



## Rapport Final de Réalisation

Document 1/3



# Geo Media Tagger



Projet n°3 : Outil Web d'annotation de textes ou d'images selon  
3 facettes (temps, espace et thème)

### Groupe

Mathieu Capdeville  
Corentin David  
Léo Dumouch

### Tuteurs et Commanditaires

Sébastien Laborie  
Christophe Marquesuzaà



## REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui nous ont aidées au cours de notre projet et en particulier :

- M. Christophe Marquesuzaà, notre tuteur et commanditaire qui s'est rendu disponible tout au long de notre projet afin de répondre à nos questions et nous donner de nombreux conseils avisés;
- M. Sébastien Laborie, notre tuteur et commanditaire, pour le temps passé à nous conseiller et à répondre à nos nombreuses questions sur les nouveaux langages à utiliser que nous n'avions pas eu l'occasion d'apprendre lors des cours dispensés à l'IUT, en particulier le RDF et le XML;
- M. Patrick Etcheverry, pour nous avoir permis d'utiliser son architecture MVC (Modèle Vue Contrôleur) ainsi pour que la documentation associée et ses conseils en PHP;
- M. Roose, pour son aide et ses conseils sur le langage PHP ;
- M. Thierry Nodenot pour son aide en JavaScript, ainsi que pour nous avoir rapidement mis en place un espace de travail sur le serveur Erozate ;
- Le groupe de projet de GeoText2Map (Marina Aguilar, Sébastien Fardilha, Romain Gérard et Guillaume Pelletier) pour la documentation ainsi que le code source fourni qui nous a permis gagner du temps sur nos phases d'analyse et de conception ;
- Mme Corine Ospital, notre professeur de communication, pour tous ses conseils concernant la rédaction de nos documents, la préparation à l'oral et la gestion de groupe ;
- M. Pierre Gastellu, notre professeur de gestion de projet, pour tous ses conseils sur l'organisation et l'avancement de notre projet ;
- Nos parents et amis pour avoir accepté de relire et de corriger notre rapport ainsi que de tester notre application.

# TABLE DES MATIERES

<b>1. Présentation du projet.....</b>	<b>5</b>
1.1 Contexte.....	6
1.2 Enjeux.....	6
1.3 Fonctionnalités.....	7
1.4 Qualité de service attendu.....	8
1.5 Limites.....	8
1.6 Utilisateurs du projet.....	9
<b>2. Analyse Fonctionnelle.....</b>	<b>10</b>
2.1 Diagramme des Cas d'Utilisation.....	10
2.2 Maquettes.....	11
2.3 Scénarios essentiels.....	14
2.4 Enchaînement des vues.....	15
<b>3. Conception et Programmation.....</b>	<b>18</b>
3.1 Diagramme des Classes.....	18
3.2 Diagramme Etats-Transitions.....	20
3.3 Framework MVC.....	21
3.3.1 Description.....	21
3.3.2 Utilisation dans le projet.....	23
3.3.3 Critiques du modèle MVC.....	23
3.4 Langages et technologies utilisés.....	24
3.4.1 Services Web.....	24
3.4.2 XHTML / CSS.....	24
3.4.3 PHP.....	25
3.4.4 JavaScript.....	25
3.4.5 JQuery.....	26
3.4.6 XML et XPath.....	26
3.4.6.1 XML.....	26
3.4.6.2 XPATH.....	27
3.4.7 RDF et SPARQL.....	27
3.4.7.1 RDF.....	27
3.4.7.2 SPARQL.....	28
3.5 APIs.....	28

3.5.1	Définition.....	28
3.5.2	CHAP Link Library .....	29
3.5.3	Google Maps API .....	29
3.5.4	RAP .....	29
3.6	Outils .....	30
3.6.1	NetBeans IDE.....	30
3.6.2	Notepad++.....	30
3.6.3	PFN .....	31
3.6.4	Suite Microsoft Office .....	31
3.6.5	Modelio .....	31
3.6.6	Pencil .....	32
3.6.7	Navigateur.....	32
3.6.8	Validateurs de code.....	33
<b>4.</b>	<b>Gestion de Projet.....</b>	<b>34</b>
4.1	Démarche de développement .....	34
4.2	Plannings prévisionnel et final .....	37
4.3	Analyse des plannings .....	38
4.4	Organisation du projet .....	38
<b>5.</b>	<b>Bilan et perspectives.....</b>	<b>39</b>
5.1	Bilan.....	39
5.2	Perspectives .....	40
<b>6.</b>	<b>Abstract .....</b>	<b>42</b>
<b>7.</b>	<b>Glossaire .....</b>	<b>43</b>
<b>8.</b>	<b>Table des Illustrations.....</b>	<b>45</b>
<b>9.</b>	<b>Webographie .....</b>	<b>46</b>

# 1. PRESENTATION DU PROJET

Notre projet, intitulé « *outil web d'annotation de texte et d'image selon trois facettes (temps, espace et thème)* », est une reprise d'un sujet traité au cours de l'année universitaire 2010 – 2011 par des étudiants de deuxième année. Il consiste en l'évolution de l'application web *GeoText2Map* (cf. figures 1 et 2), nommé ainsi par le groupe projet en 2010, et la réponse à de nouveaux besoins fonctionnels et techniques.

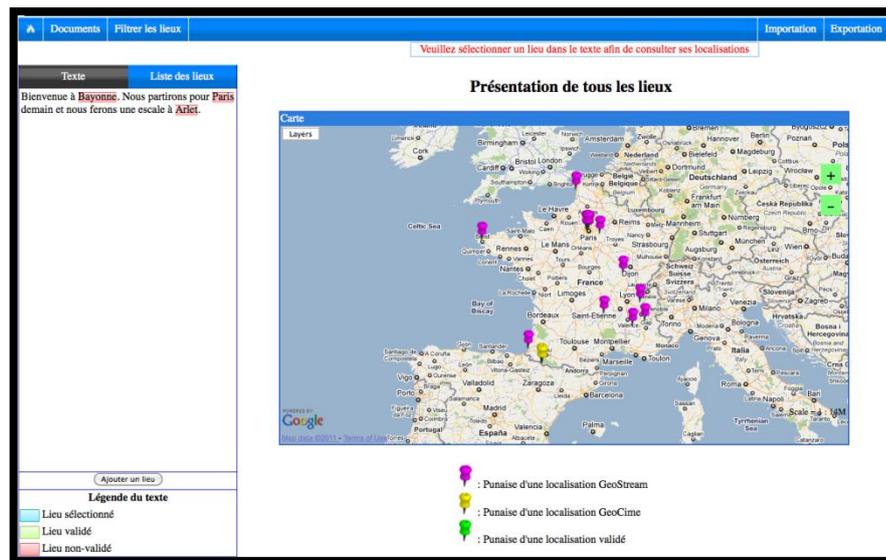


Figure 1 : Capture d'écran de GeoText2Map

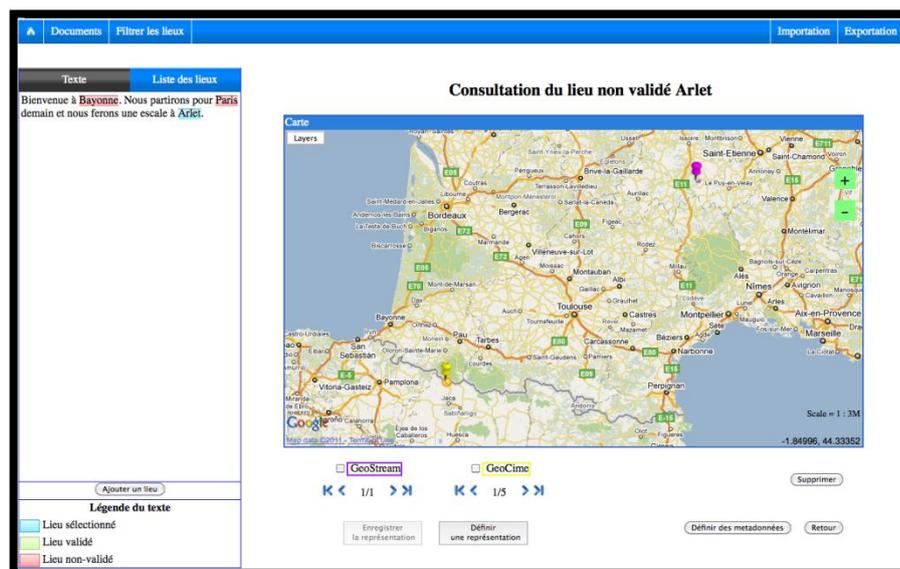


Figure 2 : Capture d'écran de GeoText2Map

## 1.1 Contexte

Ce projet s'intègre dans le cadre des recherches du Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour (LIUPPA), plus précisément, l'équipe de recherche T2i (domaine : Traitement des Informations spatiales, temporelles et thématiques pour l'adaptation de l'Interaction au contexte et à l'utilisateur), dont le responsable est M. Thierry NODENOT. Cette équipe est organisée en sous-groupes, dont le but est de récolter, traiter et valoriser des données spatio-temporelles.

Notre projet est dans la continuité de la valorisation des données spatio-temporelles de l'application existante GeoText2Map.

Nos tuteurs et commanditaires M. Christophe MARQUESUZAA et M. Sébastien LABORIE, enseignants chercheurs au sein de l'équipe T2i souhaitent que nous reprenions cette application dont les besoins ont évolués, suite au départ de l'entreprise partenaire GeoCime et à l'évolution des travaux de recherches depuis 2010.

## 1.2 Enjeux

Un des nouveaux gains attendu pour la reprise de ce projet est de valoriser les nouveaux services développés par l'équipe T2i. Depuis le projet datant de 2010, l'équipe a créé de nouveaux services que nous devons, en améliorant l'application existante, valoriser au maximum.

De plus, nous ne devons plus utiliser la base de données fournie par GeoCime car les commanditaires ne traitent plus avec cette entreprise industrielle.

Les enjeux du projet d'aujourd'hui ont donc changé par rapport à ceux de 2010, notamment par l'évolution de la capacité et du mode de traitement des informations par l'équipe T2i (regroupement des services sur un seul serveur, GeoPot). Nous devons donc assurer la continuité de la documentation et faciliter la maintenance de l'application.

### 1.3 Fonctionnalités

Nous devons à partir de l'outil GeoText2Map, imaginer, concevoir et implémenter une extension de ce dernier qui permettra d'associer à des morceaux de texte des annotations de type temporel (une date précise, une certaine période...), spatial (un lieu précis, autour d'un lieu, au nord d'un lieu etc...) et thématique (sport, ville, ami...). Il conviendra de visualiser respectivement ces annotations sur une frise temporelle, une carte ou une zone de texte.

Sur le même principe, permettre des annotations de type temporel, spatial et thématique sur des images fixes (cette photo a été prise tel jour à tel lieu...).

Toujours dans l'esprit de GeoText2Map, aider un utilisateur à annoter des textes ou des images. Par exemple pour un texte, en invoquant différents services web de reconnaissance automatique de dates ou de lieux, des suggestions d'annotations pourraient être proposés à l'utilisateur pour qu'il puisse les affiner et les valider. Pour des images, exploiter les annotations d'autres utilisateurs semble une piste intéressante à envisager.

L'évolution des fonctionnalités contient par exemple la suppression de l'utilisation de GeoCime et de GeoStream au détriment de GeoPot, un serveur qui propose trois services : deux qui reconnaissent dans un texte les entités spatiales relatives (ESR, par exemple « au nord de Bayonne » ou encore « autour d'Anglet ») et les entités spatiales absolues (ESA, par exemple « à Bayonne » ou encore « à Pau ») et un autre qui reconnaît de la même manière des entités calendaires relatives et absolues (EC, par exemple « le 14 juin 2012 » ou encore « au printemps 2008 »).

Un autre service est en développement par l'équipe de recherche. Celui-ci permet de reconnaître des thèmes (ET) présents dans un texte comme par exemple la botanique.

Ce projet n'a pas pour l'instant une finalité précise, le but est de faire fonctionner le principe, c'est un projet essentiellement lié à la recherche du groupe de recherche T2i. Nous pouvons cependant imaginer une utilité pédagogique de ce principe pour le futur.

## 1.4 Qualité de service attendu

Notre projet étant une application web, il est nécessaire que le code produit soit au maximum valide selon les standards du web (W3C).

De plus, il faut que nous produisions un code le plus modulaire possible afin que, sur le long terme, l'application puisse évoluer pour traiter différents types de médias.

Nous devons respecter des spécifications précises dans le cadre d'un travail de recherche. Par exemple la cohérence du résultat donné n'est pas prioritaire, c'est-à-dire que nous ne devons pas prendre en compte le fait que le service puisse indiquer un lieu différent de celui que l'utilisateur a préalablement saisi.

Il n'est pas non plus nécessaire de gérer le fait que plusieurs utilisateurs travaillent sur l'application simultanément.

De plus, nous n'avons pas besoin de prévoir un moyen de sécurisation tel qu'une page d'authentification.

## 1.5 Limites

Une des premières limites est que nous ne pouvons pas interagir directement avec les différents services web présents sur le serveur GeoPot. C'est-à-dire que nous n'avons pas la possibilité de modifier quoi que ce soit sur les fonctionnalités ou les performances de ces services.

Ce projet présente aussi des limites techniques, en effet, nous sommes dépendant du serveur sur lequel les services web sont hébergés. De plus, nous ne pouvons pas développer un tel projet sur nos machines car il nécessite un serveur qui supporte certains services (openssl, curl, SOAP ...).

## 1.6 Utilisateurs du projet

Tout d'abord, notre application ne gère pas les utilisateurs. C'est-à-dire que si une personne l'utilise, une autre personne interagissant avec créera de nombreux dysfonctionnements. Ensuite, il n'y a pas de cible particulière, tout le monde peut se servir de Geo Media Tagger, aussi bien une personne qui rentre de vacances et qui souhaite saisir son carnet de voyage, ou encore un chercheur qui voudrait informatiser des résultats, ou bien un enseignant qui voudrait apprendre à ses élèves, ou encore des agents de police qui souhaiteraient répertorier les lieux et dates de faits de manière informatique pour ensuite par exemple établir des statistiques. De plus, les enseignants chercheurs ayant développés les services web que nous utilisons pourront utiliser notre application pour des démonstrations afin de valoriser leurs travaux de recherche.

## 2. ANALYSE FONCTIONNELLE

---

Cette partie, ainsi que la suivante montrent l'analyse technique qui a été effectuée pour réaliser notre projet. Ce sont des extraits du Cahier Des Charges Fonctionnel et du Cahier Des Charges Technique disponibles en intégralité en Annexes (cf. Annexes 3. et 4.).

### 2.1 Diagramme des Cas d'Utilisation

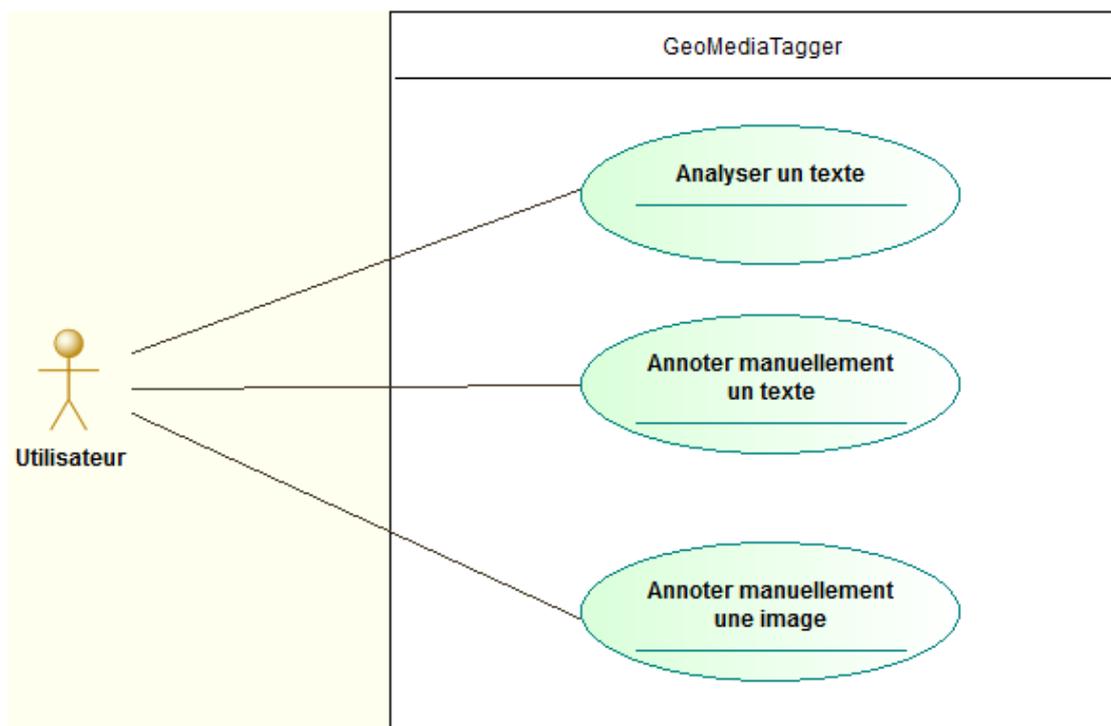


Figure 3 : Diagramme des Cas d'Utilisation

Un utilisateur peut utiliser l'application afin soit d'analyser un texte (saisie manuellement, importé depuis un fichier « .txt » présent sur son ordinateur, ou un fichier « .txt » présent sur un site), soit d'annoter manuellement une image ou bien un texte en projetant sur une carte et sur une frise chronologique ces annotations.

## 2.2 Maquettes



Figure 4 : Maquette de l'accueil

Cette maquette est la page d'accueil de notre application, elle permet à l'utilisateur de choisir entre lancer l'application à partir d'un texte, d'une image ou encore d'importer une sauvegarde. Le texte au-dessus explique brièvement comment utiliser l'application.

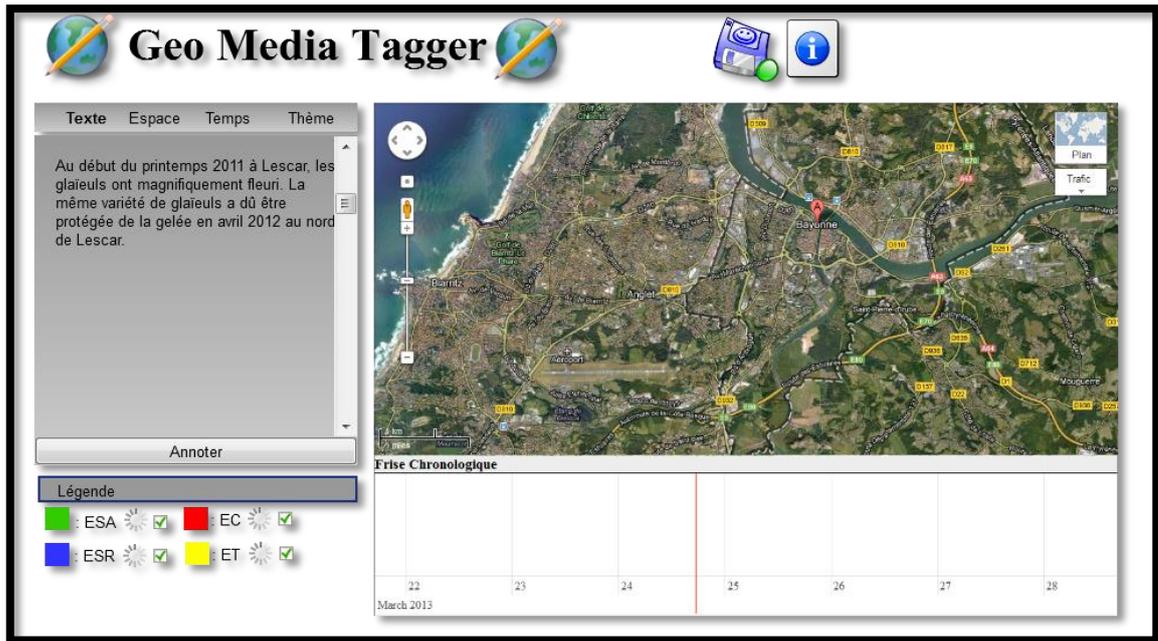


Figure 5 : Maquette de l'écran principal

Cette maquette apparaît lorsque l'utilisateur a lancé l'application à partir d'un texte. Il a ensuite plusieurs choix qui s'offrent à lui, il peut appeler le service web gérant les Entités Spatiales Absolues (ESA dans la légende). Il peut faire de même avec les Entités Spatiales Relatives (ESR dans la légende) ou encore avec les Entités Calendaires (EC dans la légende). Les Entités Thématiques (ET dans la légende) peuvent également être visualisées.

En cliquant sur le bouton « Annoter » l'utilisateur peut également annoter manuellement une partie du texte qu'il a sélectionné au préalable. Une fois cliqué sur le bouton, une fenêtre bloquante apparaît en demandant si l'on veut annoter un espace, un temps ou un thème.

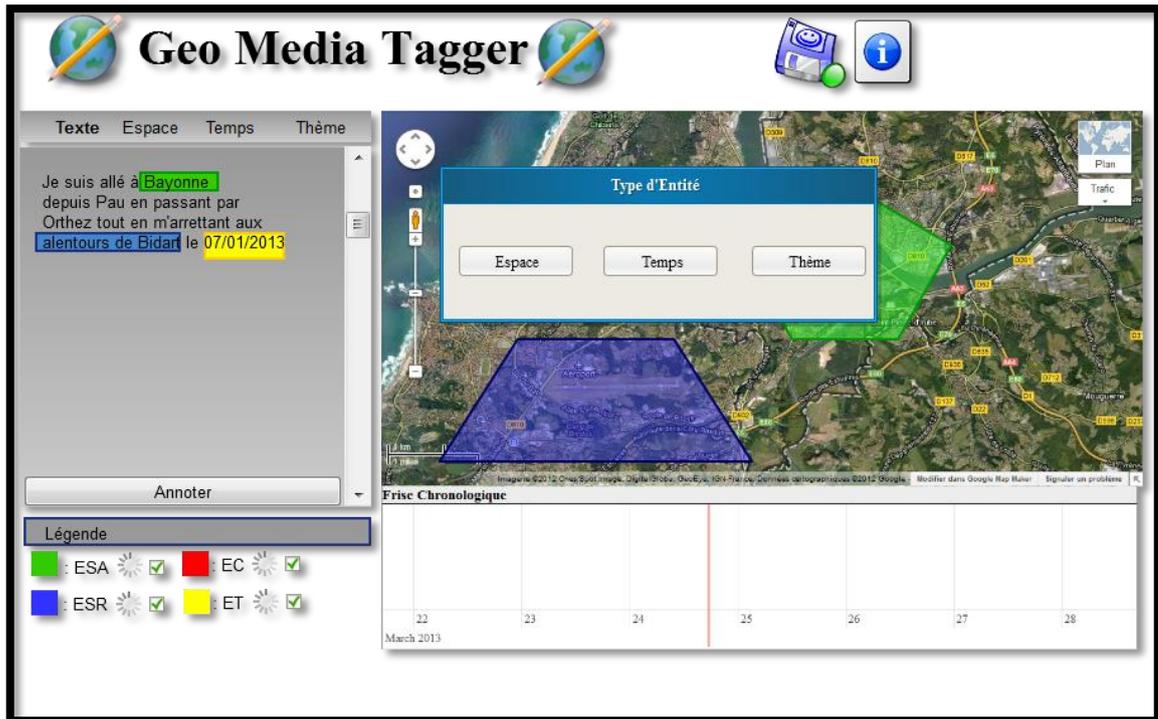


Figure 6 : Maquette de l'écran principal avec annotation

Une fois le choix fait l'utilisateur n'a plus qu'à cliquer sur la carte, la frise ou encore saisir son thème en fonction du choix précédent. S'il a choisi d'annoter de l'espace l'utilisateur pourra également choisir entre annoter un point, une ligne ou un polygone.

La disquette en haut permet de sauvegarder le travail fait afin de le reprendre plus tard. L'icône d'information « i » permet de visualiser le « à propos » de l'application.

En cliquant sur « Geo Media Tagger » en haut l'utilisateur peut retourner à la page d'accueil de l'application.

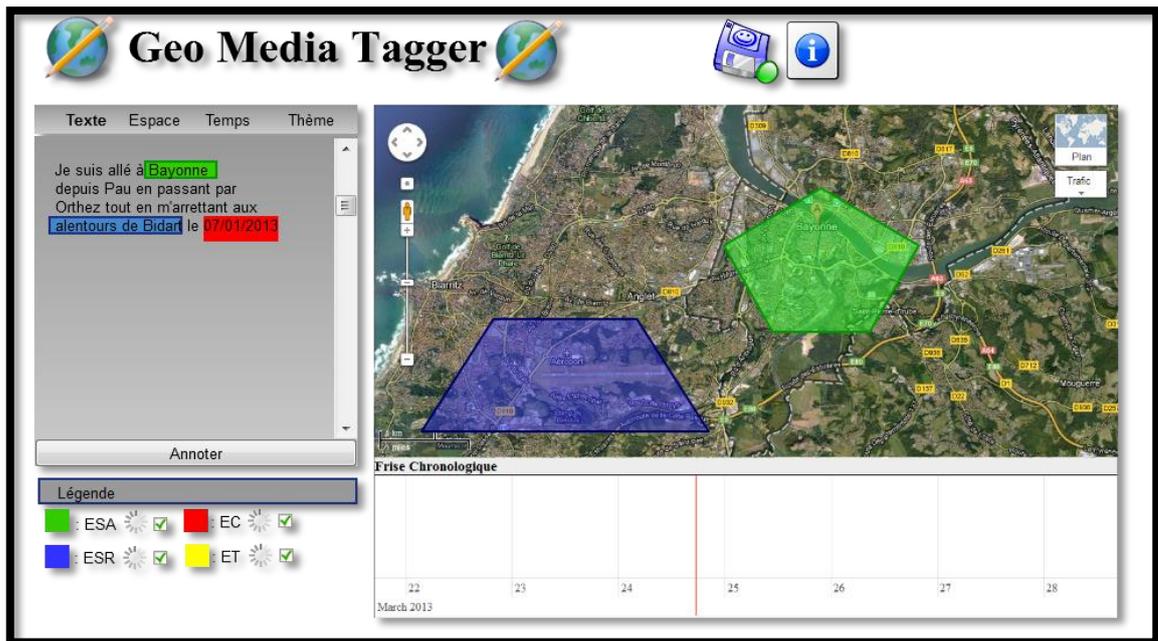


Figure 7 : Maquette présentant les entités sur la carte

Cette maquette représente notre application lorsque l'utilisateur a lancé les différents services Web. Les Entités Spatiales Absolues et Relatives sont représentées sur la carte. On peut voir dans le texte la correspondance entre les parties du texte et les services correspondants.

## 2.3 Scénarios essentiels

### Scénario du cas d'utilisation « analyser un texte »

#### Sommaire d'identification :

**Titre :** Afficher les Entités Spatiales Absolues

**Type :** réel

**Résumé :** L'utilisateur lance l'application à partir d'un texte saisi manuellement, et invoque les services pour afficher les ESA de son texte

**Acteur :** un utilisateur

**Date de création :** 12/02/2013

**Date de mise à jour :** /

**Version :** 1.0

**Responsables :** Capdeville Mathieu, David Coarentin, Dumouch Léo

**Pré-condition :** L'utilisateur a choisi de lancer l'application à partir d'un texte.

**Scénario nominal :**

- 1- Le système renvoie une page en donnant plusieurs choix à l'utilisateur.
- 2- L'utilisateur choisi « Saisie Manuelle »
- 3- Le système affiche une zone de texte ainsi que 3 boutons en dessous de celle-ci.
- 4- L'utilisateur clique sur le bouton « Exemple »
- 5- Le système affiche dans la zone de texte un texte exemple prédéfini.
- 6- L'utilisateur clique sur « Lancer l'application »
- 7- Le système renvoie une page avec le texte choisit ainsi qu'une carte et une frise. Il est en attente d'une action de l'utilisateur
- 8- L'utilisateur choisit de lancer le service de repérage d'Entités Spatiales Absolues (ESA)
- 9- Le système recharge la page en affichant dans l'onglet « Espace » les différentes ESA repérées.
- 10- L'utilisateur coche la case à cocher
- 11- Le système affiche sur la carte les différentes ESA se trouvant dans l'onglet « Espace ».

## 2.4 Enchaînement des vues

- 1- Le système renvoie une page en donnant plusieurs choix à l'utilisateur.
- 2- L'utilisateur choisi « Saisie Manuelle »



Figure 8 : Maquette d'accueil pour saisir un texte

- 3- Le système affiche une zone de texte ainsi que 3 boutons en dessous de celle-ci.
- 4- L'utilisateur clique sur le bouton « Exemple »
- 5- Le système affiche dans la zone de texte un texte exemple prédéfini.
- 6- L'utilisateur clique sur « Lancer l'application »



Figure 9 : Maquette de saisie manuelle d'un texte

- 7- Le système renvoie une page avec le texte choisit ainsi qu'une carte et une frise. Il est en attente d'une action de l'utilisateur
- 8- L'utilisateur choisit de lancer le service de repérage d'Entités Spatiales Absolues (ESA)



Figure 10 : Maquette de l'écran principal avec un texte

- 9- Le système recharge la page en affichant dans l'onglet « Espace » les différentes ESA repérées.
- 10- L'utilisateur coche la case à cocher

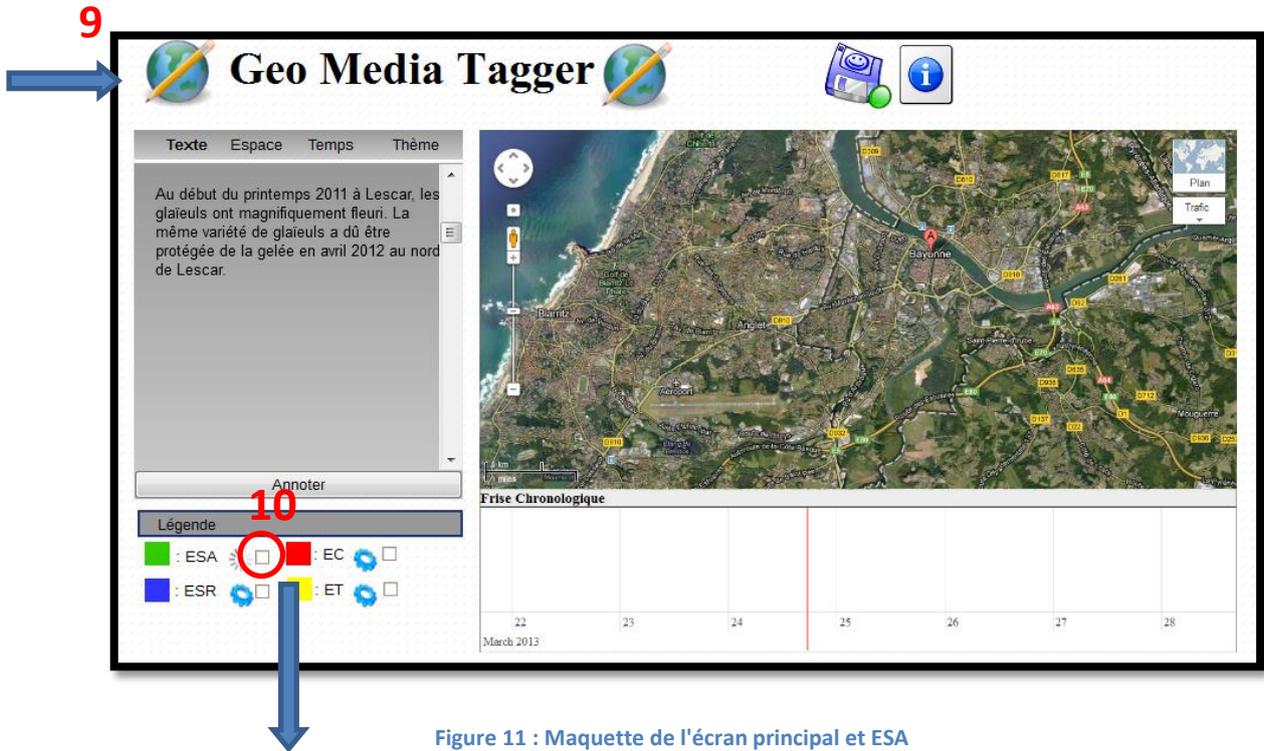


Figure 11 : Maquette de l'écran principal et ESA

- 11- Le système affiche sur la carte les différentes ESA se trouvant dans l'onglet « Espace ».

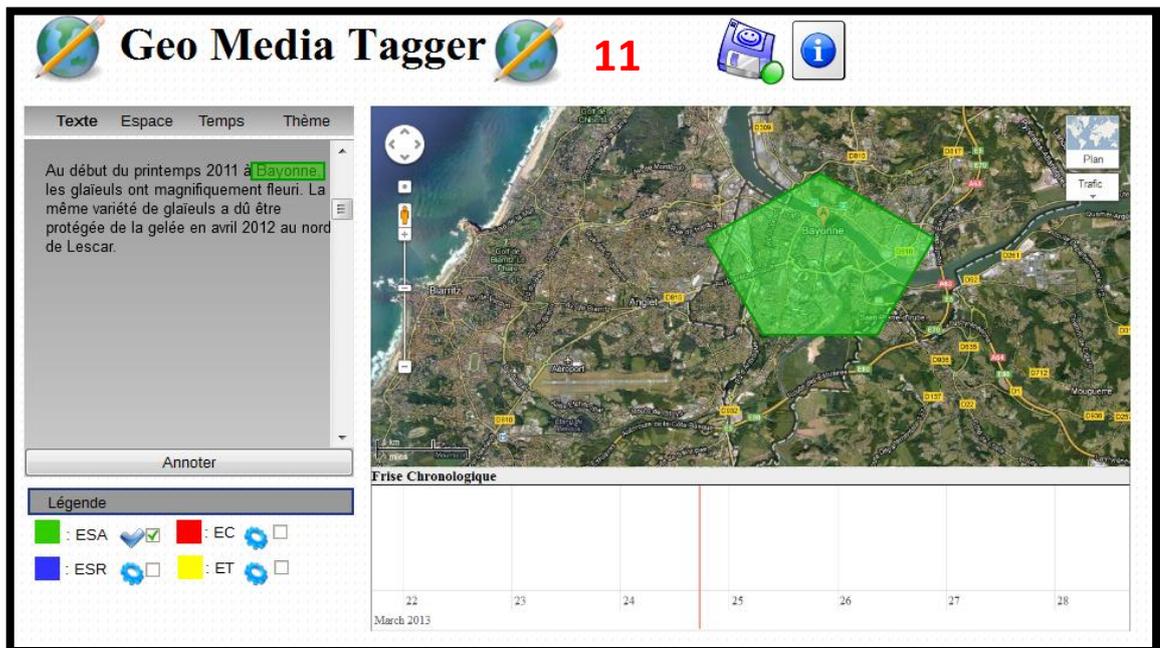


Figure 12 : Maquette présentant une ESA

## 3. CONCEPTION ET PROGRAMMATION

### 3.1 Diagramme des Classes

Dans cette partie, nous montrerons et commenterons le diagramme des classes représentant l'application telle qu'elle sera lors du rendu de ce rapport. Nous avons fait un diagramme des classes pour pouvoir coder en approche orienté objet, pour simplifier la reprise de notre code.

Ce diagramme représente l'état final de notre application, comme vous pouvez le voir, la classe Media est au centre de notre application. C'est elle qui va gérer la plupart des données de l'application.

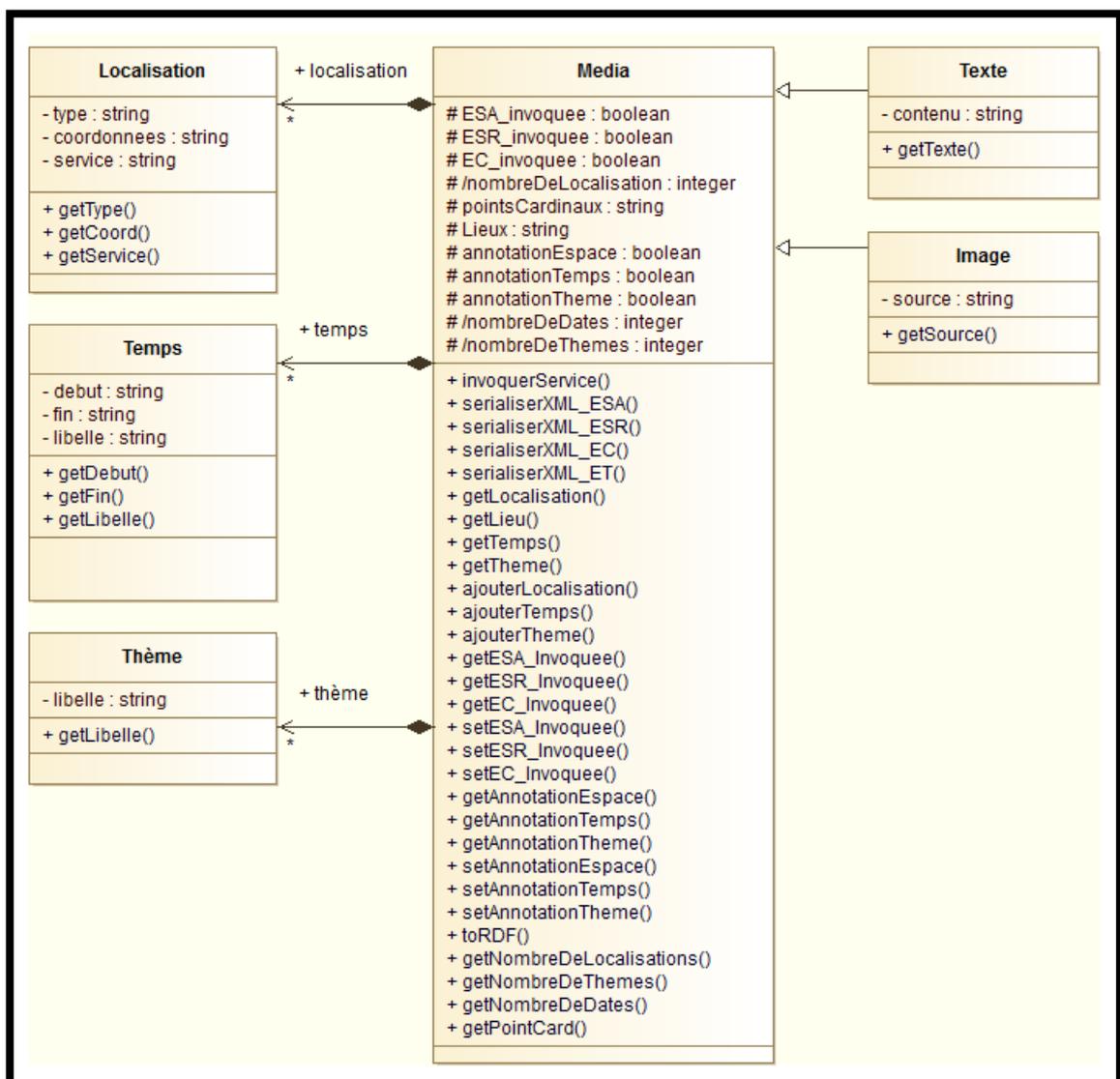


Figure 13 : Diagramme des classes au niveau logique

La classe Image et la classe Texte héritent de la classe Media, la classe Texte a un attribut qui est le contenu en plus de ceux de la classe Media. La classe Image, elle, contient une source.

Ce modèle UML est cependant susceptible d'évoluer durant la semaine de programmation.

La classe Media est composée de, au minimum 0 et au maximum plusieurs Localisation(s). Cette dernière contient un type (POINT, POLYGON, MULTIPOLYGON ...), les coordonnées de la localisation, le service web utilisé (ESA, ESR, ...) ainsi que le nom du lieu correspondant.

La classe Temps a la même composition que Localisation sur la classe Media. Elle contient la date. Il n'est pas nécessaire de préciser le service web vu qu'il n'y en a qu'un seul reconnaissant les entités calendaires.

La classe Thème a également la même composition sur la classe Media que les deux précédentes. Elle contient juste le libellé représentant la chaîne de caractères contenant le thème choisi.

Ce diagramme peut cependant être amélioré, nous n'avons pas eu le temps de l'optimiser au maximum mais nous avons cependant établi un diagramme de classes pouvant donner des perspectives pour cette application (cf : Bilan et perspectives).



## 3.3 Framework MVC

### 3.3.1 Description

Étant donné que le projet que nous reprenons a été développé grâce au Framework<sup>1</sup> MVC (Modèle Vue Contrôleur) de M. Patrick Etcheverry, nous avons continué de l'utiliser.

MVC a été mis au point en 1979 par un informaticien norvégien, Trygve Reenskaug. Le MVC est un motif de conception ou patron (modèle de solution réutilisable permettant de répondre à un type de problème récurrent) utilisable pour développer de nombreuses applications logicielles. Ce modèle recommande la séparation entre les données, les traitements et la présentation. On distingue trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.

Chaque entité a un rôle précis que nous décrivons ci-dessous :

- Le **Modèle** contient la logique métier de l'application c'est-à-dire les données et les traitements. Il regroupe les données de l'application (stockées dans des fichiers ou des bases de données), les opérations de gestion des données (ajout, modification, suppression, recherche, ...) ainsi que des opérations métiers (déclenchées par l'utilisateur à l'aide d'éléments d'interface présents dans la couche vue). Les résultats renvoyés par la couche modèle sont dépourvus de toute présentation
- La **Vue** correspond à l'interface présentée à l'utilisateur. Elle regroupe des informations sur la structure de l'interface (par exemple disposition des éléments) et parfois des données issues du modèle et des éléments d'interface. La vue n'effectue aucun traitement et permet à l'utilisateur de déclencher les opérations métiers de l'application.
- Le **Contrôleur** joue le rôle de chef d'orchestre et contrôle la dynamique de l'application. Il analyse la requête de l'utilisateur et se contente d'appeler le modèle adéquat puis de renvoyer la vue correspondante à la demande après en avoir analysé la cohérence. Il n'effectue aucun traitement, ne modifie aucune donnée et il suit toujours le même algorithme :
  1. Réceptionner la demande de l'utilisateur (par exemple, un clic sur un bouton).

---

<sup>1</sup> Framework : espace de travail modulaire qui regroupe un ensemble de bibliothèques et de conventions permettant le développement rapide d'applications

2. Analyser la cohérence de la demande en fonction de l'état courant de l'application.
3. Déclencher l'action métier du modèle demandée ou l'action qui gère les cas d'un enchaînement invalide si la demande n'a pas lieu d'être.
4. Récupérer le résultat de l'action exécutée.
5. Construire et présenter la vue réponse à l'utilisateur.

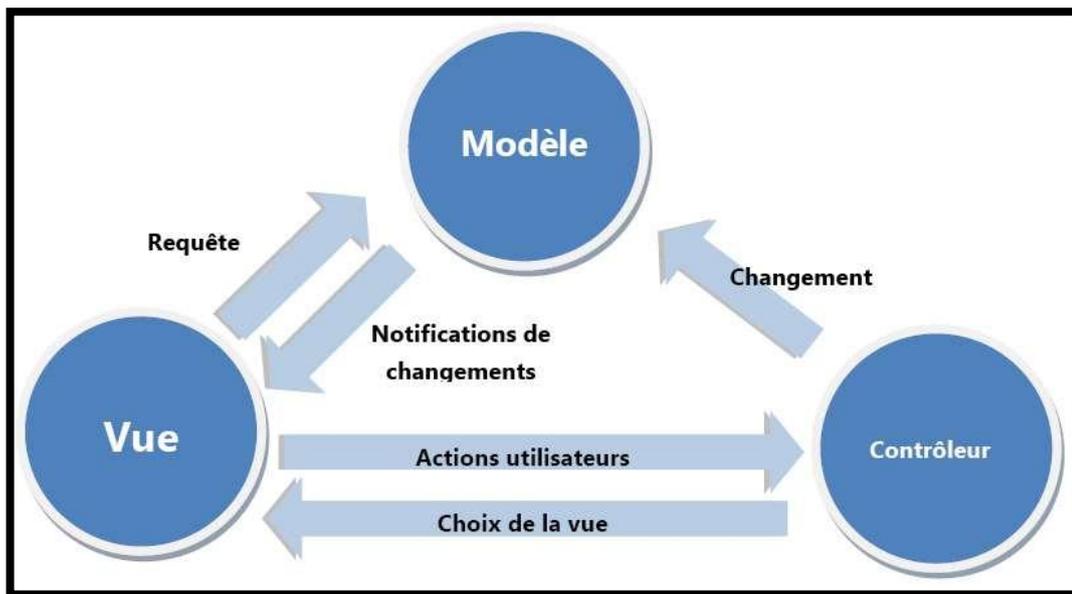


Figure 15 : Fonctionnement du MVC

Ainsi, le MVC simplifie la tâche du développeur lorsqu'il souhaite effectuer la maintenance de son site ou le faire évoluer, si toutefois il existe une documentation associée.

### 3.3.2 Utilisation dans le projet

Ce Framework MVC nous a permis d'une part de bien organiser notre code en dossiers et sous dossiers, d'autre part, il nous a facilité l'intégration des fonctionnalités développées lors des phases de prototypage.

Voici un exemple du fonctionnement du Framework lors d'une demande de projection d'un texte saisi.

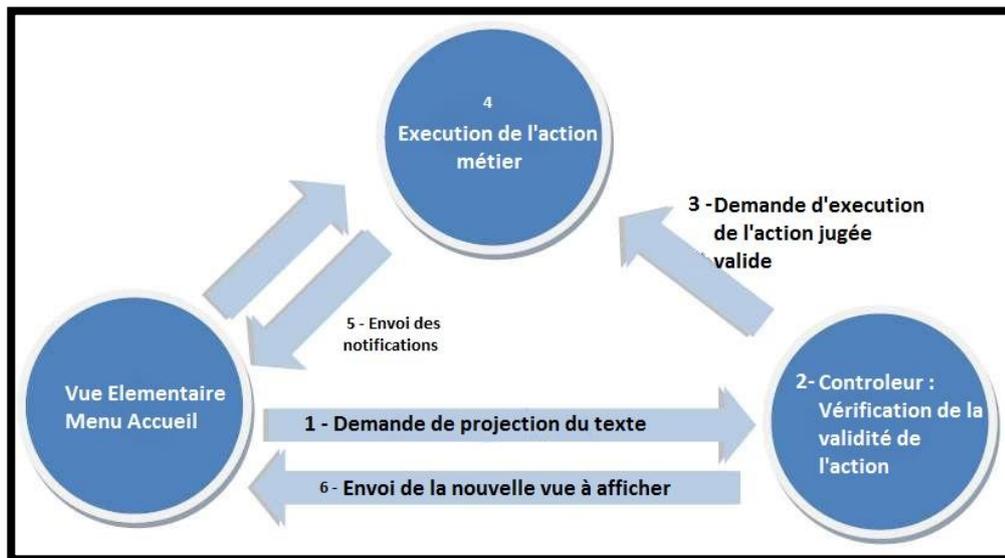


Figure 16 : Exemple de fonctionnement du MVC

### 3.3.3 Critiques du modèle MVC

L'utilisation du modèle MVC nous a paru être le meilleur choix car nous l'avons déjà utilisé lors des cours de DUT informatique durant le semestre 3. Nous avons donc eu une certaine facilité à l'utiliser vu que nous le connaissions déjà. Cependant nous avons remarqué quelques limites de ce Framework, notamment si l'on veut utiliser la technologie AJAX.

## 3.4 Langages et technologies utilisés

### 3.4.1 Services Web

Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et de manière synchrone. (Source : [http://fr.wikipedia.org/wiki/Service\\_Web](http://fr.wikipedia.org/wiki/Service_Web)). Pour notre projet, nous avons utilisé différents services web développés par l'équipe de recherche T2i.

- Le service ESA (Entités Spatiales Absolues) reconnaît dans un texte les lieux cités de manière absolue (par exemple : à Bayonne, les plages d'Anglet ...).
- Le service ESR (Entités Spatiales Relatives) qui reconnaît de la même manière les lieux cités dans un texte mais cette fois de manière relative c'est-à-dire par rapport à un point cardinal ou un adjectif de positionnement (par exemple : au nord de Lescar, aux alentours de Bayonne ...).
- Le service EC (Entités Calendaires) reconnaît lui les dates et périodes citées dans un texte. Il reconnaît autant les entités absolues (par exemple une date : le 16 mars 2013) mais aussi les entités relatives (au début des années 1970, avant l'hiver 2012 ...).

### 3.4.2 xHTML / CSS

 xHTML (*eXtensible HyperText Markup Language*) est un langage de balisage servant à écrire des pages pour le Web. Il est le successeur d'HTML dont le but est d'afficher et de formater du texte. Nous l'avons utilisé pour afficher sur une page Web, toutes les informations relatives à l'application (des images, des champs de texte...). Nous l'avons utilisé pour structurer nos pages web.

 CSS (*Cascading Style Sheets*), est un langage qui sert à décrire la présentation des documents HTML et XML. Il permet d'appliquer un style aux pages xHTML et de gérer séparément la forme du contenu. Nous l'avons utilisé pour structurer la page faite en xHTML, c'est-à-dire positionner les éléments où on le souhaite. Le CSS nous a permis de mettre en forme le contenu structuré en utilisant le xHTML.

### 3.4.3 PHP



*HyperText Preprocessor*, plus connu sous son sigle PHP (Acronyme récursif), est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes. En raison de la richesse de sa bibliothèque, on désigne parfois PHP comme une plate-forme plus qu'un simple langage. (Source : <http://fr.wikipedia.org/wiki/PHP>). Nous avons utilisé le PHP pour gérer les actions entre les différentes pages de l'application, et pour automatiser les tâches.

### 3.4.4 JavaScript



Quelques fois abrégé JS, JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. (Source : <http://fr.wikipedia.org/wiki/JavaScript>). Le Javascript nous a permis entre autre d'utiliser diverses API qui seront citées plus loin dans le rapport. De plus, ce langage nous a permis de dynamiser nos interfaces en utilisant la librairie JQuery citée au paragraphe suivant.

### 3.4.5 JQuery



JQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006 (Source : <http://fr.wikipedia.org/wiki/JQuery>). Nous avons utilisé JQuery pour afin de rendre nos interfaces dynamiques. Par exemple les onglets comprenant le texte saisie, les lieux, les dates ainsi que les thèmes sont interchangeables grâce à cette technologie. C'est aussi en utilisant JQuery que l'application permet de masquer la carte, la frise ou les deux ainsi que de rendre le menu de la page d'accueil si dynamique.

### 3.4.6 XML et XPath

#### 3.4.6.1 XML



L'*Extensible Markup Language* (XML, « langage de balisage extensible » en français) est un langage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (< >) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes (interopérabilité) (Source : <http://fr.wikipedia.org/wiki/XML>). Nous avons utilisé le XML pour récupérer les valeurs envoyées par le service GeoPot, et pour pouvoir les manipuler en utilisant un langage de manipulation de données au format XML, le XPATH.

### 3.4.6.2 XPATH



XPath est un langage (non XML) pour localiser une portion d'un document XML. Initialement créé pour fournir une syntaxe et une sémantique aux fonctions communes à XPointer et XSL, XPath a rapidement été adopté par les développeurs comme langage d'interrogation simple d'emploi. (Source : <http://fr.wikipedia.org/wiki/XPath>). Nous avons utilisé le langage XPath pour retrouver les valeurs souhaitées dans le fichier XML retourné par le service GeoPot. Le groupe projet de 2010 a développé un algorithme qui permet de parcourir (parser) toute l'arborescence d'un document XML. Ceci est un inconvénient si la structure XML est modifiée. En effet, il est nécessaire de modifier tout ou partie de leur algorithme. Alors que dans notre cas il n'y a qu'à modifier une requête.

### 3.4.7 RDF et SPARQL

#### 3.4.7.1 RDF



*Resource Description Framework* (RDF) est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées, de façon à permettre le traitement automatique de telles descriptions. Développé par le W3C, RDF est le langage de base du Web sémantique. L'une des syntaxes (ou sérialisations) de ce langage est RDF/XML. D'autres sérialisations de RDF sont apparues ensuite, cherchant à rendre la lecture plus compréhensible, c'est le cas par exemple de Notation3 (ou N3) (Source : [http://fr.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://fr.wikipedia.org/wiki/Resource_Description_Framework)). Nous avons utilisé le RDF pour permettre à l'utilisateur de sauvegarder son travail, en effet, en utilisant RDF nous avons pu créer des fichiers qui décrivent l'application à un moment donné. Ainsi, l'utilisateur enregistrer, transmettre, et reprendre un travail sur l'application.

### 3.4.7.2 SPARQL



SPARQL est au RDF ce qu'XPath est au XML. Prononcé *sparkle* en anglais « étincelle » est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet. Son nom est un acronyme qui signifie SPARQL Protocol and RDF Query Language. SPARQL est l'équivalent de SQL car comme en SQL, on accède aux données d'une base de données via ce langage de requête alors qu'avec SPARQL, on accède aux données du Web des données. Cela signifie qu'en théorie, on pourrait accéder à toutes les données du Web avec ce standard (Source : <http://fr.wikipedia.org/wiki/SPARQL>). Nous avons utilisé SPARQL pour pouvoir manipuler les informations dans les fichiers RDF. De même que pour la technologie XPath, SPARQL nous a permis de créer un code modulaire en cas de modification de la structure de donnée de l'application. Modifier quelques requêtes est plus simple que de parcourir entièrement le document. Il nous permet de pouvoir rétablir l'application à partir d'un fichier RDF préalablement généré.

## 3.5 APIs

### 3.5.1 Définition

Une interface de programmation (*Application Programming Interface* ou API) est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres, de manière analogue à une interface homme-machine, qui rend possible l'interaction entre un homme et une machine. Du point de vue technique, une API est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service. La connaissance des API est indispensable à l'interopérabilité entre les composants logiciels.

(Source : [http://fr.wikipedia.org/wiki/Interface\\_de\\_programmation](http://fr.wikipedia.org/wiki/Interface_de_programmation)).

### 3.5.2 CHAP Link Library



CHAP Link Library est une bibliothèque de visualisation basé sur le Web pour l'affichage des graphiques, de réseaux et d'échéanciers. Les outils sont développés comme les graphiques de visualisation de Google pour JavaScript. CHAP Link Library est développé par Almende dans le cadre du CHAP (Common Hybrid Agent Platform). Nous avons utilisé CHAP pour gérer la frise chronologique de notre application. Cette API nous a permis, à l'aide du langage Javascript, de générer une frise chronologique et d'y ajouter des éléments extraits du fichier résultat XML que nous fournit le service correspondant. (<http://almende.github.com/chap-links-library/>).

### 3.5.3 Google Maps API



Google Maps est une API de Google permettant de géolocaliser des adresses sur une carte à l'aide de sa latitude et de sa longitude. Cet api permet de localiser tout type de données sur une carte (routière, satellite, mixte) à partir de son adresse postale. Cet API s'avère très utile pour proposer aux internautes une vision globale et géographique de données (membre d'une communauté, restaurants d'un quartier...). Les résultats sur la carte apparaissent sous la forme d'un picto (petite icône) cliquable (Source : <http://www.dicodunet.com/definitions/google/api-google-map.htm>). De la même manière que l'API CHAP, cette API nous a permis de mettre en valeur sur un fond cartographique les données renvoyé par les services ESA et ESR. (<https://developers.google.com/maps/?hl=fr>).

### 3.5.4 RAP

RAP est une API permettant de *parser*, d'effectuer des requêtes, de manipuler et de sérialiser des modèles de données au format RDF en utilisant le langage PHP. Cette API nous permet de générer les fichiers de sauvegarde de l'application au format RDF. Nous l'avons utilisé car elle propose de nombreuses fonctions permettant de passer de variables et objets PHP à un document au format RDF en créant des triplets (Le fonctionnement de cette API et du RDF sera détaillé dans les annexes) (<http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/>).

## 3.6 Outils

### 3.6.1 NetBeans IDE



NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. (Source : <http://fr.wikipedia.org/wiki/NetBeans>)

### 3.6.2 Notepad++



NotePad++ est un éditeur de texte générique codé en C++, qui intègre la coloration syntaxique de code source pour les langages et fichiers C, C++, Java, C#, XML, HTML, PHP, JavaScript, makefile, art ASCII, doxygen, .bat, ASP, Visual Basic/VB Script, SQL, Objective-C, CSS, Pascal, Perl, Python, R, MATLAB, Lua, TCL, Assembleur, Ruby, Lisp, Scheme, Properties, Diff, Small talk, PostScript et VHDL ainsi que pour tout autre langage informatique, car ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque. Ce logiciel, basé sur la composante Scintilla, a pour but de fournir un éditeur léger (aussi bien au niveau de la taille du code compilé que des ressources occupées durant l'exécution) et efficace. Il est également une alternative au bloc-notes de Windows (d'où le nom). Le projet est sous licence GPL.

(Source : <http://fr.wikipedia.org/wiki/Notepad%2B%2B>).

### 3.6.3 PFN



Le PFN est un script PHP open source permettant de créer un explorateur de fichiers sur votre serveur. Geo Media Tagger ne pouvant pas être développé en local nous avons dû demander un emplacement sur le serveur Erozate, c'est là que nous avons dû utiliser l'outil PFN pour ajouter, modifier ou supprimer les fichiers codes sources. Nous avons cependant remarqué que cet outil n'est pas très bien adapté à un grand nombre manipulations. Il faut en effet ajouter les pages de code une par une, ce que rend la mise à jour de l'application assez longue.

### 3.6.4 Suite Microsoft Office



Microsoft Office Word (2010) est un logiciel de traitement de texte. Nous l'avons utilisé lors de la rédaction de tous les documents écrits que nous avons eu à produire.



Microsoft Office Powerpoint (2010) est un logiciel de présentation. Nous l'avons utilisé pour créer les diaporamas nécessaires pour présentations orales.



Microsoft Office Excel (2010) est un tableur. Ce logiciel nous a servi pour faire les différents plannings prévisionnels durant toute la durée du projet ou encore pour le planning final.

### 3.6.5 Modelio



Successeur de l'atelier Objecteering, Modelio est un outil de modélisation UML ou AGL (atelier de génie logiciel). Il nous a permis de réaliser les différents diagrammes UML durant la phase d'analyse.

### 3.6.6 Pencil



Pencil est un logiciel libre, distribué sous la licence GPL version 2. Il est disponible pour toutes les plates-formes que Firefox 3 peut exécuter. Il nous a permis de réaliser les différentes maquettes de l'application.

### 3.6.7 Navigateur



Mozilla Firefox est un navigateur internet personnalisable basé sur le moteur d'affichage libre de pages Web Gecko. Ce navigateur contient également un plug-in très intéressant lors de phase de programmation ou de test, *Firebug*, que nous avons également utilisé.

### 3.6.8 Validateurs de code

Nous avons aussi utilisé des outils, qui à partir d'un code, renvoient une validation, ou non, de celui-ci. Cela nous a permis de voir au fur et à mesure si le code que nous produisons est valide selon les standards W3C<sup>2</sup> (*World Wide Web Consortium*).

Par exemple nous avons utilisé le service suivant : <http://www.w3.org/RDF/Validator/> afin de voir si le schéma RDF produit pour la sauvegarde de l'application est valide.

Notre schéma étant valide, le service nous a généré un graphe qui représente notre application. En voici en extrait (figure 7) :

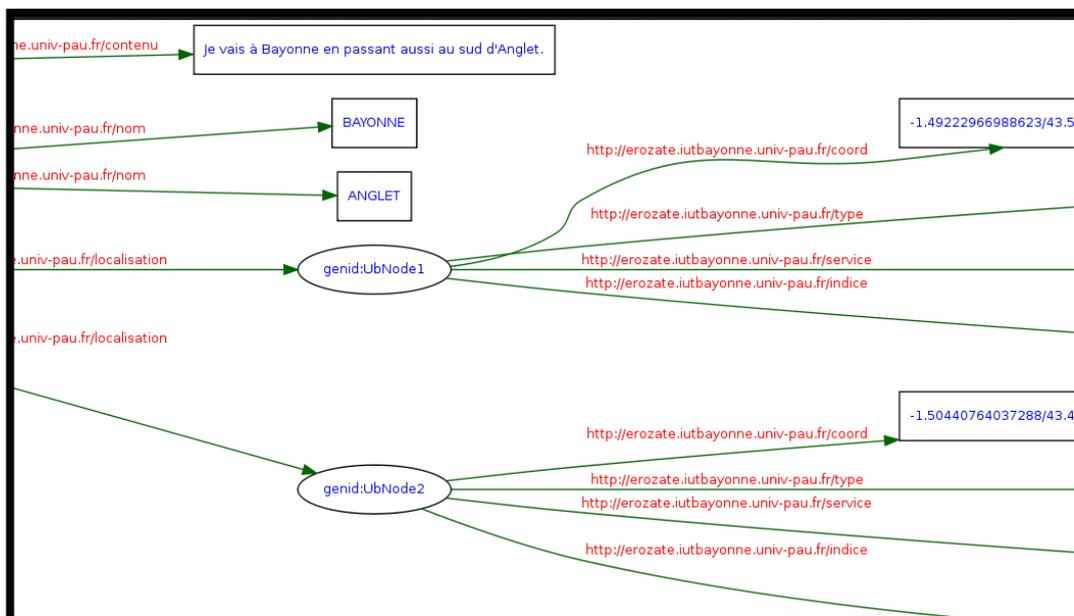


Figure 17 : Extrait du graphe RDF permettant l'export de l'application

D'autres outils permettent de valider du code XHTML ou encore du CSS, tel que le validateur W3C.

<sup>2</sup> W3C : organisme de normalisation à but non-lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, RDF, SPARQL, CSS...

## 4. GESTION DE PROJET

---

### 4.1 Démarche de développement

Dès les premières réunions avec nos commanditaires, ils nous ont conseillé d'utiliser un cycle en spirale pour ce projet. En effet, c'est le cycle de vue le plus adapté car nous devions développer un grand nombre de fonctionnalités. De plus, nous ne connaissions pas bien le domaine et le vocabulaire métier du projet. Ce cycle de vie était donc le mieux adapté pour ce type de projet, d'autant plus que notre application possédait déjà un existant.

Dans un premier temps, nous avons eu un peu de mal à comprendre le sujet, d'une part car une analyse de l'existant était nécessaire, d'autre part car nos commanditaires n'étaient pas fixés sur les améliorations à implémenter.

Ensuite, nous avons commencé à reprendre l'application que nos commanditaires nous avaient transmise en pensant implémenter les nouvelles fonctionnalités directement dans l'existant. Nos commanditaires nous ont ensuite demandé de modifier le côté graphique de l'application en essayant de créer une interface plus intuitive et plus dynamique. Nous avons donc décidé de ne plus implémenter l'existant mais de refondre totalement l'application. Dans le même temps, nous rédigeons la charte du projet.

Il s'est ensuite posé le problème de l'API à utiliser pour la gestion des fonds cartographiques. Le groupe de projet précédent avait utilisé l'API WIND, développé par The Nhân Luong dans le cadre de sa thèse. Après lecture du rapport de l'ancien groupe de projet, et quelques essais de l'API, nous nous sommes rendu compte qu'elle restait compliqué à prendre en main, et qu'il n'y avait pas de documentation décrivant précisément les fonctionnalités disponibles, et, étant donné que le créateur terminait sa thèse, nous aurions eu de grande difficultés à obtenir des informations de sa part.

De plus, l'API WIND se basait entre autres sur l'API Google Maps dans sa version 2, version qui est dépréciée depuis le 19 Mai 2010, et qui ne sera plus utilisable à partir du 19 Mai 2013 (source : site officiel *Google Developers*, cf. *Webographie*).

Nous avons donc proposé à nos commanditaires d'utiliser l'API Google Maps v3 qui possède une large communauté ainsi qu'une documentation très aboutie.

Une fois que les commanditaires ont accepté cette demande, nous avons commencé à développer selon un cycle de développement en spirale. C'est-à-dire que nous programmions des prototypes de l'application, sans pour le moment penser au côté interface.

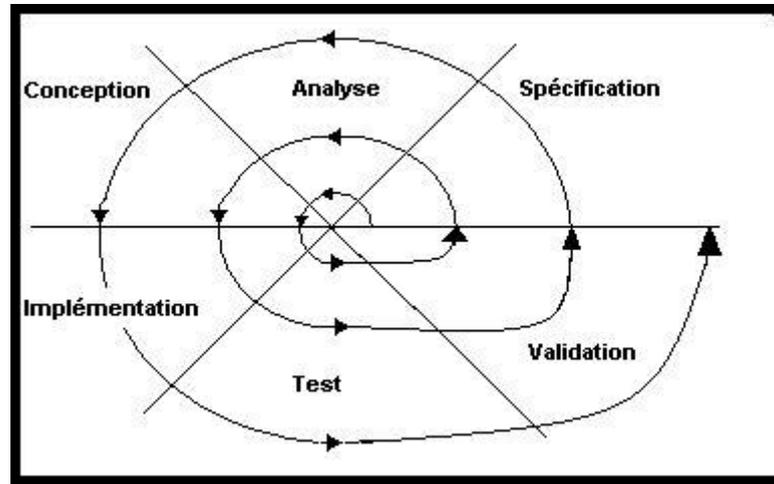


Figure 18 : Cycle de vie en spirale

Durant le développement de ces prototypes, nous avons rencontré quelques problèmes, avec les services de GeoPot étant donnés qu'ils sont basés à Pau, et que nous ne pouvons pas effectuer de test. Par exemple, lorsque les chercheurs le mettent à jour ou alors lorsque le serveur n'était pas disponible. De plus, il nous fallait récupérer les fichiers de configuration nécessaires à l'appel des services qui étaient mis à jour régulièrement. Aussi, nous avons observé que nous ne pouvons pas développer certaines fonctionnalités en local sur nos machines. Nous en avons donc discuté avec nos commanditaires qui ont alors contacté M. Thierry Nodenot afin qu'il crée et mette à notre disposition un espace de travail sur le serveur Erozate. Nous avons alors utilisé l'outil PFN afin de déposer nos fichiers sur le serveur.

Nous avons donc continué à établir des prototypes en utilisant le principe de la *Proof of Concept* (preuve de faisabilité) afin de montrer à nos commanditaires que nous arrivions à invoquer les services web et à en exploiter les résultats.

En parallèle nous avons commencé à créer les nouvelles interfaces en utilisant la librairie JQuery du langage JavaScript qui permet de créer des interfaces dynamiques.

Pour la frise chronologique, un de nos commanditaires nous a conseillé l'API CHAP LINKS qui permet, entre autres, de créer des frises dynamiques avec un design moderne, ce qui correspondait parfaitement avec le style de nos interfaces.

Une fois les interfaces bien avancées, nous avons commencé à y inclure les fonctionnalités développées dans les prototypes. Nos réunions régulières avec les commanditaires nous ont permis de faire valider et modifier nos travaux de manière rapide.

Une des fonctionnalités importante de notre application est la sauvegarde et la restauration de l'application. Cependant, à la date de rendu du rapport, cette fonctionnalité est encore en cours de développement.

## 4.2 Plannings prévisionnel et final

PLANNING PRÉVISIONNEL du 12/11/2012																														
	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
semaine universitaire																														
semaine calendaire	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14		
Date	24/9	1/10	8/10	15/10	22/10	29/10	5/11	12/11	19/11	26/11	3/12	10/12	17/12	24/12	31/12	7/1	14/1	21/1	28/1	4/2	11/2	18/2	25/2	4/3	11/3	18/3	25/3	1/4		
Renseignements semaine					vac									vac	vac										vac					
Etude des technologies et langages utilisés	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X		
Analyse et compréhension du sujet	X	X	X	X	X																									
Analyse et compréhension de l'existant						X	X	X																						
Charte							X	X	X																					
CDC Fonctionnel									X	X																				
CDC Technique										X	X	X	X																	
Programmation																X	X	X	X	X	X	X	X	X	X	X	X	X		
Codage 1ere partie - réactualisation & tests										X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X		
Codage 2eme partie - Date & tests										X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X		
Codage 3eme partie - Images & tests																	X	X	X	X	X	X	X	X	X	X	X	X		
Codage 4eme partie - Annotations & tests																			X	X	X	X	X	X	X	X	X	X		
Debugage																														
Rédaction finale du mémoire et des manuels																														
Préparation soutenance																														
Soutenance																														

Figure 19 : Planning prévisionnel

PLANNING FINAL du 26/03/2013																														
	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
semaine universitaire																														
semaine calendaire	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14		
Date	24/9	1/10	8/10	15/10	22/10	29/10	5/11	12/11	19/11	26/11	3/12	10/12	17/12	24/12	31/12	7/1	14/1	21/1	28/1	4/2	11/2	18/2	25/2	4/3	11/3	18/3	25/3	1/4		
Renseignements semaine																														
Etude des technologies et langages utilisés	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X		
Analyse et compréhension du sujet	X	X	X	X	X																									
Analyse et compréhension de l'existant							X	X	X																					
Charte									X	X	X	X																		
Documentation associée (Les personnes affectés a la tache)																														
ESA (tout le monde )																														
ESR (une personne)																														
EC (une personne) gestion de la frise																														
Mise en commun des trois services																														
Ajout des images (une ou deux personnes)																														
Mise en place des thèmes ( une deux personnes)																														
Mise à jour de import/export RDF (deux)																														
Mise à jour des interfaces (deux personnes)																														
Rédaction finale du mémoire et des manuels ( tous)																														
Préparation soutenance (tous)																														
Soutenance (tous)																														

Figure 20 : Planning final

### 4.3 Analyse des plannings

Le premier planning prévisionnel a été réalisé le 12/11/2012. A ce moment-là nous n'étions pas vraiment conscient de la charge de travail à réaliser et du temps qu'elle prendrait. De ce fait nous avons pris du retard sur le projet à cause des problèmes évoqués dans le paragraphe « 4.1 Démarches de développement ». De plus, nous avons pris du temps pour prendre en main tous les nouveaux langages et technologies dont nous avons besoin pour développer l'application. Nous voyons bien que les tâches du planning final sont plus regroupées vers la date butoir que sur le planning prévisionnel. D'autre part nous avons d'autres projets à réaliser lors de notre cursus de seconde année de DUT, en plus des cours et des partiels présents tout au long de l'année. Etant donné que c'était la première fois que nous développions un projet de cette envergure, il nous était complexe d'évaluer les durées nécessaires à la réalisation des différentes fonctionnalités que nous devions réaliser pour aboutir à une application de qualité. Après 7 versions de planning tout au long du projet, c'est au 26/03/2013 que nous avons réalisé la version finale.

### 4.4 Organisation du projet

Au commencement, nous répartissions les tâches selon les compétences de chacun, ensuite, nos tuteurs et commanditaires nous ont conseillés de répartir les tâches comme suit :

- Etant trois personnes dans le groupe de projet, définir trois tâches à faire à chaque fois.
- Affecter à chaque tâche une personne qui en est le responsable, puis mettre un autre membre du groupe en soutien sur cette tâche.

Ainsi nous avons pu continuer le projet de manière plus optimisée et mieux nous répartir les tâches ce qui nous a permis de repérer plus facilement l'avance ou le retard que nous avions sur les fonctionnalités à développer.

## 5. BILAN ET PERSPECTIVES

---

### 5.1 Bilan

Ce projet a été la concrétisation de nos deux années passées à l'IUT. Il nous a permis de tirer des enseignements aussi bien au niveau informatique qu'au niveau relationnel.

Pour commencer, au niveau informatique, ce projet nous a permis d'approfondir un grand nombre de nos connaissances techniques comme le PHP ou encore le XHTML et le CSS. Il nous a également offert la possibilité d'apprendre de nombreuses technologies que nous ne connaissions pas et qui nous serviront dans notre future vie professionnelle, comme les différentes APIs Google Maps V3 et CHAP LINKS, des librairies comme JQuery, ou encore des langages tels que JavaScript et XML couplé avec l'utilisation du XPath. Mais nous avons aussi eu l'occasion d'apprendre un autre mode de stockage de données que la base de données relationnelle apprise en cours : le RDF, avec l'utilisation de requêtes SPARQL, qui sont des technologies en pleine essor. En effet, ces dernières ne sont utilisées en majorités que par des laboratoires de recherche informatique mais commencent à gagner du terrain dans les entreprises privées. Nous avons également appris à invoquer différents services web réalisés par le laboratoire de recherche de l'Université de Pau et des Pays de l'Adour.

Au niveau de la gestion du code et des documents, il aurait été préférable que nous utilisions un gestionnaire de version du type Subversion(SVN).

Nous avons donc dû appréhender toutes ces technologies les unes après les autres et ainsi parfaire notre culture informatique. De même, ce projet nous a permis de saisir toute l'importance de l'analyse et de la documentation, notamment lors de la reprise d'un projet, l'analyse de l'existant est primordiale et sans la documentation du projet GeoText2Map nous n'aurions probablement pas réussi à atteindre l'objectif qui nous était fixé au départ.

Ensuite, au niveau relationnel, ce projet nous a démontré toute la difficulté de comprendre un sujet existant ainsi que les besoins de nos commanditaires. En effet, il nous a fallu un peu plus de un mois, ne serait-ce que pour comprendre le sujet existant et pour comprendre ce que nous avons à faire. Pour cela nous avons dû poser un grand nombre de questions souvent illustrées par des schémas, cependant cette partie était fondamentale pour réaliser de manière optimale notre application.

Par ailleurs, ce projet nous a permis d'apprendre à travailler en groupe sur une longue période, pour cela nous avons dû répartir entre nous les différentes tâches à faire afin de respecter les délais qui nous étaient imposés. La communication est donc partie intégrante d'un projet afin de limiter au maximum toute forme de conflits et, s'il y en a, de les résoudre le plus rapidement possible afin de mener à notre projet à bien.

## 5.2 Perspectives

Au départ, notre projet consistait à faire évoluer l'application GeoText2Map avec différentes fonctionnalités voulues par nos commanditaires. A l'heure où nous rendons ce mémoire les fonctionnalités principales sont implémentées et la semaine de programmation va nous permettre d'optimiser l'application avec des fonctionnalités supplémentaires. Cela n'est pas pour autant que nous considérons que notre application est terminée, nous pouvons imaginer différentes évolutions possibles à Geo Media Tagger, le diagramme des classes ci-dessus (figure 21) en fait partie.

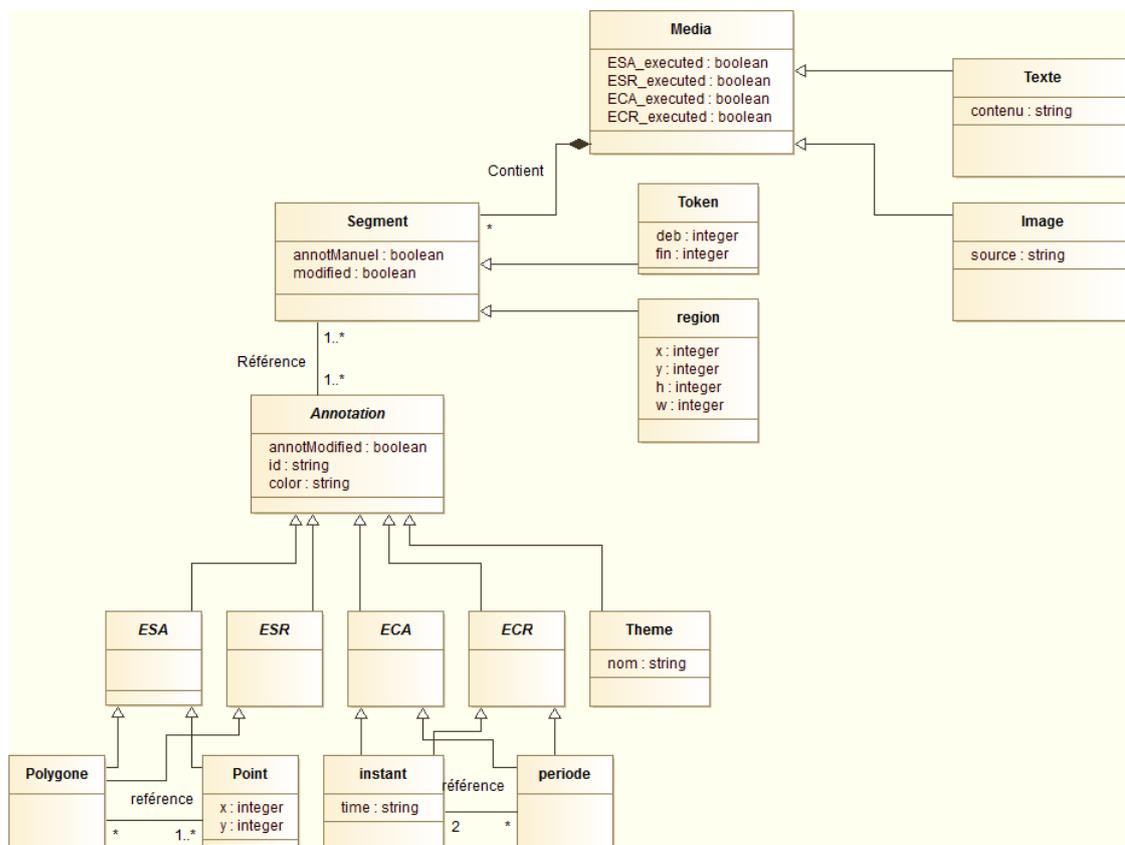


Figure 21 : Diagramme des classes contemplatif

Ce diagramme des classes contemplatif représente une modélisation possible et optimale d'une future évolution de Geo Media Tagger.

On peut remarquer l'ajout de plusieurs classes par rapport au précédent. Notamment la classe Annotation qui serait la base de toutes les classes correspondant aux différentes entités et au thème. Entre cette classe Annotation et la classe Media il y aurait une classe intermédiaire Segment représentant le segment de texte correspondant à l'annotation.

Les classes Token et Region hériteraient de la classe Segment avec les coordonnées des régions annotées pour une image ou les *tokens* de début et de fin du segment annoté.

Les classes Texte et Image, quant à elles, ont les mêmes propriétés que dans notre application actuelle.

Nous pouvons également imaginer l'annotation d'autres médias, de vidéos par exemple. Mais aussi le fait que différents utilisateurs puissent annoter une même image et ainsi voir le travail fait par les autres. Il est également imaginable d'adapter l'application à des projets pédagogiques dans les écoles primaires par exemples.

Il est donc possible d'imaginer un grand nombre d'évolutions à Geo Media Tagger, notamment en fonction de l'évolution des différents services web développés par l'équipe T2i dont nos commanditaires font partie.

## 6. ABSTRACT

---

For several years the research laboratory of the UPPA (University of Pau and Adour's Lands) works in order to collect and enhance the spatio-temporal data. Our project is part of the value of these data. We have to take an existing project in 2010 called GeoText2Map so we need to evolve this web application.

Returning a text containing place names and dates of our applications users can view on a map and a timeline this different information. There are four types of possible manipulation of data, the absolute spatial entities (like "in Pau"), the relative spatial entities (like "around Pau"), absolute entities calendar (like "14 June 2013") and finally on the calendar entities (like "in the 1980s"). They can also annotate texts, that is to say manipulate the results returned by the application. It is also expected to be able to do the same thing with an image, using a manual selection of a given image it is possible to define one or more places and a date corresponding to the image.

For this we use various tools, including the Google maps v3 API (Application Programming Interface) and Timeline, but we must also invoke a web service created by the research group of the University. Our application will be a continuation of their research work.

**Authors:** Capdeville Mathieu, David Corentin, Dumouch Léo.

**Keywords:** Computer Science, IT, geolocation, spatio-temporal data.

## 7. GLOSSAIRE

---

### *Ajax*

Ajax (Asynchronous Javascript and XML) est la combinaison de technologies telles que JavaScript, CSS, XML, dans le but de réaliser des applications internet riche (Rich Internet Application ou RIA) sans recharger toute la page. Cela offre une maniabilité et un confort d'utilisation supérieur.

### *API*

*Application Programming Interface* ou Interface de Programmation (bibliothèque de fonctionnalités mises à disposition afin de faciliter la programmation).

### *CSS*

CSS (*Cascading Style Sheets*) est un langage qui sert à décrire la présentation des documents HTML et XML. Il permet d'appliquer un style aux pages XHTML et de gérer séparément la forme du contenu.

### *ESA*

Entités Spatiales Absolues, par exemple « à Bayonne » ou encore « de Bordeaux ».

### *ESR*

Entités Spatiales Relatives, par exemple « au nord de Bayonne » ou encore « aux alentours de Bordeaux »

### *EC*

Entités Calendaires, regroupant les Entités Calendaires Absolues comme « le 14 juillet 1789 » et les Entités Calendaires Relatives comme « fin des années 1970 ».

### *Framework*

Espace de travail modulaire qui regroupe un ensemble de bibliothèques et de conventions permettant le développement rapide d'applications

### *JavaScript*

JavaScript est un langage de programmation principalement utilisé dans les pages Web interactives côté client.

### *PFN*

PHPfileNavigator (explorateur de fichiers via une interface Web).

### **PHP**

PHP (*Hypertext PreProcessor*) est un langage de script libre de droit principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété côté serveur.

### **RDF**

RDF (*Resource Description Framework*) est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées. RDF est une surcouche basée sur XML.

### **SPARQL**

SPARQL est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet. Son nom est un acronyme qui signifie *SPARQL Protocol and RDF Query Language*

### **xHTML**

xHTML (*eXtensible HyperText Markup Language*) est un langage de balisage servant à écrire des pages pour le Web. Il est le successeur d'HTML dont le but est d'afficher et de formater du texte.

### **XML**

XML (*Extensible Markup Language*) est un métalangage (formalisme conçu pour décrire un langage) informatique de balisage générique. Il sert essentiellement à stocker et/ou transférer des données de type texte structurées en champs arborescents

### **WIND**

WIND (*Web INteraction Design*) est une interface de programmation développée par The Nhân Luong qui a pour but de donner à un informaticien les outils nécessaires pour créer une application Web géographique personnalisée intégrant des cartes, sans avoir à connaître toutes les fonctionnalités sous-jacentes issues de différentes technologies et API telles que OpenLayers.

## 8. TABLE DES ILLUSTRATIONS

---

FIGURE 1 : MAQUETTE DE GEOTEXT2MAP .....	5
FIGURE 2 : MAQUETTE DE GEOTEXT2MAP .....	5
FIGURE 3 : DIAGRAMME DES CAS D'UTILISATION .....	10
FIGURE 4 : MAQUETTE DE L'ACCUEIL .....	11
FIGURE 5 : MAQUETTE DE L'ECRAN PRINCIPAL .....	12
FIGURE 6 : MAQUETTE DE L'ECRAN PRINCIPAL AVEC ANNOTATION .....	13
FIGURE 7 : MAQUETTE PRESENTANT LES ENTITES SUR LA CARTE.....	14
FIGURE 8 : MAQUETTE D'ACCUEIL POUR SAISIR UN TEXTE.....	15
FIGURE 9 : MAQUETTE DE SAISIE MANUELLE D'UN TEXTE.....	16
FIGURE 10 : MAQUETTE DE L'ECRAN PRINCIPAL AVEC UN TEXTE.....	16
FIGURE 11 : MAQUETTE DE L'ECRAN PRINCIPAL ET ESA .....	17
FIGURE 12 : MAQUETTE PRESENTANT UNE ESA.....	17
FIGURE 13 : DIAGRAMME DES CLASSES.....	<b>ERREUR ! SIGNET NON DEFINI.</b>
FIGURE 14 : DIAGRAMME ETAT-TRANSITION .....	20
FIGURE 15 : FONCTIONNEMENT DU MVC .....	22
FIGURE 16 : EXEMPLE DE FONCTIONNEMENT DU MVC .....	23
FIGURE 17 : EXTRAIT DU GRAPHE RDF PERMETTANT L'EXPORT DE L'APPLICATION .....	33
FIGURE 18 : CYCLE DE VIE EN SPIRALE .....	35
FIGURE 19 : PLANNING PREVISIONNEL.....	37
FIGURE 20 : PLANNING FINAL .....	37
FIGURE 21 : DIAGRAMME DES CLASSES CONTEMPLATIF .....	40

## 9. WEBOGRAPHIE

---

### **API Google Maps :**

<https://developers.google.com/maps/documentation/javascript/v2/>

<https://developers.google.com/maps/documentation/javascript/reference>

### **API Chap Links :**

<http://almende.github.com/chap-links-library/>

### **JQuery :**

<http://jquery.com/>

### **Wikipedia:**

[http://fr.wikipedia.org/wiki/Service\\_Web](http://fr.wikipedia.org/wiki/Service_Web)

<http://fr.wikipedia.org/wiki/PHP>

<http://fr.wikipedia.org/wiki/JavaScript>

<http://fr.wikipedia.org/wiki/JQuery>

<http://fr.wikipedia.org/wiki/XML>

<http://fr.wikipedia.org/wiki/XPath>

[http://fr.wikipedia.org/wiki/Resource\\_Description\\_Framework](http://fr.wikipedia.org/wiki/Resource_Description_Framework)

<http://fr.wikipedia.org/wiki/SPARQL>

[http://fr.wikipedia.org/wiki/Interface\\_de\\_programmation](http://fr.wikipedia.org/wiki/Interface_de_programmation)

<http://www.dicodunet.com/definitions/google/api-google-map.htm>

<http://fr.wikipedia.org/wiki/NetBeans>

<http://fr.wikipedia.org/wiki/Notepad%2B%2B>

<http://www.w3.org/RDF/Validator>

### **PHP :**

[www.php.net](http://www.php.net)

### **JavaScript :**

<http://javascript.developpez.com>