

Acquisition et exploitations pédagogiques des données sur un système pédagogique

Bilan technique et éléments de développement



Une vingtaine d'établissements de la Région Centre sont équipés du véhicule électrique **TWIZY**. Afin d'exploiter ce support avec des élèves, nous avons souhaité pouvoir relever des informations du véhicule lors d'un trajet et mettre les données en ligne. C'est ce développement qui nous a servi de base pour ce compte rendu, dans l'objectif de transposition à tout autre système pédagogique dont les données de fonctionnement pourraient s'avérer utiles pour des exploitations pédagogiques avec ou sans le système dans le laboratoire. Nous tenterons ici une description usant de SysML, qui semble pertinente pour ce type de présentation liant matériel et logiciel.

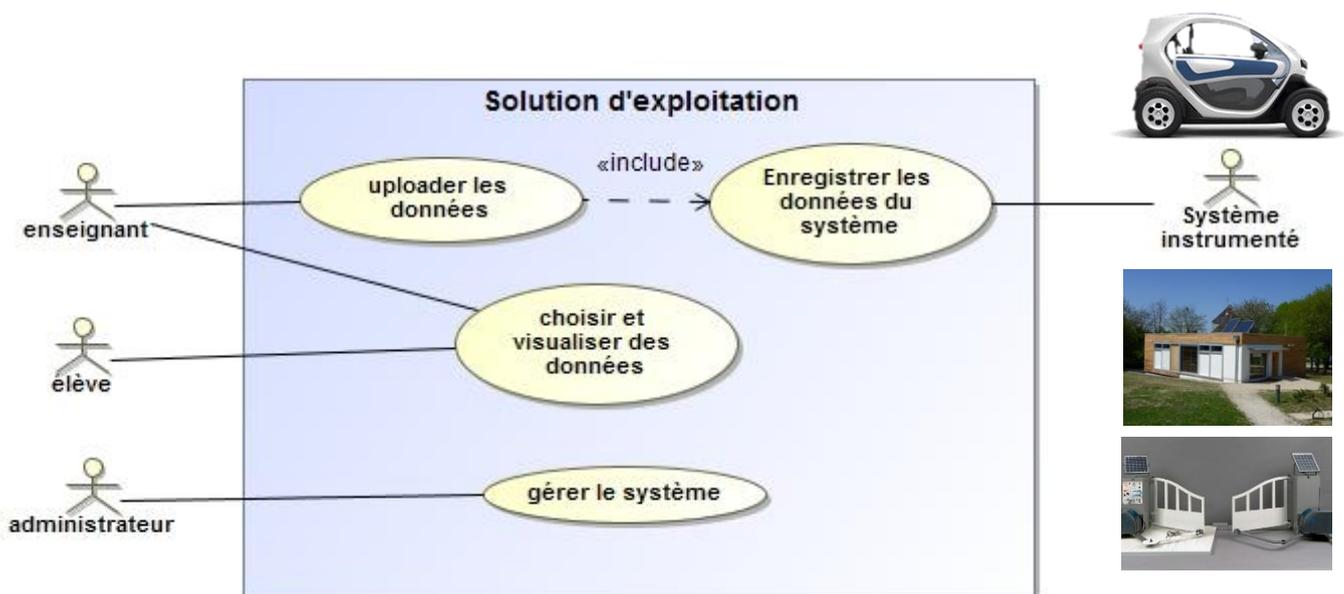
Fonctionnalités attendues

L'objectif de la solution à développer était de pouvoir mettre en ligne des données issues du véhicule électrique Twizy afin que les enseignants puissent les exploiter avec les élèves. Ces campagnes de mesures sont réalisées lorsque le véhicule roule afin d'obtenir des valeurs sur la vitesse, la charge de la batterie, etc.

On peut grossièrement indiquer les différentes phases d'utilisation de la solution et les résultats attendus :

- les mesures se font lorsque le véhicule est roulant. Elles se feront donc avec un ordinateur positionné dans la voiture. À la fin de la mesure, on possède un fichier Excel dont le format est donné en annexe (format csv).
- Ce fichier est uploadé depuis une interface web dédiée vers la base de données de l'application web. Il est traité par l'application web à la réception pour mettre les données dans la base.
- Pour l'exploitation pédagogique, l'application web aura une interface permettant de sélectionner des critères de recherche afin de pouvoir extraire des campagnes de mesures parmi toutes celles présentes dans la base. Une campagne pourra ensuite être sélectionnée puis affichée graphiquement en fonction du temps.

Il est possible de faire une synthèse de ce qui a été décrit grâce au diagramme de cas d'utilisation suivant :

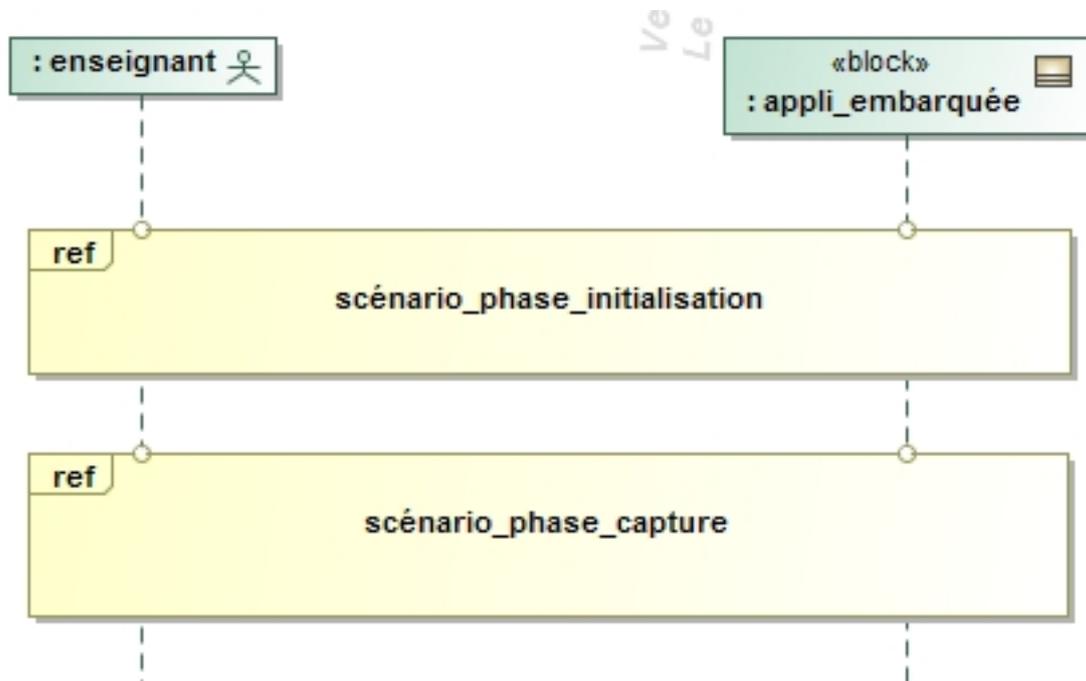


Il est à noter que ce diagramme est la base sur laquelle toute la suite du développement repose. Nous y voyons apparaître les différents acteurs et les différentes fonctionnalités attendues.

A ce stade, il est nécessaire de modéliser les comportements attendus de l'application avant de commencer à développer.

Scénarios nominaux

Scénario de capture des données :



Commentaires : afin de mieux faire apparaître le scénario nominal lors de la capture des données, il est noté ici deux phases d'utilisation détaillées ci-dessous.

Remarque : à ce stade, on peut noter sur les scénarios qu'il apparaît 2 éléments logiciels différents : l'application web et l'application embarquée. La suite va les détailler.

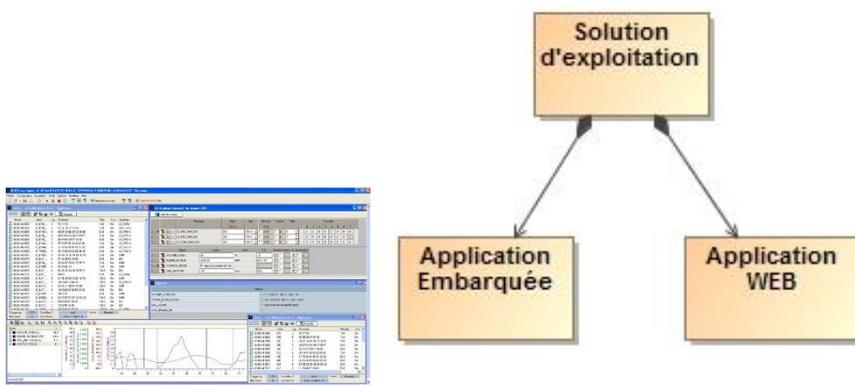
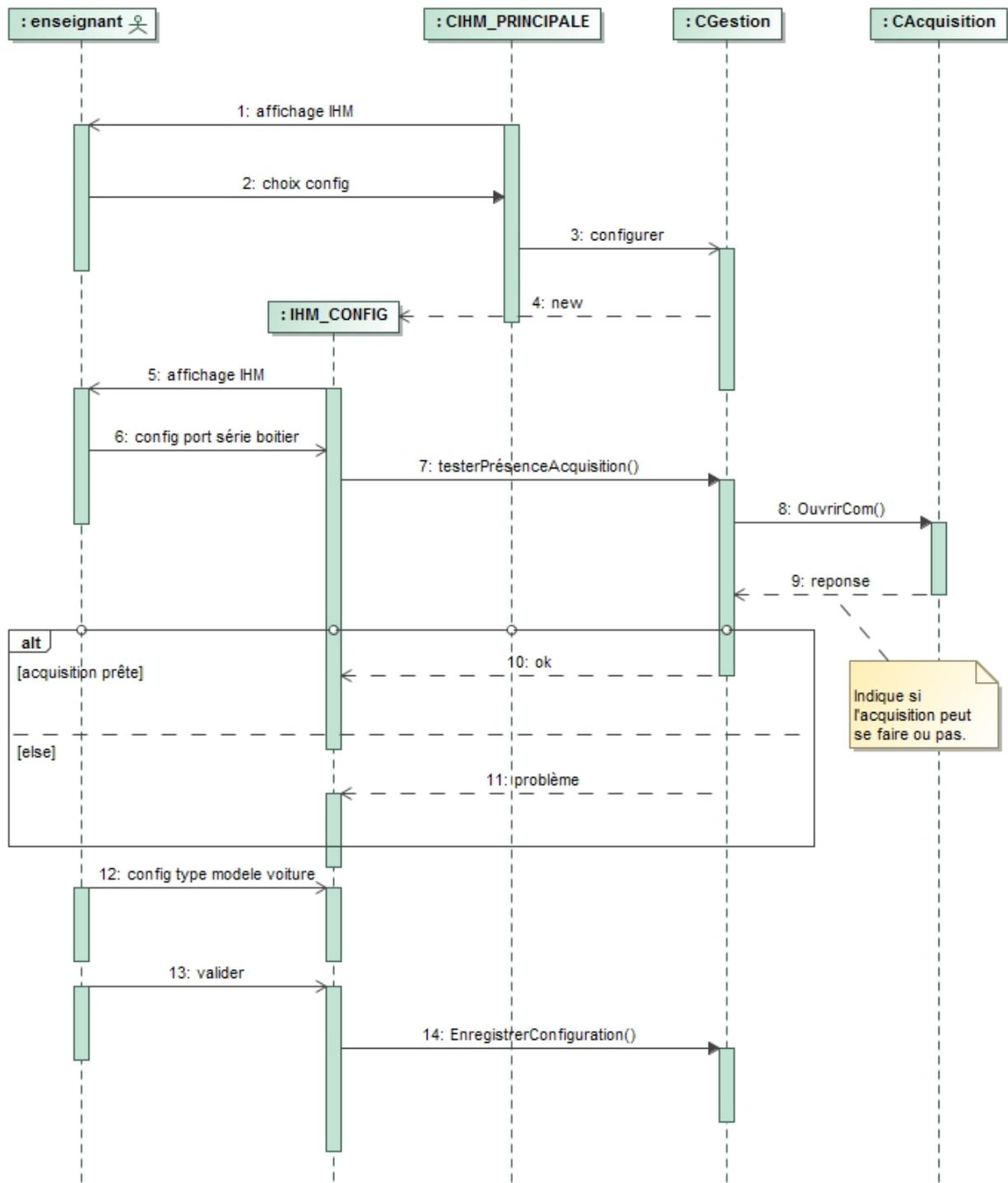


Figure 2 Solution d'acquisition en cours de développement chez Exxotest



Figure 1 Interface du prototype de l'application WEB

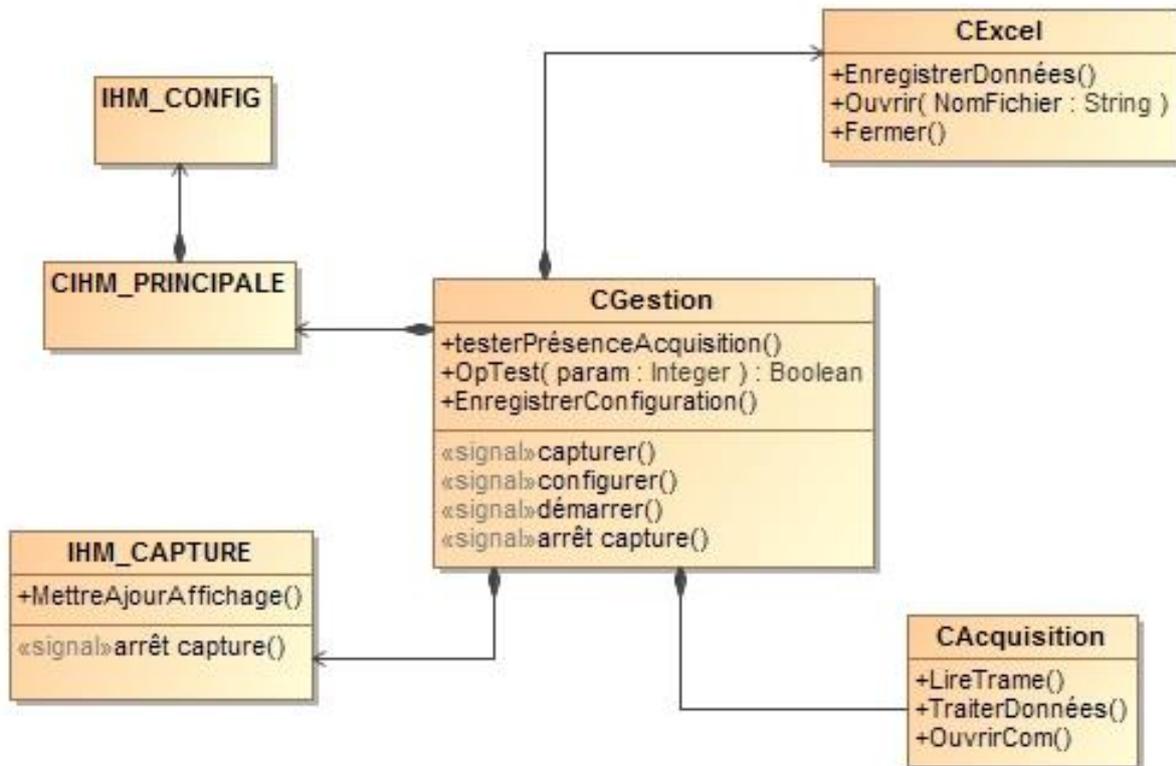
Application embarquée : scénario phase initialisation



Commentaires : dans l'idéal, il faudrait écrire un scénario où l'application embarquée est vue comme un tout (boite noire). Sauf qu'ici, il n'y a pas de difficultés particulières. On voit apparaître une classe IHM qui permet de dialoguer avec l'utilisateur (affichage et saisie) ainsi qu'une classe de gestion de l'application (cœur de l'application) et une classe de dialogue avec le matériel.

Il est important de bien scinder ces éléments pour ne pas aboutir à du code trop monolithique, c'est-à-dire difficile à mettre au point et maintenir.

Application embarquée : architecture niveau 1



Commentaires : les scénarios permettent de proposer le diagramme de classes ci-dessus. À gauche on trouve toutes les classes de dialogue avec les utilisateurs et à droite toutes les classes liées au matériel et à l'enregistrement de fichiers. Ils nous permettent aussi de faire émerger les différentes méthodes et signaux de chaque classe (l'objet n'est pas de détailler ici la différence entre ces deux concepts). À l'aide d'un logiciel de modélisation, la création des messages les ajoute automatiquement aux classes.

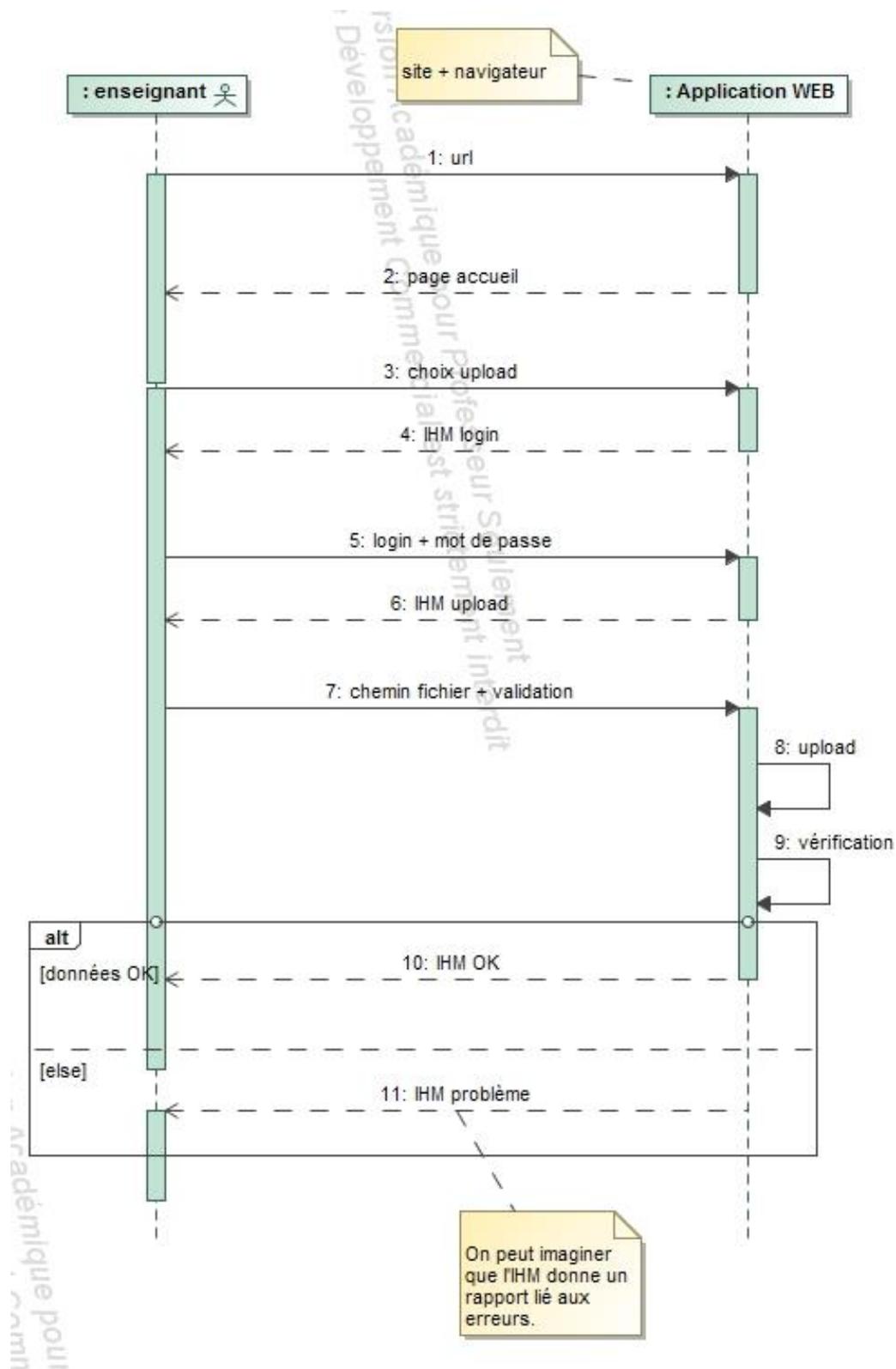
Remarque : toutes les méthodes n'apparaissent pas forcément dans les scénarios. Si on ne s'attache à ne tracer que les plus importants, alors il faudra compléter les classes de manière indépendante. Par exemple ici est complétée la classe **CExcel** avec les méthodes **Ouvrir()** et **Fermer()**.

Remarque : il est à noter ici que la classe **CAcquisition** est une classe générique. Il faudra donc adapter et modifier cette classe en fonction du système depuis lequel on souhaite effectuer des mesures. Dans notre cas, nous avons travaillé avec la TWIZY en nous branchant sur son bus CAN. Les données nous sont donc automatiquement envoyées sans avoir à les demander (ce qui est représenté ici). Par contre, sur un autre projet comme la maison PASSILAB, il faut aller interroger le système pour récupérer les données. Dans ce cas, la classe Acquisition aura un comportement légèrement différent dans le scénario de la page précédente.

On peut distinguer ainsi 3 cas :

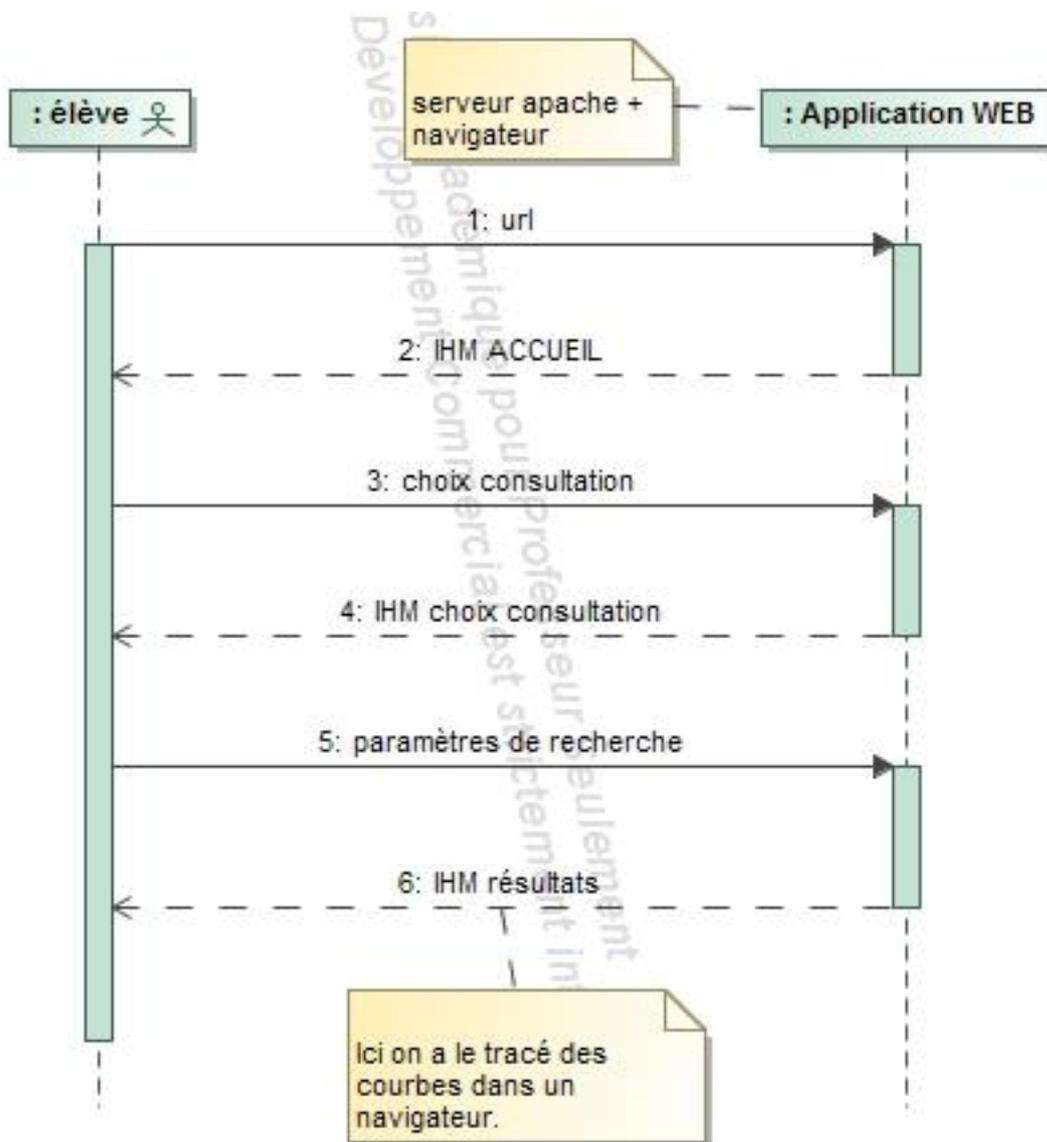
- *Système type TWIZY* : les informations sont envoyées par le biais d'un bus. Il faut donc pouvoir se brancher sur le bus par un moyen ou un autre (ici pour le cas du système TWIZY nous avons choisi le CANUSB qui est un moyen peu cher et assez facile à utiliser).
- *Système PASSILAB* : il existe une station contrôlant l'ensemble du système. Pour récupérer les données, il nous a fallu ajouter un module communicant par TCP/IP (en fait une passerelle bus KNX-> TCP/IP).
- *Système instrumenté où à instrumenter* : pour aller chercher les informations sur un système qui ne possède pas de moyen de communication, il faudra lui ajouter par exemple une solution de type microcontrôleur (Arduino par exemple) pour gérer les capteurs ainsi qu'un bus de communication. Selon les contraintes, les informations seront envoyées automatiquement par le μC ou bien elles seront récupérées à la demande par une application distante.

Application web : scénario Uploader les données



Commentaires : scénario simple avec une seule alternative. L'upload doit vérifier à la fin que le fichier contient des informations viables (pas de valeurs erronées).

Application web : Consultation des données



Commentaires : pas de problème particulier soulevé, donc pas d'alternative possible.



Application web : gérer le système

Pas de diagramme particulier ici car cela ressemble fortement à ce qui est fait pour la consultation. Une fois le choix de la consultation fait, le gestionnaire tombe sur une nouvelle IHM lui permettant de réaliser toutes les tâches d'administration souhaitées (élément classique dans une application web). On retrouvera par exemple :

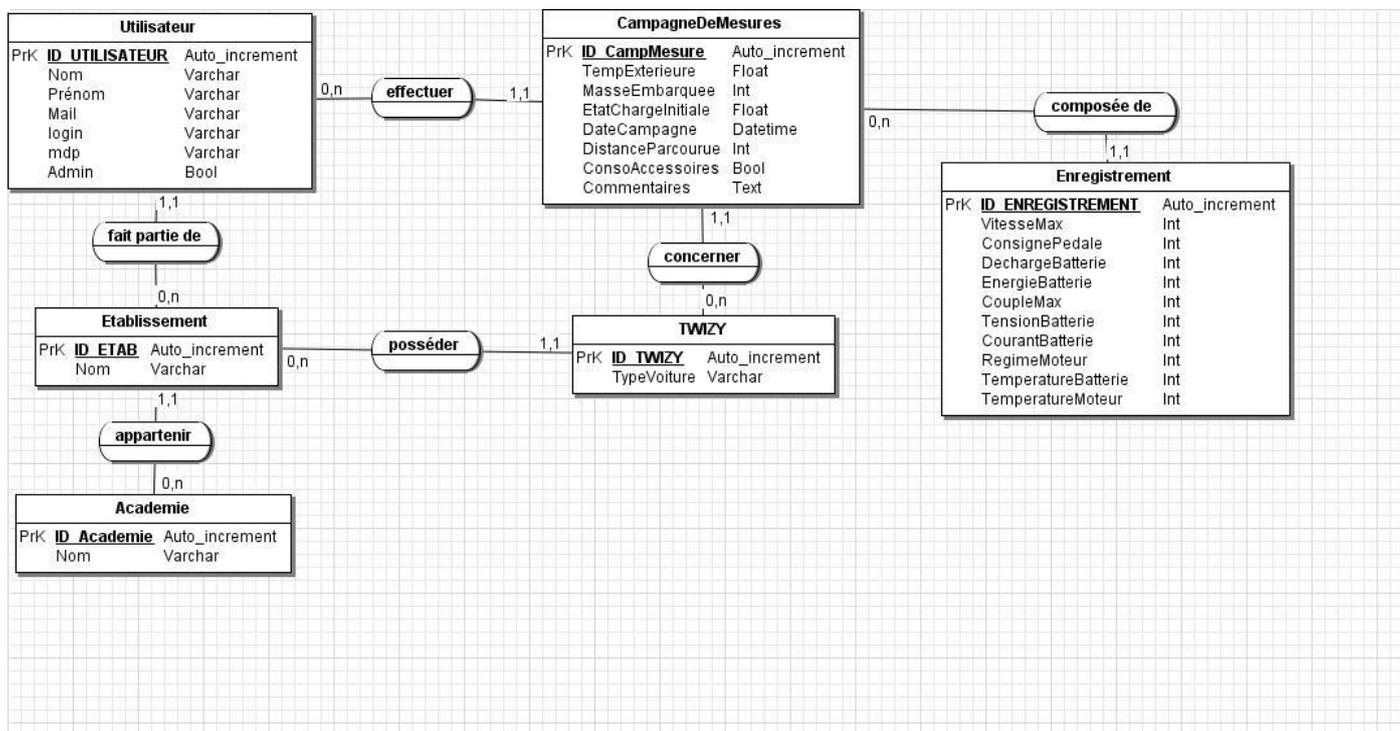
- **Gérer les établissements**, les Twizys et les comptes utilisateurs (ceux autorisés à uploader des fichiers). *Gérer* englobe *Créer, Modifier, Supprimer*. Cela fera appel à autant d'IHM nécessaires.
- **Gérer les données** (supprimer par exemple).
- **Obtenir des statistiques** sur des données comme le nombre de campagnes par établissement par exemple.

Application web : structure niveau 1

On pourrait donner une architecture de site sur la base d'un diagramme de classes, mais son implémentation peut ensuite sembler un peu difficile à expliquer. De plus il existe plusieurs solutions possibles. Dans notre cas, l'application web reste simple.

Application web : la base de données.

Une application web va avoir besoin d'une base de données. De manière assez classique, on va commencer par mettre au point le modèle conceptuel de données (MCD). C'est une sorte de diagramme de classes (schéma entités-relations) simplifié. Ci-dessous, celui qui a été mise au point pour l'application web liée à la TWIZY :



Ce diagramme a été saisi avec le logiciel gratuit JMerise. Le gros avantage dans notre cas est de pouvoir par la suite générer automatiquement le code SQL de création de la base (en plus de vérifier que le modèle est correct), une fois ce diagramme généré, aucune ligne de code n'est tapée. On copie-colle le code SQL dans un PHPMysqladmin (interface classique de gestion d'une base de données MySQL) et après validation, la base est créée et opérationnelle.

Justification des choix techniques :

- Acquisition des données : le système sur lequel nous travaillons ayant un bus CAN sur lequel circulaient les données, nous avons choisi le matériel CANUSB pour plusieurs raisons :
 - peu cher, environ une centaine d'euros.
 - permet de visualiser ce qu'il se passe sur le bus CAN avec le logiciel CAN monitor pro ainsi que de générer des trames à la demande.
 - possède une librairie utilisable avec Borland C++ Builder.
- Environnement de développement :
 - Les langages de programmation évolués comme Java ou C# sont des outils adaptés pour un tel développement. La difficulté ici est d'arriver à intégrer la DLL venant avec le CANUSB. Avec le langage Java, c'est plus technique, et en C#, ce n'est à priori pas trop difficile.
 - Nous avons donc cherché à utiliser Java dans un premier temps, mais nous avons rencontré des problèmes avec la DLL. Comme il y avait une librairie spéciale C++ Builder nous nous sommes tournés vers cet environnement. Comme avec Netbeans ou Visual Studio, cet IDE nous permet de créer des IHM en les dessinant, ce qui est beaucoup plus facile et rapide. C'est le seul que je connaisse en C++. Il existe bien sûr Qt mais là plutôt réservé aux connaisseurs. Nous sommes donc partis avec C++ Builder.
- hébergement web : ici pas de secret, nous avons pris ce que le rectorat utilise, c'est-à-dire le couple PHP/MySQL.

Conclusion générale :

L'aboutissement de ce type de projet dépend bien évidemment des compétences techniques de l'équipe. Ici c'est un projet qui est un peu électronique, mais surtout informatique industrielle, qui a demandé des compétences de développement logiciel à minima, des connaissances sur les bases de données et des connaissances sur les bus industriels.

Le 1^{er} juillet 2014

Baudouin MARTIN – Professeur au lycée Vaucanson à Tours

Émile FERRÈRE – Professeur au lycée Henri Brisson à Vierzon

Cyril MARINE – Chef de Travaux au lycée Henri Brisson

Frédéric TARAUD – Chef de Travaux au lycée Benjamin Franklin à Orléans