



HAL
open science

TéléTP : intégration de systèmes didactiques dans une plateforme d'apprentissage en ligne

Alexis Lombard

► **To cite this version:**

Alexis Lombard. TéléTP : intégration de systèmes didactiques dans une plateforme d'apprentissage en ligne. Environnements Informatiques pour l'Apprentissage Humain. 2013. dumas-01160127

HAL Id: dumas-01160127

<https://dumas.ccsd.cnrs.fr/dumas-01160127>

Submitted on 4 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL DE TOULOUSE

MEMOIRE

Présenté en vue d'obtenir

LE DIPLOME D'INGENIEUR CNAM

Spécialité : Informatique

Option : Architecture Ingénierie des Systèmes et des Logiciels

par

LOMBARD Alexis

**TéléTP : Intégration de systèmes didactiques dans une plateforme
d'apprentissage en ligne**

Soutenu le 25 Juin 2013

JURY

PRESIDENT : Mme Anne WEI

Professeur d'université Enseignant-chercheur

MEMBRES : M. Hadj BATATIA

Maître de Conférences

M. Pascal DAYRE

Maître de Conférences

M. Patrice MICHEL

M. Jean-Philippe SECHAO

M. Florian BINET

Remerciements

Je tiens à remercier :

- ✓ **Jean-Philippe SECHAO**, responsable de mon stage, de m'avoir proposé ce sujet de mémoire. Il m'a suivi et soutenu tout au long des 10 mois de mon stage. Je le remercie pour tous les conseils qu'il a pu me donner et qui m'ont permis d'avancer dans ce projet, pour toute l'aide qu'il m'a apportée et notamment pour la rédaction du présent mémoire et pour ses relectures méticuleuses.
- ✓ **Patrice MICHEL**, responsable de la filière Electrotechnique du Conservatoire National des Arts et Métiers de Toulouse, qui est à l'initiative de ce projet.
- ✓ **Cédric BARBE**, PDG de la société DMS pour m'avoir accueilli au sein de DMS.
- ✓ **Les collaborateurs de DMS** pour leur accueil chaleureux mais surtout Florian BINET pour toute sa disponibilité, qui m'a aidé dans mes recherches et a répondu à mes nombreuses questions. Il m'a permis de découvrir le monde des automates programmables industriels.
- ✓ **Pascal DAYRE**, tuteur CNAM de mon stage, pour toute sa disponibilité et ses conseils avisés qui m'ont permis d'évoluer et d'avancer tout au long de mon stage. Je le remercie également pour ses relectures et ses conseils concernant la rédaction du mémoire.
- ✓ **Hadj BATATIA**, responsable pédagogique de la filière Informatique du Conservatoire National des Arts et Métiers de Toulouse.
- ✓ **A mes parents et à ma sœur**, pour leur soutien, leur présence et leurs encouragements tout au long de mes études.
- ✓ **Brice** pour m'avoir accompagné tout au long du cursus CNAM.

Abréviations

AJAX	Asynchronous Javascript And XML
API	Automate Programmable Industriel
EIAH	Environnement Informatique pour l'Apprentissage Humain
FAD	Formation à Distance
FOAD	Formation Ouverte et à Distance
IHM	Interface Homme-Machine
ISO	International Organization for Standardization
MVC	Modèle Vue Contrôleur
NTIC	Nouvelles Technologies de l'Information et de la Communication
OSI	Open Systems Interconnection – Interconnexion de systèmes ouverts
SOAP	Simple Object Access Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TD	Travail Dirigé
TP	Travail Pratique
TIC	Technologies de l'Information et de la Communication
TICE:	Technologies de l'Information et de la Communication pour l'Enseignement
UML	Unified Modeling Language - Langage de Modélisation Unifié
XML	eXtensible Markup Language
WSDL	Web Service Description Language

Glossaire

API	Dispositif électrique programmable destiné à la commande de processus industriel. On peut voir un automate comme un type particulier d'ordinateur réactif ayant des entrées et des sorties physiques et utilisé pour automatiser des processus.
Bus de terrain	Ensemble de conducteurs permettant d'échanger des données dans un espace géographique limité (véhicule, usine, ...).
Cognitive	Science qui renvoie à l'étude interdisciplinaire de l'acquisition et de l'utilisation de la connaissance. Cette science a pour objet de décrire, d'expliquer et le cas échéant de simuler les principales dispositions et capacités de l'esprit humain - langage, raisonnement, perception, coordination motrice, planification...
Didactique	Science qui s'intéresse aux méthodes et aux contenus des enseignements en étudiant comment ils pourront être transmis et appropriés par les apprenants.
E-learning	Apprentissage par des moyens électroniques.
EIAH	Environnement Informatique pour l'Apprentissage Humain. Environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant.
Formation en présentiel	Formation classique en salle où les apprenants sont réunis dans un même lieu devant un même formateur.

Framework	Ensemble de bibliothèques et de conventions permettant le développement rapide d'applications. Il fournit suffisamment de briques logicielles et impose suffisamment de rigueur pour pouvoir produire une application aboutie et facile à maintenir.
Objet Pédagogique	Toute ressource (numérique ou non) utilisée pour effectuer une activité d'apprentissage ou de soutien.
OSI	Modèle théorique de communication entre deux dispositifs et composé de sept couches (Physique, Liaison, Réseau, Transport, Session, Présentation, Application) ou chaque couche exerce une ou plusieurs fonctions précises. Il est proposé par l'ISO.
Protocole de communication	Ensemble de règles d'échange entre deux stations ou plus à l'aide de messages ou « trames ». Ces dernières contiennent des champs de contrôle ou des données utiles.
Référentiel de Formation	Document décrivant les connaissances visées par la formation en définissant les objectifs pédagogiques, le programme ainsi que l'accompagnement prévu.
Service Web	Ensemble de fonctionnalités accessible à distance à l'aide du réseau par une autre application indépendamment des plateformes et des langages sur lesquels elles reposent.
TCP/IP	Protocole de transport fiable, c'est-à-dire qu'il permet l'établissement d'une session de communication entre deux parties qui veulent échanger des données.
Widget	Objet graphique pouvant réagir avec l'utilisateur.

Table des matières

Remerciements	2
Abréviations	3
Glossaire	4
Table des matières	6
Introduction	9
I. Etat de l'art	12
1. L'évolution de la formation à distance	12
1.1. De la formation à distance à la formation ouverte et à distance	12
1.2. L'arrivée de l'e-formation	13
1.2.1. Les téléTP	14
1.2.2. Enjeux	16
1.3. Les EIAH : Support Informatique dans la formation à distance	18
2. Le pilotage des systèmes didactiques	19
2.1. Le monde des Automates Programmable Industriels	20
2.1.1. Fonctionnement d'un API	20
2.1.2. Programmation de l'API	22
2.1.3. Les Variables	22
2.2. Les réseaux industriels locaux	23
2.2.1. ProfiBus	24
2.2.2. Modbus	25
2.3. L'IHM de commande	25
3. Les environnements ayant proposés des TéléTP sur des systèmes didactiques	27
3.1. TPLine	28
3.2. EVARISTE	29
3.3. ICTT@Lab	30
3.4. eINST	33
II. Cas d'étude	36
1. La société DMS	36
2. Le projet DIWEB	38
3. Le panneau photovoltaïque	39
3.1. Présentation et Fonctionnement	39
3.2. Composition	42
3.3. Les dispositifs environnementaux	42
3.4. Utilisation	43

III. Proposition d'intégration de systèmes didactiques dans une plateforme d'apprentissage

-----	45
1. Analyse	45
1.1. Organisation des TP en présentiel	45
1.2. Déroulement des TP en présentiel	46
2. Spécifications	47
2.1. Besoins	47
2.1.1. Contrainte temporelle	47
2.1.2. Observabilité du système lors de la manipulation	48
2.1.3. Sécurité des systèmes	48
2.1.4. Temps de manipulation d'un système	49
2.1.5. Créneaux Horaires	49
2.2. Fonctionnalités	49
2.2.1. L'Apprenant	50
2.2.2. L'Administrateur	51
2.2.3. Le formateur	51
2.3. Modélisation	52
2.3.1. Les cas d'utilisation	53
2.3.1.1. Cas d'utilisation « Réaliser l'expérimentation et la collecte de résultats d'un TP »	53
2.3.1.2. Cas d'utilisation « Interpréter les résultats d'une manipulation »	55
2.3.1.3. Cas d'utilisation « Administrer un laboratoire distant »	55
2.3.1.4. Cas d'utilisation « Administrer des systèmes didactiques »	56
2.3.1.5. Cas d'utilisation « Administrer l'utilisation des systèmes »	57
2.3.2. Diagramme de déploiement	58
2.3.3. Diagramme de classes	59
2.3.3.1. La couche Laboratoire	61
2.3.3.2. La couche Système	62
2.3.3.3. La couche IHM de commande	63
2.3.3.4. La couche Actions	63
2.3.3.5. La couche Proxy	64
2.3.4. Diagramme de composants	66
IV. La conception	68
1. Réseau Informatique d'un établissement	68
2. Choix techniques sur la communication avec les automates	71
2.1. Communication en TCP/IP entre le serveur web et l'automate	72
2.2. Communication en Modbus TCP/IP entre le serveur web et l'automate	74
2.2.1. La bibliothèque libModbus	74
2.2.2. La bibliothèque JModbus	75
2.2.3. La bibliothèque Modbus4j (Modbus For Java)	76
2.3. Le service WebDesigner	76
2.4. Choix du type de communication utilisé	77
V. Réalisation et Expérimentation	78
1. Installation et mise en place du réseau au sein de la société DMS	78
1.1. Accès à distance au système panneau photovoltaïque	80
1.2. Configuration du routeur	81

1.3.	Port forwarding et Iptables -----	81
2.	Développement de la plateforme DIWEB -----	82
2.1.	Choix du langage de développement -----	83
2.2.	Le Design Pattern Modèle Vue Contrôleur -----	84
3.	Développement de la solution proposée -----	86
3.1.	Organisation du développement -----	86
3.2.	Développement -----	90
3.2.1.	Développement des IHM de commande du panneau photovoltaïque -----	90
3.2.2.	Le pilotage du système panneau photovoltaïque -----	91
3.2.3.	Observation de la manipulation -----	95
3.2.4.	Affichage du flux vidéo et audio -----	100
4.	Expérimentation de la solution proposée -----	100
Conclusion -----		103
Annexe : La modélisation UML -----		105
Références Bibliographiques -----		108
Table des illustrations -----		110

Introduction

Depuis une quinzaine d'années, les Technologies de l'Information et de la Communication (TIC) et particulièrement Internet se sont développées à tel point qu'elles ont bouleversées notre quotidien que ce soit personnel ou professionnel. L'enseignement ne pouvait pas rester en marge de ces évolutions, les Technologies de l'Information et de la Communication pour l'Enseignement (TICE) se sont adaptées afin de familiariser les apprenants à l'utilisation des nouvelles technologies. De ce fait, de nombreux sites d'e-formation ont vu le jour.

A leur début, ces sites proposaient uniquement de l'enseignement théorique sous la forme de cours ou de travaux dirigés (TD) en ligne. Il n'y avait donc pas de réelle activité de la part de l'apprenant. Or si l'on veut faire de l'e-formation un outil de formation viable et largement utilisé, ces sites se devaient de proposer des travaux pratiques à distance ou « téléTP » pour que l'apprenant puisse manipuler des dispositifs technologiques (systèmes réels, systèmes didactiques, instruments de laboratoire, robots, ...). Des groupes de recherches se sont créés et ont permis de faire avancer le sujet des téléTP. Cependant celui-ci reste assez complexe et de nombreux problèmes techniques ont été rencontrés notamment sur la sécurité des dispositifs technologiques et des humains mais aussi sur la faisabilité de piloter ces dispositifs en temps réel. Il faut également arriver à donner la possibilité à l'apprenant de piloter le dispositif en le rapprochant au maximum des conditions, des sensations qu'il rencontre lorsqu'il pilote ce dispositif dans la salle de Travaux pratiques (TP). Ce dernier point reste le plus dur à résoudre.

Le domaine des téléTP est vaste et fait appel à de multiples disciplines comme la didactique, l'informatique, les interfaces Homme Machine (IHM), ou encore la pédagogie. Il est assez difficile de s'intéresser à toutes ces notions c'est pourquoi nous avons décidé de nous orienter sur comment rendre pilotable un système didactique à distance en se connectant sur une plateforme d'apprentissage en ligne et sur comment rapprocher cette manipulation à distance de celle qu'il serait possible de réaliser dans une salle de TP. Autrement dit, ce mémoire traitera de l'intégration des systèmes didactiques dans une plateforme d'apprentissage sous la forme d'un laboratoire distant. Un laboratoire distant est le

regroupement d'un ensemble de systèmes didactiques accessible à distance. Dans notre cas ce laboratoire sera uniquement composé de systèmes **automatisés** didactiques.

Intégrer un système automatisé didactique revient à pouvoir se connecter sur l'automate de ce système à distance et avoir la possibilité de communiquer avec lui en accédant à sa mémoire interne. L'objectif est de piloter le système à travers une IHM de commande accessible depuis la plateforme d'apprentissage.

Mais cela revient également à gérer les dispositifs environnementaux comme une caméra ou un micro pour percevoir la manipulation sur le système de façon plus réelle, sécuriser le système en ajoutant des droits d'accès mais aussi gérer la traçabilité des apprenants lors de chaque connexion sur le système pour récupérer toutes les actions faites et stocker l'état du système au fur et à mesure des actions réalisées pour permettre aux apprenants d'interpréter les résultats trouvés.

L'objectif reste de pouvoir automatiser au maximum l'intégration de ces systèmes avec notamment la génération automatique de l'IHM de commande qui pourrait être faite en fonction des différentes informations que pourrait remonter l'automate de chaque système lorsqu'il est connecté sur un réseau informatique. Mais c'est après une étude complète sur le fonctionnement des automates que nous avons pu dire qu'une automatisation complète de la génération d'une IHM de commande n'est pas possible.

C'est dans ce contexte que nous allons essayer de faciliter l'intégration des systèmes automatisés didactiques dans une plateforme d'apprentissage en ligne afin de réaliser des téléTP.

Pour plus de clarté et pour respecter le vocabulaire défini dans le domaine de l'enseignement, nous nommerons les élèves des apprenants et les enseignants, des formateurs.

Organisation du mémoire

La **première partie** permet de faire l'état des connaissances sur le sujet des téléTP et des systèmes didactiques autour de trois axes : l'historique de la formation à distance permettant de mieux comprendre les enjeux des téléTP, la notion de pilotage de système didactique et en dernier lieu un panorama des différents projets traitant des téléTP à travers une plateforme d'apprentissage.

La **deuxième partie** présente le contexte dans lequel nous avons pu travailler sur le sujet, la société partenaire DMS, le projet DIWEB ainsi que le système qui a été mis à notre disposition, le panneau photovoltaïque.

La **troisième partie** est consacrée à l'étude des besoins et des fonctionnalités liés à l'utilisation et à l'administration des systèmes didactiques au sein de la plateforme d'apprentissage pour chacun des utilisateurs. Ces utilisateurs sont l'apprenant, le formateur et l'administrateur. C'est à partir de cette étude que nous avons pu modéliser l'architecture de la solution proposée.

La **quatrième partie** présente la conception de la solution proposée. Nous décrivons les choix techniques opérés pour respecter les besoins utilisateurs et notamment ceux concernant la mise en place du réseau informatique dans lequel évolueront les systèmes didactiques et ceux permettant de communiquer en temps réel avec l'automate du système.

La **cinquième partie** décrit la réalisation et l'expérimentation de la solution. Nous expliquons l'implémentation de la solution en tenant compte des choix techniques que nous avons adoptés.

Pour conclure ce mémoire, nous faisons un bilan du travail effectué et des perspectives qui pourront être développées dans le futur.

I. Etat de l'art

Avec l'arrivée de l'informatique et plus récemment d'Internet, la formation à distance a pu se développer de façon considérable à tel point qu'il est aujourd'hui envisageable de faire des téléTP au travers de la manipulation de dispositifs technologiques à distance. Pour comprendre les enjeux des téléTP, nous retraçons dans la section suivante l'évolution de la formation à distance de ses débuts jusqu'à l'e-formation. La deuxième partie de ce chapitre décrit le fonctionnement des systèmes didactiques permettant de comprendre leur pilotage. La troisième partie de ce chapitre est consacrée à l'étude des différents projets ayant vu le jour avec l'objectif de faire des téléTP sur des systèmes didactiques ou réels.

1. L'évolution de la formation à distance

1.1. De la formation à distance à la formation ouverte et à distance

La **FAD** (Formation A Distance) n'est pas un nouveau concept. Il a été conçu en Angleterre depuis le milieu du 19ème siècle avec les cours privés par correspondance. En effet l'arrivée du timbre- poste a permis de s'assurer de l'arrivée du courrier à son destinataire. Depuis, de nombreux organismes se sont créés comme le CNED (Centre National d'Enseignement à Distance) en 1939 et propose des cours privés par correspondance. L'apprenant qui était inscrit dans une formation par correspondance recevait donc par courrier tous les supports nécessaires à la formation.

Ce n'est que plus tard que le qualificatif « ouvert » a été rajouté à la FAD et a donné la naissance à un nouveau concept, la **FOAD** (Formation Ouverte et A Distance). La notion d'ouverture de la formation à distance a créée des notions supplémentaires et vient dans le but de satisfaire les besoins de personnaliser les formations des apprenants. Il s'agit avec la FOAD d'adapter les cursus de formation aux différents besoins des apprenants. Il n'est plus question de leurs proposer des parcours standardisés mais d'adapter la formation à leurs

spécificités (parcours, besoins, expérience, contraintes, ...). Il caractérise des situations qui offrent une plus grande accessibilité, une plus grande souplesse dans l'organisation pédagogique ou par exemple aucune inscription n'est nécessaire pour accéder aux contenus pédagogiques, le rythme d'apprentissage est choisi par l'apprenant en fonction de ses capacités d'apprentissage. Tout est organisé en fonction des besoins de l'apprenant. **Ces situations ne sont pas forcément toutes en ligne, une partie de la formation peut se faire en présentiel.**

1.2. L'arrivée de l'e-formation

Avec l'arrivée du web et l'avènement des NTIC (Nouvelles Technologies de l'Information et de la Communication), de nouveaux concepts sont nés. On parle maintenant d'e-formation (ou e-learning en anglais). Les échanges entre les formateurs et les apprenants sont plus instantanés et donc beaucoup plus efficaces.

On peut utiliser la définition de l'Union Européenne pour définir **l'e-formation** par « l'utilisation des nouvelles technologies multimédias de l'internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, et d'autre part les échanges et la collaboration à distance ». L'e-formation implique donc l'utilisation de l'Internet dans l'apprentissage.

Il est donc difficile de nos jours de concevoir un dispositif FOAD sans e-formation. En revanche l'e-formation est forcément englobée dans la FOAD car l'e-formation ne pourrait pas être considérée comme une formation à part entière, il faut en effet une intervention humaine en amont et en aval pour définir des parcours pédagogiques et assurer un suivi d'apprentissage.

En fait l'e-formation comprend à la fois des situations où la technologie complète un apprentissage en présentiel, on est dans ce cas dans une situation de formation mixte (présentiel/distance), et des situations où l'ensemble du processus d'enseignement est vécu en ligne.

1.2.1. Les téléTP

Avec l'émergence d'une véritable pédagogie de « e-formation », les travaux pratiques à distance ou « téléTP » sont devenus incontournables et plus particulièrement pour les formations scientifiques et techniques. C'est dans cette optique que la faisabilité de réaliser des TP à distance est devenue primordiale.

Généralement dans ces formations, les TP en salle sont réalisés sur plusieurs types de système :

- ✓ Les **systèmes réels** : ce sont des équipements identiques à ceux que l'on peut retrouver dans une entreprise ou chez un particulier. Il peut s'agir par exemple d'une machine de production, d'un véhicule automobile, d'un équipement grand public.
- ✓ Les **systèmes didactiques** : l'exploitation de systèmes réels ne permet pas toujours l'accès aux organes qui assurent les différentes fonctions (dimensions, encombrement, ...). Il s'agit de systèmes isolés de leur contexte c'est-à-dire qu'ils mettent en œuvre un ou plusieurs organes réels de l'équipement industriel ou grand public. Ils sont généralement instrumentés ce qui permet de relever à certains points des données chiffrées sur le système en fonctionnement. Il peut s'agir de mesures comme la pression, le débit, la température.

Ils sont généralement fournis avec une série de documents : la notice du système didactique, un dossier technique contenant l'ensemble des spécificités techniques du système didactique et un ensemble de représentations électrique, mécanique et comportementale. Ceci permet à l'apprenant une meilleure connaissance du système didactique.

- ✓ Les **systèmes simulés** : il s'agit de systèmes simulés informatiquement permettant de visualiser le comportement du processus et un mode opératoire afin d'appréhender le fonctionnement du matériel ou du système.

On dénote quelques inconvénients sur les systèmes simulés comme le manque de sensations (visuels, auditif, ...). Tous ces manques peuvent empêcher l'apprenant d'avoir une totale compréhension de ce qu'il fait malgré des représentations en 2D et 3D des systèmes.

C'est pourquoi les systèmes réels et didactiques sont souvent privilégiés.

Voici quelques exemples de systèmes didactiques :

- ✓ La **liaison sol** arrière du modèle 1200GS de la moto BMW permettant d'expérimenter 2 types de suspensions et de transmission. La photo de droite ci-dessous représente le système didactique.

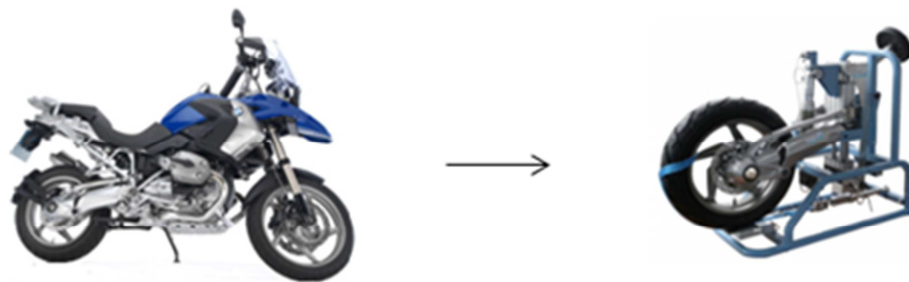


Figure 1: La liaison au sol

- ✓ La **boîte de vitesse C3** est la véritable boîte de vitesses du véhicule de la Citroën C3 qui intègre des capteurs sur les chaînes d'information et d'énergie. L'ensemble des mesures est conditionné et exploité dans un logiciel fourni.



Figure 2: La boîte de vitesse C3

- ✓ Le pont **Gustave Flaubert** est un pont levant routier issu du pont réel situé à Rouen. Ce système permet de mener des activités pratiques comportementales sur les structures du pont.



Figure 3: Le pont Gustave Flaubert

1.2.2. Enjeux

D'un point de vue **pédagogique**, l'e-formation présente un fort enjeu pour l'apprenant et les formateurs et ce sur plusieurs points :

- ✓ L'autonomie de l'apprenant : aptitude à organiser ses tâches de façon à les mener à bien.
- ✓ Permet un accès plus rapide aux connaissances sans limite de distance (élimination des contraintes de temps et de lieux).
- ✓ La familiarisation des apprenants avec le numérique. Le numérique qui est et sera présent dans la vie de tous les jours.
- ✓ L'apprenant doit se familiariser avec un nouveau processus d'apprentissage.
- ✓ L'accompagnement des apprenants est plus individualisé.
- ✓ Les nouveaux rôles des formateurs : plus de coordination pour intégrer et réguler les accompagnements des apprenants.

En se repositionnant dans le cadre des téléTP, nous pouvons relever d'autres enjeux **pédagogiques** mais également des enjeux **économiques** et **matériels**.

En effet les travaux pratiques en présentiel peuvent présenter quelques limites : [LELEVE et al, 2008]

- ✓ Au niveau du **temps** : les heures des travaux pratiques sont limitées dans l'emploi du temps et les rotations de TP peuvent ne pas être les meilleures pour certains groupes. En effet ils peuvent commencer par le TP alors que l'idéal est de commencer par le cours et les travaux dirigés. Aussi, un groupe peut ne pas arriver à finir tout le TP pendant la séance mais comme l'utilisation de la salle est limitée aux heures de TP, ils ne peuvent pas revenir le terminer à un autre moment ce qui peut être un frein à la compréhension de certaines notions.
- ✓ Au niveau de la **documentation** : le sujet du TP n'est donné que pendant les heures de TP.
- ✓ Au niveau des **systèmes didactiques** : ces systèmes sont lourds et coûteux ; il peut être difficile pour certains établissements de pouvoir en acheter. Des problèmes économiques et matériels sont rencontrés.

Les téléTP offrent aux apprenants et aux formateurs de nouvelles perspectives du point de vue **temporel** (les apprenants peuvent accéder aux systèmes didactiques à n'importe quel moment, accès aux cours si l'apprenant ne comprend pas certaines théories présent dans le TP) et **spatial** (possibilité de se connecter à distance sur les systèmes des autres établissements et donc possibilité de faire des TP sur des systèmes non présents dans l'établissement).

On retrouve donc des enjeux pédagogique, matériel et économique.

1.3. Les EIAH : Support Informatique dans la formation à distance

On ne peut pas parler de la formation à distance sans parler des EIAH (Environnement Informatique pour l'Apprentissage Humain). Ils sont indispensables pour faire de la formation à distance. Un EIAH est un environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissance chez un apprenant. En effet l'e-formation est entièrement médiatisée par les ordinateurs et les réseaux électroniques.

Au début de l'informatique, dans les années 1970 les EAO (Enseignement Assisté par Ordinateur) sont apparus mais ne permettaient que l'affichage du contenu pédagogique. Puis avec les progrès fait dans l'intelligence artificielle et des sciences cognitives, sont ensuite venus les EIAO (Environnement Intelligemment Assisté par Ordinateur) qui eux étaient capables de donner un ordre aux leçons destinées à l'apprenant, donc plus de dialogue d'interaction entre l'environnement et l'apprenant. Comme nous l'avons vu dans la section précédente avec l'avènement des TICE, ce sont développés les EIAH.

C'est un environnement qui intègre des agents humains (apprenant et formateur) et artificiels (informatique) et leur offre des conditions d'interactions locales ou à travers des réseaux informatiques ainsi que des conditions d'accès à des ressources formatives (humaines et ou médiatisées) locales ou distribuées [BENMOHAMED, 2007]. Un EIAH a pour objectif de favoriser des apprentissages, de les accompagner et de les valider.

Voici quelques exemples d'EIAH : [LIRIS, 2009]

- ✓ **Plateforme d'apprentissage/pédagogique** : environnement proposant des fonctionnalités à des ressources formatives (support de cours, exercices, pointeurs vers des ressources externes, ...) et à des fonctionnalités spécifiques (outils de communication, outils de suivi des apprenants, ...). On les retrouve généralement sous la forme de sites Internet.
- ✓ **Tuteur intelligent** : logiciel amenant un apprenant à réaliser une tâche et capable d'assurer un certain nombre d'actions, généralement inspirées de l'enseignement humain : explication de la démarche à utiliser pour la réalisation des tâches, interprétation des actions de l'apprenant afin d'élaborer une certaine

compréhension de son comportement permettant de générer une aide, une correction, une explication, ...

- ✓ **Environnement de simulation** : environnement permettant l'immersion de l'apprenant dans un monde virtuel conçu pour favoriser l'atteinte d'objectif pédagogique.
- ✓ **Environnement d'apprentissage mobile** : environnement s'articulant autour des technologies nomades (PDA, téléphone mobile, ...) à partir duquel on peut créer des tâches spécifiquement liées au besoin d'un apprenant.

2. Le pilotage des systèmes didactiques

Comme expliqué dans la partie 1.2.1 traitant des téléTP, il est possible de faire des TP sur plusieurs types de systèmes. Nous nous sommes uniquement intéressés aux systèmes **automatisés** didactiques dans le but de les intégrer dans une plateforme d'apprentissage en ligne et de les piloter à distance afin de faire des téléTP.

Un système automatisé est un système qui exécute toujours le même cycle de travail pour lequel il a été programmé. Ce genre de système est composé d'une **partie opérative**, d'une **partie commande** et de l'**IHM** permettant de commander le système.

- ✓ La **partie opérative** : elle est composée des actionneurs (agit sur la partie mécanique du système) et des capteurs permettant d'acquérir les divers états du système.
- ✓ La **partie commande** : c'est elle qui gère le fonctionnement du système automatisé. Généralement au travers d'un automate.
- ✓ L'**IHM** : il permet à l'opérateur (apprenant ou formateur) de commander le système en lui envoyant des ordres.

La partie centrale du système et celle qui nous intéresse le plus est l'API (Automate Programmable Industriel). En effet, c'est en se connectant et en interagissant sur les API que nous pourrions piloter à distance les systèmes au travers de l'IHM de commande.

2.1. Le monde des Automates Programmable Industriels

Le marché des API est dominé par les constructeurs Schneider-Electric, Siemens et Rockwell. Dans le domaine de l'éducation, Schneider-Electric reste le constructeur le plus impliqué. Il présente de nombreux avantages pour les enseignants : des prix attractifs et une certaine facilité de mise en place. La plupart des systèmes automatisés didactiques sont donc équipés d'automate de la marque Schneider-Electric.

2.1.1. Fonctionnement d'un API

Un **API** est un dispositif électrique destiné à la commande de processus industriel. Pour cela un programme est installé sur l'API et tourne de façon cyclique. En fonction des données récupérées sur les capteurs, sondes ou autres (on parle des entrées), le programme de l'automate réagit et met à jour les actionneurs nécessaires (on parle des sorties). La figure 4 illustre le fonctionnement d'un API.

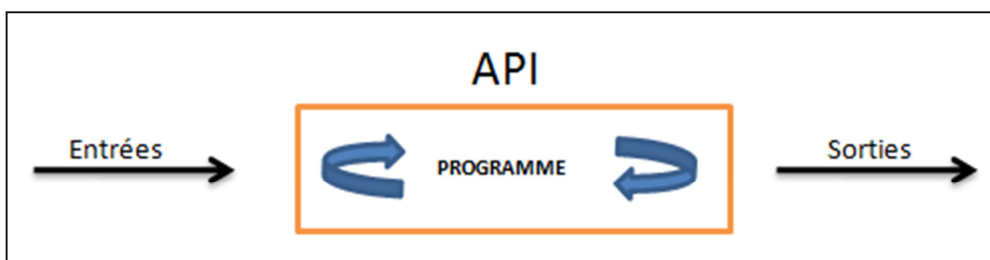


Figure 4: Fonctionnement d'un API

Un API contient au minimum une alimentation et un processeur (CPU). Il est organisé sur un système de rack dans lequel on peut rajouter un certain nombre de modules :

- ✓ Des **cartes entrées-sorties** (Input/Output, I/O en anglais), qui peuvent être numériques (tout ou rien, en quelque sorte un interrupteur, à 2 états qui peut avoir la valeur soit 1 soit 0) ou analogiques pour des signaux à évolution continue (une température, un courant compris entre 0 et 10 Volts par exemple).

- Des cartes d'entrée pour brancher des capteurs, des boutons poussoirs, ...
 - Des sorties pour brancher des actionneurs (ventilateur, éclairage, ...), des voyants ou autre.
- ✓ Des **modules de communication** pouvant utiliser les protocoles MODBUS, ProfiBus, InterBus, TCP/IP, CANOpen pour ne citer qu'eux et permettant de communiquer avec d'autres automates, serveur web ou avec des entrées/sorties déportées (un afficheur peut être considéré comme une sortie dérivée, une sonde comme une entrée dérivée).
 - ✓ Des **modules de dialogue homme-machine**, tel qu'un pupitre ou un IHM de commande.
 - ✓ Un **serveur web** qui a pour but de stocker des pages Web, que ce soit des pages HTML ou des applets Java. Ces pages Web sont là pour afficher les caractéristiques techniques du serveur ou des outils permettant de dialoguer avec le serveur. Il ne possède pas d'interpréteur ou de compilateur. Il a généralement très peu de mémoire (de l'ordre de 8 Mo).

La CPU contient deux types de mémoire:

- ✓ ROM : contenant les variables systèmes de l'automate, elle n'est accessible qu'en lecture.
- ✓ RAM : elle est elle-même divisée en 2 zones mémoires :
 - La zone **mémoire programme** contenant le programme. Ce sont les instructions du cycle dans l'API.
 - La zone **mémoire des données** contenant les variables entrées et sorties, les variables internes et les constantes. Leur valeur évolue en fonction du déroulement du cycle. Cette zone est découpée en deux ensembles distincts : la mémoire **bit** que l'on appelle également **coil** et la mémoire **registre**. L'automaticien peut allouer 1 bit ou 1 registre (équivalent à 16 bits) pour une variable en fonction des besoins.

2.1.2. Programmation de l'API

L'écriture d'un programme correspond à l'établissement du cycle d'un système automatique. Pour représenter le fonctionnement du système automatisé, on utilise le langage graphique **GRAF CET** (Graphe de Commande Etape Transition). Il permet de représenter chaque étape du fonctionnement du système sur lesquelles on associe une ou plusieurs actions à réaliser.

Après avoir fait le GRAFCET du système automatisé, on s'appuie sur un langage de programmation pour coder les actions associées aux étapes. On utilise généralement le **Ladder**, langage graphique ressemblant aux schémas électriques ou le **List**, langage textuel assez proche de l'assembleur. Le ladder pour des raisons de facilité est le plus souvent choisi par les automaticiens.

2.1.3. Les Variables

Ce sont les variables créées pour le programme de l'API par l'automaticien qui vont nous permettre d'**interagir** avec l'API.

Les différents types d'une variable dans un programme automate peuvent être le **booléen** (BOOL - 1 bit), l'**entier** (INT - 16 bits), le **mot** (WORD - 16 bits), le **double mot** (DWORD - 32 bits), la **chaîne de caractères** (STRING), le **nombre à virgule flottante** (REAL sur 32 bits), le **TIME** (correspond au nombre de secondes, codé sur 32 bits) ou les **tableaux** (ARRAY).

La mémoire des données de l'automate est découpée en deux ensembles distincts, les bits et les registres. Les variables de type booléen seront stockées dans la zone bit tandis que les autres types de variables (entier, réel, time, string, mot, double mot et tableau) seront stockées dans la zone registre sur 1 ou plusieurs registres ou chaque registre équivaut à 16 bits. Pour exemple, le réel étant codé sur 32 bits, il sera stocké sur 2 registres.

Il existe plusieurs types de variables, les variables internes au programme, les constantes, les variables systèmes qui nous donnent des informations sur le système, mais également les variables pour les entrées et sorties physiques de l'automate.

Quel que soit le type de variable, le format des adresses des variables sera toujours le même : %Vtn ou **V** représente le type de variables (M pour variable interne, K pour constante, S pour variable système, I pour entrée physique de l'automate, Q pour sortie physique de l'automate), **t** représente la typologie d'une variable (X pour un bit, W pour un mot, D pour un mot double et b pour octet) et **n** le numéro de la variable dans la mémoire des données.

2.2. Les réseaux industriels locaux

Les **réseaux industriels locaux** ou **bus de terrain** permettent de faire dialoguer des équipements industriels nombreux et divers (capteurs, actionneurs, API, informatique) en intégrant une caractéristique importante, celle de fournir des services contraints par le temps avec souvent du temps réel. C'est avec ces bus de terrain que la communication avec les API des systèmes est possible.

Ils se déclinent sous quatre niveaux. Le plus bas niveau (niveau 0) correspond à la communication entre les actionneurs, capteurs et API au sein d'un système, le niveau 1 représente la communication entre les API et les modules d'interface pour la commande de mouvement tels que les variateurs de vitesse ou démarreur, le niveau 2 la communication entre les API, et le niveau le plus haut (niveau 3) représente généralement la communication entre les API et un poste informatique. La figure 5 ci-dessous schématise ces quatre niveaux.

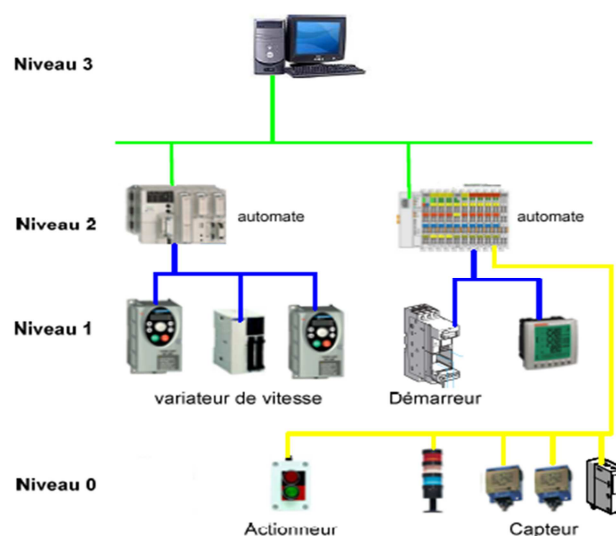


Figure 5: Les 4 niveaux des bus de terrain

On peut constater que notre problématique se situe au dernier et troisième niveau. Cette communication est possible grâce à l'existence de protocoles comme **Modbus** ou **Profibus**. Ces deux derniers étant considérés comme les principaux de leur catégorie.

Tout ceci a pu prendre forme par l'adoption progressive de standards communs entre les mondes de l'informatique et de l'automatisme ce qui a fait tomber la frontière entre ces deux mondes. La communication entre ceux-ci converge grâce à l'adoption de protocoles standards mondiaux comme Ethernet et TCP/IP (Transmission Control Protocol/Internet Protocol). Ethernet est un protocole connu, largement implémenté et ses performances augmentent continuellement avec l'évolution des technologies (notamment sa bande passante).

2.2.1. Profibus

Profibus (Process Field Bus) est le nom du protocole de communication inventé par Siemens. Il s'agit d'un protocole propriétaire, il est donc intégré quasi exclusivement dans les automates Siemens.

A ce jour, on trouve deux variantes du protocole, où la différence se trouve essentiellement dans le mode de transmission. En effet, les deux variantes échangent le même type de trames.

- ✓ **Profibus-DP** (Decentralized Peripherals / Périphériques décentralisés) : à ce jour, c'est la variante la plus utilisée. Il est basé sur la transmission RS-485 (possibilité de débit de l'ordre de 12 Mbits/seconde)
- ✓ **Profibus-PA** (Process Automation / Automatisation de Processus) : conçu pour les zones à risques (d'explosion notamment), il utilise la transmission MBP (limité à un débit de 31 Kbits/seconde)

Avec l'évolution de l'automatisme, une nouvelle spécification du protocole s'est mise en place : **PROFINET** où les échanges sont assurés par le média de communication Ethernet.

2.2.2. Modbus

Modbus est un protocole de communication non propriétaire proposé par la société Modicon et crée en 1979. Il relève du niveau 7 (applicatif) du modèle OSI. Ce protocole a été porté sur Ethernet pour constituer Modbus/TCP protocole totalement ouvert sur Ethernet. Le développement d'une connexion Modbus TCP/IP ne nécessite aucun composant propriétaire, ni achat de licence. Grâce au débit important d'Ethernet (100 Mbit/seconde), Modbus TCP/IP offre des performances totalement adaptées au besoin temps réel mais ne le garantit pas forcément en fonction de demandes plus importantes ou à des latences ponctuelles.

Il existe donc deux types de protocole Modbus :

- ✓ le protocole **Modbus RTU** de liaison série (la plupart du temps, un câble de type RS-485).
- ✓ Le protocole **Modbus TCP/IP**, qui est identique au Modbus RTU à la différence qu'il transmet les informations par paquets de données TCP/IP. On a donc une trame RTU qui est encapsulée dans une trame TCP/IP.

La structure d'une trame Modbus RTU (Remote Terminal Unit) est la suivante : l'adresse de l'esclave sur 1 octet, les codes fonctions sur 1 octet, les données, sur 252 octets et les codes de contrôle CRC sur 2 octets.

Modbus/TCP encapsule une trame Modbus dans une trame TCP. Il utilise le plus souvent un schéma de communication maître/esclave ou client/serveur.

2.3. L'IHM de commande

Sur les systèmes automatisés didactiques, une **IHM de commande** permet à l'utilisateur de contrôler le système. Elle est généralement développée à l'aide d'un logiciel propriétaire. Par exemple, Schneider-Electric propose l'application Vijeo Designer pour les développer. Ce logiciel permet de créer des pages au format WYSIWYG (What You See Is What You Get). Il s'agit d'une interface intuitive où l'utilisateur voit directement le résultat de ce qu'il crée à

l'écran.

A l'aide de cette IHM, l'utilisateur va générer des actions qui vont **modifier** la valeur des variables du programme de l'automate gérant les actionneurs. Ces changements de valeur vont permettre la modification du comportement des actionneurs et donc du système.

On peut également **lire** la valeur de ces variables.

L'image ci-dessous illustre le fonctionnement de la lecture et écriture des variables d'un API d'un système de panneau photovoltaïque.

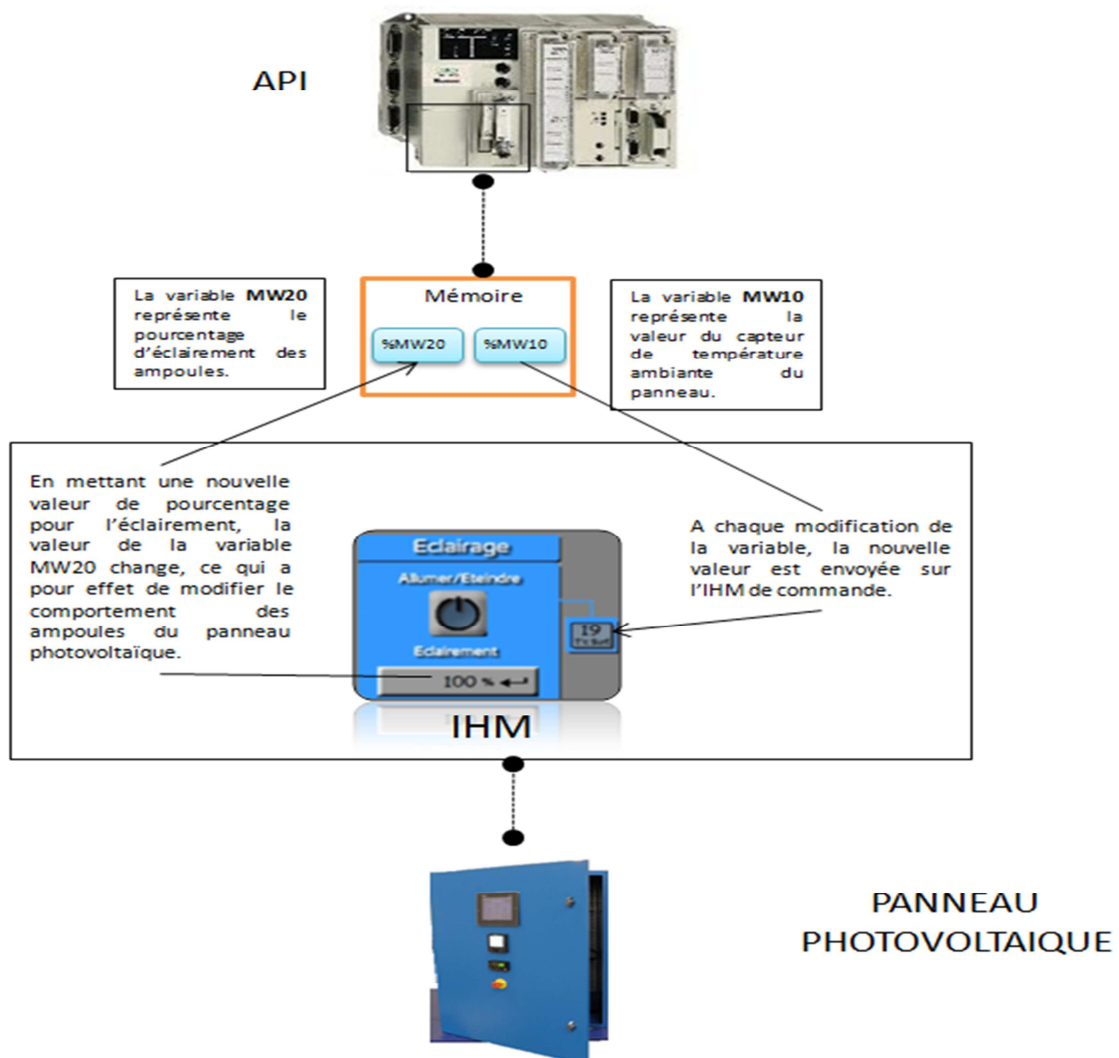


Figure 6: Lecture / Ecriture des variables d'un automate

Il s'agit de ces IHM de commande qui devront être reproduites au format Web afin de les intégrer dans la plateforme d'apprentissage. Les systèmes seront contrôlés à travers la plateforme dans le même principe que si on se trouvait en local.

3. Les environnements ayant proposés des TéléTP sur des systèmes didactiques

Au début du 21eme siècle les téléTP ont commencé à émerger dans les travaux de recherche pour combler les manques de la formation à distance dans ce domaine-là. Une diversité de projets ou d'études de faisabilité traitant des téléTP dans divers domaines ont vu le jour résultant pour la plupart en environnement propriétaires et utilisables uniquement dans le laboratoire ou l'université à l'origine du projet. C'est pourquoi malgré de nombreux sites traitant des téléTP que nous avons vu sur Internet, peu sont visibles et donc encore moins en démonstration.

Nous avons fait l'étude de quatre projets nous permettant de retracer le panorama des solutions existantes en terme de téléTP. Nous avons écarté toutes les applications permettant la télémanipulation via la simulation de systèmes. Nous avons pris en compte uniquement les environnements proposant des téléTP sur des systèmes réels ou didactiques et ayant une intention pédagogique. Tous ces projets ont été réalisés par des laboratoires de recherche ou bien par des enseignants.

Cette étude est nécessaire afin de comprendre ce qui a déjà été réalisé dans ce domaine -là et d'analyser comment l'utilisation des systèmes a été pensée d'un point de vue technique dans le but de faire des téléTP.

3.1. TPLine

TPLine est une plateforme gratuite mettant à la disposition des apprenants et des formateurs des travaux pratiques à partir de mesure faite sur des systèmes industriels réels distants.

Les TP proposés sont tous niveaux en partant du collège jusqu'au Post Bac en passant par le lycée et lycée professionnel.

Cette plateforme est le fruit de l'enseignement supérieur et de la recherche. Il a fallu également trouver des sociétés partenaires permettant d'instrumentaliser les systèmes réels tels que la SNCF ou Schneider Electric. Actuellement des TP sont proposés sur trois systèmes réels :

- ✓ Plusieurs **stations météo** aux abords de la ligne de TGV Paris-Marseille. Ces stations météo sont situées dans la vallée du Rhône là où le changement de vent est le plus sensible. On y mesure donc la vitesse et direction du vent.
- ✓ Un **tapis roulant**, qui est géré par un automate programmable. Les principales mesures effectuées sont la vitesse du tapis roulant, ainsi que la tension et le courant électrique générés en fonction de la vitesse.
- ✓ Le contrôle de la consommation électrique dans un **logement** implanté dans le Sud de l'Île de France. On a la mesure de l'énergie consommée chaque jour sur le site entier ainsi que quelques appareils comme le réfrigérateur, la cafetière ou le cumulus présents dans le logement.

Le fonctionnement de TPLine est très pratique pour les apprenants. En prenant exemple sur les stations météo, un apprenant voulant réaliser un TP sur ce système peut choisir les tranches horaires des mesures sur lesquelles il veut travailler et faire son TP. Il peut choisir la tranche horaire correspondant à des vents forts ou à des vents plus faibles pour avoir une comparaison. Si l'apprenant quitte sa session alors qu'il n'a pas fini son travail, ses configurations et ses réponses sont sauvegardées. Il pourra le terminer à n'importe quel moment. Une fois le TP terminé, l'apprenant le valide, ses réponses sont envoyées au

formateur pour correction.

Des cours sont également associés à chaque TP pour que les apprenants n'ayant pas compris toutes les notions liées au TP puissent revoir et comprendre ces notions.

Les mesures sont réalisées quotidiennement à des intervalles de temps régulier et envoyées en temps réel sur le serveur de TPLine via Internet. Sur le système des stations météo, les informations de vent sont mesurées par les stations anémométriques toutes les 5 secondes et moyennées toutes les 10 minutes. Les moyennes sont stockées dans la mémoire de la station, sont lues par GSM (onde radio) toutes les heures et envoyées au serveur TPLine où elles sont stockées en mémoire.

L'ensemble des mesures faites sur les systèmes réels sont stockées sur le serveur TPLine.

Cette plateforme ne permet pas réellement le « pilotage » des systèmes réels, mais simplement l'utilisation des mesures faites sur ces systèmes. On retrouve en revanche la notion de session de manipulation avec la possibilité de sauvegarde des configurations et réponses de l'apprenant et la possibilité pour l'apprenant de charger ses sessions de manipulation précédentes. Tout ceci concerne la traçabilité des apprenants.

3.2. EVARISTE

Dans le même style l'association **EVARISTE** (Etudes et Valorisations des Applications de la Recherche en Informatique sur les Systèmes Tutoriels d'Enseignement) a créé un ensemble de logiciels permettant « de promouvoir l'enseignement des sciences de l'école à l'université au travers d'activités expérimentales d'acquisition de données et d'exploitation pédagogique en temps réel et en interactivité avec l'expérience ». Ces logiciels proposent des TP sur différents domaines : physique, chimie, Science et Vie de la Terre, ...

Dans ces logiciels les TP sont réalisés sur des données déjà acquises, les apprenants interprètent les données, mais ne pilotent pas les systèmes.

3.3. ICTT@Lab

ICTT@Lab (generiC framework for remoTe and virTu@l LABoratory integration) est un framework d'intégration des téléTP au sein d'une plateforme de formation en ligne existante mais assure également toute la chaîne d'édition de scénarios pédagogiques de type téléTP. [BENMOHAMED, 2007] Il fait en quelque sorte l'interface entre le laboratoire distant d'un établissement (composé de systèmes automatisés didactiques) et la plateforme d'apprentissage existante. En effet, toutes les fonctions classiques de gestion et de diffusion de ressources d'apprentissage (cours, TD, projet, ...), de gestion des formateurs et apprenants, de gestion et de planification de l'apprentissage sont assurés au niveau de l'EIAH d'origine. De ce projet, nous avons pris en compte uniquement l'intégration des systèmes didactiques dans l'EIAH existant, toute la partie chaîne d'édition des scénarios pédagogiques a été occultée.

C'est un projet résultant des travaux de thèse de Hcene Benmohamed rendu en 2007 et développé dans le laboratoire ICTT de l'INSA de Lyon.

Pour qu'il puisse supporter les téléTP, une partie **interface** a été rajoutée à l'environnement d'origine pour faire le lien entre l'EIAH existant et le laboratoire distant comme on peut le voir sur la figure 7. Cette partie est composée de deux composants **ELaMS** et **OntoServ**.

- ✓ **ELaMS** (Electronic Laboratory Management System) représente le middleware, c'est-à-dire la partie assurant le dialogue entre la couche téléformation et la couche téléopération. Il permet d'installer et de référencer les nouveaux dispositifs technologiques disponibles dans le laboratoire distant mais également de gérer l'utilisation de ces derniers par les apprenants et les formateurs (gestion des droits d'accès aux dispositifs, choix et utilisation du dispositif, ...).
- ✓ **OntoServ** (Serveur d'Ontologies) est le conteneur des fichiers d'ontologies créés. Une ontologie est une formalisation logique, structurée et exhaustive des connaissances pour un domaine donné. Ici, les ontologies sont utilisées pour décrire et représenter de façon graphique les dispositifs technologiques à disposition dans le laboratoire distant. Pour chaque dispositif, on retrouve une **ontologie mère** décrivant le

dispositif de façon globale avec ses principaux composants. Les **ontologies filles** décrivent quant à elles chaque composant et ses principales fonctionnalités. Hcene Benmohamed s'est inspiré des techniques du Web sémantique en adoptant le langage d'ontologie web OWL (Web Ontology Language). L'objectif de cette modélisation est avant tout l'intégration des dispositifs technologiques présents dans le laboratoire dans les scénarios pédagogiques.

Tout ceci donne une architecture du framework ICTT@Lab qui est structurée en 3 niveaux bien distinct :

- ✓ Le niveau **téléformation** qui est ni plus ni moins l'EIAH existant permettant l'intégration de scénarios pédagogiques. Généralement les EIAH sont des plateformes d'apprentissage.
- ✓ Le niveau **téléopération** représentant l'ensemble des systèmes didactiques mis à disposition dans le laboratoire distant.
- ✓ L'**interface** faisant le lien entre les deux couches. Elle est composée d'ELaMS et d'OntoServ.

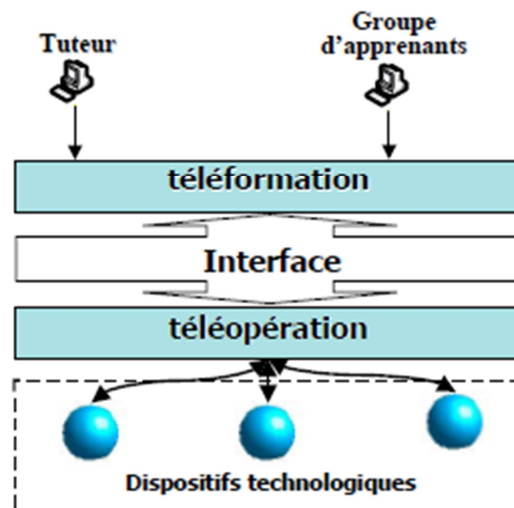


Figure 7: Architecture du projet ICTT@Lab

Ces travaux de thèse ont expérimentés des téléTP seulement sur deux systèmes automatisés didactiques (le magasin vertical et le four industriel).

- ✓ Le **magasin vertical** stocke des pièces qui sont introduites et retirées manuellement. L'objectif est de faire monter ou descendre les nacelles où sont positionnées les pièces pour mettre ou prendre la pièce voulue.
- ✓ Le **four industriel** permet de travailler sur la régulation de sa température.

La communication avec les API de ces deux systèmes didactiques a été réalisée par l'ajout d'un module serveur web permettant la communication TCP/IP.

Pour que les apprenants puissent les piloter à distance, des IHM de commande ont été construites à l'aide du langage de description d'univers virtuel 3D, **VRML** (Virtual Reality Modeling Language). C'est avec ce langage de présentation que les IHM en 3 dimensions ont été créées ce qui leur permet d'avoir un meilleur rendu à l'affichage sur le poste des apprenants.

Une fois les IHM de commandes réalisées, il est possible de contrôler les actionneurs du système et d'avoir accès à l'état du système en faisant de la lecture et de l'écriture des variables dans la mémoire de l'API. Malheureusement, dans ces travaux de thèse Hcene Benmohamed ne présente pas les protocoles de communication utilisés pour communiquer avec l'API.

L'intégration d'un nouveau dispositif dans ELaMS n'est pas automatique, dans le sens où quand on connecte le dispositif au réseau, tout ne se fait pas tout seul. L'administrateur doit rentrer des informations au travers de formulaires et créer les IHM de commande pour que les apprenants puissent piloter le système didactique. Pour déclarer un nouveau dispositif auprès d'ELaMS, l'administrateur via un formulaire rentre les informations nécessaires telles que le nom du dispositif, associe les URL permettant de trouver l'IHM de commande réalisés avec le langage VRML, les dispositifs environnementaux (caméra, micro) et de l'automate permettant de communiquer avec lui.

Concernant les temps de réponse entre l'application et les dispositifs, on est sur du temps réel, bien entendu les temps de réponse augmenteront en fonction du type de liaison

Internet des apprenants et de la distance qui sépare les apprenants des dispositifs technologiques.

Ce framework a été intégré sur la plateforme d'apprentissage Moodle (qui est libre).

3.4. eINST

eINST signifie E-INSTRumentation. L'eInstrumentation permet de désigner la manipulation distante de ressources, qu'elles soient industrielles, ou d'un autre domaine et assisté par ordinateur. [GRAVIER, 2007] Il s'adresse aux étudiants universitaires de 1^{er} cycle pour les cursus en science et ingénierie.

Ce projet est issu des travaux de thèse de Christophe Gravier fait en 2007 dont l'objectif est de permettre la généralisation des manipulations sur des dispositifs technologiques distants dans le but de faire des téléTP. Il traite de la problématique de la mise en ligne de laboratoires distants. Les laboratoires distants sont composés de dispositifs technologiques accessibles à distance.

L'objectif principal est de créer un prototype permettant de déployer de nouveaux dispositifs avec le minimum de développement requis.

L'architecture générale du projet eINST est caractérisée par trois parties :

- ✓ La partie entre le poste local (présent en local dans l'établissement) et le dispositif technologique lui-même. C'est grâce à ce poste que le dispositif sera accessible à distance.
- ✓ La partie **IHM** (Interface Homme-Machine) permettant de relier l'apprenant à l'application distante.
- ✓ La **partie centrale**, le **middleware** ou **intergiciel** permettant de transmettre via Internet, les commandes ou requêtes entrées par les apprenants de l'application jusqu'au dispositif et inversement, transférer les résultats.

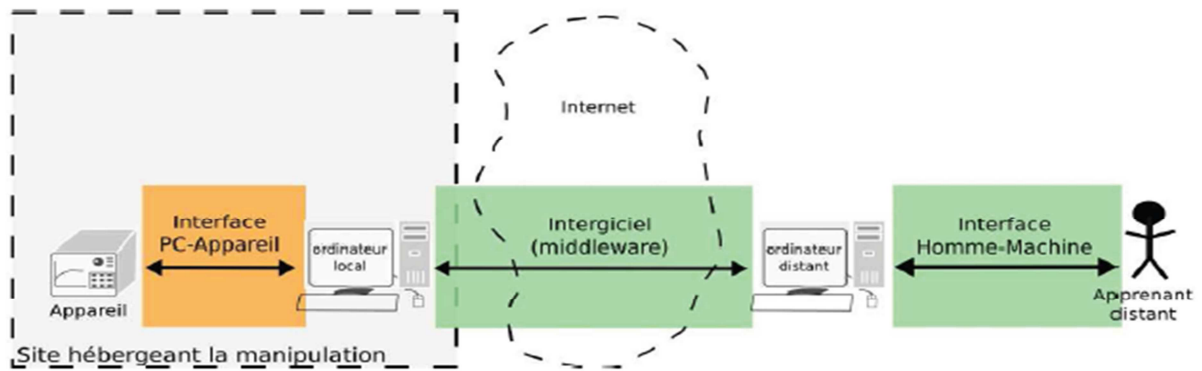


Figure 8: Architecture du projet eINST

Dans ces travaux, Christophe Gravier traite des dispositifs pour l'électronique et la micro-électronique comme les instruments de mesure parmi lesquels on peut trouver un oscilloscope, un générateur de signaux par exemple. Ce ne sont pas les dispositifs électroniques que nous souhaitons rendre accessibles à distance. Nous ne pourrions donc pas nous intéresser aux protocoles de communication utilisés entre l'IHM de commande et les instruments. En effet, les technologies **VISA** (Virtual Instrument Software Architecture) et **IVI** (Interchangeable Virtual Instruments) sont utilisées pour ces dispositifs. Elles permettent de rendre les drivers (pilotes) d'accès aux instruments les plus indépendants possibles de l'instrument piloté, en regroupant les dispositifs par catégories en s'affranchissant notamment du type de connexion utilisée pour contrôler les équipements (si on utilise de l'USB, de l'Ethernet, ou autre comme les liens série RS-232).

C'est donc sur la génération des IHM de commande de l'équipement que nous nous sommes intéressés. Dans ces travaux, ils devaient reproduire les différentes façades des instruments. Comme dans le projet ICTT@Lab, il a été choisi les ontologies OWL pour structurer et représenter les IHM. Ils existent des logiciels spécifiques pour développer ce genre d'ontologies.

Pour représenter l'IHM d'un dispositif avec un maximum de détails, OWL va jouer le rôle permettant d'avoir le maximum de connaissances sur l'instrument représenté par l'IHM. Grâce à cette ontologie, chaque composant ou widget affiché sur la façade de l'instrument pourra être représenté et relié à une propriété (localisation sur la façade, couleur, taille, ...)

ou à une action à l'aide d'une relation.

Le client en charge de l'affichage de la façade de l'instrument parcourt l'ontologie pour présenter cette façade. En fait, un client générique instancie la façade à partir de la lecture de l'ontologie correspondant à cette façade.

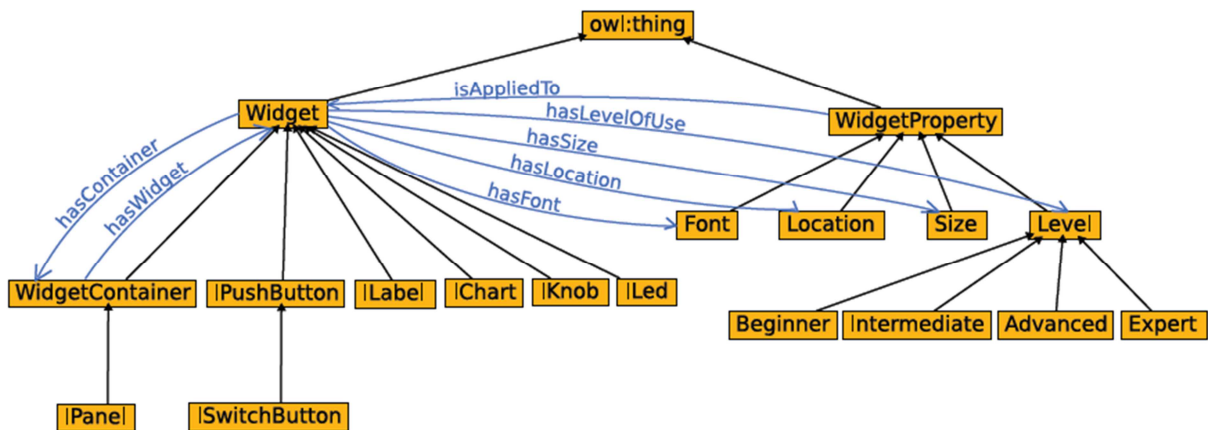


Figure 9: Exemple d'ontologie

Les flèches bleues représentent les relations entre les widgets et les propriétés de ces derniers. On retrouve la relation ternaire Widget, Propriété, Relation.

Il existe deux possibilités pour écrire des ontologies OWL. Soit en les écrivant à l'aide d'un éditeur de texte, soit en utilisant des éditeurs de fichiers OWL. La première solution n'est pas idéale car les fichiers OWL que l'on doit produire peuvent rapidement devenir assez fastidieux et long à écrire. Heureusement il existe sur le marché des éditeurs de fichiers OWL dont le principal est **Protégé** créée par l'université médicale de Stanford ou des environnements de développement comme le framework **Jena** (écrit en Java). Ces deux types d'outil sont gratuits et fournissent des environnements facilitant le développement d'OWL.

Néanmoins, même s'il est plus aisé d'écrire des ontologies avec ces outils, leur utilisation reste assez difficile. Dans notre cas, du fait de leur complexité, elle ne constitue pas pour l'instant une solution viable pour le développement de nos IHM de commande.

II. Cas d'étude

Ce mémoire a été réalisé au sein de la société DMS que nous présenterons dans la section suivante. Pour DMS, « la prise en compte des Technologies de l'Information et de la Communication dans l'éducation suppose une évolution profonde du système éducatif ». En effet le développement du numérique doit conduire à terme à l'évolution de l'organisation des enseignements et des pratiques pédagogiques.

C'est dans le cadre du développement de ses activités sur l'enseignement que DMS a souhaité réaliser une plateforme d'apprentissage en ligne intitulée **DIWEB** (Didactique WEB) permettant la mise à disposition des apprenants d'un ensemble de ressources pédagogiques pouvant être **matériels** (systèmes didactiques) ou **immatériels** (cours, Travaux Dirigés, Travaux Pratiques, ...) sur un ensemble de matières pour les formations professionnelles et technologiques.

1. La société DMS

La société **DMS** (DidaLab Matelco STCI), créée en 2003 a pour activité la conception, réalisation et commercialisation de systèmes didactiques ainsi que des cours et travaux pratiques destinés aux établissements présentant des formations technologiques et professionnelles dans les domaines :

- ✓ Des sciences de l'ingénieur, génie mécanique, électrotechnique, électronique, maintenance.
- ✓ De la maintenance automobile (outils de diagnostic, bancs moteurs, équipements de garage, ...).
- ✓ De la construction mécanique.

Par la suite, DMS a fait l'acquisition de la société GTI fin 2005 ayant pour principale activité la

maintenance industrielle et les sciences de l'ingénieur, ainsi que de la société MEDIA en 2006 s'occupant de la technologie au collège. Ces deux acquisitions ont pu élargir le domaine d'activité de DMS.

Ci-dessous les détails de l'historique de la société DMS sont présentés en détail.

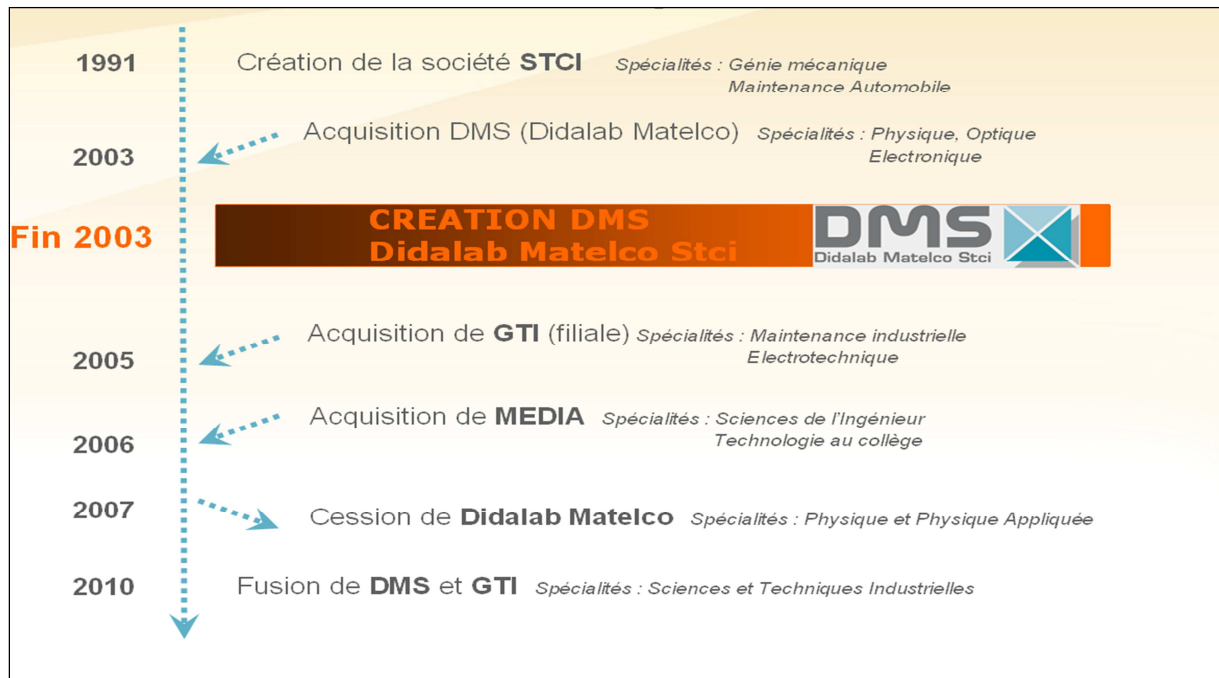


Figure 10: Historique de la société DMS

Tous les systèmes didactiques réalisés par DMS sont issus de systèmes réels mais sont adaptés à l'enseignement. Ils sont ensuite didactisés, c'est-à-dire qu'ils sont instrumentés et documentés pour être rendu appropriés à l'enseignement.

En effet l'ensemble de ces systèmes didactiques proposés par DMS sont accompagnés d'un dossier technique, d'un dossier pédagogique et d'un dossier de ressources. Ainsi les différents niveaux de formation utilisant les produits DMS sont principalement :

- ✓ **Niveau supérieur** : CPGE (Classes Préparatoires aux Grandes Ecoles), BTS (Brevet de Technicien Supérieur), DUT (Diplôme Universitaire de Technologie).
- ✓ **Niveau secondaire** : Bac S, SI, Bac Général, Bac Technologique, Bac Pro, Collège, BEP

et CAP (Certificat d'Aptitude Professionnelle).

- ✓ Les **centres de formation professionnels** : CFA, CFPA, ...

La société DMS a son siège à l'Aéroparc Saint Martin du Touch proche de Toulouse. En dehors de son activité en France, DMS est présent également dans les pays du Maghreb comme la Tunisie, le Maroc, l'Algérie ou la Lybie mais également au Liban.

2. Le projet DIWEB

DIWEB est une **plateforme d'apprentissage en ligne** qui aura pour objectif de faire acquérir des compétences à un apprenant pour qu'il puisse arriver au niveau requis de sa formation défini par les programmes et référentiels actuels. Un référentiel permet de définir les compétences (savoirs et savoirs faire) à maîtriser à la suite d'une formation. Pour cela la plateforme lui mettra à disposition un ensemble d'objets pédagogiques qui peuvent être matériel (systèmes) ou immatériels (cours, Travaux dirigés, Travaux Pratiques, ...).

Cette plateforme sera également accessible aux formateurs qui pourront suivre, évaluer les apprenants et adapter le parcours de formation initial en parcours individuel pour chaque apprenant pour qu'il acquière l'ensemble des compétences souhaitées à leur rythme d'apprentissage.

La partie administration sera confiée à un administrateur qui aura la charge de la gestion des objets pédagogiques disponibles dans DIWEB (ajout, suppression et mise à jour des ressources).

En ne faisant pas de distinction du type d'utilisateur, on peut relever les principales composantes de la plateforme DIWEB qui seront :

- ✓ La mise en place d'un ensemble d'outils et de services de formation utilisables par les formateurs et les apprenants.
- ✓ La mise à disposition d'un ensemble d'objets pédagogiques (cours, projet, Travaux Pratiques, Travaux Dirigés, Evaluation, ...)
- ✓ Le pilotage et l'acquisition de données issues de manipulations effectuées sur des

systèmes didactiques hétérogènes accessibles à distance du moment qu'ils soient reliés au réseau Ethernet.

A travers le pilotage à distance de ces systèmes didactiques, la plateforme proposera la réalisation de téléTP dont la partie manipulation pourra être réalisée à travers le système didactique et la partie exploitation des résultats sera facilitée par l'IHM de la plateforme.

- ✓ L'administration de ces différents objets pédagogiques (mise à jour, modification, suppression, insertion).

DIWEB sera mise en œuvre dans un premier temps pour tous les établissements proposant des formations technologiques et professionnelles mais avec pour objectif à terme de l'étendre à l'ensemble du système éducatif.

3. Le panneau photovoltaïque

Au sein de DMS, le système didactique qui a été mis à notre disposition pour réaliser les expériences a été un **panneau photovoltaïque**.

3.1. Présentation et Fonctionnement

Tous les jours, le soleil fournit de l'énergie à la Terre. On peut utiliser cette énergie gratuite grâce à une technologie appelée **photovoltaïque**, qui transforme l'énergie solaire en électricité. Les modules ou panneaux photovoltaïques sont composés de semi-conducteurs qui permettent de transformer directement la lumière du soleil en électricité.

Ces modules peuvent s'avérer une source d'énergie sûre, fiable, sans entretien et non polluante pendant très longtemps. La majorité des modules sur le marché aujourd'hui sont pourvus de garanties de plus de 20 ans, et ils fonctionneront bien au-delà de cette période.

C'est dans ce sens que DMS a décidé de réaliser le système didactique panneau photovoltaïque. Le fonctionnement des ampoules simule le soleil tandis que les ventilateurs simulent les conséquences du vent sur les panneaux photovoltaïque. On peut jouer sur l'intensité lumineuse et l'intensité des ventilateurs et calculer l'énergie produite en fonction de ces deux paramètres.

La première image (figure 11) représente le système dans son ensemble avec à l'avant du système le coffret électrique et, l'IHM de commande ainsi que la centrale de mesures (voir figure 12) situés sur la porte avant. On retrouve à l'arrière du système la cage contenant l'ensemble de la partie opérative et les panneaux photovoltaïques (voir la figure 13). La cage est accessible par une porte.

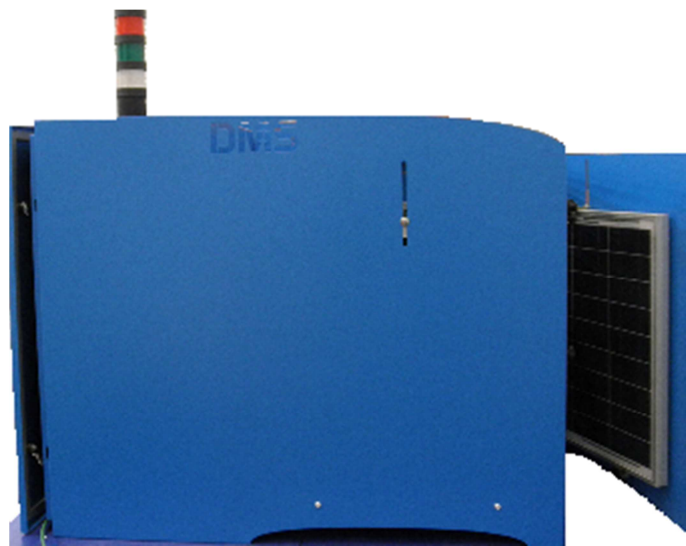


Figure 11: Vue de profil du Panneau Photovoltaïque



Figure 12: IHM de commande du Panneau Photovoltaïque

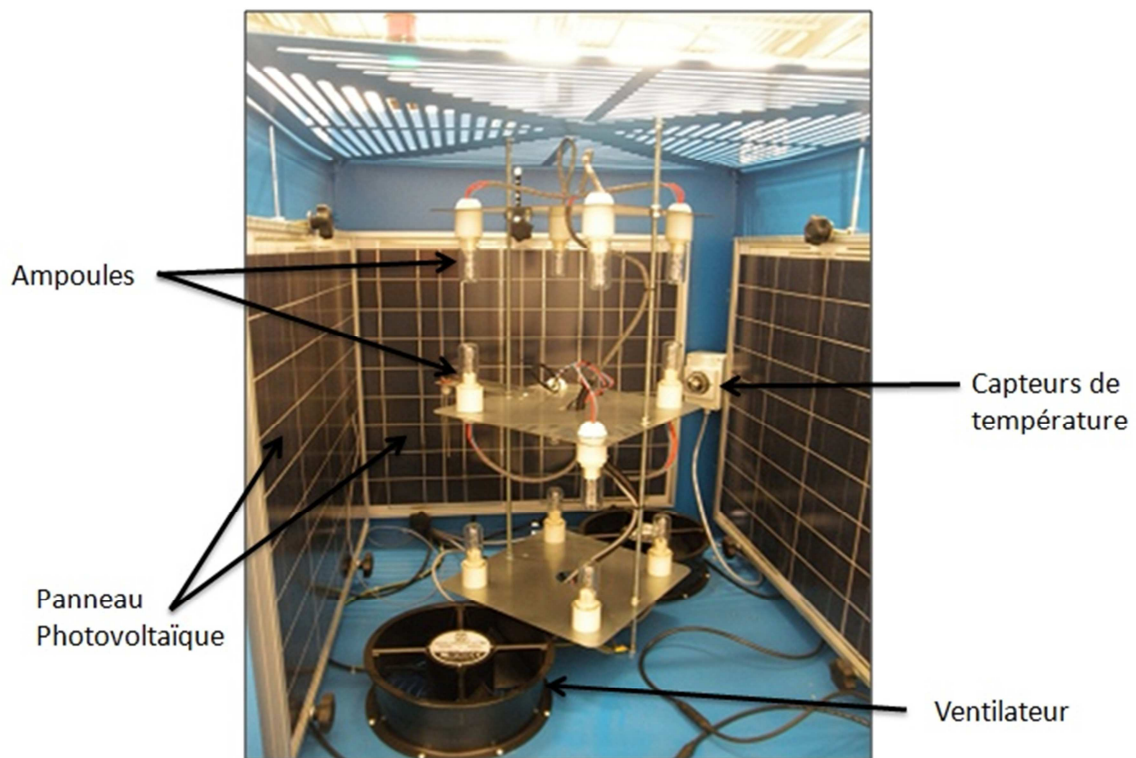


Figure 13: Système Panneau Photovoltaïque – Intérieur de la cage

3.2. Composition

Comme tout système automatisé, le panneau photovoltaïque est composé de trois parties :

- ✓ La **partie opérative** : est constituée d'un ensemble de onze ampoules et de deux ventilateurs. Le tout est contenu dans une cage à l'intérieur du système didactique. La cage est entourée de quatre panneaux photovoltaïques. Des capteurs de température ambiante à l'intérieur de la cage et à la surface des panneaux sont également présents.
- ✓ La **partie commande** : est quant à elle constituée d'un coffret électrique industriel comportant les alimentations et les protections électriques, d'un gradateur, d'un variateur mais également d'un automate programmable industriel, le M340 de la marque Schneider-Electric. Un module serveur web mais aussi des modules de communication Ethernet et Modbus ont été rajoutés à l'API.
- ✓ L'**IHM de commande** : permet de contrôler la puissance d'éclairage des ampoules et de marche des ventilateurs.

3.3. Les dispositifs environnementaux

Pour rendre encore plus réel une manipulation à distance sur le système panneau photovoltaïque, nous avons rajouté une **caméra IP** équipé d'un micro qui filme l'intérieur du système et donc permet de visualiser le fonctionnement des ampoules et des ventilateurs. Le micro est nécessaire pour entendre les ventilateurs tourner.

Cependant comme on peut le voir sur la photo ci-dessous, la façon dont est positionnée la caméra n'est pas optimale car cela nécessite de laisser la porte ouverte de la cage contenant la partie opérative du système. Or cette porte contient côté intérieur un panneau photovoltaïque ce qui en la laissant ouverte va diminuer le rendement de ce panneau et donner des résultats qui ne correspondront pas forcément à la réalité.

Un autre système devra être trouvé, en faisant une ouverture sur le dessus de la cage et fixer la caméra pour qu'elle filme dans cette ouverture afin de ne pas fausser les rendements

sur les panneaux photovoltaïques.



Figure 14: Caméra IP

3.4. Utilisation

Pour utiliser le système, il existe 3 modes permettant de réaliser des mesures d'énergie produite par les panneaux :

- ✓ **Manuel** : on a la main sur tout le système, il n'y a aucune contrainte, on peut régler l'éclairage et la ventilation comme on l'entend. Pour des raisons de sécurité, l'éclairage s'arrête lorsque la température ambiante arrive à 50°C. Ce mode n'est accessible que par les formateurs.

- ✓ **Automatique** : on donne les consignes de température à la surface des panneaux solaires et d'éclairage. Lorsqu'on arrive à la température souhaitée, le système se régule tout seul pour essayer de stabiliser la température à celle souhaitée. On peut également faire rentrer en jeu le paramètre de ventilation en activant les ventilateurs.

- ✓ **Régulation** : Ce mode est moins souple que le mode manuel, c'est une sorte de pilotage automatique en fonction des consignes de température de surface des panneaux solaires et ambiante et PID donnés (P -> Gain, I -> Intégrale, D -> Dérivée) qui va jouer sur l'éclairement ou la ventilation pour permettre de stabiliser la température de surface ou ambiante. Si on ne mettait pas ces consignes, la température ferait du yoyo (soit au-dessus, soit en dessous).

III. Proposition d'intégration de systèmes didactiques dans une plateforme d'apprentissage

En s'intéressant à l'intégration des systèmes automatisés didactiques dans une plateforme d'apprentissage, nous nous sommes donc recentrés sur la partie téléTP de DIWEB. Dans cette section nous ferons une analyse sur l'organisation et le déroulement des TP en présentiel pour les formations professionnelles et technologiques afin de savoir comment il est possible de l'adapter aux téléTP. A partir de là, nous pourrons dégager les spécifications qui recenseront l'ensemble des fonctionnalités à développer. Enfin nous ferons une proposition d'architecture permettant d'intégrer des systèmes automatisés didactiques dans une plateforme d'apprentissage. Ces systèmes seront regroupés dans un laboratoire distant.

1. Analyse

1.1. Organisation des TP en présentiel

Comme dit précédemment, nous nous intéresserons aux formations professionnelles et technologiques. La couverture du référentiel de ce type de formation nécessite le découpage du cursus de formation en un certain nombre matières ou de centres d'intérêts. Pour chaque matière on retrouve l'organisation suivante :

- ✓ Des heures de **cours** pour la synthèse et la structuration des connaissances.
- ✓ Des heures de **TD** consacrées à des applications de cours.
- ✓ Des heures de **TP** destinées à l'acquisition, la consolidation ou la validation des compétences et des connaissances.

Actuellement, un nombre d'heures, qui est défini dans le référentiel, est consacré aux TP. Ils se déroulent généralement dans les salles TP de l'établissement ou plusieurs systèmes

didactiques sont regroupés. Des groupes de TP sont constitués (généralement des binômes ou des trinômes). Cependant, certains systèmes sont coûteux et assez lourds en terme de place ; pour certains établissements, il est impossible d'en posséder plusieurs identiques. Durant les mêmes créneaux horaires de TP, tous les groupes ne peuvent donc pas travailler en même temps sur le même système. Pour qu'à la fin de l'année scolaire, l'ensemble des groupes de TP aient pu travailler sur chaque système, un roulement sur les systèmes s'opère de façon à ce que chaque semaine chaque groupe travaille sur un système différent.

1.2. Déroulement des TP en présentiel

Les TP sont un type d'enseignement fondé sur l'apprentissage pratique avec en particulier la réalisation d'expérience permettant de vérifier et compléter les connaissances dispensées dans les cours théoriques. Ils concernent généralement les sciences expérimentales. Contrairement aux autres types de cours qui se passent exclusivement à l'oral ou à l'écrit, les séances de travaux pratiques nécessitent souvent un matériel spécifique. Dans notre étude, ce matériel sera en l'occurrence des **systèmes automatisés didactiques**.

Pour un apprenant, la réalisation d'un TP peut être décrite en quatre étapes :

- ✓ **Lecture du protocole expérimental** : Il s'agit de la liste de tâches expérimentales organisées de façon temporelle et/ou logique, dont l'objectif est de déterminer des valeurs spécifiques en relation avec les hypothèses scientifiques qui sous-tendent l'expérience. Chaque tâche est caractérisée par une intention explicite, par une procédure et par un ensemble de paramètres dont les valeurs doivent être fixées. Le terme de protocole expérimental fait donc à la fois référence aux actions à réaliser dans un certain ordre (l'expérimentation elle-même) et aux textes et schémas construits pour décrire et communiquer cette expérimentation. [MARZIN et al, 2010]
- ✓ **Expérimentation** : Les manipulations proprement dites consistant à tester par des expériences la validité d'une hypothèse. Pour des systèmes automatisés didactiques, les apprenants se servent de l'IHM de commande pour donner des ordres au système. En fonction des ordres donnés, l'état du système change.
- ✓ **Collecte de résultats** : l'élève doit évidemment détailler tous les résultats

expérimentaux collectés pendant le TP, ainsi que leur contexte précis (sans lequel ils sont inutiles). Les résultats sont liés au protocole, à l'expérimentation, et attendent une interprétation. Ces résultats ne peuvent pas être enregistrés et doivent être notés par les apprenants au fur et à mesure du déroulement du TP.

- ✓ **Interprétation des résultats** : C'est la phase clé de la méthode expérimentale, la plus importante du compte-rendu. A cette étape, l'élève tire des conclusions de ses résultats, et surtout des variations de résultats lorsqu'on modifie le protocole expérimental.

L'intégration des systèmes automatisés didactiques dans la plateforme couvrent les étapes 2 et 3 soit l'**Expérimentation** et la **Collecte de résultats** dans le déroulement d'un TP. Par la suite et grâce à la collecte des résultats qui seront sauvegardés, l'apprenant pourra **interpréter** les résultats obtenus pendant l'expérimentation.

2. Spécifications

Avant de dégager les fonctionnalités des utilisateurs, nous avons relevé les besoins essentiels qui devront être respectés dans la proposition finale.

Avant toute chose, le déroulement d'un TP qu'il soit en présentiel ou à distance doit se dérouler de la même manière pour les apprenants. On doit retrouver les quatre étapes de la réalisation d'un TP. Dans notre étude nous avons décidés de ne pas prendre en compte le fait que les groupes de TP pouvaient être composés de deux ou trois apprenants. En effet, on dira qu'un groupe de TP est constitué d'un **seul apprenant**.

2.1. Besoins

2.1.1. Contrainte temporelle

Des contraintes temporelles nécessaires à la faisabilité du téléTP devront être respectées pour que les conditions du TP en présentiel soient reproduites, du moins au niveau de la

réaction du système en fonction des actions lancées par l'apprenant au travers de l'IHM de commande. En effet, on ne peut pas se permettre que les actions lancées par l'apprenant soient prises en compte par l'automate du système ne serait-ce que 10 secondes plus tard car ceci pourra altérer les résultats obtenus. C'est pourquoi la communication entre l'IHM de commande et l'automate du système piloté devra se faire en **temps réel**.

Si ces contraintes temporelles ne peuvent pas être respectées, il ne sera pas possible de réaliser le téléTP sur le système. Nous verrons donc plus tard que les choix techniques utilisés pour la communication avec l'automate seront très importants.

2.1.2. Observabilité du système lors de la manipulation

Le fait de ne pas avoir une vision du système lors d'une manipulation à distance peut constituer un problème majeur. En effet, pouvoir observer le système évoluer en fonction des actions lancées peut permettre aux apprenants de comprendre certaines notions qu'ils ne pourraient pas comprendre si le système n'était pas visible. Ceci peut être pallié par l'installation de dispositifs environnementaux comme une caméra ou un micro qui peuvent reproduire la vue et l'écoute autour des systèmes en fonctionnement. Il sera beaucoup plus compliqué voire impossible en revanche de reproduire l'odeur ou le toucher, ce qui sur certaine manipulation peut poser un problème.

La reproduction du système peut être également envisagée avec l'usage d'images virtuelles en 2 ou 3 dimensions.

2.1.3. Sécurité des systèmes

La sécurité des systèmes mais également des humains peut être problématique. En effet, on peut imaginer que pendant une manipulation à distance faite par des apprenants, le système subisse une panne technique ce qui peut emmener des comportements non maîtrisés et dangereux du système. Il paraît donc peu probable de laisser l'utilisation des systèmes à distance aux apprenants 24h/24. Une présence humaine serait donc souhaitable durant les

périodes d'utilisation des systèmes.

De plus les systèmes étant connectés sur le réseau informatique de l'établissement, pour y accéder depuis l'extérieur, un point d'entrée sur ce réseau devra exister. C'est dans ce cas que des failles de sécurité peuvent arriver (intrusions, virus, ...). Pour éviter ces failles, il faudra prévoir la mise en place d'un système de sécurité performant.

2.1.4. Temps de manipulation d'un système

Il faut bien prendre en compte le fait qu'un système ne pourra être piloté que par un seul groupe à la fois ce qui implique que lorsqu'un groupe est déjà en train de piloter le système, un autre groupe ne pourra pas le piloter. Pour que l'équité soit respectée entre tous les apprenants, il faudra mettre en place un temps de connexion par apprenant sur chaque système pour que le temps de connexion soit le même pour tous. Lorsque l'apprenant se connectera sur le système, le temps se décomptera. Lorsque le temps sera écoulé, l'apprenant ne pourra plus se connecter sur le système pour le piloter mais pourra toujours interpréter les résultats trouvés.

2.1.5. Créneaux Horaires

Pour éviter les pics d'affluence sur un système, il faudra prévoir la mise en place sur chaque système d'un calendrier de réservation ou chacun pourra réserver des créneaux horaires durant lesquels il souhaitera piloter le système. Une fois le créneau horaire fini, l'apprenant sera déconnecté. S'il souhaite pouvoir de nouveau piloter le système, il devra réserver un nouveau créneau.

2.2. Fonctionnalités

Trois utilisateurs sont concernés par la mise en place et l'utilisation des systèmes didactiques à distance :

- ✓ L'apprenant
- ✓ Le formateur
- ✓ L'administrateur

Nous allons décrire les principales fonctionnalités pour chacun de ces utilisateurs.

2.2.1. L'Apprenant

L'apprenant est l'utilisateur principal de la plateforme d'apprentissage et celui qui fera les TP à distance.

Activités possibles	Fonctionnalités	Description
Se connecter sur un système	Visualiser le laboratoire distant	Voir l'ensemble des systèmes didactiques présents dans le laboratoire distant de l'établissement.
	Ouvrir une session de manipulation	Lancer une connexion sur le système sélectionné.
Communiquer avec le système	Piloter le système	Démarrer/Fermer une connexion. Remise A Zéro (R.A.Z) du système.
		Modifier l'état des actionneurs du système par l'IHM de commande.
	Collecter des résultats	Enregistrer la session de manipulation en cours qui contient toutes les données de la manipulation en train d'être réalisée.
Observer la manipulation sur le système	Visualiser le système	Visualiser le système en cours de fonctionnement au travers d'images réelles issues d'une caméra.
	Ecouter le système	Entendre le système en cours de fonctionnement au travers du son capté par un micro.
	Visualiser les résultats	Observer l'état des capteurs/actionneurs (données sous forme numérique ou graphique).
Exploiter les données de manipulation	Interprétation des résultats	Visualiser les résultats obtenus lors de la manipulation du système. Visualisation des résultats sous forme numérique ou sous forme de courbes.

Lors du pilotage du système, la possibilité de remise à zéro (R.A.Z) du système ne sera pas

possible pour tous les systèmes ou du moins elle ne sera pas immédiate. En prenant exemple du panneau photovoltaïque, un certain temps doit être respecté pour que la température de surface des panneaux solaires et la température ambiante redescendent et retrouvent un niveau normal pour que l'apprenant puisse commencer à l'utiliser.

2.2.2. L'Administrateur

C'est l'administrateur qui s'occupera de la gestion du laboratoire distant de l'établissement.

Activités possibles	Fonctionnalités	Description
Administrer le laboratoire distant de l'établissement	Connecter/Déconnecter un système	Connecter un système sur le laboratoire distant pour qu'il puisse être utilisable ou déconnecter un système en cas de panne ou autre.
	Enregistrer le laboratoire	Enregistrer les données du laboratoire distant.
	Connecter/Déconnecter un dispositif environnemental	Connecter un dispositif environnemental d'un système pour qu'il puisse être utilisable ou en déconnecter en cas de panne ou autre.
Administrer les systèmes didactiques	Créer un système didactique	Connecter l'API du système, lier l'IHM de commande au système, renseigner les informations du système (nom, fiche technique, ...), ajouter la documentation du système (notice, documents techniques).
	Modifier un système	Modifier les configurations sur le système.
	Supprimer un système	Supprimer le système du laboratoire distant.
	Ajouter/Supprimer des dispositifs environnementaux	Ajouter/Supprimer des dispositifs environnementaux au système (caméra, micro)

2.2.3. Le formateur

Le formateur se limitera aux fonctionnalités de type administration, c'est-à-dire la gestion

des apprenants et la supervision de l'utilisation des systèmes didactiques présents dans le laboratoire distant. On ne rentre pas dans toutes les fonctionnalités liées à la supervision des TP des apprenants.

Activités possibles	Fonctionnalités	Description
Gérer les apprenants	Gérer groupes de TP	Créer/Modifier/Supprimer un groupe de TP. Un groupe de TP est constitué d'un seul apprenant.
Superviser l'utilisation des systèmes	Gérer les droits d'accès	Gérer les droits d'accès d'utilisation des systèmes. Les apprenants ne peuvent pas avoir accès à tous les systèmes du laboratoire distant, c'est pourquoi pour chaque système des droits d'accès seront nécessaires.

2.3. Modélisation

Dans cette partie nous utiliserons la **modélisation** pour mieux comprendre comment nous pouvons intégrer des systèmes automatisés didactiques dans une plateforme d'apprentissage. C'est une étape importante dans la vie d'un projet informatique, permettant d'avoir une bonne architecture et par la suite un code de qualité et plus facilement maintenable.

De nombreuses méthodes de modélisation existent. Il n'y a pas forcément de bonnes ou mauvaises méthodes de modélisation mais nous avons choisi **UML**, méthode de modélisation orientée-objet la plus répandue dans le monde informatique et appropriée pour ce type de projet. Nous avons fait une explication plus détaillée d'UML à l'annexe 1.

2.3.1. Les cas d'utilisation

Les cas d'utilisation permettent d'effectuer une bonne délimitation des fonctionnalités du système informatique attendu et également d'améliorer la compréhension de son fonctionnement.

Ici, les cinq cas d'utilisations schématisés vont structurer les besoins de l'apprenant, du formateur et de l'administrateur mais également du serveur web applicatif où sera stocké la plateforme d'apprentissage lorsqu'il communiquera avec l'automate du système didactique.

2.3.1.1. Cas d'utilisation « Réaliser l'expérimentation et la collecte de résultats d'un TP »

Ce cas d'utilisation représente les besoins nécessaires à l'étape 2 et 3 de la réalisation d'un TP, soit l'**expérimentation** et la **collecte des résultats**. L'acteur principal est l'apprenant qui va réaliser le TP.

Avant de pouvoir faire une manipulation sur le système, l'apprenant doit être authentifié sur la plateforme. Il se connecte sur le laboratoire distant de son établissement et choisit le système qu'il souhaite manipuler pour pouvoir faire le TP. Une **session de manipulation** sur le système est créée s'il a les droits nécessaires pour le faire ou que le système soit disponible c'est-à-dire qu'il ne soit pas en cours d'utilisation ou en panne (Activité « Lancer une session de manipulation »).

Une fois la session de manipulation créée, au travers de l'IHM de commande, il peut piloter le système en lançant des actions (Activité « Piloter le système ») mais également observer le système (Activité « Lire les données de manipulation »). Pour ces deux activités il y a communication avec l'automate du système didactique pour lire ou écrire sur un ensemble de bits ou registres présent dans la mémoire de l'automate. Ces blocs de bits ou registres représentent les variables. Avant de réaliser des opérations de lecture ou d'écriture il faut que la connexion avec l'automate du système didactique soit effective.

On peut visualiser ou écouter le système en cours de fonctionnement en affichant le flux

vidéo de la caméra ou en lisant le flux audio du micro si ces deux dispositifs environnementaux sont disponibles sur le système choisi.

Concernant l'enregistrement des données liées à la manipulation du système, l'apprenant à la possibilité de sauvegarder au travers de la session de manipulation les données suivantes durant tout le temps qu'il manipule le système :

- ✓ Toutes les actions liées à la modification de l'état des actionneurs du système ou actions de simulation. La sauvegarde de ces actions sera nécessaire à l'apprenant lors de l'interprétation des résultats, pour qu'il puisse se souvenir des modifications précises qu'il a apportées au système.
- ✓ Toutes les données obtenues sur les capteurs et les actionneurs du système en temps réel.

Les sessions de manipulation sont sauvegardées en base de données.

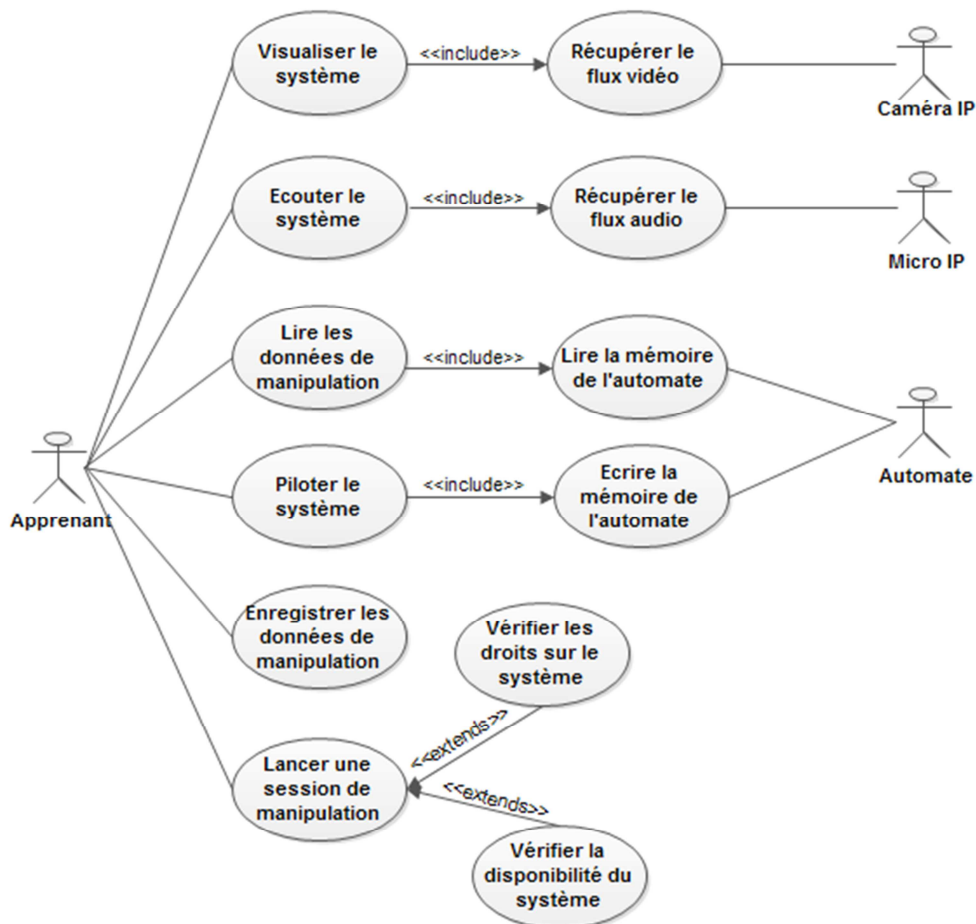


Figure 15: Cas d'utilisation "Réaliser l'expérimentation et la collecte des Résultats"

2.3.1.2. Cas d'utilisation « Interpréter les résultats d'une manipulation »

Une fois la manipulation terminée sur le système, l'apprenant doit interpréter les résultats en exploitant les données de manipulation qu'il a pu collecter. Il s'agit de l'étape 4 (**interprétation des résultats**) de la réalisation d'un TP.

L'apprenant doit être authentifié sur la plateforme d'apprentissage pour afficher les différentes sessions de manipulation sauvegardées. Après avoir chargé une session de manipulation, il peut naviguer dans l'espace de données et afficher les données qui ont été sauvegardées durant la manipulation sur le système (représenté par l'activité « Restituer une session de manipulation »). De ces données on retrouve l'ensemble des résultats sauvegardés sur les capteurs et actionneurs du système par l'apprenant.

Il peut consulter l'historique de ses sessions de manipulation dans le but d'en charger une.

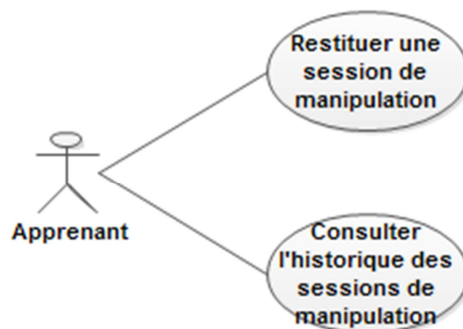


Figure 16: Cas d'utilisation "Interpréter les résultats d'une manipulation"

2.3.1.3. Cas d'utilisation « Administrer un laboratoire distant »

L'administration d'un laboratoire distant d'un établissement est faite par un administrateur. Pour rappel, un laboratoire distant est composé de systèmes didactiques accessibles à distance.

En cas de panne ou de problèmes techniques d'un système, l'administrateur peut déconnecter le système du laboratoire distant de l'établissement pour qu'il ne soit plus visible par les apprenants (symbolisé par l'activité « Connecter/Déconnecter un système »). On est sur le même principe pour les dispositifs environnementaux des systèmes, en cas de problèmes il y a possibilité de déconnecter n'importe lequel pour qu'il ne soit plus utilisable lors des manipulations (Activité « Connecter / Déconnecter un dispositif environnemental »).

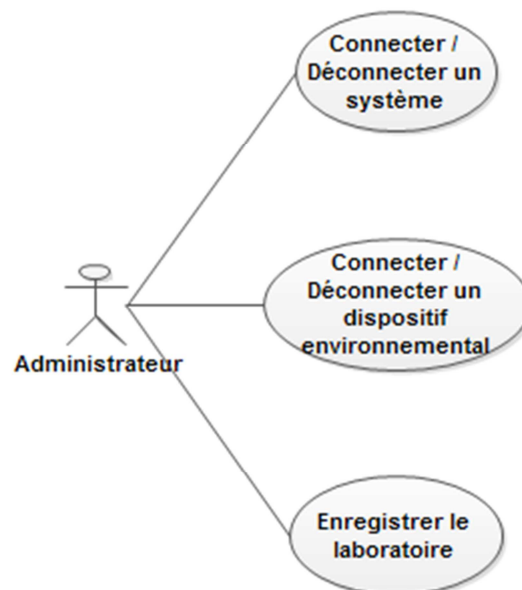


Figure 17: Cas d'utilisation "Administrer un laboratoire distant"

2.3.1.4. Cas d'utilisation « Administrer des systèmes didactiques »

L'administration d'un système didactique est également faite par l'administrateur. La gestion de l'ensemble des systèmes du laboratoire distant (création, modification, suppression) est son rôle principal. Les principaux paramètres dans la création d'un système et donc l'ajout de celui-ci dans le laboratoire distant sont la connexion avec l'automate, l'ajout de l'IHM de commande, l'ajout de la documentation du système mais aussi la gestion des dispositifs environnementaux (caméra, micro).

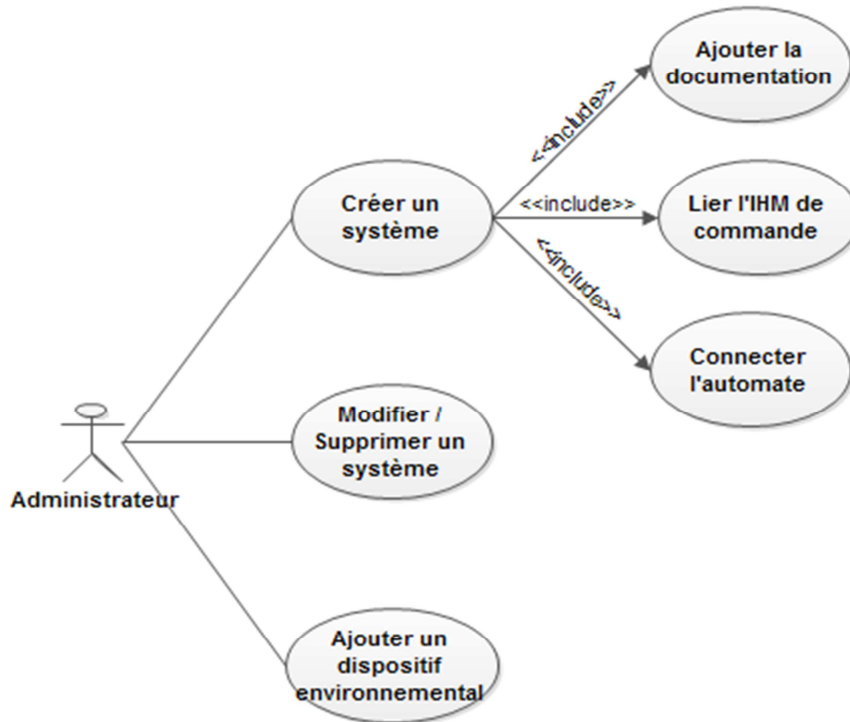


Figure 18: Administrer un système didactique

2.3.1.5. Cas d'utilisation « Administrer l'utilisation des systèmes »

Ce cas d'utilisation montre le rôle du formateur dans l'organisation des téléTP et l'utilisation des systèmes. C'est lui qui va attribuer les droits d'accès permettant aux apprenants la possibilité de se connecter sur les systèmes présents dans le laboratoire distant de l'établissement. C'est également le formateur qui va gérer les apprenants en formant les groupes de TP. Dans notre cas la question ne se pose pas puisque un groupe de TP est constitué d'un seul apprenant.

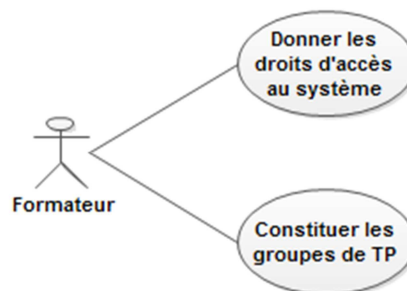


Figure 19: Cas d'utilisation "Administrer l'utilisation des systèmes"

2.3.2. Diagramme de déploiement

Après avoir défini l'ensemble des besoins, le diagramme de déploiement va nous permettre d'avoir une première approche sur la disposition physique des matériels qui composent notre solution et la répartition des composants sur ces matériels.

On retrouve les trois matériels principaux de notre architecture physique :

- ✓ le **serveur web applicatif** : contenant la plateforme d'apprentissage et la base de données.
- ✓ les **dispositifs environnementaux** : essentiellement la caméra IP et le micro IP pour rendre plus réelle les manipulations à distance. La communication entre le serveur web applicatif et les dispositifs IP se fait en utilisant le protocole TCP/IP.
- ✓ les **automates de chaque système automatisé didactique du laboratoire distant** : pour être accessible à distance, un automate doit avoir au minimum soit une passerelle TCP/IP, soit un serveur web pour établir une communication TCP/IP. Il est possible par la suite d'atteindre les variables du programme automate stockées dans la CPU de l'automate à l'aide de bus de terrain comme Modbus ou Profibus.

Trois types d'utilisateurs (apprenant, formateur et apprenant) sont matérialisés par les trois postes. C'est en utilisant la plateforme d'apprentissage, qu'ils communiquent avec le serveur web applicatif. L'accès à la plateforme est sécurisé, un identifiant et un login sont demandés à l'utilisateur pour pouvoir s'y connecter.

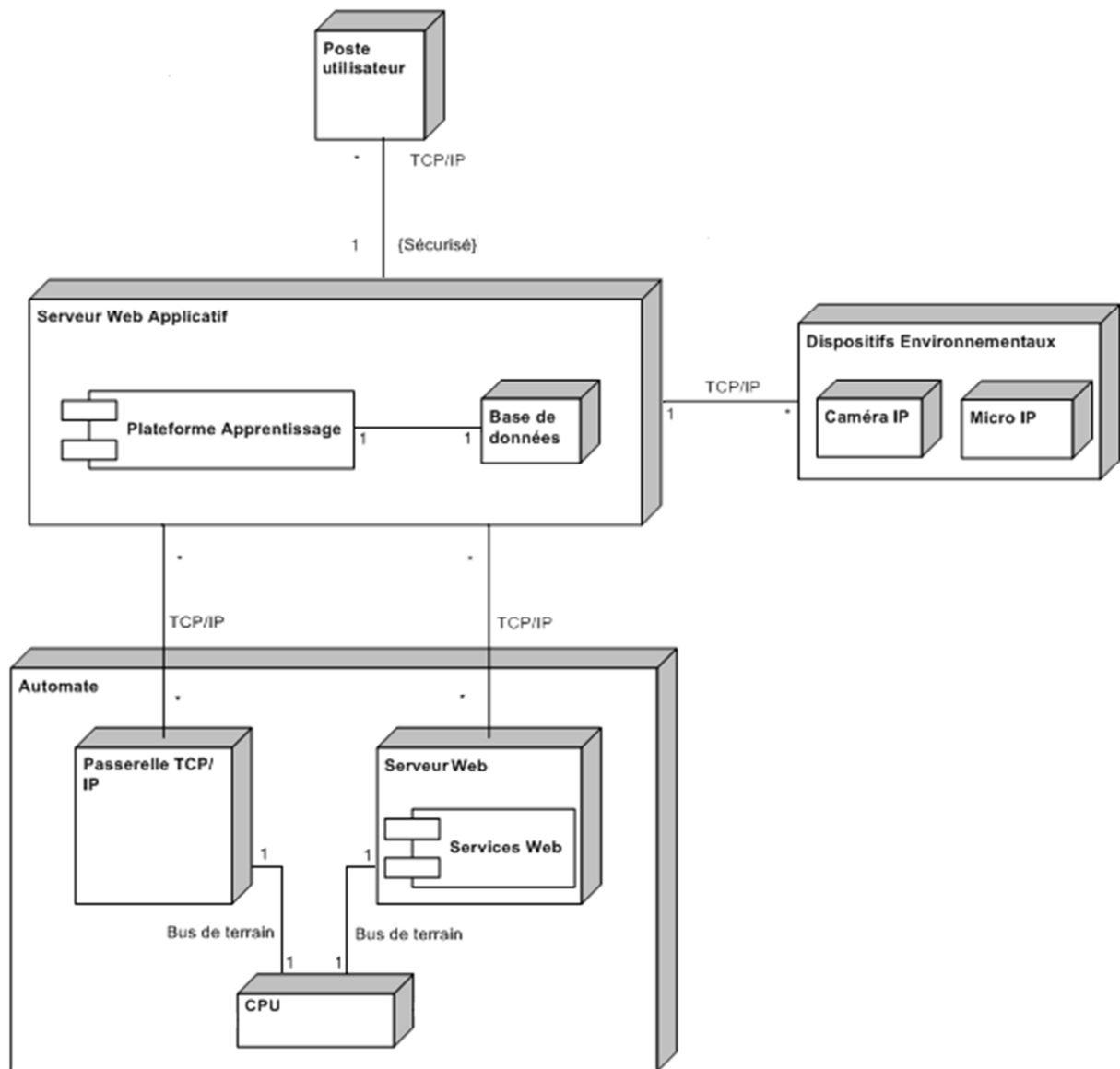


Figure 20: Diagramme de déploiement

2.3.3. Diagramme de classes

Avec ce diagramme nous allons avoir une connaissance de l'ensemble des classes qui vont composer notre proposition de mise en place d'intégration d'un laboratoire distant dans une plateforme d'apprentissage. On décrit la structure interne statique précise de chacune des classes ainsi que les relations entre les classes mises en œuvre.

Nous avons également introduit les designs patterns pour simplifier l'organisation des

classes. Ces **designs patterns** ou **patron de conception** décrivent des organisations pratiques de classes d'objets qui forment une solution à un problème connu et fréquent. Ces designs patterns sont constitués par un ensemble d'objets décrits par des classes et des relations liants ces objets. [DEBRAUWER, 2009]

Ils répondent à des problèmes de conception de logiciels dans le cadre de la programmation par objets. Ce sont des solutions connues et éprouvées dont la conception provient de l'expérience de programmeurs. Il n'y a aucun aspect théorique dans les designs patterns et surtout pas de formalisation.

Ils sont classés en trois catégories :

- ✓ Les **patrons de conception** : ils délèguent à d'autres classes la construction des objets.
- ✓ Les **patrons de structure** : ils encapsulent la composition des objets augmentant le niveau d'abstraction du système.
- ✓ Les **modèles de comportement** : ils tentent de répartir les responsabilités entre chaque classe.

Nous avons fait le choix de décomposer notre architecture en **cinq couches** (Laboratoire, Système, IHM, Actions et Proxy), des couches haut niveau aux couches bas niveau. Chaque niveau de couche a ses responsabilités et fournit ses services au niveau supérieur. Dans notre cas avoir un niveau d'abstraction assez élevé permet de cacher la complexité de la communication avec l'automate. En effet, ce sont les classes bas niveau qui vont assurer cette communication avec l'automate et donc utiliser un langage compréhensible par l'automate pour pouvoir communiquer avec lui. Les données récupérées sont retransmises et traitées par les couches des niveaux supérieurs. Cela permet entre autre d'avoir une maintenance du code beaucoup plus facile.

On retrouve aussi la couche **persistance** des données respectant le patron de conception **DAO** (Data Access Object) qui est chargée de l'accès aux données et de leur manipulation et

ceux indépendamment du système de gestion de base de données (SGBD) choisi. Elle permet de faire l'illusion que notre base de données relationnelle soit une base de données objet. La classe mère **DAO** doit implémenter les quatre méthodes CRUD (Create, Retrieve, Update et Delete), qui constitue les opérations de base dans la persistance des données. C'est cette couche qui va faire le lien entre les objets métiers et notre SGBD contenant la connexion vers la base de données et l'ensemble du code SQL. A travers les classes **LaboratoireDAO** et **SystèmeDAO**, il est possible de manipuler les objets métiers Laboratoire et Système en base de données.

2.3.3.1. La couche Laboratoire

Elle constitue la couche haute de notre architecture et représente le laboratoire distant de l'établissement. Cette couche est constituée de la classe « Laboratoire ».

La plateforme d'apprentissage utilise cette classe pour pouvoir intégrer les systèmes didactiques. Les principales méthodes disponibles sont :

- ✓ **connecter ()** : connecte un apprenant sur le laboratoire distant. L'apprenant rentre dans la zone de téléTP de la plateforme.
- ✓ **déconnecter ()** : déconnecte un apprenant du laboratoire distant. L'apprenant quitte la zone de téléTP de la plateforme.
- ✓ **listeSysteme ()** : liste l'ensemble des systèmes présent dans le laboratoire distant de l'établissement.
- ✓ **ajouterSysteme ()** : permet d'ajouter un système dans le laboratoire distant.
- ✓ **modifierSysteme ()** : modifie un système (informations, IHM de commande, automate, documents, dispositifs environnementaux).
- ✓ **supprimerSysteme ()** : supprime un système du laboratoire.
- ✓ **enregistrer ()** : enregistre l'état du laboratoire distant en base de données.

2.3.3.2. La couche Système

On descend d'un niveau, cette couche représente les systèmes didactiques. On retrouve les classes « Système », « Dispositifs », « Caméra » et « Micro ». La classe principale est la classe « Système ». Ses principales méthodes sont :

- ✓ **piloter ()** : cette méthode ouvre une session de manipulation à l'apprenant, s'il en a les droits nécessaires. L'IHM de commande du système va s'afficher et la connexion à l'automate va s'opérer pour que les actions puissent être lancées.
- ✓ **connecter ()** : connecte le système pour qu'il soit accessible aux apprenants. Cela revient à se connecter à l'automate du système à l'aide de son adresse IP.
- ✓ **déconnecter ()** : déconnecte le système du laboratoire de façon à ce qu'il ne soit plus visible par les apprenants. Cela revient à se déconnecter de l'automate.
- ✓ **lierIHM ()** : permet de lier l'IHM de commande au système.

Elle possède également la propriété « Droits » qui est un tableau contenant les apprenants ayant le droit de piloter le système.

Cette classe est un **singleton** pour s'assurer qu'elle ne possède qu'une seule instance. Pour cela, elle fournit une méthode de classe unique retournant cette instance. Il s'agit de la méthode **Instance ()**. Elle nous est très utile pour s'assurer que le système ne soit piloté que par un apprenant. Lorsqu'un apprenant commence la manipulation sur le système, la méthode Instance signale que le système est en cours d'utilisation et qu'il ne peut être de nouveau utilisable que lorsque la manipulation actuelle a été fermée.

La classe mère « Dispositif » dispose de la méthode **connecter ()** permettant de créer la communication avec un dispositif environnemental. Elle possède deux classes filles « Caméra » et « Micro » qui hérite donc de cette méthode et qui par la suite permet d'utiliser ces deux dispositifs environnementaux.

2.3.3.3. La couche IHM de commande

Cette couche est composée de la classe « IHM ». Cette classe représente l'IHM de commande pour piloter le système à distance. L'affichage de l'IHM se fera par la méthode **afficher ()**.

2.3.3.4. La couche Actions

Cette couche représente l'ensemble des actions qu'il sera possible de réaliser sur le système ou sur les dispositifs environnementaux du système. Elle est constituée des classes « ActionSysteme », « ActionDispositif », « Observation », « Simulation », « Etat », « CommandeCamera » et « CommandeMicro ».

Nous avons distingué deux types d'actions réalisables à travers l'IHM de commande : les actions système et les actions dispositif.

Les Actions système

Les actions système représentent les actions lancées sur l'automate du système. Nous avons décidé de les classer en trois catégories :

- ✓ Action d'**observation** : permettent d'obtenir la valeur d'une variable de l'automate. Variable pouvant stocker la valeur de l'état des capteurs du système (température, ...) ainsi que des données sur ces actionneurs en temps réel (valeur du courant, tension, intensité, puissance, ...).
- ✓ Action de **simulation** : permettent de modifier la valeur de la variable de l'automate gérant l'état de l'actionneur que l'on souhaite voir changer de comportement. En prenant exemple sur les ventilateurs du système panneau photovoltaïque, changer la variable gérant l'état du fonctionnement de ces derniers reviendrait à les faire ventiler plus ou moins fort.
- ✓ Action d'**état** : dans cette catégorie, on retrouve des actions comme la remise à zéro du système, ou l'enregistrement des données de manipulation sur la session de manipulation de l'apprenant.

Les Actions des dispositifs environnementaux

Ce type d'action représente les actions permettant de contrôler les dispositifs environnementaux (caméra, micro) du système.

- ✓ Pour une **caméra**, ce sont les actions donnant la possibilité de visualiser le flux vidéo de la caméra, de jouer sur le zoom, de diriger la caméra (de bas en haut ou de droite à gauche) ou de jouer sur la luminosité.
- ✓ Pour un **micro**, il s'agit des actions permettant de lire le flux audio du micro ou de monter ou descendre le son.

2.3.3.5. La couche Proxy

Dans cette couche, nous avons introduit le patron de conception **Proxy** correspondant à la conception d'un objet qui se substitue à un autre objet et qui en contrôle l'accès. C'est dans ce sens que nous avons mis en place la couche Proxy qui fait le lien entre les actions systèmes lancées par un apprenant et l'automate du système. Cette couche est composée des classes « AutomateProxy », « Memoire », « Registre » et « Bit » qui vont substituer les composants de l'automate du système.

La classe « **AutomateProxy** » représente l'automate, « **Memoire** » la mémoire des données de l'automate découpée en bit et registre qui sont représentées par les classes « **Bit** » et « **Registre** ». Ces deux dernières classes sont des classes filles de « Memoire » et héritent notamment des méthodes **lire ()** et **ecrire ()** permettant de lire et écrire sur la mémoire de l'automate du système.

Cette couche, comme son nom l'indique, joue le rôle de proxy et permet de filtrer les accès à la mémoire de l'automate. Il est donc tout à fait possible de régler l'accès aux variables en ajoutant des filtres qui vont les rendre accessibles ou non accessibles.

De plus, si n'importe quel problème est détecté sur l'automate (alerte, dysfonctionnement, ...), la classe « AutomateProxy » jouera le rôle de tampon. Les demandes seront bloquées par cette classe au lieu d'être envoyées directement sur l'automate.

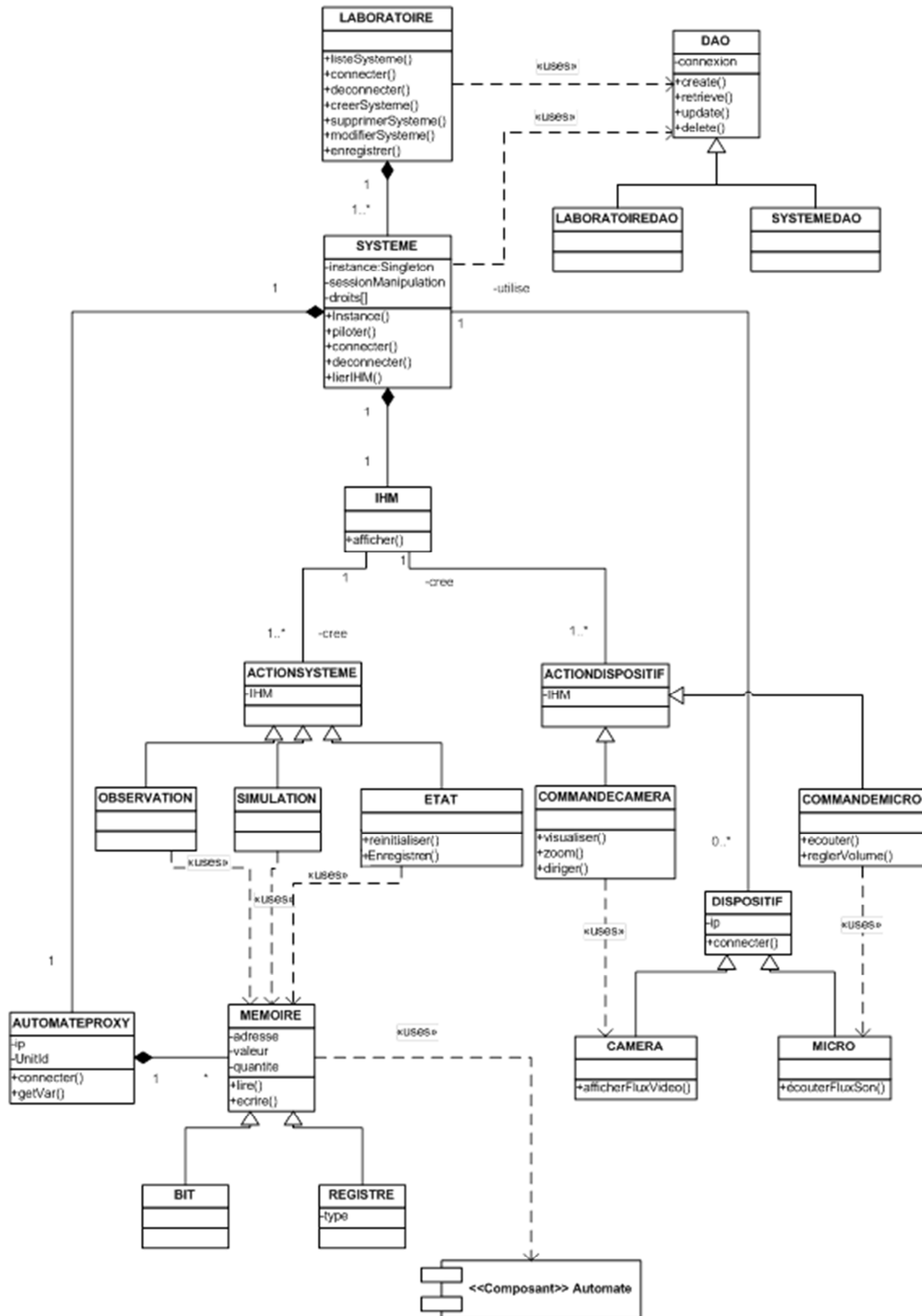


Figure 21: Diagramme de classes Laboratoire Distant

Le diagramme des classes suivant fait état du lien entre la plateforme d'apprentissage sur laquelle les apprenants sont connectés et le laboratoire distant. En utilisant la méthode **connecter()** de la classe Laboratoire on ouvre une connexion pour l'apprenant demandeur. De même lorsqu'une apprenant souhaite piloter un système présent dans le laboratoire distant, une session de manipulation se crée via la création d'un objet **SessionManipulation**. On retrouve les méthodes **enregistrer()** et **charger()** permettant respectivement d'enregistrer une session de manipulation en base de données ou de charger une session pour observer les données de manipulation.

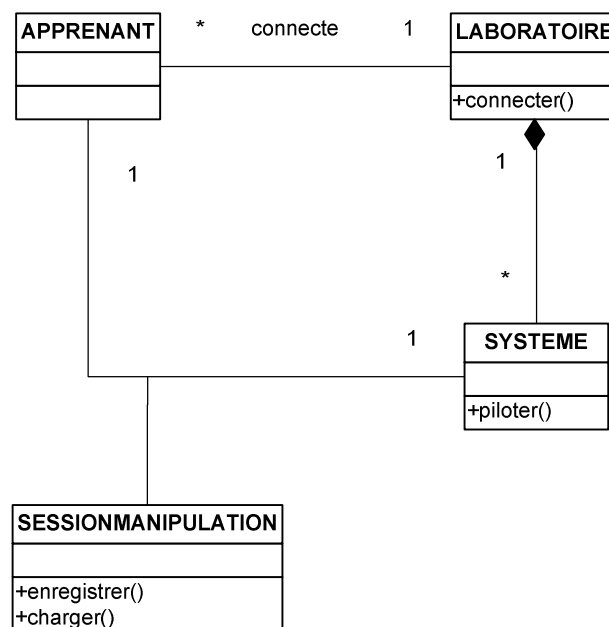


Figure 22: Diagramme de classes "Apprenant - Laboratoire"

2.3.4. Diagramme de composants

Ce diagramme de composants nous permet d'avoir une vue statique de l'implémentation de la solution et illustre les choix de réalisation.

On distingue de manière plus distincte les cinq couches de notre architecture (Laboratoire, Système, IHM, Actions et Proxy). La couche Proxy pourra dialoguer avec l'automate à l'aide de la bibliothèque « Services Web » qui présente des services permettant d'accéder aux variables du programme automate. Si cette bibliothèque n'est pas présente, il sera possible

de communiquer via la passerelle TCP/IP de l'automate que nous n'avons pas présenté sur ce diagramme.

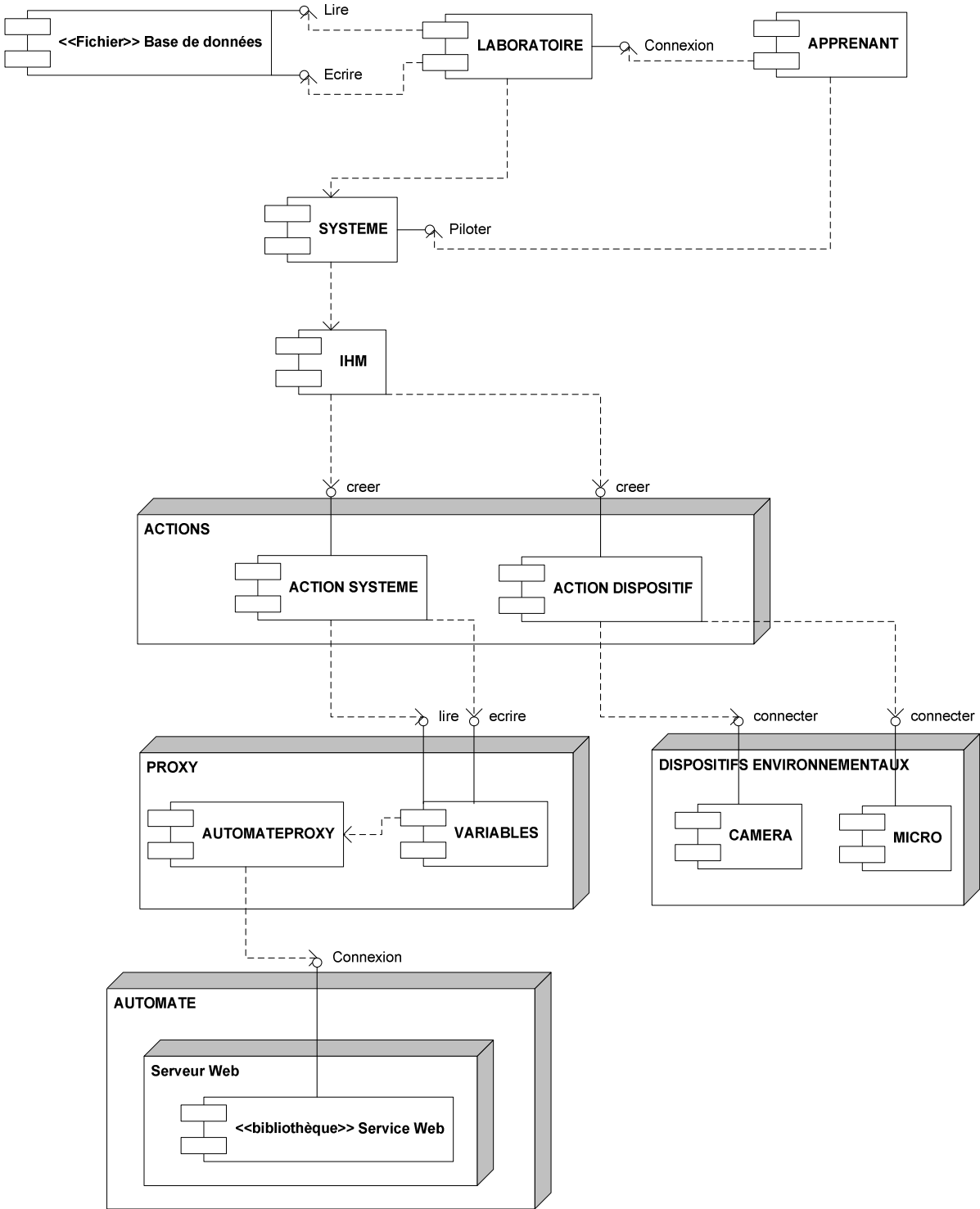


Figure 23: Diagramme de composants

IV. La conception

Dans cette section, nous allons mettre en œuvre la proposition d'intégration de systèmes automatisés didactiques au sein d'une plateforme d'apprentissage. Cette mise en œuvre concerne d'une part la structure du réseau informatique d'un établissement qui accueillera la plateforme et d'autre part les choix techniques concernant la communication entre le serveur web applicatif et l'automate du système didactique pour qu'elle soit en temps réel.

1. Réseau Informatique d'un établissement

Pour pouvoir se connecter sur les systèmes didactiques, l'ouverture du réseau informatique de l'établissement sur le monde extérieur est nécessaire, ce qui peut laisser entrevoir un risque accru de failles informatiques en tout genre. Il s'agit donc de sécuriser au maximum le réseau informatique de l'établissement pour éviter le moins de failles possibles.

En terme d'architecture technique, la plateforme d'apprentissage est stockée sur un serveur web applicatif contenant également une base de données pour le stockage des données. C'est à partir de ce serveur web applicatif que la communication avec les systèmes didactiques présents dans le laboratoire distant de l'établissement est réalisée.

La sécurisation du réseau local d'un établissement doit être faite de la façon suivante :

- ✓ Le **parefeu** est un système se trouvant derrière le routeur, juste avant le réseau local de l'établissement. Son rôle est de filtrer les données échangées avec le réseau extérieur. Le filtre peut se faire sur la couche 3 du modèle OSI en analysant les adresses IP et les numéros de port mais également au niveau de la couche 7 du modèle OSI en étudiant les protocoles utilisés dans les échanges de données.
- ✓ Une **DMZ** (Zone Démilitarisée ou DeMilitarized Zone anglicisé), est le sous réseau isolé du réseau local de l'établissement par le parefeu. Cette zone fait partie du réseau local de l'établissement. Elle contient toutes les machines étant susceptibles

d'être atteintes depuis l'extérieur par Internet. Dans cette zone on va placer le **serveur web applicatif** et le **laboratoire distant**. Seul le serveur web applicatif possède un nom de domaine, les systèmes didactiques n'ont, eux, qu'une adresse IP locale, ce qui les rend inaccessibles depuis l'extérieur sans passer par le serveur web. Cette zone est configurée de façon à ce que les postes du sous réseau local puissent quand même communiquer avec le serveur web étant dans le DMZ mais les composants de la DMZ n'ont pas d'accès sur ceux du sous réseau local.

- ✓ Un **sous réseau local** composé des postes apprenant, formateur et administrateur. C'est le pare-feu qui bloque les accès au sous réseau local pour garantir sa sécurité. En cas de faille, le pirate aura accès uniquement aux composants de la DMZ.

Ci-dessous, la figure 24 illustre l'organisation du réseau local de l'établissement.

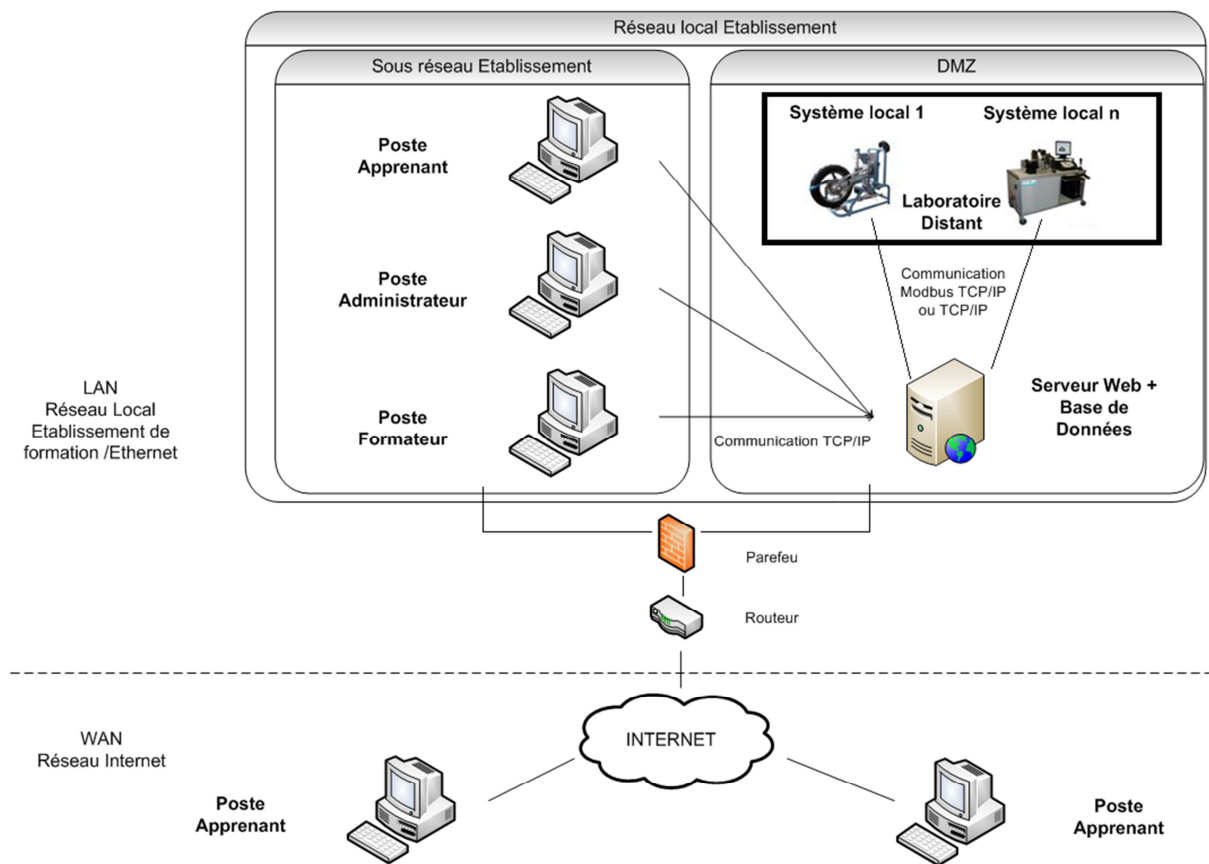


Figure 24: Schéma réseau local d'un établissement

Afin d'avoir une meilleure compréhension des échanges possibles entre les différentes zones, un tableau récapitulatif des actions possibles a été fait.

Action	Réseau	Source	Destination	Protocole, Port
Autoriser	Internet	LAN	WAN	HTTP, 80
Refuser	Internet	WAN	Sous réseau local	*
Autoriser	Internet	WAN	DMZ	HTTP, 80
Refuser	Réseau Local	DMZ	Sous réseau local	*
Autoriser	Réseau Local	Sous réseau local	DMZ	HTTP, 80

A terme, une interconnexion des réseaux de différents établissements sera possible. Un établissement ne possédant pas un système pourra accéder au système manquant en se connectant sur l'automate du système présent dans le laboratoire distant d'un autre établissement. De ce fait l'établissement pourra proposer des TP à distance sur ces systèmes. On aura donc des réseaux de laboratoires distants et accessibles par la plateforme d'apprentissage.

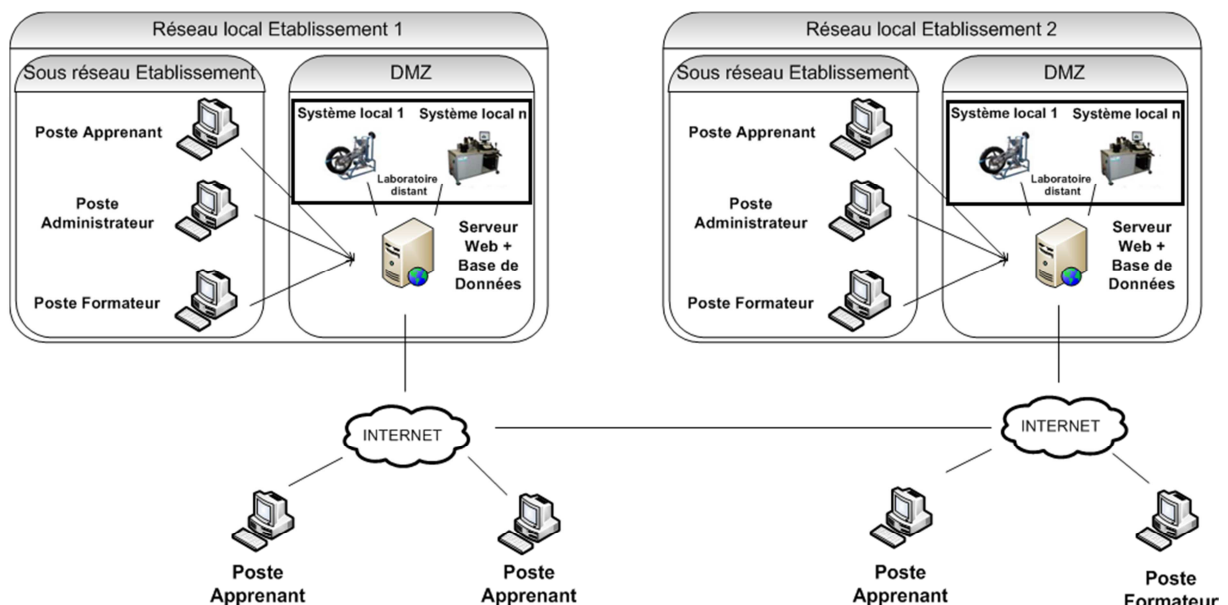


Figure 25: Interconnexion de réseau d'établissement

2. Choix techniques sur la communication avec les automates

Le choix technique primordial réside dans la communication entre le serveur web applicatif et l'automate du système didactique. Les requêtes envoyées à l'automate devront être effectuées en temps réel pour respecter la contrainte temporelle imposée. Les deux protocoles de communication viables pour communiquer avec un automate sont **Modbus** et **ProfiBus**.

Comme dit dans le chapitre 2.2 sur les réseaux locaux industriels, ProfiBus est un protocole de communication propriétaire de Siemens. Or la plupart des systèmes automatisés didactiques étant équipés d'automate de la marque Schneider-Electric, il paraît incompatible d'utiliser le protocole ProfiBus. Modbus étant un protocole de communication non propriétaire et compris par tous les automates, c'est donc vers ce protocole que nous nous sommes orientés.

En utilisant le protocole de communication Modbus, il existe plusieurs façons de communiquer entre le serveur web applicatif et l'automate du système didactique.

- ✓ Communication via TCP/IP entre le serveur web applicatif et le serveur web de l'automate du système. Généralement des **services web** sont utilisables pour accéder à la mémoire des données de l'automate. Le principal étant **ModbusXMLDa**. Ceci est possible uniquement si l'automate est équipé d'un module serveur web. Par ailleurs, la communication entre le serveur web de l'automate et la CPU de l'automate se fait en Modbus.
- ✓ Communication par le protocole de communication Modbus TCP/IP entre le serveur web applicatif et l'automate en passant par une passerelle TCP/IP présent sur l'automate. Trois bibliothèques viables ont été trouvées : **libModbus** (langage C), **JModbus** et **Modbus4j** (langage Java).

Les critères de comparaison des bibliothèques ont été les suivants :

- ✓ La documentation : savoir s'il y a une description exhaustive des différentes fonctions ou classes qui composent la bibliothèque (documentation sur Internet, forum, exemple de codes, ...).
- ✓ Coût : bibliothèque open-source (libre) ou payante lorsqu'elle est utilisée à des fins commerciales.
- ✓ Temps d'exécution des requêtes : temps d'exécution des requêtes en lecture et écriture. L'objectif est d'avoir du temps réel. Un test des temps d'exécution devra être fait au sein du réseau local de l'établissement ou à distance sur la lecture et l'écriture de variables. Bien entendu, les temps d'exécution à distance vont dépendre du débit. Nos tests à distance se sont déroulés dans un lieu présentant un débit de 6Mbits par seconde. Les tests se feront entre le serveur web applicatif et l'automate du système panneau photovoltaïque.
- ✓ Productivité : Facilité de mise en place et d'utilisation permettant un apprentissage rapide pour les développeurs et une maintenance du code simplifiée.

2.1. Communication en TCP/IP entre le serveur web et l'automate

Avec l'implémentation des services web dans les modules serveurs web des API, il est possible d'interfacer simplement sa propre application développée avec n'importe quel langage de programmation avec le monde de l'automatisme. Celle-ci accèdera en temps réel à l'information souhaitée directement dans la mémoire de l'automate, sans interface intermédiaire, ni protocoles propriétaires.

Le principal service web, permettant de faire le lien entre les mondes de l'informatique et de l'automatisme, mis en place dans la plupart des serveurs web des API est de la famille SOAP (Simple Object Access Protocol). Il s'agit du service web **ModbusXMLDa** (Modbus XML Data Access). Ce service web est mis en œuvre pour permettre l'accès aux variables du programme de l'automate que l'on souhaite atteindre.

SOAP désigne un protocole permettant des appels de procédures à distance qui permet l'échange d'informations de type client/serveur.

Les échanges d'informations s'appuient sur les standards du web, **XML** qui est le format universel d'échange de données pour la structure des messages, et **HTTP** pour le transport. Les messages SOAP vont être écrits en XML et être encapsulés dans HTTP.

Les services web sont décrits par des documents **WSDL** (Web Service Description Language) qui précisent les méthodes disponibles, leurs signatures et les points d'accès du service (URL, port). C'est grâce à ce document que les services web sont utilisables. Celui de ModbusXMLDa se trouve sur le serveur web de l'automate.

Voici les principales caractéristiques du service web ModbusXMLDa.

Documentation	<p>La documentation est surtout fournie par le site de Schneider-Electric avec des détails de l'ensemble des fonctions utilisables.</p> <p>Il y a malgré tout, peu d'exemples d'utilisation, ni sur le site Schneider, ni sur le Web en général.</p> <p>Peu de forums, ni d'aide sur le site de Schneider sont disponibles en cas de problèmes rencontrés.</p>
Coût	Open-source.
Productivité	<p>Les services web ne nécessitent aucune installation préalable, ModbusXMLDa ne déroge pas à la règle. Il faut simplement se connecter sur le fichier WSDL.</p> <p>La mise en place et l'utilisation sont très faciles. Il existe six méthodes pour faire l'ensemble des opérations de lecture et d'écriture sur la mémoire de l'automate et une méthode permettant d'avoir l'identification complète du serveur web mis en place sur l'automate.</p> <p>Parmi les six méthodes de lectures et écriture, on retrouve les méthodes permettant de lire et d'écrire sur les registres (ReadMultipleRegisters et WriteMultipleRegisters) mais aussi celles permettant la lecture et l'écriture sur les bits (ReadCoils et WriteMultipleCoils).</p>
Temps d'exécution des requêtes	Que l'on soit en local ou à distance, l'exécution des actions en lecture et écriture se fait en temps réel.

2.2. Communication en Modbus TCP/IP entre le serveur web et l'automate

La principale différence entre ces trois bibliothèques réside dans le langage de programmation utilisé. La bibliothèque libModbus est écrite en **C** tandis que les deux autres (JModbus et Modbus4J) sont écrites en **Java**.

2.2.1. La bibliothèque libModbus

Cette bibliothèque est encore maintenue à ce jour. La dernière version stable remonte au 26/05/2012 avec la version 3.0.3.

Documentation	<p>La documentation est assez complète sur le site officiel avec description de toutes les fonctions et exemples sur quasiment chacune d'entre elle.</p> <p>Il existe peu de forums sur lesquels compter en cas de problèmes mais il y a la possibilité d'écrire au créateur de la bibliothèque en cas de bugs problèmes rencontrés.</p>
Coût	Open-source.
Productivité	<p>L'installation est facile, il suffit d'inclure libModbus au compilateur GCC pour que ses fonctions soient prises en compte.</p> <p>On retrouve un ensemble de huit fonctions permettant de faire l'ensemble des opérations de lecture et d'écriture sur les bits et registres de la mémoire de l'automate et rend son utilisation assez intuitive.</p> <p>Le principal avantage est l'existence de deux fonctions :</p> <ul style="list-style-type: none">✓ modbus_get_float, permet à partir d'un tableau contenant 2 entités de 16 bits d'obtenir un réel (32 bits). Cette fonction est pratique pour la lecture de réels sur la mémoire de l'automate qui sont codés sur 2 registres de 16 bits.✓ modbus_set_float, est l'inverse de la fonction précédente, c'est-à-dire qu'elle permet d'obtenir 2 entités de 16 bits à partir d'un nombre réel. Elle facilite l'écriture de réels sur la mémoire de l'automate.

Temps d'exécution des requêtes	On est sur du temps réel que l'on soit en local ou à distance.
---------------------------------------	--

2.2.2. La bibliothèque JModbus

La dernière version de cette API est la version 0.1.1 datant du 8/11/2001. Malheureusement, elle n'est plus maintenue à ce jour.

La communication entre le serveur web applicatif et l'automate se fait par **sockets**. C'est un mécanisme permettant la communication entre deux hôtes à l'aide d'un couple d'adresse IP et de port écrit de la manière suivante : (@IP1,Port1;@IP2,Port2). La première partie du couple concerne le serveur web applicatif et la deuxième partie l'automate. Le port est le 502 correspondant au Modbus TCP/IP.

Documentation	La Javadoc de cette API est très complète avec une description de toutes les classes et méthodes. Peu de forums et d'informations sont présents en cas de problèmes.
Coût	Open-source.
Productivité	Il n'y a pas d'installation, juste à mettre l'archive jmodbus.jar dans le classpath lors de la compilation. La mise en place est assez facile mais l'utilisation est beaucoup moins évidente due au manque de documentation. Il existe huit méthodes pour faire l'ensemble des opérations de lecture et d'écriture sur les bits et les registres de la mémoire de l'automate.
Temps d'exécution des requêtes	Que l'on soit en local ou à distance, l'exécution des actions en lecture et écriture se fait en temps réel.

2.2.3. La bibliothèque Modbus4j (Modbus For Java)

Cette bibliothèque a été créée en 2007, mais est seulement disponible depuis 2008. Elle est encore maintenue à ce jour.

Son fonctionnement ressemble beaucoup à celui de JModbus, décrit précédemment avec l'utilisation de sockets pour la communication avec l'automate.

Documentation	La javadoc est incomplète sur le site officiel. On retrouve bien la liste de toutes les classes et des méthodes, cependant aucune explication n'est fournie sur l'utilité de chacune d'entre elle. On retrouve peu de forums ou d'informations en cas de problèmes.
Coût	Open-source
Productivité	Il n'y a pas d'installation à faire, juste à mettre les archives modbus4J.jar et seroUtils.jar dans le classpath lors de la compilation. Il existe huit fonctions pour faire l'ensemble des opérations de lecture et d'écriture. Son principe de mise en place est plus intuitif que JModbus, ce qui la rend plus agréable à utiliser.
Temps d'exécution des requêtes	Que l'on soit en local ou à distance, l'exécution des actions en lecture et écriture se fait en temps réel.

2.3. Le service WebDesigner

De manière plus spécifique, il est possible d'utiliser le service WebDesigner. C'est un outil mis à disposition sur les serveurs web des automates de la marque Schneider-Electric consistant à visualiser l'IHM de commande comme si on était en local devant le système. Lors des tests, nous avons constatés que les temps de réponses ne sont pas satisfaisants et ne correspondent pas au besoin temps réel. Cette option ne peut pas être prise en compte.

2.4. Choix du type de communication utilisé

On obtient des résultats égaux dans le temps d'exécution des actions de lecture et d'écriture, que l'on utilise la communication Modbus TCP/IP ou la communication TCP/IP via le serveur web monté dans l'automate. Dans ces deux cas, la communication se fait en temps réel.

Les choix techniques en terme de communication se décideront en fonction du langage de programmation utilisé pour le développement de la plateforme d'apprentissage. Cependant, l'utilisation du service web ModbusXMLDa à l'avantage de ne dépendre d'aucun langage, il peut être attaqué par n'importe lequel.

V. Réalisation et Expérimentation

L'expérimentation de la solution proposée afin d'intégrer des systèmes didactiques dans une plateforme d'apprentissage a été réalisée au travers de la plateforme DIWEB, souhaitée par la société DMS. Les développements de DIWEB n'étant pas commencés à ce jour, il a fallu dans un premier temps créer les principales fonctionnalités de cette plateforme pour pouvoir intégrer la partie concernant les téléTP. Le système didactique intégré dans DIWEB est le panneau photovoltaïque que l'on a décrit dans la partie 2 du chapitre 3 sur le cadre d'étude.

Dans cette partie nous verrons les principales étapes qui nous ont permis d'expérimenter notre proposition et nous discuterons des résultats trouvés ainsi que des évolutions envisageables. Toute cette phase d'expérimentation s'est déroulée dans les locaux de DMS.

1. Installation et mise en place du réseau au sein de la société

DMS

La mise en place du réseau informatique constitue une étape importante pour l'établissement qui accueillera la plateforme DIWEB. Pour cela j'ai installé les différents composants et organisé le réseau informatique de la société DMS comme si nous nous trouvions dans un établissement.

Notre architecture technique comprend les composants suivants :

- ✓ Le **serveur web applicatif** dédié au projet, équipé du système d'exploitation DEBIAN qui est une distribution de Linux. J'ai installé le serveur web Apache2 pour qu'il soit capable d'interpréter les requêtes HTTP des clients web (navigateur) arrivant sur le port 80 associé au protocole HTTP et de fournir une réponse avec ce même protocole. La base de données MySQL est également installée.
- ✓ **2 postes informatiques** faisant office de poste apprenant. Pour accéder à la

plateforme, des navigateurs (Google Chrome et Mozilla Firefox) ont été installés.

- ✓ **Le système panneau photovoltaïque.** Il est le seul élément de notre laboratoire distant.
- ✓ Une **caméra IP** pivotable de la marque VIVOTEK. C'est le dispositif environnemental du système panneau photovoltaïque capable de le filmer et de l'écouter en cours de fonctionnement puisqu'elle elle est également équipée d'un micro. Cette caméra embarque un serveur web ce qui permet d'accéder aux images et de la configurer directement depuis un navigateur web.
- ✓ Un **routeur**, permettant de faire le lien entre le réseau local de DMS et le réseau extérieur (Internet). Il s'agit d'un routeur de la marque Netgear.

Le schéma ci-dessous illustre l'organisation du réseau local mise en place à DMS avec les adresses IP associées à chaque élément.

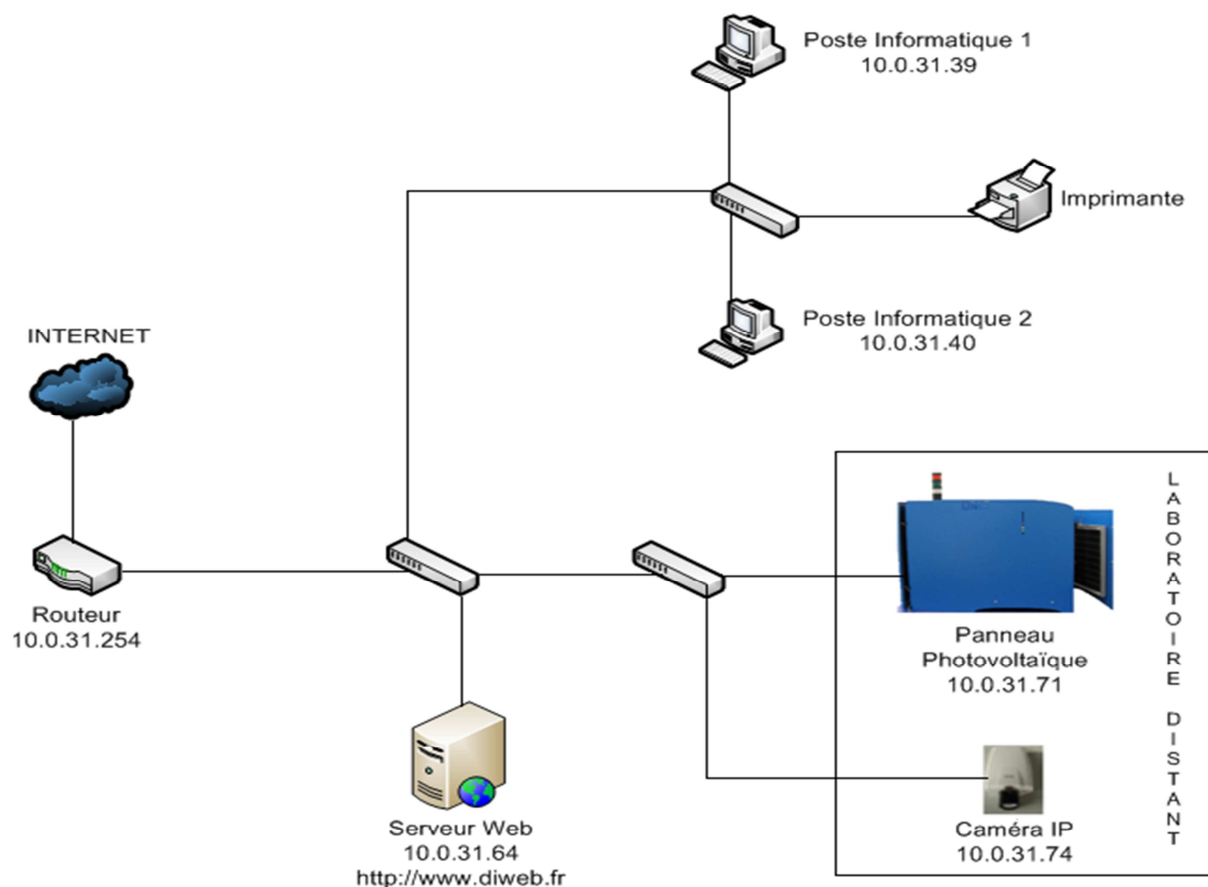


Figure 26: Schéma du réseau informatique local de la société DMS

Voici un tableau récapitulatif des adresses IP du réseau local DMS

Composant	Adresse IP locale	Nom de domaine
Routeur	10.0.31.254	-
Serveur Web	10.0.31.64	http://www.diweb.fr
Système Panneau Photovoltaïque	10.0.31.71	-
Caméra IP	10.0.31.74	-
Poste Informatique 1	10.0.31.39	-
Poste Informatique 2	10.0.31.40	-

1.1. Accès à distance au système panneau photovoltaïque

C'est le serveur web qui fait la passerelle vers le système panneau photovoltaïque. Il faut donc que ce serveur soit accessible depuis l'extérieur du réseau local de DMS. Cependant, l'adresse IP du serveur est attribuée par le fournisseur d'accès à Internet. Cette adresse IP est généralement dynamique et ne dure qu'un certain temps avant d'être de nouveau modifiée. Dans ces conditions il est impossible de faire correspondre une adresse IP avec un nom de domaine. C'est dans ce but que j'ai installé sur ce serveur web, l'outil **No-IP** qui permet en fait d'obtenir une adresse IP fixe en effectuant une résolution de nom de domaine vers l'adresse IP dynamique actuelle. Il sert donc à rendre transparent la modification d'adresse IP dynamique du serveur web pour la faire correspondre au nom de domaine choisi. En l'occurrence, le nom de domaine qui a été réservé pour DIWEB est <http://www.diweb.fr>. C'est à travers cette URL que DIWEB sera accessible.

Les deux principaux outils permettant de rendre des IP fixes sont DynDNS et No-IP. Je suis parti sur le choix de No-IP car pour les mêmes prestations il a l'avantage d'être gratuit. Le seul inconvénient est qu'il faut réactiver le compte tous les mois.

1.2. Configuration du routeur

Les ports **80** (protocole HTTP), **502** (protocole Modbus TCP/IP) et **5004** (pour la caméra du système panneau photovoltaïque) ont été ouverts sur le routeur en TCP/IP pour le client serveur web. Le réseau informatique de DMS est géré par une société tierce, c'est par cette société que je suis passé pour l'ouverture des ports. C'est grâce à cette manipulation que le serveur web applicatif pourra être atteint depuis l'extérieur uniquement sur les ports 80, 502 et 5004.

1.3. Port forwarding et Iptables

Le **port forwarding** (redirection de port en français) permet de rediriger un port vers une machine spécifique du réseau local. Dans notre cas, l'objectif est de rediriger les ports 502 et 5004 vers les dispositifs correspondants. Lorsque la communication se fait via le protocole Modbus TCP/IP, les paquets arrivant sur le port 502 du serveur web seront redirigés vers l'automate du système panneau photovoltaïque tandis que ceux arrivant sur le port 5004 pour une communication avec la caméra IP seront quant à eux redirigés vers la caméra IP installée pour filmer les panneaux photovoltaïques.

Pour cela, j'ai utilisé l'outil **Iptables**. Cet outil est une interface en ligne de commande, présent sur toutes les distributions Linux dont Debian, et permettant de configurer le module **Netfilter**, module du noyau Linux offrant la possibilité de contrôler, modifier et filtrer les paquets IP ainsi que de suivre les connexions.

J'ai mis en place le fichier script shell `BootCommandes.sh` qui définira les configurations réseau et les règles de routage mises en place sur le serveur web à chaque démarrage de celui-ci. Nous allons détailler les principales commandes contenues dans ce fichier :

- ✓ **Activation du port forwarding** : il est indispensable que le port forwarding soit activé sur le serveur. S'il n'est pas activé, nos règles de routage ne servent à rien. Pour l'activer, il suffit de taper la commande suivante, `echo 1 > /proc/sys/net/ipv4/ip_forward`
- ✓ **Configuration de l'interface réseau de la machine** : notre serveur dispose d'une

interface Ethernet, eth1 correspondant au réseau local. On écrit la commande `ifconfig eth1 10.0.31.64 netmask 255.255.255.0` pour assigner l'adresse IP locale et le masque de sous réseau à l'interface réseau eth1.

- ✓ **Modification de la table de routage** : on ajoute la route par défaut dans la table de routage vers la passerelle par défaut par la commande `route add default gw 10.0.31.254`. Ceci permet de définir pour le serveur web l'hôte du réseau local vers lequel envoyer tous les paquets destinés à des hôtes situés hors du réseau local.
- ✓ **Redirection des paquets IP reçus sur le port 502** : la commande `iptables -t NAT -A PREROUTING -j DNAT -i eth1 -p tcp -dport 502 -to-destination 10.0.31.71:502` permet de rediriger les paquets IP arrivant sur le port 502 de l'interface réseau Ethernet eth1 du serveur web sur le port 502 de l'automate du système panneau photovoltaïque.
- ✓ **Redirection des paquets IP reçus sur le port 5004** : la commande `iptables -t NAT -A PREROUTING -j DNAT -i eth1 -p tcp -dport 5004 -to-destination 10.0.31.74:80` permet quant à elle de rediriger les paquets IP arrivant sur le port 5004 de l'interface réseau Ethernet eth1 du serveur web sur le port 80 du serveur web de la caméra IP.

2. Développement de la plateforme DIWEB

Comme la plateforme DIWEB n'existait pas, pour pouvoir tester notre solution, il a bien fallu développer cette plateforme. Avant de commencer le développement, une question s'est posée. N'est-il pas possible d'utiliser une plateforme d'apprentissage existante en la configurant comme DMS le souhaite. En effet, il n'existe pas loin de 200 plateformes d'apprentissage en ligne (**Web CT** étant la plus connue) dont une trentaine est sous licence GPL (General Public License), ce qui signifie que le code source est tout à fait modifiable et adaptable par n'importe qui souhaitant utiliser le code source. En effet, dans les plateformes d'apprentissage existantes on retrouve les principales fonctionnalités comme la gestion des ressources formatives ou les outils de communication tels que les forums, chats, mails, ...

Mais cette licence est dite contaminante et indique également qu'un logiciel utilisant le code

source d'un logiciel sous licence GPL doit être diffusé avec ses sources et peut être librement adapté et modifié. Il tombe lui aussi automatiquement sous le coup de la licence GPL. Les principales plateformes d'apprentissage sous ce type de licence étant **Moodle**, **Claroline** ou **Chamilo**. On aurait pu dans ce cas adapter le code source et rajouter un module de téléTP par la suite.

Cependant DMS souhaite que le code source de DIWEB soit protégé, pour pouvoir vendre cette plateforme dans le futur, c'est pourquoi l'adaptation de ces plateformes ne sera pas possible.

2.1. Choix du langage de développement

DIWEB est une plateforme d'apprentissage en ligne, bâtie sur une architecture **client-serveur**, ce qui signifie que le serveur est chargé de répondre aux requêtes envoyées par les clients.

Pour développer DIWEB, de nombreuses solutions permettant de réaliser du client-serveur étaient possibles. Parmi celles-ci, on retrouve le langage **PHP** (Personal Home Page). Ce langage est très simple à mettre en œuvre. Il possède les caractéristiques suivantes :

- ✓ Il est souple (les variables sont faiblement typées) et permissif, il propose un confort d'utilisation inégalé alors que d'autres langages nécessitent beaucoup de rigueur et de connaissances théoriques.
- ✓ Il est facilement extensible et accepte de multiples fonctionnalités à travers son système d'extensions ce qui augmente le nombre de classes et fonctions natives disponibles.
- ✓ Il est optimisé pour économiser des ressources ce qui a pour but d'augmenter ses performances, les temps d'exécution étant réduit.

De plus, il a accumulé au fil des ans suffisamment de composants spécialisés pour le Web qu'il est capable de gérer tout type d'applications web.

Le framework **Java** reste également une solution viable. En effet, au travers de la brique JEE

(Java Enterprise Edition) destinée au développement d'applications client-serveur, on retrouve notamment les spécifications **Servlet** et **JSP** présentes pour répondre à la problématique des contenus dynamiques des applications web. Les principaux avantages de Java sont :

- ✓ La **portabilité du code**, chaque application pourra être déployée sur tous les dispositifs possédant une machine virtuelle Java (ou JVM pour Java Virtual Machine en anglais).
- ✓ La **gratuité** : le compilateur, l'accès à toutes les API de Java, la documentation et toutes les spécifications sont librement disponibles sur Internet via le site Web d'Oracle.
- ✓ La **sécurité** : la JVM préserve la sécurité et l'intégrité du système sur lequel tourne l'application.

L'**ASP** (Active Server Pages) de Microsoft possède aussi toutes les fonctionnalités nécessaires au développement web mais à la différence de Java et PHP, il est résolument propriétaire, utilisable uniquement dans un environnement de serveurs Windows. Cette solution ne sera donc pas envisagée.

J'ai donc longuement hésité entre PHP et Java mais j'ai finalement opté pour le PHP qui correspond aux besoins de DIWEB. En effet, malgré les apparences c'est un langage professionnel qui a fortement évolué au cours des 10 dernières années et offre la possibilité de développer du code haut niveau notamment quand on l'associe à la puissance du modèle objet. De plus la souplesse de ce langage permet de développer plus rapidement qu'en Java.

2.2. Le Design Pattern Modèle Vue Contrôleur

Le Design Pattern **Modèle Vue Contrôleur** (MVC) permet de structurer le code d'une application en 3 couches :

- ✓ La **Vue** représente tout ce qui touche à l'interaction avec les utilisateurs et la présentation des données (mise en forme, affichage). Elle n'effectue aucun traitement des données.

- ✓ Le **Modèle** concerne le traitement, le stockage et la mise à jour des données. C'est dans cette couche que l'on définit l'interaction avec la base de données.
- ✓ Le **Contrôleur** gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue par la suite la synchronisation entre le modèle et les vues. Le contrôleur représente le point central du Design Pattern MVC.

Cette organisation présente l'avantage de séparer les IHM et les données d'une application. Ceci permet une maintenance plus facile de l'application par la suite. Elle facilite également le travail d'équipe.

Le diagramme d'activités ci-dessous représente les différentes actions réalisées par le contrôleur, le modèle ou la vue.

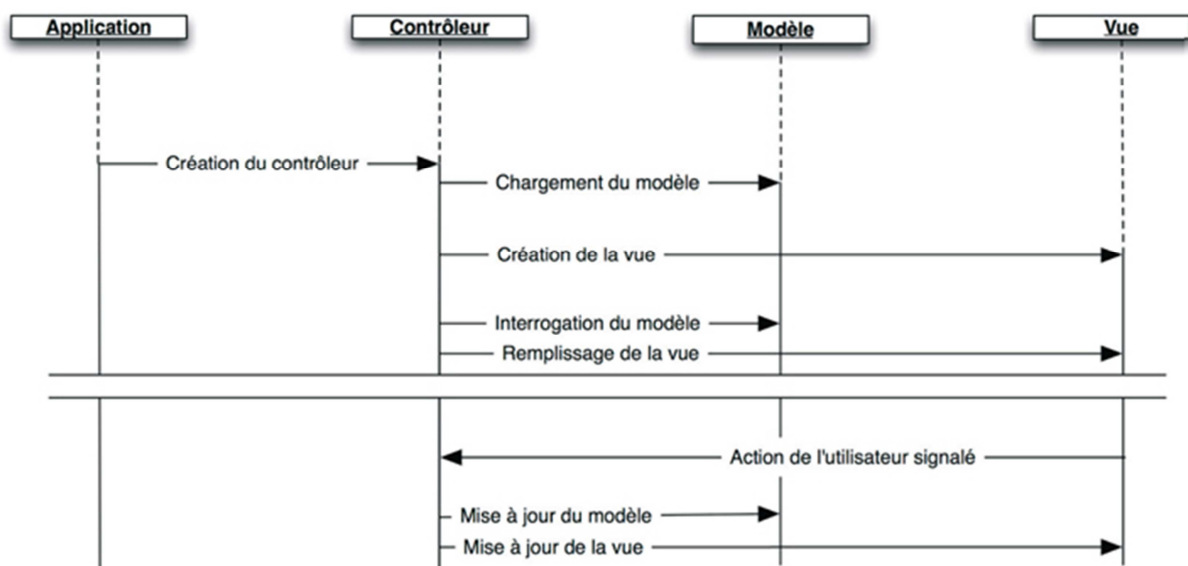


Figure 27: Diagramme d'activités du Design Pattern MVC

C'est sur cette architecture que DIWEB a été conçue. En PHP, de nombreux framework existent, la plupart open source, et permettent de réaliser des applications de type MVC. Le plus connu étant Zend Framework, mais on peut citer en citer d'autres comme Symfony ou CakePHP. Cependant ce type de framework n'est pas réputé pour leur performance brute, de nombreuses sous couches sont appelées ce qui rend les traitements plus lourds. Même si le développement est plus rapide en utilisant un framework comme Zend, C'est pour cela

que je n'ai pas choisi de m'appuyer sur l'un de ces framework mais de créer mon propre système MVC.

3. Développement de la solution proposée

Une fois la structure de DIWEB développée, la mise en place de la solution proposée est possible.

3.1. Organisation du développement

Avant de se lancer dans le développement de l'intégration de systèmes automatisés didactiques dans DIWEB, l'**organisation** de celui-ci reste indispensable. Même si elle n'a pas constitué une phase primordiale étant la seule ressource sur le projet, elle m'a tout de même permis de définir toutes les tâches de développement mais également l'ensemble des activités pour l'intégration du projet à réaliser permettant de le dégrossir et donc le rendre moins complexe.

La décomposition a été faite par la méthode **WBS** (Work Breakdown Structure) qui m'a aidé à recenser l'ensemble des tâches de développement du projet sous forme arborescente.

La construction de cet arbre se fait en effectuant :

- ✓ L'inventaire exhaustif de toutes les tâches à réaliser.
- ✓ L'identification des lots de travaux.
- ✓ La détermination du niveau hiérarchique de tous ces lots.

La structure initiale du WBS concerne l'ensemble des tâches de façon globale. L'entrée de l'arbre se situe au lot **0000** qui se trouve être le développement final, c'est le lot global. Celui-ci est découpé en trois lots d'activités traitant des fonctionnalités de l'apprenant, du formateur et de l'administrateur. Ces lots contiennent eux-mêmes d'autres lots. On obtient

la structure initiale suivante :

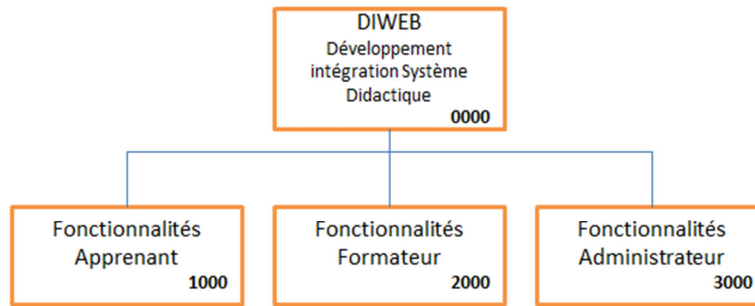


Figure 28: Structure initiale WBS du projet

Lot **1000** : ce sont les fonctionnalités de l'apprenant. On retrouve la visualisation du laboratoire distant, le pilotage d'un système didactique, l'observation de la manipulation sur un système, l'enregistrement des résultats d'une manipulation et l'exploitation des données de manipulation. Ce lot est le plus important, c'est donc celui-ci qui a été le plus abouti en terme d'avancement.

Lot **2000** : ce sont les fonctionnalités du formateur. Ce lot d'activité comprend la gestion des apprenants, ainsi que la gestion des droits d'accès sur les systèmes. Par manque de temps ce lot n'a pas été développé.

Lot **3000** : ce sont les fonctionnalités de l'administrateur de DIWEB. Ce lot comprend les fonctionnalités de gestion du laboratoire distant de l'établissement ainsi que la gestion des systèmes didactiques contenu dans le laboratoire distant. Par manque de temps ce lot n'a pas été développé.

Ces trois lots contiennent des tâches détaillées faisant référence aux spécifications techniques. On retrouve également les tests unitaires qui eux assurent une certaine qualité et rigueur au niveau du développement.

Lot 1000 : Les Fonctionnalités Apprenant

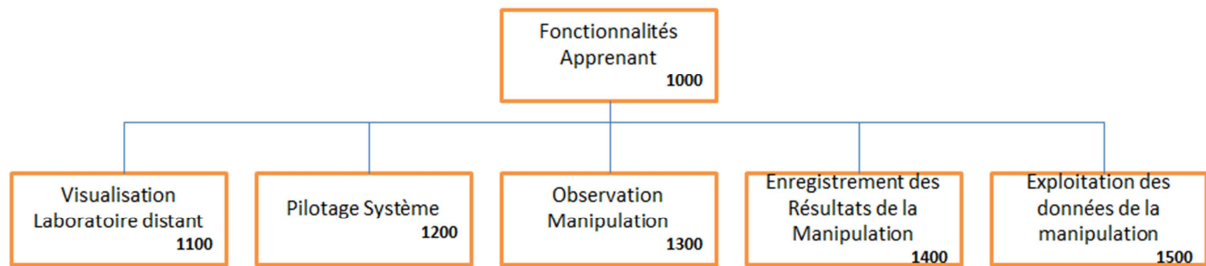


Figure 29: Structure du Lot 1000 "Fonctionnalités Apprenant"

Lot 1100 : correspond à la visualisation du laboratoire distant. On retrouve l’affichage de la liste des systèmes du laboratoire distant de l’établissement (lot 1110) ainsi que l’analyse de chaque système permettant de savoir leur disponibilité ou non en cas de panne, d’utilisation ou autres. (lot 1120).

Lot 1200 : on retrouve dans la gestion de la connexion (lot 1210), la possibilité pour un apprenant de se connecter ou déconnecter d’un système mais également toutes les fonctionnalités permettant de lire ou modifier l’état des actionneurs du système (actions de simulation ou d’observation) correspondant au lot 1220. Ceci se fait au travers de l’IHM de commande.

Lot 1300 : lecture des flux audio et vidéo issu des dispositifs environnementaux installés sur chaque système (lot 1310 et 1320). L’observation de l’état des capteurs du système manipulé peut se faire sous forme graphique ou numérique (lot 1330).

Lot 1400 : on retrouve toutes les fonctionnalités permettant d’enregistrer dans la session de manipulation ouverte par l’apprenant des données du système sur une durée choisie ou à un instant t (lot 1410).

Lot 1500 : dans ce lot, il s’agit de l’exploitation des résultats obtenus au travers des différentes sessions de manipulation.

Lot 2000 : Fonctionnalités Formateur

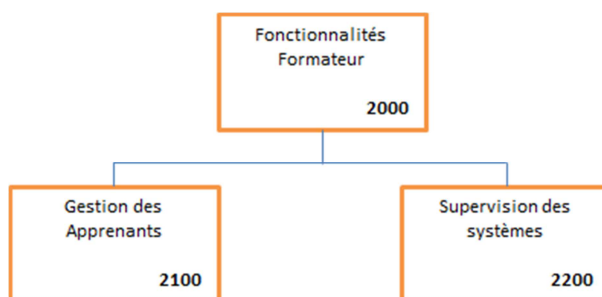


Figure 30: Structure du Lot 2000 "Fonctionnalités Formateur"

Lot 2100 : fonctionnalités de gestion des groupes de TP dans la plateforme d'apprentissage.

Lot 2200 : module de supervision des systèmes permettant de gérer l'accès à l'ensemble des systèmes du laboratoire distant.

Lot 3000 : Fonctionnalités Administrateur

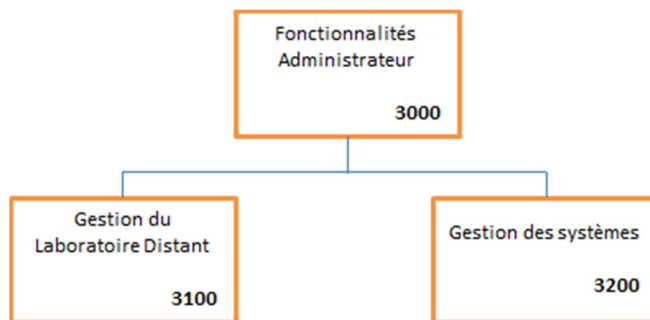


Figure 31: Structure du lot 3000 "Fonctionnalités Administrateur"

Lot 3100 : on retrouve la gestion du laboratoire distant de l'établissement pour tout ce qui est ajout, modification ou suppression de systèmes mais également la connexion ou la déconnexion de chaque système pour le rendre utilisable ou pas (lot 3110). Il y a aussi le module de connexion/déconnexion des dispositifs environnementaux (lot 3120).

Lot 3200 : module de gestion des systèmes présents dans le laboratoire distant que ce soit au niveau des dispositifs environnementaux où des informations sur chacun d'eux.

3.2. Développement

Je me suis particulièrement attaché à développer les lots **1200** et **1300** correspondant au pilotage du panneau photovoltaïque et à l'observation de la manipulation. Pour pouvoir faire ces deux lots il a fallu dans un premier temps réaliser les IHM de commande du panneau photovoltaïque.

3.2.1. Développement des IHM de commande du panneau photovoltaïque

Cette phase consiste à adapter l'IHM de commande présent sur le panneau photovoltaïque au format Web. Cette IHM de commande a été initialement développée à l'aide du logiciel Vijeo Designer, outils propriétaire de Schneider-Electric, il m'est donc impossible d'utiliser ce format d'IHM pour les convertir au format Web. Pour cela j'ai fait de l'intégration web à partir des maquettes de l'IHM (transmises au format d'image JPEG) en utilisant les technologies suivantes :

- ✓ **HTML 5** : permet de créer le squelette de l'IHM à l'aide de balises en la structurant sémantiquement. La version 5 de HTML est la dernière version de ce langage. Cette version à l'avantage d'être compatible avec l'ensemble des navigateurs web présents sur le marché même si toutes les fonctionnalités ne sont pas intégrées sur tous.
- ✓ **CSS 3** : pour réaliser la mise en page des contenus définis avec les balises HTML. Cette mise en forme est faite au moyen de styles.
- ✓ **JavaScript** : cette technologie est utilisée pour ajouter de l'interactivité sur l'IHM de commande en modifiant le comportement de certains éléments.

L'objectif de l'intégration est donc de mettre en place l'IHM de commande au format Web de façon à pouvoir l'intégrer dans DIWEB. Cette IHM est un ensemble de pages accessibles les unes aux autres par des liens hypertextes. Ci-dessous le résultat obtenu pour la page

d'accueil (en haut à gauche), le mode manuel (en bas à gauche) et le mode régulation (à droite).

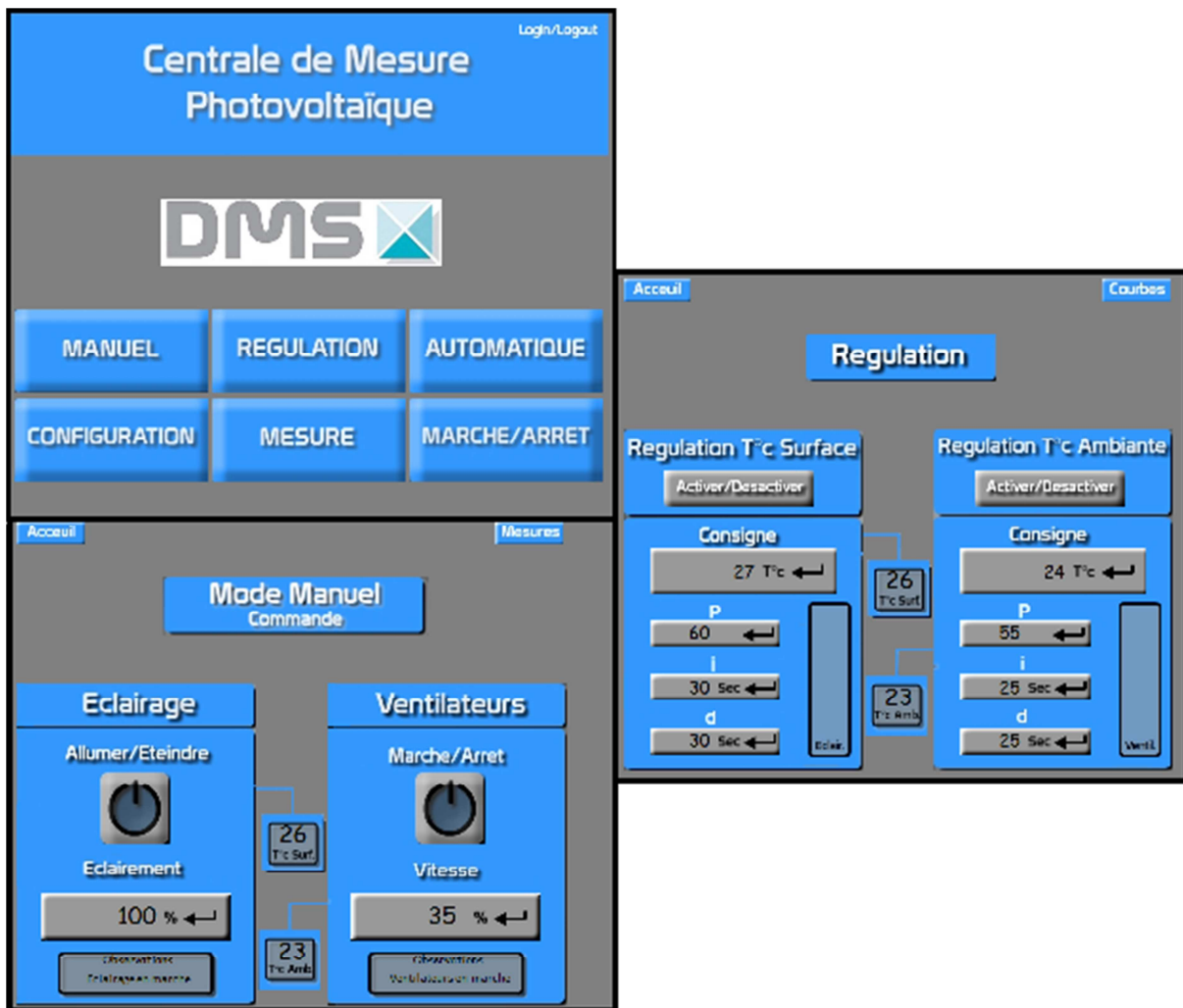


Figure 32: IHM de commande au format Web

3.2.2. Le pilotage du système panneau photovoltaïque

Les actions de **simulation** correspondent au pilotage du système tandis que les actions d'**observation** constituent l'observation de la manipulation. Dans le cas du panneau photovoltaïque, le tableau suivant liste les actions de simulation qui ont été développées pour un apprenant. Cette liste ne comporte pas toutes les actions de simulation réalisables sur le système.

Nom Action	Description
Consigne-Eclairment	Modifie l'intensité lumineuse sur les panneaux solaires.
Consigne-Vitesse-Ventilateur	Modifie la vitesse des ventilateurs présents dans la cage du panneau photovoltaïque.
Gestion-Eclairage	Allume ou éteint l'éclairage du panneau photovoltaïque.
Gestion-Ventilateurs	Allume ou éteint les ventilateurs du panneau photovoltaïque.
Consigne-Température-Ambiante	Modifie la consigne de température ambiante.
Consigne-Température-Surface	Modifie la consigne de température à la surface des panneaux solaires.

Une action de simulation va écrire dans les variables du programme de l'automate dans le but de modifier son comportement.

Je vais détailler le développement de l'action de simulation **Gestion-Eclairage** que l'on peut déclencher dans le mode manuel de l'IHM de commande. La première étape du développement de cette action consiste à la lecture du GRAFCET de l'automate pour visualiser toutes les étapes liées à la gestion de l'éclairage. Cette gestion est traitée par l'automate en trois étapes :

- ✓ Lecture de la variable **%M16** (IHM_BP_Commande_Lampe), c'est un booléen correspondant à l'état des ampoules. En effet les ampoules peuvent être dans deux états différents, en marche ou à l'arrêt. Si le booléen est positionné à marche, les ampoules sont allumées avec l'intensité définie par la variable **%MW14** sinon les ampoules sont éteintes. Les deux étapes suivantes se déroulent **uniquement** si l'état des ampoules est à marche.
- ✓ Lecture de la variable **%MW14** (IHM_Consigne_Eclairment) correspondant à la valeur de la consigne d'éclairage rentrée par l'apprenant sur l'IHM. Cette variable est de type réel, elle est codée sur 32 bits.
- ✓ Ecriture de la variable **%MW18** (Consigne_Eclairment) avec la valeur de la variable **MW%14** lue précédemment. Elle est également du type réel donc codée sur 32 bits. C'est cette variable qui sera prise en compte par l'automate pour donner la consigne d'éclairage au gradateur. Cette valeur est donc transmise au **gradateur**, dispositif électronique permettant de faire varier la valeur d'une tension alternative, qui va changer l'intensité lumineuse des ampoules présentes dans le panneau

photovoltaïque. En fonction du pourcentage saisi par l'apprenant sur l'IHM de commande l'intensité lumineuse sur les panneaux solaires sera plus ou moins importante.

Le **diagramme de séquence** de la méthode UML permet d'avoir une meilleure visibilité dans l'enchaînement des étapes entre les objets dans le développement de l'action Gestion-Eclairage. En effet ce diagramme va représenter graphiquement le déroulement de l'action et montrer tous les objets impactés. On considère que la connexion sur l'automate du panneau photovoltaïque est effective.

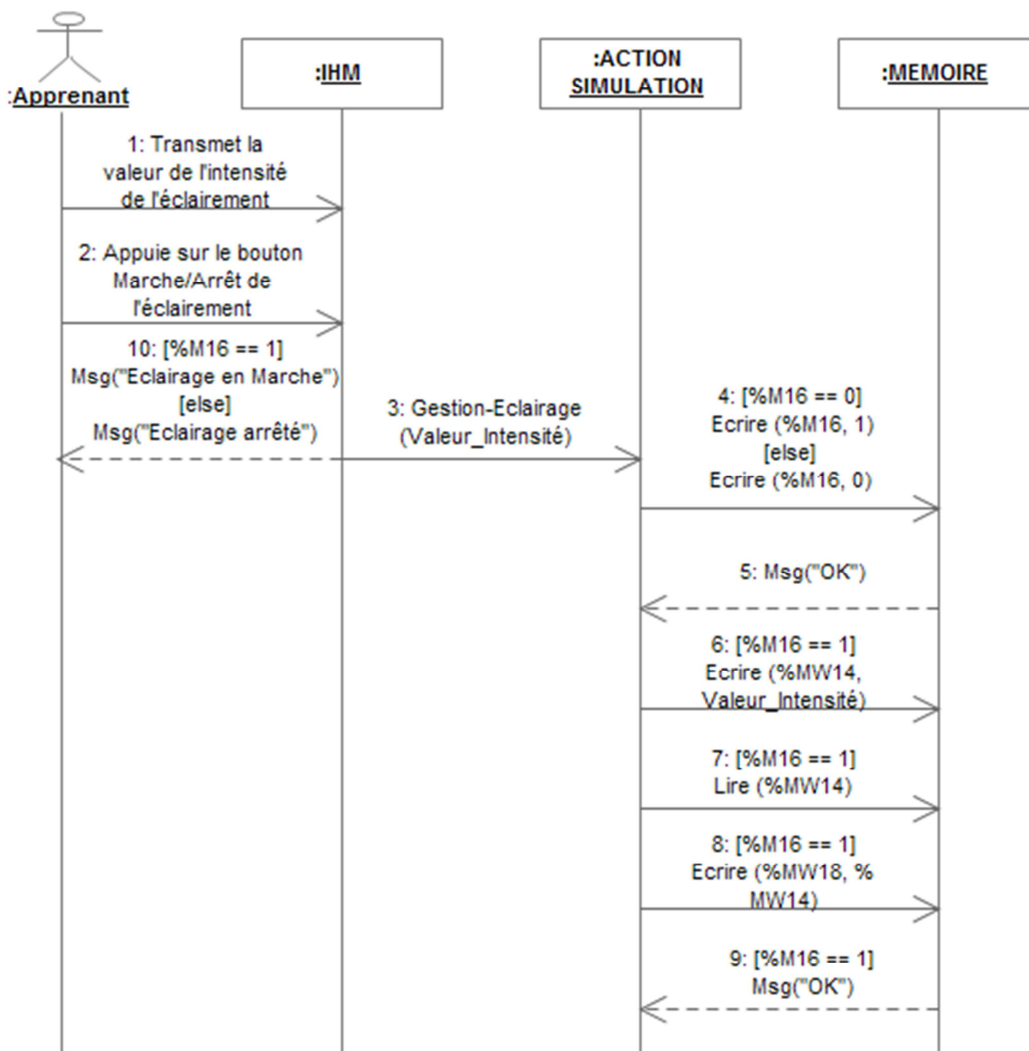


Figure 33: Diagramme de séquence de la méthode "Gestion-Eclairage"

L'action **Gestion-Eclairage** est déclenchée par l'apprenant en cliquant sur le bouton Marche/Arrêt de la page Mode Manuel de l'IHM de commande. Auparavant, l'apprenant aura saisi la valeur de l'intensité de l'éclairage souhaitée. Dans un premier temps, cette action teste si l'éclairage va passer en marche ou en arrêt. Ce test va conditionner la suite des opérations. En effet si l'éclairage est en marche (si la variable %M16 est égal à 1), on écrit la valeur de l'intensité choisit par l'apprenant dans la variable %M14 (étape 6 sur le diagramme de séquence) puis on lit la variable %MW14 pour voir quelle est la valeur de l'intensité rentrée par l'apprenant dans l'IHM de commande (étape 7). C'est cette valeur que l'on va écrire dans la variable %MW18 à l'aide de la méthode Ecrire de l'objet Mémoire (étape 8). Pour finir on affiche à l'écran le message « Eclairage en marche » pour indiquer à l'apprenant que l'action s'est correctement déroulée. Si l'éclairage passe en mode arrêt on affiche à l'écran le message « Eclairage arrêté ».

Ayant choisi d'utiliser le langage PHP pour développer la solution, je me suis servi du service web **ModbusXMLDa** pour communiquer avec l'automate. En faisant de la lecture et de l'écriture sur la mémoire de l'automate du panneau photovoltaïque, les méthodes **Lire** et **Ecrire** de l'objet Mémoire font appel aux méthodes de ce service web. Etant codé sur 32 bits, la variable %MW14 est lue avec la méthode **ReadMutipleRegisters** de ModbusXMLDa. Pour rappel 1 registre vaut 16 bits, pour lire une variable codée sur 32 bits, la méthode ReadMutipleRegisters permet de lire ces 2 registres. La méthode **WriteMutipleRegisters** est utilisée sur le même principe pour écrire sur les 2 registres constituant la variable %MW18.

Pour faciliter l'utilisation des variables du programme de l'automate, j'ai mis en place un **fichier de configuration** permettant de stocker l'ensemble des adresses mémoires de ces variables ainsi que les noms et le type associés à chacune d'entre elle. Ce fichier recensera toutes les variables de l'automate accessibles à distance et utilisées dans les différentes actions. En effet toutes les variables ne sont pas accessibles à distance depuis un programme externe. Seules les variables internes, les constantes et les variables systèmes sont accessibles à distance. Les variables entrées-sorties physiques ne sont pas disponibles à distance que ce soit en lecture ou en écriture, il faut que l'automaticien est prévue de recopier les valeurs de ces variables dans des variables internes au programme pour avoir

accès en lecture et écriture à distance.

Ce fichier est structuré en deux parties, la première rassemble toutes les variables susceptibles d'être accédées en lecture et la deuxième toutes les variables susceptibles d'être accédées en écriture. Il permettra une meilleure maintenance en cas de changement dans les variables du programme de l'automate.

3.2.3. Observation de la manipulation

L'observation de la manipulation en cours est possible à l'aide des actions d'observation. Le tableau suivant liste les actions d'observation qui ont été développées pour un apprenant. Cette liste ne comporte pas toutes les actions d'observation réalisables sur le système.

Nom Action	Description
Etat-Eclairment	Récupère le pourcentage d'éclairment mis en place.
Etat-Vitesse-Ventilateur	Récupère le pourcentage de la vitesse des ventilateurs mis en place.
Etat-Température-Ambiante	Récupère la température relevée à l'intérieur de la cage du panneau photovoltaïque.
Etat-Température-Surface	Récupère la température à la surface des panneaux solaires.
Mesures-Panneaux-Solaires	Récupère un ensemble d'informations sur les panneaux solaires comme la tension, l'intensité, la puissance électrique de l'énergie produite.

Ce type d'action a pour vocation de lire uniquement les variables du programme de l'automate avec pour objectif de récupérer une valeur et de l'afficher. Cependant certaines données comme l'évolution de la température ambiante ou à la surface des panneaux solaires nécessitent d'être représentées graphiquement. Pour cela l'utilisation d'une bibliothèque graphique a été obligatoire. En effet l'évolution de la température a été représentée au moyen d'une courbe graphique avec des informations disponibles sur chaque point de la courbe.

Afin de choisir la meilleure bibliothèque graphique possible, j'ai réalisé un comparatif des principales bibliothèques sur le marché (développé dans un langage serveur ou client). Les critères de sélection ont été le design, l'esthétique des graphes proposés ainsi que la facilité

d'utilisation. En fonction des tests réalisés, chaque critère de chaque bibliothèque a été noté sur 10. La bibliothèque choisie devra également être gratuite à des fins commerciales.

Bibliothèque	Version	Langage	Prix/Licence	Maintenance	Compatibilité	Graphes proposés	Assistance	Design (sur 10)	Simplicité, Facilité d'utilisation (sur 10)
JpGraph	3.0.7 (11/01/2011)	PHP	Gratuit pour une utilisation personnelle, mais payante pour une utilisation commerciale (105€).	NON	Tout navigateur (Firefox, IE, Google Chrome, Safari, Opera)	Camembert, courbes, aire, colonnes, ...	Email, forum, tutoriaux, FAQ, sites	2	8, très simple et orienté-objet.
pChart	2.1.2 (4/08/2011)	PHP	Gratuit	OUI	Tout navigateur (Firefox, IE, Google Chrome, Safari, Opera)	Camembert, courbes, aire, colonnes, ...	Email, forum, tutoriaux, FAQ, sites	7	6, simple ; orienté-objet, difficulté pour afficher le graphe
HighChart	2.2.4 (31/05/2012)	JavaScript	Gratuit pour une utilisation personnelle, mais payante pour une utilisation commerciale (360€).	OUI	Tout navigateur (Firefox, IE, Google Chrome, Safari, Opera)	Camembert, courbes, aire, colonnes, ...	Email, forum, tutoriaux, FAQ, sites	9	9, très simple, écriture au format JSON.
JQPlot	<u>1.0.0b2_r1012</u> (12/11/2011)	JavaScript	Gratuit	OUI	Tout navigateur (Firefox, IE, Google Chrome, Safari, Opera)	Camembert, courbes, aire, colonnes, ...	Email, forum, tutoriaux, FAQ, sites	8	9, très simple, écriture au format JSON. Plugin JQuery.

Au vu des résultats présents dans le tableau ci-dessus, j'ai finalement opté pour une bibliothèque générant la courbe côté client à l'aide du langage Javascript, notamment pour une meilleure interactivité et un meilleur design par rapport aux bibliothèques PHP. Le choix s'est ensuite porté vers **JQPlot** au détriment de HighChart pour des raisons de coût. C'est en plus un plugin de la célèbre bibliothèque JavaScript JQuery qui est très utilisée par les développeurs.

Pour retracer l'évolution de la température à la surface des panneaux solaires et la température ambiante, on retrouve sur le même graphique les deux courbes représentant la température à la surface des panneaux solaires et la température ambiante avec en abscisse le temps en **secondes** et en ordonné la température en **degré Celsius**. Cette courbe est mise à jour toutes les 5 secondes mais les courbes ne sont visibles à l'écran que dans un intervalle de 25 secondes. Bien entendu l'ensemble des valeurs enregistrées lors de la manipulation du système sont conservées pour afficher la courbe durant toute la manipulation une fois celle-ci terminée.

Pour mettre à jour toutes les 5 secondes les courbes de température, un appel **AJAX** (Asynchronous Javascript And XML) est réalisé côté client vers le serveur pour récupérer les nouvelles valeurs des températures. C'est avec les actions **Etat-Température-Ambiante** et **Etat-Température-Surface** exécutées côté serveur que les variables stockant les valeurs de la température ambiante et la température à la surface des panneaux solaires sont lues sur l'automate du panneau photovoltaïque et transmises côté client.

AJAX représente une nouvelle approche utilisant un ensemble de technologies existantes dont JavaScript, XML, HTML et permettant d'envoyer des requêtes asynchrones vers le serveur, requête qui ne «bloque» pas l'utilisateur en attendant la réponse du serveur. La page n'est pas rechargée intégralement mais seulement le bloc de la page nécessaire ce qui diminue très nettement les temps de réponse.

Voici la représentation des deux courbes retraçant l'évolution de la température ambiante et à la surface des panneaux solaires avec l'éclairage réglé à 100% et la ventilation réglée à 30%. La première image retrace l'évolution des températures sur les 25 premières secondes de la manipulation tandis que la deuxième image de la 10ème à 35ème secondes

de la manipulation. L'apprenant a en visuel à l'écran l'évolution des courbes sur 25 secondes de la manipulation.

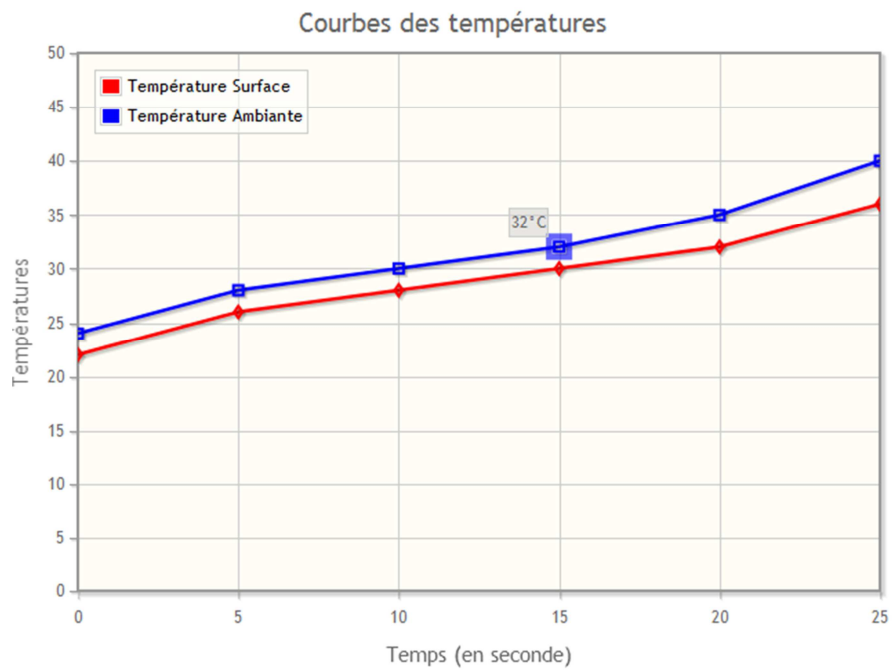


Figure 34: Courbes des températures sur les 25 premières secondes de la manipulation

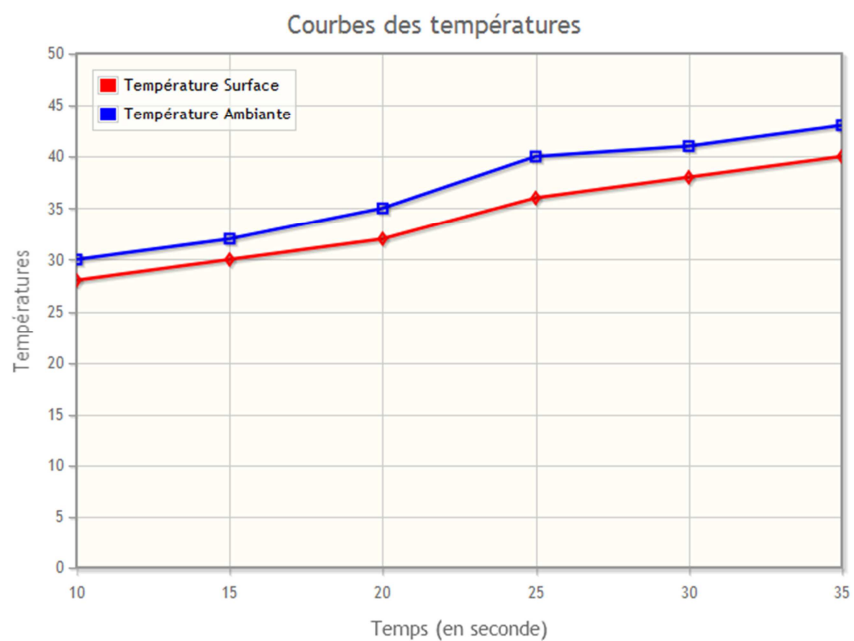


Figure 35: Courbes des températures de la 10eme à la 35eme secondes de la manipulation

3.2.4. Affichage du flux vidéo et audio

Les flux vidéo et audio envoyés par la caméra IP installée pour le panneau photovoltaïque sont accessibles sur le serveur web qu'elle embarque. J'ai ajouté un encart sur l'IHM de commande contenant la vidéo avec son en direct du panneau photovoltaïque en me connectant sur l'adresse IP du serveur web définie lors de la configuration à l'installation de la caméra. Ceci permet à l'apprenant d'avoir une vision en continu du système.

4. Expérimentation de la solution proposée

Après la phase de développement, la phase d'expérimentation reste primordiale. C'est durant l'expérimentation que les tests du pilotage du panneau photovoltaïque ainsi que de l'observation des données suite à la manipulation ont notamment pu être réalisés dans des conditions réelles au travers de la plateforme d'apprentissage DIWEB. Tout ceci dans le but de savoir si la qualité des **performances** de la solution est satisfaisante ou pas.

Les actions ont été lancées en utilisant le panneau photovoltaïque en mode **manuel**. Ce mode permet d'avoir la main sur le système sans aucune contrainte, on peut jouer sur l'éclairage et la ventilation comme on le souhaite.

Pour manipuler le panneau photovoltaïque je me suis connecté sur DIWEB dans deux endroits différents :

- ✓ En **local** au sein de la société DMS pour simuler les apprenants utilisant la plateforme à l'intérieur de l'établissement.
- ✓ En **extérieur** de la société DMS (distance significative d'environ 15km) pour simuler les apprenants utilisant DIWEB en dehors de l'établissement.

A noter que pour ces deux lieux de test, j'utilisais une connexion haut débit de l'ordre de 6Mbits par seconde.

Pour chaque facteur de test, une appréciation est donnée correspondant à la satisfaction des performances perçue. Les appréciations possibles sont en partant de la moins forte :

Faible, Acceptable, Elevée, Très élevée. Pour que les facteurs soient considérés comme satisfaisants il faut que l'ensemble des appréciations soient Elevée ou Très élevée.

Trois facteurs ont été testés : Le pilotage du panneau photovoltaïque en jouant sur l'éclairage et les ventilateurs, l'observation des données de la manipulation sous forme numérique et graphique mais également la qualité du dispositif environnemental mis en place. On notera que pour l'observation de la manipulation, les tests pour les données sous forme graphique ont été faits à l'aide de la courbe des températures tandis que ceux réalisés pour les données sous forme numérique à l'aide de la page affichant les informations du gradateur (tension, intensité, cosinus phi, ...).

Le tableau ci-dessous recense les différents facteurs testés et les appréciations attribuées pour chacune d'elle.

Numéro	Facteur	Local	Extérieur	
A	Pilotage du Panneau Photovoltaïque	Mise en marche de l'éclairage	Très Elevée	Très Elevée
		Modification de la puissance de l'éclairage	Très Elevée	Très Elevée
		Extinction de l'éclairage	Très Elevée	Très Elevée
		Mise en marche des ventilateurs	Très Elevée	Très Elevée
		Modification de la puissance des ventilateurs	Très Elevée	Très Elevée
		Extinction des ventilateurs	Très Elevée	Très Elevée
B	Observation des données de la manipulation	Affichage des données du gradateur	Elevée	Elevée
		Affichage de la courbe des températures.	Très Elevée	Très Elevée
C	Qualité du dispositif environnemental	Qualité du flux vidéo	Elevée	Elevée
		Qualité du flux audio	Elevée	Elevée

D'après ces résultats on peut s'apercevoir que les performances sont globalement très satisfaisantes puisque aucune appréciation n'est en dessous de élevée. Un petit bémol est à souligner sur l'observation des données et plus précisément sur l'affichage de la courbe des températures. Pour l'instant la courbe est mise à jour toutes les 5 secondes mais les tests

ont montré qu'il n'était pas possible de passer à un temps de rafraîchissement plus faible que 2 secondes du au temps nécessaire de réception des données de la part du serveur et à l'affichage de la courbe.

Au-delà des performances pures de la solution, la commandabilité du système au travers de l'IHM de commande reste tout à fait acceptable malgré le fait qu'elle ne soit pas tactile. De même pour l'observabilité où la caméra IP installée rend encore plus réelle la manipulation malgré qu'on ne soit pas face au système. De plus, la caméra peut être pivotée à l'aide des commandes prévues à cet effet ce qui permet d'avoir un angle de vision encore plus important.

Conclusion

L'essor d'Internet a contribué au développement d'un ensemble de dispositifs de formation que l'on retrouve sous la forme de cours, TP ou projets en ligne mais sans réelles activités pratiques de la part de l'apprenant. L'état de l'art présenté dans ce mémoire a montré que de nombreux enseignants-chercheurs se sont penchés sur le sujet des téléTP mais pour la plupart ne permettait pas la manipulation de systèmes didactiques réels.

Un des principaux objectifs de la plateforme d'apprentissage DIWEB est justement d'intégrer un ensemble de systèmes didactiques sous la forme d'un laboratoire distant pour que les apprenants puissent réaliser des TP à distance. Ce n'est en aucun cas de remplacer les TP en présentiel. Il faut considérer cet outil comme une possibilité donnée aux apprenants de passer plus de temps dans la compréhension des systèmes en dehors des heures de TP dans le but de compléter une manipulation et obtenir des résultats supplémentaires, mais également de pouvoir interpréter les résultats qui pourront être observés à partir du chargement d'une session de manipulation.

Les essais de manipulation n'ont été portés que sur un système didactique automatisé, le panneau photovoltaïque. Ces expérimentations ont révélé qu'il était possible de manipuler ce système comme si on était en présentiel répondant donc avec succès aux contraintes du temps réel. A terme DIWEB devra être capable de proposer des manipulations sur un large panel de systèmes didactiques et non plus seulement sur des systèmes automatisés mais également sur des instruments de laboratoire ou sur des systèmes réels.

D'un point de vue personnel, le travail que j'ai pu réaliser sur DIWEB m'a permis d'appréhender de nombreuses difficultés. La plus importante fut la connaissance du monde de l'automatisme qui m'était totalement inconnu au départ et dont l'acquisition d'un maximum de connaissances était primordiale dans le développement et le choix des solutions. Le fait de travailler de manière autonome tout au long du projet m'a aidé à m'organiser, à rechercher des informations, à faire des comparatifs de solutions

équivalentes pour déterminer la plus efficace dans le cadre de DIWEB ce qui a également enrichi mes compétences techniques.

Toutes fonctionnalités n'ont malheureusement pas pu être développées et être intégrées dans DIWEB, notamment celles sur l'enregistrement et la restitution des sessions de manipulation de l'apprenant mais aussi toute la partie administration du laboratoire et des systèmes didactiques. C'est sans doute le point le plus frustrant de ne pas avoir réussi à terminer ce que j'avais prévu de réaliser. Mais je pense avoir traité le plus intéressant et le plus important en réussissant à piloter le panneau photovoltaïque et à afficher sur l'IHM des résultats liés à la modification de son comportement.

Annexe : La modélisation UML

A.1 – Définition

UML est un langage de modélisation graphique à base de digrammes. Il a pour principal objectif de faciliter les transitions lors du développement d'un projet, du besoin originel à la phase d'implémentation.

Ce langage, fusion de trois méthodes orientées-objet très utilisées (Booch, OOSE (Object Oriented Software Engineering) et OMT (Object Modeling Technique)) dont le développement a débuté en 1994, et qui a été standardisé en 1997 par l'OMT (Object Management Group) dans sa version 1.1, s'appuie essentiellement sur la technologie objet et les concepts qu'elle véhicule.

UML est largement utilisé dans les projets logiciels. Elle est indépendante des langages et des environnements.

On parle volontairement de langage UML et non pas de méthode, car UML propose des outils et des concepts de modélisation, et non pas une méthodologie à respecter impérativement.

A.2 – Formalisme d'UML

Depuis la version 2, UML propose 13 types de diagrammes et se décompose en plusieurs sous-ensembles :

- ✓ Les **vues**: Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.
- ✓ Les **diagrammes**: Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.

- ✓ Les **modèles d'élément**: Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagramme. Exemple d'élément : cas d'utilisation, classe, association, ...

A.3 - Mise en œuvre d'une démarche à l'aide d'UML : les vues

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système :

- ✓ Vue des **cas d'utilisation**: c'est la description du modèle « vue » par les acteurs du système. Elle correspond aux besoins attendus par chaque acteur.
- ✓ Vue **logique**: c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs.
- ✓ Vue **d'implémentation**: cette vue définit les dépendances entre les modules.
- ✓ Vue des **processus**: c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, ...
- ✓ Vue de **déploiement**: cette vue décrit la position géographique et l'architecture physique de chaque élément du système.

A.4 - Les diagrammes

Les 13 diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

A.4.1 - Diagrammes structurels ou statiques

Diagramme de classes : il représente les classes intervenant dans le système.

Diagrammes d'objets : il sert à représenter les instances de classes (objets) utilisées dans le système.

Diagramme de composants : il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèque, bases de données ...).

Diagramme de déploiement : il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage ...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.

Diagramme des paquetages : un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le diagramme (de) Hcenc Benmohamed sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions.

Diagramme de structure composite : permet de décrire sous forme de boîte blanche les relations entre composants d'une classe.

A.4.2 - Diagrammes comportementaux

Diagramme des cas d'utilisation : il permet d'identifier les possibilités d'interactions entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

Diagramme états-transitions : permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.

Diagramme d'activité : permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

A.4.3 - Diagrammes d'interaction ou dynamiques

Diagramme de séquence : représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

Diagramme de communication : représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.

Diagramme global d'interaction : permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).

Diagramme de temps : permet de décrire les variations d'une donnée au cours du temps.

Références Bibliographiques

[LELEVE et al, 2008]

LELEVE A., MEYER C., PREVOT P., 2008. TéléTP : Premier pas vers une modélisation. Publication, INSA de Lyon.

[BENMOHAMED, 2007]

BENMOHAMED H., 2007. ICTT@Lab: un environnement informatique pour la génération et l'exécution des scénarios de téléTP. Thèse de Docteur Informatique, INSA de Lyon, 279p.

[GRAVIER, 2007]

GRAVIER C., 2007. Vers la généralisation de manipulations distantes et collaboratives d'instruments de haute technologie. Thèse de Docteur Informatique, Université Jean Monnet de Saint Etienne, 238p.

[MBALA, 2003]

MBALA HIKOLO A., 2003. Analyse, Conception, Spécification et Développement d'un système multi-agents pour le soutien des activités en formation à distance. Thèse de Docteur Informatique et Automatique, Université 302p.

[SECHAO, 2010]

SECHAO J-P., 2010. Développement d'une plateforme d'apprentissage et de pilotage de systèmes industriels pour la réalisation de travaux pratiques. Mémoire d'ingénieur informatique, IPST-CNAM, 251p.

[MARZIN et al, 2010]

MARZIN P., SANCHEZ E., DEVALLOIS D., 2010. Concevoir des protocoles expérimentaux. *Biologie-Géologie*, 135-147

[DEBRAUWER, 2009]

DEBRAUWER L., 2009. Design Patterns, Les 23 modèles de conception. ENI Editions, 329p.

[LIRIS, 2009]

Document au format PDF présentant les fonctionnalités des EIAH. Il s'agit d'un cours dispensé à l'université de Lyon, LIRIS. Disponible sur <<http://liris.cnrs.fr/stephanie.jean-daubias/enseignement/M1/EIAH&IHM-CM6-FonctionnalitesEIAH.pdf>> (consulté le 05/01/2013)

[DMS, 2012]

Site de la société DMS. Disponible sur <<http://www.dmseducation.net>> (consulté le 05/01/2013)

[TPLINE, 2012]

Site de la plateforme TPLine. Disponible sur <<http://www.tpline.net>> (consulté le 05/01/2013)

Table des illustrations

<i>Figure 1: La liaison au sol</i>	15
<i>Figure 2: La boîte de vitesse C3</i>	15
<i>Figure 3: Le pont Gustave Flaubert</i>	16
<i>Figure 4: Fonctionnement d'un API</i>	20
<i>Figure 5: Les 4 niveaux des bus de terrain</i>	23
<i>Figure 6: Lecture / Ecriture des variables d'un automate</i>	26
<i>Figure 7: Architecture du projet ICTT@Lab</i>	31
<i>Figure 8: Architecture du projet eINST</i>	34
<i>Figure 9: Exemple d'ontologie</i>	35
<i>Figure 10: Historique de la société DMS</i>	37
<i>Figure 11: Vue de profil du Panneau Photovoltaïque</i>	40
<i>Figure 12: IHM de commande du Panneau Photovoltaïque</i>	41
<i>Figure 13: Système Panneau Photovoltaïque – Intérieur de la cage</i>	41
<i>Figure 14: Caméra IP</i>	43
<i>Figure 15: Cas d'utilisation "Réaliser l'expérimentation et la collecte des Résultats"</i>	54
<i>Figure 16: Cas d'utilisation "Interpréter les résultats d'une manipulation"</i>	55
<i>Figure 17: Cas d'utilisation "Administrer un laboratoire distant"</i>	56
<i>Figure 18: Administrer un système didactique</i>	57
<i>Figure 19: Cas d'utilisation "Administrer l'utilisation des systèmes"</i>	57
<i>Figure 20: Diagramme de déploiement</i>	59
<i>Figure 21: Diagramme de classes Laboratoire Distant</i>	65
<i>Figure 22: Diagramme de classes "Apprenant - Laboratoire"</i>	66
<i>Figure 23: Diagramme de composants</i>	67
<i>Figure 24: Schéma réseau local d'un établissement</i>	69
<i>Figure 25: Interconnexion de réseau d'établissement</i>	70
<i>Figure 26: Schéma du réseau informatique local de la société DMS</i>	79
<i>Figure 27: Diagramme d'activités du Design Pattern MVC</i>	85
<i>Figure 28: Structure initiale WBS du projet</i>	87
<i>Figure 29: Structure du Lot 1000 "Fonctionnalités Apprenant"</i>	88
<i>Figure 30: Structure du Lot 2000 "Fonctionnalités Formateur"</i>	89
<i>Figure 31: Structure du lot 3000 "Fonctionnalités Administrateur"</i>	89
<i>Figure 32: IHM de commande au format Web</i>	91
<i>Figure 33: Diagramme de séquence de la méthode "Gestion-Eclairage"</i>	93
<i>Figure 34: Courbes des températures sur les 25 premières secondes de la manipulation</i>	99
<i>Figure 35: Courbes des températures de la 10eme à la 35eme secondes de la manipulation</i>	99

**Intégration de systèmes didactiques dans une plateforme d'apprentissage en ligne.
Mémoire Ingénieur CNAM, Toulouse 2013.**

RESUME

Le développement des Technologies de l'Information et de la Communication ont conduit petit à petit à une nouvelle forme d'enseignement : l'e-formation. Désormais, l'apprentissage peut se faire à distance au travers de moyens techniques comme des plateformes d'apprentissage. Pour que l'e-formation soit considérée comme un outil de formation viable, elle doit permettre la réalisation de travaux pratiques à distance et ainsi ajouter la possibilité de manipuler des dispositifs technologiques à distance.

C'est dans ce contexte que nous nous sommes intéressés à l'intégration des systèmes automatisés didactiques sous la forme d'un laboratoire distant dans une plateforme d'apprentissage en ligne.

Ce mémoire s'articule autour de 5 parties :

- ✓ L'étude de cas présentant l'évolution de la formation à distance, le pilotage des systèmes automatisés didactiques et l'étude de projets fait dans ce domaine.
- ✓ La description du cadre du sujet.
- ✓ La proposition d'architecture pour une intégration de systèmes automatisés didactiques dans une plateforme d'apprentissage.
- ✓ La conception décrivant les choix techniques mis en place pour la communication avec les automates programmables industriels des systèmes didactiques.
- ✓ La réalisation permettant de tester la proposition donnée et les choix techniques.

Il se conclut par un bilan du projet et un bilan personnel.

Mots-clés: e-formation, téléTP, systèmes automatisés didactiques, automate, UML, Modbus

ABSTRACT

The development of information and communication technologies has gradually led to a new type of teaching: e-learning. Henceforward, learning can be made with remote tools like learning platforms. In order to consider e-learning as a sustainable tool of training, the ability to make remote practicals has to exist for the handling of remote technological devices.

It's in this context that we were interested in the integration of automated didactic systems in the form of a remote laboratory integrated in an online learning platform.

This dissertation is composed of 5 parts:

- ✓ The case study showing the progress of remote training, control of automated didactic system and study of projects performed in this domain.
- ✓ The description of subject environment.
- ✓ An architecture proposal for an integration of automated didactic systems in a learning platform.
- ✓ The design describing technical choices set up for the communication with didactic system's automatons.
- ✓ The achievement allowing testing proposal and technical choices.

To conclude, it will end with a project assessment and a personal assessment.

Keywords: e-learning, distance TP, automated didactic system, automaton, UML, Modbus