

Travail de bachelor 2010

Filière Informatique de gestion

Génération d'étiquettes énergétiques pour les bâtiments



Etudiant : Hervé Sierro

Professeur : Jean Hennebert



Préface

De nos jours, nous sommes de plus en plus attentifs à la dimension environnementale liée à nos comportements. Que ce soit l'augmentation des coûts de l'énergie, les prises de position politique ou encore les tendances écologiques actuelles, la diminution des impacts environnementaux est devenue un objectif social.

Dans le but de diminuer notre consommation d'énergie, plusieurs projets visant à aider les particuliers ou les institutions publiques ont vu le jour. Ces projets sont accompagnés de logiciels permettant non seulement de réaliser un suivi hebdomadaire des consommations mais aussi de détecter des anomalies dans ces consommations.

Ce travail s'inscrit dans la thématique de la réduction des impacts environnementaux. Il vise à fournir aux logiciels existants et aux futurs logiciels deux outils supplémentaires permettant de visualiser et comparer les consommations d'énergie. Le premier outil fournira un service d'affichage des consommations sous forme de graphiques interactifs. Le second permettra la représentation des consommations sous forme d'étiquettes énergétiques telles que l'on en trouve sur certains électroménagers actuels avec une échelle d'appréciation de A à E.

Ce rapport final de travail de bachelor décrit les différentes étapes qui ont été nécessaires au développement des deux outils précédemment cités.



Remerciements

Je tiens à remercier cordialement toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce travail de bachelor.

Une attention particulière à :

M. Jean Hennebert, pour m'avoir encadré dans la réalisation de ce projet tout au long de ces trois mois. Merci pour les précieux conseils et idées prodigués.

M. Pascal Pitteloud, adjoint scientifique II à la HES-SO Valais, pour m'avoir fourni tout ce dont j'avais besoin pour accéder au logiciel SBat3 et au serveur de développement. Merci pour sa collaboration lors des séances pour le transfert technologique.

M. Simon Martin, adjoint scientifique II à la HES-SO Valais, également pour sa collaboration lors des séances pour le transfert technologique.

M. Camille Ançay, responsable de l'énergie et installations à l'Etat du Valais, pour avoir pris le temps de m'expliquer les aspects énergétiques liés au logiciel SBat3 et pour avoir répondu à mes nombreuses questions.

Merci enfin à toutes les personnes m'ayant aidé à relire et à peaufiner ce document. Merci pour vos efforts et votre patience.



Résumé

Ce travail de bachelor fournit aux logiciels web de surveillance énergétique des bâtiments deux outils supplémentaires permettant de visualiser et comparer les consommations d'énergie. Le premier outil fournit un service d'affichage des consommations sous forme de graphiques interactifs. Le second fournit un service permettant la représentation des consommations sous forme d'étiquettes énergétiques avec une échelle d'appréciation de A à G.

Ces deux services web pourront être implémentés par le logiciel de surveillance énergétique SBat3 qui est utilisé par le canton du Valais et également par un projet CTI en cours de réalisation à l'institut IIG.

Le logiciel client peut interroger les deux modules via l'adresse du serveur PHP où sont implémentés les deux services. L'adresse interrogée est accompagnée d'un paramètre correspondant à l'adresse du fichier XML contenant les données énergétiques à traiter. Cette solution permet donc aux modules de fonctionner indépendamment de la technologie dans laquelle est implémenté le logiciel client.

Dans le cadre du module de génération de graphiques, le client reçoit une réponse sous forme de code HTML et JavaScript représentant le graphique. Les graphiques sont générés à l'aide de la librairie JavaScript Highcharts.

Dans le cadre du module de génération d'étiquettes énergétiques, le client reçoit en réponse une image de type png ou un document PDF représentant l'étiquette énergétique. Les fichiers réponses sont générés par l'intermédiaire des utilitaires wkhtmltopdf et wkhtmltoimage.

Ce projet a été réalisé dans le cadre d'un travail de bachelor HES en informatique de gestion et a été effectué du 17 mai au 16 août 2010, pour un total de 364 heures.



Table des matières

1	Présentation du travail	1
1.1	Description	1
1.2	Contexte du projet	1
1.3	Délivrables du projet	1
1.4	Déclaration sur l'honneur	2
2	Gestion de projet	3
2.1	Introduction	3
2.2	Planning et suivi de projet	3
2.3	Réalisation	4
3	Choix technologiques	6
3.1	Introduction	6
3.2	Server Side	6
3.2.1	PHP	7
3.2.2	XML	8
3.3	Client Side	9
3.3.1	HTML	9
3.3.2	CSS	10
3.3.3	JavaScript	11
3.3.4	Navigateurs Internet	11
4	Module de génération de graphiques	12
4.1	Introduction	12
4.2	Contexte	12
4.2.1	Graphiques SBat3	12
4.3	Librairies graphiques	17
4.3.1	Choix final	17
4.4	Implémentation	20
4.4.1	Processus	20
4.4.2	Communication client-serveur	21
4.4.3	Traitement des données	24



4.5	Exemples d'utilisations.....	26
4.5.1	PHP	26
4.5.2	ASPX	26
4.5.3	HTML	26
4.6	Exemples de graphiques	27
5	Module de génération d'étiquettes énergétiques	30
5.1	Introduction	30
5.2	Contexte.....	30
5.3	Librairie de génération d'images et de documents PDF	33
5.3.1	Librairie PDF	33
5.3.2	Librairie image.....	35
5.4	Implémentation	36
5.4.1	Processus.....	36
5.4.2	Communication client-serveur.....	37
5.4.3	Traitement des données	38
5.5	Exemples d'utilisations.....	41
5.6	Exemples d'étiquettes énergétiques.....	42
6	Déploiement.....	44
6.1	Module de génération de graphiques.....	44
6.2	Module de génération d'étiquettes énergétiques.....	44
7	Problèmes rencontrés.....	45
8	Conclusion	46
9	Bibliographie	47
10	Tables des illustrations	49
10.1	Tableaux.....	49
10.2	Images	49
11	Tableau des annexes.....	51

1 Présentation du travail

1.1 Description

L'Etat du Valais utilise actuellement un logiciel web pour la surveillance énergétique de ses bâtiments. Ce logiciel, appelé SBat3 (technologie ASP), permet le suivi hebdomadaire des relevés des compteurs électriques, d'eau et de gaz, et permet de tracer l'évolution des consommations sous forme de graphique. Il permet également de détecter des anomalies dans les consommations.

Le projet a pour objectif principal la mise au point d'un calcul d'étiquettes énergétiques pour les bâtiments. Ces étiquettes sont similaires aux étiquettes que l'on trouve sur certains électroménagers actuels avec une échelle d'appréciation de A à E. Il s'agira tout d'abord de comprendre les fonctionnalités d'analyse énergétique des bâtiments telles qu'implémentées dans SBat3. Ensuite, une analyse de l'état de l'art du calcul des étiquettes énergétiques devra être réalisée. Finalement, il s'agira de sélectionner et d'implémenter le calcul de ces étiquettes dans un nouveau logiciel en cours de réalisation (en lien avec un projet CTI réalisé à l'institut IIG sur une technologie PHP). Il s'agira également d'analyser et de sélectionner un Framework PHP pour la création de graphiques dans une application web afin de réimplémenter les diverses générations de graphiques disponibles dans SBat3 et d'implémenter la génération des étiquettes énergétiques.

1.2 Contexte du projet

Ce projet a été réalisé dans le cadre d'un travail de bachelor HES en informatique de gestion.

Le temps disponible pour la réalisation de ce travail a été de 360 heures, à raison de 20 heures de travail par semaine durant les six premières semaines et 45 heures par semaine durant les six dernières semaines. La première partie s'étant déroulée en parallèle avec les cours du 6^{ème} semestre de la formation en informatique de gestion.

Ce travail de bachelor s'est déroulé du 17 mai au 16 août 2010.

1.3 Délivrables du projet

Ce travail de bachelor se compose de trois parties distinctes: un rapport final et deux applications fonctionnelles.

Les applications ont été réalisées en respectant le cahier des charges établi avec le professeur responsable du projet. Elles sont accompagnées par une documentation exhaustive et suffisamment fournie pour permettre à une tierce personne de poursuivre le travail.



1.4 Déclaration sur l'honneur

Je déclare, par ce texte, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré.

2 Gestion de projet

2.1 Introduction

Ce chapitre décrit le déroulement du projet, de sa planification à sa réalisation. Les différentes phases du projet y sont identifiées, décrites, planifiées et hiérarchisées.

Le cahier des charges de ce travail peut être consulté en annexe de ce document.

2.2 Planning et suivi de projet

Ce tableau (Tableau 1) représente l'état du projet lors de son rendu. Nous y trouvons également la planification initiale et le suivi du projet. Chaque activité accomplie durant le projet a été attribuée à une tâche définie durant la planification initiale. Cette solution permet de visualiser quotidiennement la progression de chaque tâche et permet de facilement discerner le travail restant du travail effectué.

De plus, la colonne d'écart permet de distinguer les tâches achevées qui ont été sous-évaluées ou surévaluées ainsi que les tâches en cours risquant de ne pas correspondre avec la planification initiale. Voici l'état du projet lors de son rendu :

N°	Tâche	Planifié	Réalisé	Ecart	Statut
1	Recherche et assimilation	h	h	h	
1	1 Compréhension de la situation initiale du projet	4	4	0	✓
2	2 Réflexion cahier des charges	7	7	0	✓
3	3 Rédaction cahier des charges	7	7	0	✓
4	4 Planification initiale du projet	2	2	0	✓
2	Analyse				
1	1 Introduction à SBat3	10	14	4	✓
3	3 Analyse des graphiques SBat3	38	16	-22	✓
2	2 Analyse génération d'étiquettes énergétiques	29	24	-5	✓
3	Implémentation				
1	1 Définir un Framework PHP de développement	7	0	-7	✓
2	2 Définir une librairie/technologie pour la génération de graphiques	31	32	1	✓
3	3 Implémentation modules graphiques	51	52	1	✓
4	4 Implémentation génération d'étiquettes énergétiques	48	64	16	✓
4	Remise du projet				
1	1 Refactoring et documentation du projet	32	48	16	✓
2	2 Rédaction du rapport final	54	54	0	✓
3	3 Préparation du rendu du projet	10	10	0	✓
5	Divers				
1	1 Séances hebdomadaires	18	18	0	✓
2	2 Suivi de projet	12	12	0	✓
Total Heures		360	364	4	

Tableau 1 : Gestion et suivi de projet

Globalement, le projet n'a pas subi d'écart trop important. Néanmoins, l'analyse des graphiques de SBat3 a été effectuée plus rapidement que prévue alors que l'implémentation du module de génération d'étiquettes énergétiques a demandé d'avantage de temps que planifié. Ce dernier écart étant dû aux difficultés rencontrées lors de l'installation de certaines librairies sur un serveur Linux (voir chapitre 7. Problèmes rencontrés).

2.3 Réalisation

Ce chapitre présente un résumé du déroulement du travail durant les douze semaines du projet. Le temps de travail durant les semaines une à six s'est élevé à 50% et 100% pour les semaines sept à douze.

Des informations détaillées peuvent être consultées dans le document de planification annexé à ce document.

Semaine 01, du 17 au 21 mai 2010

Cette semaine correspond à la phase d'introduction au travail de bachelor et à la phase d'analyse du projet.

Tout d'abord, il a fallu comprendre les enjeux du projet et les besoins utilisateurs pour ensuite définir le cahier des charges. Il a été décidé de commencer le développement par le module de génération de graphiques pour ensuite se concentrer sur le module de génération d'étiquettes énergétiques.

Durant cette semaine, la personne en charge du projet a pu faire connaissance avec les collaborateurs amenés à utiliser le produit final dans de futurs développements.

Semaine 02 et 03, du 24 mai au 4 juin 2010

Ces deux semaines correspondent à la phase d'analyse du logiciel SBat3.

L'analyse de ce logiciel a été effectuée dans le but de bien comprendre le rôle des graphiques dans l'analyse des comportements énergétiques d'un bâtiment.

Les types de graphiques à mettre en place ainsi que les différentes fonctionnalités devant apparaître sur les graphiques ont pu être déterminés.

Semaine 04 à 07, du 7 juin au 2 juillet 2010

Ces quatre semaines correspondent aux phases d'analyse et d'implémentation du module de génération de graphiques.

Tout d'abord, il a fallu déterminer la technologie à utiliser pour permettre au module d'être développé indépendamment du programme par lequel il sera interrogé.

Ensuite, plusieurs librairies de génération de graphiques ont été sélectionnées et testées pour finalement en retenir une seule. La librairie retenue a été implémentée et les données traitées rendues dynamiques pour permettre l'utilisation du module à distance.

Pour terminer, la documentation du développement a été établie et une présentation du travail accompli aux collaborateurs a été effectuée.

Semaine 08 à 10, du 5 juillet au 23 juillet 2010

Ces trois semaines correspondent aux phases d'analyse et d'implémentation du module de génération d'étiquettes énergétiques.

Premièrement, un état de l'art des solutions permettant la génération d'étiquettes énergétiques a été effectué. Suite à cette analyse, il a été décidé de ne pas implémenter l'algorithme permettant de calculer la valeur de l'étiquette mais de concentrer le développement sur l'affichage de l'étiquette. L'affichage comprenant la génération d'étiquettes dans un format image et PDF.

Ce module devant également être développé indépendamment du programme par lequel il sera interrogé, la technologie utilisée par le module précédent a été reprise.

Par la suite, différentes solutions de génération d'images et de documents PDF ont été testées, la solution retenue implémentée et les données traitées rendues dynamiques pour permettre l'utilisation du module à distance.

Au final, la documentation du développement a été établie et une présentation du travail accompli aux collaborateurs a été effectuée.

Semaine 11 à 12, du 26 juillet au 6 août 2010

Ces deux semaines correspondent à la phase de remise du projet.

Ces semaines ont été consacrées à la rédaction du rapport final, à la finalisation des deux outils développés ainsi qu'à la préparation des livrables demandés.

3 Choix technologiques

3.1 Introduction

Ce chapitre dresse un panorama des différentes technologies utilisées lors du développement.

Les choix technologiques concernent autant le module de génération de graphiques que le module de génération d'étiquettes énergétiques pour bâtiments. En effet, les problématiques sont les mêmes : les deux modules doivent pouvoir être interrogés par un programme indépendamment de la technologie dans laquelle est développé ce dernier.

Notons que les modules développés sont des services web, d'où l'intérêt de distinguer la technologie implémentée côté serveur de celle implémentée du côté client. Un langage de programmation dit "Server Side" est un langage qui est interprété sur le serveur, au contraire d'un langage "Client Side" qui est interprété sur l'ordinateur de l'utilisateur. Concrètement, un langage "Server Side" permet d'interroger une base de données, de traiter et calculer des informations qui seront ensuite affichées chez le client à l'aide de langages dit "Client Side".

3.2 Server Side

Actuellement le logiciel SBat3 est développé en ASP. Cependant, une version PHP du logiciel sera prochainement implémentée. Le module de génération de graphiques ainsi que le module de génération d'étiquettes ont donc été développés de manière à être facilement intégrables aussi bien à la future version du logiciel SBat qu'à d'autres types de projets.

En effet, le module de génération d'étiquettes énergétiques devra pouvoir être ajouté à la version actuelle du logiciel SBat (technologie ASP). D'autre part, il est possible que les deux modules soient intégrés à un projet CTI réalisé à l'institut IIG sur technologie PHP.

En somme, ce développement étant étroitement lié à la future version du logiciel SBat développé en PHP, il a été demandé que les deux modules de ce travail soient également réalisés à l'aide de cette même technologie. Cependant, afin que ce travail puisse être repris dans d'autres projets, la communication entre le client et les modules réalisés a été implémentée de manière standardisée. Pour ce faire, la technologie retenue a été l'XML.

3.2.1 PHP

PHP¹ (Hypertext Preprocessor) est un langage multiplateforme utilisé comme langage de script côté serveur. Le serveur qui héberge le site web va interpréter le code PHP, interroger une base de données au besoin et générer une réponse sous forme de code HTML, CSS ou JavaScript et l'envoyer au navigateur du client. Ce langage permet la création d'applications dynamiques, la plupart du temps destinées au web. C'est un langage libre et gratuit. Son utilisation est simple et son installation rapide. La syntaxe PHP est très proche de celle du C dont il est un des nombreux descendants.



Figure 1 : Choix technologiques – Schéma PHP

Source : wikipedia.org

Les avantages principaux du PHP sont [1] :

- Sa grande communauté de développeurs et le grand nombre de scripts disponibles sur le web
- Sa licence GNU GPL signifiant la gratuité et la disponibilité du code
- La possibilité de mélanger du code HTML au code PHP, l'intégration au sein de nombreux serveurs web ou encore la simplicité d'interfaçage avec des bases de données.

La version actuelle du PHP est la version 5. Apparue en 2004, elle permet le développement orienté objet comme on le connaît en Java : constructeur de classe, interface, héritage, définition de la portée des variables,... De plus, elle permet de lire avec plus de facilité les fichiers XML à l'aide de la librairie SimpleXML.

¹ <http://www.php.net>

3.2.2 XML

XML² (Extensible Markup Language) est un langage de balisage permettant de structurer des données que l'on désire stocker ou transférer. On dit d'XML qu'il est extensible car il donne la possibilité à l'utilisateur de définir ses propres balises.

Contrairement à l'HTML, XML permet de décrire le contenu plutôt que de le représenter. La mise en page des données est assurée par un langage tiers comme par exemple le CSS³ (Cascading StyleSheet) ou l'XSL⁴ (eXtensible StyleSheet Language).

Dans notre cas, la récupération et le traitement du XML sont effectués directement en PHP.

Les avantages principaux du XML sont [2] :

- La lisibilité et l'accessibilité : compréhension facile du contenu
- Auto descriptif et extensible : liberté dans la définition des balises
- Intégrabilité : toute application possédant un parseur peut interpréter/lire un XML
- Structure arborescente : permettant de représenter la majorité des problèmes informatiques

```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```

Figure 2 : Choix technologiques – Exemple XML

Source : kirupa.com

² <http://www.xml.com>

³ <http://www.w3.org/Style/CSS/>

⁴ <http://www.xml.com/pub/a/2000/08/holman/index.html>

3.3 Client Side

Dans le cadre de ce projet, les principaux langages interprétés du côté client sont l'HTML, le CSS et JavaScript :

3.3.1 HTML

HTML⁵ (Hypertext Markup Language) est un langage de balisage permettant de mettre en forme des pages web. Il permet également la lecture de documents sur Internet à partir de machines différentes, grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée URL [3].

La version actuelle est l'HTML 4. Publiée en 1997, elle vise à standardiser le support des scripts, les feuilles de style ou encore d'éléments HTML comme les cadres (frames). La version 5 est à l'étude depuis 2008 et commence à être interprétée par différents navigateurs web. Cette version a pour objectif d'harmoniser les médias et de structurer la mise en page par des éléments plus "sémantiques" tels que l'élément <nav> représentant une section de liens de navigation ou encore l'élément <dialog> utilisé pour définir une conversation [4].



Figure 3 : Choix technologiques – Exemple de code HTML

⁵ <http://www.w3schools.com/html/>

3.3.2 CSS

CSS⁶ (Cascading Style Sheets) permet la stylisation de documents HTML par l'intermédiaire d'un document de style séparé.

Les avantages sont donc multiples [5]:

- Toutes les pages d'un site web faisant référence à la même feuille de style, la présentation est uniformisée.
- Un même document peut donner le choix entre plusieurs feuilles de style, par exemple une pour l'impression et une pour la lecture à l'écran. Certains navigateurs web permettent au visiteur de choisir un style parmi plusieurs.
- Le code HTML est considérablement réduit en taille et en complexité, puisqu'il ne contient ni balise ni attribut de présentation.


Code HTML	Code CSS
<pre><table id="table"> <tr> <td id="tdLeft">Hello</td> <td id="tdRight">World</td> </tr> </table></pre>	<pre>#table { border:1px solid black; text-align:center } #tdLeft { width:50px; background-color: red ;} #tdRight { width:60px; font-weight:bold ;}</pre>
Résultat	
	

Figure 4 : Choix technologiques – Exemple de code CSS

⁶ <http://www.w3.org/Style/CSS/>

3.3.3 JavaScript

JavaScript est un langage de script qui peut être intégré dans une page HTML et qui permet d'exécuter des commandes du côté client [6]. Dans le cas d'une telle intégration, c'est le navigateur web qui prend en charge l'exécution des scripts.

On utilise généralement le JavaScript pour contrôler les données saisies dans les formulaires HTML, pour interagir avec le document HTML ou de plus en plus, pour interroger des scripts se trouvant sur un serveur.

JavaScript

```
<script type="text/javascript">
<!--
    function clickBtn(){
        var valBtn = document.getElementById("btn").value ;
        alert(valBtn) ;
    }
//-->
</script>

<input type="button" id="btn" value="Click" name="btnName" onClick="clickBtn();"/>
```

Résultat



Figure 5 : Choix technologiques – Exemple de code JavaScript

3.3.4 Navigateurs Internet

L'interprétation de ces langages par les différents navigateurs Internet n'étant pas toujours identique, les modules ont été testés sur les plateformes suivantes : Internet Explorer 7 et 8, Firefox 3, Opera 10, Google Chrome 5 et Safari 4.

Le choix des navigateurs testés a été basé sur les résultats de la dernière étude du site MarketShare de Net Applications [7] publiée en juillet 2010 et qui présente la liste des navigateurs les plus utilisés.

4 Module de génération de graphiques

4.1 Introduction

Ce chapitre décrit comment le module de génération de graphiques a été réalisé.

Il dresse un panorama des graphiques existants dans le logiciel SBat3, justifie le choix de la librairie graphique utilisée pour ce projet puis décrit de quelle façon cette dernière a été implémentée.

Pour finir, il indique de quelle façon un développeur pourra utiliser le module et propose des exemples concrets de graphiques réalisés.

4.2 Contexte

Le module de génération de graphiques pouvant potentiellement être intégré à la version actuelle de SBat3 et à sa future version, les graphiques présents dans SBat ont représenté la ligne directrice de ce développement. Dès lors, une description de ces graphiques a été réalisée, accompagnée en préambule, d'une brève description des fonctionnalités offertes par le logiciel SBat3 :

SBat3 est un logiciel ayant pour objectif la surveillance énergétique des bâtiments. Son développement est assuré par l'institut de recherche de la HES-SO Valais/Wallis et l'utilisateur principal est l'Etat du Valais qui gère quotidiennement plus de 150 bâtiments publics. La version actuelle du logiciel est la version 3 qui est composée d'une architecture web client/serveur sous technologie Microsoft ASP [8].

Le logiciel permet d'administrer des bâtiments et offre pour chacun d'entre eux, une interface permettant aux concierges de saisir les données mensuelles de consommation énergétique. Ces données sont ensuite affichées sous forme de tableaux récapitulatifs ou sous forme de graphiques permettant leur comparaison. Les résultats affichés dépendent non seulement des chiffres saisis par les concierges mais également des données architecturales du bâtiment et de l'emplacement géographique de ce dernier.

Notons encore que le logiciel permet de visualiser l'historique des données saisies pour chaque bâtiment et donne la possibilité de définir des objectifs de consommation qui seront ensuite comparés à la consommation réelle.

4.2.1 Graphiques SBat3

La liste des graphiques proposée dans ce document n'est pas exhaustive. En effet, SBat3 permettant de créer des compteurs à la volée selon le bâtiment, il est possible que tous les graphiques ne soient pas cités.

Cependant, l'important n'est pas de citer tous les graphiques mais de relever les différents types de graphiques. Les graphiques SBat3 peuvent être catégorisés de la manière suivante : graphiques Flash ou graphiques HTML.

Graphiques Flash

Les graphiques suivants ont été créés en flash en raison du besoin d'interaction. En effet, ils offrent la possibilité de sélectionner un point représentant une semaine de l'année courante et permettent de visualiser les données détaillées de la semaine sélectionnée.

Contrairement à l'HTML, la technologie Flash permet à l'utilisateur d'interagir avec le graphique. Une animation Flash peut être lue à l'aide d'un Flash Player : un logiciel Adobe qui permet aux internautes de visualiser des applications et des contenus interactifs sur le web.

Le langage de programmation utilisé pour la création de ces animations est l'Action Scripts. L'apparition de la version 3.0 d'Action Script en 2007 a complètement bouleversé la manière de programmer en faisant d'Action Script un langage orienté objet.

Chauffage

Ce graphique permet de représenter la relation entre les degrés jours et l'énergie consommée en chauffage : plus la température est froide, plus la consommation sera élevée.

La ligne de régression bleue "Signature", calculée sur la base des données de l'année courante, indique le comportement énergétique du bâtiment. En effet, plus la pente est forte plus la qualité énergétique du bâtiment est considérée comme faible et inversement. En cliquant sur un des points représentant une semaine, le responsable pourra consulter dans l'encadré nommé "Infos points" la consommation et les degrés jours de la semaine concernée.

Quant à la droite noire "Année précédente", elle représente le comportement énergétique calculé sur la base des données de l'année précédente. Seuls les points de l'année courante sont affichés.

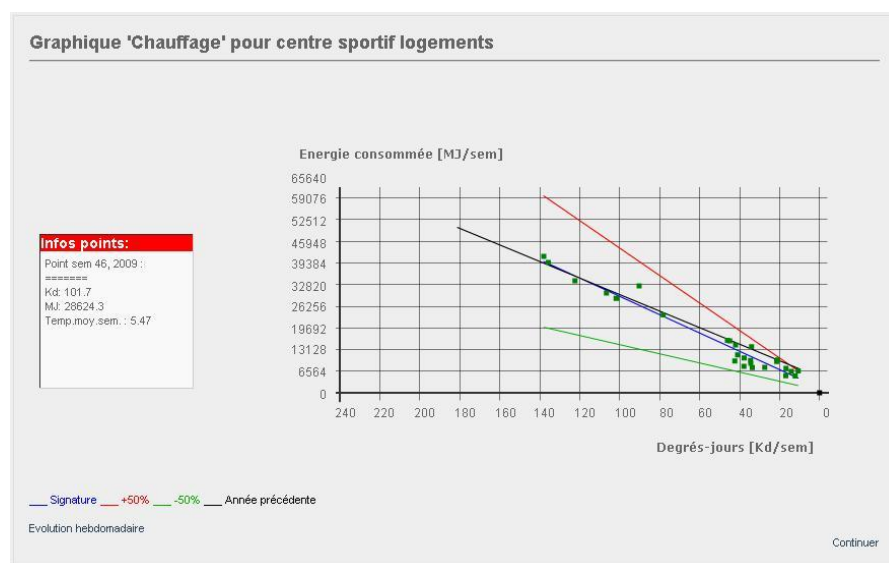


Figure 6 : SBat3 – Graphique énergétique chauffage en Flash

Source : SBat3

Solaire

Ce graphique permet d'analyser la production de l'installation solaire par rapport à l'ensoleillement hebdomadaire. On remarque logiquement que plus l'ensoleillement est fort, plus la production est grande. En cliquant sur un des points représentant une semaine, le responsable pourra consulter dans l'encadré nommé "Infos points" la production et l'ensoleillement de la semaine concernée.

La droite de régression nommée "Signature" permet de mieux visualiser le degré d'association existant entre la production et l'ensoleillement. Notons que plus la pente de la droite est élevée, plus la production l'est également.

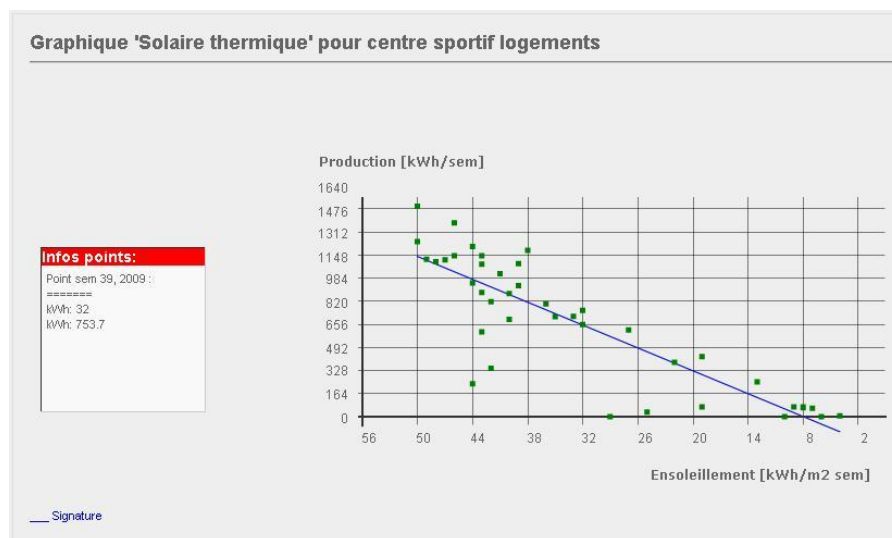


Figure 7 : SBat3 – Graphique énergétique solaire en Flash

Source : SBat3

Bilan annuel

La rubrique bilan annuel peut être consultée en sélectionnant la dernière semaine d'une année. En effet, pour que les données présentées soient exhaustives, il est nécessaire d'effectuer les calculs sur l'ensemble d'une année.

Cette page permet de visualiser un résumé des différentes énergies consommées. Pour chacune d'elle, trois représentations sont affichées : la consommation dans l'unité d'origine, la transformation de cette dernière en mégajoule par an et pour finir l'exergie⁷ retirée. Il est important de déterminer le pouvoir calorifique de chaque consommation en la transformant en mégajoule, ceci afin de pouvoir comparer les différentes énergies. Le facteur de transformation en mégajoule peut être déterminé pour chaque compteur.

Au dessous, le total des mégajoules consommés par an est divisé par la surface de référence du bâtiment pour représenter l'indice de dépense d'énergie par mètre carré.

⁷ Exergie : grandeur permettant de calculer l'énergie maximale que peut fournir un système à l'extérieur lorsqu'il se met à l'équilibre thermodynamique avec son environnement. Source : <http://fr.wikipedia.org/wiki/Exergie>

Pour finir, un graphique permet de représenter l'évolution des différents indices de dépense d'énergie sur plusieurs années. Par exemple, l'énergie consommée en mazout en 2009 a été de 1227938 MJ/an. Si l'on divise ce nombre par la surface 2268 m² du bâtiment, on obtient 541 comme indice de dépense d'énergie pour le mazout. Chaque point présent sur le graphique est calculé de cette manière.

Le graphique permet également d'analyser l'influence que peut avoir une énergie sur une autre. Ainsi, le responsable du bâtiment pourra constater que l'ajout d'une installation solaire en 2008 a fait diminuer la consommation de mazout.



Figure 8 : SBat3 – Graphique énergétique bilan annuel en Flash

Source : SBat3

Graphiques HTML

Les graphiques réalisés uniquement en HTML n'offrent aucune possibilité d'interaction à l'utilisateur, l'HTML étant un langage de représentation de données.

Ces graphiques de type histogramme ont pour objectif la représentation de la consommation d'une énergie donnée pour chaque semaine de l'année sélectionnée. Par exemple, il est possible de consulter la production solaire pour chaque semaine de l'année ou l'énergie consommée pour le chauffage d'un bâtiment.

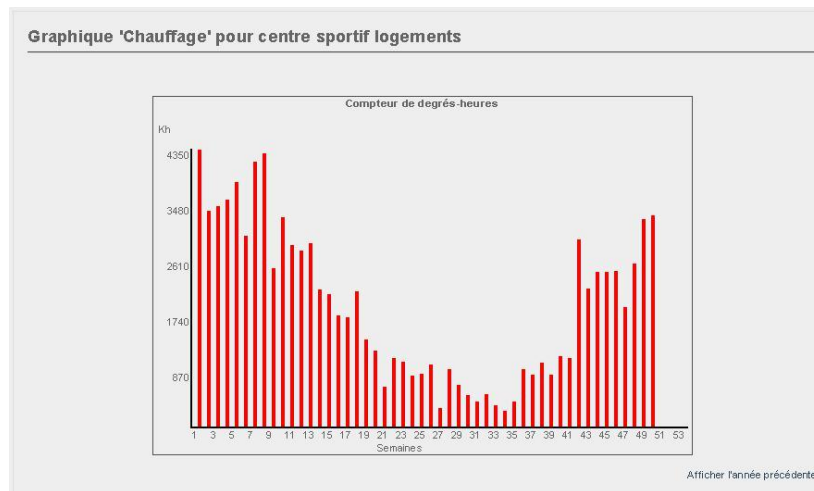


Figure 9 : SBat3 – Graphique énergétique chauffage en HTML

Source : SBat3

Pour chaque histogramme, il est possible d'afficher les données de l'année précédente :

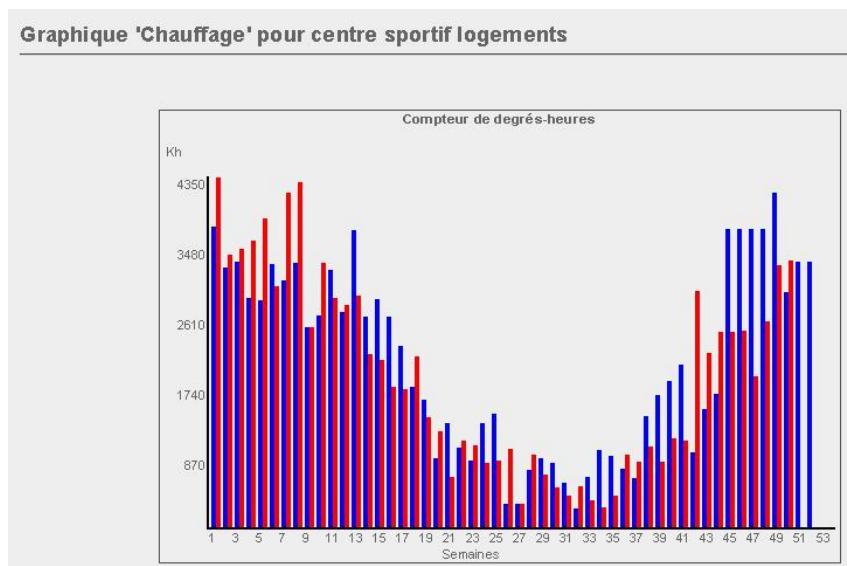


Figure 10 : SBat3 – Graphique énergétique année précédente en HTML

Source : SBat3

4.3 Bibliothèques graphiques

Les graphiques permettent aux responsables de visualiser rapidement le comportement énergétique d'un bâtiment. L'utilisateur devant gérer une multitude de données, le graphique doit faciliter la comparaison de ces données et indiquer la ligne directrice énergétique à suivre.

En conséquence, il est important d'implémenter une bibliothèque graphique dont le rendu visuel puisse aider l'utilisateur au mieux dans son travail quotidien.

La première tâche à effectuer pour la génération des graphiques est le choix d'une bibliothèque permettant de le faire. Dès le début les recherches se sont portées sur une bibliothèque JavaScript.

Le choix d'une bibliothèque PHP permettant de générer des graphiques n'aurait pas été pertinent pour notre travail. En effet, ces dernières traitent les données et renvoient à l'utilisateur un graphique en format image avec lequel aucune interaction ne sera possible. Ceci s'explique par le fait que PHP est un langage de programmation interprété du côté serveur et non par le navigateur du client. Dès lors, il serait impossible pour l'utilisateur de consulter les données précises d'un point présenté sur le graphique.

Il existe une multitude de bibliothèques JavaScript permettant de générer des graphiques. Dans un premier temps, pour rassembler un premier groupe de bibliothèques, les deux critères suivants ont été établis : une licence libre ainsi qu'une compatibilité avec les navigateurs suivants : IE 6, IE7, IE8, Firefox, Opera, Safari et Chrome.

Les bibliothèques qui ont été testées sont Flot⁸, JqPlot⁹, Highcharts¹⁰ et Google Chart Tools¹¹.

4.3.1 Choix final

Pour effectuer notre choix final parmi les bibliothèques retenues, les critères suivants ont été déterminés en fonction des caractéristiques des graphiques de SBat3:

Critères fonctionnels

1. Type des séries : des points, des lignes et des colonnes doivent pouvoir être gérés.
2. La graduation des axes doit pouvoir être inversée. Dans SBat3, les unités degrés-heures sont affichées dans l'ordre décroissant.
3. Une info bulle doit pouvoir être ajoutée sur n'importe quel point du graphique. Dans SBat3, l'utilisateur peut visualiser les données détaillées d'un point du graphique.
4. L'utilisateur doit pouvoir masquer chaque série (ligne, colonne, points, ...). Fonctionnalité inexistante dans SBat3 mais pratique lorsque beaucoup de données sont affichées sur le graphique.

⁸ <http://code.google.com/p/flot/>

⁹ <http://www.jqplot.com/>

¹⁰ <http://www.highcharts.com/>

¹¹ <http://code.google.com/apis/charttools/>

- (optionnel) L'utilisateur doit pouvoir exporter le graphique en format PDF ou en image. Fonctionnalité inexistante dans SBat3 mais pratique lorsque l'utilisateur désire sauvegarder les graphiques sur son poste de travail.

Critères d'implémentation

- Le support et la documentation
- La communauté : forum ou Wiki
- Adaptabilité et accessibilité du code et options offertes par la librairie

Radar

Toutes les librairies graphiques offrant à peu de chose près les mêmes fonctionnalités, une pondération sous forme de radar a été effectuée pour chaque librairie à l'aide des critères précédemment cités.

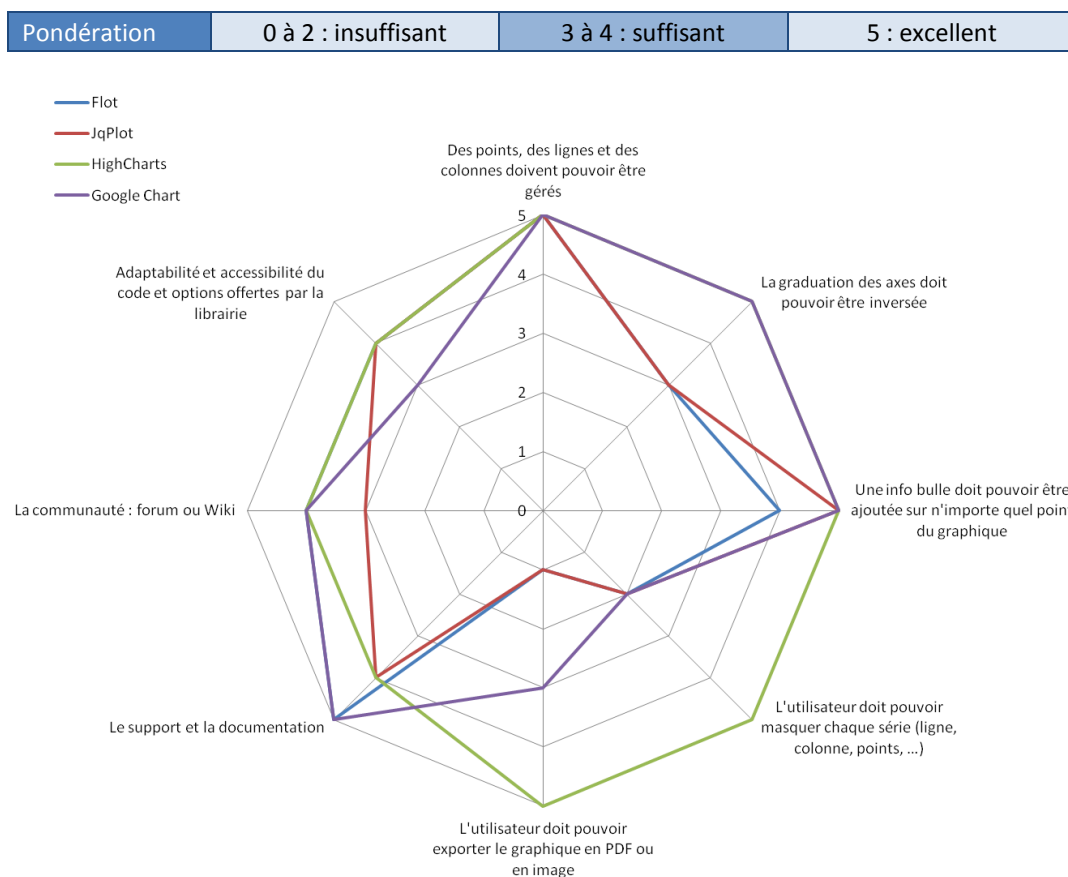


Figure 11 : Librairie graphique – Radar pour le choix final

En analysant ce radar, nous constatons que la librairie HighCharts se rapproche le plus des critères demandés. En effet, cette dernière est la seule à posséder toutes les fonctionnalités demandées telles que la possibilité d'exporter en format PDF ou en image le graphique affiché ou encore la possibilité de masquer une série donnée.

De plus, elle possède une documentation complète qui permet de visualiser toutes les propriétés des objets JavaScripts nécessaires au développement. Une communauté d'utilisateurs est également très active sur le forum de la librairie.

Highcharts Options Reference

<< [Go to the Highcharts home page](#)
 Navigate the structure by clicking plus and minus. Click the blue object names for details.

The options object

```

var chart = new Highcharts.Chart({
  chart: {
    alignTicks          : true,
    backgroundColor     : null,
    borderColor         : "#4572A7",
    borderRadius        : 5,
    borderWidth         : 0,
    className           : "",
    defaultSeriesType   : "line",
    events: [...],
    height              : null,
    ignoreHiddenSeries : true,
    inverted             : false,
    margin              : [50, 50, 60, 80],
    plotBackgroundColor : null,
    plotBackgroundImage : null,
    plotBorderColor     : "#C0C0C0",
    plotBorderWidth    : 0,
    plotShadow          : false,
    shadow              : false,
    showAxes            : false,
    style               : null,
    width               : null,
    zoomType            : ""
  },
  colors: [...]
});
      
```

chart

Options regarding the chart area and plot area as well as general chart options.

alignTicks : Boolean	When using multiple axes, the ticks of two or more opposite axes will automatically be aligned by adding ticks to the axis or axes with the least ticks. This can be prevented by setting <code>alignTicks</code> to false. If the grid lines look messy, it's a good idea to hide them for the secondary axis by setting <code>gridLineWidth</code> to 0. Defaults to <code>true</code> .
backgroundColor : Color	The background color or gradient for the outer chart area. Defaults to <code>null</code> .
borderColor : Color	The color of the outer chart border. The border is painted using vector graphic techniques to allow rounded corners. Defaults to <code>"#4572A7"</code> .
borderRadius : Number	The corner radius of the outer chart border. Defaults to 5.
borderWidth : Number	The pixel width of the outer chart border. The border is painted using vector graphic techniques to allow rounded corners. Defaults to 0.
className : String	A CSS class name to apply to the charts container <code>div</code> , allowing unique CSS styling for each chart. Defaults to <code>""</code> .
defaultSeriesType : String	The default series type for the chart. Can be one of line, spline, area, areaspline, column, bar, pie and scatter. Defaults to <code>"line"</code> .
events : chart.events	Event listeners for chart events. Defaults to <code>.</code>
height : Number	An explicit height for the chart. By default the height is calculated from the offset height of the containing element. Defaults to <code>null</code> .

Figure 12 : Librairie graphique - Highcharts Options Reference

Source : highcharts.com

Un manuel d'utilisation est aussi disponible pour aider et former les nouveaux utilisateurs au fonctionnement de la librairie.

How To Use

1. Installation

1. Highcharts uses either jQuery or MooTools for some common JavaScript tasks. In addition, Internet Explorer needs ExCanvas which emulates the Canvas element. You need to include these JavaScript files in the section of your web page. If you already have jQuery or MooTools included, you can skip the first one.

```

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"
type="text/javascript"></script>
<script src="/highcharts/highcharts.js" type="text/javascript"></script>
<!--[if IE]>
  <script src="/highcharts/excanvas.compiled.js" type="text/javascript"></script>
<![endif]-->
      
```

Figure 13 : Librairie graphique - Highcharts manuel d'utilisation

Source : highcharts.com

Pour finir, la librairie offre une collection d'options d'affichage fort intéressante dont l'implémentation reste facile d'approche.

En conclusion et sur la base des tests et des analyses effectués, le choix de la solution s'est arrêté sur la librairie JavaScript Highcharts [9] dont l'usage est libre pour toute utilisation non commerciale.

4.4 Implémentation

4.4.1 Processus

Le schéma ci-dessous explique le fonctionnement global du module, chaque point étant détaillé dans les chapitres suivants du document.

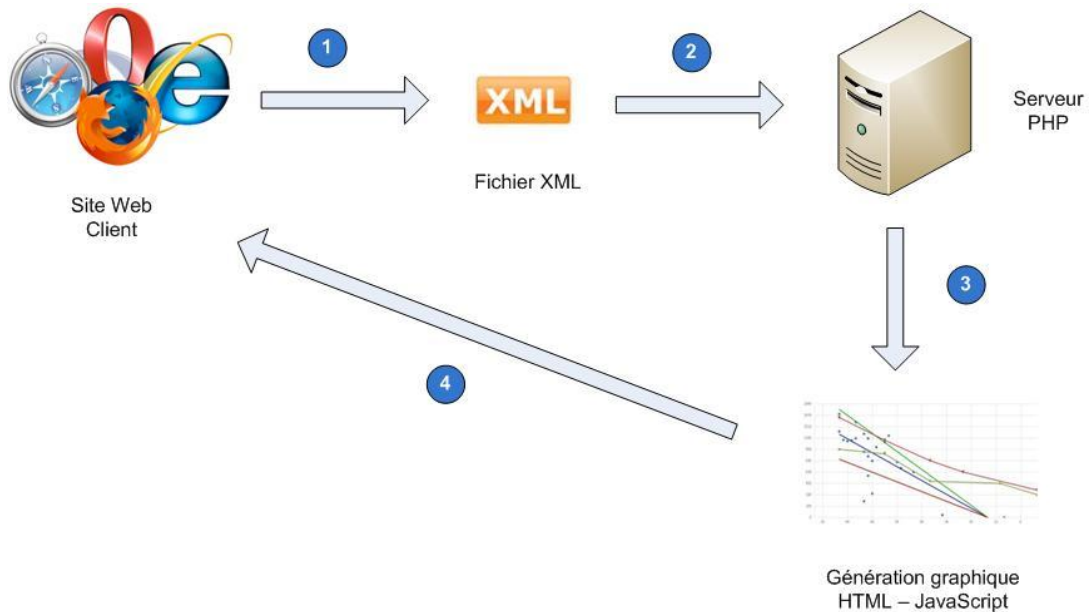


Figure 14 : Module graphique – Processus

1. Le site client web récolte les données nécessaires pour la génération du graphique. Ces données seront transmises au serveur PHP sous forme d'un fichier XML. Le format de l'XML peut être récupéré en consultant le schéma XSD mis à disposition sur le serveur PHP.
2. Le client interroge l'adresse internet du serveur PHP en lui transmettant le chemin de l'XML précédemment généré. Exemple :
`http://adresse_ip_serveur/libraryGraphique.php?xml=url_chemin_xml_du_client.`
3. Le serveur PHP récupère les informations contenues dans l'XML et génère le code du graphique en implémentant la librairie JavaScript Hightcharts.
4. Le code HTML/JavaScript est retourné au client par le serveur. Le client récupère simplement le buffer résultant de l'accès à la page web citée au point 2.

4.4.2 Communication client-serveur

XML

La communication XML a pour rôle la transmission des données entre le client et le serveur. L'avantage de ce type de communication est que le langage de programmation du client peut être totalement différent de celui du serveur, en l'occurrence le PHP.

L'XML demandé peut être consulté dans sa totalité à l'annexe B.1. Il se divise en trois parties : les propriétés du graphique, les données des axes et les données des séries à afficher.

```

<?xml version="1.0" encoding="UTF-8"?>
<Flot>
  <Properties>...</Properties>
  <Axes>
    <XAxis>...</XAxis>
    <YAxes>
      <YAxis>...</YAxis>
      <YAxis>...</YAxis>
      <YAxis>...</YAxis>
    </YAxes>
  </Axes>
  <Series>...</Serie>
    <Serie>
      <Properties>...</Properties>
      <Points>...</Points>
    </Serie>
    <Serie>
      <Properties>...</Properties>
      <Points>...</Points>
    </Serie>
  </Series>
</Flot>
  
```

Figure 15 : Module graphique – Données XML

Propriétés du graphique

Propriété	Utilité
Width	Longueur du graphique en pixel
Height	Hauteur du graphique en pixel
ZoomType	Zoom effectué sur X = l'axe X, Y = axe Y, XY = deux axes
BackgroundColor	Couleur de fond du graphique
BorderColor	Couleur de la bordure du graphique
BorderRadius	Angle de l'arrondi de la bordure du graphique
StyleCss	Style CSS de l'info bulle affichée lors du passage de la souris sur un point du graphique
BorderWidth	Epaisseur de la bordure du graphique
Animation	Si 1, un effet "fondu" est appliqué pour l'apparition de chaque ligne
Margin	Marges autour du graphique
Title	Définition du titre du graphique ainsi que sa couleur, grandeur et police d'écriture
Plot	Définition du quadrillage du graphique : épaisseur, couleur et couleur de fond
Legend	Définition des propriétés de la légende : alignement vertical/horizontal, bordure, ombre, couleur, ...

Tableau 2 : Module graphique – XML – Propriétés du graphique



Les axes

Un graphique doit posséder au moins un axe X et un ou plusieurs axes Y. Chaque axe possède les propriétés suivantes :

Propriété	Utilité
Label	Texte du label, de l'unité (ex : Degrés-heure [Kd/Sem]), la couleur, la police d'écriture et la distance qui sépare le label du graphique.
Unit	Valeur minimale, maximale, saut, couleur et police d'écriture des graduations
Grid	Couleur et épaisseur des lignes de l'axe
Line	Couleur et épaisseur de la ligne de l'axe où se trouvent les graduations
Opposite	Seulement pour axes Y : Si 0 = graduation et label à gauche du graphique, si 1 = graduation et label à droite du graphique
Offset	Seulement en cas d'axes multiples : distance entre les axes

Tableau 3 : Module graphique – XML – Propriétés des axes

Les séries

On entend par séries, les données affichées sur le graphique.

Propriété	Utilité
Legend	Titre de la série
Type	Type de la série : ligne, barre, colonne, points
RegressiveLine	En cas de nuage de points, possibilité d'afficher une ligne de régression
VariationLine	En cas de ligne de régression, possibilité d'afficher deux lignes représentant la variation de x% des points Y par rapport aux points X
YAxis	Seulement en cas d'axes multiples : détermine à quel axe Y correspondant la série.
Color	Couleur de la série
Points	Liste des points de la série : coordonnée x, coordonnée y et une information quelconque sous forme de texte

Tableau 4 : Module graphique – XML – Propriétés des séries

XSD

Un schéma XSD (XML Schema Definition) permet d'une part de définir la structure d'un document XML mais également de vérifier si le format de ce dernier est correct. Un schéma XSD a donc été créé pour permettre au client de connaître le format de l'XML à transmettre au serveur. Le schéma XSD peut être consulté dans sa totalité à l'annexe B.2.

Ci-dessous, un extrait de l'XML définissant les options de l'affichage du titre du graphique et la partie correspondante du schéma XSD.

XML	XSD
<pre> <Title> <Text>Graphique Chauffage pour Castalieeee</Text> <Align>center</Align> <Color>#3E576F</Color> 16px "Lucida Grande", "Lucida Sans Unicode", Verdana, Arial, Helvetica, sans-serif <Margin>10px 0px 0 0</Margin> </Title> </pre>	<pre> <xsd:complexType name="TitleType"> <xsd:sequence> <xsd:element name="Text" type="xsd:string" /> <!-- left / center / right --> <xsd:element name="Align"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="left" /> <xsd:enumeration value="right" /> <xsd:enumeration value="center" /> </xsd:restriction> </xsd:simpleType> </xsd:element> <xsd:element name="Color" type="xsd:string" /> <xsd:element name="Font" type="xsd:string" /> <!-- 10px 0px 0 0 --> <xsd:element minOccurs="0" name="Margin" type="xsd:string" /> </xsd:sequence> </xsd:complexType> </pre>

Figure 16 : Module graphique – Exemple de code XML/XSD

Ci-dessous, un extrait de l'XML définissant le type de série affichée et la partie correspondante du schéma XSD définissant toutes les options possibles.

XML	XSD
<pre> <Type>scatter</Type> </pre>	<pre> <xsd:element name="Type"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="area" /> <xsd:enumeration value="areaspline" /> <xsd:enumeration value="bar" /> <xsd:enumeration value="column" /> <xsd:enumeration value="line" /> <xsd:enumeration value="pie" /> <xsd:enumeration value="scatter" /> <xsd:enumeration value="spline" /> </xsd:restriction> </xsd:simpleType> </xsd:element> </pre>

Figure 17 : Module graphique – Exemple de code XML/XSD

4.4.3 Traitement des données

Introduction

Ce chapitre présente tout d'abord l'architecture des fichiers mis en place sur le serveur PHP.

Ensuite, il présente de quelle façon les données XML transmises par le client sont réceptionnées et utilisées pour générer le graphique.

Architecture fichiers

Le répertoire `_lib/class` contient le système de classes PHP permettant de stocker les informations contenues dans le XML envoyé par le client.

Le répertoire `_lib/js` contient les bibliothèques JavaScript highcharts et jQuery utiles pour la génération des graphiques.

Le fichier `_config.inc.php` regroupe les différentes variables d'environnement du projet. Lors d'un éventuel déplacement physique du projet, les changements interviendront uniquement dans ce fichier.

Le fichier `index.php` est le fichier que le client interroge en lui donnant en paramètre le lien de l'XML contenant les données du graphique. Ce fichier instancie le système de classes PHP et y stocke les données provenant de l'XML. Puis en utilisant les objets PHP instanciés, il génère le code JavaScript nécessaire à la bibliothèque highcharts pour l'affichage du graphique.

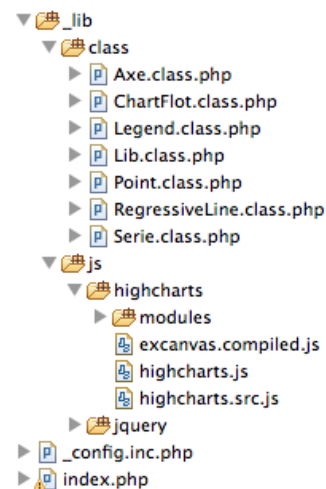


Figure 18 : Module graphique – Architecture fichiers

Génération du graphique

Les données XML reçues par le serveur PHP sont récupérées à l'aide de l'API PHP SimpleXML [10]. En donnant à cette librairie le chemin de l'XML, cette dernière renvoie un objet PHP de type SimpleXMLElement. Ensuite, le parcours de l'objet et la recherche d'informations sont facilités grâce à des méthodes telles que `SimpleXMLElement::attributes` ou encore `SimpleXMLElement::children`. Il est également possible d'effectuer des requêtes XPath à l'aide de la méthode `SimpleXMLElement::xpath`.

Finalement, les données collectées dans l'objet SimpleXML sont stockées dans un système d'objets PHP dont la structure est similaire à celle de l'XML :

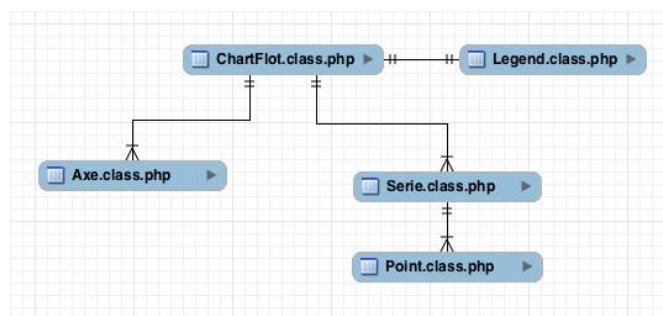


Figure 19 : Module graphique – Diagramme de classes

L'étape suivante consiste à générer le code HTML et JavaScript qui sera retourné au client et qui permettra d'afficher le graphique. Ci-dessous un exemple du code JavaScript demandé par la librairie Highcharts pour l'affichage des séries du graphique et sa construction à l'aide du système d'objets PHP.



Figure 20 : Module graphique – Exemple de code PHP/JavaScript

4.5 Exemples d'utilisations

Adresse du serveur hébergeant le module de génération de graphiques :

- <http://153.109.141.33/chart/>

Adresse de l'XML du client contenant les données du graphique :

- <http://153.109.141.33/chart/download/xml/regressive.xml>

4.5.1 PHP

En PHP, le client peut utiliser la fonction `file_get_contents()` afin de récupérer le contenu d'une page web puis l'afficher.

```
<?php
$xml = "http://153.109.124.94/chart/download/xml/regressive.xml" ;
$src = "http://153.109.124.94/chart/index.php" ;

$buffer = file_get_contents($src.'?xml='.$xml) ;
echo $buffer ;
?>
```

4.5.2 ASPX

En ASPX, il est également possible de récupérer le contenu d'une page web puis l'afficher :

```
<%
Dim xml = "http://153.109.124.94/chart/download/xml/regressive.xml"
Dim src = "http://153.109.124.94/chart/index.php"

Dim GotothisURL = src + "?xml=" + xml

Dim GetConnection = CreateObject("Microsoft.XMLHTTP")

GetConnection.Open("get", GotothisURL, False)
GetConnection.Send()
Dim ResponsePage = GetConnection.responseText

Response.Write(ResponsePage)
GetConnection = Nothing
%>
```

4.5.3 HTML

Une autre solution envisageable, purement HTML, est l'utilisation de la balise HTML `<iframe>` qui permet d'inclure un site web externe dans sa propre page web :

```
<iframe
src="http://153.109.124.94/chart/index.php?xml=http://153.109.124.94/c
hart/download/xml/histogram1.xml" scrolling="no" frameborder="0"
width="1000" height="450"></iframe>
```


4.6 Exemples de graphiques

Ci-dessous différents graphiques générés à l'aide du module de génération de graphiques.

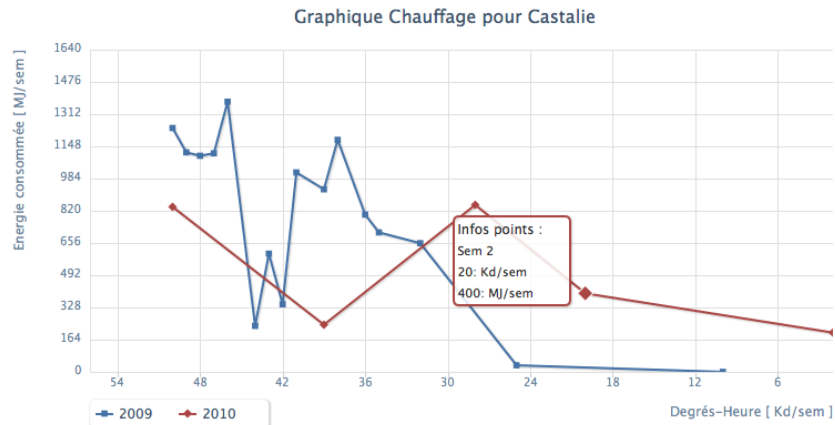


Figure 21 : Module graphique – Exemple d'un graphique avec des lignes

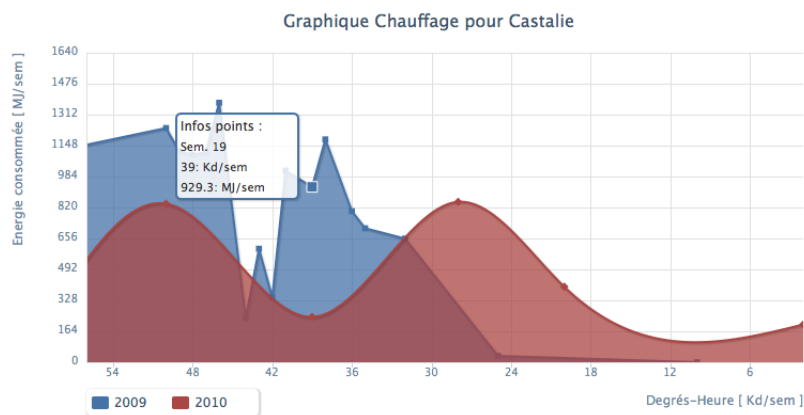


Figure 22 : Module graphique – Exemple d'un graphique avec des aires

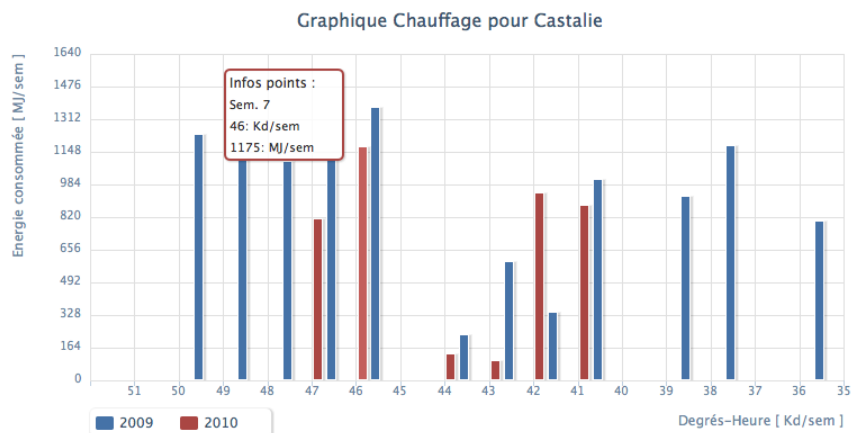


Figure 23 : Module graphique – Exemple d'un graphique avec des colonnes

Ci-dessous un graphique comportant un nuage de points avec la ligne de régression correspondante ainsi que deux lignes de variations.

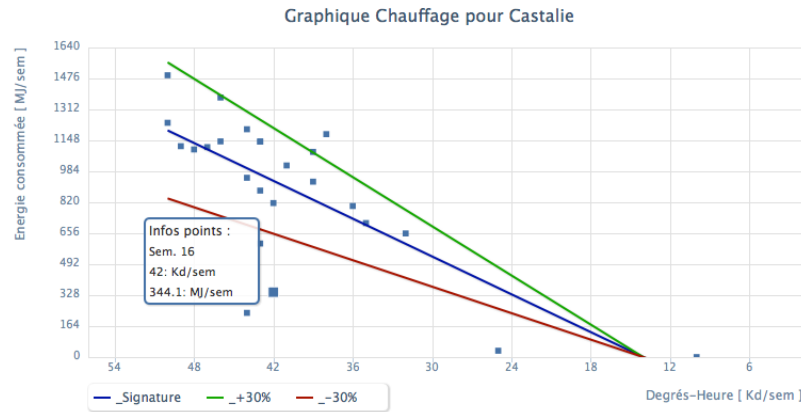


Figure 24 : Module graphique – Exemple d'un graphique avec une droite de régression

Ci-dessous un graphique possédant l'option de zoom. L'utilisateur peut sélectionner la zone qu'il désire agrandir.

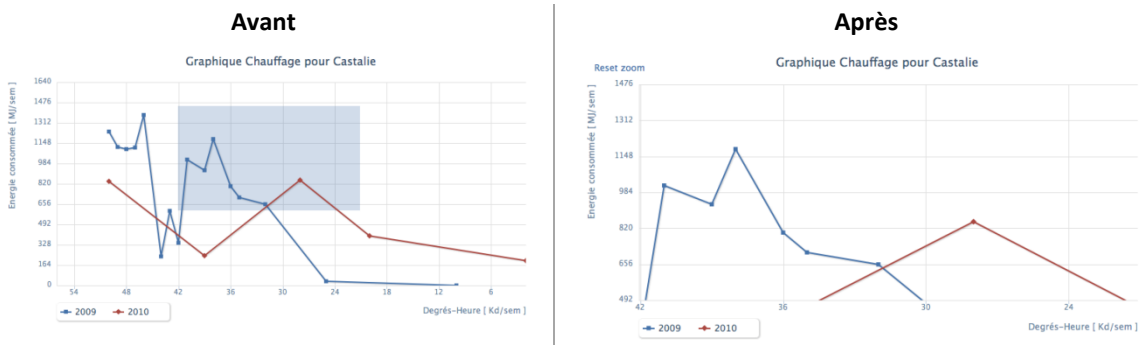


Figure 25 : Module graphique – Exemple d'un graphique avec l'option zoom

Ci-dessous, un graphique possédant plusieurs axes Y.

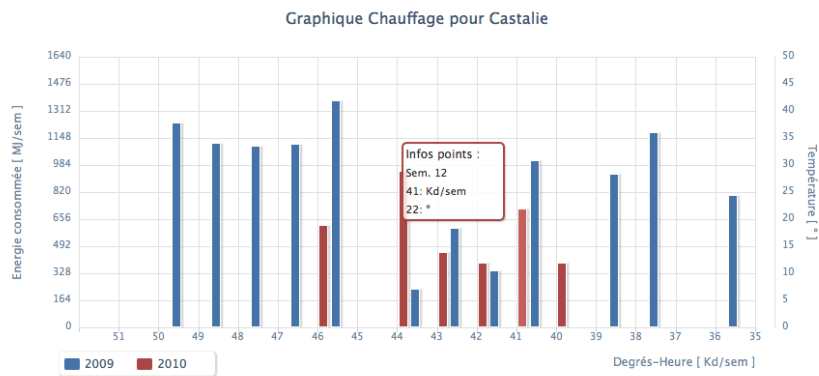


Figure 26 : Module graphique – Exemple d'un graphique avec axe Y multiple

Au clic sur la légende d'une série, il est possible de la faire disparaître :

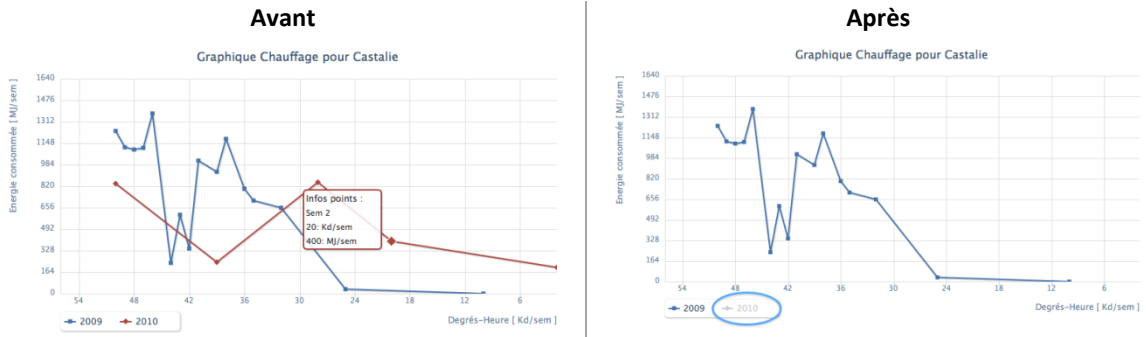


Figure 27 : Module graphique – Exemple d'un graphique avec une série masquée

L'utilisateur peut exporter et imprimer le graphique :

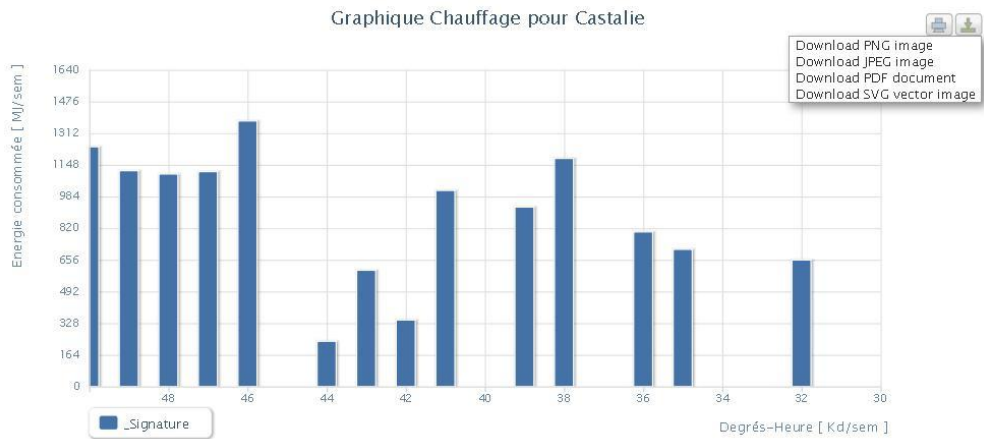


Figure 28 : Module graphique – Exemple d'un graphique avec possibilité d'exportation

5 Module de génération d'étiquettes énergétiques

5.1 Introduction

Ce chapitre décrit comment le module de génération d'étiquettes énergétiques a été réalisé.

Il introduit les différents acteurs œuvrant à la génération d'étiquettes énergétiques sur le marché Suisse et en déduit qu'il n'est pas possible d'implémenter un algorithme de calcul d'étiquettes mais de devoir se concentrer sur l'affichage de ces dernières.

Puis, il justifie les choix technologiques, indique de quelle façon un développeur pourra utiliser le module et propose des exemples d'étiquettes énergétiques réalisées.

5.2 Contexte

Depuis 1950, la consommation énergétique suisse a quadruplé et croît en moyenne de 2% par année. Si aucune mesure corrective n'est entreprise afin de diminuer cette augmentation, la consommation pourrait augmenter de 60% d'ici les 30 années à venir. Le parc immobilier suisse, composé de 1'450'000 habitations, compte environ 950'000 habitations individuelles datant d'avant 1980 et nécessitant une rénovation thermique. Sachant qu'une rénovation thermique du bâtiment permet de réduire de 30 à 70% la consommation d'énergie en chauffage, ceci représente un potentiel d'économie substantiel à l'échelle de la consommation Suisse [11].

Les gouvernements deviennent de plus en plus attentifs à la dimension environnementale. En Suisse, lors de la session d'hiver 2008, le gouvernement a relevé de 14 à 100 millions de francs la somme inscrite au budget 2009 pour les mesures de promotion dans le domaine de l'utilisation de l'énergie et des rejets de chaleur. Dans cette somme, notons qu'au maximum 18 millions de francs seront attribués à un programme de sensibilisation destiné aux propriétaires immobiliers, qui pourront bénéficier de conseils énergétiques personnalisés et faire établir un certificat de performance énergétique pour leur bâtiment. La visite d'un conseiller, y compris l'établissement du certificat énergétique, coûte environ 1'200 francs, dont 1'000 francs seront pris en charge par le programme de sensibilisation. Les cantons, en collaboration avec SuisseEnergie, organiseront ce programme et la formation des conseillers [12].

De plus, selon Novatlantis¹² le chauffage des bâtiments et la consommation électrique sont, en moyenne, responsables d'environ 40% du bilan énergétique des ménages et produisent plus de 20% des émissions de gaz à effet de serre. Par ailleurs et comme le démontre la Figure ci-dessous, les bâtiments ont un large potentiel d'amélioration en termes d'impact énergétique.

¹² <http://www.novatlantis.ch/>

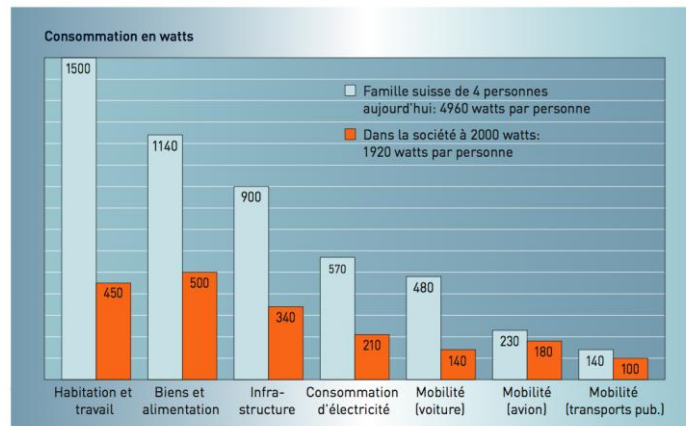


Figure 29 : Module étiquette – Consommation en watts par personne et par pays

Source : novatlantis.ch

Dans ce contexte environnemental, le Conseil fédéral a approuvé en juin 2009 un texte relatif à la modification de la loi sur l'énergie à l'intention du Parlement. L'objectif de cette révision partielle était de créer les bases légales nécessaires à l'établissement d'un certificat de performance énergétique pour les bâtiments qui soit uniforme pour toute la Suisse. Ces bases légales ont depuis lors été inscrites dans la loi. Toutefois, les cantons restent libres de rendre cette norme contraignante. Actuellement, aucun canton n'a encore déclaré le CECB® obligatoire, même si certains d'entre eux examinent cette question [13].



CERTIFICAT ÉNERGÉTIQUE CANTONAL DES BÂTIMENTS

Figure 30 : Module étiquette – Logo CECB

Source : cecb.ch

CECB¹³ signifie certificat énergétique cantonal des bâtiments. Il permet de démontrer sous forme d'une étiquette énergétique la consommation d'un bâtiment lorsqu'il est utilisé de manière standard. Sa réalisation a été soutenue par la Confédération et l'Association suisse des

propriétaires fonciers. Le calcul de l'étiquette CECB se base sur la norme SIA 380/1[14] qui permet de calculer l'efficacité de l'enveloppe du bâtiment et l'efficacité énergétique globale. L'efficacité de l'enveloppe du bâtiment exprime la qualité de la protection thermique de cette enveloppe, qui englobe les murs extérieurs, la toiture et le plancher de l'étage chauffé le plus bas, ainsi que les fenêtres. L'efficacité énergétique globale prend en compte le chauffage et les autres installations techniques dont la génération de chaleur, notamment pour l'eau chaude sanitaire [15].

¹³ <http://www.cecb.ch/>

L'application de la norme SIA 380/1 est un exercice complexe. Une étude réalisée en 2006 [16] par le service cantonal de l'énergie du canton de Vaud, montre que sur l'ensemble des dossiers traités, 60% des dossiers ne sont pas conformes à la norme SIA 380/1. Les erreurs les plus fréquentes sont le calcul de la surface de l'enveloppe¹⁴ ou encore le calcul de la valeur des ponts thermiques¹⁵. Cependant, l'étude note que le potentiel d'économies et le rapport "argent investi / kWh économisé" serait énorme si la norme était appliquée.

Outre le CECB, la campagne Display¹⁶, lancée en 2003 en Europe par des experts de 20 villes et municipalités européennes et introduite dès 2004 en Suisse, permet, tout comme le CECB l'affichage d'une étiquette énergétique permettant de mesurer différentes consommations. Cependant, Display n'est pas en opposition mais en complémentarité avec le CECB. En effet, Display permet l'affichage des performances énergétiques des bâtiments publics, ce qui n'est pas le cas du CECB qui concerne principalement les bâtiments privés. De plus et d'après SuisseEnergie [17], le CECB est plus «lourd» à réaliser que Display car nécessitant une visite du bâtiment et la récolte d'un grand nombre de données techniques.



Figure 31 : Module étiquette – Logo Display

Source : display-campaign.org

Après consultation de la norme SIA 380/1 et analyse des différentes solutions existantes sur le marché Suisse, nous avons constaté qu'il n'était soit pas possible de maîtriser les connaissances nécessaires au calcul d'une étiquette dans le temps imparti à ce travail de bachelor, soit qu'il n'était pas possible de connaître les algorithmes utilisés par les acteurs du marché. Dès lors, il a été décidé de laisser tomber l'élaboration de cet algorithme demandant la connaissance d'une multitude de facteurs techniques que nous ne maîtrisons pas.

En conclusion, le module de génération d'étiquettes énergétiques reçoit en entrée des valeurs/lettres pré calculées et se concentre sur la visualisation des étiquettes énergétiques depuis un site web sous forme d'images ou de documents PDF. Le premier objectif a donc été de définir des bibliothèques permettant de générer ces images et documents PDF.

¹⁴ Surface de l'enveloppe : peut comprendre en plus de la surface de référence, la couverture du toit, le revêtement de façade ou la surface vitrée

¹⁵ http://herve.silve.pagesperso-orange.fr/deperditions/ponts_thermiques.htm

¹⁶ <http://www.display-campaign.org/>

5.3 Librairie de génération d'images et de documents PDF

5.3.1 Librairie PDF

Pour la génération de documents PDF, deux types de solutions pouvaient être implémentées : une librairie PHP ou un logiciel tiers. Afin d'effectuer des tests sur ces différentes librairies, une représentation HTML d'une étiquette a été implémentée pour comparer les résultats des différentes solutions testées :



Concernant les librairies PHP permettant de générer des documents PDF, les librairies suivantes ont été testées : fpdf, html2pdf et dompdf.

```

<?php
require('fpdf.php');

$pdf=new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'Hello World!');
$pdf->Output();
?>
  
```

Figure 32 : Module étiquette – Exemple d'implémentation de la librairie fpdf

Source : fpdf.org

Fpdf¹⁷ (version 1.6) est une librairie dans laquelle l'utilisateur doit positionner manuellement chaque élément présent sur le document PDF. Ci-contre, un exemple de code PHP utilisant la librairie fpdf. Le fait de devoir placer chaque élément n'apporte que peu de flexibilité lors du développement ainsi que lors d'éventuelles futures modifications visuelles de l'étiquette. Dans ce sens, nous

n'avons pas jugé utile d'essayer de représenter une étiquette à l'aide de cette librairie. Dès lors, les recherches se sont dirigées vers des librairies permettant de transformer du code HTML en format PDF telles que html2pdf.

¹⁷ <http://www.fpdf.org/>



Html2pdf¹⁸ (version 4.01) est une librairie PHP qui étend la librairie fpdf. Son utilisation simple demande en entrée un code HTML et renvoie comme sortie un document PDF. Cependant, les tests effectués avec le code HTML représentant l'étiquette énergétique n'ont pas été concluants :

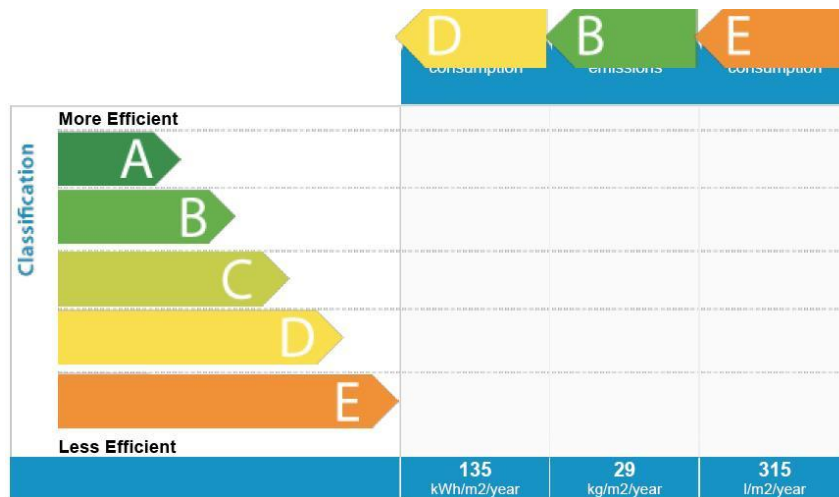


Figure 33 : Module étiquette – Test de la librairie Html2pdf

En effet, plusieurs éléments de mise en page tels que les positions absolues ou les images en arrière plan ne sont pas interprétés de manière optimale. Bien entendu, le code HTML testé aurait pu être simplifié pour obtenir un meilleur rendu mais dans l'optique de sélectionner la meilleure librairie, le code HTML n'a pas été modifié.

La dernière librairie PHP testée est DomPdf¹⁹ (version 5.1) qui permet également de transformer un code HTML en document PDF. Les résultats du test ont été décevants :

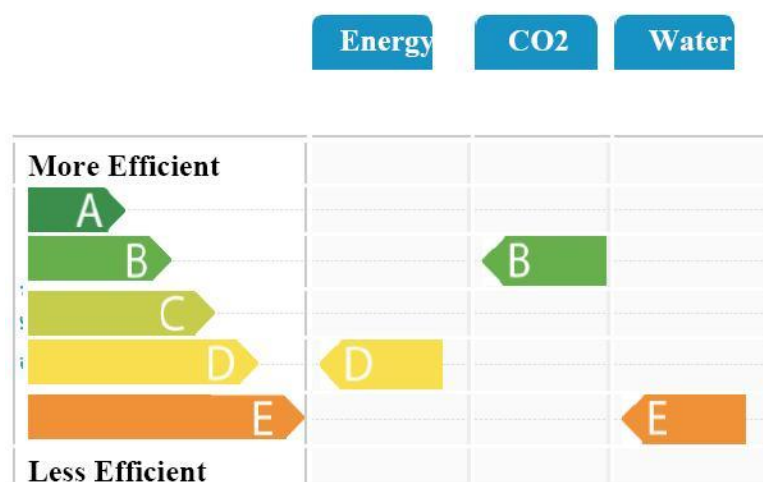


Figure 34 : Module étiquette – Test de la librairie DomPDF

¹⁸ <http://html2pdf.fr/>

¹⁹ <http://www.digitaljunkies.ca/dompdf/>

Effectivement, le rendu du document PDF était loin de la représentation HTML désirée. Notons cependant que la version testée de DomPDF est une version beta.

Pour finir, le seul logiciel tiers testé se nomme wkhtmltopdf. Wkhtmltopdf est un utilitaire en ligne de commande utilisant le moteur de rendu WebKit²⁰. L'utilitaire reçoit en entrée le lien de la page web à transformer en document PDF ainsi que le chemin de destination du fichier qui sera généré. Les résultats obtenus par l'intermédiaire de wkhtmltopdf ont été très satisfaisants :

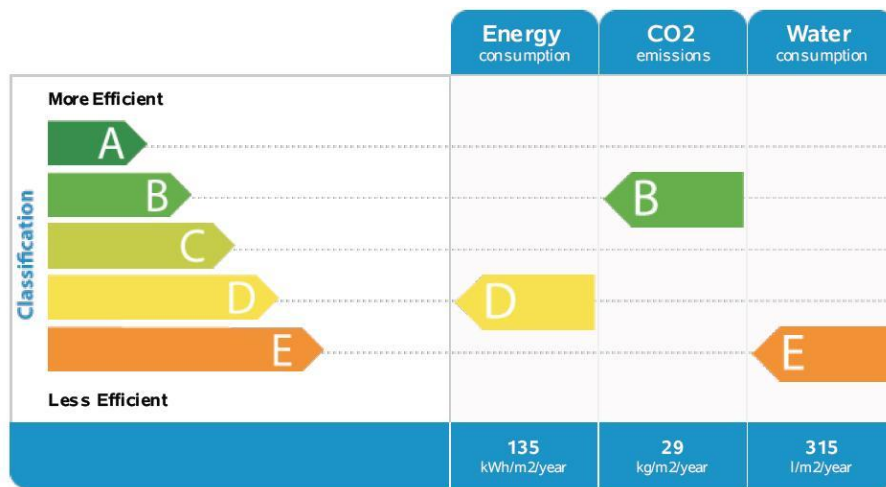


Figure 35 : Module étiquette – Test du WebKit wkhtmltopdf

En plus du fait de pouvoir générer un document PDF quasiment identique au rendu HTML, cet utilitaire est multiplateforme (Windows, Apple, Linux) et est accompagné d'une documentation complète ainsi que d'un forum de discussions.

En conclusion et à la lumière des précédents arguments, la solution retenue pour la génération des documents PDF est le WebKit wkhtmltopdf [18].

5.3.2 Librairie image

La solution retenue pour la génération d'images est l'utilitaire wkhtmltoimage. Cet utilitaire en ligne de commande possède les mêmes caractéristiques que wkhtmltopdf et permet donc de transformer un code HTML en une image. L'utilisation similaire des deux utilitaires ainsi que les très bons résultats obtenus nous ont convaincus dans notre choix et ne nous ont pas poussés à rechercher d'autres solutions pour la génération des images.

²⁰ WebKit : bibliothèque de fonctions permettant aux développeurs d'intégrer facilement un moteur de rendu de pages Web dans leurs logiciels. Source: <http://fr.wikipedia.org/wiki/WebKit>

5.4 Implémentation

5.4.1 Processus

Le schéma ci-dessous explique le fonctionnement global du module, chaque point étant détaillé dans les chapitres suivants du document.

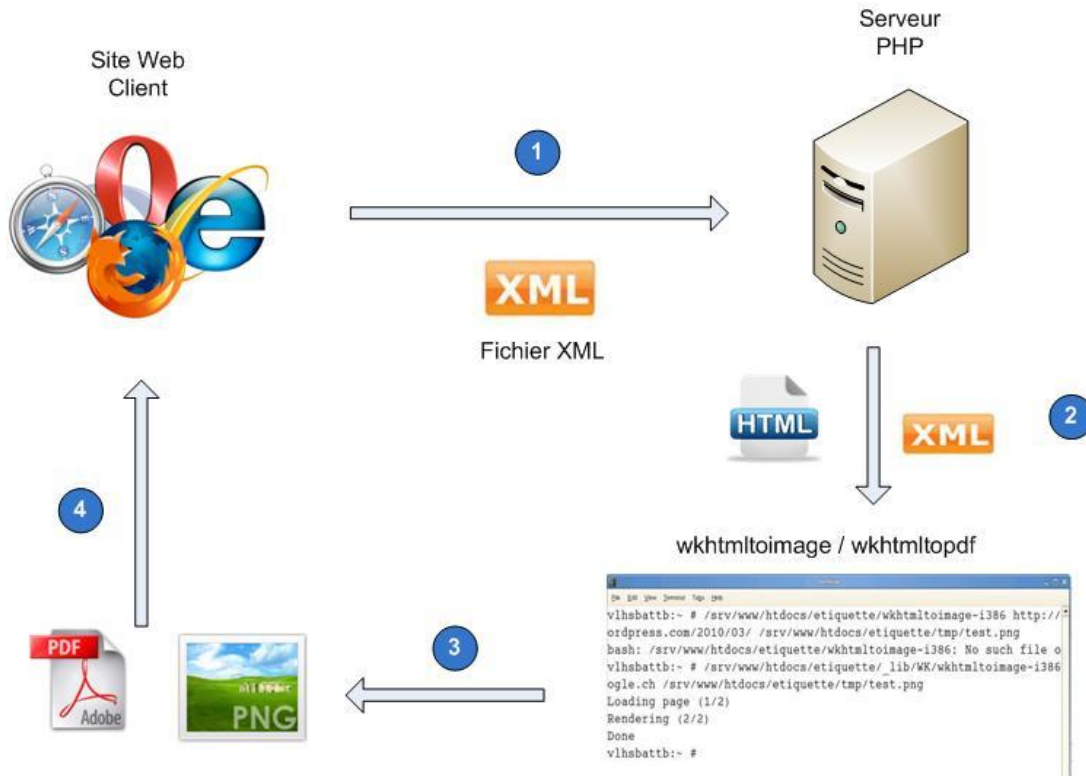


Figure 36 : Module étiquette – Processus

1. Le client web récolte les données nécessaires à la génération de l'étiquette et les transmet au serveur PHP sous forme de données XML. Exemple:
http://adresse_serveur/libraryEtiquettes.php?xml=url_chemin_xml_du_client.
2. Le serveur PHP récupère les informations et exécute l'utilitaire en ligne de commande wkhtmltopdf ou wkhtmltoimage. Les données transmises à l'utilitaire sont le lien de la page Internet contenant le code HTML de l'étiquette énergétique avec en paramètre le chemin du fichier XML du client ainsi que la destination et le nom du fichier à générer.
3. En fonction de la demande du client, l'utilitaire wkhtmltopdf ou wkhtmltoimage génère soit un document PDF soit une image qu'il stocke temporairement sur le serveur.
4. Finalement le serveur transmet l'image ou le document PDF au client et supprime les fichiers temporaires du serveur.



5.4.2 Communication client-serveur

Comme pour le module de génération de graphiques, la technologie retenue pour la transmission des données est l'XML. Rappelons que l'avantage de ce type de communication est que le langage de programmation du client peut être totalement différent de celui du serveur, en l'occurrence le PHP. L'XML demandé peut être consulté dans sa totalité à l'annexe C.1 suivi de son schéma à l'annexe C.2. L'XML se décompose en trois parties :

Propriétés de l'étiquette

Propriété	Utilité
NumberOfLetter	Nombre de lettres affichées. Maximum de A à G
LabelTop	Label se trouvant au dessus de la liste des lettres, exemple : "More Efficient"
LabelBottom	Label se trouvant au dessous de la liste des lettres, exemple : "Less Efficient"
LabelLeft	Label se trouvant à gauche de la liste des lettres, exemple : "Classification"

Tableau 5 : Module étiquette – XML – Propriétés de l'étiquette

Propriétés pour le PDF

Ces propriétés sont uniquement utilisées lors de la génération de documents PDF.

Propriété	Utilité
Title	Titre de l'étiquette
Text	Texte descriptif de l'étiquette
InfosBatTitle	Titre du tableau "Informations sur le bâtiment"
InfosBat	Informations sur le bâtiment présentées sous forme d'un tableau. Pour être générique, toutes les informations sont stockées dans un seul champ et dans le format suivant : le séparateur "#" permet de distinguer les cellules et le séparateur "###" permet de distinguer les lignes. Exemple : titreDeLaDonnee1#valeurDeLaDonnee1### titreDeLaDonnee2#valeurDeLaDonnee2.
Logo	Image se trouvant au sommet droit du PDF

Tableau 6 : Module étiquette – XML – Propriétés pour le document PDF

Types d'énergies

Une étiquette peut contenir plusieurs types d'énergie. Pour chacune d'entre elles, les propriétés suivantes sont demandées :

Propriété	Utilité
Title	Nom de l'énergie
Subtitle	Autres indications libres sur l'énergie
Value	Valeur de l'énergie, format numérique
Unit	Unité de l'énergie, exemple : kWh/m2/year
Letter	Lettre calculée : de A à G

Tableau 7 : Module étiquette – XML – Propriétés des types d'énergie

5.4.3 Traitement des données

Introduction

Ce chapitre présente tout d'abord l'architecture des fichiers mis en place sur le serveur PHP.

Ensuite, les utilitaires `wkhtmltopdf` et `wkhtmltoimage` se basent sur un code HTML pour générer un document PDF ou une image, ce chapitre décrit premièrement la génération du code HTML représentant l'étiquette énergétique pour terminer avec l'exécution des deux utilitaires générant le document PDF ou l'image à transmettre au client.

Architecture des fichiers

Le répertoire `_lib/class` contient le système de classes PHP permettant de stocker les informations contenues dans le XML envoyé par le client.

Le répertoire `_lib/WK` contient les deux utilitaires `wkhtmltopdf` et `wkhtmltoimage` permettant la génération des documents PDF et des images.

Le répertoire `tmp` stocke temporairement les fichiers générés par les deux utilitaires précédemment cités.

Le répertoire `tpl` contient les fichiers `pdf.php` et `image.php`. Les utilitaires de génération des fichiers interrogent ces deux fichiers en leur donnant en paramètre le lien de l'XML contenant les données de l'étiquette. Ces fichiersinstancient le système de classes PHP et y stockent les données provenant de l'XML. Puis en utilisant les objets PHP instanciés, ils génèrent le code HTML représentant l'étiquette. Ce code sera récupéré par les deux utilitaires pour générer un document PDF ou une image.

Le fichier `img.php` est le fichier que le client interroge lorsqu'il désire créer une étiquette énergétique en format image. Le fichier `pdf.php` est le fichier que le client interroge lorsqu'il désire créer une étiquette énergétique en format PDF. Le client fait appel à ces fichiers en leur donnant en paramètre le lien de l'XML contenant les données de l'étiquette. Ces fichiers exécutent les deux utilitaires `wkhtmltoimage` et `wkhtmltopdf` en leur transmettant le lien de l'XML contenant les données de l'étiquette. Finalement ils transmettent le fichier généré au client.

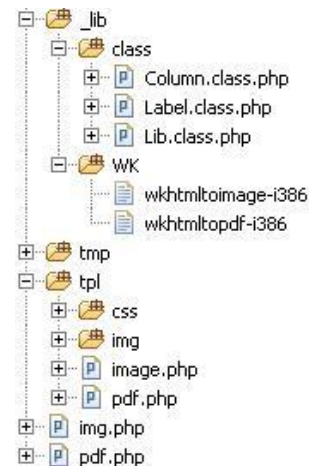


Figure 37 : Module étiquettes – Architecture fichiers

Création de l'étiquette énergétique

Comme pour le module précédent, les données XML reçues par le serveur PHP sont récupérées à l'aide de l'API PHP SimpleXML qui renvoie un objet PHP de type SimpleXMLElement. Ce dernier est alors parcouru et les informations collectées sont stockées dans un système d'objets PHP cachant ainsi toute la complexité de l'accès aux données. Ce système d'objets comprend deux classes. La classe "Label" gère les données de l'étiquette et peut posséder plusieurs "Column" qui représentent les différentes énergies affichées :

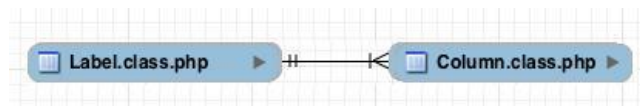


Figure 38 : Module étiquette – Diagramme de classes

L'étape suivante consiste à générer le code HTML représentant l'étiquette énergétique. Ci-dessous un exemple d'une partie du code HTML généré et de sa construction à l'aide du système d'objets PHP.

PHP	HTML
<pre> <tr class="result"> <td colspan="2">&nbsp;</td> <?php \$xmlData = simplexml_load_file(\$xml); \$data = new Label(\$xmlData) ; foreach(\$data->cols as \$col) { <? <td class="label"> <?php echo \$col->value;?>
 <?php echo \$col->unit;?> </td> <?php } <? </tr> </pre>	<pre> <tr class="result"> <td colspan="2">&nbsp;</td> <td class="label"> 135
 kWh/m2/year </td> <td class="label"> 29
 kg/m2/year </td> <td class="label"> 315
 l/m2/year </td> </tr> </pre>

Résultat web

	135 kWh/m2/year	29 kg/m2/year	315 l/m2/year
--	--------------------	------------------	------------------

Figure 39 : Module étiquette – Exemple de code PHP/HTML

Génération du fichier

Deux types de fichiers peuvent être générés : des images et des documents PDF. Comme déjà mentionné, les utilitaires `wkhtmltoimage` et `wkhtmltopdf` sont respectivement utilisés.

Notre serveur PHP fonctionnant sur un système d'environnement Linux Suse, pour pouvoir utiliser les deux utilitaires demandés, nous nous sommes d'abord assurés que les programmes X11, gcc et openssl soient installés. (Exemple de commande utile à l'installation : `"zypper install openssl"`). Ensuite et après avoir téléchargé les paquets d'installation sur le site web²¹ des deux utilitaires, la commande `"tar -xjvf wkhtmltopdf-0-10.tar.bz2"` nous a permis de décompresser les paquets sur le serveur. Pour finir, un test à l'aide de la commande `"/wkhtmltopdf-i386 http://www.google.ch test.pdf"` nous a permis de vérifier la bonne marche de la solution installée.

Afin d'obtenir son image ou son document PDF, le client fait appel à la page `pdf.php` ou `img.php` se trouvant sur le serveur en lui donnant en paramètre le chemin de l'XML contenant les données de l'étiquette. Le processus engendré par la page PHP interrogée est le suivant :

1. Définition du dossier temporaire où stocker le fichier généré:

Si PDF :

```
$dir = tempnam('tmp/', 'pdf') ;
```

Si image :

```
$dir = tempnam('tmp/', 'image/png') ;
```

2. Définition de l'url de la page permettant de générer le code HTML de l'étiquette. Le chemin de l'XML est passé en paramètre (La génération du code HTML de l'étiquette a été expliquée au chapitre précédent).

Si PDF :

```
$url = "tpl/pdf.php?xml=".$pathXML ;
```

Si image :

```
$url = "tpl/image.php?xml=".$pathXML ;
```

3. Exécution de l'utilitaire `wkhtmltopdf` ou `wkhtmltoimage`.

Si PDF :

```
exec("wkhtmltopdf-i386 --lowquality --margin-left 20  
\"\"".$url."\" \" ".$dir) ;
```

Si image :

```
exec("wkhtmltoimage-i386 --crop-w ".$width."  
\"\"".$url."\" \" ".$dir.".png") ;
```

²¹ <http://code.google.com/p/wkhtmltopdf/>

4. Envoi du fichier au client

Si PDF :

```
header('Content-type: application/pdf');  
header('Content-Disposition: attachment;  
filename="downloaded.pdf");  
echo file_get_contents($dir);
```

Si image :

```
header('Content-type: image/png');  
echo file_get_contents($dir);
```

5. Destruction du fichier temporaire

```
unlink($dir);
```

5.5 Exemples d'utilisations

Adresse du serveur hébergeant le module de génération d'étiquettes énergétiques :

- <http://153.109.124.94/etiquette/>

Adresse de l'XML contenant les données du graphique :

- <http://153.109.124.94/etiquette/download/xml/etiquette.xml>

Images

Ci-dessous, un exemple d'intégration d'une étiquette sous forme d'image dans une page HTML.

```

```

PDF

Ci-dessous, un exemple d'intégration d'une étiquette sous forme d'un lien HTML permettant de télécharger le document PDF.

```
<a  
href="http://153.109.124.94/etiquette/pdf.php?xml=http://153.109.124.94  
4/etiquette/download/xml/etiquette.xml">Télécharger le PDF</a>
```

Au clic sur le lien "Télécharger le PDF", l'utilisateur sera invité à télécharger le document PDF généré par le serveur.

5.6 Exemples d'étiquettes énergétiques

Images

Ci-dessous, deux exemples d'étiquettes générées dans un format image. La première image affiche les lettres de A à E avec deux énergies et la deuxième image les lettres de A à G avec 3 énergies. Ces différentes propriétés sont définies dans l'XML fourni par le client.

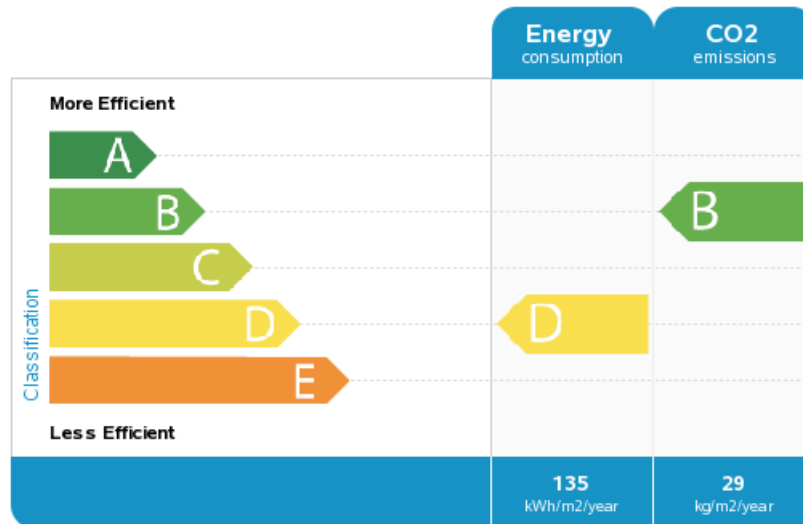


Figure 40 : Module étiquette – Exemple d'étiquette en format image 1

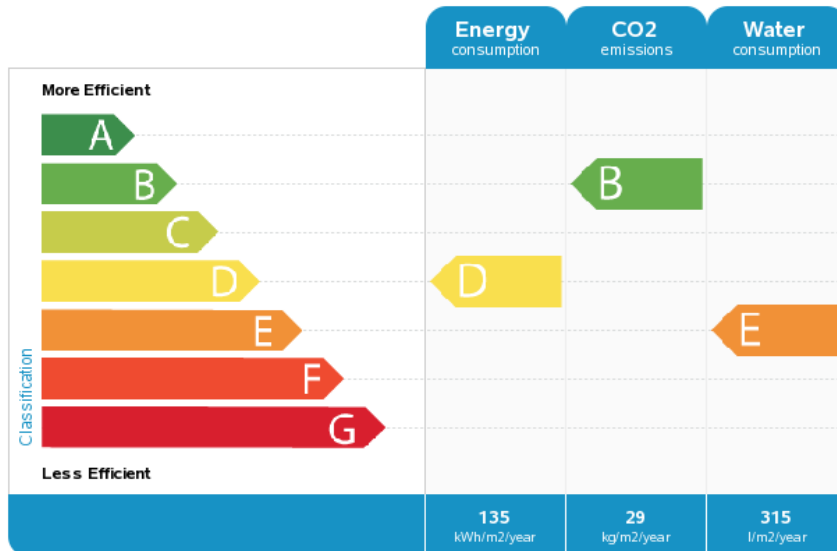


Figure 41 : Module étiquette – Exemple d'étiquette en format image 2

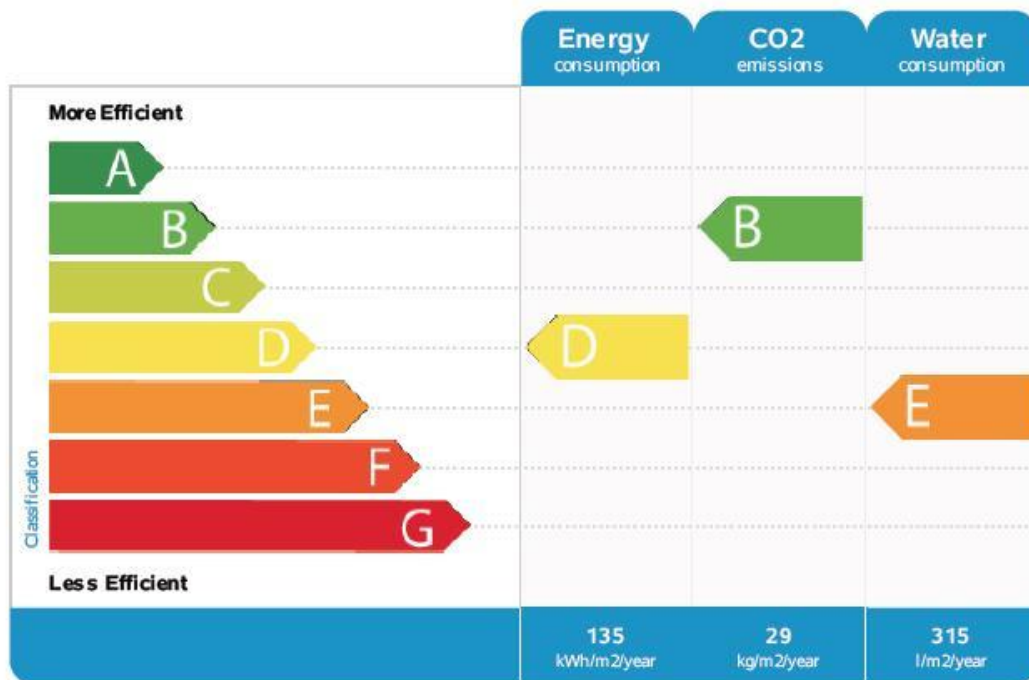
PDF

Ci-dessous, un exemple d'étiquette énergétique générée dans un document PDF. L'XML correspondant ainsi que le document PDF peuvent être consultés respectivement aux annexes C.1 et C.3.

Local pompiers



La production, le transport et la consommation de toutes les unités qui utilisent de l'énergie fossile sont prises en compte. L'indice climat caractérise la qualité de gaz à effet de serre émise par unité d'énergie consommés par le bâtiment.



Building Data	
Surface de référence	12125 m ²
Taux de renouvellement d'air	0.7 l/h
Surface de l'enveloppe	15602 m ²
Volume chauffé	35303 m ³

Figure 42 : Module étiquette – Exemple d'étiquette en format PDF

6 Déploiement

Actuellement, les deux modules sont disponibles sur un serveur Linux Suse atteignable depuis le réseau interne de la HES-SO Valais de Sierre à l'adresse suivante : <http://153.109.124.94/>

Si les services doivent être déplacés sur un autre serveur, il est nécessaire de prendre en considération la remarque suivante : le serveur sur lequel sont implémentés les modules doit pouvoir interpréter le PHP. Sur la machine actuelle, un serveur web Apache est utilisé.

Les modules ne sont rattachés à aucune base de données. Les seuls éléments qui doivent être déplacés sont donc les fichiers sources.

6.1 Module de génération de graphiques

En cas de déplacement de ce module, le serveur PHP doit autoriser les fonctions qui permettent d'accéder à des fichiers distants par le protocole HTTP. Cette autorisation permettra au logiciel client de récupérer le code généré par la page `index.php` qui se trouve à la racine du module. Le code généré par cette page permet l'affichage du graphique demandé par le client.

Pour autoriser ce genre de fonction, il est nécessaire de modifier le fichier de configuration PHP qui se nomme "`php.ini`". Notons que l'emplacement de ce fichier peut varier selon le serveur PHP qui est installé. A l'intérieur de ce fichier, il faut rechercher les lignes où sont inscrits les codes suivants : "`allow_url_fopen = Off`" et "`allow_url_include = Off`". La valeur de ces deux instructions doit être positionnée à "`On`".

Pour finir, afin de s'assurer de travailler avec des librairies à jour, ce module utilise la librairie JavaScript jQuery mise à disposition par Google. Si cet outil devait être déplacé dans un réseau fermé sans connections Internet, il serait nécessaire d'utiliser la librairie jQuery se trouvant dans le dossier du module : "`_lib/js/jquery`".

6.2 Module de génération d'étiquettes énergétiques

En cas de déplacement de ce module, il est nécessaire d'installer sur le serveur web les utilitaires `wkhtmltopdf` et `wkhtmltoimage` qui permettent de générer les fichiers demandés par le client. Les deux utilitaires peuvent être téléchargés à l'adresse suivante : <http://code.google.com/p/wkhtmltopdf/>.

Les utilitaires sont multiplateformes. Cependant, il est obligatoire d'installer certaines librairies pour qu'ils puissent fonctionner. La liste des librairies nécessaires pour chaque système d'exploitation est disponible sur le site web des utilitaires.

Pour finir, il est important de vérifier que l'utilisateur par lequel est lancée l'instance du serveur web possède les droits en écriture sur le dossier "`tmp`" se trouvant à la racine du module pour qu'il puisse y stocker et supprimer les fichiers temporaires.

7 Problèmes rencontrés

Ce chapitre dresse la liste des principaux problèmes survenus durant la réalisation du projet «Génération d'étiquettes énergétiques pour les bâtiments».

Compréhension des problèmes énergétiques

N'ayant aucune connaissance dans le domaine énergétique, nous aurions pu nous contenter de traiter des données sans même comprendre leurs significations.

Cependant, notre rôle dans ce projet a été de comprendre en partie la problématique énergétique pour amener des solutions adaptées aux besoins du client final.

L'analyse du logiciel SBat3 ainsi que la rencontre avec un collaborateur spécialiste du domaine énergétique nous a permis d'assimiler les concepts de base nous permettant de réaliser ce projet.

Recherche et formation

Les connaissances requises pour accomplir ce projet vont au delà du cadre des cours que nous avons pu suivre à la HES-SO.

Bien que durant nos études à la HES-SO nous ayons eu la chance de découvrir les technologies web, nous n'avons jamais assimilé de manière approfondie les langages de programmation, tels que le CSS ou le JavaScript, qui sont utiles à la création d'un site web dynamique.

Dès lors, passablement de temps a été investi dans la formation et la documentation sur les langages PHP, JavaScript et CSS.

Serveur Linux

Les deux modules développés dans le cadre de ce projet ont été implémentés sur un serveur Linux Suse. L'installation du serveur web Apache n'a pas posé de problème particulier.

Les problèmes sont survenus lors de l'installation du WebKit wkhtmltopdf permettant la génération des étiquettes énergétiques dans le cadre du deuxième module développé. La documentation fournie pour l'installation du WebKit et des bibliothèques requises à son fonctionnement se rapportant principalement aux distributions Linux Debian/Ubuntu et n'étant pas familier avec le monde Linux, la mise en place fut plus longue que prévue.

Il nous a donc été nécessaire de nous documenter sur la distribution Linux Suse afin de pouvoir mener à bien l'installation du WebKit.

8 Conclusion

Toutes les fonctionnalités décrites dans le cahier des charges ont été réalisées dans le respect des contraintes imposées. Les deux modules développés sont disponibles sur un serveur interne à la HES-SO Valais et pourraient, en l'état, être utilisés pour réimplémenter les graphiques vieillissants du logiciel SBat3 ou être implémentés dans d'autres projets en cours de réalisation à la HES-SO Valais.

Les deux modules sont livrés avec une documentation complète consultable sur le wiki créé pour ce projet ou directement dans le code source commenté.

Deux séances se sont tenues afin de transmettre le travail accompli aux collaborateurs de la HES-SO Valais enclins à reprendre le projet.

D'un point de vue personnel

Ce projet m'a permis de m'initier au domaine de l'énergie car auparavant mes connaissances sur le sujet étaient fortement limitées. J'ai pu découvrir l'aspect environnemental lié à la consommation d'énergie dans un bâtiment et prendre conscience de la place importante qu'occupe l'aspect énergétique dans notre société.

D'un point de vue technique, ce travail de bachelor m'a donné la possibilité de perfectionner mes compétences en programmation web et d'approfondir mes connaissances de l'environnement Linux.

Durant ces trois mois, ce projet m'a permis de m'investir de manière individuelle dans un travail de recherche, de développer mes compétences et de mieux connaître mes limites. Sortir du cadre scolaire et se voir attribuer un projet concret furent une expérience enrichissante et motivante.

À la lumière de ce qui précède, il m'a paru pertinent d'implémenter de tels outils. En effet, dans le contexte environnemental actuel, le développement de logiciels de surveillance énergétique me paraît de nature à soutenir l'homme dans sa démarche pour réduire sa consommation d'énergie.

9 Bibliographie

- [1] Introduction à PHP. [Internet]. [version du 23 Juin 2010] : <http://www.commentcamarche.net/contents/php/phpintro.php3>.
- [2] Introduction à XML. [Internet]. [version du 24 Juin 2010] : <http://www.commentcamarche.net/contents/xml/xmlintro.php3>.
- [3] Introduction à l'HTML. [Internet]. [version du 24 Juin 2010] : <http://www.commentcamarche.net/contents/html/htmlintro.php3>.
- [4] HTML5 se dévoile. [Internet]. [version du 28 Juin 2010] : <http://www.alsacreation.com/article/lire/750-HTML5-nouveautes.html>.
- [5] Feuille de style CSS en ligne. [Internet]. [version du 28 Juin 2010] : <http://www.sitego.fr/feuille-de-style-css.html>.
- [6] Le langage Javascript. [Internet]. [version du 2 Juillet 2010] : <http://www.commentcamarche.net/contents/javascript/jsintro.php3>.
- [7] Browser Market Share. [Internet]. [version du 28 Juillet 2010] : <http://www.netmarketshare.com/browser-market-share.aspx?qprid=0>.
- [8] Monitoring énergétique des bâtiments. [Internet]. [version du 24 Mai 2010] : <http://iig.hevs.ch/valais/monitoring-energetique-batiments-mandat-sbat3.html>.
- [9] Highcharts - Librairie Graphique. [Internet]. [version de Juin - Juillet 2010] <http://highcharts.com/>.
- [10] SimpleXML. [Internet]. [version de Juin 2010] <http://php.net/manual/fr/book.simplexml.php>.
- [11] Ecobuilding - Rénovation thermique réussie. [Internet]. [version du 14 Juillet 2010] : <http://www.ecobuilding.ch/?link=content&cid=1.1&menu=whatis>.
- [12] Le Conseil fédéral favorable à un programme national d'assainissement des bâtiments. [Internet]. [version du 14 Juillet 2010] : <http://www.bfe.admin.ch/energie/00588/00589/00644/index.html?lang=fr&msg-id=25533>.
- [13] Certificat de performance énergétique pour les bâtiments. [Internet]. [version du 14 Juillet 2010] : <http://www.bfe.admin.ch/energie/00567/00569/00601/index.html?lang=fr#>.

- [14] Energie - Norme SIA 380/1 et cahier technique 2031. [Internet]. [version du 14 Juillet 2010] : http://www.sia.ch/f/actuel/news/20090318_nouveau.cfm.
- [15] Classes étiquette énergie. [Internet]. [version du 14 Juillet 2010] : <http://www.cecb.ch/Pages/Light/About/ClassificationPage.aspx>.
- [16] Evaluation de l'application de la norme SIA 380/1-2001. [Internet]. [version du 13 Juillet 2010] : <http://www.eben-energie.ch/rapport%20380-1/rapport%20380-1.pdf>.
- [17] Lettre d'info de SuisseEnergie pour les communes. [Internet]. [version de Juillet 2010] http://www.energiestadt.ch/f/joomla/pop_up.php?option=com_content&task=view&id=505.
- [18] Convert html to pdf using webkit (qtwebkit). [Internet]. [version de Juillet 2010] <http://code.google.com/p/wkhtmltopdf/>.
- [19] openSUSE. [Internet]. [version de Juillet 2010] <http://fr.opensuse.org/>.

10 Tables des illustrations

10.1 Tableaux

Tableau 1 : Gestion et suivi de projet.....	3
Tableau 2 : Module graphique – XML – Propriétés du graphique	21
Tableau 3 : Module graphique – XML – Propriétés des axes.....	22
Tableau 4 : Module graphique – XML – Propriétés des séries	22
Tableau 5 : Module étiquette – XML – Propriétés de l'étiquette	37
Tableau 6 : Module étiquette – XML – Propriétés pour le document PDF	37
Tableau 7 : Module étiquette – XML – Propriétés des types d'énergie.....	37

10.2 Images

Figure 1 : Choix technologiques – Schéma PHP.....	7
Figure 2 : Choix technologiques – Exemple XML	8
Figure 3 : Choix technologiques – Exemple de code HTML.....	9
Figure 4 : Choix technologiques – Exemple de code CSS	10
Figure 5 : Choix technologiques – Exemple de code JavaScript	11
Figure 6 : SBat3 – Graphique énergétique chauffage en Flash	13
Figure 7 : SBat3 – Graphique énergétique solaire en Flash	14
Figure 8 : SBat3 – Graphique énergétique bilan annuel en Flash	15
Figure 9 : SBat3 – Graphique énergétique chauffage en HTML	16
Figure 10 : SBat3 – Graphique énergétique année précédente en HTML.....	16
Figure 11 : Librairie graphique – Radar pour le choix final	18
Figure 12 : Librairie graphique - Highcharts Options Reference	19
Figure 13 : Librairie graphique - Highcharts manuel d'utilisation	19
Figure 14 : Module graphique – Processus.....	20
Figure 15 : Module graphique – Données XML	21
Figure 16 : Module graphique – Exemple de code XML/XSD	23
Figure 17 : Module graphique – Exemple de code XML/XSD	23
Figure 18 : Module graphique – Architecture fichiers	24
Figure 19 : Module graphique – Diagramme de classes	25
Figure 20 : Module graphique – Exemple de code PHP/JavaScript.....	25
Figure 21 : Module graphique – Exemple d'un graphique avec des lignes	27
Figure 22 : Module graphique – Exemple d'un graphique avec des arcs	27
Figure 23 : Module graphique – Exemple d'un graphique avec des colonnes.....	27
Figure 24 : Module graphique – Exemple d'un graphique avec une droite de régression	28



Figure 25 : Module graphique – Exemple d'un graphique avec l'option zoom	28
Figure 26 : Module graphique – Exemple d'un graphique avec axe Y multiple	28
Figure 27 : Module graphique – Exemple d'un graphique avec une série masquée	29
Figure 28 : Module graphique – Exemple d'un graphique avec possibilité d'exportation	29
Figure 29 : Module étiquette – Consommation en watts par personne et par pays	31
Figure 30 : Module étiquette – Logo CECB.....	31
Figure 31 : Module étiquette – Logo Display	32
Figure 32 : Module étiquette – Exemple d'implémentation de la librairie fpdf	33
Figure 33 : Module étiquette – Test de la librairie Html2pdf	34
Figure 34 : Module étiquette – Test de la librairie DomPDF.....	34
Figure 35 : Module étiquette – Test du WebKit wkhtmltopdf	35
Figure 36 : Module étiquette – Processus	36
Figure 37 : Module étiquettes – Architecture fichiers	38
Figure 38 : Module étiquette – Diagramme de classes.....	39
Figure 39 : Module étiquette – Exemple de code PHP/HTML	39
Figure 40 : Module étiquette – Exemple d'étiquette en format image 1	42
Figure 41 : Module étiquette – Exemple d'étiquette en format image 2	42
Figure 42 : Module étiquette – Exemple d'étiquette en format PDF	43

11 Tableau des annexes

À la suite de ce document se trouvent les annexes suivantes :

A. Gestion de projet

- A.1. Cahier des charges du projet
- A.2. Document de planification et de suivi de projet.
- A.3. Rapports hebdomadaires de travail

B. Module de génération de graphiques

- B.1. Exemple d'XML à envoyer au serveur pour la génération d'un graphique
- B.2. Schéma XSD permettant la définition de l'XML

C. Module de génération d'étiquettes énergétiques

- C.1. Exemple d'XML à envoyer au serveur pour la génération d'une étiquette énergétique
- C.2. Schéma XSD permettant la définition de l'XML
- C.3. Exemple d'étiquette énergétique en format PDF

D. Déclaration sur l'honneur

Annexe

A. Gestion de Projet

A.1 Cahier des charges du projet



Annexe

A. Gestion de projet

A.2 Document de planification et de suivi

Annexe

A. Gestion de projet

A.3 Rapports hebdomadaires de travail

Annexe

B. Module de génération de graphiques

B.1 Exemple d'XML à envoyer au serveur pour la génération d'un graphique

Annexe

B. Module de génération de graphiques

B.2 Schéma XSD permettant la définition de l'XML

Annexe

C. Module de génération d'étiquettes énergétiques

C.1 Exemple d'XML à envoyer au serveur pour la génération d'une étiquette énergétique

Annexe

C. Module de génération d'étiquettes énergétiques

C.2 Schéma XSD permettant la définition de l'XML



Annexe

C. Module de génération d'étiquettes énergétiques

C.3 Exemple d'étiquette énergétique en format PDF

Annexe

D. Déclaration sur l'honneur



Déclaration sur l'honneur

Sierro, le 16 août 2010.

Je déclare, par ce document, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré.

Hervé Sierro