

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE TOULOUSE

MÉMOIRE

Présenté en vue d'obtenir

Le DIPLOME D'INGÉNIEUR CNAM

Spécialité : Informatique

Option : Systèmes d'information

Par

Jean-Philippe SECHAO

**Développement d'une plateforme d'apprentissage et de
pilote de systèmes industriels pour la réalisation de
travaux pratiques**

Soutenu le 6 décembre 2011

JURY

PRÉSIDENT : Mme Anne Wei

MEMBRES : M. Hadj Batatia (CNAM)
M. Patrice Michel (CNAM)
M. Jean-Paul Duboué (DMS)
M. Philippe Vidal (IRIT)
M. Julien Broisin (IRIT)

Remerciements

Je tiens tout d'abord à remercier M. Patrice MICHEL, pour m'avoir proposé ce sujet de stage qui m'a tout de suite plu et a continué à m'intéresser tout au long du projet. Il m'a fait découvrir un monde que je ne connaissais pas : celui de l'enseignement. De même, il a suivi mon travail tout au long du projet, m'a fait découvrir mes méthodes formelles et m'a encouragé dans mes choix. Je le remercie pour toute l'aide qu'il m'a apporté, en particulier pour la rédaction du présent mémoire et pour ses relectures méticuleuses.

Un grand merci à mes collaborateurs de DMS pour les réunions toujours très intéressantes et riches en réflexion qui m'ont beaucoup apporté, tant au niveau professionnel que personnel. Je les remercie également pour leur aide dans la rédaction de mémoire et pour leurs relectures.

A l'achèvement de cette étude, je souhaite exprimer le plaisir que j'ai eu à travailler avec M. Patrice MICHEL comme collaborateur et partenaire de projet. Sa pédagogie a facilité l'acquisition des connaissances dans l'enseignement et son contact humain a beaucoup contribué à ce que je réalise mon stage dans les meilleures conditions. Qu'il soit donc remercié pour son appui tout au long de cette étude.

J'exprime ma reconnaissance à,

<i>à Madame Anne Wei.</i>	Professeur au Conservatoire Nationale des Arts et Métiers de Paris, qui me fait l'honneur de présider ce jury.
<i>à Monsieur Batatia</i>	Responsable pédagogique de la filière Informatique du Conservatoire Nationale des Arts et Métiers de Toulouse.
<i>à Mesdames et Messieurs</i>	Les professeurs du Centre National des Arts et Métiers de Toulouse, pour leurs compétences et la qualité de leurs enseignements.
<i>à Madame Monferran</i>	Secrétaire responsable de la section Informatique au Conservatoire Nationale des Arts et Métiers de Toulouse.

J'adresse mes sincères remerciements à,

<i>à Monsieur Cédric Barbe</i>	PDG (DMS)
<i>à Monsieur Jean-Paul Duboué</i>	Responsable de production (DMS)
<i>à mes collègues de travail</i>	DMS
<i>aux enseignants et personnels</i>	Lycée Déodat de Séverac
<i>à Monsieur Philippe Vidal</i>	Professeur d'université (IRIT)
<i>à Monsieur Julien Broisin</i>	Maître de conférence (IRIT)

Je dédie ce mémoire,

<i>à mes Parents</i>	Pour leur soutien, leur présence et leurs encouragements tout au long de mes études.
<i>à Sarah</i>	Pour m'avoir accompagné dans cette aventure difficile que sont les cours du soir.
<i>A mes Amis</i>	Qui ont supportés mon absence pendant si longtemps.

Glossaire

Applet	Petit programme Java destiné à être lancé à partir d'une autre application. Contrairement aux programmes ordinaires, un applet ne peut pas être exécuté directement par le système d'exploitation.
Base de données	Ensemble de données organisé en vue de son utilisation par des programmes correspondant à des applications distinctes et de manière à faciliter l'évolution indépendante des données et des programmes.
Compatibilité	Qualité d'un matériel ou d'un logiciel conforme aux règles d'interface d'un système informatique défini, et dont l'introduction n'altère pas les conditions de fonctionnement de ce système.
Connexion	Procédure permettant à un utilisateur de se mettre en relation avec un système informatique et, si nécessaire, de se faire reconnaître de celui-ci.
CRC	Cyclical redundancy Check, Système de calcul garantissant l'intégrité des données transmises.
Ethernet	Protocole de communication constitué de câbles, de cartes et d'un logiciel permettant à des ordinateurs de communiquer entre eux sur un réseau local. L'Ethernet standard permet de communiquer à 10 Mbit/s, l'Ethernet 100 Base-T 100 Mbits/s, et l'Ethernet 1000 Base-T à 1 GBit /s.
Firewall	Pare-feu, Élément logiciel ou matériel qui permet de contrôler le trafic (entrant et sortant généralement) et agit comme une barrière entre votre ordinateur et le réseau (internet).
FTP	File Transfer Protocol, protocole utilisé par Internet pour le transfert de fichiers.
Full-duplex	Technologie permettant une communication bidirectionnelle en temps réel, par opposition au half-duplex, où un PC reçoit ou envoie des données, ces deux actions n'étant pas simultanées.
Hub	Concentrateur, Boîtier utilisé pour recevoir des câbles et jouer le rôle d'aiguille. On l'utilise notamment dans les réseaux informatiques.
Implémenter	Réaliser la phase finale d'élaboration d'un système qui permet au matériel, aux logiciels et aux procédures d'entrer en fonction.
Interactif	Qualifie les matériels, les programmes ou les conditions d'exploitation qui permettent des actions réciproques en mode dialogué avec des utilisateurs ou en temps réel avec des appareils.
Interface	Jonction entre deux matériels ou logiciels leur permettant d'échanger des informations par l'adoption de règles communes, physiques ou logiques.
Lan	Local Area Network, Réseau informatique local reliant plusieurs ordinateurs et permettant aussi de partager des matériels, par exemple des imprimantes.
Mappe	Représentation de la localisation d'ensembles de données en vue d'en faciliter l'accès.

Mise en réseau	Constitution et exploitation d'un réseau informatique.
Ping	C'est le temps que met une machine pour répondre à la demande qui lui est faite par une autre machine. cette donnée est exprimée en ms (millisecondes).
Réseau informatique	Ensemble des moyens matériels et logiciels mis en oeuvre pour assurer les communications entre ordinateurs, stations de travail et terminaux informatiques <i>Abréviation : réseau. n. m.</i>
Réseau local	Ensemble connexe, à caractère privatif, de moyens de communication établie sur un site restreint pourvu de règles de gestion du trafic et permettant des échanges internes d'informations de toute nature, notamment sous forme de documents, images, etc. <i>Note : Le réseau local, ainsi défini en informatique, ne doit pas être confondu avec la notion de réseau local de raccordement, utilisée dans les télécommunications.</i>
Serveur	Organisme exploitant un système informatique permettant à un demandeur la consultation et l'utilisation directe d'une ou de plusieurs banques de données.
Serveur Proxy	Assure le stockage des pages Web les plus consultées. Le client évite la connexion aux sites les plus embouteillés du Réseau.
Télémaintenance	Maintenance d'une unité fonctionnelle, assurée par télécommunication directe entre cette unité et un centre spécialisé.
Temps réel	Mode de traitement qui permet l'admission des données à un instant quelconque l'obtention immédiate des résultats..
Terminal	Appareil permettant l'accès à distance à un système informatique.
URL	Uniform Resource Locator, C'est l'adresse des sites internet, autrement dit-le nom de domaine.

Liste des abréviations

Apprentissage	Ensemble d'activités qui permet à une personne d'acquérir ou 'approfondir des connaissances théoriques et pratiques, ou de développer des aptitudes.
e-formation	L'e-formation est à l'origine un sous-ensemble de la FOAD, qui s'appuie sur les réseaux électroniques. Aujourd'hui le concept d'e-learning est de plus en plus employé, attestant de l'évolution fondamentale de ce domaine de formation.
e-learning	terme anglophone pour e-formation. Utilisation des nouvelles technologies multimédias et de l'Internet, pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que les échanges et la collaboration à distance (commission européenne - 2000). Le e-learning définit tout dispositif de formation qui utilise un réseau local, étendu ou l'internet pour diffuser, interagir ou communiquer, ce qui inclut l'enseignement à distance, en environnement distribué, l'accès à des sources par téléchargement ou en consultation sur le net. Il peut faire intervenir du synchrone ou de l'asynchrone, des systèmes tutorés, des systèmes à base d'autoformation, ou une combinaison des éléments évoqués. Le e-Learning résulte donc de l'association de contenus interactifs et multimédia, de supports de distribution (PC, internet, intranet, extranet), d'un ensemble d'outils logiciels qui permettent la gestion d'une formation en ligne et d'outils de création de formations interactives. L'accès aux ressources est ainsi considérablement élargi de même que les possibilités de collaboration et d'interactivité.
ENT	Acronyme d' « environnement numérique de travail ». Dispositif global fournissant à la communauté éducative l'accès, à travers les réseaux, à la totalité des services, ressources et outils numériques en rapport avec son activité.
FAD / Formation à distance	La formation à distance est un système de formation conçu pour permettre à des individus de se former sans se déplacer sur le lieu de formation et sans la présence physique d'un formateur. La formation à distance recouvre plusieurs modalités (cours par correspondance, e-learning ...) et est incluse dans le concept plus général de Formation Ouverte et à Distance (AFNOR). La formation à distance couvre l'ensemble des dispositifs techniques et des modèles d'organisation qui ont pour but de fournir un enseignement ou un apprentissage à des individus qui sont distants du l'organisme de formation prestataire du service.
Interactions	Les interactions désignent les échanges entre le tuteur et l'apprenant, l'apprenant et les membres de sa classe virtuelle, le tuteur et la classe virtuelle.
Interactivité	Capacité d'un programme informatique à créer une situation d'échange avec un utilisateur. Possibilité pour l'apprenant d'intervenir au cours de son processus d'apprentissage et d'entraîner, par cette intervention, une modification dans le contexte de cet apprentissage.
LMS	<i>Learning management system</i> , plate-forme de (télé)formation Ce terme anglo-saxon désigne la plate-forme d'e-formation. Système informatique conçu pour optimiser, sur un réseau Internet ou Intranet, la

gestion de l'ensemble des activités de formation, depuis l'information sur l'offre, l'inscription des participants, la distribution des ressources, l'organisation de parcours individualisés, le suivi par le tuteur et du tutorat (gestion intégrée des interactions apprenants-formateur), l'animation de communautés d'apprentissage. Selon leur conception, elles favorisent l'entrée par les contenus ou les compétences, le travail collaboratif ou individuel, l'acquisition de compétences ou l'organisation de connaissances (les grains...).

- Objectif de formation** Compétence(s) à acquérir, à améliorer ou à entretenir exprimée(s) initialement par les commanditaires et/ou les formés. L'objectif de formation est l'élément fondamental des cahiers des charges. Il sert à évaluer les effets de la formation (AFNOR).
- Objectif pédagogique** Capacité(s) que le formé doit avoir acquise(s) à l'issue d'une action de formation, définie(s) par le formateur, à partir d'un objectif de formation. L'objectif pédagogique sert à construire et à conduire l'action de formation et à évaluer les capacités acquises (AFNOR).
- TIC** Acronyme de "Technologies de l'Information et de la Communication", qui désigne l'ensemble des technologies numériques (ordinateurs, réseaux, multimédia...)

Liste des acronymes

RAD	Rapid Application Development
HTTP	HyperText Transfer Protocol
PHP	Hypertext PreProcessor
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations
SOAP	Simple Object Access Protocol
WSDL	Web Service Description Language
AJAX	Asynchronous Javascript and XML
REST	Representational State Transfer
W3C	World Wide Web Consortium
UML	Unified Modeling Language
DOM	Document Object Model
W3C	Consortium World Wide Web
HTML	Hyper Text Markup Language
XHTML	Hyper Text Markup Language eXtended
DHTML	Dynamique Hyper Text Markup Language
CSS	Cascading Stylme Sheets

TABLE DES MATIERES

Remerciements.....	2
Glossaire.....	3
Liste des abréviations.....	5
Liste des acronymes.....	7
Tables des matières.....	8
Introduction.....	11
[PREMIERE PARTIE : L'ENVIRONNEMENT].....	12
I. CADRE DU STAGE.....	13
II. PRESENTATION DES STRUCTURES.....	13
II.1 – DMS Education.....	13
II.2 – Le Lycée Déodat de Séverac.....	16
[DEUXIEME PARTIE : L'ANALYSE FONCTIONNELLE].....	18
III. LA PEDAGOGIE NUMERIQUE : SUJET PROBLEMATIQUE.....	19
III.1 – Les TIC pour l'enseignement : Pourquoi ?.....	20
III.2 – Les plateformes de réalisations de TP sur Internet.....	20
IV. LE LYCEE DEODAT DE SEVERAC.....	26
IV.1 – L'enseignement via les TP dans la filière BTS Electrotechnique.....	26
IV.2 – Les moyens et outils mise à disposition des enseignants.....	27
IV.3 – Problématique de l'organisation des TP.....	30
V. LA SOCIETE DMS.....	31
V.1 – Enjeux et défis pour DMS dans les logiciels TICE.....	31
V.2 – Partenariat entre DMS et le Lycée Déodat de Séverac.....	31
V.3 – Elaboration du cahier des charges.....	32
VI. CAHIER DES CHARGES GENERALES - LE PROJET DIWEB.....	32
VI.1 – Rappel de l'objectif global.....	32
VI.2 – Situation du marché, tendances, analyse de la concurrence.....	33
VI.3 – Les innovations technologiques du projet.....	33
VI.4 – Présentation du projet.....	34
VI.5 – La plateforme.....	35
VI.6– Les utilisateurs.....	35
[TROISIEME PARTIE : L'ETUDE TECHNIQUE].....	36
VII. ETAT DES LIEUX.....	37
VII.1 – Réseau informatique du lycée Déodat de Séverac.....	37
VII.2 – Réseau informatique de la zone Electrotechnique.....	41
VII.3 – Environnement technique des systèmes communicants.....	43
VII.4 – Fonctionnement des systèmes communicants.....	44
VII.5 – Essais et analyse des systèmes.....	46
VII.6 – Les activités des étudiants sur les systèmes.....	47
VII.7 – Conclusion des activités.....	47
VII.8 – Analyse des modules industriels.....	48

VIII. CAHIER DES CHARGES DETAILLES (APRES ETUDES TECHNIQUES)	56
VIII.1 – Principales fonctionnalités de la plateforme.....	56
VIII.2 – Verrous technologiques.....	59
VIII.3 – Synthèse de fonctionnement.....	60
[QUATRIEME PARTIE : MODELISATION DU SYSTEME].....	63
IX. MODELISATION DU SYSTEME D'INFORMATION.....	64
IX.1 – UML.....	64
IX.2 – Modélisation métier avec UML.....	67
IX.3 – Modélisation des objets métiers des fonctionnalités de Diweb.....	68
IX.4 – Représentation UML des objets métiers – Cas d'utilisation.....	71
IX.5 – Représentation UML des objets métiers – Diagramme de composants.....	76
[CINQUIEME PARTIE : MISE EN PLACE].....	79
X. MISE EN PLACE ET ARCHITECTURE DU RESEAU INFORMATIQUE.....	79
X.1 – Définition du modèle d'architecture matérielle 3-tiers.....	79
X.2 – Installation et mise en place du réseau de la zone Electrotechnique.....	82
X.3 – Configuration des adresses IP.....	89
X.4 – Liaison depuis l'extérieur vers le serveur Diweb.....	89
X.5 – Réseau de connexion externe entre DMS-Lycée Déodat de Séverac.....	95
X.6 – Préparation des serveurs.....	96
XXI. INSTALLATION ET ADMINISTRATION DE SERVEUR.....	100
XI.1 – Mise en place et optimisation.....	100
XI.2 – Installation du serveur de routage et de contrôle d'accès.....	105
XI.3 – Accès à distance aux systèmes.....	108
XI.4 – Génération automatique des tables de routage & IPTable.....	109
XI.5 – Mappage du réseau.....	111
XI.6 – Ressources humaines.....	113
XXII. REDEVELOPPEMENT DES INTERFACES IHM.....	115
XII.1 – IHM d'un système communicant.....	115
XII.2 – Objectifs de l'étudiant Rémy David.....	116
XII.3 – Caractéristiques du matériel :.....	116
XII.4 – Choix de la solution la plus appropriée :.....	116
XII.5 – Présentation du logiciel :.....	117
XII.6 – Prise en main du logiciel :.....	117
XII.7 – Exemple d'IHM dans les systèmes industriels.....	118
[SIXIEME PARTIE : CONDUITE ET REALISATION].....	119
XIII – DEROULEMENT ET ORGANISATION.....	120
XIII.1 – Introduction.....	120
XIII.2 – Préparation et mise en place.....	121
XIII.3 – Méthodologie.....	136
XIII.4 – Environnement de développement.....	139
XIII.5 – Architecture du système.....	139
XIII.6 – Définition des données.....	143
XIII.7 – Modélisation et conception de la base de données.....	144
XIII.8 – Découpage des tâches de développement.....	147
XIII.9 – Répartition des tâches.....	149

XIV – DEVELOPPEMENT	155
XIV.1 – Introduction à Zend Framework.....	155
XIV.2 – Représentation UML des objets métiers – Diagrammes de classes.....	158
XIV.3 – Présentation des IHM.....	168
XIV.4 – Intégration sous Zend Framework.....	174
XIV.5 – Ajout des bibliothèques javascript et répertoire web.....	188
XIV.6 – Méthodes de récupération des données.....	191
XIV.7 – Tests unitaires.....	192
XIV.8 – Difficultés rencontrés de mise en œuvre.....	194
Conclusion	196
Liste des annexes	197
Bibliographie	248
Liste des figures	250
Résumé	251

INTRODUCTION

Ce dossier constitue le rapport de mon stage d'ingénieur, effectué au sein de la société DMS Education.

Dans une première partie, je présenterais le contexte dans lequel j'ai effectué mon stage, ainsi que ma société et nos partenaires avec lesquels le projet a été réalisé.

En seconde partie, je ferais une synthèse des informations et des thèmes que j'ai pu abordés et consultés afin de préparer l'appréhension des fonctions que j'aurais à mettre en œuvre dans le cadre du projet.

En troisième partie, je présenterais l'environnement technique, à savoir les différentes techniques issues de l'industrie pour la communication des systèmes sur lesquelles je me suis appuyé pour la réalisation de ce projet. Ainsi que le cahier des charges techniques définitif.

En quatrième partie, je décrirais le système d'information issu du cahier des charges techniques en la modélisant sous UML.

En cinquième partie, j'évoquerais commenterai la mise en place du projet en expliquant les différentes tâches d'installation ainsi que la présentation des IHM.

Ensuite, je détaillerais la conduite et la réalisation de l'application en mettant en avant les méthodes et outils utilisés.

Pour terminer, je présenterais le déroulement de mon stage et je résumerai les différents travaux effectués, en me penchant sur les difficultés rencontrées.

Bonne lecture !

PREMIERE PARTIE : L'ENVIRONNEMENT

I. CADRE DU STAGE

Le stage s'est déroulé du 21 septembre 2009 au 30 juin 2010 dans les locaux de DMS Education, une entreprise produisant des matériels didactiques destinés aux établissements scolaires. Mes tâches ont été dans un premier temps, la prise en main et la compréhension des différentes méthodes de fonctionnement de la société ainsi que le rôle de chacun.

La société ne possède pas en son sein de service informatique, il a donc fallu définir les besoins matériels et logiciels, ainsi que sa mise en place, pour la réalisation du projet. Celui-ci a été réalisé en collaboration avec le Lycée Déodat de Séverac, pour cela, des moyens m'ont été fournis pour la mise en œuvre d'une zone afin d'accueillir la plateforme pédagogique. J'ai par ailleurs, été dans ces tâches bien aidés par les enseignants et personnels de l'établissement. Que cela soit avec l'administrateur informatique du Lycée ainsi que la mission T.I.C.E¹ qui sont responsables des infrastructures et matériels informatiques dans les établissements publics de la région Midi-Pyrénées.

Le réel intérêt de ce stage était la conduite d'un projet innovant par DMS et le Lycée Déodat de Séverac, à savoir une plateforme d'apprentissage en ligne via l'accès à des systèmes industriels développés par la société DMS : le projet Diweb.

Développés avec les nouvelles technologies du web, ce projet présentait un aspect innovant non négligeable et était articulé autour d'une gestion et d'un pilotage systèmes industriels que j'ai pris beaucoup de plaisir à mettre en place et à diriger, cela m'a permis par la même occasion de favoriser mon travail d'équipe et la communication avec mes collaborateurs, ce qui convient parfaitement au cursus du diplôme d'ingénieur que j'ai suivi.

II. PRESENTATION DES STRUCTURES

II.1 – DMS Education

II.1.1 – Présentation Générale

La société DMS (Didalab Matelco STCI), créée en 2003, a pour activités la conception, réalisation et commercialisation du matériel et des cours destinés aux établissements d'enseignements technologique, professionnel et en sciences physiques. Et ceci dans les 3 domaines suivants :

- Sciences et techniques industrielles
- Sciences de l'ingénieur, génie mécanique, électrotechnique, électronique, maintenance industrielle, information et réseaux
- Maintenance automobile
- Outils de diagnostic, équipements de garage, maquettes didactiques, bancs moteurs
- Technologie au collège

¹ Technologie de l'Information et de la Communication pour l'Enseignement

Un catalogue de matériels didactiques, présenté sous forme de maquettes associées çà un programme d'enseignement spécifique, est disponible dans la partie annexe du cahier des charges.

Par la suite, DMS a fait l'acquisition de la société GTI fin 2005 ainsi que de la société MEDIA en 2006, qui respectivement ont pour spécialités la maintenance industrielle et sciences de l'ingénieur, technologie au collège. Elargissant ainsi son domaine d'activité.

Tous les produits DMS sont accompagnés d'un dossier technique, d'un dossier pédagogique et d'un dossier de ressources. Ainsi les différents niveaux de formations utilisant les produits de DMS sont principalement :

- Supérieur : CPGE, BTS, DUT
- Intermédiaire : Bac S SI, Bac Général, Bac Technologique, Bac Pro
- Initiation : Collège, BEP et CAP
- Les centres de formation professionnelle : CFA, CFPA, ...

La société possède son siège situé à au parc Aerospace de Saint Martin du Touch, proche de Toulouse. En dehors de son activité en France, DMS à une présence internationale dans des pays comme la Tunisie, l'Algérie, le Maroc, la Lybie et le Liban.

II.1.2 – Organigramme de L'Entreprise

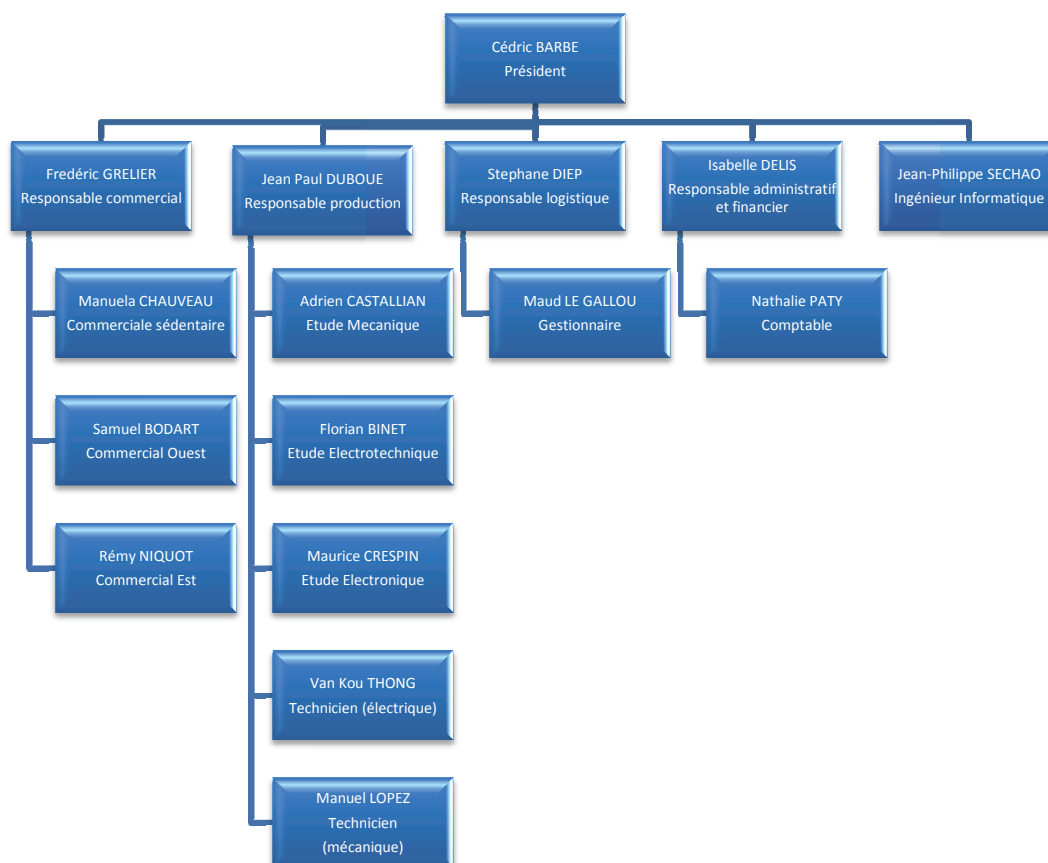


Fig. II.1.2 – Organigramme de l'entreprise DMS

II.1.3 – Historique de l'Entreprise

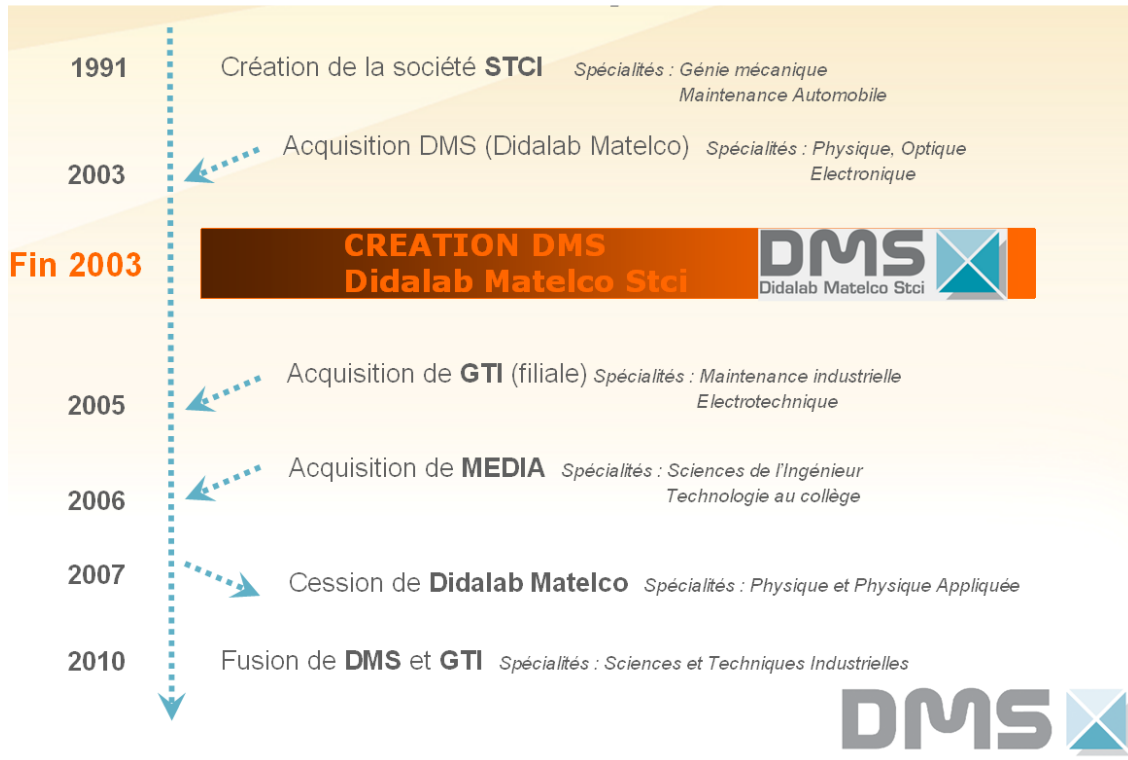


Fig. II.1.3 – Historique de l'entreprise DMS

II.1.4 – Les produits de l'entreprise



Fig. II.1.4 – Les produits de l'entreprise DMS

II.2 – Le Lycée Déodat de Séverac

II.2.1 – Présentation Générale

Le lycée Déodat de Séverac² est établissement d'enseignement secondaire technologique et professionnel qui offre un enseignement en relation avec les métiers des entreprises, afin d'acquérir des compétences, des connaissances et des savoir-faire dans un domaine professionnelle.

L'enseignement permet ainsi de préparer en 3 ans aux baccalauréats de la voie technologique répartie en 2 grandes catégories :

- S.T.L : sciences et technologies de laboratoire
- S.T.I : sciences et technologies industrielles dont les spécialités sont les suivantes :
 - Génie mécanique, Génie civil, Génie énergétique, Génie des matériaux
 - Génie électronique
 - Génie électrotechnique
 - Génie optique

Par la suite, l'établissement dispose d'enseignement dans le supérieur afin de préparer en 2 ans au BTS (Brevet de Technicien Supérieur) réparties dans les domaines suivants :

- Systèmes électroniques
- Electrotechnique
- CIRA (Contrôle Industriel et Régulation Automatique)
- GOP (Génie Optique Photonique)
- Maintenance Industrielle
- Assistance Technique d'Ingénieur
- Chimie

II.2.2 – Filière BTS Electrotechnique

II.2.2.1 – Généralités

Le diplôme BTS Electrotechnique forme des spécialistes de l'étude, de la mise en œuvre, de l'utilisation et de la maintenance des équipements électriques. Ces équipements, de plus en plus sophistiqués en raison de l'évolution des technologies de l'informatique et de l'électronique, peuvent faire intervenir des procédés d'hydraulique, de pneumatique, d'optique ...

Le technicien supérieur peut exercer son activité dans différents secteurs tels que les équipements et le contrôle industriels, la production et la transformation de l'énergie, les automatismes et la gestion technique du bâtiment, le froid et l'agroalimentaire, ou les équipements publics. C'est en maintenance et en conseil technique que les emplois se développent le plus rapidement.

Il intervient dans les secteurs de la production industrielle, du tertiaire, de l'habitat, du transport et de la distribution de l'énergie électrique. Ses interventions s'exercent sur des processus industriels de fabrication ou dans les services techniques liés à des infrastructures mettant en œuvre différentes formes d'énergie.

² Voir annexes photographies du Lycée Déodat de Séverac

II.2.2.2 – La section BTS Electrotechnique du Lycée

Les étudiants de BTS Electrotechnique se forment à l'utilisation de l'énergie électrique sur des thèmes tels que :

- La conversion d'énergie,
- La production, le transport et la distribution de l'énergie électrique,
- Les équipements communicants,
- La communication technique appliquée aux infrastructures, à l'habitat, aux bâtiments industriels et tertiaires,
- L'organisation de chantier.

Domaines d'intervention :

- Pré-étude : définition du besoin,
- Etude et développement : choix du matériel, schéma, conception,
- Réalisation et mise en œuvre : management, organisation de chantier,
- Maintenance.

Ainsi le cursus est composée de plusieurs thématiques abordés tout au long de l'enseignement tels que les sciences appliquées (séances de travaux pratiques, acquisition et modélisation, principes physiques), essais de systèmes industriels (axe z, surpressic, microcentrale), modélisation via des logiciels, automatismes (programmation de consoles, programmation d'automates)³.

³ Voir annexes photographies de la zone Electrotechnique

DEUXIEME PARTIE : L'ANALYSE FONCTIONNELLE

III. LA PEDAGOGIE NUMERIQUE : SUJET PROBLEMATIQUE

Cette partie du document fait suite à une étude réalisée lors d'une collecte et recherche d'information afin d'aborder le thème la pédagogie par le numérique et ainsi de proposer au mieux une solution qui favorisera le déploiement de ces ressources numériques adaptées aux besoins de la communauté éducative, notamment dans les filières technologiques.

L'enseignement est l'action de transmettre des connaissances à un élève, le but de celui-ci est surtout l'acquisition de savoirs ou de connaissances au moyen d'études, d'exercices et de contrôle de connaissances. Mais chaque élève apprend de manière différente et la méthode qui réussit bien à certains peut s'avérer inadaptée pour les autres. Cela peut notamment s'expliquer par la manière d'enseigner de l'enseignant. Il est de ce fait, il est encore plus difficile d'imaginer l'absence de celui-ci. Dans le cas de la pédagogie numérique, cela entraîne chez l'élève un besoin d'auto-adaptation⁴.

Cette étude nous a démontré l'importance d'initialiser un projet qui fera l'objet d'une coopération nouvelle et fructueuse entre un établissement technique, représentant l'Education Nationale et une entreprise industriel, représentant le monde professionnel. Pour cela, l'action à mener est de développer et favoriser les compétences numériques des élèves. En effet, aujourd'hui, un fort pourcentage des enfants scolarisés disposent d'un ordinateur et d'une connexion à internet à leurs domiciles. Il est important de mettre en amont l'intérêt que porte ces enfants aux numériques et aux nouvelles technologies.

Or, l'Education nationale n'est pas toujours en mesure de répondre à ce besoin et de mettre à disposition les ressources et les outils nécessaires à la formation numérique que les établissements scolaires peuvent dispenser. Il est donc nécessaire de développer ces possibilités avec l'Education Nationale afin de permettre aux élèves de bénéficier d'outils numériques. Il a été noté aussi qu'en matière d'équipements, de services numériques, d'infrastructures, la qualité du service fourni aux enseignants, aux élèves et aux personnels est fondamentale. Il ne saurait en effet y avoir de pratiques ou d'usages pédagogiques réguliers si la fiabilité, la disponibilité et la pérennité ne sont pas au rendez-vous.

De ce fait, on peut en déduire qu'il est important de mettre en place une plateforme d'identification et de présentation des ressources, des usages et bonnes pratiques afin de mieux informer les enseignants sur les ressources disponibles pour l'enseignement de leur discipline. Il s'agira ainsi de valoriser les ressources numériques éducatives de toute nature (documents, vidéos, logiciels, etc.) en le renvoyant sur cette plateforme numérique et de le laisser à disposition des autres établissements.

La difficulté sera notamment de proposer aux enseignants, un outil qui leurs permettra d'élaborer des stratégies pédagogiques en fonction de leurs difficultés et d'organiser des activités. Afin de s'adapter à l'élève, il faut déterminer son profil. Il sera donc nécessaire d'effectuer des entretiens et audit réguliers sur les différentes méthodes de travail afin que l'outil puisse s'adapter.

⁴ Cette méthode est donnée dans l'article [7].

III.1 – Les TIC pour l'enseignement : Pourquoi ?

En effectuant des recherches sur le réseau internet, il paraît évident que l'emploi des nouvelles technologies de l'information et la communication peuvent apporter de meilleurs moyens d'enseignement, du moment que ceux-ci soient utilisés à bon escient. La pédagogie doit être de bonne qualité avec une cohérence entre les outils, les méthodes et les objectifs.

La question à se poser serait-il possible de pouvoir enseigner dans les filières technologiques avec un outil informatique ? L'enseignement peut-être aussi vu comme la mise à disposition de moyens et de matériels d'apprentissage. C'est un processus interactif et une activité intentionnelle. L'objectif serait le gain de connaissances, un approfondissement de la compréhension, le développement de compétences en « résolution de problèmes » ou encore des changements dans les perceptions, les attitudes, les valeurs et le comportement.

Il faudra ainsi mettre à disposition un ensemble cohérent constitué de ressources, de stratégies, de méthodes et d'acteurs interagissant dans un contexte donné pour atteindre un but. Le but du dispositif pédagogique est de faire apprendre quelque chose à quelqu'un ou mieux (peut-on faire apprendre ?) de permettre à « quelqu'un » d'apprendre « quelque chose ». Avec l'outil technologique, nous ajoutons « quelque part » et à « n'importe quel moment »⁵.

Il n'est pas étonnant de constater que l'usage de plus en plus répandu d'Internet dans le domaine de l'apprentissage favorise l'apparition de nombreux cours en ligne et d'outils pour les créer et les diffuser. L'enseignement à distance et notamment la réalisation de TP a montré l'intérêt d'adapter le cours aux particularités de l'élève afin de faciliter l'assimilation des connaissances.

Dans le cadre des plateformes d'apprentissage en ligne, des techniques d'adaptations se sont développées afin d'individualiser l'apprentissage. Cela nous amène à relever deux types de problématiques que l'on se pose couramment lorsque l'on développe une telle application.

D'abord sur le plan des spécifications, il est nécessaire de trouver des moyens pour adapter l'ordre de présentation et l'apparence des contenus pédagogiques principalement en fonction du style d'apprentissage de l'élève. Des modèles de cours sont d'ailleurs implémentés en vue de ces adaptations.

Ensuite sur le plan technique, la plateforme produite est souvent basée sur des Web Services⁶ afin de permettre la création et la présentation de cours en ligne. Les technologies pour la mise en œuvre de ces services et de l'Interface Homme Machine restent toutefois à définir.

III.2 – Les plateformes de réalisations de TP sur Internet

Internet est un réservoir inépuisable de connaissance, lieu de rencontre pour échanger et fabriquer des outils de diffusion du savoir. D'ailleurs, l'utilisation du Web dans la formation présente plusieurs avantages. Un nombre illimité d'étudiants peut accéder à un cours en ligne, à n'importe quel moment et de n'importe où dans le monde.

L'utilisation de médias complexes, tels que les vidéos et les éléments interactifs comme les simulations et les questionnaires corrigés automatiquement, peuvent potentiellement améliorer la qualité de l'enseignement. De même, si les moyens de communication comme les espaces de travail

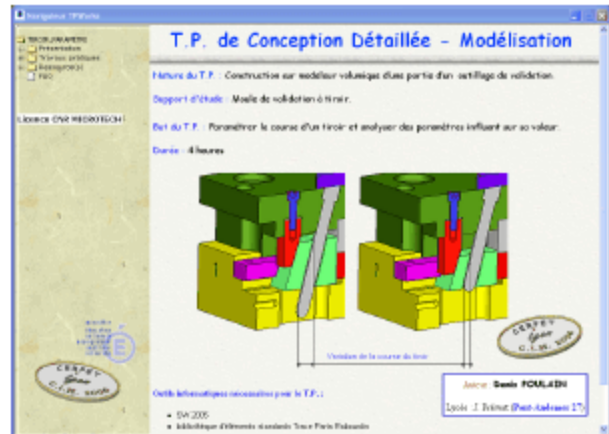
⁵ Voir article [8]

⁶ Programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués.

collaboratifs sont utilisés de façon pertinente, ils favorisent les échanges personnalisés entre étudiants et enseignants.

Exemple d'outils en ligne pour la réalisation de TP :

TP Works : C'est un logiciel qui offre une plateforme de création, de diffusion et d'échange de documents électroniques et de ressources multimédias.



TP Line : est un site internet gratuit de travaux pratiques pour l'enseignement de la technologie au collège et des sciences industrielles pour l'ingénieur (S2I) au lycée. Ces travaux pratiques vous mettent en liaison en temps réel avec de vrais systèmes industriels distants : une station anémométrique pour le TGV, une chaîne de fabrication chez Schneider-Electric, etc.



On s'aperçoit qu'en effectuant une recherche rapide sur le moteur de recherche « Google » via des mots clés tels que « TP en ligne », « TP à distance », etc., qu'il existe beaucoup de projets et d'outils liés à ce contexte. Il est donc d'actualité ce nouveau concept de l'enseignement via les TICE.

Il est toutefois à noter a plupart de ces plateforme utilisent des standards que je vais maintenant décrire.

III.2.1 – Les standards du e-learning, une contrainte ?

Un standard e-Learning⁷ est un ensemble de spécifications permettant de décrire les éléments de e-Learning et leurs modes d'utilisation.

Il ne s'agit pas ici de décrire en profondeur l'ensemble des standards les plus couramment utilisés pour d'en donner une précision sur les techniques et moyens employés dans la réalisation d'une application e-Learning. Ces standards pourraient d'ailleurs être utilisés dans le cadre du projet Diweb. Cela permettrait notamment de me sensibiliser aux processus de normalisation dans les domaines de la formation à distance. Il en expose d'ailleurs les enjeux majeurs et présente les principaux acteurs du marché.

Par ailleurs, ce phénomène commence d'ailleurs à prendre de l'ampleur dès lors que les entreprises ont compris qu'en adoptant les normes en vigueur elles pourraient rationaliser leurs coûts de formation, tout en optant pour une diffusion de connaissances rapide, voir instantanée, quelle que soit la localisation géographique de son personnel.

L'accroissement de la disponibilité des accès à haut débit à l'Internet, doublé de l'amélioration continue des performances des ordinateurs de bureau, ne font qu'accompagner le mouvement. De là, l'enjeu majeur consiste à pouvoir disposer de contenus pédagogiques indépendants du système d'exploitation, du logiciel, du matériel, du contexte et de la présentation.

Pour répondre à ces besoins les standards déjà opérationnels utilisent les concepts suivants :

- Les métadonnées⁸ et le « packaging⁹ » de cours :
 - Facilitent la recherche et le référencement des cours et de leur contenu ainsi que leur transport.
 - Facilitent la création de parcours pédagogiques et leur flexibilité par des règles associées (pré-requis, objectifs...)
- Mise en place de processus de suivi (tracking¹⁰) :
 - Permet l'enrichissement des informations sur un apprenant et le suivi de sa progression dans un contenu, dans des évaluations
- Les possibilités de dialogue entre systèmes :
 - Permettent l'échange d'informations sur les utilisateurs entre systèmes (par exemple en utilisant des profils d'apprenant).

⁷ Apprentissage en ligne, est une méthode d'apprentissage qui repose sur la mise à disposition de contenus pédagogiques via un support électronique.

⁸ Une métadonnée est une donnée qui a pour but de décrire une autre donnée.

⁹ Ensemble de ressources électronique

¹⁰ Traçage et historique des actions d'un utilisateur

III.2.1.1 - Les métadonnées

Depuis de nombreuses années, de nombreuses organisations s'impliquent dans le domaine du « e-learning » notamment le gouvernement, les entreprises, les écoles et les universités. Elles contribuent toutes à réaliser des outils performants et innovants tant les besoins en matière d'enseignement sont variés.

Mais il est fort de constater que tous ces outils ont des points en communs, plus précisément au niveau des données à manipuler. Ces données sont le plus souvent des cas appelés des métadonnées car elles constituent des données sur les données. En effet, les contenus pédagogiques sont eux-mêmes assimilables à des données.

Un besoin d'interopérabilité est également apparu entre les outils afin de distribuer la complexité et l'étendu que représente la connaissance au sens encyclopédique du terme. Afin d'éviter que chacun n'ait sa propre représentation des métadonnées il est apparu un certain nombre de standards ayant trait en général à leur représentation dans le langage XML¹¹.

L'avantage dans l'utilisation des métadonnées est qu'il va permettre d'indexer les objets pédagogiques tels que les ressources élémentaires (documents, photographies ou archives sonores) et les éléments interactifs (simulation, questionnaires, etc) qui sont à la base des cours en ligne. Elles sont utilisées pour trouver des ressources, pour les trier, elles facilitent l'interopérabilité en constituant un vocabulaire commun. Elles permettent aussi d'informer sur le cycle de vie des documents, de protéger les droits de propriété intellectuelle et de procéder à des authentifications des contenus.

Il existe une multitude de standard, mais le plus utilisé est le LOM¹², qui est une norme IEEE¹³ utilisée pour indexer des objets pédagogiques. Il comprend 78 éléments regroupés en 9 catégories :

1. Général : La fonction de cette catégorie est la description et l'identification de la ressource. Les informations contenues permettent de se référer aux autres catégories et d'établir en général les interactions avec les systèmes d'information.
2. Cycle de vie : Le cycle de vie permet d'avoir des informations sur l'histoire de la ressource et sur ses contributeurs. Les informations collectées ici serviront à la gestion de la ressource.
3. Méta-métadonnées : Les méta-métadonnées permettent de garder une trace (assez détaillée) du schéma de métadonnées utilisé pour décrire une ressource. Ces informations seront utiles pour envisager l'interopérabilité des différents systèmes et pour l'échange de données.
4. Technique : Cette catégorie présente les caractéristiques techniques qui permettront l'exécution de la ressource sur un système informatique. Les contraintes techniques d'utilisation peuvent être exprimées ici. Cette catégorie sert à décrire, évaluer et gérer les ressources.
5. Pédagogique : La pédagogie est le coeur du LOM. C'est ce qui le rend spécifique par rapport à d'autres modèles. Cette catégorie accomplit une fonction de description pédagogique de la ressource. S'y retrouvent, pour ne citer que certains des éléments, le contexte d'utilisation (scolaire, secondaire, supérieur, formation continue, etc.), le type de public cible (enseignant, auteur, apprenant, gestionnaire), la difficulté, l'âge des apprenants auxquels la

¹¹ *eXtensible Markup Language* (voir Glossaire)

¹² *Learning Object Metadata* (voir Glossaire)

¹³ *Institute of Electrical and Electronics Engineers, association professionnelle* qui a pour but de promouvoir la connaissance dans le domaine de l'ingénierie électrique.

ressource est destinée, le temps d'apprentissage type, la langue des apprenants, le niveau et le type d'interactivité.

6. Droits : Ici figurent les conditions légales d'utilisation de la ressource : indications de coût, de licences, de droits d'auteur.
7. Relation : Cette catégorie permet de lier deux ressources physiques entre elles, en donnant une valeur à cette relation, un sens (est partie de, est version de, est format de, est référencée par, etc.).
8. Commentaire : L'annotation ou commentaire permet de faire des remarques à propos de la ressource, tout en gardant en mémoire la description de la personne qui commente et la date de ce commentaire. Cette catégorie apporte des informations complémentaires utiles à l'évaluation de la ressource.
9. Classification : Il s'agit de la catégorie qui permet d'assigner des classifications et des vocabulaires contrôlés à une ressource. Des classifications peuvent être attribuées pour la discipline, les pré-requis pédagogiques, les objectifs pédagogiques, l'accessibilité, les restrictions, les compétences.

III.2.2 – Autres exemples de plateformes d'e-Learning en ligne

Le but de cette étude se portant principalement sur la possibilité de centraliser les notions d'apprentissage, il est fort intéressant de se pencher sur les outils orientés principalement pour l'enseignement et la formation autodidacte. Ces applications se différencient par les fonctionnalités qu'ils proposent mais se basent tous sur une notion de métadonnées.

Moodle se définit comme un CMS¹⁴. Il s'appuie sur une large communauté de 100 000 utilisateurs, parlant plus de 70 langues à travers 150 pays. C'est un produit robuste qui offre la plupart des fonctionnalités utilisées aujourd'hui par l'apprentissage en ligne.



ATutor (Adaptive Tutoring system) est un LCMS¹⁵ offrant des fonctionnalités d'adaptation. Il est conforme au WCAG¹⁶ du W3C¹⁷ pour l'accessibilité aux contenus web. Il permet par exemple de lire en voix synthétique les textes écrits d'une page web. Une gestion des préférences inclut le thème graphique permettant une adaptation de l'environnement. L'étudiant peut également choisir la façon de parcourir le cours à travers une navigation dans l'arbre des chapitres ou bien en utilisant un parcours séquentiel de cet arbre.



¹⁴ Content Management System, outil de gestion de contenu web.

¹⁵ Learning Content Management System, outil de gestion de contenu web orienté pédagogie.

¹⁶ Web Content Accessibility Guidelines, Règles pour l'accessibilité des contenus Web

¹⁷ World Wide Web Consortium (voir Glossaire)

Claroline

C'est un outil de e-learning mais aussi de « e-working » ou de travail à distance. Claroline a été lancé par l'université de Louvain en Belgique et depuis le centre de l'ECAM contribue également au projet.

Claroline est organisé autour du concept d'espace en relation avec un cours ou une activité pédagogique. Chaque espace met à disposition une liste d'outils qui permettent la création de contenus, la gestion des activités et d'interagir avec les élèves.



Dokeos

Une solution portée par une société du même nom. Dokeos propose une intégration des outils bureautique de Microsoft avec des fonctionnalités comme « Oogie Rapid Learning » qui transforme une présentation PowerPoint en un support de cours en ligne. Les éditions peuvent être exportées vers Excel ou tout autre outil d'analyse de données. Des fonctions de vidéo-conférence sont disponibles pour les sessions de formation.



Néanmoins, ces plateformes ne proposent pas de fonctionnalités de séquençement complexe. Les cours sont organisés en chapitres et non à l'aide d'un arbre d'objectifs. Elles ne permettent pas non plus de gérer le style d'apprentissage des étudiants qui est une des bases de l'adaptation souhaitée.

En outre, elles sont écrites en PHP (PHP: Hypertext Preprocessor), langage qui ne convient pas à la réalisation d'une architecture complexe et évolutive. Pour ces raisons, elles ne correspondaient pas au besoin exprimé.

IV. LE LYCEE DEODAT DE SEVERAC

IV.1 – L'enseignement via les TP dans la filière BTS Electrotechnique

Comme notre objectif se situe notamment au niveau de la section BTS Electrotechnique et plus précisément d'offrir aux étudiants la possibilité d'utiliser les systèmes présents dans leur atelier. Il faut approfondir la connaissance métier du rôle de l'enseignant.

Dans un premier voici ce qui résulte de l'audit réalisés auprès des enseignants lorsque j'ai demandé posé une série de questions sur leur mode de fonctionnement.

D'un point de vue « enseignant » :

L'idée de base est de permettre aux étudiants de profiter pleinement de la séance de TP en arrivant bien préparés. D'une façon générale, une séance de TP d'API¹⁸ vise à transmettre aux étudiants un savoir-faire concernant les outils et les langages de programmation des automates mais surtout la possibilité d'appliquer les concepts théoriques vus en cours pour aboutir à une « bonne » partie commande.

Le plus souvent les étudiants doivent se rendre compte de leurs erreurs pour pouvoir assimiler un principe théorique (problèmes de « fond »). Malheureusement, par manque de temps, la séance de TP permet souvent uniquement à l'étudiant d'acquérir un savoir-faire partiel. Cette expérimentation a pour objectif de mettre en évidence s'il est possible ou non à l'étudiant d'acquérir une partie du savoir-faire pendant une séance de télé-TP en vue de se consacrer principalement aux problèmes de « fond » pendant la séance en salle de TP.

Le point de vue « étudiant »

Si l'on reste objectif, pour une majorité des étudiants, l'objectif principal est de parvenir à bien réussir la séance de TP en vue d'obtenir une bonne note. Cela revient donc à réaliser l'ensemble des consignes décrites dans l'énoncé du sujet de TP dans un temps imparti. Les étudiants se regroupe au nombre de 2 ou 3. Il y'a au préalable une phase d'introduction au TP afin d'appréhender celui-ci et se familiariser avec les tâches demandées.

IV.2 – Les moyens et outils mise à disposition des enseignants

L'atelier dispose de plusieurs équipements matériels et logiciels. Ces équipements sont composés principalement de matériels informatiques et de plateforme didactiques destinés à la réalisation de T.P réparties en quatre espaces, qui disposent de ressources dans un objectif pédagogique :

- Espace Sciences Appliquées, composés de :
 - Séances de Travaux Pratiques,
 - Acquisition et Modélisation,
 - Principes Physiques

- Espace Essais de Systèmes, composés de :
 - Axe Z
 - Surpressic
 - Microcentrale

¹⁸ Automate Programmable Interface

- Espace Modélisation, composés de :
 - Psim : logiciel de simulation,
 - XRelais : Elaboration de schémas électriques (commande et puissance),
 - Ecodial : bilan de puissance,
 - Autocad Electrical : schémas électriques

- Espace Automatisation, composés de :
 - Programmation de consoles :
 - Vijeo Designer
 - Siemens : WinCC Flexible
 - Programmation d'automates :
 - Schneider Télémecanique : PL7 Pro, Unity Pro
 - Siemens : Step7
 - Omron : CXOne

IV.2.1 – Environnement technique des systèmes industriels

IV.2.1.1 – La salle de TP

Elle contient tout l'appareillage automatisation, et notamment :

- Plusieurs API (de type TSX¹⁹) relié au réseau local de l'atelier par l'intermédiaire d'un coupleur Ethernet,
- Plusieurs micro-ordinateur de type PC pour la programmation des automates. Ils sont tous reliés au réseau local de l'atelier et peuvent aussi dialoguer directement avec l'API par une liaison série RS-485²⁰. Ces PC sont essentiellement utilisés en TP pour programmer en direct l'API au moyen du logiciel PL7 Pro ou d'Unity Pro S.

IV.2.1.2 – La salle des "étudiants"

Cette salle n'étant pas encore complètement aménagée, elle sera notamment équipée de plusieurs PC, relié au réseau de l'atelier. Ainsi que de mobilier pour optimiser l'espace de travail.

Ces machines disposeront en interne de :

- Un serveur OPC²¹ (sur le PC). Le serveur OPC (OLE for Process Control) permet à des applications locales ou distantes (munies d'un client OPC), de communiquer directement avec l'API situé dans la salle de TP.
- PL7 Pro. Il s'agit du module logiciel communiquant avec l'API. Ce logiciel permet d'une part l'écriture du programme API, puis son transfert vers l'API, et enfin le suivi du déroulement du programme. Les étudiants avaient déjà travaillé avec le logiciel lors de précédents TP d'API. La communication du PC avec l'API se fait au moyen du protocole XIP (Xway²² sur IP).

¹⁹ Gamme d'automates chez Schneider Télémecanique

²⁰ Norme qui définit les caractéristiques électriques de la couche physique

²¹ Standard d'interface ouvert qui permet des connexions de type « Plug and Play » entre les différents composants d'un système d'automatisation.

²² Type de communication basé sur le modèle TCP/IP par Schneider Télémecanique.

- Un simulateur de partie opérative (PO). Il s'agit d'un moteur de calcul (simulateur auteur) permettant de simuler, une maquette physique de la salle de TP, tout en restant pilotable par l'API réel et dialoguant avec le logiciel de supervision (en DDE²³ ou avec une DLL²⁴ particulière) (cf. figure 2).

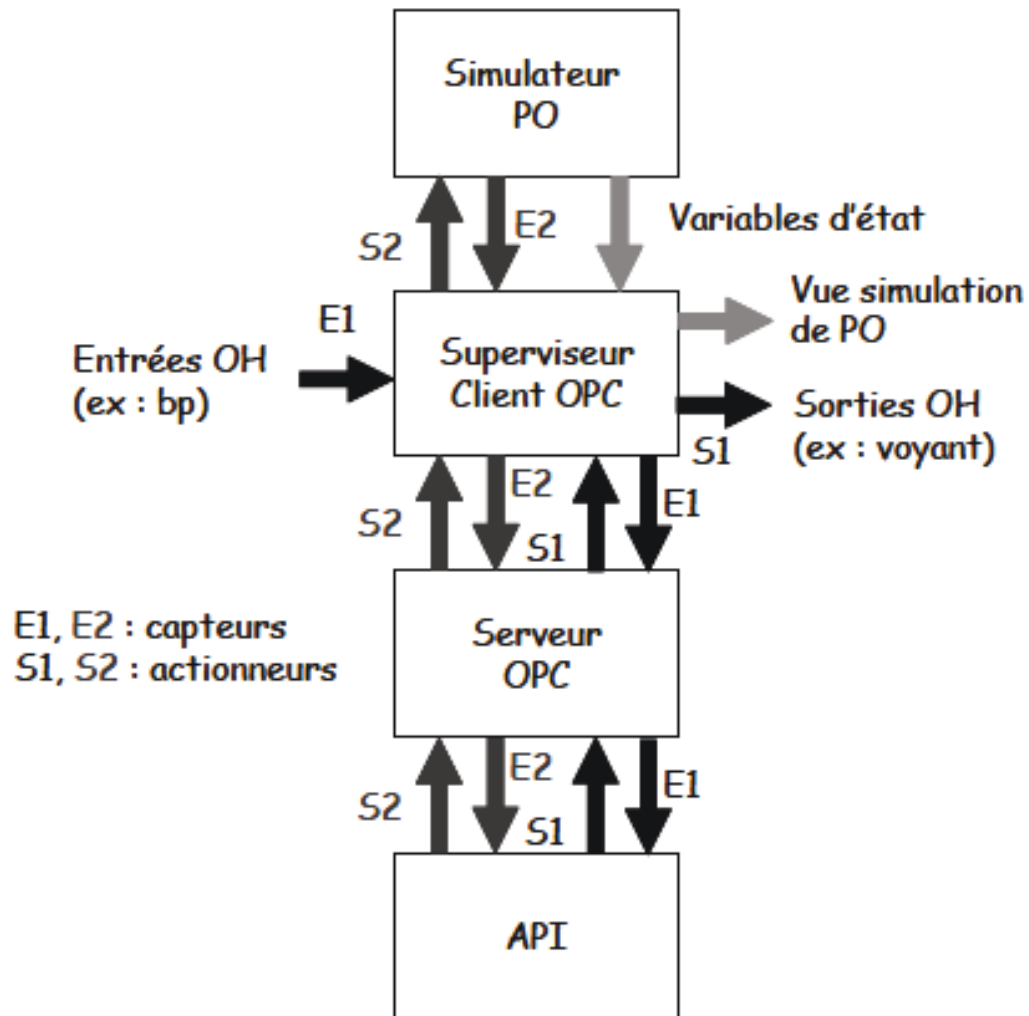


Fig. IV.2.1.2 Connexions simulateur de PO, superviseur, API

Il est important de noter que certains comportements n'ont pas volontairement été simulés. L'idée était de permettre aux étudiants durant la « véritable » séance de TP d'aborder plus rapidement les problèmes intéressants d'un point de vue pédagogique²⁵.

- Un logiciel de supervision qui gère l'affichage graphique de la maquette virtuelle. Le choix est libre. Toutefois, le logiciel de supervision utilisé doit être un client OPC pour communiquer avec le serveur OPC et client DDE pour dialoguer avec le simulateur.

IV.2.1.3 – La salle "enseignant"

La salle des enseignants se compose principalement de PC et d'imprimantes reliés sur le réseau avec un accès à Internet. Les enseignants conservent tous les logiciels et documents techniques relatives aux différents systèmes.

²³ Serveur ethernet fournissant un accès aux automates et aux équipements Modbus TCP.

²⁴ Bibliothèque logicielle

²⁵ Ces informations sont issues de l'article [5] de B. Philippe « L'instrumentation virtuelle comme outil pédagogique »

IV.2.2 – L'organisation des TP

Les travaux pratiques constituent 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.

Les travaux pratiques 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, il s'agit en l'occurrence ici des systèmes et maquettes didactiques représentatifs de systèmes réels.

Ils sont une mise en application (et une mesure de la maîtrise par les étudiants) de la méthode scientifique, basée sur la pose d'hypothèse, la conception d'un protocole expérimental, l'expérimentation, l'interprétation des résultats et le raffinement des hypothèses initiales.

L'enseignant doit ainsi savoir conduire des travaux pratiques avec méthode, et en retirer le maximum. Ainsi il n'y a pas de méthode à proprement parler pour rédiger un TP, l'enseignant est libre de faire son TP comme il le souhaite.

Toutefois, je donne ici des indications qui sont en général considérées comme des exigences sous-entendues par les correcteurs de TP.

La méthode pour faire un TP est la même que celle enseignée à l'université, et se déroule schématiquement en quatre phases :

1. **Protocole expérimental** : Il est souvent fourni dans le sujet de TP. l'élève peut le rappeler dans le compte-rendu pour en faciliter la lecture, ou au minimum préciser l'étape que l'on est en train de décrire (référence précise au plan du sujet).
2. **Expérimentation** : Les manipulations proprement dites. Si elles ne découlent pas immédiatement du sujet de TP, l'étudiant doit les préciser en détails.
3. **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. Ils ne constituent pas en soi un élément de conclusion dans un rapport de TP.
4. **Interprétation** : C'est la phase clé de la méthode expérimentale, la plus importante du compte-rendu, et celle qui est le plus souvent négligée. A cette étape, l'élève tire des conclusions de ses résultats, et surtout des variations de résultats lorsqu'on modifie le protocole expérimental. C'est à cette étape, par la qualité de la rédaction, que l'on évalue la compréhension du problème par l'élève.

L'élève ne doit pas oublier que l'un des principes de base de la méthode scientifique consiste à modifier un des paramètres de départ sans toucher aux autres, à constater des différences dans les résultats et à les interpréter en concluant sur l'influence du paramètre modifié. C'est pour cela qu'il est important de préciser pour chaque jeu de résultats, quels sont les paramètres précis de l'expérimentation.

Dans sa conclusion, l'élève doit soigneusement expliquer toute affirmation par l'observation effectuée en TP et la déduction logique qu'il en fait. Si plusieurs explications sont possibles, il doit les mentionner. Si une conclusion a été écartée, expliquez par quel raisonnement (fondé sur les observations d'une manipulation).

IV.2.3 – Le dialogue entre les enseignants et les étudiants

Les travaux pratiques sont faits en binômes voire trinômes. La capacité à travailler en équipe est donc un facteur de réussite et fera l'objet d'une évaluation par l'enseignant. Ainsi au sein du groupe chaque membre doit se partager et se répartir équitablement le travail à faire pendant le TP, à assurer que tous comprennent ce que fait chacun, et à en rédiger le compte-rendu à soumettre à l'enseignant.

Toutefois, il peut arriver que les élèves n'arrivent pas à résoudre certains problèmes. Ils s'en réfèrent à leurs enseignants. Ils les sollicitent sur des problèmes techniques ou de manipulations. Les TP sont à réaliser dans un laps de temps d'environ 3 heures à la fin du quelle les élèves soumettent leur rapport.

IV.3 – Problématique de l'organisation des TP

Plusieurs points peuvent être pris dans les problèmes de réaliser des T.P, parmi celles-ci, on retrouve notamment les suivantes qui ont une fâcheuse tendance à être redondant et causé une perte de temps :

1. La gestion de plusieurs groupes par l'enseignant, celui-ci est indisponible lorsqu'il est sollicité par un groupe et par conséquent retarde les autres si ceux-ci ont besoin de l'aide de leur enseignant,
2. L'installation du matériel et logiciels informatiques, lorsque l'environnement n'est pas préparé, les élèves perdent du temps à chercher les outils nécessaires (d'autant que si ceux-ci ne sont pas rangés correctement, il faut les chercher),
3. Le manque de préparation aux TP, les élèves négligent trop souvent cette étape, ils commencent donc le TP sans avoir de bases et de notions fondamentaux et perdent du temps à se remémorer les cours et à assimiler l'objectif du TP,
4. Le temps accordés (3 heures) aux TP peut souvent s'avérer insuffisant dès lors que les expériences à mener prennent beaucoup de temps (ex. bain régulateur où on doit chauffer l'eau à une température de 100 degrés), les élèves se retrouvent ainsi à attendre et à patienter inutilement,
5. L'indisponibilité des machines, que celle en panne ou indisponible car utilisé par un autre groupe,

V. LA SOCIÉTÉ DMS

V.1 – Enjeux et défis pour DMS dans les logiciels TICE

Avec l'évolution des moyens informatiques pour la communication, l'utilisation des TIC²⁶ dans et pour l'éducation est vue maintenant dans le monde entier comme une nécessité et une opportunité. En effet, celui-ci augmente rapidement dans beaucoup de pays.

DMS accorde donc une priorité élevée à l'utilisation des TIC pour un développement plus équitable et pluraliste en éducation notamment dans les domaines technologiques, visant ainsi à augmenter la base des connaissances, à la fois dans l'organisation de l'enseignement, et sur le plan national, travailler avec les partenaires industriels afin d'établir des principes et des conseils pour une bonne utilisation.

En cela, il est tout d'abord logique de se poser les questions suivantes :

- Comment peut-on employer les TIC pour accélérer le progrès vers l'éducation pour tous ?
- Comment les TIC peuvent-elles entraîner un meilleur équilibre entre l'équité et l'excellence dans l'éducation ?
- Et comment l'éducation peut-elle préparer les individus et la société à maîtriser et à tirer bénéfice des TIC qui imprègnent de plus en plus tous les domaines de la vie ?

Plusieurs points doivent être présents à l'esprit quand on suit ces questions.

D'abord, les TIC sont seulement une partie d'un continuum de technologies, commençant par la craie et les livres, et pouvant toutes soutenir et enrichir l'apprentissage.

En second lieu, les TIC, comme tous les outils, doivent être considérées en tant que telles, et être employées et adaptées pour servir des buts éducatifs.

Troisièmement, beaucoup de questions morales et légales interviennent dans une utilisation plus large des TIC dans l'éducation, telle que les questions de la propriété du savoir, l'échange croissant de l'éducation comme bien commercial, et la globalisation de l'éducation en relation avec la diversité culturelle.

C'est autant de réponses que de questions à se poser sur le sujet et c'est ce qui rend la tâche compliquée. C'est ainsi dans ce contexte, que DMS souhaite y trouver un moyen d'y répondre.

V.2 – Partenariat entre DMS et le Lycée Déodat de Séverac

L'idée d'un tel projet remonta bien avant mon arrivée au sein de chez DMS, en effet, une collaboration était déjà présente avec le Lycée Déodat de Séverac et DMS. En effet, la société a plus d'une fois vendue du matériel pédagogique au lycée. C'est ainsi que dans le cadre du mémoire, afin de faciliter le déroulement du projet et d'aider des élèves à réaliser leurs projets, DMS a mis à disposition 2 systèmes didactiques.

Cela a permis notamment à Patrice Michel de rencontrer des collaborateurs venant de DMS et d'initialiser le projet. C'est par ailleurs M. Michel qui a soumis le sujet de mémoire au Cnam et m'a permis de pouvoir le réaliser.

²⁶ Technologie de l'Information et de la Communication

V.3 – Elaboration du cahier des charges

Afin d'initialiser le projet, la première étape a été de formaliser les besoins et à les expliquer aux différents acteurs à l'aide d'un cahier des charges pour s'assurer que tout le monde soient en accord. En particulier, le cahier des charges permet de cadrer la ou les missions des acteurs impliqués.

Cette phase de rédaction fut notamment difficile car tout au long du projet, la communication, l'échange et le partage d'informations fut difficile dans un premier temps. Cela était dû notamment à la difficulté de pouvoir dialoguer ou communiquer de façon « informatique » sur certains aspects techniques et de solliciter les acteurs qui étaient peu disponible et voir même indisponible.

VI. CAHIER DES CHARGES GENERALES - LE PROJET DIWEB

VI.1 – Rappel de l'objectif global

Dans le cadre du développement de ces activités sur l'enseignement, ce projet s'inscrit dans la volonté de la société DMS de réaliser un outil de réalisation de TP à distance. L'objectif sera, dans un premier temps, d'adapter la solution aux différents systèmes présents au sein du lycée Déodat de Séverac et dont la totalité de leurs manipulations pourront s'effectuer à distance.

Des ressources fournies par les différents acteurs du projet (fournisseurs, partenaires, fabricants) doivent être recueillies, modifiées, synthétisées et scénarisées. Les étudiants disposeront alors de cours et d'exercices dans le but de réaliser des TP. L'étudiant obtiendra une note finale sur l'évaluation de son travail uniquement par la validation de l'équipe pédagogique de sa composante ou plus simplement par son enseignant.

Le fait d'utiliser Internet et les nouvelles technologies pour enseigner ou apprendre à distance, de manière autonome, en tout lieu, en ayant accès à des cours, exercices, tests et TP présente différents avantages.

En outre, il permet :

- De former et/ou d'informer un plus grand nombre de personne quelle que soit sa situation géographique,
- De diffuser différemment l'information et proposer un contenu évolutif,
- De proposer des entrées de parcours adaptées aux besoins de chacun,
- D'avoir recours à des modes d'apprentissage variés et d'intégrer différents degrés d'interactivités,
- De proposer de meilleures possibilités d'évaluation.

Ainsi, les étudiants :

- Ont accès à l'ensemble des matériels présents dans l'atelier (ou à distance)
- Effectue les données de mesures puis libèrent le système
- Ont accès à l'ensemble des documents utiles (ressources pédagogiques, supports de cours ...)
- Rédigent en toute autonomie leurs TP sur leur PC grâce à une interface personnalisé

Et le professeur :

- Organise son enseignement en gérant les groupes d'élèves, l'accès aux TP et aux matériels, l'enseignement par centres d'intérêts est facilité,
- Supervise le déroulement de l'ensemble des TP :
 - Peut avoir accès au travail de chaque élève (fichiers d'acquisition, TP en cours de rédaction)
 - Peut partager les données entre élèves
 - Peut centraliser l'ensemble des fichiers d'acquisition (base de données)

VI.2 – Situation du marché, tendances, analyse de la concurrence

Actuellement, une plateforme de ce genre n'existe pas sur le marché. En effet, aucune entreprise n'a proposé un outil de ce type. La tendance actuelle du marché de l'éducation nationale tend à développer des outils ENT²⁷ ou TICE aux établissements scolaires. Les moyens que disposent actuellement les entreprises concurrentes pour développer ce genre de plateforme est peu important car cela nécessite de profondes connaissances du métier de la pédagogie.

De plus, dans le cadre de ce projet, un partenariat avec un établissement scolaire montre que la tendance vers le développement de ce genre d'outils est de plus en plus sollicitée. En effet, nos relations avec les enseignants nous permettent de comprendre leurs attentes dans des outils technologiques informatiques. De plus, étant directement lié dans un environnement pédagogique, nous sommes sensibles aux recommandations fournis par les enseignants.

VI.3 – Les innovations technologiques du projet

A long terme, ce projet s'inscrit dans le cadre de plusieurs innovations technologiques, dont notamment :

- La plateforme s'inscrit dans le cadre des activités d'une entreprise expert en construction de matériels didactiques au moyen de systèmes industriels, qui seront combinés avec de fortes connaissances pédagogiques et des compétences informatiques,
- La plateforme permettra de communiquer à distance avec des systèmes industriels,
- La plateforme permettra de développer un réseau social avec tous les établissements scolaires (échange de cours et de TP),
- Cette plateforme permettra à un étudiant de réaliser des TP à distance,
- Cette plateforme permettra d'allouer des systèmes pour des établissements à distance.

²⁷ Espace Numérique de Travail

VI.4 – Présentation du projet

Le projet Diweb porte sur l'étude et la mise en place d'un réseau local dans la zone BTS Electrotechnique du lycée Déodat de Séverac afin de superviser le fonctionnement des systèmes.

La plateforme est installée sur un serveur web situé au lycée Déodat de Séverac. Ce serveur est accessible par tout type d'utilisateur. L'utilisateur qui souhaite utiliser l'application en local et à distance.

Contexte de l'application

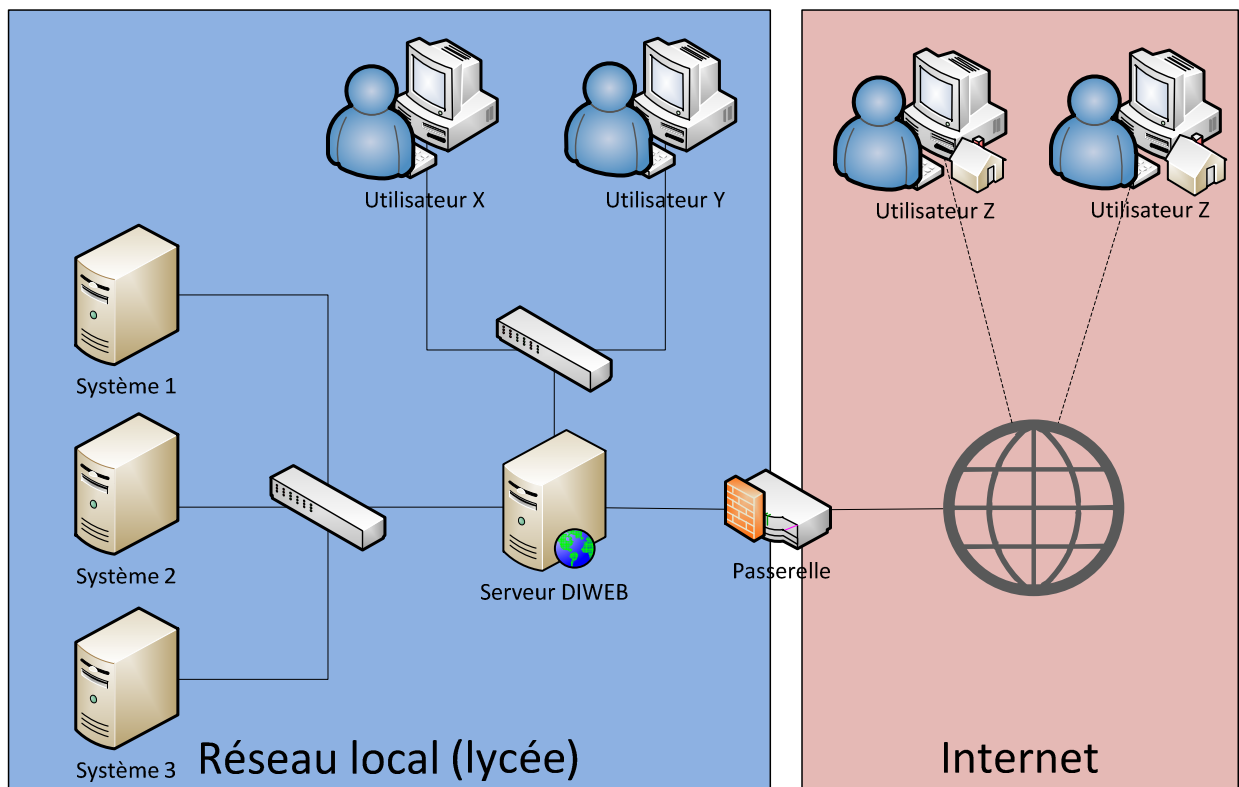


Fig. VI.1 – Schéma de fonctionnement global

Commentaires :

L'application est installée sur un serveur web situé au lycée Déodat de Séverac. Ce serveur est accessible par tout type d'utilisateur. L'utilisateur qui souhaite utiliser l'application en local et l'utilisateur souhaitant l'utiliser à distance. Sur le schéma présenté ci-dessus, on distingue clairement la partie locale et la partie internet de l'environnement.

Ces deux parties sont reliées par un accès internet protégé par une passerelle/pare-feu directement rattaché au serveur Diweb. Nous pouvons configurer la passerelle/pare-feu depuis les locaux de DMS à travers son interface WEB accessible sur internet.

VI.5 – La plateforme

La plate-forme est une application sécurisée (nécessitant une authentification) permettant l'accès à de nombreuses fonctionnalités dépendant du type d'utilisateur connecté. Le principal intérêt de la plate forme est la réalisation de TP à distance, en effet l'élève pourra avoir accès aux systèmes depuis la plate-forme et ainsi réaliser des TP en temps réel en utilisant les données renvoyées. Ainsi la principale composante de cette plate-forme est le moteur de TP.

Elle est décomposée en trois grandes parties :

- **Le bureau** : c'est la partie information & communication de la plate-forme, elle est composée de plusieurs blocs d'information dédiée à l'utilisateur courant comme ses messages, les annonces relatives à sa filière ou encore les cours récemment ajoutés. Un calendrier est également mis à disposition et met en valeurs événements et annonces du mois courant.
- **Les TP** : c'est le cœur du projet, cette partie comprend la réalisation, la gestion des TP mais aussi la prise en main à distance des systèmes. Elle comprend aussi toutes les composantes annexes comme la communication audio/vidéo.
- **L'administration** : c'est le « backstage²⁸ » de l'application, à savoir la partie gestion de toutes les données.

VI.6 – Les utilisateurs

L'application est développée pour trois types d'utilisateurs qui, une fois authentifiés, sont capable d'accéder aux fonctionnalités de la plate-forme. Ces utilisateurs sont l'élève, l'enseignant et l'administrateur.

L'élève doit être capable d'accéder aux TP disponibles pour sa filière. Il doit pouvoir avoir accès à ses informations personnelles et être capable de les modifier. Il doit avoir accès à son bureau et ainsi à toutes les informations proposées par la plate-forme (messages, annonces, événements, calendrier, nouveaux cours, etc.).

L'enseignant doit être capable de gérer les TP relatifs à sa filière. Il doit pouvoir gérer et visualiser ces TP. Il aura lui aussi accès à son bureau et à ses informations personnelles. Il peut être enseignant-administrateur et ainsi avoir accès à toutes les fonctionnalités définies pour l'utilisateur administrateur décrit ci-dessous.

L'administrateur doit être capable de gérer les informations importantes de la plate-forme, à savoir les filières ou les établissements par exemple. Il peut ajouter, modifier ou supprimer tout type d'information (ajouter, modifier ou supprimer un enseignant par exemple). C'est le super utilisateur de la plate-forme.

²⁸ Outil de gestion d'administration

TROISIEME PARTIE : L'ETUDE TECHNIQUE

VII. ETAT DES LIEUX

VII.1 – Réseau informatique du lycée Déodat de Séverac

VII.1.1 - Introduction

Il existe trois types de services nécessaires dans l'établissement : les services communs, représentant les services obligatoires et mutualisés entre le réseau pédagogique et le réseau administratif, et les services à but pédagogique permettant de développer des usages autour des nouvelles Technologies d'Information et de Communication pour l'Enseignement (T.I.C.E).

Ces services en ligne prennent en compte l'établissement dans son ensemble, avec ses activités pédagogiques, administratives, de communication interne, de communication externe (ouverture sur le milieu socio-économique, liaison avec les élèves et les enseignants à domicile ...).

VII.1.2 – Les services communs

Les services communs sont les suivants :

- Un service d'annuaire, fédérant les données relatives aux utilisateurs,
- Un service d'authentification, sur lequel s'appuient les services nécessitant une authentification, visant à garantir l'identité de l'utilisateur connecté,
- Un service de sécurisation, protégeant les différents réseaux de l'établissement ou de l'école contre les intrusions en provenance d'Internet ou des éventuels autres réseaux locaux,
- Un service d'accès au réseau, offrant un mécanisme de filtrage et de journalisation des accès des utilisateurs de l'établissement aux ressources présentes sur Internet.

VII.1.3 – Les services à but pédagogique

Les services à but pédagogique sont les suivants :

- Un service de messagerie électronique d'établissement, assurant une communication interne ou externe entre les différents acteurs de la communauté éducative de l'établissement ou de l'école;
- Un service de publication, rendant possible la mise en ligne de documents sur l'intranet de l'établissement ou de l'école et éventuellement sur Internet selon le niveau d'accréditation de l'utilisateur;
- Un service de travail collaboratif, visant à gérer des espaces de travail partagés, permettant à des utilisateurs de déposer ou consulter des documents, et de communiquer entre eux au sein d'un groupe identifiés;
- Un service de gestion documentaire, permettant l'indexation des documents internes de l'établissement ou de l'école et leur exploitation par un moteur de recherche local.

VII.1.4 – Les services à but administratif

Les services à but administratif sont les suivants :

Les données saisies par les établissements scolaires alimentent des bases locales à leur structure sur un serveur de type Netware de Novell hébergeant les applications diffusées par le Ministère et permettant de réaliser la gestion de l'établissement.

Le transfert des informations de l'administration de l'établissement vers les bases de données académiques. L'administrateur du Lycée n'a pas souhaité m'en communiquer davantage à ce sujet car ces données restent confidentielles.

La plupart de ces informations m'ont été communiqué par Vincent Murat qui est responsable du service informatique du Lycée ainsi que Jean-François Chaumont de la Mission T.I.C.E. Ils m'ont été d'une grande aide dans la compréhension du réseau et leurs participations dans le projet Diweb. Toutefois, on peut comprendre que l'administration correspond aux logiciels informatiques mis à disposition des personnels ayant la responsabilité de faire fonctionner la structure en termes de :

- Gestion de personnel, comme la gestion des heures réalisées par les professeurs, les notes des élèves, le suivi de leurs absences ...,
- Gestion de la comptabilité et de ses budgets.

VII.1.5 - La mission au TICE

Placée sous la responsabilité de la Conseillère aux Technologies de l'Information et de la Communication pour l'Enseignement (C.T.I.C.E), cette structure a les missions suivantes :

- Le développement et les usages pédagogiques des outils des T.I.C.E.,
- L'intégration de ces outils dans les pratiques,
- La mise en œuvre et le suivi des expérimentations,
- La valorisation et la mutualisation des expériences,
- L'aide à la production multimédia (en ligne et hors ligne) ,
- La visioconférence.

Elle mène son action en lien avec des établissements « pôles d'appui » chargés de la mise en œuvre des projets innovants et du rôle de conseil et d'accompagnement pour l'ensemble des enseignants engagés dans le développement des usages des T.I.C.E. La mission gère notamment la sécurité du réseau dans l'établissement, en filtrant les données entrantes et sortantes sur les réseaux TCP/IP.

VII.1.6 – La sécurité dans les réseaux TCP/IP

La sécurité dans les réseaux d'établissements scolaires est un enjeu très important, surtout lorsque cette établissement scolaire est reliée à Internet. Dans cette section du document nous allons nous attacher à décrire la sécurité uniquement sous un aspect technique. Mais il ne faut pas perdre de vue que la technique n'est qu'une petite partie des nombreux aspects que la sécurité peut revêtir.

VII.1.6.1 – Sécurité interne : les pare-feux

Actuellement nombreuses sont les établissements scolaires qui disposent d'un accès Internet. Commerce électronique, échange d'informations avec ses partenaires, sont autant de services nécessaires de nos jours au bon fonctionnement d'une structure. Mais cet accès offre une porte d'entrée sur les systèmes d'information à toute personne bien ou mal intentionnée qui se connecterait.

Par défaut, le protocole Ipv4²⁹ (le plus utilisé) ne permet pas de disposer de sécurité en natif, c'est pourquoi il existe un nombre important de services s'appuyant sur ce protocole et permettant de protéger le réseau interne du Lycée. Le plus connu de tous est certainement le pare-feu, garde-barrière ou encore coupe-feu.

Un pare-feu est un matériel ou un logiciel qui se situe à l'intersection de plusieurs réseaux, qui, ainsi délimités, sont appelés zones. Le plus souvent on le retrouve à l'intersection d'un réseau local et du réseau Internet, permettant alors de filtrer les accès en provenance de la zone la moins sûre, en l'occurrence Internet. Il permet de réaliser le filtrage des informations qui transitent d'une zone à une autre, ou encore de modifier le contenu de ces dernières avant de le transmettre.

Nous pouvons comparer ce dispositif à un routeur. A l'instar de son ancêtre, il dispose de plusieurs interfaces, servant à raccorder les différentes zones. Mais il offre des possibilités supplémentaires en assurant un contrôle du trafic, à l'aide de tables de filtrage.

L'administrateur du Lycée renseigne ces tables selon la politique de sécurité qu'il souhaite mettre en application. Elle peut être de deux types :

- Tout autoriser et bloquer certaines connexions,
- Ou à l'inverse, tout interdire et autoriser uniquement les informations qui peuvent circuler.

²⁹ Protocole IP version 4, la première utilisé

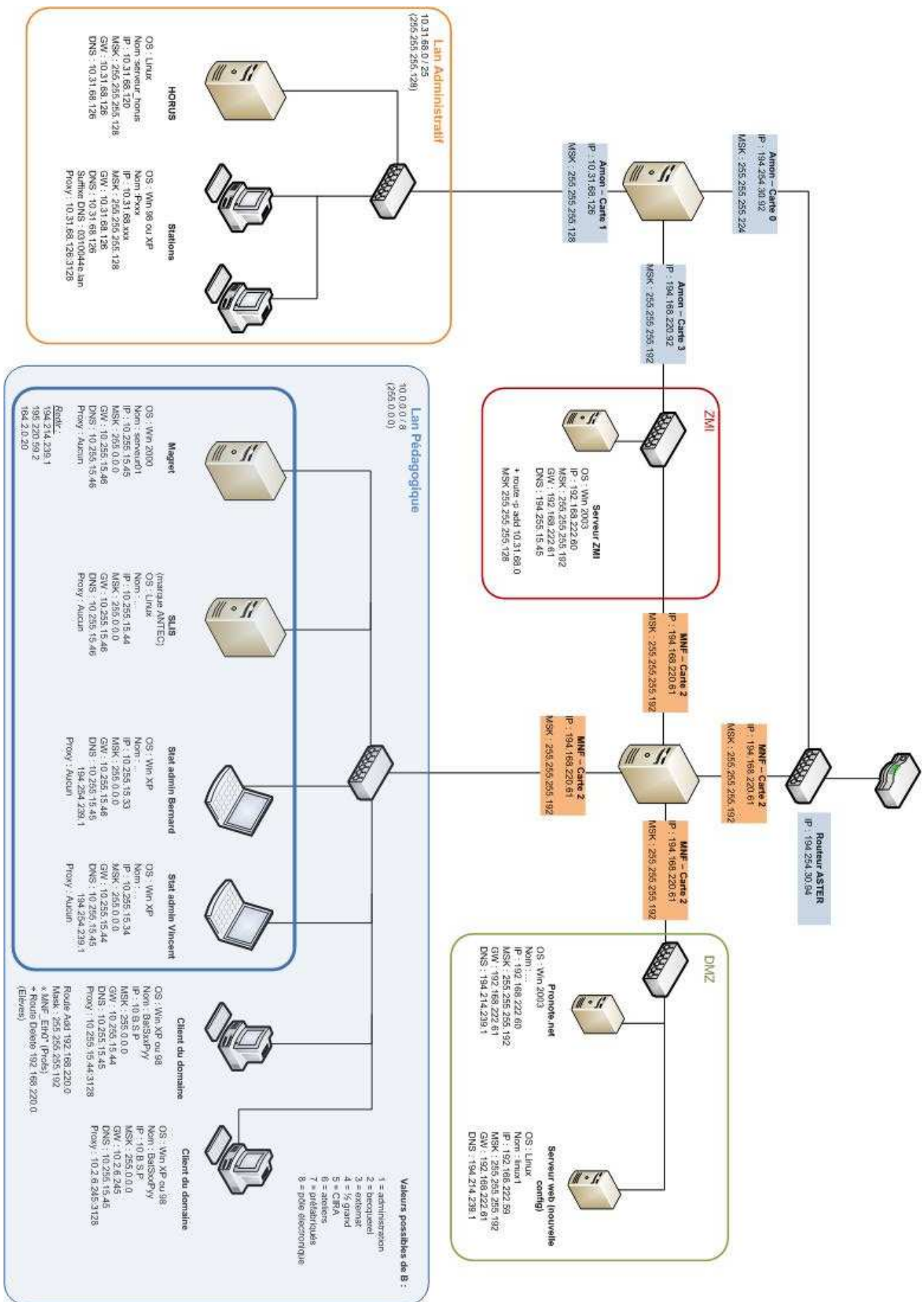


Fig. VII.1 – Architecture générale du réseau du Lycée

VII.2 – Réseau informatique de la zone Electrotechnique

Afin d'adapter les systèmes aux nouvelles technologies de communication, l'atelier a été aménagé et instrumenté de manière spécifique, tout d'abord par l'installation du réseau informatique. Cet aménagement a notamment permis de relier les postes informatiques ainsi que les systèmes automates sur le réseau local de la zone.

Grâce à cela, les étudiants sont capables de pouvoir réaliser des projets très intéressants et avancés. On peut notamment faire référence au projet de « Supervision des systèmes de la zone Electrotechnique Invitation au contrôle-commande via le réseau Internet ».

Ce projet a été réalisé en 2008, grâce à la généralisation de systèmes communicants qui a amené naturellement à proposer des sujets qui permettent d'aborder le contrôle et la commande de systèmes automatisés et d'équipements industriels. Des étudiants ont ainsi pu réaliser un projet portant sur la possibilité d'une communication des systèmes présents dans la zone. Ce travail a demandé la mise en place d'un réseau LAN³⁰ dans la zone électrotechnique afin de superviser le fonctionnement de trois systèmes de l'atelier. Cette première plateforme expérimentale a permis d'appréhender ces nouvelles technologies de l'Internet dans le milieu de l'Electrotechnique.

L'idée de départ était d'utiliser les IHM³¹ embarqués dans les systèmes qui sont principalement des fonctions du contrôle et de la commande des procédés qui assurent la surveillance et la maîtrise mais aussi la maintenance d'installations réparties sur différents sites parfois éloignés les uns des autres. En effet, l'évolution des technologies Internet se répand dans le domaine de l'industrie si bien qu'aujourd'hui il semble être le meilleur vecteur pour contrôler, commander et mesurer à distance, à partir d'une simple page HTML³² et depuis n'importe quel terminal, en milieu industriel comme en domotique. Cette fonction est possible par l'équipement par les systèmes de serveurs web embarqués.

Les objectifs de ce projet est de mettre en œuvre l'aspect communicant des systèmes. D'un point de vue pédagogique, il a pour but de disposer d'une plate-forme qui intègre des technologies innovantes qui se sont imposées dans l'industrie. Le cahier des charges prend d'ailleurs en compte les exigences sous-jacentes liées à l'application et son contexte, soit :

- Une interface homme-machine ergonomique et conviviale avec des menus déroulants intuitifs,
- Une émulation de système et de son bandeau de commande. Cela signifie que l'on devra avoir une image en temps réel de l'état du système et de sa signalisation sur la console de supervision,
- De pouvoir paramétrer le système de la console à distance.

Afin de pouvoir effectuer une communication, il a été nécessaire dans un premier temps, de réaliser un chantier pour la mise en place de l'armoire de brasage ainsi que du réseau informatique. Cet armoire a pour élément central une plateforme de communication ou d'interfaçage homme-machine plus communément appelé console de supervision, modèle G308C du constructeur Redlion.

Cette console à écran tactile est un superviseur équipé d'un système d'exploitation dédié. Elle supporte une plateforme de communication équipée d'un convertisseur de protocole ainsi que de 5 ports de communications RS232/422/485³³, d'un port USB et Ethernet permettant à des systèmes différents de communiquer ensemble et en particulier avec un automate programmable industriel.

³⁰ *Local Area Network* : Réseaux Local

³¹ *Interface Human Machine* : Interface Homme-machine pour le contrôle

³² HyperText Meta Language (voir Glossaire)

³³ Norme standardisant un bus de communication de type série

Le serveur Web embarqué permet de contrôler le système à distance. C'est une console à écran tactile qui a pour fonction de superviser l'ensemble des systèmes industriels d'une zone via un système d'exploitation dédié à la machine. Elle donne ainsi la possibilité à des produits différents de communiquer ensemble et en particulier avec un automate programmable industriel équipé d'un coupleur Ethernet.

Le nœud local, constitué par un switch industriel, permet d'interconnecter la console G308C et les systèmes reliés sur le réseau Ethernet. Les systèmes mis en réseaux possèdent tous une adresse IP, et un environnement graphique spécifique est développé à la fois dans les serveurs web embarqués de chaque système et à la fois sur la supervision permettant ainsi d'avoir la main sur chacun des systèmes en émulant la PO (Partie Opérative) ainsi que le bandeau de commande sur l'écran tactile du superviseur.

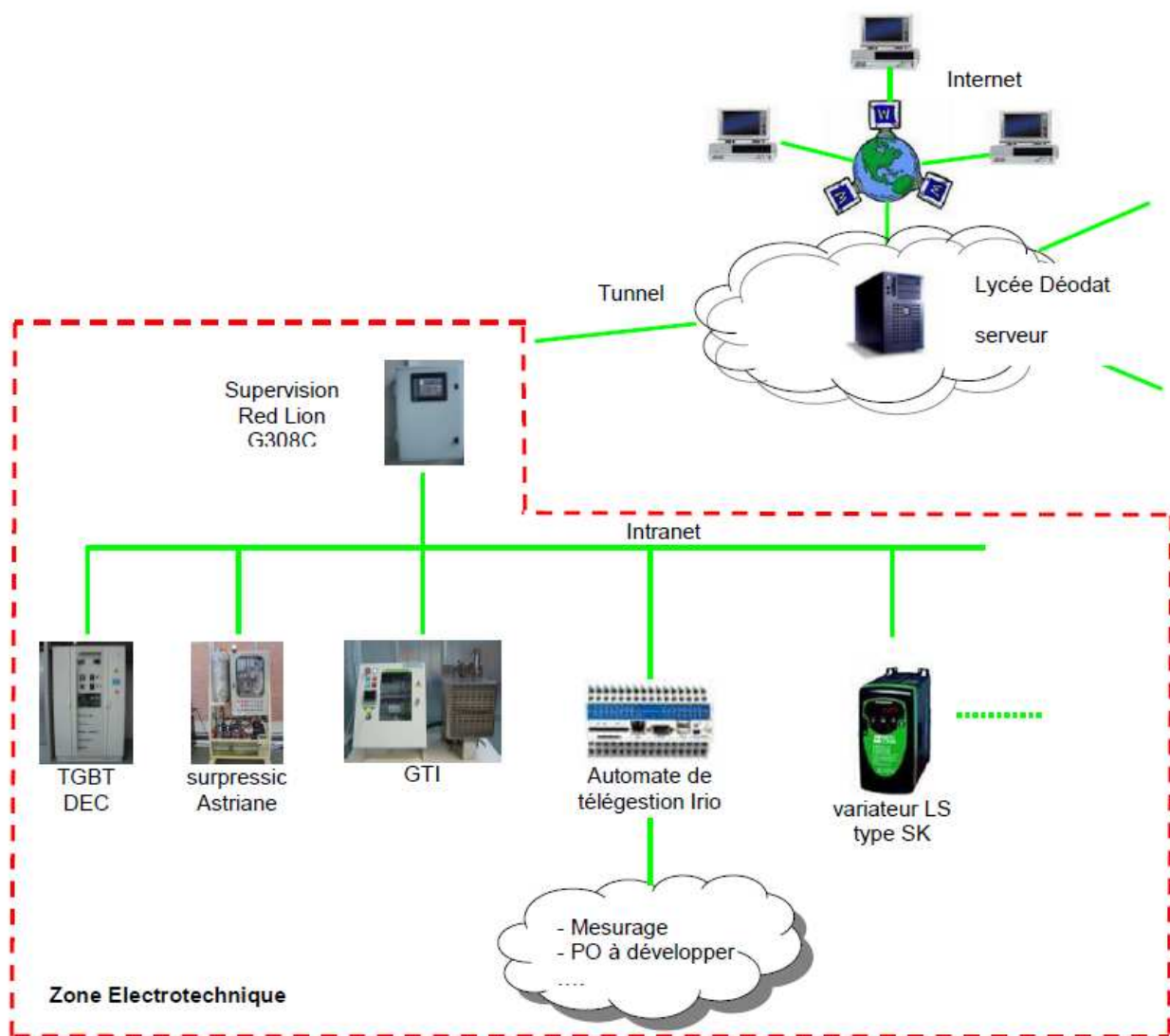


Fig. VII.2.a – Architecture du réseau informatique de la zone Electrotechnique (Projet 2008)

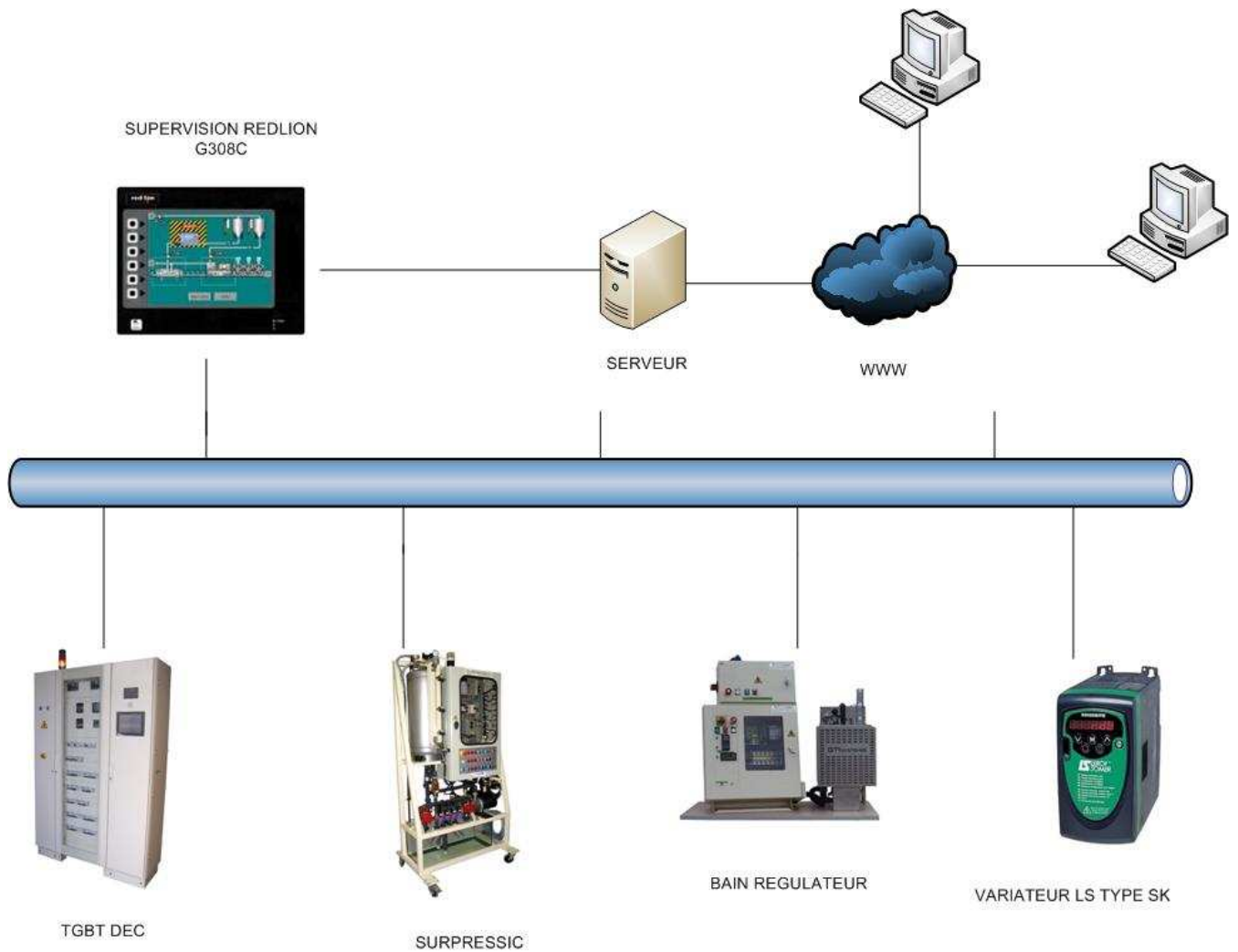


Fig. VII.2.b – Architecture du réseau informatique de la zone Electrotechnique (Projet 2010)

VII.3 – Environnement technique des systèmes communicants

Le pilotage de système à distance fait partie des possibilités de nouvelles modalités d'enseignement offertes par le développement des outils informatiques, matériels et logiciels. Ma démarche se place dans le cadre de l'étude des potentialités de la mise en réseau de systèmes didactiques existants au sein de l'atelier. En autorisant un accès distant, plusieurs étudiants, peuvent ainsi partager la même partie opérative, issue d'un système industriel réel.

On trouve de nombreuses publications et articles traitant de l'instrumentation virtuelle permettant des mesures à distance³⁴, acquisition de température, mesures de résistance ou de caractéristiques fréquentielles par exemple. De nombreux systèmes utilisés pour la réalisation de TP sont également « mises à distance » grâce à un ordinateur, ayant servi auparavant à la commande et aux mesures sur site et désormais dédié à l'interface web.

Notre projet ayant pour objectif de communiquer avec des systèmes afin de pouvoir en retirer des données, il est judicieux de connaître quels systèmes³⁵ peuvent s'adapter à notre plateforme.

³⁴ Voir Bibliographie, articles [1], [2] et [3]

³⁵ Voir Annexe 2 sur les systèmes communicants

Comme d'autres, nos systèmes devront intégrer une boucle d'action sur le procédé et pas seulement sur l'instrumentation qui l'entoure et permet également les mesures. Par ailleurs, le feedback du système vers l'étudiant est présent en image et son, via une caméra et un micro qui mettent en scène, qui concrétisent, qui rendent plus « réels », les mouvements des systèmes didactiques. Ils peuvent être ainsi utilisés comme outils de formation.

VII.4 – Fonctionnement des systèmes communicants

VII.4.1 – Configuration générale d'un système communicant

Un système communicant est basé sur une structure serveur-client. Le système est construit de manière modulaire et se compose de :

1. **Un processus physique** avec ses différents capteurs et organes de commande,
2. **Une cible temps réel** où les algorithmes de traitement de données, de commande et de réglage sont implantés localement,
3. **Un serveur d'expérience** dans un environnement Linux, à proximité du processus physique, jouant le rôle d'interface entre le réseau Internet et la cible temps réel,
4. **Un module d'interface client** comprenant les protocoles de communication et d'application, avec panneaux de commande et de visualisation,
5. **Un protocole de communication** adéquat gérant les échanges serveur-client. Ce protocole assure le transfert bidirectionnel des données entre le serveur d'expérience et le client distant,
6. **Une plateforme** offrant des services Web (documentations, exercices) et la gestion des clients et des expériences (enregistrement, réservation, etc.). Cette plate-forme est de préférence liée à une base de données dynamique.

VII.4.2 – Le serveur d'expérience

Chaque processus temps réel est connecté à un serveur local par un protocole d'application propriétaire. Ce serveur d'expérience tournant sous Linux, joue le rôle d'interface entre le réseau Internet et la cible temps réel dédiée. Pour certains processus, le serveur d'expérience Linux peut également être utilisé comme cible temps réel où les différents algorithmes de commande et de régulation sont exécutés.

A la demande du client distant, le serveur est en mesure de transmettre de manière continue et bidirectionnelle, tous les signaux nécessaires à la supervision, ainsi que le type et les paramètres de l'algorithme de traitement.

Le serveur d'expérience a également la tâche de gérer les clients : un seul client à la fois détient la main, les autres clients doivent attendre leur tour. De plus, le premier client connecté a toujours priorité sur les autres clients. Si un utilisateur client exécute de mauvaises manipulations conduisant par exemple, à des instabilités du processus ou à l'atteinte de butée, le serveur envoie un message d'avertissement lui enjoignant de ne pas dépasser les limites prescrites dans le mode d'emploi.

Les tâches de gestion des clients et des temps de connexion des utilisateurs, ainsi que le pilotage des dialogues et des services de sécurité peuvent également être paramétré et géré au niveau du protocole de communication.

VII.4.3 – Application client

L'interface utilisateurs est essentiellement composée d'animations ou éléments pour la visualisation des signaux mesurés. D'autres éléments permettent de modifier différents paramètres comme la consigne, la période d'échantillonnage, le type et les paramètres du régulateur.

Le client à distance peut également introduire une perturbation ou simuler une panne, pour observer ensuite et analyser le comportement du régulateur dans ce mode de travail.

La transmission d'une image vidéo a l'avantage de rendre l'expérience à distance, plus réaliste et attractive pour le client. Une telle transmission doit évidemment avoir un temps de réponse suffisamment faible et donc bénéficier d'une large bande passante. Une image animée, construite sur la base de signaux mesurés est certainement nettement plus parlante et explicite qu'une image retardée générée par exemple par une Web Cam.

VII.5.4 – Communication Serveur-Client

L'échange en continu de données entre le serveur et le client peut être reposé sur différentes technologies :

- Application des variables partagées entre le serveur et le client « Network-Published Shared Variables » utilisant la technologie 'Sockets' basée sur un protocole de communication bas niveau comme TCP/IP. Cette application permet une transmission bidirectionnelle fiable et particulièrement rapide. A titre d'exemple 'LabVIEW Real-Time' offre la possibilité d'échanger facilement des données entre 2 ou plusieurs systèmes distants, par simple configuration des variables à transmettre. Il est aisé d'ajouter des services de données, tels que dialogues, messages d'erreur, enregistrements, alarmes, sécurités et autres événements.
- Utilisation d'un client http et d'un navigateur Web pour le transfert de données entre le serveur d'expérience et les panneaux virtuels intégrés dans les pages Web. Le protocole de communication entre le serveur et le client http repose sur la technologie hypertexte développée pour le World Wide Web. Les données échangées entre les systèmes distants peuvent être sécurisées par différents niveaux de confidentialité.

Il convient de noter que, dans la première méthode, le client recourt à un programme d'exécution pour piloter l'installation à distance. Par contre, dans la deuxième technologie, un simple navigateur du commerce (Internet Explorer, Netscape, ...) suffit au client http, pour télé-manipuler les différents paramètres.

L'application client est un client http pouvant être appelée, en se connectant à l'adresse suivante :

<http://192.168.1.1:95>

On note que l'adresse (192.168.1.1) correspond à l'adresse IP du PC serveur d'expérience, le deuxième numéro (95) est le numéro du port du PC où les données transitent, suivi du nom du programme d'application 'html'.

Lié au contrat d'établissement du Lycée, ce projet innovant, relevant de la didactique des sciences et de la technologie, avait pour objectifs :

- D'initier l'utilisation de systèmes pilotés à distance dans les dispositifs de formation des enseignants et en définir les modalités d'utilisation,
- De faire bénéficier les professeurs stagiaires en génie électrique de systèmes industriels pilotables à distance présents au Lycée, et leur permettre de les utiliser depuis l'établissement scolaire dans lequel ils intervenaient en stage,
- De permettre aux enseignants du Lycée de suivre les évolutions technologiques, d'accompagner ou d'anticiper les modifications des référentiels dans les lycées techniques et professionnels.

VII.5– Essais et analyse des systèmes

Après avoir effectué plusieurs essais sur les systèmes, j'ai pu identifier les contraintes de sécurité liées à l'utilisation à distance des systèmes et sur l'interconnexion des réseaux au sein de l'établissement scolaire.

Cela a mis en lumière tout l'intérêt de disposer d'une zone sécurisée (DMZ³⁶) ou d'une ligne spécialisée dans les établissements scolaires. L'installation d'une caméra placée dans la zone Electrotechnique va permettre d'observer les systèmes.

De plus, les essais de mise en service d'un système piloté à distance, au moyen de connexions Ethernet utilisant le protocole TCP/IP, via l'adjonction d'une interface de liaison série Ethernet ETZ510 jouant le rôle de serveur web au module automate TSX 37 est nécessaire et dont en sont équipés un bon nombre de systèmes ont été concluantes. J'ai pu ainsi m'apercevoir des limites des fonctionnalités des systèmes.

Il est nécessaire d'avoir les outils suivants pour effectuer les essais :

- Un navigateur pour l'accès au serveur embarqué ETZ510 et au paramétrage de la caméra distante, de préférence Internet Explorer,
- Unity Pro S (configuration des drivers et paramétrage de l'automate programmable),

D'autre part, la mise au point des liaisons distantes a nécessité la mise en place sur le site du Lycée technique, par le service chargé de la gestion du réseau, d'une adresse IP publique pour un accès direct au réseau local de la zone Electrotechnique.

Cela a été possible par les services TICE du Rectorat de l'Académie de Toulouse, ont quant à eux autorisé l'utilisation d'une adresse IP publique par l'établissement scolaire, imposé l'utilisation d'un pare-feu autorisant la connexion aux adresses IP de la caméra et du coupleur Ethernet, et refusant toute autre connexion.

³⁶ Zone démilitarisée : sous-réseau séparé du réseau local et isolé de celui-ci et d'Internet par un pare-feu

VII.6 – Les activités des étudiants sur les systèmes

Les activités proposées sur les systèmes doivent tenir compte du niveau d'enseignement et des pré-acquis des élèves, mais comportaient une phase commune, guidée par une aide multimédia intégrée :

- Découverte du système (approches fonctionnelle et matérielle),
- Configuration du driver permettant de communiquer avec l'automate programmable industriel,
- Mise en service et observation du fonctionnement du système, initialement paramétré par défaut,
- Paramétrage du programme automate, observation puis interprétation du comportement du système

J'ai réalisé les essais avec l'aide de plusieurs élèves, à la fin, ils m'ont fait part des difficultés rencontrées, notamment dans l'utilisation simultanée de différents outils informatiques et le pilotage de la caméra distante.

Leurs avis ont été également recueillis sur l'utilité, l'aspect attrayant, l'efficacité de l'apprentissage à distance, par référence à l'apprentissage « traditionnel », en mode local, à proximité du système didactique.

VII.7 – Conclusion des activités

La mise en œuvre des TP est similaire à celle d'un TP en « présentiel », mais elle nécessite à l'évidence une implication supplémentaire très importante de l'enseignant pour la mise au point de l'interface informatique :

- Le paramétrage du réseau (ouverture de ports, protections),
- Les tests de connexion à distance,
- La résolution de dysfonctionnements inattendus, inhérents à la nature même du réseau internet.

Par ailleurs, de fortes compétences dans le domaine des Technologies de l'Information et de la Communication dans l'Enseignement sont indispensables pour ce genre de projet. La mise en place du Certificat Informatique et Internet Niveau 2 Enseignant (C2i2e), devenue obligatoire en formation initiale des enseignants, constitue un début de solution pour les enseignants qui manquent de connaissances en informatique.

Il semble toutefois primordial qu'au niveau d'un établissement scolaire comme le Lycée Déodat de Séverac, la responsable des systèmes communicants puisse être également administrateur du réseau du lycée. A défaut, un contact privilégié avec cette personne doit être établi par l'enseignant.

On peut aussi noter que la rapidité et l'efficacité des protocoles de transmission et d'application réduisent virtuellement la distance entre le client distant et l'expérience à télé-manipuler, et renforce ainsi le sentiment d'interaction et de proximité avec un vrai système.

L'accès à un réseau d'expériences télé-manipulables reste donc très intéressant à divers titres :

- Abaisser les coûts d'achat, par un partage à distance d'installations souvent coûteuses,
- Optimiser le taux d'utilisation des expériences,
- Offrir l'opportunité de se former à distance et d'accéder à des installations non disponibles sur place,
- Préserver la partie pratique de la formation et la mettre en valeur,
- Offrir la possibilité de faire des démonstrations depuis une salle de classe.

Les systèmes proposés ouvrent la voie à de nombreuses applications industrielles, notamment pour faciliter les travaux de mise en service et de surveillance des installations.

Malheureusement, mon champ d'expériences n'a pas encore été suffisamment exploité et éprouvé pour tirer des conclusions sur le taux d'occupation des manipulations et le degré de satisfactions des visiteurs virtuels potentiels. Un audit auprès des enseignants sera nécessaire pour en tirer des conclusions.

VII.8 – Analyse des modules industriels

Afin de pouvoir mettre en œuvre la prise en main à distance des systèmes industriels via des interfaces web, il a été nécessaire que je comprenne davantage les composants industriels utilisés. Il s'agit notamment du couple Automate + Serveur web embarqué qui sont le cœur de la communication des systèmes.

VII.8.1 – Automate programmable industriel

Un automate programmable industriel (API) est un dispositif électronique programmable destiné à la commande de processus industriels par un traitement séquentiel. Il envoie des ordres vers les pré-actionneurs (partie opérative ou PO côté actionneur) à partir de données d'entrées (capteurs) (partie commande ou PC côté capteur), de consignes et d'un programme informatique.

Plus précisément, on peut comparer ce dispositif à un ordinateur ayant des entrées et des sorties, utilisé pour automatiser des processus comme la commande des machines sur la une ligne de montage dans une usine.

Là où les systèmes automatisés plus anciens emploieraient des centaines ou des milliers de relais et de cames, un simple automate suffit. On nomme automaticiens les programmeurs de ces Automates Programmables Industriels.

L'API est structuré autour d'une unité de calcul ou processeur (en anglais Central Processing Unit, CPU), d'une alimentation (depuis des tensions AC ou DC) et, de modules suivant les besoins de l'application, tel que:

- Des cartes d'entrées – sorties (en anglais Input – Output, I/O) numériques (Tout ou rien) ou analogiques
 - Cartes d'entrées pour brancher des capteurs, boutons poussoirs, ...
 - Cartes de sorties pour brancher des actionneurs, voyants, vannes, ...
- Des modules de communication ModbusPlus, Profibus, Interbus, DeviceNet, LonWorks, Ethernet, FIPIO, FIPWAY, RS232, RS-485, AS-i, CANOpen pour dialoguer avec d'autres automates, des entrées/sorties déportées, des supervisions ou autres interfaces homme-machine (IHM), ...
- Des modules dédiés métiers, tels que de comptage rapide, de pesage, ...
- Des modules d'interface pour la commande de mouvement, dits modules Motion, tels que démarreurs progressifs, variateurs de vitesse, commande d'axes,
- Des modules de dialogue (homme-machine) tel que le pupitre (tactile ou avec clavier) dialoguant avec l'automate par réseau industriel propriétaire ou non et affichant des messages ou une représentation du procédé.

D'autres, plus anciens, étaient constitués d'une simple mémoire dont l'adresse d'entrée était constituée d'une concaténation de données d'entrée (senseurs, horloge) et de l'état précédent. Beaucoup moins onéreux, ils se prêtaient en revanche mal à une augmentation rapide du nombre d'états. Ils sont restés très utilisés pour des automatisations simples du style Anti-Blocker-System (ABS) ou *feux de signalisation* aux carrefours.

Les programmes des API sont traités selon un cycle précis : acquisition de toutes les entrées (recopie dans une mémoire image) - traitement des données (calculs) - mise à jour des sorties. Le temps d'un cycle d'API varie selon la taille du programme, la complexité des calculs et de la puissance de l'API. Le temps de cycle est généralement de l'ordre d'une vingtaine de millisecondes et est protégé par un chien de garde.

Les API se caractérisent par rapport aux ordinateurs par leur fiabilité et leur facilité de maintenance (bien que les ordinateurs industriels atteignent également un très bon degré de fiabilité). Les modules peuvent être changés très facilement et le redémarrage des API est très rapide.

L'absence d'interface Homme-machine pour visualiser l'action et le fonctionnement du programme sur la partie opérative font que les automates sont très souvent reliés par une communication à un pupitre opérateur, une interface graphique (écran d'affichage ou écran tactile) ou un PC. Dans ce dernier cas, on parle de supervision (dans certains cas, le PC peut d'ailleurs être utilisé seul en regroupant les fonctions de l'API et de la supervision, grâce à l'utilisation d'un softplc).

En automatisme industriel, on parle aussi beaucoup d'automate communiquant pour les "machines" (Automates de télégestion) : Machines d'emballage, chaudières collectives, stations d'épuration,... Dans ce cas précis, on vient, via Internet, modifier ou visualiser à distance les données ou le programme des automates de gestion de ces "machines". Elle se fait par le biais de modem-routeurs souvent associés à un logiciel assurant une liaison sécurisé (VPN³⁷).

³⁷ *Virtual Private Network* : est une extension des réseaux locaux et préserve la sécurité logique que l'on peut avoir à l'intérieur d'un réseau local. Ce réseau peut par ailleurs être distant.

VII.8.2 – Module serveur web embarqué

La plupart des systèmes de l'atelier sont équipés du modèle ETZ510 de Schneider-Télemécanique, ce sera donc sur celui-ci que se portera mon analyse. Le produit TSX ETZ 510 est un module passerelle TCP-IP / Uni-Telway autonome qui permet de réaliser la connexion des automates TSX Micro sur un réseau TCP-IP. Il communique avec les automates TSX Micro (au minimum TSX 31-10) via la prise « Terminal AUX ».

Le module TSX ETZ 510 est configurable à l'aide d'un serveur Web intégré, il n'est pas reconnu par les logiciels PL7. Il ne s'insère pas dans un rack d'automate. Ce module est donc extérieur à l'automate et peut se fixer sur un profilé DIN ou sur une platine perforée Telequick. Il est alimenté en 24 VCC et intègre une liaison série RS232 pour connecter un modem externe ou pour configurer le module. L'adaptation en vitesse sur le médium Ethernet (10/100 Mbits) est automatique.

L'automate est muni d'un pupitre de simulation.

VII.8.2.1 – Caractéristiques du module ETZ510

Les plateformes d'automatisme Micro se connectent au Réseau Ethernet TCP/IP par l'intermédiaire d'un coupleur externe TSX ETZ 510. Le coupleur Ethernet ETZ 510 dispose de :

- Un profil de communication Modbus/Uni-TE TCP/IP sur Ethernet 10/100 M bits/s ou TCP/IP par liaison série RS 232 reliée à un Modem externe à 56 K bits/s.
- La fonction serveur web embarqué. Le serveur web embarqué permet l'accès à :
 1. La configuration du coupleur,
 2. La fonction de diagnostic système automate, « Rack Viewer »,
 3. La fonction de diagnostic communication,
 4. La fonction d'accès aux variables et données automates, « Data Editor »,
 5. La fonction éditeur graphique d'objets pour l'aide à la conception de pages Web utilisateur,
 6. Un outil de configuration du serveur web embarqué.

Les applications d'automatismes distribués peuvent aujourd'hui utiliser un réseau de communication unique répondant aussi bien aux exigences de performances temps réel de l'atelier qu'à des besoins d'ouverture vers des logiciels de contrôle/commande basés sur des produits utilisant des protocoles de communication standards ou des applications utilisant la technologie Internet.

Ethernet et le protocole point à point (PPP) par liaison série répondent à ces différents besoins en terme de débit, de capacité d'ouverture sur TCP/IP et de souplesse en terme de topologie. La communication Ethernet vise essentiellement les applications de :

- Coordination entre automates programmables.
- Supervision locale ou centralisée.
- Communication avec l'informatique de gestion de production.
- Communication avec des entrées/sorties distantes.

Ce serveur est un serveur de données temps réel automate. Toutes les données du processeur de l'automate Micro supportant l'un de ces coupleurs sont présentées sous forme de pages standards Web au format HTML et sont ainsi accessibles par tout navigateur standard, capable d'exécuter du code Java embarqué, ou par le logiciel FactoryCast (fourni sur CD-Rom avec chaque coupleur Ethernet).

Le coupleur TSX ETZ 510 dispose en plus d'une mémoire type Flash EPROM³⁸ de 8 M octets (1), accessible comme un disque dur, et permettant l'accueil (hébergement) de pages Web définies par l'utilisateur. Ces pages Web peuvent être créées avec tout outil standard permettant la création et l'édition au format HTML. 8 pages peuvent être enrichies par l'insertion d'objets graphiques animés fournis par l'éditeur graphique d'objets.

Ces pages Web ainsi créées permettent entre autres de :

- Visualiser toutes variables automate en temps réel.
- Réaliser des hyperliens vers des serveurs externes (documentation, fournisseurs...).

Cette fonction convient particulièrement à la création de graphismes et d'images destinés à des fins de :

- Visualisation, contrôle, diagnostic.
- Elaboration de rapports de production temps réel.
- Aide à la maintenance.
- Guides opérateur.

La configuration d'un serveur web³⁹ embarqué nécessite le logiciel « FactoryCast » qui permet de gérer/personnaliser le site web de ce serveur. Ce logiciel est par ailleurs fourni avec le module.

Par la suite, le site est accessible par le biais d'un navigateur web et permet de visualiser et/ou modifier les réglages de la partie opérative gérée par l'automate.

³⁸ Erasable Programmable Read-Only Memory : est un type de mémoire morte reprogrammable.

³⁹ Voir Annexe 3 sur la configuration d'un module ETZ510

Factory Cast contient trois composants :

- Un serveur web intégré au module TSX ETZ 510 de l'automate.
 - Il permet d'accéder à toutes les pages web et applets Java nécessaires pour lire ou modifier les données d'exécution et les réglages de l'automate.
 - Il est possible d'ajouter des pages à ce site web ou de le modifier totalement (le module ETZ contient un serveur FTP pour transférer ces pages)
- Un logiciel de configuration qui permet de gérer le serveur web, de créer une base de données avec les symboles (variables) ou les adresses propres de l'automate et d'assurer la maintenance de la partie opérative.
- Un site web qui permet d'effectuer des diagnostics de l'exécution du programme automate via des pages Web prédéfinies. Ce site comporte aussi un éditeur de données, un éditeur graphique et un viewer de châssis pour développer nos propres pages Web.
 - ***L'éditeur de données*** permet de visualiser et de modifier des variables (symboles et adresses directes).
 - ***L'éditeur graphique*** permet de créer et d'utiliser des objets représentant des variables ou des adresses de l'automate sur le serveur web embarqué. Un synoptique de fonctionnement de l'application peut être créé par cette manière.
 - ***Le viewer de châssis*** permet de visualiser la configuration actuelle et l'état de l'automate ainsi que les différents modules additionnels ajoutés (modules E/S par exemple).

VII.8.2.1 – Connexion de l'automate avec le module ETZ510

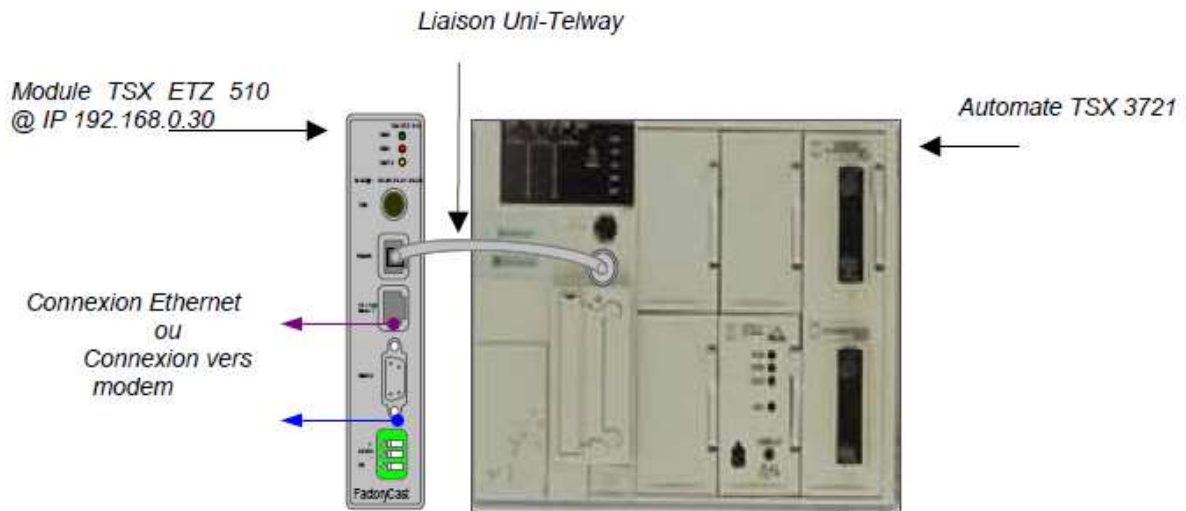


Fig. VII.8.2.1.a – Automate et coupleur de communication



Fig. VII.9.2.1. – Description du coupleur



Fig. VII.8.2.2 – Interface de configuration pour la communication avec un système

VII.8.2.3 – Modélisation sous UML du principe de fonctionnement du module web

Afin de mieux comprendre le fonctionnement d'un API, j'ai pris l'initiative de les modéliser sous forme de diagramme de cas d'utilisation. Ceux-ci afin d'appuyer ma compréhension d'un automate et par la suite de le transmettre à mes interlocuteurs.

Cas d'utilisation « Communication avec l'API » :

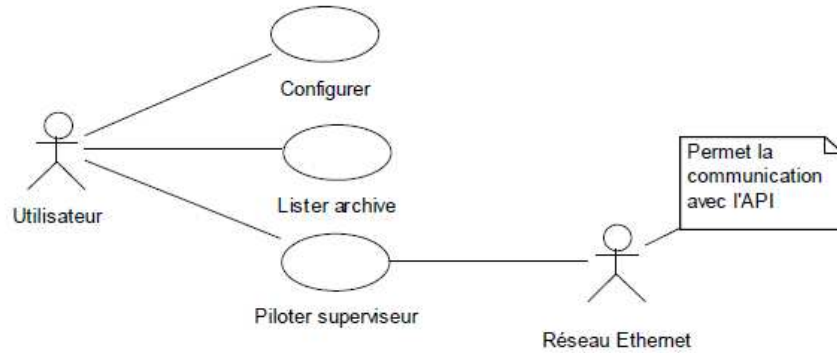


Fig. VII.8.2.3.a – Interface de configuration pour la communication avec un système

Cas d'utilisation « Piloter Superviseur » :

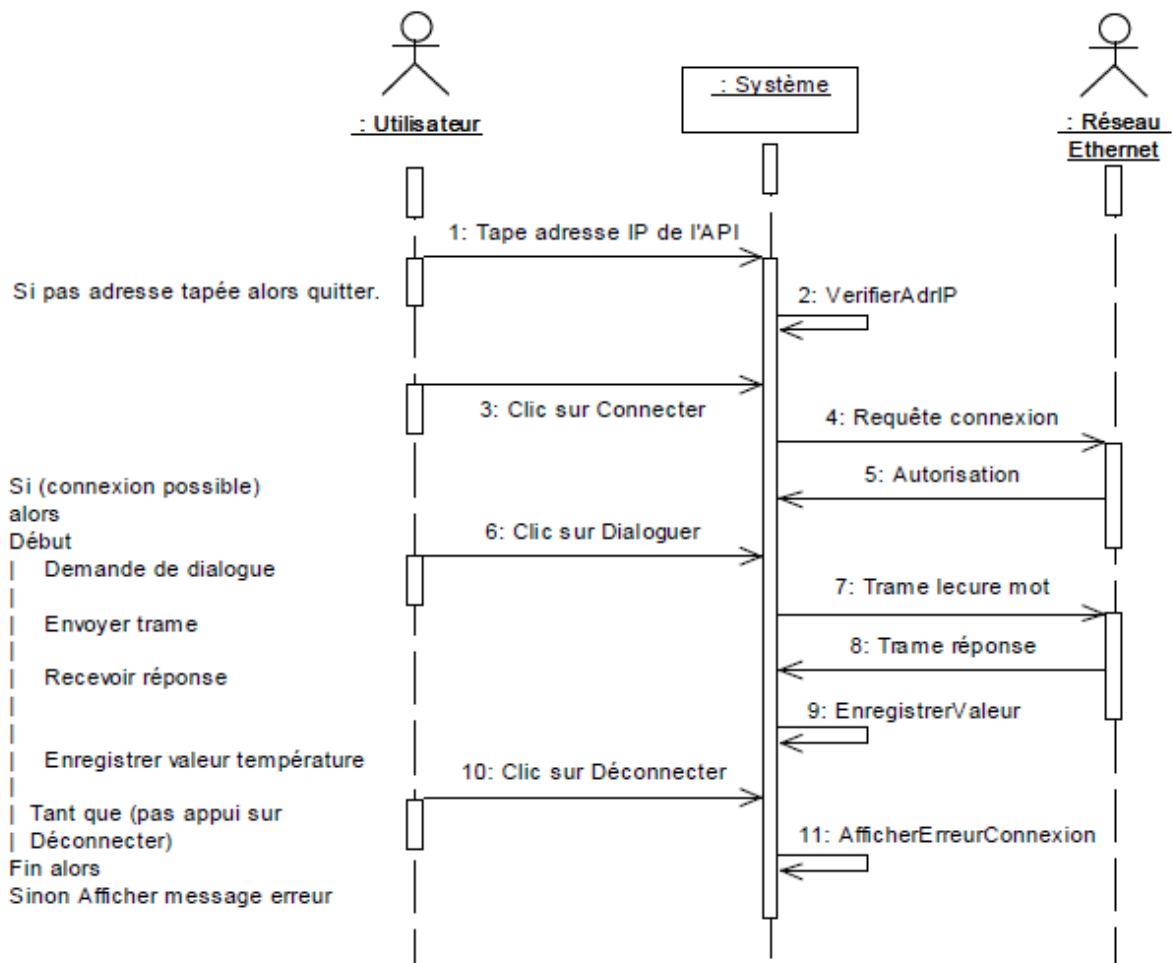


Fig. VII.8.2.3.b – Use case « Pilotage Superviseur »

Autres conception préliminaires modéliser sous UML :

Diagramme de séquence de la « Connexion, de la lecture et la déconnexion » :

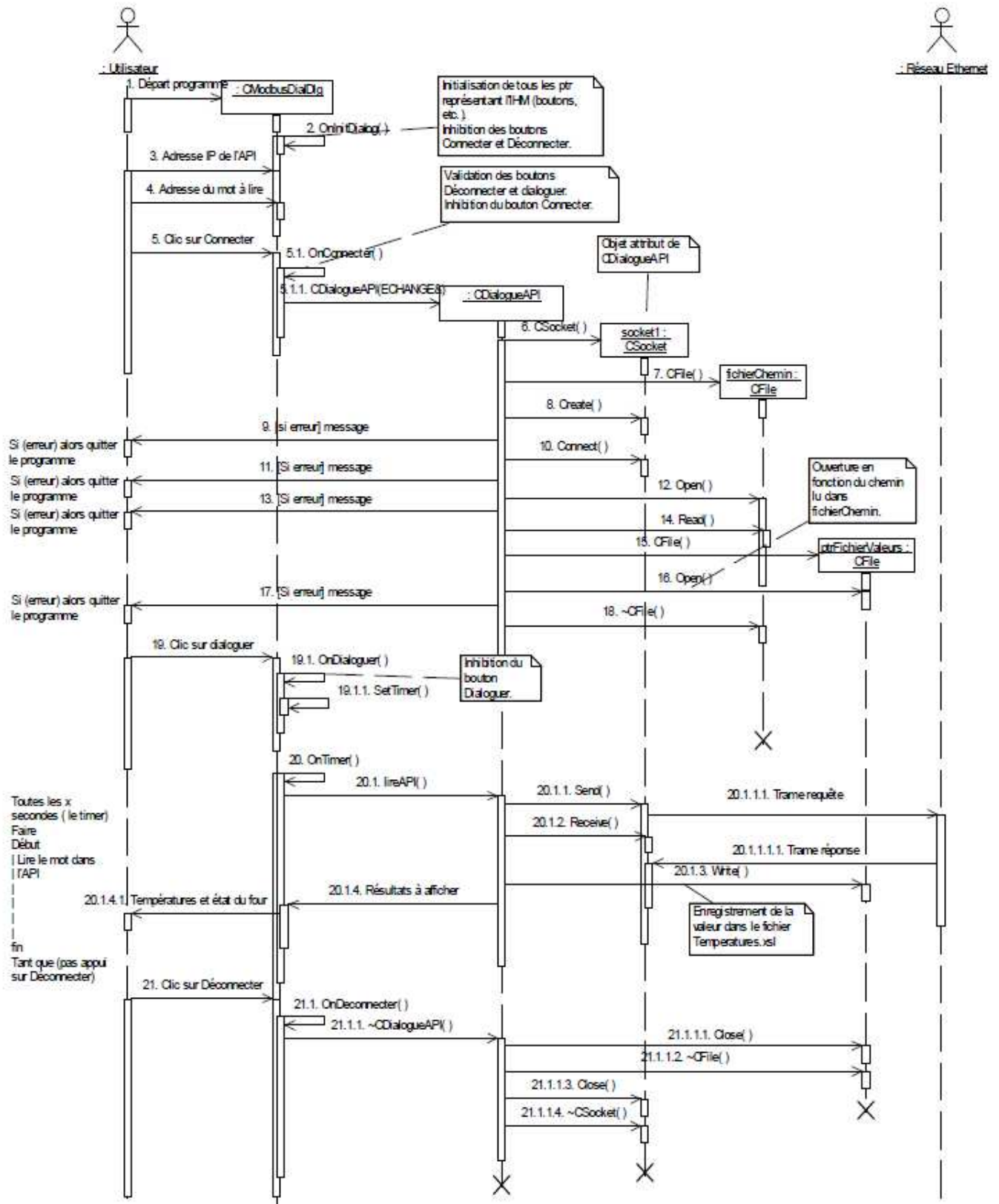


Fig. VII.8.2.3.c – Diagramme de séquence de la « Connexion, de la lecture et la déconnexion »

VII.8.3 – Le protocole Modbus

La plupart des automates dont sont équipés les systèmes de l'atelier sont du constructeur Schneider-Télemécanique. Le constructeur est propriétaire du protocole de communication Modbus⁴⁰, par conséquent tous ses modules automates programmables communiquent avec celui-ci. Je ne vais pas énumérer et détailler tous les protocoles existants, mais juste m'intéresser à celui-ci, car elles ont tous le même principe de fonctionnement.

Un protocole de communication de données définit les règles et la structure des messages utilisés par l'ensemble des périphériques d'un réseau pour l'échange de données. Ce protocole définit également l'échange correct des messages et la détection des erreurs

VIII. CAHIER DES CHARGES DETAILLES (APRES ETUDES TECHNIQUES)

Après avoir vu en détails le fonctionnement de communication, compris l'organisation des TP et ses problématiques. J'ai pu en définir les possibilités de l'application et dresser une liste des fonctionnalités possibles de la plateforme. En cela, à partir du cahier des charges initial. J'ai rédigé un document de spécifications techniques (ou Cahier des Charges Techniques).

VIII.1 – Principales fonctionnalités de la plateforme

1. Communication avec les systèmes pédagogiques au sein d'un réseau local.
 - La plateforme permet de prendre la main sur un système si celui-ci est connecté sur un réseau local relié à celui-ci.
2. Communication avec les systèmes pédagogiques au sein d'un réseau extérieur.
 - La plateforme permet de prendre la main sur un système si celui-ci est connecté sur un réseau local accessible à distance.
3. Développement des interfaces web pour communiquer avec les systèmes pédagogiques de la plateforme Diweb.
 - La plateforme permet d'accéder aux systèmes pédagogiques ne peut s'effectuer que par les interfaces de communication de celui-ci.
4. Développement des interfaces pour sauvegarder les mesures dans les systèmes pédagogiques.
 - La plateforme permet de pouvoir enregistrer et stocker les mesures relevées sur les interfaces des systèmes pédagogiques dans la base de données de la plateforme. Il est ainsi possible de pouvoir reprendre ces mesures ultérieurement.
5. Centralisation des travaux et tâches des étudiants.
 - La plateforme permet aux enseignants de pouvoir centraliser le travail des étudiants et avoir un suivi de l'avancement des TP. Les tâches ainsi centralisés permettront d'avoir une meilleure compréhension des problèmes rencontrés. Les étudiants

⁴⁰ Voir Annexe 5 sur la description du protocole Modbus

disposent d'un compte associé à la plateforme Diweb qui fournira un ensemble d'outils de rédaction afin de rendre leurs travaux (traitement de texte en ligne).

6. Outil de gestion, d'organisation et de création de TP.

- La plateforme est un outil afin de permettre aux enseignants d'organiser les TP, d'associer les TP aux systèmes pédagogiques.

7. Communication et acquisition audio et vidéo.

- La plateforme permet de aux usagers de la plateforme Diweb de pouvoir dialoguer avec une interface d'acquisition vidéo (webcam) et sonore (casque, microphone, téléphone IP). Les élèves peuvent soit dialoguer avec les enseignants soit dialoguer entre les membres d'un même groupe.

8. Gestion et affectation des groupes de TP.

- La plateforme permet aux enseignants de gérer les groupes et les affectations des élèves selon leurs niveaux.

9. Stockage des ressources techniques des systèmes pédagogiques sur la plateforme.

- La plateforme permet de sauvegarder les documents techniques des systèmes pédagogiques sous format électronique et de les consulter à volonté à travers une bibliothèque.

10. Stockage des ressources pédagogiques des systèmes pédagogiques sur la plateforme

- La plateforme permet de sauvegarder les documents pédagogiques des systèmes pédagogiques sous format électronique et de les consulter à volonté à travers une bibliothèque.

11. Stockage des ressources logicielles sur la plateforme.

- La plateforme permet de sauvegarder les logiciels fournis par les systèmes pédagogiques et permettre de les télécharger à volonté à travers une librairie.

12. Cartographie du réseau local.

- La plateforme permet de cartographier le réseau local technique des systèmes pédagogiques, il permet ainsi de connaître les systèmes connectés sur chaque adresse IP (Wifi, PC, Systèmes, etc.) et connaître les ports de communication de chacun. Il est ainsi possible d'accéder à un système sans connaître son adresse IP.

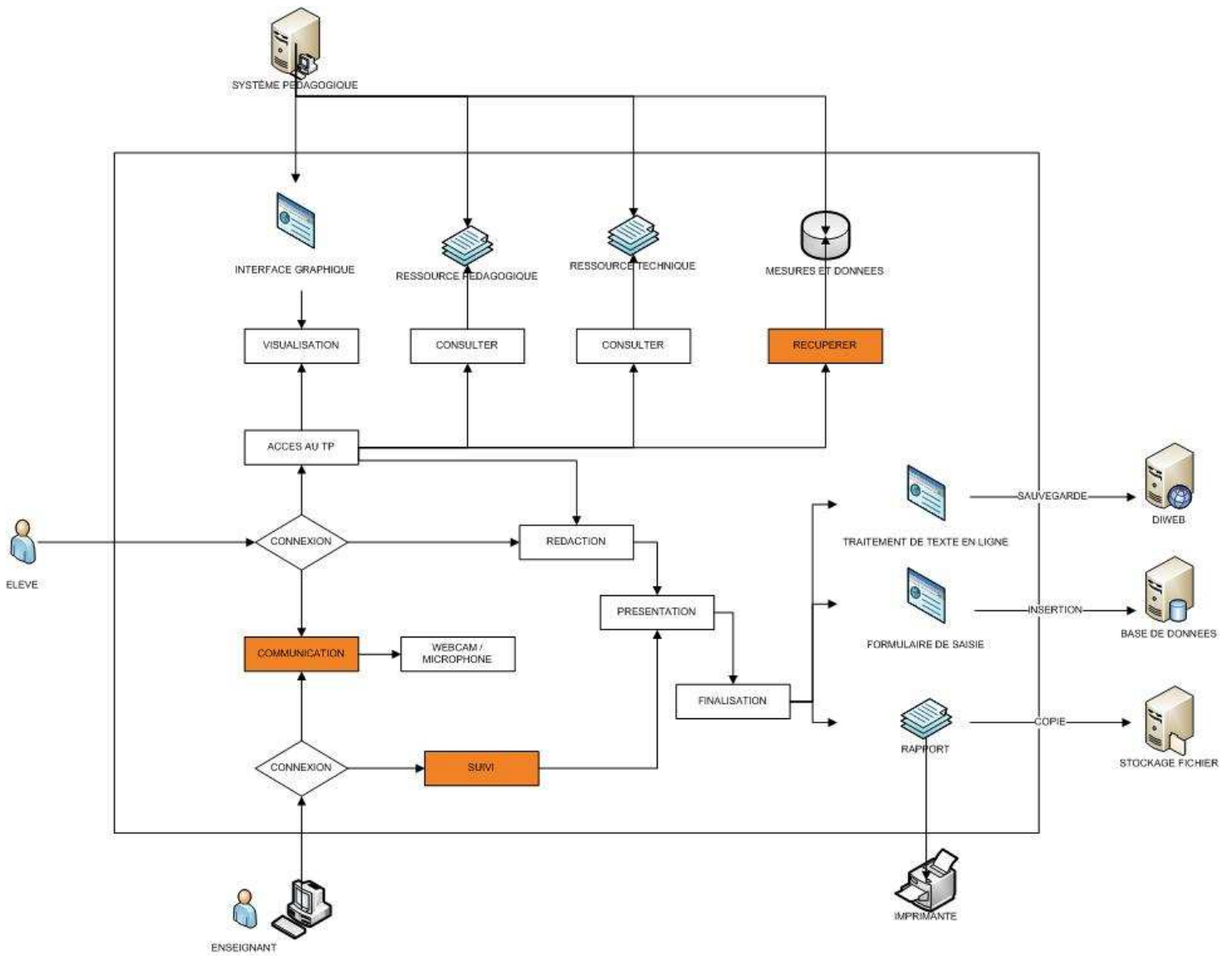


Fig.VIII.1 – Schéma de principe de la plateforme

VIII.2 – Verrous technologiques

Il n'est pas rare de trouver dans toute conception d'application, de rencontrer des obstacles d'ordre technique, ou fonctionnel, en cela il est important dès le départ de pouvoir les cibler afin d'anticiper les problèmes susceptibles d'intervenir lors des phases de développement et ainsi les lever. Les verrous technologiques apparaissent notamment lorsqu'on souhaite rechercher de l'innovation dans un projet et auquel personne n'y a encore fait face.

Les principaux verrous technologiques sur le projet sont :

1. Mutualisation des modules automates-serveur web du réseau informatique, la plateforme ne peut pas dialoguer avec tous les modules automates. Le développement d'interfaces spécifiques pour chaque module est nécessaire.
2. Acquisition des données sur du long terme dans les automates programmables, les modules automates-serveur web ne stocke pas de données à long terme (mémoire interne trop faible).
3. Ergonomie des IHM embarqués dans les systèmes pédagogiques, les IHM embarqués dans les systèmes pédagogiques sont créés à partir de logiciels fournis par le constructeur de l'automate, ces logiciels sont limités par leurs possibilités. Il est possible de développer des interfaces plus riches et dynamiques avec les technologies web actuelles mais il est nécessaire d'obtenir la librairie du constructeur pour comprendre son protocole de communication. Il n'est actuellement pas possible de généraliser et d'automatiser les interfaces, les IHM sont spécifiques et uniques pour chaque système.
4. Dématérialiser le système pédagogique, la plateforme Diweb ne permet pas encore de fournir des interfaces et un environnement pour permettre à l'élève d'être autonome du système (systèmes non complètement autonome et pilotable à distance ou systèmes trop complexes pour être simulés informatiquement).
5. Connaissances de l'internet par les enseignants + formation, il est nécessaire de comprendre l'outil internet et informatique par les utilisateurs pour utiliser la plateforme. Des enseignants à l'heure actuelle ne connaissent pas l'outil informatique. Des formations sont nécessaires pour l'utiliser. Il sera nécessaire pour les enseignants de changer leurs habitudes.
6. Choix des logiciels de l'éducation imposé par l'éducation nationale, ainsi notre plateforme doit s'adapter aux logiciels qu'utilisent le plus souvent l'éducation nationale pour ses enseignants.
7. Accès internet et communication extérieur pour les établissements, la plateforme doit être accessible de l'extérieur, l'établissement doit donc ouvrir une connexion réseau. De plus, le débit doit être suffisamment important pour transférer les paquets de données transmises par les systèmes pédagogiques.
8. Logiciel web, les outils développés avec des technologies web sont limités par leurs possibilités graphiques et traitements temps réels. Une version locale et logiciel bureautique sera plus adaptée pour certains systèmes pédagogiques (modélisation 3D, graphiques temps réels).

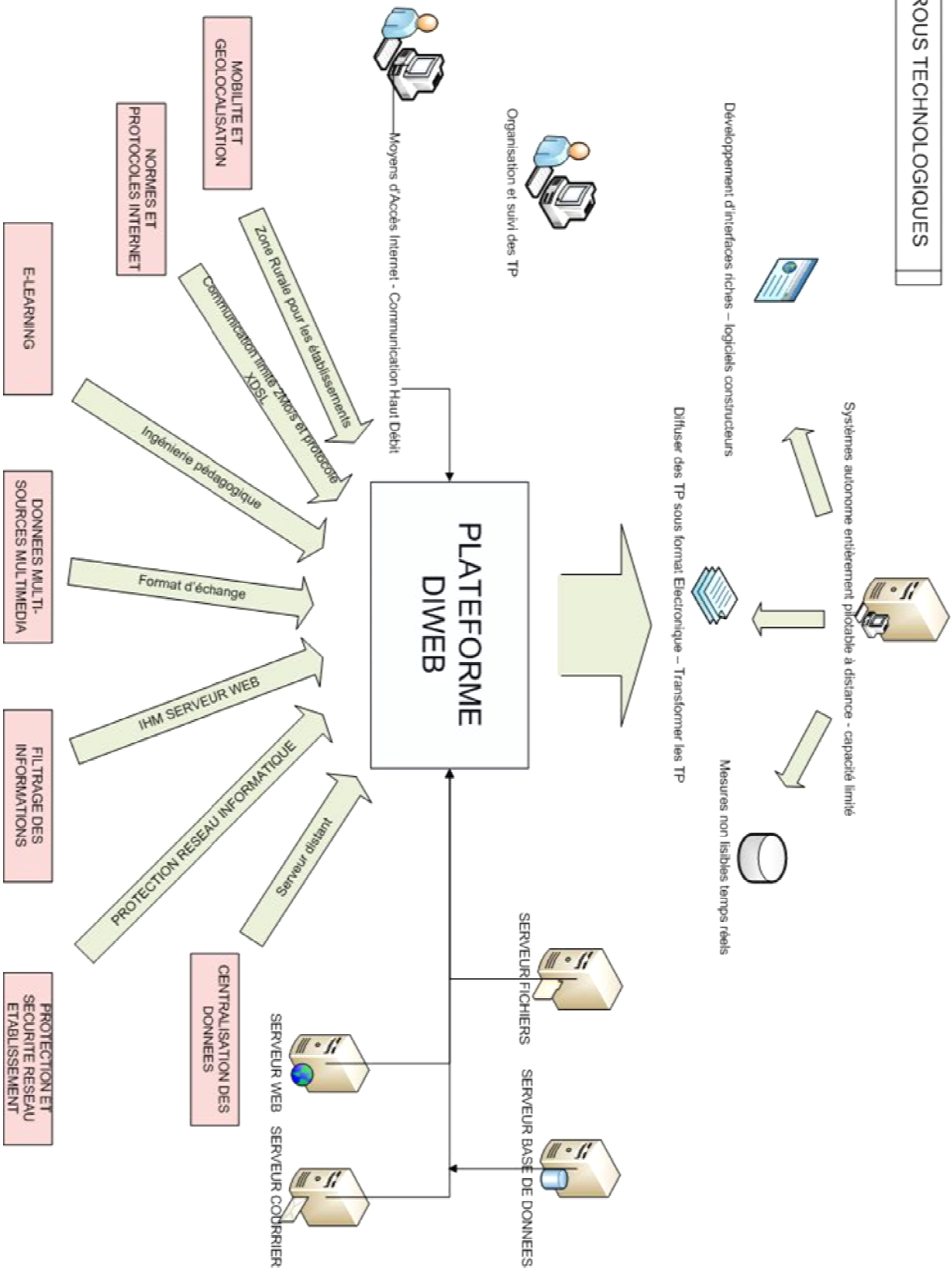


Fig. VIII.2 – Schéma des verrous technologiques

VIII.3 – Synthèse de fonctionnement

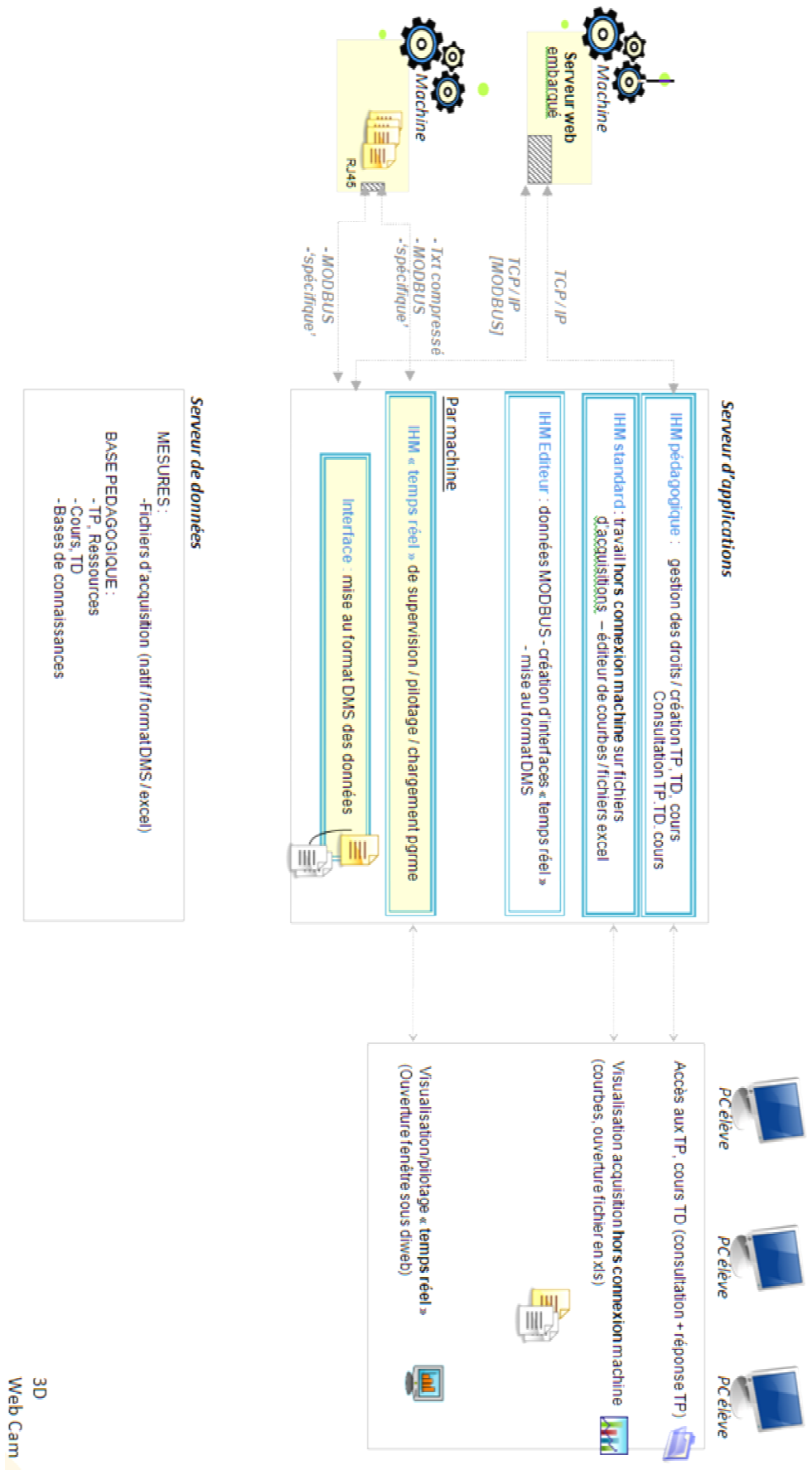


Fig. VIII.3. – Schéma de synthèse de fonctionnement

VIII.4 – Planning prévisionnel

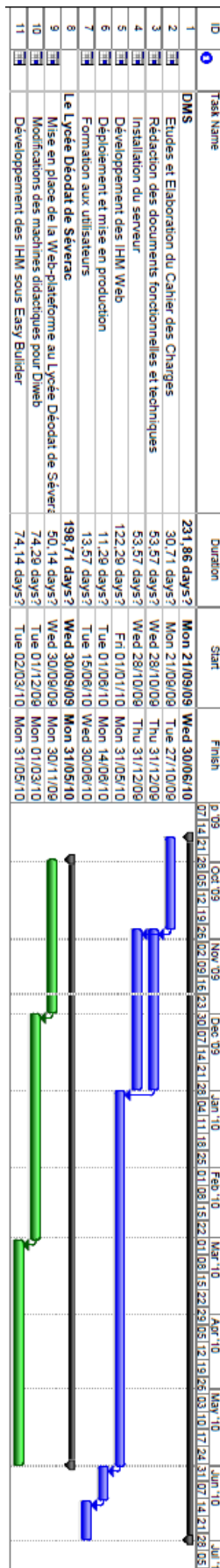


Fig. VIII.4. – Planning prévisionnel

QUATRIEME PARTIE : MODELISATION DU SYSTEME

IX. MODELISATION DU SYSTEME D'INFORMATION

A ce stade du projet et à la suite du cahier des charges techniques, il m'est possible de commencer à penser à la future application à développer. Une des étapes importante dans un projet informatique est la partie modélisation, qui permettra par la suite d'avoir un code source de qualité et de définir une bonne architecture. C'est par ailleurs une pratique conseillé ...

Il n'existe pas de bonnes méthodes de modélisation, chacun présentant tous ses avantages et inconvénients, mais mon choix se portera sur l'UML qui est l'une des méthodes les plus répandus dans ce type de projet et que de par ma formation d'ingénieur il a été la méthode la plus pratiquée.

IX.1 – UML

IX.1.1 – Définition⁴¹

UML (en anglais *Unified Modeling Language* ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la conception « orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

IX.1.2 - Le formalisme d'UML

UML propose 9 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, etc.

IX.1.3 - 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 (c'est le QUOI et le QUI).
- Vue **logique** : c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs (c'est le COMMENT).
- Vue **d'implémentation** : cette vue définit les dépendances entre les modules.

⁴¹ Définition UML issu de Wikipédia : http://fr.wikipedia.org/wiki/Unified_Modeling_Language

- Vue des *processus* : c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.
- Vue de *déploiement* : cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le Où).

Nota : le POURQUOI, n'est pas défini dans UML.

IX.1.4 – 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.

IX.1.4.1 – Diagrammes structurels ou statiques

Les diagrammes structurels ou statiques rassemblent :

- **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 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.

IX.1.4.2 – Diagrammes comportementaux

Les diagrammes comportementaux rassemblent :

- **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.

IX.1.4.3 – Diagrammes d’interaction ou dynamiques

Les diagrammes d’interactions rassemblent :

- **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.

IX.1.4.3 – Les modèles

Un *modèle* est une vision simplifiée d’une fonctionnalité. Il est l’abstraction de comment fonctionne l’application.

La modélisation doit prendre en compte ...

- **Le Pourquoi ?** Pourquoi est ce que le processus métier existe ...
- **Quoi ?** Les activités métiers détaillées qui doivent être accomplies pour atteindre les objectifs assignés au processus.
- **Qui ?** Le système ou la partie responsable pour chacune des activités.
- **Où ?** Le lieu où doivent se dérouler les activités.
- **Quand ?** Le métier est sensible au facteur temps ses processus le sont également.

IX.1.4.4 - Qu’est ce qu’il faut attendre de la modélisation ?

- Communication
 - Utiliser une notation commune
 - Eviter les effets de mode et les notations propriétaires
- Standards
 - Important pour assurer la pérennité des spécifications
 - Mais pas pour le standard
- Outils
 - Doivent fournir une assistance pour tout le cycle de vie
 - Ne pas se limiter à des boîtes à outils (non intégrées)

IX.1.4.5 - Difficultés de la modélisation

- Processus larges et complexes,
- Les acteurs de ces processus n'ont pas toujours une maîtrise de tous les facteurs qui influent sur le processus,
- Divers individus ont différentes visions du processus,
- Les processus sont souvent transversaux (peuvent concerner diverses organisation),
- Absence d'un vocabulaire commun,
- Couvrent plusieurs aspects et à différents niveaux d'abstraction et de détail.

Le rôle des modèles

Un modèle métier est une sorte de plan qui conduit le métier.

- Il permet une meilleure compréhension du processus,
- Il sert de base à l'amélioration du processus,
- Il sert de base à l'aide de la décision, affecte les décisions en fixant les priorités sur les objectifs, sert de base pour obtenir les ressources.
- Il permet d'anticiper les changements et évolution,
- Même s'il n'apporte pas toutes les réponses, il fournit une vision de la stratégie à suivre.

IX.2 – Modélisation métier avec UML

Une architecture centrée sur les processus métier

L'intérêt d'utiliser un langage tel que UML est de :

- Disposer d'un langage visuel permettant la représentation simple des fonctions et leurs relations,
- Utiliser un langage reconnu comme standard,
- Utiliser des techniques éprouvées,
- Utiliser des concepts similaires (tout pourra être rapporté à des objets, relations et interactions entre objets),
- Utiliser la même notation que celle utilisée pour la modélisation du système et resserrer ainsi l'écart entre les deux modélisations.

Tout métier utilise l'information, il est après tout celui qui détermine les besoins d'un système d'information. Il est donc important que celui-ci soit conçu pour répondre à ses besoins. UML a l'avantage de s'adapter facilement à la notion de métier qui est ici celui de l'enseignant.

Modéliser l'architecture métier

Les modèles architecturaux définissent la structure du métier sont primordiaux pour comprendre le métier et ses fonctionnalités.

D'après Eriksson et Penker « une bonne architecture permet au concepteur d'abstraire le métier en divers aspects et vues et de se focaliser sur un seul aspect à la fois »

Les concepts métier :

- Ressource
 - Tout objet humain, matériel, informationnel utilisé ou produit par le métier. Les ressources sont organisées en structures ayant des relations entre elles.
- Processus
 - Sont les activités exécutées par le métier et durant lesquelles l'état des ressources change. Les processus décrivent comment faire les choses et sont dirigées par les règles.
- Objectif
 - Ils décrivent la finalité du métier et peuvent être décomposés en sous objectifs.
- Règle
 - Traduisent la connaissance du métier. Elles définissent des contraintes sur les différents aspects du métier.

IX.3 – Modélisation des objets métiers des fonctionnalités de Diweb

Il s'agit d'exprimer les besoins actuels de l'application, dans le cadre du fonctionnement du système éducatif français d'aujourd'hui. Même si avec le développement rapide des TIC, cela va bouleverser l'organisation des formations et remettre en cause le concept de classe et d'horaires d'enseignement pratiqués dans les établissements scolaires. Le projet Diweb doit prendre en compte ces évolutions à venir.

Description générale des activités pédagogiques actuelles dans les STI

Les programmes et référentiels actuels font une place importante aux activités de « Travaux pratiques » comme moyen d'acquisition de compétences et de savoirs associés. Contrairement aux sciences appliquées, les activités pédagogiques de STI débutent le plus souvent par des TP et dont appel à la démarche inductive. Les horaires d'enseignement sont actuellement constitués par :

- Des heures de cours (enseignement théorique),
- Des heures de TD consacrés à des applications liées aux cours,
- Des heures de TP destinées à l'acquisition de compétences et de connaissances (TP d'acquisition), à la consolidation des acquis (TP de consolidation) et à l'évaluation (TP d'évaluation).

La cohérence de l'enseignement exige une grande proximité entre cours TD et TP. Ces trois approches de la formation trouvent leur cohérence dans la formulation d'objectifs pédagogiques communs, identifiés pour une période donnée (2 à 4 semaines) constituant un module de formation appelé « centre d'intérêt ». Chaque cursus de formation est découpé en « centres d'intérêt ». Chaque activité de TP se rapporte donc à un centre d'intérêt, elle met en relation l'élève avec une situation technique sur laquelle il intervient.

Organisation de l'enseignement

Pour une formation donnée, qui se déroule généralement sur 2 années (BEP, BAC, BTS, CPGE), la couverture du référentiel ou du programme nécessite le découpage du cursus en une vingtaine de centres d'intérêts en moyenne.

Pour un centre d'intérêt donné, abordé en TP par groupe de 28 élèves en moyenne travaillant actuellement en binôme, il faut mettre en œuvre 9 supports de formation (systèmes techniques) simultanément, car $28/3$ (groupes) = 9 à peu près.

Les produits didactiques, supports de TP, doivent couvrir un maximum de centre d'intérêt. Comme, il est difficilement possible, pour un produit, de couvrir la totalité des centres d'intérêt, la plateforme technique doit être équipée, actuellement d'un minimum de 12 systèmes.

Organisation des activités de TP

Chaque TP est identifié par le centre d'intérêt auquel il se rapporte. Tous les TP associés au même centre d'intérêt visent le ou les mêmes objectif(s) pédagogique(s).

Un TP est introduit par une problématique technique réelle.

La résolution du problème impose une démarche décomposée en étapes (ou activités), nécessitant la mise en œuvre du système et l'acquisition de grandeurs physiques. Le TP se termine par la production d'un résultat (solution au problème posé) et par l'identification des acquis. Chaque TP est accompagné d'un corrigé.

Evolutions et perspectives

Les plateformes techniques doivent illustrer des technologies récentes. Le renouvellement périodique des équipements est une nécessité incontournable. Afin d'optimiser les équipements il faut donc envisager de multiplier le nombre d'utilisateurs par système technique.

Le projet Diweb consiste à envisager de faire intervenir plusieurs apprenants simultanément sur chaque support.

Cette organisation optimise l'utilisation des équipements et limite la dispersion des activités ce qui permet à l'enseignant de concentrer davantage ses interventions sur les objectifs pédagogiques visés.

Par ailleurs, le caractère formatif des TP, doit s'adapter à la diversité des temps d'apprentissage. Il faut donc que l'apprenant puisse avoir accès à certains TP en dehors des heures d'enseignement.

Le projet doit permettre également la réalisation de TP en dehors de la plateforme ou du laboratoire.

Quelques pistes possibles pour satisfaire le besoin.

1. **Multiplier le nombre de postes de travail autour de chaque système.** Pour un centre d'intérêt retenu, chaque système est équipé de plusieurs postes permettant, chacun, l'acquisition de données et le pilotage du système. Autour d'un système, chaque élève est confronté à la même démarche de TP. Les temps d'attente d'intervention sur le système sont utilisés à la manipulation de maquettes virtuelles associées au système et utiles à la résolution du problème.
2. **Associer plusieurs systèmes à chaque centre d'intérêt.** Pour un centre d'intérêt retenu, plusieurs systèmes sont accessibles simultanément, depuis le laboratoire ou à distance. Il est possible d'acquérir des données ou de piloter chaque système dans les mêmes conditions que précédemment.

Procédure de mise en œuvre des TP

La conception des TP obéit à de nouvelles contraintes qui éliminent, notamment, l'utilisation de documents papiers. L'accès aux bases de données (ressources techniques et pédagogiques) se fait par liens. Principales étapes de la procédure à mettre en œuvre depuis un poste informatique :

1. Identifier la filière et le niveau (exemple CPGE : PCSI),
2. Visualiser le tableau présentant les centres d'intérêts et les TP,
3. Ouvrir le TP à réaliser : présentation des systèmes disponibles. Soit le système est imposé, soit l'élève a accès à plusieurs systèmes.
4. La page TP comporte les onglets suivants :
5. Fiche pédagogique générique
6. Problème technique
7. Travail demandé (activités, parmi lesquelles l'acquisition et le pilotage),
8. Ressources techniques (dossier technique, modélisation, maquette virtuelle),
9. Ressources pédagogiques (éléments de cours ...)
10. Corrigé

Le projet Diweb doit également permettre l'organisation de la progression de la formation ainsi que le suivi global et individuel de son déroulement.

IX.4 – Représentation UML des objets métiers – Cas d'utilisation

Après avoir bien définis l'aspect métier de l'enseignant et le rôle qu'il a joué. On peut comprendre comment les élèves peuvent interagir avec les systèmes.

IX.4.1 – Cas d'utilisation Application – Elève

L'application élève utilise le serveur principal pour toutes les fonctionnalités dont elle a besoin mis à part pour la lecture du flux vidéo pour lequel elle accède directement au serveur machine.

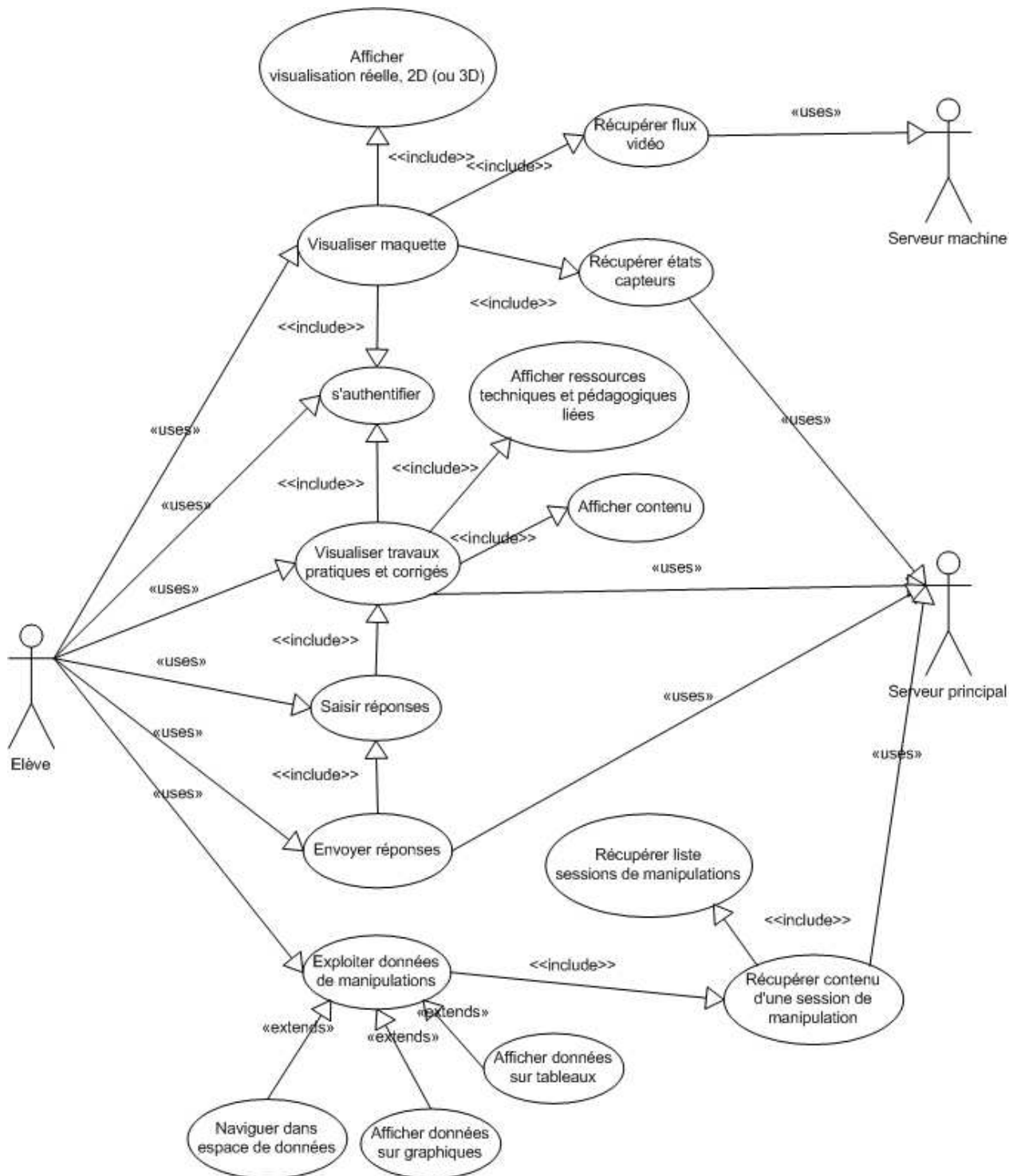


Fig. IX.5.1 – Diagramme de cas d'utilisation Application – Elève

IX.4.2 – Cas d'utilisation Application – Professeur

Tous les cas d'utilisation de l'application élève font partie de la première partie de l'application enseignant. La deuxième partie est la suivante :

- Les deux utilisateurs, élèves et enseignant utilisent chacun leurs interfaces pour accéder à leurs fonctionnalités via le serveur principal à l'exception de l'accès au flux vidéo.

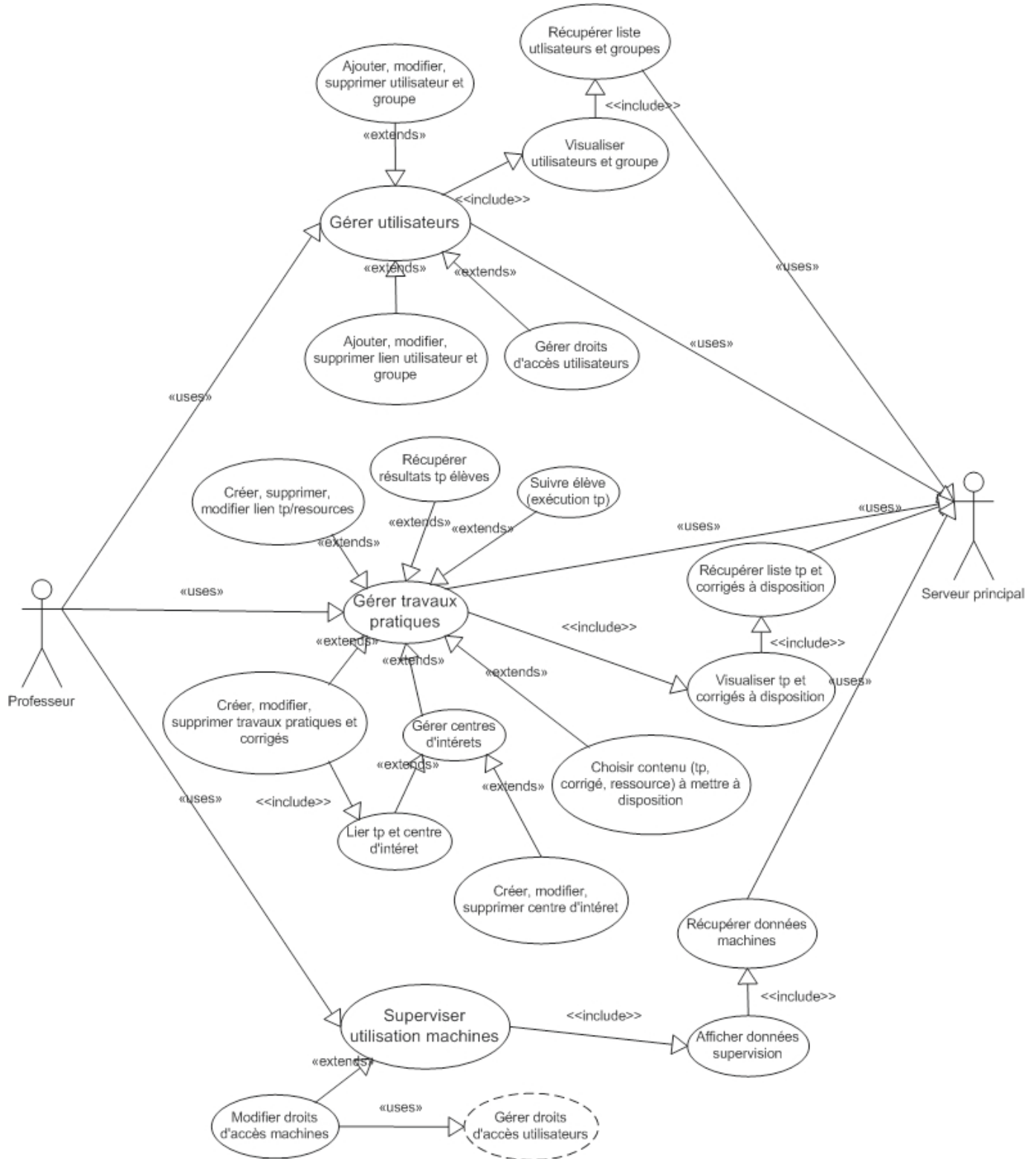


Fig. IX.5.2 – Diagramme de cas d'utilisation Application - Professeur

IX.4.3 – Cas d'utilisation IHM-Serveur

Le serveur machine utilise cette interface pour communiquer avec la machine connectée et pour lire les échantillons de données disponibles sur le disque dur local.

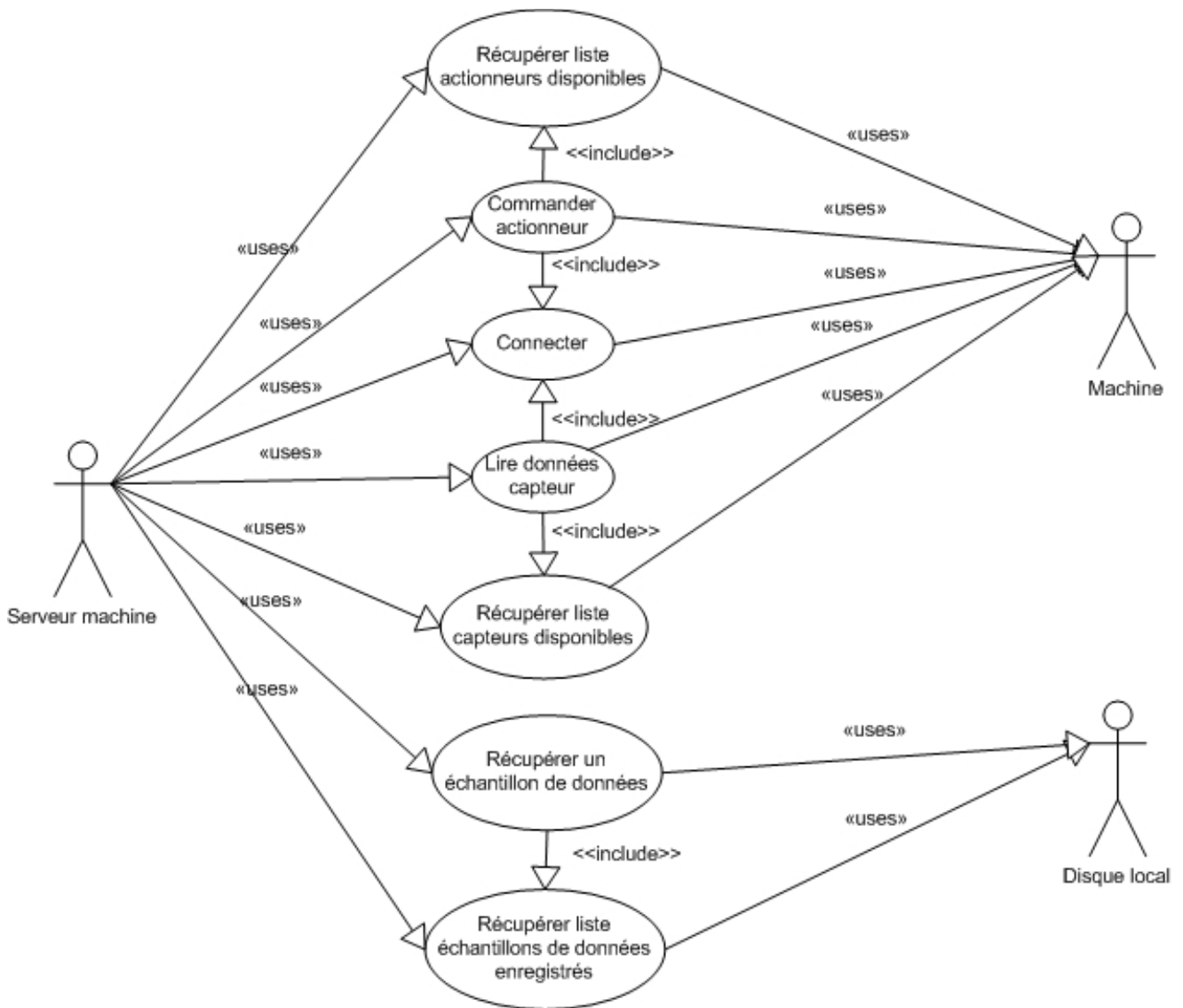


Fig. IX.5.3 – Diagramme de cas d'utilisation IHM-Serveur

Commentaires :

Les serveurs web embarqués disposent de très peu d'espaces de stockage (8 Mo). La difficulté que l'on va rencontrer vient du fait que lors de la connexion avec l'automate via le serveur, il faudra systématiquement récupérer les données résultantes de l'expérimentation.

Auquel cas, ils ne seront pas conservés longtemps par le serveur, en effet celui-ci pourra être écrasé par un autre fichier résultant d'une autre expérimentation.

IX.4.4 – Cas d'utilisation Serveur-Machine

Le serveur principal utilise le serveur machine pour communiquer avec la machine via l'interface machine/serveur et récupérer les échantillons de données.

Le serveur machine permet à l'application élève et enseignant de visualiser le flux vidéo issu de la caméra orientée sur la machine.



Fig. IX.5.4 – Diagramme de cas d'utilisation Serveur-Machine

IX.4.5 – Cas d'utilisation Serveur principal

Les applications élèves et professeur utilisent le serveur principal pour communiquer avec la machine, une fois authentifiées. Elles utilisent, également après authentification, toutes les fonctions d'accès aux données stockées en base de données. Le serveur de mise à jour utilise le serveur principal pour lancer sa procédure.



Fig. IX.5.5 – Diagramme de cas d'utilisation Serveur Principal

IX.5 – Représentation UML des objets métiers – Diagramme de composants

Le diagramme de composants décrit l'organisation ou l'architecture physique et statique du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ils montrent la mise en œuvre physique des modèles de la vue logique avec l'environnement de développement.

Ce diagramme permet de mettre en évidence les dépendances entre les composants (*Qui utilisent quoi ?*). Par ailleurs, les dépendances entre composants permettent notamment d'identifier les contraintes de compilation et de mettre en évidence la réutilisation de composants. Il permet d'accentuer la compréhension du rôle du serveur.

IX.5.1 – Diagramme de composants Application-Elève

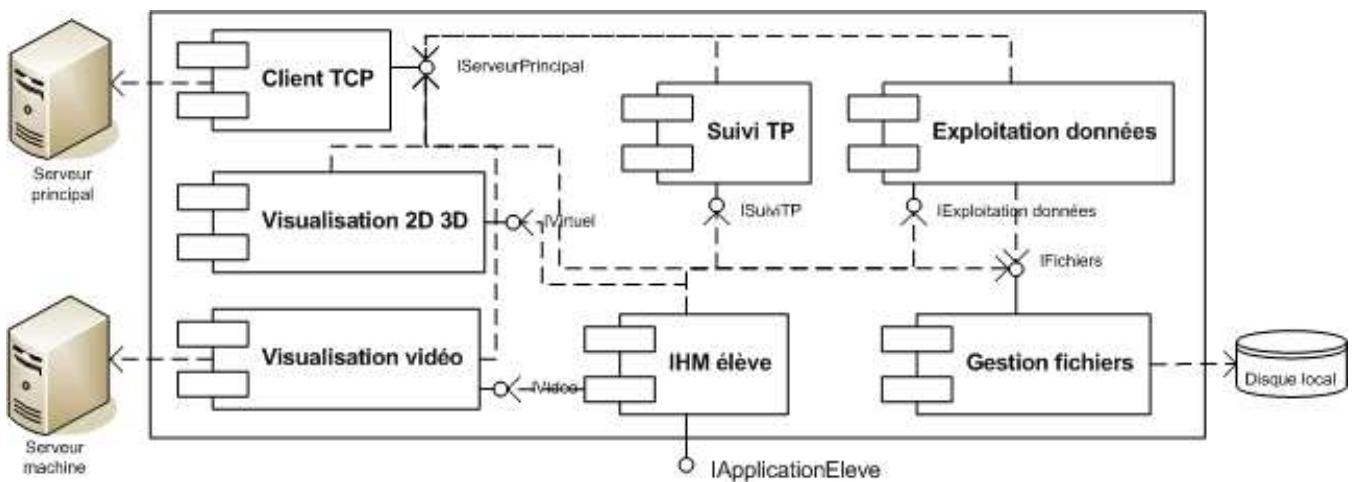


Fig. IX.6.1 – Diagramme de composants Application-Elève

Sur ce schéma, on distingue 7 modules dont un central, le module IHM élève, qui fournit une interface à l'utilisateur élève. Ce module fait appel à tous les autres modules mis à part le client TCP. Le module Visualisation vidéo permet d'afficher le flux vidéo issu de la caméra placée sur la machine. Le mode Visualisation 2D/3D permet l'affichage d'une vision virtuelle du système en rapport avec les données enregistrées dans un fichier, ou avec les données récupérées en temps réel sur la machine.

Le module Suivi TP permet à l'élève de suivre l'exécution d'un TP depuis la lecture de l'énoncé jusqu'à la saisie de résultats, l'envoi de réponses et la consultation de corrigé. Le module Exploitation données permet la visualisation au travers de graphiques et de tableaux de données, récupérées à partir du disque local par l'intermédiaire du module Gestion fichiers, ou du serveur principal par l'intermédiaire du module TCP.

IX.5.2 – Diagramme de composants Application-Professeur

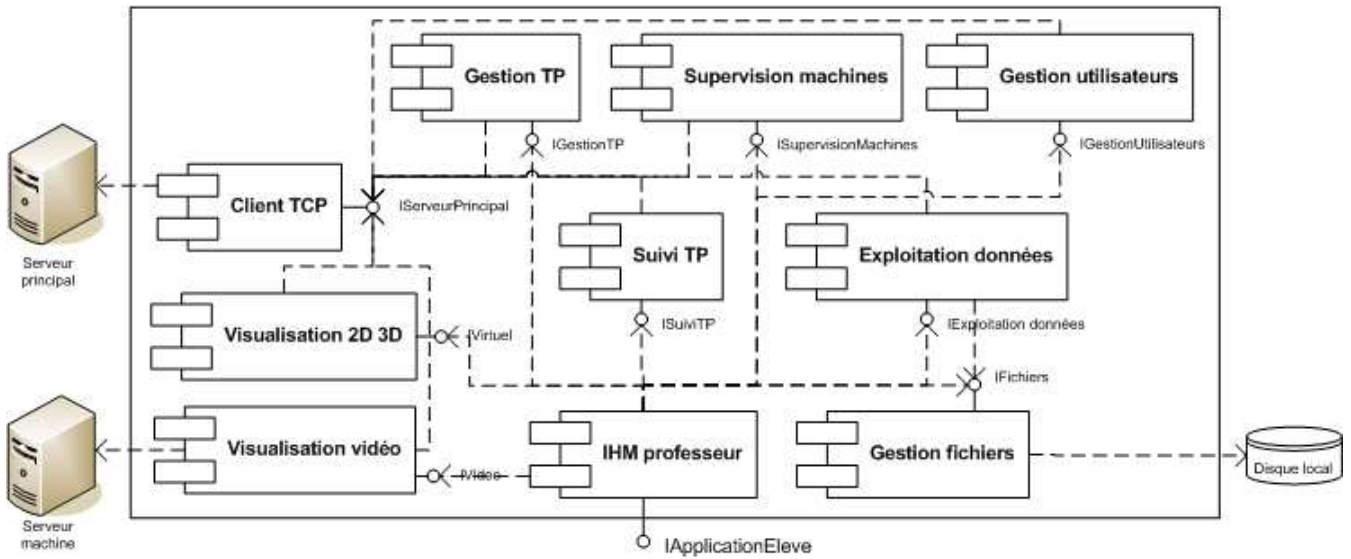


Fig. IX.6.2 – Diagramme de composants Application-Professeur

Seuls les 3 modules de la partie haute du schéma sont ajoutés à l'application professeur. Ces modules servent à l'IHM Professeur pour lui permettre de gérer les TP par le module Gestion TP, et ainsi autoriser la création, modification, suppression et mise à disposition de TP, corrigés et ressources liées sur le serveur principal. Et les deux autres modules Supervision machines et gestion utilisateurs permettent à l'IHM enseignant de superviser l'utilisation des machines et de gérer les utilisateurs et groupes d'utilisateurs.

IX.5.3 – Diagramme de composants Serveur-Machine

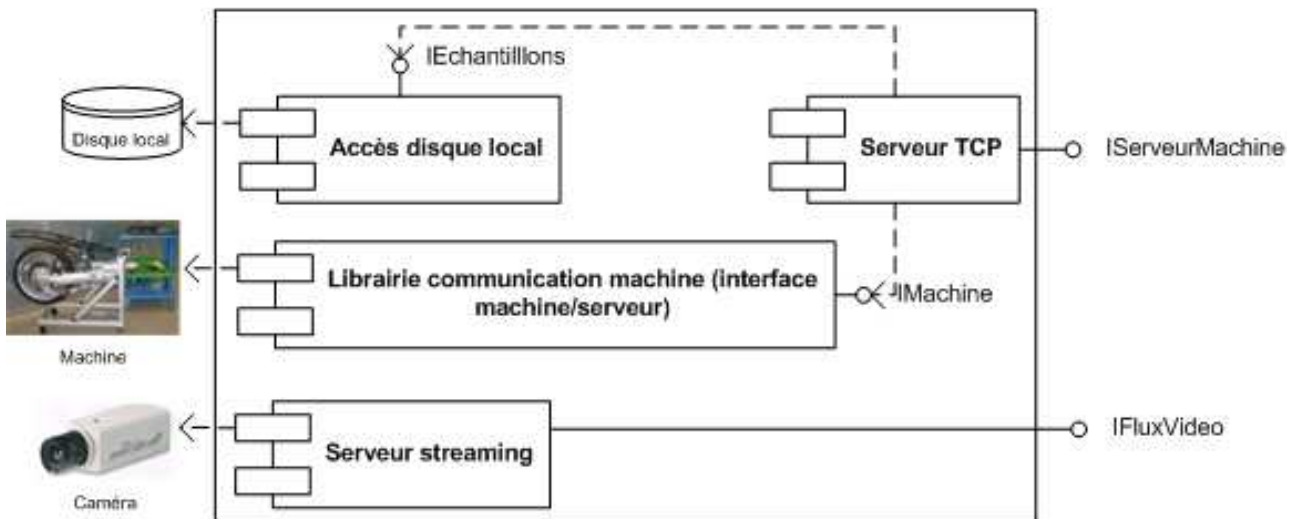


Fig. IX.5.3 – Diagramme de composants Serveur-Machine

Le serveur sera composé de 4 modules principaux dont un central, le serveur TCP, chargé de récupérer les données stockées sur le module d'accès au disque local et de communiquer avec la machine par l'intermédiaire de la librairie de communication machine. Ensuite un module sera chargé de mettre à disposition un flux vidéo issu d'une caméra connectée au serveur.

IX.5.4 – Diagramme de composants Serveur Principal

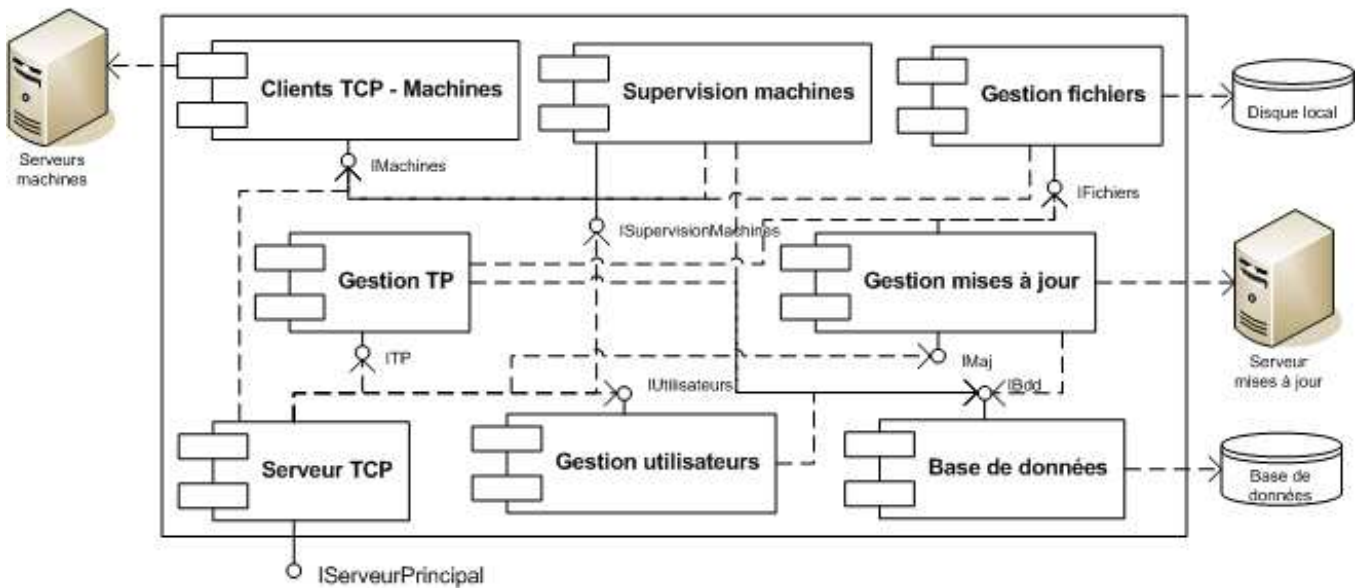


Fig. IX.5.4 – Diagramme de composants Serveur Principal

Sur ce schéma, on distingue 8 modules dont un central qui fournit la fonction de sortie du système, le serveur TCP, chargé de communiquer avec les machines par l'intermédiaire du module Clients TCP-machines, de fournir un service de gestion des utilisateurs par l'intermédiaire du module Gestion utilisateurs, de permettre la gestion des travaux pratiques pour les deux types d'applications par l'intermédiaire du module Gestion TP, et de permettre la gestion des mises à jour, ainsi que la supervision des machines par l'intermédiaire du module Supervision machines.

Le module Gestion TP devra nécessairement accéder à la base de données pour accéder aux données de TP (caractéristiques, énoncé, corrigé et ressources) par l'intermédiaire du module Base de données, ainsi qu'aux fichiers stockés relatifs aux TP (manipulations) par l'intermédiaire du module Gestion fichiers.

Ce module Gestion fichiers accédera aux fichiers stockés sur la machine locale et distante via respectivement le disque local et le module Clients TCP-Machines. Le module Gestion utilisateurs devra accéder à la base de données par l'intermédiaire du module Base de données, pour obtenir toutes les informations sur les utilisateurs et groupes d'utilisateurs.

Le module Supervision machine fera appel au module Clients TCP-machines pour obtenir les informations machines ainsi qu'au module Base de données pour obtenir des données calculées à partir de données enregistrées. Enfin le module Gestion mises à jour accédera aux fichiers stockés par l'intermédiaire du module Gestion fichiers pour effectuer les mises à jour de fichiers, et au module Bases de données.

CINQUIEME PARTIE : MISE EN PLACE

X. MISE EN PLACE ET ARCHITECTURE DU RESEAU INFORMATIQUE

En amont de la phase de développement, il est important que l'infrastructure réseau soit correctement mise en place afin d'effectuer la communication aux systèmes présents dans l'atelier. Cette communication doit par ailleurs être disponible depuis l'extérieur hors de la zone de TP.

Afin de rendre cela possible il est nécessaire d'installer au sein du réseau de l'atelier un serveur qui aura pour rôle de filtrer et de surveiller les demandes de connexion aux systèmes depuis l'extérieur ou intérieur par les élèves. Ce type de connexion se base sur une architecture matérielle dite de type « 3 tiers » car 3 éléments interviennent lors de cette action (PC Elève, PC Machine et Serveur). On peut considérer dans ce type d'architecture que le serveur machine est considérée comme une base de données.

Afin d'avoir une meilleure compréhension, je vais détailler la notion d'architecture 3-tiers.

X.1 – Définition du modèle d'architecture matérielle 3-tiers

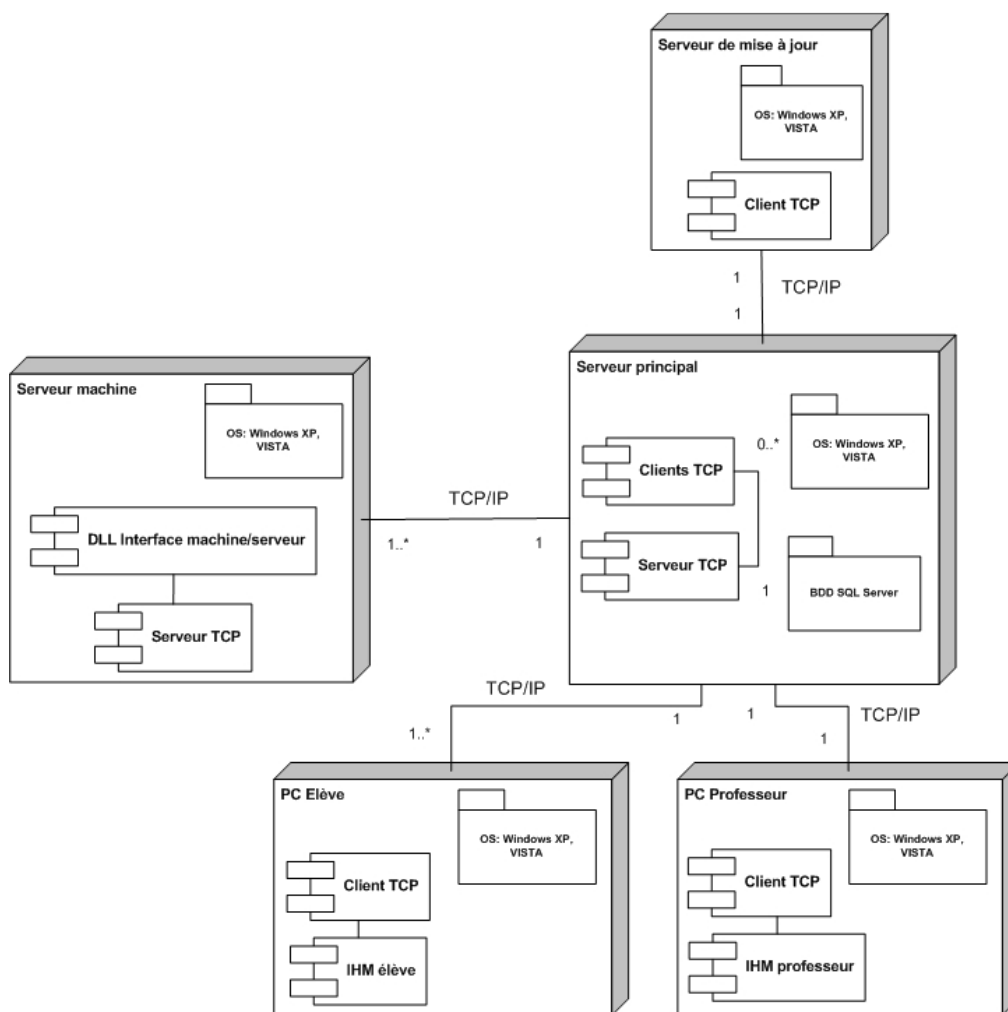


Fig. X.1.a - Schéma de l'architecture logicielle de la plateforme

Commentaires :

Sur ce diagramme, on retrouve chacun des éléments de l'architecture matérielle sur lequel sont ajoutés les éléments logiciels nécessaires au fonctionnement du système. Sur le serveur machine, on distingue la DLL de communication machine, spécifique à chaque machine vendue par la société

DMS et dont les fonctions de sortie sont génériques à toutes les machines, ainsi que le serveur TCP nécessaire à la communication avec le serveur principal.

Sur le serveur principal seront nécessaire un client TCP par machine connectée afin de s'assurer une connexion simultanée, ainsi qu'avec le serveur de mise à jour.

Il sera placé sur ce serveur, la base de données, où seront enregistrées toutes les données du système. Chacun des postes de travail élève et professeur disposera d'un client TCP pour permettra la communication avec le serveur principal ainsi que l'interface utilisateur spécifique.

Quant au serveur de mise à jour, il disposera uniquement d'un client TCP pour mettre à jour le contenu sur le serveur principal. Chaque élément évoluera dans un environnement Windows.

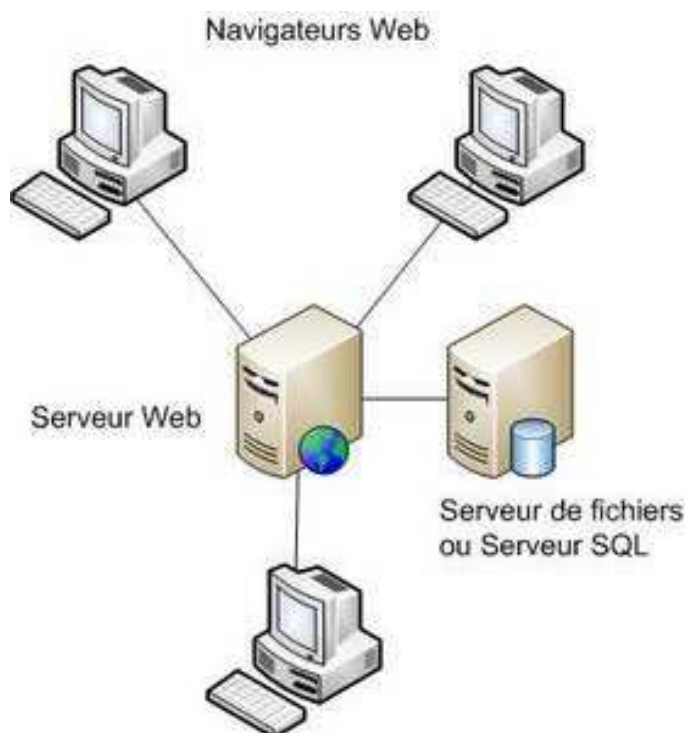


Fig. X.1. b - Schéma d'architecture matérielle

Commentaires :

Le système, dont la partie centrale est le serveur principal où sont stockées toutes les ressources techniques et pédagogiques ainsi que les données utilisateur, s'articule sur deux applicatifs (élèves et enseignants), et permet la récupération de données situées sur chacun des serveurs dédiés aux machines. L'architecture, telle qu'elle est envisagée permettra la visualisation d'une machine située en dehors du réseau local, et la mise à jour du système par le biais d'un serveur de mise à jour distant.

X.2 Installation et mise en place du réseau de la zone Electrotechnique

Des travaux ont été effectués dans l'atelier afin d'étendre les possibilités de communications des systèmes et d'y ajouter des matériels tels que PC, caméra, serveurs et autres. Ces installations ont été réalisées en collaboration avec des étudiants en BTS Electrotechnique dont les principales ont été les suivantes :

- Installation de nouvelles prises Ethernet afin d'interconnecter plus de systèmes,
- Câbler les caméras IP sur le réseau,
- Installer et configurer les nouveaux systèmes sur le réseau,
- Installer de nouveaux Switch et agrandir la baie de brassage,
- Installer et préparer les postes informatiques ainsi que tous les outils nécessaires pour la connexion avec les automates,
- Aménager le nouvel environnement de la zone pour organiser les systèmes et le mobilier,



Fig. X.2.a – Plan d'aménagement de la zone Electrotechnique

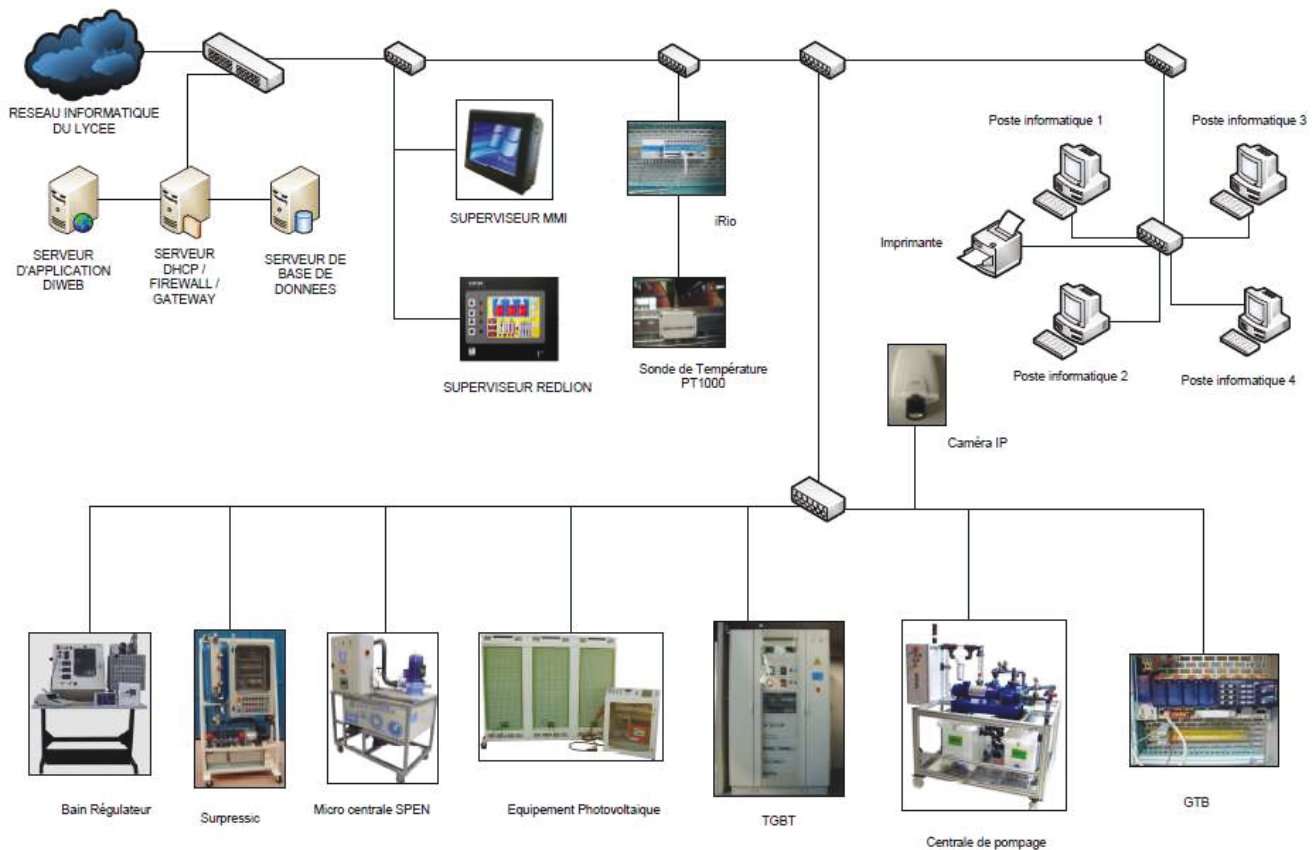


Fig. X.2.b – Schéma du réseau local de l'atelier BTS Electrotechnique

X.2 – Configuration des adresses IP

Mes tâches ont été diverses mais surtout demandant des compétences et notions avancées en informatique, Pour cela il me fallait réorganiser tout le réseau de départ.

Parmi ces tâches primaires ont retrouvé notamment :

- La configuration des adresses IP sur chaque système communicable et les postes informatiques afin que ceux-ci puissent communiquer ensemble,
- L'installation des serveurs informatiques afin de sécuriser la zone et préparer la phase de développement.
- L'aide aux étudiants pour le développement de nouvelles IHM embarqués sur les systèmes,
- L'ajout du terminal MMI pour la supervision des systèmes du réseau,
- L'ajout d'une connexion internet afin que les étudiants puissent récolter de l'information sur le web (il sera donc nécessaire de définir des règles d'accès vers l'extérieur),
- L'installation et le paramétrage d'une caméra pilotable pour un retour vidéo des systèmes communiquant.

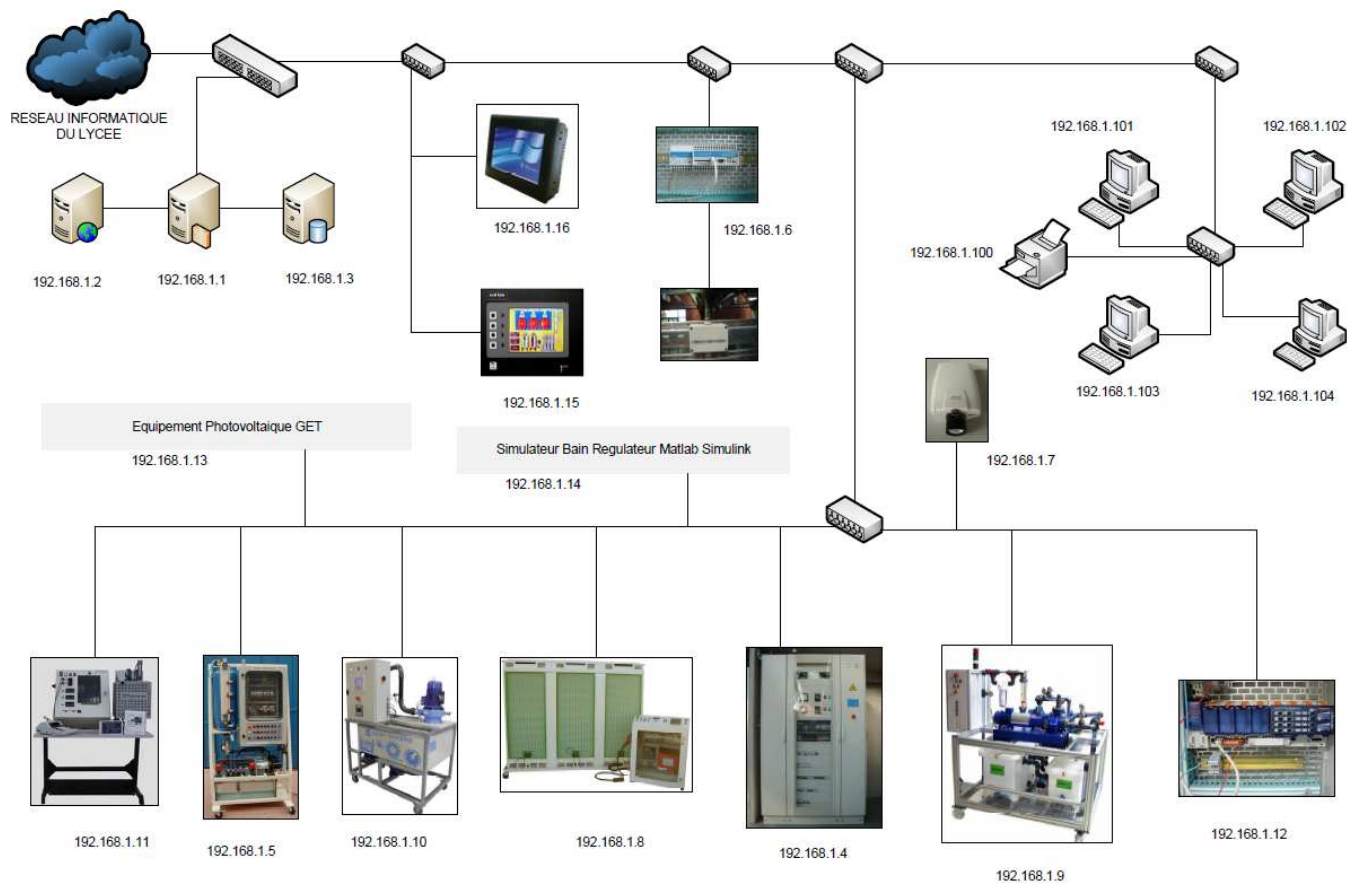


Fig. X.2.a – Schéma du réseau local de l'atelier BTS Electrotechnique avec les adresses IP

SYSTEMES	URL	IP
Serveur DHCP	http:///srv_dhcp	192.168.1.1
Serveur d'application DiWeb	http://srv_diweb	192.168.1.2
Serveur de bases de données	http://srv_bdd	192.168.1.3
TGBT	http://tgbt	192.168.1.4
SURPRESSICC	http://surpressic	192.168.1.5
iRio	http://irio	192.168.1.6
Caméra IP	http://camera	192.168.1.7
Equipement PV	http://photovoltaïque	192.168.1.8
Centrale de pompage	http://pompage	192.168.1.9
Micro-centrale SPEN	http://spen	192.168.1.10
Bain régulateur GTI	http://bain_regulateur	192.168.1.11
Equipement PV GET	http://photovoltaïque_get	192.168.1.12
Simulateur bain régulateur	http://simul_bain_regulateur	192.168.1.13
Superviseur RedLion G308C	http://redlion	192.168.1.14
Terminal MMI 4156F	http://superviseur	192.168.1.15
Caméra IP électrique	http://camera_elec	192.168.1.21

Fig. X.2.b – Récapitulatifs des adresses IP distribuées

Avec la configuration des adresses ainsi que leurs DNS, il est dorénavant possible de dialoguer avec ces systèmes au sein du réseau informatique de l'atelier. En cela, j'ai réservé des plages d'adresses que je pourrais par la suite configurer au serveur afin qu'il puisse les attribuer de façon automatique. Par exemple, la plage d'IP 192.168.1.100 – 192.168.1.140 sont réservés pour les postes informatiques connectés.

10.2.1 – Installation d'une borne WIFI

Par un souci de confort et afin d'élargir les possibilités des connexions, j'ai eu l'idée d'ajouter au sein de la zone Electrotechnique une connexion WIFI qui permettra aux étudiants et aux enseignants de pouvoir s'y connecter via leurs propres matériels informatiques (smartphone, ordinateur portables, pda etc...).



Ainsi les étudiants auront une liberté de mouvements lors de leurs séances de TP. Bien sûr, cette connexion est sécurisée et devra faire l'objet d'une demande auprès de l'enseignant pour pouvoir s'y authentifier. L'authentification par ailleurs requiert de fournir l'adresse mac du matériel. Celui-ci étant configurable via son interface de gestion sur l'adresse IP 192.168.1.250.



L'interface de gestion permet de configurer d'effectuer les configurations et paramétrages sur l'attribution des adresses IP dynamiques, le filtrage MAC des appareils et l'accès à internet (celui-ci sera configuré par la suite). Le mode Wifi configuré ici est WPA (TKIP + AES), qui représente le mode de sécurité maximal. Mais le débit en est réduit.

The screenshot displays the Netgear SmartWizard configuration interface for a RangeMax™ NEXT Wireless Router (model WNR854T). The interface is divided into three main sections:

- Left Navigation Panel:** Contains a tree view with categories: Setup Wizard, Setup (Basic Settings, Wireless Settings, Content Filtering, Logs, Block Sites, Block Services, Schedule, E-mail), Maintenance (Router Status, Attached Devices, Backup Settings, Set Password, Router Upgrade), and Advanced (Wireless Settings, Port Forwarding / Port Triggering, WAN Setup, LAN IP Setup, Dynamic DNS, Static Routes, Remote Management, UPnP).
- Basic Settings Page:**
 - Does Your Internet Connection Require A Login?:** Radio buttons for Yes and No (No is selected).
 - Account Name (If Required):** Text input field.
 - Domain Name (If Required):** Text input field.
 - Internet IP Address:** Radio buttons for Get Dynamically From ISP (selected) and Use Static IP Address. Below are input fields for IP Address, IP Subnet Mask, and Gateway IP Address, each with four digit boxes.
 - Domain Name Server (DNS) Address:** Radio buttons for Get Automatically From ISP (selected) and Use These DNS Servers. Below are input fields for Primary DNS and Secondary DNS, each with four digit boxes.
 - Router MAC Address:** Radio buttons for Use Default Address (selected) and Use Computer MAC Address.
- Help and Documentation Panel:**
 - Text: "The Basic Settings pages allow you to configure, upgrade and check the status of your NETGEAR Wireless Router." and "Click an item in the leftmost column. The current settings or information for that area appear in the center column."
 - Text: "Helpful information related to the selected Settings page appears in this column. If you are using Internet Explorer, you may click an item in the center column to jump directly to the related help section, otherwise, scroll down until you reach it."
 - Text: "For the most current documentation, go to <http://documentation.netgear.com/wnr854t>"
 - Basic Settings Help:**
 - Does Your Internet Connection Require A Login?:** "Select this option based on the type of account you have with your ISP. If you need to enter login information every time you connect to the Internet or you have a PPPoE account with your ISP, select **Yes**. Otherwise, select **No**."
 - Note:** "If you are setting up the router for the first time, the default settings may work for you with no changes."
 - Note:** "If you have installed PPP software such as WinPoET (from Earthlink) or Eternet (from PacBell), then you have..."

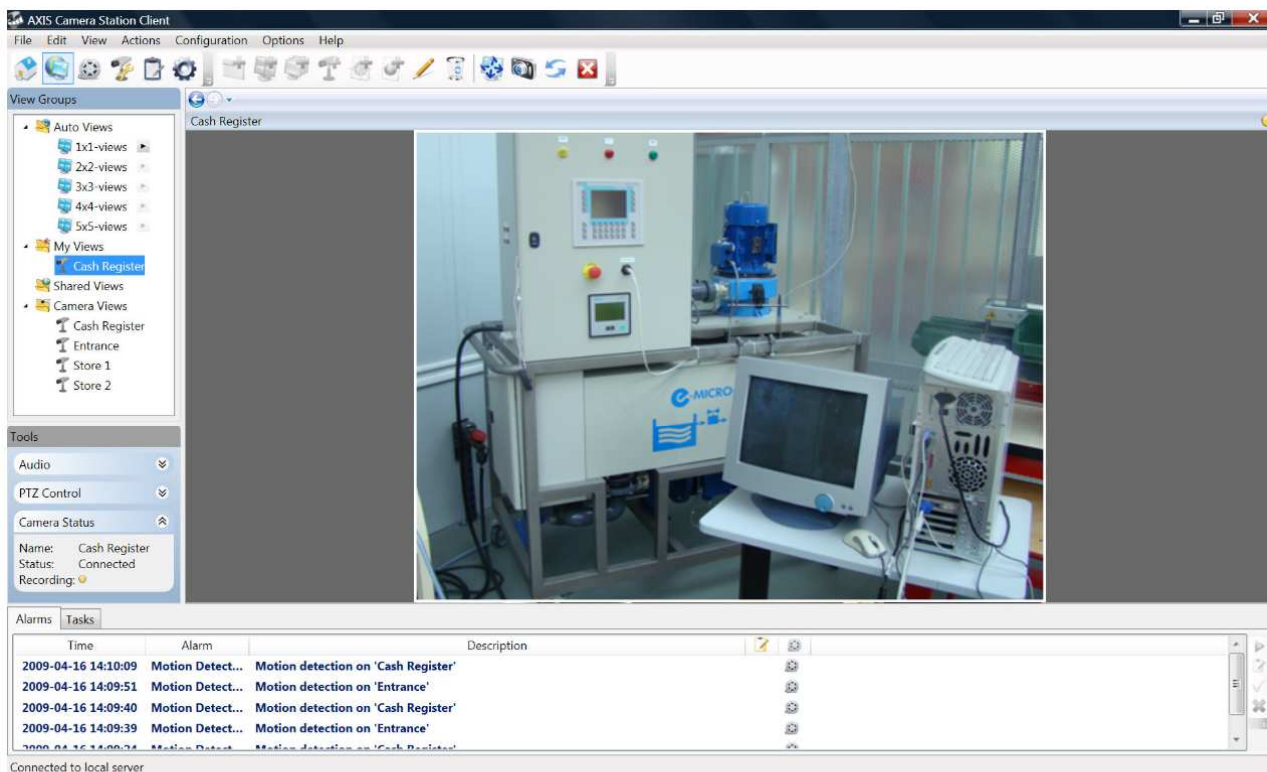
10.2.2 – Installation d'une caméra pivotable

Nécessaire dans la réalisation du projet afin d'avoir un retour vidéo, l'atelier s'est doté d'une caméra IP pivotable donnant une visibilité sur l'ensemble de l'atelier est notamment sur les systèmes, par ailleurs les systèmes communicants ont d'ailleurs été aménagé de façon à ce que ceux-ci soit facilement zoomable et visible de la caméra. Ce qui donne une expérience plus enrichissantes et apporte aussi plus d'intérêts lorsque l'on réalise des TP à distance. Cette intégré notamment une option de capture photo et vidéo de type Motion-JPEG ou MPEG-4 qui sont les formats les plus reconnus.

De plus, la transmission audio est bidirectionnelle qui permettra d'avoir des échanges orales en temps réels avec un professeur. L'intérêt technique non négligeable est que le flux vidéo de cette caméra peut-être récupéré et réutilisé dans notre future plateforme.



L'interface d'administration de la caméra permet des possibilités assez larges dans l'utilisation de celle-ci comme les fonctionnalités de mouvement vidéo intégrée et la gestion avancée des événements et de commander celle-ci à distance.



10.2.3 – Installation du superviseur MMI

Le MMI-4156F est un panel PC industriel idéal pour une utilisation dans le domaine de l'industrie, l'agroalimentaire, le tertiaire, la chimie et autres applications spécifiques de Contrôle/Commande.

L'atelier a fait l'acquisition de cette matériel dans l'optique d'élargir les logiciels industriels de supervision pour les étudiants et permettre ainsi aux étudiants de développer les synoptiques des systèmes de l'atelier plus facilement.

La plupart des systèmes communicants sont équipés de serveur web ce qui rend possible le stockage des IHM pour le piloter à distance, d'autres systèmes comme le bain régulateur et l'axe XY n'ont s'en pas équiper mais dispose d'une passerelle, la différence est que avec une passerelle on ne dialogue avec les systèmes que par l'intermédiaire de logiciels de supervision. Cela évite de modifier la partie opérative des systèmes pour y inclure des serveurs qui sont coûteux.



Caractéristiques techniques du Superviseur :

PORTS	
Ethernet	1 port 10/100 Mbps
USB	2 ports USB 2.0 (2 ports supplémentaires) *
VGA	1 port VGA * - Intel 855GME
Série	COM1 : RS232 COM2 : RS232 ou RS422 ou RS485 * (ports supplémentaires) *
Parallèle	1 port SPP/EPP/ECP *
PS/2	2 ports Clavier/Souris
E/S Audio	1 MIC, 1 line-in, 1 line-out
Compact Flash	1 port CF
Extension	Non

CONFIGURATION	
Processeur	Celeron M 1 GHz à 1.5 GHz Pentium M 1.6 GHz à 1.8 GHz
Mémoire	256 Mo SDRAM jusqu'à 1 Go
Stockage (DD)	2.5" slim DD anti vibration 40, 60, 80 Go (à définir) 1 port Compact Flash
OS	Windows CE 4.2, 5.0, XP Embedded, XP PRO, VISTA
Lecteurs	Non
Connexions	Wifi, Bluetooth, GPRS

X.3 – Liaison depuis l'extérieur vers le serveur Diweb

Dû aux contraintes de disponibilités et aux horaires d'accès à la zone, il m'était impossible d'avoir une présence permanente afin d'effectuer mes tâches de développement de l'application. La nécessité d'avoir accès au serveur depuis l'extérieur me permettait à la fois de pouvoir travailler et agir à distance sur les serveurs web et de sécuriser la zone.

Le niveau des équipements informatiques dans le Lycée est très important et constitue l'une des grandes difficultés à surmonter dans la mise en œuvre du projet. Il dispose d'une structure disposant de sept cents postes en réseaux s'appuyant sur des V.L.A.N⁴² et mettant en œuvre des serveurs Windows 2003 et Linux avec plusieurs contrôleurs de domaines, proxy, pare-feu ...

Ce parc informatique se compose, comme cela a déjà été mentionné, de deux entités : l'administration et la pédagogie. La pédagogie met à disposition des services informatiques à usage pédagogique. Ils viennent apporter un soutien à l'enseignement des professeurs, par le biais de logiciels éducatifs et de mise à disposition de ressources d'informations disponibles sur Internet. C'est aussi un instrument indispensable dans certaines sections spécialisées dans le domaine comme entre autres les B.T.S. informatique ou encore les B.E.P secrétariats.

Ces outils deviennent plus variés encore selon la ou les spécialités de l'établissement (Lycée d'enseignement professionnel par exemple) ou encore selon la politique et les moyens de ce dernier. Ainsi n'est-il pas rare de rencontrer des besoins de C.A.O ou de montage vidéo. De façon générale l'informatique pédagogique regroupe tous les services qui peuvent être apportés par l'informatique dans une salle de classe quelle que soit son domaine d'enseignement.

D'un point de vue réseaux informatiques le plus souvent ces deux communautés sont dissociées. En regard des particularités de ces deux entités, on peut constater que nous mettons en relation une structure gérant des informations sensibles (l'administration), et une autre qui met à disposition des stations de travail pour des élèves. L'architecture réseau du Lycée met en œuvre la technologie des V.L.A.N. On retrouve trois réseaux virtuels : un administratif, un pédagogique, et un de supervision.

X.3.1 – Configuration réseau au sein du Lycée

Un des problèmes que j'ai rencontré lors de la mise en place du réseau Electrotechnique au sein de la zone⁴³ fut que celle-ci est difficilement accessible si on passe par le réseau du Lycée. Avec tous ses blocages et redirections de ports. J'ai eu beaucoup de mal à pouvoir rediriger la connexion extérieur vers la zone Electrotechnique.

Par la suite, il a été décidé que notre réseau serait indépendamment de celui du Lycée ce qui me permettrait d'obtenir un accès total sur l'ensemble des ports de communication. L'inconvénient est que notre zone ne sera pas accessible pour les autres postes du Lycée.

La difficulté d'avoir une liaison externe présentait des failles de sécurité que n'appréciait guère l'administrateur du Lycée. Cela a donc pris beaucoup de temps pour en avoir l'autorisation. Cette connexion a toutefois permis par la suite permis à la zone Electrotechnique de fournir un accès à Internet. Afin de rendre cet accès possible, il a été demandé à la mission TICE de fournir une adresse IP fixe qui ciblerait à un routeur via un port de communication du pare-feu de l'établissement. Suite à de nombreux

⁴² *Virtual Local Area Network*, technologie permettant de créer plusieurs réseaux virtuels au sein d'un seul et unique réseau local physique

⁴³ Voir article [9] sur les réseaux industriels

problèmes de connexion, la mission a ouvert l'ensemble des ports de communication vers ce routeur, j'ai donc été chargé de veiller à son bon fonctionnement et de mettre en place les règles de sécurité.

Ce routeur permettrait notamment permettait de filtrer les entrées et sorties de la zone Electrotechnique et de définir des règles de routage et de redirections de ports. L'adresse qui m'a été confié pour une utilisation extérieure est <http://194.254.30.90>.

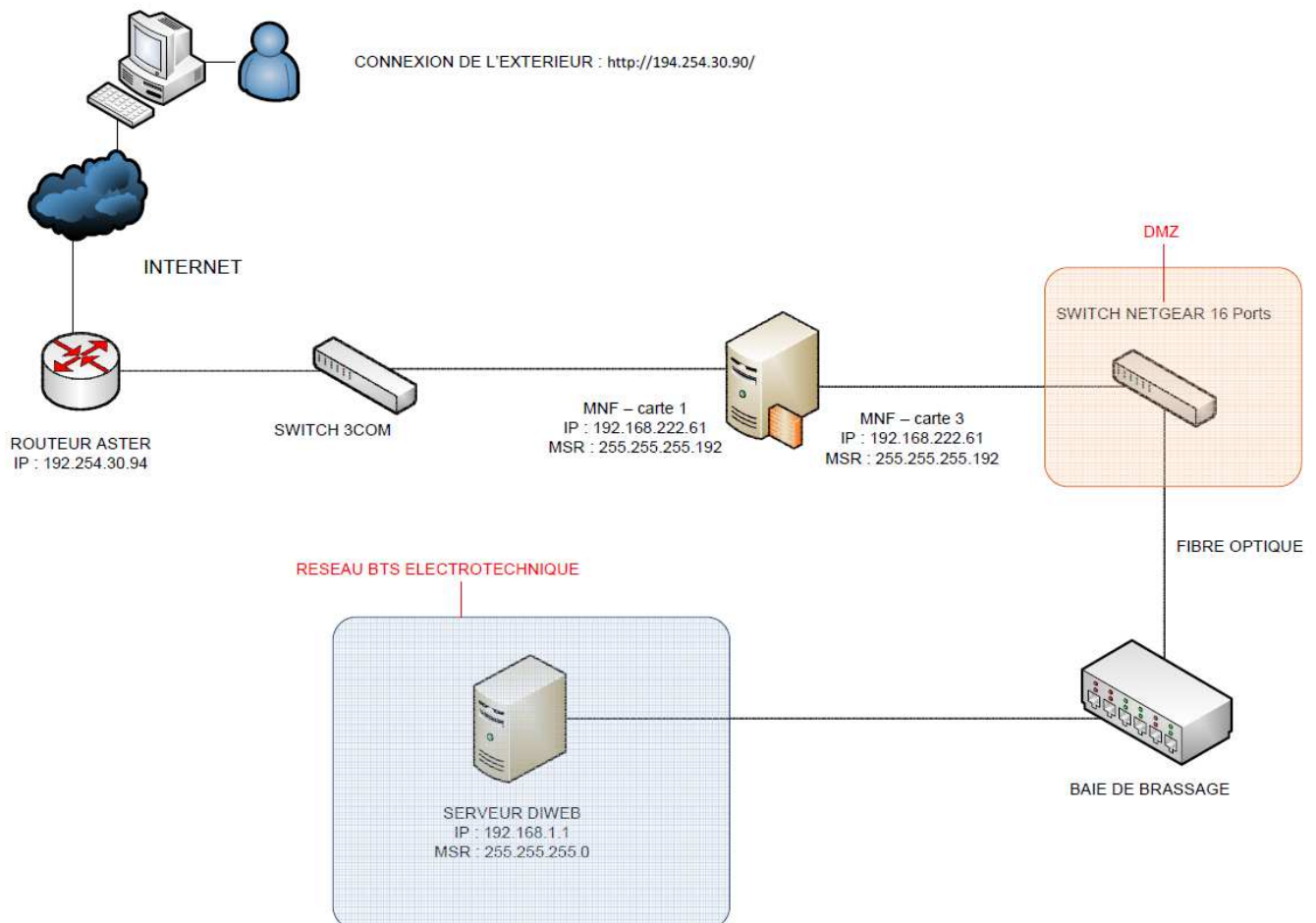


Fig. X.3.a – Ouverture de la connexion vers l'extérieur

X.3.1 Rôle du routeur

Le routeur est un élément intermédiaire dans un réseau informatique assurant le routage des paquets indépendants. Son rôle est de faire transiter des paquets d'une interface réseau vers une autre au mieux, selon un ensemble de règles formant la table de routage.

Comme la zone dispose de plusieurs ordinateurs, je dois configurer le routeur afin de ne pas rencontrer de conflit d'adresse IP si jamais une connexion simultanée de plusieurs ordinateurs. Il est préférable d'activer le DHCP⁴⁴ afin de simplifier la connexion à Internet. C'est un équipement de couche 3 au modèle OSI. Il ne doit pas être confondu avec un commutateur (couche 2).

La fonction de routage traite les adresses IP en fonction de leur adresse réseau définie par le masque de sous-réseaux et les redirige selon l'algorithme de routage et sa table associée. Ces protocoles de routage

⁴⁴ Dynamic Host Configuration Protocol (voir Glossaire)

sont mis en place selon l'architecture de notre réseau et les liens de communication inter sites et inter réseaux. Ils utilisent le protocole TCP/IP.

Ici notre routeur sert d'intermédiaire entre Internet (réseau extérieur) et le réseau de la zone Electrotechnique. Il comporte donc 2 interfaces réseaux pour relier entre eux ces deux réseaux. Dans notre réseau local, chaque machine dispose d'une adresse IP unique qui lui permet d'être identifiée sur le réseau. Un serveur DNS (Domain Name System) effectuent la conversion entre l'adresse symbolique et l'adresse physique réelle qui est un nombre de 32 bits que l'on représente habituellement sous la forme d'une suite de 4 nombres de 0 à 255 comme par exemple '66.94.230.34'. Par ailleurs, j'ai activé l'option DHCP qui permet d'attribuer les adresses IP automatiquement sur les nouveaux postes connectés au sein du réseau.

Product Page:DIR-100 Déconnexion Version du matériel: A1 Version du microprogramme:v1.00FR

D-Link

DIR-100 // CONFIGURATION AVANCÉE OUTILS ÉTAT ASSISTANCE

INTERNET
PARAMÈTRES RESEAU

CONFIGURATION DU RÉSEAU

Cette section vous aidera à configurer les paramètres du réseau interne sur votre routeur ainsi que le serveur DHCP intégré pour assigner une adresse IP aux ordinateurs du réseau. L'adresse IP configurée ici est celle que vous utilisez pour accéder à l'interface de gestion Web.. Si vous la modifiez, il sera peut-être nécessaire d'ajuster les paramètres réseau de votre PC pour pouvoir accéder à nouveau au réseau.

Enregistrer les paramètres Ne pas enregistrer les paramètres

PARAMÈTRES DU ROUTEUR

Cette section vous aidera à configurer les paramètres réseau internes de votre routeur. L'adresse IP configurée ici est celle que vous utilisez pour accéder à l'interface de gestion Web. Si vous la modifiez, il sera peut-être nécessaire d'ajuster les paramètres réseau de votre PC pour pouvoir accéder au réseau.

Adresse IP du routeur: 192.168.1.254
 Masque de sous-réseau par défaut: 255.255.255.0
 Nom du domaine local:
 Activer le relais DNS:

PARAMÈTRES DU SERVEUR DHCP

Cette section vous aidera à configurer le serveur DHCP intégré pour attribuer une adresse IP aux ordinateurs de votre réseau.

Activer le serveur DHCP:
 Plage d'adresses IP DHCP: 101 sur 120 (adresses sur le sous-réseau local)
 Durée d'attribution DHCP: 60 (minutes)

LISTE DES CLIENTS DHCP DYNAMIQUES

Nom d'hôte	Adresse IP	Adresse MAC	Durée écoulée
	192.168.1.108	00-22-15-5f-3a-66	46 minutes 8 seconds

10 - RESERVATION DHCP

	Nom d'hôte	Adresse IP	Adresse MAC	
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<< Nom d'ordinateur
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<< Nom d'ordinateur

Astuces.

- Si votre réseau comporte déjà un serveur DHCP ou si vous utilisez des adresses IP statiques sur tous les périphériques du réseau, décochez la case Activer le serveur DHCP pour désactiver cette option.
- Si votre réseau comporte des périphériques dont l'adresse IP doit être fixe, ajoutez une réservation DHCP à chacun de ces appareils.

X.3.1.1 – Ouverture et redirection des ports de communication

La **redirection de port** ou **port forwarding** ou **port mapping** en anglais, consiste à rediriger des paquets réseaux reçus sur un port donné d'un ordinateur ou un équipement réseau vers un autre ordinateur ou équipement réseau sur un port donné. Cela permet entre autres de proposer à des ordinateurs extérieurs à un réseau d'accéder à des services répartis sur plusieurs ordinateurs de ce réseau.

Afin de pouvoir communiquer avec le serveur Diweb installé dans la zone, il m'a été nécessaire d'ouvrir les ports de communication les plus utilisés afin de pouvoir administrer et gérer celui-ci à distance :

Service réseau	N° de Por	Type	Commentaire
ICMP	7	TCP/UDP	Commandes Ping
Netstat	15	TCP/UDP	Etat du réseau
FTP	21	TCP	Transfert de fichiers
SSH	22	TCP/UDP	SSH Remote Login Protocol
Telnet	23	TCP	Connexion via terminal réseau
SMTP	25	TCP	Envoi de courrier
DNS	53	TCP/UDP	Serveur de noms de domaine
HTTP	80	TCP	Serveur web
POP3	110	TCP	Réception de courrier
IMAP	143	TCP/UDP	Protocole d'accès messagerie Internet
SNMP	161	UDP	Administration de réseau

Exemple de configuration des règles de redirection de port :

	Nom	Adresse IP	Port	Type de trafic
<input type="checkbox"/>	SRV_WEB1	192.168.1.1	Début: 1, Fin: 87	Tout
<input type="checkbox"/>	SRV_WEB2	192.168.1.1	Début: 89, Fin: 1000	Tout
<input type="checkbox"/>	SRV_WEB3	192.168.1.3	Début: 1001, Fin: 65000	Tout
<input type="checkbox"/>			Début: , Fin:	Tout
<input type="checkbox"/>			Début: , Fin:	Tout
<input type="checkbox"/>			Début: , Fin:	Tout

Dans mon exemple, les plages de ports 1 -> 87 et 89 -> 1000 sont redirigés vers le serveur d'application ayant l'adresse IP : 192.168.1.1. Le port 88 étant réservé à l'interface d'administration du routeur. N'ayant pas encore l'utilité des autres ports, ceux-ci sont redirigés vers un serveur pare-feu d'adresse IP 192.168.1.3.

X.3.1 Règles de Pare-feu et l'hôte DMZ

DMZ :

En informatique, une zone démilitarisée (ou DMZ) est un sous-réseau isolé du réseau local par un pare-feu. Ce sous-réseau contient les machines étant susceptibles d'être accédées depuis Internet.

Le pare-feu bloquera donc les accès au réseau local pour garantir sa sécurité. Et les services susceptibles d'être accédés depuis Internet seront en DMZ (donc sans protection par le firewall). En cas de compromission d'un des services dans la DMZ, le pirate n'aura accès qu'aux machines de la DMZ et non au réseau local.

En l'occurrence, les postes informatiques situés dans la DMZ sont les serveurs d'adresse IP 192.168.1.1 et 192.168.1.3.

PARAMETRES DE PARE-FEU

Les options de filtrage Web permettent d'élaborer une liste des sites Web autorisés, qui peut être consultée par plusieurs utilisateurs. Lorsque le filtrage Web est activé, tous les sites qui ne figurent pas sur cette page sont bloqués.

CONFIGURATION DU PARE-FEU

Activer SPI:

Activer la prévention DoS:

HOTE DMZ

L'option DMZ (zone démilitarisée) permet de placer un ordinateur de votre réseau hors de la couverture du routeur. Si un ordinateur rencontre des difficultés avec certaines applications Internet lorsqu'il est derrière le routeur, vous pouvez le placer dans la zone démilitarisée pour lui conférer un accès illimité à Internet.

Remarque: placer un ordinateur dans la zone démilitarisée peut l'exposer à un certain nombre de risques de sécurité. Il est conseillé de n'utiliser cette option qu'en dernier ressort.

Activer l'hôte DMZ:

Adresse IP de l'hôte DMZ:

Activation de la DMZ sur le routeur

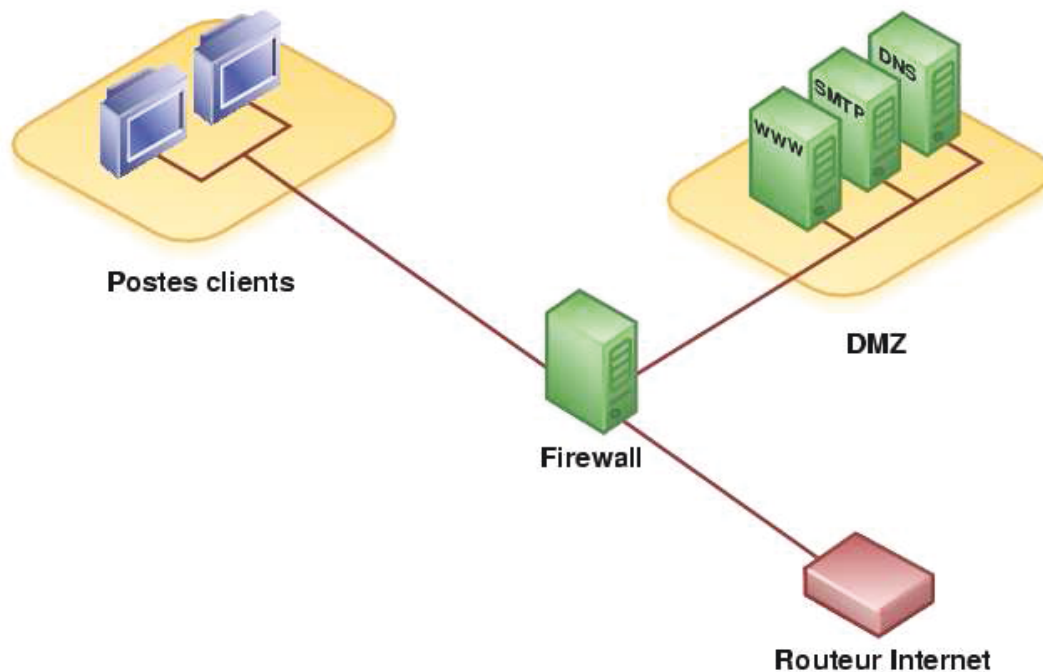


Fig. X.3.1 - Principe de la DMZ

Pare-Feu (Firewall) :

Un **pare-feu** (appelé aussi *coupe-feu*, *garde-barrière* ou **firewall** en anglais), est un système permettant de protéger un ordinateur ou un réseau d'ordinateurs des intrusions provenant d'un réseau tiers (notamment internet). Le pare-feu est un système permettant de filtrer les paquets de données échangés avec le réseau, il s'agit ainsi d'une passerelle filtrante comportant au minimum les interfaces réseau suivante :

- une interface pour le réseau à protéger (réseau interne) ;
- une interface pour le réseau externe.

Le système firewall est un système logiciel, reposant parfois sur un matériel réseau dédié, constituant un intermédiaire entre le réseau local (ou la machine locale) et un ou plusieurs réseaux externes. Il est possible de mettre un système pare-feu sur n'importe quelle machine et avec n'importe quel système pourvu que :

- La machine soit suffisamment puissante pour traiter le trafic ;
- Le système soit sécurisé ;
- Aucun autre service que le service de filtrage de paquets ne fonctionne sur le serveur.

Un système pare-feu contient un ensemble de règles prédéfinies permettant :

- D'autoriser la connexion,
- De bloquer la connexion.

L'ensemble de ces règles permet de mettre en œuvre une méthode de filtrage dépendant de la **politique de sécurité** adoptée par l'entité. On distingue habituellement deux types de politiques de sécurité permettant :

- soit d'autoriser uniquement les communications ayant été explicitement autorisées ;
- soit d'empêcher les échanges qui ont été explicitement interdits.

REGLES DE PARE-FEU

Activée Désactivée

Nom:

Action: Autoriser Refuser

Interface: Adresse IP de début: Adresse IP de fin: Protocole: Plage de ports:

Source: LAN

Destination: WAN TCP -

Programmation: Always On

LISTE DES RÈGLES DE PARE-FEU

Action	Nom	Source	Destination	Protocole,Port
<input type="checkbox"/> Refuser	INTERNET	LAN, 192.168.1.4 - 192.168.1.253	WAN, *	*, *
<input checked="" type="checkbox"/> Autoriser	RESEAU LOCAL	LAN, *	WAN, *	*, *
<input checked="" type="checkbox"/> Autoriser	DMZ Server	*, *	LAN, 192.168.1.1	*, *

X.4 – Réseau de connexion externe entre DMS-Lycée Déodat de Séverac

Afin de renforcer la sécurité de la zone et en amont de la phase de développement. J’ai mis en place un tunnel entre le lycée et DMS. Afin de créer un réseau privé virtuel (VPN) entre celui de DMS et celui du réseau local de l’atelier du Lycée. Ainsi, cela serait plus aisé pour les collaborateurs impliqués dans le projet de chez DMS de pouvoir effectuer des tests sur les systèmes communicants et voir l’avancée du projet.

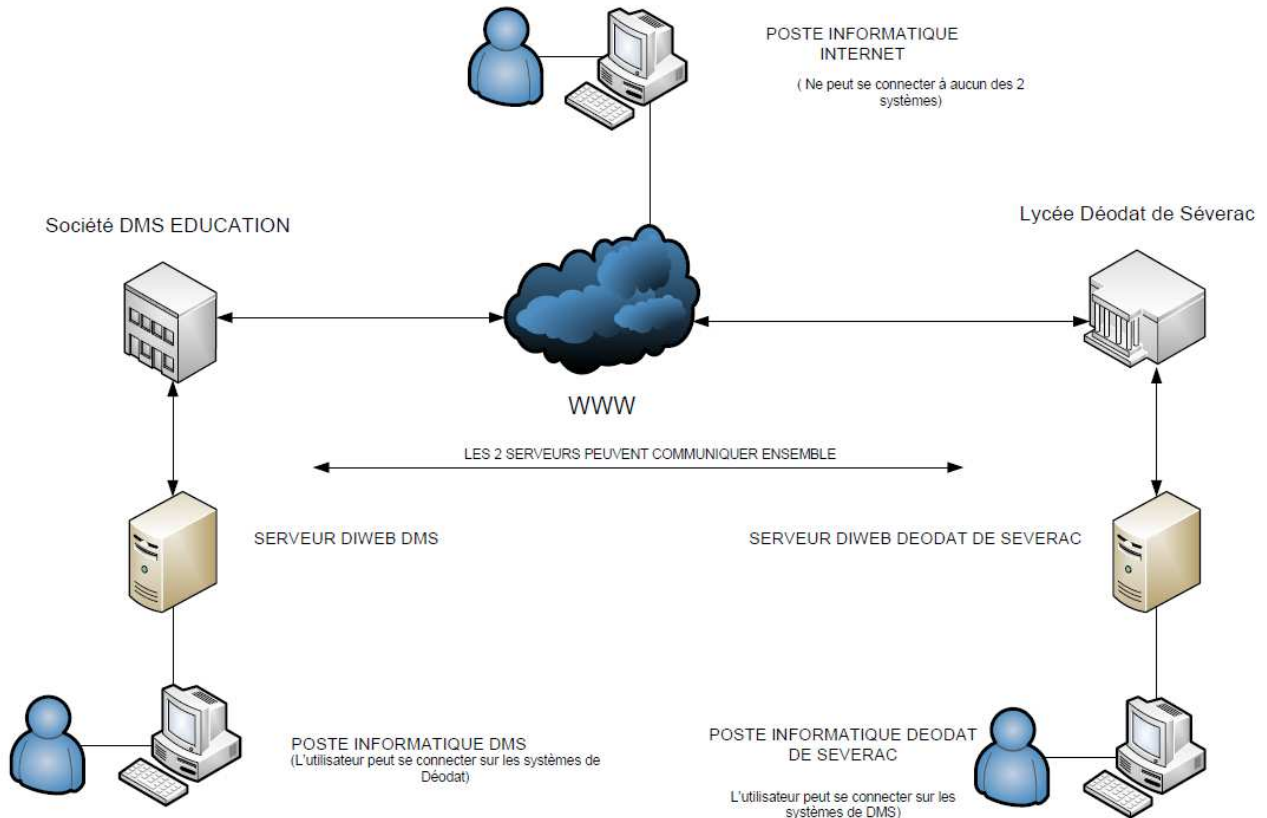


Fig. X.4 – Schéma du réseau de connexion DMS-Lycée Déodat de Séverac

Il aura fallu toutefois, que le réseau de DMS permet cette connexion au niveau du routeur et pare-feu de l'entreprise. J'ai demandé à ce que des ports soient redirigés vers mon poste informatique afin d'avoir la possibilité de prendre en main à distance les serveurs et systèmes communicants. De plus, au niveau du routeur du Lycée, seul mon poste est autorisé à réaliser certaines type de connexions.

X.5 – Préparation des serveurs

Une fois que l'architecture et la structure des réseaux est mise en place, je me suis occupé de l'installation des serveurs. Les serveurs jouant un rôle important dans la sécurité d'un parc informatique et dans le développement de notre plateforme, c'était une tâche importante à effectuer.

Pour l'installation des serveurs, la société DMS souhaite que j'opte pour des distributions libres comme Linux par exemple. Cette décision, m'a enthousiasmé car ayant moi-même des connaissances avancée dans les distributions Linux, je n'aurais pas eu de mal à pouvoir les administrer.

Pour cela, la société DMS m'a fourni 2 postes informatiques ainsi que le Lycée Déodat de Séverac qui s'est doté d'un super serveur d'une capacité technique importante. Avec ces PC à ma disposition, je me suis penché sur les distributions Linux que je pouvais installer dessus. La plupart de ces distributions sont des plateformes LAMP⁴⁵ et le choix peut-être rapide si l'on sait quel genre d'application on souhaite développer.

Dans un réseau informatique, un **serveur** est à la fois un ensemble de logiciels et l'ordinateur les hébergeant dont le rôle est de répondre de manière automatique à des demandes de services envoyées par des clients — ordinateur et logiciel — via le réseau.

Les serveurs sont d'usage courant dans les centres de traitement de données, les entreprises, les institutions, et le réseau Internet, où ils sont souvent un point central et sont utilisés simultanément par de nombreux utilisateurs pour stocker, partager et échanger des informations. Les différents usagers opèrent à partir d'un *client* : ordinateur personnel, poste de travail, ou terminal.

Les utilisations courantes des serveurs sont le serveur de fichiers, d'impression, de base de données, de courrier, ainsi que le serveur web, le serveur d'applications, le proxy ou le serveur de jeu.

Un serveur de fichiers (anglais *file server*) est utilisé pour le stockage et le partage de fichiers entre plusieurs clients. Un serveur d'impression est utilisé comme intermédiaire entre un ensemble de clients et un ensemble d'imprimantes, tandis qu'un serveur de base de données est utilisé pour stocker et manipuler des données contenues dans une ou plusieurs bases de données et partagées entre plusieurs clients.

Un serveur de courrier est utilisé pour stocker et transmettre du courrier électronique. Un serveur web stocke et manipule les pages d'un site web et les transmet sur demande au client (voir navigateur web). Un serveur de jeu arbitre et suit l'évolution d'un jeu en mettant en communication les différents joueurs.

Un serveur d'applications effectue les traitements d'un ou plusieurs logiciels applicatifs à architecture client/serveur. Un serveur mandataire (anglais *proxy*) reçoit des demandes, les contrôle, puis les transmet à d'autres serveurs¹⁷.

L'auto-hébergement consiste pour un particulier à utiliser son ordinateur personnel comme serveur, et de le mettre à disposition par l'intermédiaire de sa connexion Internet personnelle. L'utilisateur évite ainsi les frais dus à un contrat avec un hébergeur web. Il peut ainsi avoir son propre serveur web

⁴⁵ Serveur web : Linux, Apache, MySql et Php

(LAMP, WAMP, ...) et, en utilisant un DNS dynamique (**en**), avoir son propre nom de domaine. Certains FAI font payer les adresses IP fixes, mais elles ne sont pas indispensables.

Un serveur d'impression répond à des demandes en rapport à des travaux d'impression numérique. Les documents sont placés dans des files d'attente, puis envoyés aux imprimantes

Un serveur de base de données répond à des demandes de manipulation de données stockées dans une ou plusieurs bases de données. Il s'agit typiquement de demandes de recherche, de tri, d'ajout, de modification ou de suppression de données²³.

Un serveur web répond à des demandes de consultation de documents tels que ceux du World wide web. Chaque consultation d'un nouveau document hypertexte par parcours d'un hyperlien dans le logiciel client (la plupart du temps un navigateur web) provoque l'envoi d'une nouvelle demande. Chaque demande peut être envoyée à un serveur différent.

En réponse à une demande, le serveur peut envoyer un document qui est stocké en tant que fichier dans les mémoires de masse du serveur ou exécuter un traitement puis envoyer le résultat sous forme d'un document. HTTP et HTTPS sont les protocoles des serveurs web et HTML est le format de fichier prévu à cet effet.

Mon besoin se situe principalement sur l'installation d'un serveur web et d'un serveur d'applications. Les systèmes d'exploitation pour lesquels j'opte seront principalement des distributions Linux. Il existe une multitude de distribution Linux, tous ayant ses avantages et ses inconvénients, surtout au niveau de leurs utilisations. Pour ma part, j'ai une nette préférence pour les distributions Linux Debian, auquel je suis assez bien familiarisé.

Il peut-être à la fois complexe ou simple à utiliser selon les besoins que l'on a. Mais il reste un système d'exploitation complet présentant un large panel d'outils lors de son installation. Etant une plateforme de type LAMP, Je vais détailler et fournir quelques explications sur le contenu d'une plateforme LAMP.

X.5.1 – Plateforme LAMP

X.5.1 – Définition

LAMP est un acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de sites Web. L'acronyme original se réfère aux logiciels suivants :

- « Linux », le système d'exploitation (GNU/Linux) ;
- « Apache », le serveur Web ;
- « MySQL », le serveur de base de données ;
- « PHP » à l'origine, puis « Perl » ou « Python », les langages de script.

Les rôles de ces quatre composants sont les suivants :

- Linux assure l'attribution des ressources aux autres composants ;
- Apache est le serveur web « frontal » : il est « devant » tous les autres et répond directement aux requêtes du client web (navigateur) ;
- MySQL est un système de gestion de bases de données (SGBD). Il permet de stocker et d'organiser des données ;
- le langage de script PHP permet la génération de pages web dynamiques et la communication avec le serveur MySQL.

Tous les composants peuvent être situés :

- sur une même machine ;
- sur deux machines, généralement Apache et le langage de script d'un côté et MySQL de l'autre ;
- sur de nombreuses machines pour assurer la haute disponibilité (répartition de charge et/ou *failover*).

X.5.2 – Apache

Apache http Server, souvent appelé Apache, est un logiciel de serveur http produit par l'Apache Software Foundation . C'est le serveur http le plus populaire du web. C'est un logiciel libre avec un type spécifique de licence, nommée licence Apache.

Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, Common Gateway Interface, Server Side Includes, réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc.

Les possibilités de configuration d'Apache sont une fonctionnalité phare. Le principe repose sur une hiérarchie de fichiers de configuration, qui peuvent être gérés indépendamment. Cette caractéristique est notamment utile aux hébergeurs qui peuvent ainsi servir les sites de plusieurs clients à l'aide d'un seul



X.5.3 – MySQL

MySQL est un système de gestion de base de données (SGBD⁴⁶). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server.

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

C'est un logiciel libre développé sous double licence en fonction de l'utilisation qui en est faite : dans un produit libre ou dans un produit propriétaire. Dans ce dernier cas, la licence est payante,

⁴⁶ *Système de Gestion de Base de Données*

sinon c'est la licence publique générale GNU⁴⁷ qui s'applique. Ce type de licence double est utilisé par d'autres produits comme le Framework de développement de logiciels Qt (pour les versions antérieures à la 4.5).



X.5.4 – PHP

A ce stade du projet, je ne savais pas en langage j'aurais développé la plateforme, PHP étant un package par défaut lors de l'installation de la distribution LAMP, je ne l'ai pas supprimé.

PHP (sigle de *PHP: HyperText Preprocessor*³) est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP³, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes. En raison de la richesse de sa bibliothèque, on désigne parfois PHP comme une plate-forme plus qu'un simple langage.

Le langage PHP est utilisé principalement en tant que langage de script côté serveur, ce qui veut dire que c'est le serveur (la machine qui héberge la page Web en question) qui va interpréter le code PHP et générer du code (constitué généralement d'XHTML ou d'HTML, de CSS, et parfois de JavaScript) qui pourra être interprété par un navigateur. PHP peut également générer d'autres formats en rapport avec le Web, comme le WML, le SVG, le format PDF, ou encore des images bitmap telles que JPEG, GIF ou PNG.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent dédiées au Web. PHP est très majoritairement installé sur un serveur Apache, mais peut être installé sur les autres principaux serveurs HTTP du marché, par exemple IIS. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

Il est multiplateforme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (prendre en compte les règles d'arborescences de répertoires qui peuvent changer).

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage.

PHP appartient à la grande famille des descendants du C, dont la syntaxe est très proche. En particulier, sa syntaxe et sa construction ressemblent à celles des langages Java et Perl, à la différence que du code PHP peut facilement être mélangé avec du code HTML au sein d'un fichier PHP.

⁴⁷ *General Public Licence*, licence pour la distribution de logiciels libres

Dans une utilisation Web, l'exécution du code PHP se déroule ainsi : lorsqu'un visiteur demande à consulter une page Web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension .php), le serveur appelle l'interprète PHP qui va traiter et générer le code final de la page (constitué généralement d'HTML ou de XHTML, mais aussi souvent de CSS et de JS). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client.

Ce schéma explique ce fonctionnement :



Une étape supplémentaire est souvent ajoutée : celle du dialogue entre PHP et la base de données. Classiquement, PHP ouvre une connexion au serveur de SGBD voulu, lui transmet des requêtes et en récupère le résultat, avant de fermer la connexion.



XXI. INSTALLATION ET ADMINISTRATION DE SERVEUR

XI.1 – Mise en place et optimisation

Un serveur Web nécessite aussi des services bien spécifiques utilisés pour les phases de développement et c'est ces derniers qu'il a fallu installer et configurer pour les rendre opérationnels.

L'ensemble des services que nous souhaitons pour notre serveur web sont donc :

- **Apache 2** : Le serveur Web très utilisé,
- **PHP** : Service couplé à Apache permettant l'utilisation du langage PHP sur le dit serveur,
- **MySQL**: Service de gestion de base de données qui couplé à Apache et PhpMyAdmin permet la gestion de notre base de données,
- **PhpMyAdmin** : Interface de gestion de notre base de données en ligne,
- **SVN** : Service de gestion de versions, permet le travail en équipe sur le même projet sans porter atteinte aux modifications des autres.

Installer Apache est la base d'un serveur Web, c'est lui qui s'occupe faire fonctionner les autres services et de les faire interagir entre eux. PHP et Mysql sont les deux services qui viennent en second, le premier permettant d'interpréter le code de l'application et le second permettant le stockage en base de données des informations nécessaires.

PhpMyAdmin n'étant qu'une couche « graphique » en ligne de MySQL, il n'est pas indispensable mais permet d'avoir une vue simplifiée de la base de donnée et même, maintenant, de modéliser cette dernière. C'est le principal outil que j'ai utilisé pour gérer la base de données à l'instar de MySQL cc.

SVN est indispensable si une équipe de projet souhaite travailler sur le même projet. En effet SVN permet à plusieurs personnes de travailler sur une même base de fichiers sans que les modifications faites sur un même fichier par deux personnes différentes ne s'écrasent mutuellement, les fichiers sont fusionnés et si il y'a conflit entre les modifications, c'est reporté et cela peut ainsi être résolu.

De plus SVN est une application multi-utilisateur, on peut ainsi tracer les modifications de chacun et revenir à une version précédente en cas d'erreur d'un des membres de l'équipe.

J'ai donc opté pour la distribution Linux Debian qui est un système très largement utilisé et apprécié en tant que OS d'un serveur.



Je ne vais pas rentrer en détails dans le processus d'installation d'un serveur web sous Debian car il existe un large éventail de tutoriels concernant cela.

MySQL

MySQL est historiquement lié à PHP. Si les SGBD courants privilégient avant tout l'intégrité des données, MySQL a vu le jour pour privilégier les performances. Aujourd'hui, cet aspect performance est toujours au rendez-vous, mais l'utilisateur peut choisir plusieurs moteurs en fonction de ses besoins et ainsi répondre pleinement à des problématiques web ou non web.

On utilisera MySQL pour tout type d'application web, de petite taille, de taille moyenne et même de grande taille. Il s'agit d'un bon choix dans tous les cas.

Pour accéder à une base de données avec PHP, une extension spécifique est nécessaire. Aujourd'hui il existe deux sortes d'extensions : les extensions indépendantes et les pilotes PDO (PHP Data Objects).

- Les extensions *indépendantes* sont les toutes premières à avoir vu le jour. Elles proposent des fonctions et parfois des classes permettant d'effectuer des opérations sur le SGBD.
- Les extensions PDO permettent de lier le SGBD à PDO, qui propose des classes de manipulation standard, quel que soit le SGBD utilisé. L'utilisation de PDO est intéressante aussi bien pour simplifier la maintenance que pour assurer la durabilité des développements.

PDO (*PHP Data Objects*) est une extension un peu particulière qui permet d'utiliser la même interface (*classes et méthodes*) quel que soit le SGBD sous-jacent. Les extensions qui font la liaison entre PDO et les SGBD s'appellent des pilotes (*drivers*) : pdo_mysql, pdo_oci, pdo_sqlite, etc.

XI.1.1 – Installation du serveur web sous Debian

Le projet étant développé sur un serveur situé au sein du réseau local de la zone Electrotechnique du Lycée. Un serveur dédié au projet a été acquis par le Lycée afin que je puisse le configurer de façon à optimiser les conditions de développement informatique.



Aperçu du nouveau serveur utilisé pour l'application : HP ProLiant ML

XI.1.2 – Caractéristiques techniques du serveur

Spécifications techniques	
Processeurs	Intel® Xeon® E5649 (6 cœurs, 2,53 GHz, 12 Mo L3, 80 W)
Nombre de processeurs	1
Noyau processeur disponible	6
Mémoire en standard	6 Go
Logements pour la mémoire	18 logements DIMM
Mémoire	PC3-10600, modules DDR3-1333 enregistrés
Logements d'extension	9
Contrôleur réseau	(1) 4 ports multifonction NC375i 1GbE
Type d'alimentation électrique	(1) 750 watts CS à haut rendement
Contrôleur de stockage	(1) Smart Array P410i/512 Mo BBWC
Logiciel d'administration	Sans objet
Type du lecteur optique	DVD-ROM (SATA)
Format (entièrement configuré)	4U
Garantie (pièces, main d'œuvre, sur site)	3/3/3

XI.1.3 – Installation des packages

On vérifie la présence d'aptitude (gestionnaire de paquets) sur le serveur Debian si l'on souhaite installer via une interface de gestion sinon il est possible d'effectuer l'installation manuellement via les lignes de commandes Shell via invite de commande :

- Installer apache : `apt-get install apache2`
- Installer mysql-server5.0 : `apt-get install mysql-server5.0`
- Installer php5 : `apt-get install php5`
- Installer phpmyadmin : `apt-get install phpmyadmin`
- Installer usvn : `apt-get install usvn`

XI.1.4 - Configuration d'apache2

Définition du "DocumentRoot"

Afin que le serveur Apache sache où se trouve la plate-forme Diweb il faut lui indiquer le document root, la racine du site. Pour ce faire il va falloir d'abord vérifier dans les fichiers suivants qu'il n'existe pas déjà une ligne DocumentRoot avant de le redéfinir (Une simple recherche dans ces fichiers devrait suffire) :

- `/etc/apache2/sites-enabled/000-default`
- `/etc/apache2/apache2.conf`

S'il existe une ligne du type DocumentRoot alors on la supprime. Ensuite il faudra éditer le fichier /etc/apache2/httpd.conf (normalement vide depuis la version 2 d'apache, fichier de configuration pour les utilisateurs) et y ajouter la ligne suivante :

- DocumentRoot "/chemin/vers/le/dossier/public/de/diweb/"

XI.1.5 - Configuration de mysql-server5.0

Opérations basiques

Le serveur MySQL démarre automatiquement après l'installation du système d'exploitation. Il est aussi possible de le démarrer manuellement la commande suivante `/etc/init.d/mysql start` via l'invite de commande. Par la suite, à chaque mise à jour effectuée (changer le nom de domaine dans la base mysql par exemple), il sera nécessaire de relancer le serveur avec `/etc/init.d/mysql restart`

Débloquer l'accès à distance de la base

Dans le fichier my.cnf, il faut commenter la ligne `bind_adress=127.0.0.1` afin que l'accès la base soit possible à distance.

XI.1.6 - Mise en place des règles de routage

Connaitre l'interface que l'on va utiliser

Il est impératif de bien connaître l'interface ethernet (carte réseau du PC) que l'on va utiliser. En utilisant la commande `ifconfig` on peut connaître les interfaces existantes et opérationnelles. Il faudra en choisir une si il y'en a plusieurs. En général elles portent le nom de eth0 et eth1.

Connaitre son réseau logique

Pour mettre en place les règles de routage, il faut connaître son réseau. En général il est du type 192.168.1.0 mais peut tout aussi bien être différent.

Lancer le script de mise en place automatique

Pour mettre en place les règles de routage simplement, il suffit de lancer le script `IptablesMaker.sh` avec deux paramètres. Par exemple avec une interface eth1 et un réseau de type 192.168.1.0 ça donnerait : `./IptablesMaker.sh 192.168.1 eth1`. Le premier paramètre étant les trois premiers membres de l'ip du réseau et le second étant l'interface. Le script se chargera tout seul d'appliquer les règles de routage.

Activer l'ip forwarding

Il est indispensable que l'ip forwarding soit activé sur la machine, c'est ce qui permet le relaiage des trames et de l'information. S'il n'est pas activé nos règles de routage ne servent à rien. Pour activer l'ip forwarding il faut taper dans l'invite de commande : `echo 1 > /proc/sys/net/ipv4/ip_forward`. Pour vérifier l'état de l'ip forwarding il faudra taper : `cat /proc/sys/net/ipv4/ip_forward`. Un 1 correspond à activé, un 0 à désactivé.

XI.1.7 - Configuration de USVN

Lancer le daemon SVN

Pour que le serveur puisse répondre aux requêtes, il faut lancer le service qui s'en occupera et pour se faire il faut taper : `svnserve -d` dans l'invite de commande. Si un message d'erreur apparait c'est que vous l'avez probablement déjà lancé.

Faire un dump du repository actuel

Pour pouvoir transférer un repository il faut d'abord "dumper" l'original. Pour dumper un repository il faut taper : `svnadmin dump NomDuRepository`. Un fichier sera alors généré et pourra être réutilisé dans l'étape suivante.

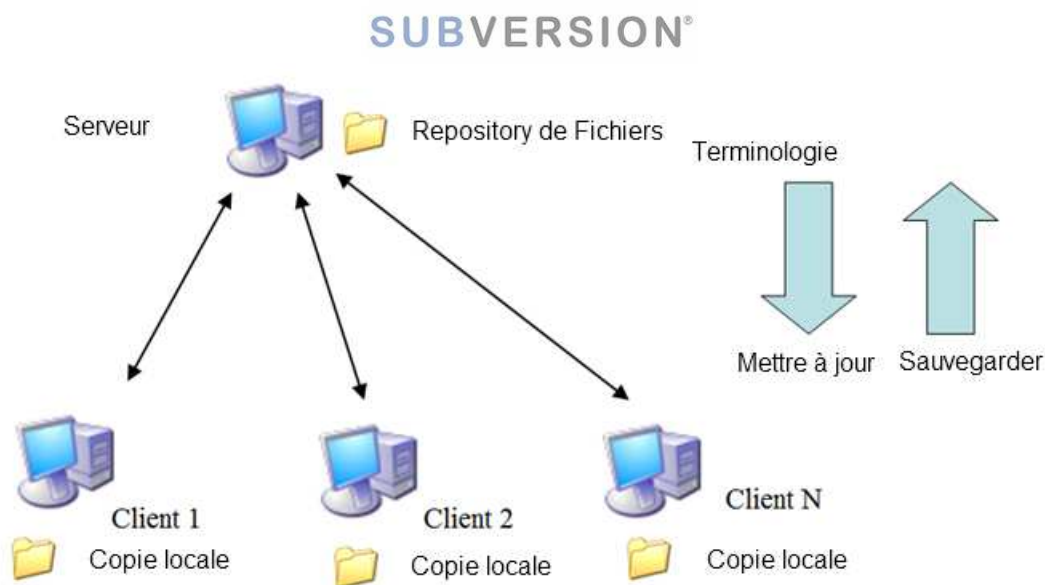
Décompresser le dump

Pour pouvoir décompresser le dump du repository que nous venons de faire, il faut savoir où vous souhaitez le situer. Ensuite, il faut créer le repository avec la commande : `svnadmin create NomDuRepositorySouhaité`. Ensuite, il faut charger notre dump avec la commande : `svnadmin load NomDuDump`. Normalement, on voit toutes les révisions s'afficher et à la fin le repository devrait être chargé.

Principe du SVN

Subversion (en abrégé svn) est un système de gestion de versions, distribué sous licence Apache et BSD. Il fait la gestion concurrente de versions ou de la gestion des versions concurrentes. Il permet ainsi d'historiser les différentes évolutions du code source d'un logiciel, ainsi que de gérer les éventuels conflits entre les modifications effectuées par plusieurs développeurs.

Tout seul ou en équipe, un tel outil devient vite indispensable pour gérer les sources. Il aussi répond à un besoin de simplification et de sécurisation pour la gestion du code source. Aujourd'hui, des centaines de milliers de projets utilisent un gestionnaire de versions (CVS ou Subversion), cela permet de gérer un projet dans la continuité et de simplifier le travail à plusieurs.



Principe de Subversion

Installation d'USVN

USVN est une interface web d'administration et de gestion de dépôts Subversion, écrite en PHP. Elle autorise l'administration des utilisateurs, groupes, et projets de votre dépôt sans avoir à utiliser un outil d'administration en ligne de commande. Son but est aussi de faciliter la création de nouveaux projets sans avoir à utiliser une interface en ligne de commande, et potentiellement, sans avoir accès

L'installation de USVN requiert la présence de :

- PHP 5 (5.1.2 <= ver < 5.3)
- apache2
- mod_dav enable (dans Apache httpd.conf - décommenter "LoadModule dav_module modules/mod_dav.so")
- mod_dav_fs enable (dans Apache httpd.conf - décommenter "LoadModule dav_fs_module modules/mod_dav_fs.so")
- mod_rewrite enable (dans Apache httpd.conf - décommenter "LoadModule rewrite_module modules/mod_rewrite.so")
- ajout de la configuration AllowOverride
- Subversion - ajouter les modules suivant au modules d'Apache
 - mod_authz_svn enable (ajouter dans httpd.conf "LoadModule authz_svn_module modules/mod_authz_svn.so")
 - mod_dav_svn enable (ajouter dans httpd.conf "LoadModule dav_svn_module modules/mod_dav_svn.so")

A la suite de l'installation, le repository du projet se retrouvera à l'adresse suivante : <svn://194.254.30.90/var/www/usvn/files/diweb2/trunk/application>



XI.2 – Installation du serveur de routage et de contrôle d'accès

Ce serveur a été installé avec une distribution Linux SME Server qui permettra de créer une passerelle sous Linux afin de partager la connexion Internet sur un réseau local de la zone Electrotechnique. SME Server est une distribution Linux basée sur CentOS.

Cette distribution permettra également de créer nos propres services internet (messagerie, http, ftp, vpn, ...) et nos services de réseau local (serveur de fichiers, serveur d'impression, filtre antivirus sur la messagerie ...). Une interface web très intuitive offre une prise en main rapide de toutes les fonctions d'administration du serveur.

Même si les possibilités du serveur sont multiples, nous nous intéresserons surtout aux fonctionnalités suivantes du serveur :

- Règles de pare-feu (firewall) préconfigurées pour bloquer certaines attaques intérieurs. En effet, si l'on configure une adresse IP manuellement au sein du réseau local sans passer par le

routeur, une personne peut prendre en main un système à distance. Donc globalement, l'aspect sécurité est pris en compte à tous les niveaux.

- Serveur FTP pour partager les données sur Internet (notamment les supports de cours ou documents électroniques).
- Serveur de messagerie sécurisé pour envoyer et recevoir des mails directement, avec possibilité d'accéder à distance (via Internet, par exemple) à ses comptes de messagerie depuis un client de messagerie ou une interface WebMail sécurisée.
- Filtres antivirus et anti-pourriels pour la messagerie qui permettent de se prémunir de ces fléaux de plus en plus répandus.
- Serveur de fichier Samba.
- Redirection de certains ports entrants vers des machines spécifiques.
- Possibilité de faire des sauvegardes de données.

A la base, ce serveur devait faire le lien entre le réseau Electrotechnique et celui du Lycée, effectivement, les adresse IP et Masque de sous réseau sont différentes. La passerelle devait donc être un routeur DHCP et firewall à la place du routeur Netgear. Nous utiliserons donc ce serveur autrement.

Configuration du système :

Nom de domaine : deodat-severac-diweb.fr.nf

Nom du système : sme_server - IP locale : 192.168.1.3 - Masque de sous-réseau : 255.255.255.0

Accès à la configuration du serveur : [htTP://192.168.1.3/server-manager](http://192.168.1.3/server-manager)

SME Server™
Server manager

admin@avempace.yakati.selfip.net ? Logout

Collaboration
Utilisateurs
Groupes
Quotas
Pseudonymes
I-bays
Récupération de mails

Administration
Sauvegarde et restauration
Visualisation des fichiers journaux
Journaux de messagerie
Arrêt et redémarrage

Sécurité
Accès à distance
Gestion des réseaux locaux
Renvoi de port
Services Proxy

Configuration
Mise à jour logicielle
Date et heure
Groupe de travail
Annuaire LDAP
Imprimantes
Gestion des noms d'hôte et des adresses

Bienvenue dans le gestionnaire du serveur

Bienvenue dans SME Server, la distribution Linux dédiée aux petites et moyennes entreprises. SME Server est mis à votre disposition par SME Server, Inc., organisation à but non lucratif dont l'objectif est d'assurer la promotion, le support et l'existence légale de SME Server.

SME Server est gratuitement exploitable, selon les termes de la Licence "GNU General Public License", et n'existe que grâce aux efforts de la communauté formée autour de SME Server. Toutefois, la disponibilité et la qualité de SME Server dépendent de notre capacité à régler nos dépenses telles que les coûts liés à l'hébergement, au serveur matériel, etc.

De ce fait, nous demandons une petite contribution financière afin de couvrir nos dépenses et d'assurer les développements à venir.

Veillez aller sur <http://www.smeserver.org/donate/> pour effectuer votre donation.

Ce système d'exploitation ne dispose d'AUCUNE GARANTIE. Veuillez cliquer ici pour afficher les informations détaillées concernant le support technique, la garantie et la licence d'exploitation.

Pour effectuer une tâche d'administration du système, cliquez sur l'un des liens dans le menu situé à la gauche de votre écran.

SME Server 7.5.1
Copyright 1999-2006 Mitel Corporation
Tous droits réservés.
Copyright 2006 SME Server, Inc.

Par ailleurs le routeur Wifi est configuré sur le serveur de passerelle, celui-ci faisant office de DHCP pour les connexions entrantes et sortantes sur le Wifi. Cette architecture assure donc une sécurité et une protection maximale contre toutes intrusions sur le réseau.

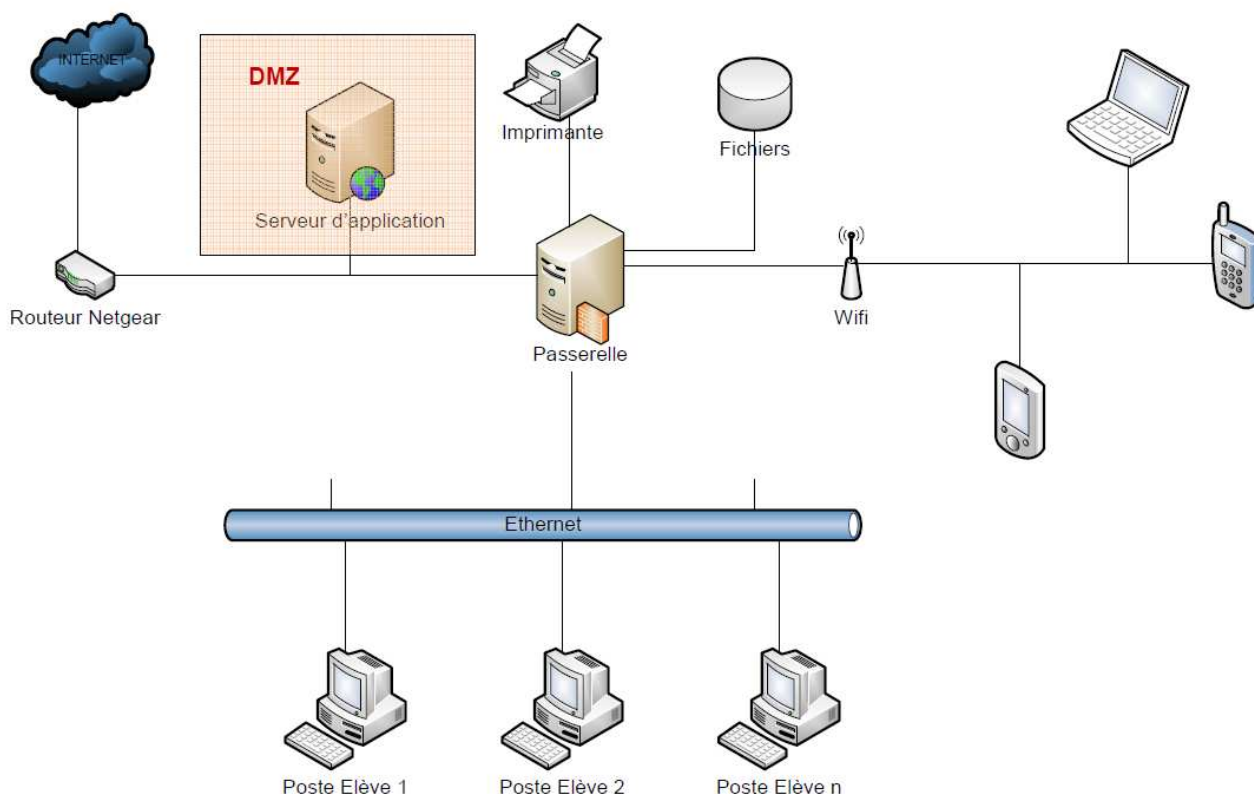


Fig. Schéma d'architecture de la passerelle dans le réseau local de l'atelier

XI.3 – Accès à distance aux systèmes

La structure de l'application est limitée dans le cas où l'on souhaite accéder à l'interface de commande d'un système communicant depuis le réseau internet. Alors que les systèmes n'étaient accessibles que par le réseau local de l'atelier, il était prévu dans un premier temps que ceux devaient n'échanger des informations qu'avec le serveur web afin d'alimenter celui-ci en données exploitables lors des TP. Mais je me suis vite rendu-compte que les enseignants souhaitent dans un premier temps avoir accès et tester la communication externe depuis chez eux, notamment la caméra pivotable.

Pour pallier à cette demande, j'ai rendu tous les serveurs embarqués des systèmes accessibles en routant depuis la passerelle les ports du serveur vers un système. Par exemple, pour accéder à un système, on spécifie un port de communication depuis l'url de la plateforme. Ces ports étant réservés de 9001 à 9020 au niveau du routeur Netgear, celui-ci route la demande vers le serveur passerelle. Celui-ci, en fonction du numéro de port reçu va router la requête vers le système correspondant.

Par exemple, en tapant l'adresse <http://diweb-serveur.com:9001> on accèdera à un système X, en tapant l'adresse <http://diweb-serveur.com:9002> on accèdera à un système Y. On remarque que le résultat dépend du numéro de port spécifié.

Mais ce type de fonctionnement se retrouve bien vite limité car seul les ports 80 de chaque système est accessible, chaque serveur embarqué communique avec bien plus de ports que cela. De plus, il

est nécessaire d'exécuter une demande d'autorisation du serveur pour chaque système qui sera ajouté par la suite ... Ce qui est assez embêtant lorsque l'on n'a pas de notions avancées en informatique !

XI.4 – Génération automatique des tables de routage & IPTables

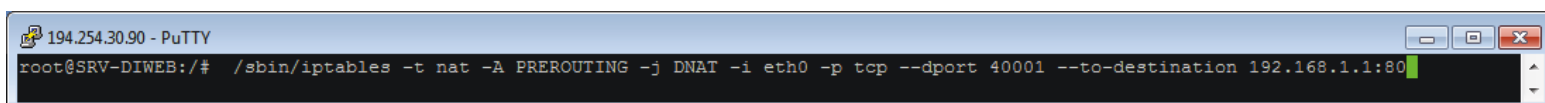
Comme expliqué précédemment, le système de routage, qui en fonction du port, permet de cibler un système en particulier. Afin de rendre accessible tout nouveau système au sein du réseau local et rendre accessible ses ports, il faut automatiser le traitement.

C'est en utilisant l'outil « Iptables » que j'ai procédé à une définition générale des routes à appliquer. Prenons par exemple, un système X sachant dont je souhaite rendre son serveur accessible via son protocole http (port 80). La requête provenant de l'utilisateur distant sera accompagnée du port 9001. En connaissant le port 9001, le serveur saura qu'il faut rediriger la requête vers le système X en changeant le port vers le port 80 (port pour le protocole http) pour consulter l'interface.

Mais les systèmes utilisent d'autres ports que le numéro 80, d'autres ports existent comme le port 502 et qui est utilisé par les serveurs embarqués pour communiquer (protocole Modbus). Par conséquent, il faut définir une liste de ports de communications correspondant à chacun de ceux utilisés par les serveurs web embarqués connectés. Heureusement ces serveurs n'utilisent que 5 ports⁴⁸ soit les suivants :

- HTTP : port 80
- FTP : port 21
- SSH : 22
- TELNET : 51
- MODBUS : 502

Les serveurs web embarqués de type Siemens utilisent un port supplémentaire pour leur connexion VNC.



```
194.254.30.90 - PuTTY
root@SRV-DIWEB:~# /sbin/iptables -t nat -A PREROUTING -j DNAT -i eth0 -p tcp --dport 40001 --to-destination 192.168.1.1:80
```

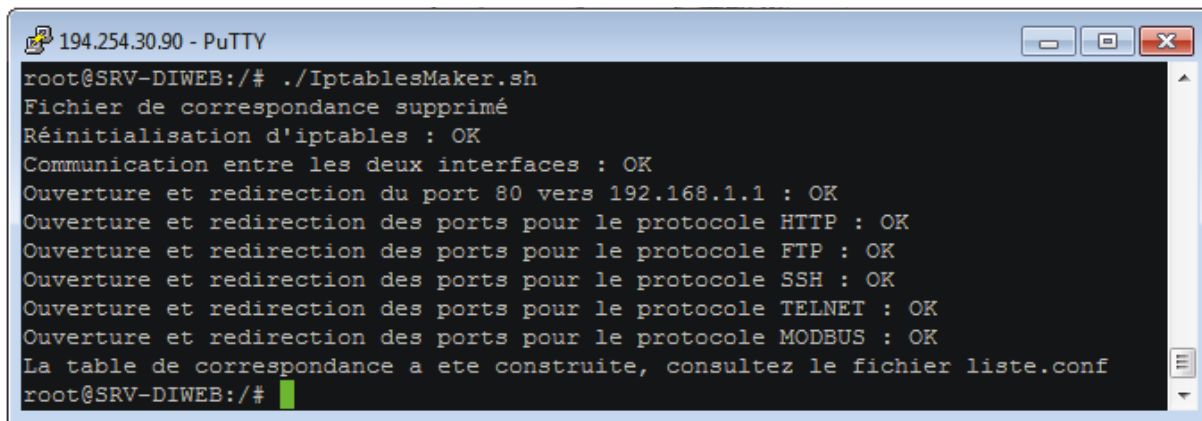
Lancement du fichier bash IpTableMaker.sh

Exemple de commande pour ouvrir une route

J'ai donc programmé un script Shell définissant automatiquement les ports pour chaque protocole. Dans notre cas nous avons utilisé une plage de ports allant de 40000 à 42000, ce sont des ports disponibles par le serveur dont nous allons les mettre à nos profits.

Le résultat de l'exécution du script dans un terminal donne :

⁴⁸ Ports de communications d'un serveur



```
194.254.30.90 - PuTTY
root@SRV-DIWEB:/# ./IptablesMaker.sh
Fichier de correspondance supprimé
Réinitialisation d'iptables : OK
Communication entre les deux interfaces : OK
Ouverture et redirection du port 80 vers 192.168.1.1 : OK
Ouverture et redirection des ports pour le protocole HTTP : OK
Ouverture et redirection des ports pour le protocole FTP : OK
Ouverture et redirection des ports pour le protocole SSH : OK
Ouverture et redirection des ports pour le protocole TELNET : OK
Ouverture et redirection des ports pour le protocole MODBUS : OK
La table de correspondance a ete construite, consultez le fichier liste.conf
root@SRV-DIWEB:/#
```

Affichage du script shell permettant de créer les différentes routes

Le script contient plusieurs étapes :

- Suppression du fichier de correspondance, qui est le fichier contenant un tableau de correspondance entre IP, ports et protocoles si le script a déjà été exécuté.
- Réinitialise de la table de routage.
- Activation de la communication entre les deux interfaces (cartes réseau) du serveur.
- Redirections des requêtes entrantes avec le port 80 sur le serveur web afin de rendre le site accessible.
- Création de toutes les routes pour tous les protocoles définis tout en construisant la table de correspondance.

Code source du script de « IpTableMaker.sh »

Le script IptablesMaker mets en place toutes les règles de routage sur une machine. Ce script est à exécuter dans un terminal sous Linux. L'affichage informe l'utilisateur de l'avancement du processus.

```
1  #!/bin/bash
2  ## Suppression du fichier liste.conf si il existe
3  `rm ./liste.conf`
4  echo "Fichier de correspondance supprimé"
5  ## Réinitialisation des tables
6  /sbin/iptables -t nat -F
7  /sbin/iptables -t nat -X
8  echo "Réinitialisation d'iptables : OK"
9  ## On rend transparent la communication entre les deux interfaces du serveur
10 /sbin/iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
11 /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
12 echo "Communication entre les deux interfaces : OK"
13 ## On définit la redirection pour le port 80 sur le serveur de développement
14 /sbin/iptables -t nat -A PREROUTING -j DNAT -i eth0 -p tcp --dport 80 --to-destination 192.168.1.1:80
15 echo "Ouverture et redirection du port 80 vers 192.168.1.1 : OK"
16 a=2;
17 for (( nbP=0; nbP<5; nbP++ ))
18 do
19     case $nbP in
20         0)
21             protocole=80
22             nomProtocole=HTTP;;
23         1)
24             protocole=21
25             nomProtocole=FTP;;
26         2)
27             protocole=22
28             nomProtocole=SSH;;
29         3)
30             protocole=23
31             nomProtocole=TELNET;;
32         4)
33             protocole=502
34             nomProtocole=MODEBUS;;
35     esac
36     ip=2;
37     for (( a=2; a<=254; a++ ))
38     do
39         ip=`expr $ip + 1`
40         max=`expr $nbP \* 254`
41         plage=`expr $max + 40001`
42         port=`expr $a + $plage`
43         /sbin/iptables -t nat -A PREROUTING -j DNAT -i eth0 -p tcp --dport $port --to-destination 192.168.1.$ip:$protocole
44         correspondance="$port : 192.168.1.$a pour le protocole $nomProtocole"
45         echo $correspondance >> ./liste.conf
46     done
47     echo "Ouverture et redirection des ports pour le protocole $nomProtocole : OK"
48 done
49 echo "La table de correspondance a été construite, consultez le fichier liste.conf"
50 exit 0
```

XI.5 – Mappage du réseau

Afin d'assister les enseignants dans l'utilisation des communications à distance des systèmes. J'ai ajouté une petite application afin qu'il puisse avoir une visibilité sur l'état du réseau et les systèmes en marche. Il était d'ailleurs évident que sans connaître de notion de ports de communication, les enseignants auraient eu de la difficultés à retenir les ports de l'adresse IP. Ce second programme est écrit en script shell & php.

Il permet à un utilisateur de voir l'état de toutes les systèmes connectées sur le réseau et de pouvoir générer un lien qui redirigerait l'accès à ce système sans avoir à consulter une table de correspondance. Il permet par la même occasion de voir les ports ouverts sur un système donné, avec la mise en place d'une base de données afin qu'ils puissent associer un nom à une IP et ainsi simplifier la lecture.

Lancement du script de mapping

```

194.254.30.90 - PuTTY
root@192:/home# ./PingTest.sh
192.168.1.1 : Systeme connecte !
192.168.1.2 : Systeme connecte !
192.168.1.3 : Systeme connecte !
192.168.1.4 : Systeme deconnecte ou inexistant !
192.168.1.5 : Systeme deconnecte ou inexistant !
192.168.1.6 : Systeme connecte !
192.168.1.7 : Systeme connecte !
192.168.1.8 : Systeme deconnecte ou inexistant !
192.168.1.9 : Systeme connecte !
192.168.1.10 : Systeme connecte !
192.168.1.11 : Systeme deconnecte ou inexistant !
192.168.1.12 : Systeme connecte !
192.168.1.13 : Systeme deconnecte ou inexistant !
192.168.1.14 : Systeme deconnecte ou inexistant !
192.168.1.15 : Systeme deconnecte ou inexistant !
192.168.1.16 : Systeme deconnecte ou inexistant !
192.168.1.17 : Systeme deconnecte ou inexistant !
192.168.1.18 : Systeme deconnecte ou inexistant !
192.168.1.19 : Systeme deconnecte ou inexistant !
192.168.1.20 : Systeme connecte !
192.168.1.21 : Systeme deconnecte ou inexistant !

```

Exécution du script de mapping

IP	Statut	Nom	Action	Internet	Local
192.168.1.1	On	Serveur d'application	<input type="button" value="Changer le nom"/>	Go !	Go !
192.168.1.2	On	Pas de nom	<input type="button" value="Changer le nom"/>	Go !	Go !
192.168.1.3	On	Serveur d'application	<input type="button" value="Changer le nom"/>	Go !	Go !
192.168.1.4	Off	Pas de nom	<input type="button" value="Changer le nom"/>		
192.168.1.5	Off	Pas de nom	<input type="button" value="Changer le nom"/>		
192.168.1.6	On	Thermomètre extérieur	<input type="button" value="Changer le nom"/>	Go !	Go !
192.168.1.7	On	Caméra Axis	<input type="button" value="Changer le nom"/>	Go !	Go !
192.168.1.8	Off	Pas	<input type="button" value="Changer le nom"/>		
192.168.1.9	On	Centrale de pompe	Nom <input type="text" value="Centrale"/> <input type="button" value="Modifier ce nom"/>	Go !	Go !
192.168.1.10	On	Pas de nom	<input type="button" value="Changer le nom"/>	Go !	Go !
192.168.1.11	Off	Pas de nom	<input type="button" value="Changer le nom"/>		
192.168.1.12	On	Pas de nom	<input type="button" value="Changer le nom"/>	Go !	Go !
192.168.1.13	Off	Pas de nom	<input type="button" value="Changer le nom"/>		
192.168.1.14	Off	Pas de nom	<input type="button" value="Changer le nom"/>		
192.168.1.15	Off	Pas de nom	<input type="button" value="Changer le nom"/>		
192.168.1.16	Off	Pas de nom	<input type="button" value="Changer le nom"/>		
192.168.1.17	Off	Pas de nom	<input type="button" value="Changer le nom"/>		

Interface de l'application de mapping

```

1  #!/bin/bash
2  #On réinitialise le fichier map.php contenant les données obtenues au sein de tous les systèmes
3  rm /var/www/map.php
4  #On écrit l'entête du fichier html
5  echo "<html>
6      <head>
7          <title>Index</title>
8          <link rel='stylesheet' type='text/css' href='style.css'>
9          <script language='javascript' src='script.js'></script>
10     </head><body>" >> /var/www/map.php
11 #On écrit les opérations de connexion à la base de données
12 echo "<?php \$connect=mysql_connect('localhost','root','mcs1ap31');
13 mysql_select_db('hostname',\$connect);
14 \$res=mysql_query('SELECT * FROM name ORDER BY "ID"');
15 ?>" >> /var/www/map.php
16 #On écrit l'entête du tableau
17 echo "
18     <center>
19         <table border=1>
20             <tr>
21                 <th>IP</th>
22                 <th>Statut</th>
23                 <th>Nom</th>
24                 <th>Action</th>
25                 <th>Internet</th>
26                 <th>Local</th></tr>" >> /var/www/map.php
27 #Pour chaque ordinateur du réseau
28 for (( ip=1;ip<255;ip++ ))
29 do
30 #On ping le système courant et on récupère son retour
31 ping=`ping -c 1 -W 1 192.168.1.$ip >/dev/null && echo 0 || echo 1`
32 #On écrit les données concernant le système courant
33 echo "<?php \$tab=mysql_fetch_array(\$res); ?>" >> /var/www/map.php
34 #On teste le retour du ping
35 if [ $ping = 0 ]
36 then
37 #On écrit dans le fichier les données correspondant au système courant et on informe qu'il est sous tension
38 echo -e "${BLUE}192.168.1.$ip : Systeme connecte !"
39 port=`expr 40000 + $ip`
40 #On écrit la ligne concernant le système
41 balise="<tr><tr><td><b><a href='ports.php?ip=192.168.1.$ip&start=1&end=65000'>192.168.1.$ip</a></b></td><td><fo
42 echo $balise >> /var/www/map.php
43 else
44 #On écrit dans les fichier les données correspondant au système courant et on informe qu'il est hors tension
45 echo -e "${RED}192.168.1.$ip : Systeme deconnecte ou inexistant !"
46 #On écrit la ligne concernant le système
47 balise="<tr><tr><td><b>192.168.1.$ip</b></td><td><font color=red>Off</font></td><td id='name$ip'><?php echo str_
48 echo $balise >> /var/www/map.php
49 fi
50 done

```

Code source du script de mapping réseau

XI.6 – Ressources humaines

XI.6.1 – Etudiants en BTS Electrotechnique

Dans la cadre du projet, il m'a été confié la charge de suivre et diriger un groupe d'étudiants qui étaient de 2^{ème} année de BTS Electrotechnique. Afin de pouvoir valider leurs diplômes, ils avaient pour objectif de présenter lors d'une soutenance orale, un projet qu'ils auraient mené et finaliser dans leurs domaines de compétences.

Pour cela, leurs professeur principal, en l'occurrence M. Patrice Michel leurs a proposé de participer au projet Diweb. Leurs principales tâches seront d'améliorer des systèmes communicants existants afin d'étendre leurs capacités de pilotage à distance et de développer de nouvelles IHM plus conviviales incluant de nouvelles fonctionnalités liés aux améliorations qu'ils apporteront sur leur système.

C'est donc avec enthousiasme qu'ils ont acceptés de collaborer avec moi. Et moi avec beaucoup de plaisir que j'ai eu à travailler avec eux que cela soit lors de mes interventions au Lycée ou lors de leurs stages qu'ils ont effectués au sein de la société DMS.

Voici les étudiants qui ont travaillé sur leurs systèmes ainsi que les tâches respectifs :

- Rémy David – Système SPEN
- Benjamin Jousot – Système Surrpressic
- Julien Cressence – Systèmes de pompage de piscine
- Thomas Balti – Système Tapiris

Lors de leurs stages en entreprise, j'ai pu leurs confier différentes tâches :

	Benjamin JOUSSOT	Julien CRESENCE	Rémi DAVID	Thomas BALTI
2				
3	Généralités			
4	xxx	x	xxx	xxx
5	xx	xx	xxx	xx
6	x	x	xxx	x
7	xx	x	xxx	x
8	x	x	xx	x
9	x	x	x	x
10	Connaissances informatiques réseaux et logiciels			
11	xxx	xx	xxx	xxx
12	x	x	xx	x
13				
14	x			xx
15				
16				
17				
18		x		
19		x		
20				
21	x		xx	x
22	Travaux sur les systèmes industrielles			
23				
24				
25				
26				
27				
28				
29				
30				

Je ne détaillerais pas l'ensemble des travaux qu'ils ont effectués dans le cadre du projet Diweb, notamment au niveau industriel des systèmes, je m'attarderais plutôt sur la partie de la refonte des IHM dont je prendrais le cas de Rémy David dans celui du système SPEN qui été pour moi celui le plus intéressant.

XI.6.2 – Etudiant en DUT Informatique

Pendant les phases de mise en place de la zone technique, je me suis rendu compte qu'étant le seul « informaticien » du projet m'handicapé énormément au niveau du projet Diweb. En effet, de par le temps que je passais à la préparation du projet (serveurs, zone, stages des étudiants, cahier des charges, spécifications, modélisation, etc ...).

Il ne me restait guère le temps pour passer au développement de celui-ci car je finissais prochainement la modélisation du système.

J'ai donc proposé à DMS de recruter des étudiants en Informatique afin de m'épauler durant cette lourde tâche. Celui-ci accepta donc je recrute des étudiants à l'IUT Informatique de Toulouse Rangueil. Je souhaitai aussi par ailleurs recruter un stagiaire en Infographie afin que celui travaille sur la partie graphique de l'application, ce qui aurait été un élément important dans la phase de développement. Malheureusement, m'étant pris trop tard pour cette période je n'ai pu trouver aucun étudiant en Infographie.

Après avoir fait passer plusieurs entretiens, j'ai accueilli deux étudiants en deuxième année chez DMS afin de renforcer notre équipe déjà présente. Ayant été très intéressé par le domaine d'application et les technologies mises en œuvre sur ce projet, c'est avec beaucoup de bonne volonté qu'ils ont pu participer au projet afin de le soutenir à l'oral pour obtenir leurs diplômes. A ces étudiants, je leur ai confié des tâches principalement de développement. J'ai en cela rédigé un planning sur l'ensemble des parties sur lesquels ils devaient travailler.

XI.7 – Ressources logiciels

Voir Annexe 6 pour consulter la liste des logiciels industriels et leurs utilités.

XI.8 – Tests et essais sur le système industriel Tapiris

Il m'a été permis d'approcher le système Tapiris afin de le manipuler et effectuer des essais dans le but de mieux comprendre le fonctionnement d'un automate programmable. Il est possible de consulter les résultats issus de cette manipulation dans l'annexe [4].

XXII. REDEVELOPPEMENT DES INTERFACES IHM

Afin de mieux comprendre le fonctionnement des interfaces et leurs interactions avec le système mécanique et opérative. J'ai suivi l'élève Rémy David durant la mise à jour de l'IHM de son système SPEN ainsi que la modification de la partie mécanique de celui-ci. Je vais détailler le travail qu'il a effectué après avoir rappeler quelques notions sur l'objectif et le principe de fonctionnement de ces interfaces.

XII.1 – IHM d'un système communicant

L'interface homme-machine, interaction humain-machine (IHM) définit, les moyens et outils mis en œuvre, afin qu'un humain puisse contrôler et communiquer avec une machine. Les ingénieurs en ce domaine étudient la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes qui soient ergonomiques, efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation.

Les IHM sont des pages pouvant contenir du texte, des images, des vidéos, des animations 3D ou 2D, des dessins. Elles permettent la présentation d'un système ou d'une unité de production complète, cette vision du système permet une utilisation plus claire des systèmes ainsi qu'une prise en main de n'importe quelle personne à n'importe quel endroit du moment que le terminal de dialogue soit communiquant avec un réseau (BUS : ProfiBus, LAN : Ethernet ...)

Ces IHM peuvent aussi simplifier la maintenance en indiquant l'emplacement du défaut directement sur l'IHM, ce qui permet de gagner du temps et ainsi réduire les coûts de maintenance du à l'arrêt de la production. Grâce à ce procédé on peut visualiser simultanément l'état du système en temps réel, visualiser des courbes, des mesures, des alarmes, des défauts. On peut aussi agir sur le process à distance.

XII.2 – Objectifs de l'étudiant Rémy David

Un besoin de refaire ces pages est apparu très vite dès lors que les objectifs du projet ont été définis. Ce projet doit avoir comme but principal l'aspect pédagogique et ergonomique des systèmes, or comme nous l'avons constaté précédemment les IHM présentes dans le pupitre opérateur sont nullement ergonomique.

Donc, Il faut que ces IHM est un aspect clair pour la compréhension du système, ces pages doivent- être pédagogique dans un but d'apprentissage, le référentiel de l'éducation nationale explicite bien le fait d'utiliser des supports pluri techniques réels locaux ou à distance.

La vision des pages à distance doit-être en « pleine page », le serveur web étant déjà présent dans le pupitre opérateur ce dernier doit supporter les nouvelles IHM, la société SPEN nous interdit l'accès au programme automate pour des raisons évidentes de garantie.

Le serveur web n'accepte pas la programmation HTML, le seul protocole supporté est le VPN (c'est une capture d'écran à intervalles rapprochés) programmable à l'aide du logiciel *WinCC Flexible* (présenté plus loin). Sur un écran d'ordinateur les IHM représentent un petit carrée, dans un coin de l'écran (résolution 320x240 pixels).

XII.3 – Caractéristiques du matériel :

Le pupitre opérateur permet une vision délocalisée du système grâce d'une part à sa liaison par bus de terrain vers l'automate et d'une autre part la vue globale du système à l'aide d'interface graphique. Ce pupitre tactile SIEMENS peut être raccordé à un réseau LAN grâce à une connectique Ethernet. Il embarque une mémoire de 40Mb pour son fonctionnement. La dimension de son écran (5,7'') est un désavantage pour la vision « pleine page » des interfaces.

XII.4 – Choix de la solution la plus appropriée :

Afin de répondre au mieux aux nouvelles exigences, l'étudiant a procédé par étape; tout d'abord il a organisé une réunion avec SIEMENS afin qu'il lui propose une solution. Cette solution consisté à changer le pupitre opérateur actuel par un pupitre de 12'', ce qui aurait permis de conserver les paramètres ainsi que les interfaces de bases.

Problème posé par cette solution est le budget d'un tel écran (env. 3000€). Il s'est donc ensuite rapproché de la société SPEN afin d'avoir un deuxième avis, nous avons convenus d'une réunion au lycée afin qu'il me présente leurs solutions ; pour la société SPEN, nous pouvons garder le pupitre opérateur actuel, il est juste nécessaire d'ajouter un *Runtime* (moteur d'exécution : Un moteur d'exécution est un programme ou une bibliothèque qui permet l'exécution d'un autre programme) ce qui permettrait l'affichage en pleine page tout en conservant la structure matériel présente.

Cette solution sera retenue car sa mise en place n'engendre pas de coût, le système est livré avec un dossier technique et pédagogique ainsi que tous les logiciels nécessaire à son utilisation. (*Runtime* et *WinCC Flexible advanced*). Une fois la solution acquise, il faut maintenant déterminer l'apparence des IHM ; nous avons pu constater que les interfaces de bases ne sont ni claires, ni ergonomiques.

XII.5 – Présentation du logiciel :

SIMATIC HMI offre une gamme complète permettant de couvrir toutes les tâches de contrôle-commande. SIMATIC HMI permet de maîtriser le processus à tout instant et de maintenir les machines et installations en état de marche.

Les systèmes SIMATIC HMI simples sont p. ex. de petites consoles à écran tactile mises en oeuvre sur site. A l'autre extrémité de la gamme SIMATIC HMI se trouve des systèmes utilisés pour la conduite et la surveillance de chaînes de production. Il s'agira en l'occurrence des puissants systèmes client-serveur.

Utilisation de SIMATIC WinCC flexible :

WinCC flexible est le logiciel IHM pour la réalisation, par des moyens d'ingénierie simples et efficaces, de concepts d'automatisation évolutifs, au niveau machine. WinCC flexible réunit les avantages suivants : simplicité, ouverture et flexibilité.

- *WinCC flexible Advanced :*

SIMATIC WinCC flexible est un logiciel IHM innovant sous Windows, qui convient pour toutes les applications au pied de la machine dans le domaine de la construction de machines, de machines de série et d'installations. La gamme des pupitres opérateur s'étend des Micro Panels conçus pour les applications avec des automates SIMATIC S7-200 à des solutions sur site avec des Panel PC ou des IPC SIMATIC.

WinCC flexible apporte une efficacité de configuration maximale : des bibliothèques contenant des objets préconfigurés, des blocs d'affichage réutilisables, des outils intelligents allant jusqu'à la traduction automatisée des textes dans le cadre de projets multilingues.

- *WinCC flexible Runtime :*

WinCC flexible Runtime est le logiciel de visualisation de processus. Dans Runtime, vous exécutez le projet en mode processus.

XII.6 – Prise en main du logiciel :

Afin de pouvoir créer de nouvelles interfaces il faut d'abord prendre en main le logiciel d'interface homme-machine de chez SIEMENS ; *WinCC Flexible Advanced*, qui offre, pour les applications au niveau machine un considérable gain d'efficacité dans la configuration ainsi que des concepts d'automatisation innovateurs.

Ce logiciel permet de créer, modifier, transférer, sauvegarder et restaurer des projets. Il intègre une bibliothèque graphique complète qui lui confère une simplicité d'utilisation accrue. Malgré sa prise en main complexe il reste relativement simple d'utilisation par la suite.

Pour ma part j'ai effectué des exemples fournis avec le programme mais aussi des nouveaux projets afin de bien comprendre son fonctionnement. Environ 1 mois a été nécessaire pour sa compréhension et son utilisation.

En ayant suivi, la programmation de ces interfaces⁴⁹, j'ai pu comprendre les problèmes récurrents que rencontre les techniciens du milieu de l'industriels et voir comment procéder. Ce fut une expérience enrichissante.

Pour autant, les méthodes de travail dans le milieu de l'industrie sont différents de ceux de l'informatique tertiaire. Ces interfaces sont axées sur l'informatique industrielle, mais presque tous les domaines de l'électrotechnique sont référencés, comme l'automatisme, le choix de matériel et le câblage.

L'informatique industrielle est particulièrement utilisée par le BTS maintenance industrielle, mais tend à rentrer de plus en plus dans le cursus du BTS Electrotechnique. Ce qui rend ce BTS très attractive grâce maintenant à son large domaine d'apprentissage.

XII.7 – Applet de connexion automate

Durant cette phase de développement, j'ai pu constater que tous les interfaces des systèmes communicants avaient un point commun. En effet, les serveurs web embarqués dans les systèmes ne sont pas apte à réaliser des traitements ou des compilations de fichiers. Ces systèmes sont limités par leurs capacités techniques et jouent principalement le rôle de stockage de fichiers. Lorsque on consulte une interface, le navigateur interprète un fichier web, de type html le plus souvent des cas, Le fichier HTML afin de pouvoir communiquer avec l'automate utilise une librairie écrite en Java qu'il communiquera par la suite dans un applet.

Donc le seul moyen de communiquer avec un automate avec un programme informatique est le Java. Ce qui rend la nécessité d'installer un runtime sur tous les postes élèves qui souhaiterais accéder à ces interfaces.

Cette information m'a permis de comprendre que les dispositions à prendre lors des développements de la plateforme est d'orienter celles-ci vers l'utilisation du langage Java. Afin de mieux comprendre le fonctionnement de ces applets, je me suis intéressé au système Tapiris qui est doté d'un automate TSX37 et d'un module web 5TZ510 du constructeur Schneider Télémécanique, les modules les plus faciles à exploiter.

Comme on veut utiliser des applets Java pour lire et modifier la valeur des variables dans la mémoire de l'automate pilotant le Tapiris. Ces applets sont stockées sur le serveur web embarqué du coupleur Ethernet TSX 510 de l'automate. Le système Tapiris est décrit un peu plus haut dans cette partie. Afin d'effectuer les tests, j'ai suivi un tutoriel⁵⁰ utilisant l'éditeur de codes Jbuilder.

Par exemple, on pourra lire et/ou modifier :

- Les codes de produits évacués au poste 1 ou au poste 2,
- Lire le nombre de produits évacués au poste 1 ou au poste 2,
- Régler les délais d'éjection des vérins ...

Remarque importante : L'exécution des applets construites avec JBuilder nécessitent une machine virtuelle Java Sun version 1.4 ou supérieur associée au navigateur web de consultation. Cette machine virtuelle est téléchargeable gratuitement sur le site de SUN (<http://fr.sun.com/>).

⁴⁹ Voir Annexe 7, présentation des différents IHM développés par l'étudiant.

⁵⁰ Voir Annexe 8 pour suivre le tutoriel de connexion à un automate via un applet Java

SIXIEME PARTIE : CONDUITE ET REALISATION

XIII – DEROULEMENT ET ORGANISATION

XIII.1 - Introduction

A ce stade du projet, je dispose de tous les éléments en main pour commencer la phase de développement. Pour cela, il me reste encore à définir quels sont les outils et technologies que je vais mettre en œuvre.

Au début, de cette phase j'ai pu être appuyé par l'arrivée de mes stagiaires en DUT Informatique sur une période d'environ 3 mois du mois d'avril à juin, ce qui a été une aubaine pour moi car j'avais pris du retard dans le déroulement du projet. Il me fallait optimiser au maximum le temps restant qu'il m'était accordé pour finaliser le projet.

J'ai donc redéfinis le planning général en détaillant les parties succinctes de développement. Afin de pouvoir le réaliser dans les meilleures conditions possibles, j'ai mis en pratiques les connaissances que j'ai pu acquérir lors des cours de génie logiciel et d'ingénierie des systèmes d'informations.

Ma formation d'ingénieur en systèmes d'information se base notamment sur ces principes. Je vais rappeler des notions fondamentales rapidement car j'ai déjà bien avancée celui-ci (Etude préliminaires, de faisabilité, et technique).

Afin de mener ce projet sérieusement, j'ai mise en avant mes compétences de gestionnaire afin d'avoir une directive bien précise. Il était important que durant le développement les objectifs soient claires et que tous les travaux à réaliser soient utiles. Je ne pouvais pas me permettre de perdre de temps vu qu'il m'en restait peu et que j'arrivais à terme à la fin de mon stage.

13.1.1- Méthodologie de développement

Avec deux étudiants à ma disposition pour le développement de la plateforme, j'ai souhaité avoir une approche rigoureuse et qualitative de mon développement.

Ayant un rôle proche de chef de projet, il me fallait mettre une place de méthodologie et des procédures de conception⁵¹. L'occasion était pour moi de mettre en avant mon savoir-faire dans ce domaine, et ceux acquises durant les cours. L'expérience a été par ailleurs très enrichissante et bénéfique car nouvelle.

A la suite d'une étude de comparaison de différentes méthodologies, mon choix s'est porté sur Scrum d'Agile. Il m'a paru la plus adapté à la situation.

⁵¹ Voir Annexe 9 pour une description des méthodologies de développement

XIII.2 – Préparation et mise en place

Ma première tâche était de revoir mon planning et de définir les tâches prioritaires que moi et les stagiaires informatiques auront à effectuer. Je pourrais ainsi par la suite définir et attribuer les parties de chacun.

Il me fallait aussi préparer et définir plusieurs éléments essentiels pour le développement de la plateforme, je ne souhaitais pas me lancer dans cette phase sans préparation au risque de perdre du temps cruciaux.

Ainsi les éléments nécessaires sont les suivants et dans l'ordre :

1. **Le planning détaillé**, celui intègre les sous-ensembles des principales tâches du projet.
2. **Une méthodologie de conception (ou de développement)**, utiliser une méthode pour favoriser et diriger la phase de conception.
3. **Les environnements de développement web**,
4. **Les outils de développement web**,
5. **L'architecture du système**, c'est à dire le noyau, les couches logicielles et la manière dont elles interagissent entre elles.
6. **Les données**, les principales données et objets de l'application (par ex. composant élève).
7. **Les phases et tâches de conception (ou développement)**, un digramme de Gantt sur les tâches.
8. **La modélisation des classes à développer**,
9. **La modélisation et la conception de la base de données**,
10. **Les interfaces graphiques (maquettes visuels)**, nécessaires avant le développement afin d'avoir une vue d'ensemble des futurs IHM de l'application.

Cette phase consiste à étudier différentes solutions ou architectures techniques et fonctionnelles en fonction de contraintes de compétences, d'équipement, de délais. Les choix doivent être ensuite validés par la réalisation de maquettes ou de prototypes et éventuellement la mise en place de la plateforme test.

Ces éléments en main, je pourrais passer en suite à la phase de développement.

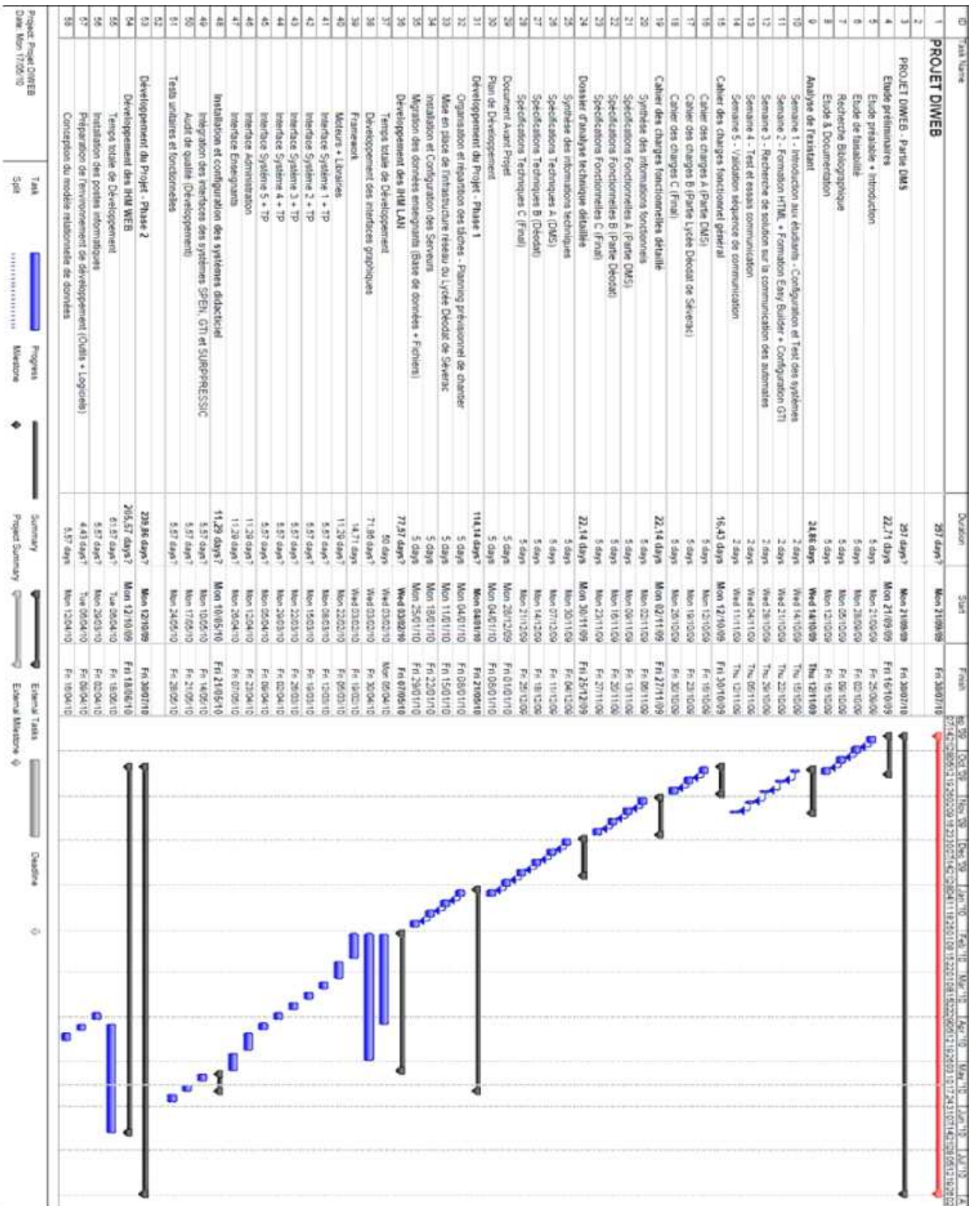


Fig.XIII.2 – Planning globale détaillé

XIII.3.4 – Environnement de développement

« *Un bricoleur ne peut pas travailler correctement s'il ne dispose pas de bons outils et adaptés à la tâche qu'il doit accomplir* », il en est de même avec le développement informatique. Des choix judicieux s'imposent si on souhaite réaliser l'application avec les meilleurs outils. Encore une fois, une contrainte de n'utiliser que des interfaces Open Source m'a été demandée par DMS.

Heureusement, le monde de l'informatique fournit une quantité importante d'outils dédié au développement. En raison de cela, il est difficile de faire ses choix. Je me contenterais donc d'utiliser les outils les plus populaires. Ce choix doit de plus être orienté dans une perspective de développement web. En effet, notre application est une application web. Toutes les méthodes et outils doivent être dans un objectif de développement informatique web, ceci dans tous les niveaux.

Par ailleurs, lors de la rédaction des spécifications techniques et avant que ne soit préconiser l'utilisation du langage PHP pour le développement de notre plateforme. J'avais proposé l'utilisation du langage Java et de ses outils. Après étude et tests, je me suis vite aperçu que celui-ci était limité dans une utilisation web dont je détaillerais la raison ci-dessous.

Tout simplement par sa simplicité de mise en œuvre et de coûts, les applications du web (internet et intranet) sont à 80%, des applications développées en Php. On ne compte le nombre de d'outils et de packages que comptent celui-ci. Je vais d'abord commencer par préciser quelques notions et concepts techniques propres à notre phase de développement.

XIII.3.4.1 – Les « Design Patterns »

Tout langage digne de ce nom propose de manipuler des objets. Mère des projets de développement les plus ambitieux, catalyseur des systèmes d'information les plus complexes et reine de la modularité, l'utilisation d'un design pattern est aujourd'hui importante.

Pour le développeur, elle est un véritable tremplin qui lui permettra de créer des applications de grande qualité. Après un rapide tour d'horizon des concepts de base liés aux « Design Patterns », nous nous intéresserons à son implémentation, puis aux outils de modélisation qui permettent de structurer efficacement un programme en amont.

Les design patterns (aussi appelés *motifs de conception*) représentent des structures objet. Programmer avec des objets n'est efficace que si leur structuration est bonne. Combien doit-il y'avoir de classes ? Quels objets possèdent quelles responsabilités ? Comment les objets interagissent-ils entre eux ? Comment faire que l'impact d'un changement dans le programme soit maîtrisé et limité ?

Toutes ces réponses sont données par les motifs de conception, qui assurent au final une application cohérente, découplée, testable et maintenable. Les motifs proposent des façons de concevoir son code ; ils sont reconnus dans le génie logiciel et indépendant du langage de programmation utilisé. Ils répondent à des problèmes connus et récurrentes du développement logiciel.

Comprendre les motifs de conception

Pour schématiser, on peut comparer les design patterns à un moteur de voiture. Celui-ci est ‘une étonnante complexité, et d’une incroyable précision. C’est un ensemble de pièces (d’objets) qui fonctionnent toutes ensemble. La manière dont fonctionnent ces pièces les unes avec les autres, et la responsabilité attribuée à chaque représentant un ensemble de motifs de conception.

Lorsqu’une pièce casse dans un moteur, on est capable de l’isoler. On est capable aussi de la remplacer, et même de la tester à part. Ce n’est que parce que chaque pièce possède un rôle unique, et parce qu’elles ont toutes été prévues pour fonctionner ensemble d’une manière précise et réfléchie, que le tout, le moteur, finit par tourner et faire avancer le véhicule.

Les design patterns représentant ainsi la manière dont chaque objet va réagir, soit individuellement, soit avec les objets voisins. Comme un moteur peut ne pas tourner rond – ou ne pas tourner du tout – si ses pièces sont mal assemblés, il en va de même pour une application : si ses objets sont mal conçus, ou s’ils n’arrivent pas à communiquer correctement ensemble, alors c’est toute l’application qui en paiera les conséquences.

Il est possible de créer une application web (raisonnable) dans un seul objet, tout comme il est possible de la créer avec des milliers d’objets. Dans les deux cas, l’application ne sera facile ni à maintenir, ni à tester. Depuis très longtemps, les architectes logiciel ont analysé et évalué des manières de concevoir les objets, de manière individuelle, ou dans un tout. Il en est ressorti des dizaines de façons remarquables de les faire communiquer, chacune répondant à un problème précis et récurrentes : chacune de ces façons représente alors un motif de conception. Le but ultime est d’obtenir un programme à forte cohésion et à faible couplage.

XIII.3.4.2 – Le langage de programmation, PHP ou Java ?

Pendant un long moment, j’avais préconisé l’utilisation du Java, non par choix technique et stratégique pour le projet, mais je dois l’avouer pour moi-même, même cela était égoïste de ma part, je souhaitais approfondir mes connaissances en Java et découvrir de nouveaux outils à mettre en œuvre.

Par un souci de gain de temps notamment, je souhaitais implémenter un outil de base afin d’améliorer et d’ajouter des fonctionnalités propres à Diweb. Lors de la rédaction des spécifications techniques, je me suis installé un serveur sous Tomcat, serveur d’application web Java, afin de pouvoir tester plusieurs outils en Java.

L’idée de départ était de partir sur un outil existant qui servira de base à notre plateforme. Après une longue recherche sur le web, je me suis aperçu rapidement qu’il n’existait que peu d’outils dédiés à la programmation orienté web en Java.

Parmi celle-ci, Je me suis penché sur deux outils en particuliers Jetspeed et Jahia qui proposent des fonctionnalités importantes et basiques dès le départ, comme par exemple, la gestion des utilisateurs, des pages, des contenus et autres.

Des fonctionnalités que je n'aurais pas à développer dès le départ et qui m'aurait fait gagner du temps pour le développement. Ces fonctionnalités permettront une plus grande souplesse dans l'administration de la plateforme par les utilisateurs, en l'occurrence les enseignants.



Jahia ou Jetspeed, le combat !

JAHIA

Jahia est un serveur applicatif remplissant les fonctions de portail d'entreprise, de serveur de gestion de documents, de serveur de gestion d'activité, de moteur de recherche et de suite collaborative. Jahia met à disposition des utilisateurs un système d'édition et de contribution en ligne (directement en front office), un support multi-langue pour le contenu et l'interface.

Jahia possède également au niveau de son fonctionnement un système de traitement des workflows et des activités, une gestion multi-site avec support du clustering, met à disposition des portlets prêts à l'utilisation (forum de discussion, calendrier, etc.), supporte l'authentification unique (SSO⁵²) avec une connexion LDAP⁵³ ou Active Directory, et dispose d'un serveur de gestion du cache pour améliorer les performances sous forte charge.



JETSPEED

Apache Jetspeed-1 est un outil de portail Web Java libre développé par la Fondation Apache depuis 1999, ce qui en fait l'un des premiers outils de portail Web. Jetspeed est à la fois un portail opérationnel et un framework de portail. Le fonctionnement est du type « MyYahoo », c'est à dire qu'il permet à chaque utilisateur d'agencer ses pages à sa guise. Notons que Jetspeed est entièrement basé sur une configuration Xml, et ne nécessite pas de base de données pour fonctionner - même si une base de données est préférable pour la tenue en charge en environnement de production.

⁵² *Single Sign-On*, méthode permettant à un utilisateur de ne procéder qu'à une seule authentification

⁵³ *Lightweight Directory Access Protocol*, protocole permettant l'interrogation et la modification des services d'annuaire.



Jahia souffre d'une certaine complexité dans sa configuration sur serveur de production (notamment sous websphere d'IBM) et dans sa mise en œuvre. De plus, je me suis rendu-compte après que Jahia est distribuée sous une licence. Il existe une version libre mais celui-ci est très limité. Même si celui-ci aurait pu être fort intéressant, il ne m'était pas permis d'utiliser Jahia.

Me restant que Jetspeed à exploiter, j'ai consulté les informations autour de celui-ci, la première chose est de constater que la communauté autour de cet outil est peu nombreuses, on y trouve donc peu de mises à jour ainsi que peu d'informations à leurs mise en œuvre. Ce qui peut s'avérer problématique lors des phases de développement.

L'interface est Jetspeed est sobre et peu efficace, de plus, par l'utilisation de son système de portlets, Jetspeed est peu souple au niveau de son code informatique voire même complexe à mettre en œuvre. Jetspeed n'est d'ailleurs plus mise à jour depuis 2007.

Download Jetspeed-2 Distribution

Jetspeed-2 is distributed in several formats for your convenience and distributed under the Apache License, version 2.0 .

There are two kinds of binary distributions. Neither one is dependent on the other. The demo is best for seeing more features. The minimal installer could be useful for basing your own portal distribution upon, or just seeing the basic feature set of Jetspeed.

1. Minimal Installer and EAR - only contains a Jetspeed Portal and the Jetspeed Administrative Portlets, with a minimal Site map and set of users
2. Full or Demo Installer and EAR - contains a Jetspeed Portal, the Jetspeed Administrative Portlets, a full demo Site Map, 10 users. Additionally, there several portlet applications including an RSS application, a Database Browser application, a Web Content application, and a set of demo portlets including Google Maps, weather portlets, and lots of programming examples to get you started.

Distribution	Mirrors	Checksum	Signature
Jetspeed-2 Standard with only the required Administrative Portlets	jetspeed-installer-minimal-2.2.0.jar	here	here
Jetspeed-2 Demo With lots of demo Portlets: RSS, Database Browser, Google Maps, Portlet Events, Web Content and many more	jetspeed-installer-demo-2.2.0.jar	here	here

Both the installers support the following databases for Jetspeed: Derby (default), DB2, MySQL, MSSQL, Oracle, PostgreSQL, SapDB, as well as manual (do it yourself) configuration of other databases.

An Ant script is provided for after installation, reinitializing or switching to another database in a one simple step operation.

Complete instructions for getting started using the installer is available [here](#).

Interface de gestion de Jetspeed

Conclusion :

Je me suis donc rendu-compte que le langage ne correspondait pas à ce à mes attentes. Je me suis résigné à l'utilisation du Java pour celui du Php. Connaissant particulièrement bien ce langage et sachant que ce choix technique sera en ma faveur par la suite. J'ai décidé de ne pas prendre un outil comme base de développement et de partir d'une base vierge de développement.

Cela va me demander de développer des fonctionnalités basiques comme l'authentification, pages et autres. Je me contenterais de développer les fonctions essentielles de la plateforme sans les autres (gestion des pages, des contenus etc....) ? L'important est de fournir un prototype fonctionnel qui

sera amélioré par la suite. J'utiliserais tout de même le langage Java si j'avais à développer des applets pour les serveurs web des systèmes communicants.

XIII.3.4.3 - Php

Le choix que j'ai fait dans l'utilisation du langage Php est que celui-ci est un programme qui a séduit des millions de développeurs. La manière dont il a été développé est en partie responsable de cette popularité. Voici quelques caractéristiques stratégiques de PHP :

- PHP est souple et permissif, il propose un confort d'utilisation inégalé alors que d'autres langages nécessitent beaucoup de rigueur et de connaissances théoriques ;
- PHP fonctionne aussi bien indépendamment qu'en couple : on peut utiliser l'exécutable PHP tout seul, ou l'employer à travers un autre exécutable, tel qu'Apache ;
- PHP est facilement extensible, il accepte de multiples fonctionnalités à travers son système d'extensions. Cela augmente le nombre de classes et de fonctions natives disponibles, donc de fonctionnalités possibles ;
- PHP est optimisé pour économiser des ressources, que ce soit celles de la machine ou celles du développeur. Cela augmente les performances de la machine ... et du développeur.

Principe de fonctionnement

Comme nous le disions, PHP peut fonctionner de manière autonome ou couplé à un autre programme. Le plus fréquemment, PHP est associé à un serveur HTTP comme Apache. Ce dernier gère la couche *web* de l'ensemble, c'est-à-dire les interactions avec les visiteurs, via le protocole http, tandis que PHP gère la couche dynamique, c'est-à-dire la génération du code (x)HTML et la gestion des données utilisateurs.

Ce type d'utilisation est la plus simple. Il consiste simplement, comme illustré sur la figure F-1, à utiliser un exécutable (php.exe sous Windows, php sous Unix/Linux) pour exécuter du code PHP. Il est intéressant dans le cadre du développement, notamment si celui débute à partir d'une base, de fournir un framework qui sera capable de donner une directive au code.

XIII.3.4.4 – Comparatif de framework PHP

Les framework⁵⁴ sont aujourd'hui des outils stratégiques qui favorisent les bonnes pratiques : gagner du temps, réutilisé, testé, stabiliser.

Un framework se greffe sur un langage. On peut considérer le langage comme étant de *bas niveau* et le framework comme étant de *haut niveau*, car les ressources offertes par le framework sont plus proches du raisonnement humain (métier) que celles proposées par le langage.

Concrètement, ce la signifie que pour développer une même fonctionnalité, on aura besoin de moins de lignes de code avec un framework que sans.

⁵⁴ Kit de composant logiciels structurels

Les objectifs pratiques d'un framework sont multiples :

- Fournir un ensemble de composants fiables et testés ;
- Favoriser le travail à plusieurs en instaurant des règles de programmation cohérentes ;
- Proposer une méthodologie et une organisation du travail qui permettent de maintenir un code source organisé.

Ainsi, le framework trouve toute son utilité en particulier dans le monde de l'entreprise, puisqu'il permet :

- *Aux développeurs* : d'utiliser des composants communs que tout le monde connaît, en évitant ainsi de réinventer ce qui existe déjà ;
- *Aux chefs de projet* : de fiabiliser les ressources humaines et d'organiser des développements sur la base d'un outil commun et maîtrisé
- *Aux architectes et ingénieurs logiciel* : de bâtir des modèles basés sur la programmation orientée objet et une organisation pratique des classes, des interfaces, des paquetages et des composants

Le framework permet d'homogénéiser les pratiques de développement au sein d'une équipe de projet, en proposant *une base de travail* et des *conventions* prédéfinies.

De plus, il favorise la réutilisation de par son organisation et ses ressources prédéveloppées. Il en résulte *un gain de temps et de qualité* extrêmement important dans le cadre de développements professionnels.

XIII.3.4.5 – Inconvénients des framework PHP

Nul n'est parfait et le framework n'échappe pas à cette règle. Tout d'abord, sa manipulation peut être difficile à apprendre et à maîtriser, en particulier pour les développeurs qui ne connaissent pas bien la programmation objet. En effet, un framework comme Zend Framework exploite largement le modèle objet de PHP, avec, entre autres, de nombreux design patterns, concepts objets souvent assez poussés, qu'il convient de bien comprendre.

Ensuite, étant donnée qu'il se situe à un niveau au dessus de PHP et qu'il fait appel à des technologies très diverses (XML, JSO⁵⁵N, http, SQL ...) il est important de maîtriser tout cela avant de se lancer dans la découverte de ses composants.

Car s'il est certes possible de conduire une voiture sans en connaître tous les rouages techniques, comprendre son fonctionnement permet une meilleure conduite, plus économique, et lorsqu'une panne arrive, on sait en général la réparer, ou tout du moins la localiser.

⁵⁵ Javascript Object Notation : outil de description de données issu du javascript

Enfin, l'impact d'un framework sur les performances du serveur est loin d'être négligeable. Il est ainsi nécessaire de bien appréhender le fonctionnement du framework, celui de PHP et de tout son écosystème, notamment le serveur web, afin de faire face à cette difficulté qui trouve heureusement de nombreuses solutions.

XIII.3.4.6 – En résumé

Il faudra retenir au moins l'essentiel : sa définition et ses quatre rôles fondamentaux. Un framework est une base de travail qui permet d'organiser et d'accélérer les développements. Il permet de :

- Fournir des composants réutilisables ;
- Proposer une architecture, c'est-à-dire des règles d'organisation pour les dossiers, les fichiers et ce qu'ils doivent contenir ;
- Fournir des conventions de développement ;
- Fournir éventuellement des outils de développement et de maintenance.

XIII.3.4.7 – Technologies du web

L'utilisation de Php pour le développement n'est qu'une des technologies employés pour développer des applications web. Mais l'utilisation simple de Php limite les possibilités d'interaction avec l'utilisateur et n'apporte que peu de convivialité et de dynamisme aux interfaces. Voici quelques précisions sur ces technologies du web :

XIII.3.4.7.1 - HTML

L'**Hypertext Markup Language**, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des *applets*.

Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

```

1 <html>
2 <head>
3 <title>Cours HTML - APerezMas</title>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
5 <meta name="author" content="Arsène Perez-Mas">
6 <meta name="description" content="Cours HTML pour débutants">
7 <meta name="keywords" content="HTML">
8 <script language="JavaScript" src="../Communs/javascripts/isnav.js"></scr
9 <script language="JavaScript" src="../Communs/javascripts/fonctions_dates
10 <script language="JavaScript" src="../Communs/javascripts/notes.js"></scr
11 </head>
12 <body bgcolor="#FFFFFF" background="../Communs/fonds/fond_courge_APM_esta
13 <a name="debut"></a>
14 <table width="47%" cellpadding="5" align="center">
15 <tr>
16 <td width="12%">
17 <h2 align="center"><br>
18 <a href="../index.html" target="Hors"><br>
19 </h2>
21 </td>
22 </tr>
23 </table>
24 </body>
25 </html>

```

XIII.3.4.7.2 – Javascript

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur¹. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.

```

10 $.fn.ajaxNavigation = function(params){
11 /*
12  * Extending parameters
13  */
14 var params = typeof params == 'object' ? params : {};
15 params = $.extend({
16   basepath: '',
17   event: 'click',
18   not: ['href^=#'], 'a.noajax', 'a.ajaxified',
19   onStart: function($to, href, hash){},
20   onExecution: function($to, href, hash, data){},
21   onComplete: function($to, href, hash){},
22   to: '#general-container'
23 }, params);
24
25 /*
26  * Main function called as callback (can be entirely hooked in the future)
27  * Its power: everything is pluggable
28  */
29 var _main_callback = function($to, href, hash){
30   hash = typeof hash == 'undefined' ? '' : hash;
31   $to.data('hash', hash);
32   params.onStart.call(this, $to, href, hash);
33
34   $.ajax({
35     type: 'GET',
36     url: href,
37     success: function(data){ params.onExecution.call(this, $to, href, hash, data); },
38     complete: function(data){ params.onComplete.call(this, $to, href, hash); }
39   });
40
41   console.log('onComplete: %s (#%s)', $to.data('initialized'), $to.data('hash'));
42 }

```

Il existe une bibliothèque JavaScript fortement intéressante à mettre en œuvre dans notre application. Celui-ci a la particularité de donner beaucoup de dynamique dans notre application même si il reste complexe à utiliser.

jQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant AJAX⁵⁶) et HTML, et a pour but de simplifier des commandes communes de JavaScript. jQuery contient notamment les fonctionnalités suivantes :

⁵⁶ *Asynchronous Javascript And Xml* : librairie Javascript pour du contenu web dynamique

- Parcours et modification du DOM⁵⁷ (y compris le support des sélecteurs CSS 1 à 3 et un support basique de XPath) ;
- Événements ;
- Effets et animations ;
- Manipulations des feuilles de style en cascade (ajout/suppression des classes, d'attributs...)
- AJAX ;
- Plug-ins ;
- Utilitaires (version du navigateur...).



XIII.3.4.7.3 – Css (Cascading Style Sheets)

CSS (Cascading Style Sheets : feuilles de style en cascade) est un langage informatique qui sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C).

L'un des objectifs majeurs des CSS est de permettre la mise en forme hors des documents. Il est par exemple possible de ne décrire que la structure d'un document en HTML, et de décrire toute la présentation dans une feuille de style CSS séparée.

Les styles sont appliqués au dernier moment, dans le navigateur web des visiteurs qui consultent le document. Cette séparation fournit un certain nombre de bénéfices, permettant d'améliorer l'accessibilité, de changer plus facilement de présentation, et de réduire la complexité de l'architecture d'un document.

<pre> h1 h1 a, h1 a:link h1 a:visited h1 a:hover h2 h2 a, h2 ...2 a:visited h2 a:hover h3 h3 a, h3 ...3 a:visited h3 a:hover h4 h4 a, h4 ...4 a:visited h4 a:hover h5 h5 a, h5 ...5 a:visited h5 a:hover </pre>	<pre> 26 h1 { 27 color: #8000FF; 28 font-size: 2.5em; 29 margin-left: 0.5em; 30 font-weight: bold; 31 } 32 h1 a, h1 a:link { 33 color: #8000FF; 34 border-bottom-style: dotted; 35 border-bottom-width: 1px; 36 } 37 h1 a:visited { 38 color: #a900ff; 39 border-bottom-style: dotted; 40 border-bottom-width: 1px; 41 } 42 h1 a:hover { 43 color: #a900ff; 44 border-bottom-style: solid; 45 border-bottom-width: 1px; 46 text-decoration: none; 47 } </pre>
---	---

⁵⁷ Document Object Model : interface indépendante de description

XIII.3.4.7.4 – Les Web Services

L'utilisation des web services est importante dans le cadre de notre projet Diweb, en effet, il est important de noter que pour dialoguer avec un système communicant, nous procédons de 3 manières suivantes :

- On accède à l'IHM du système via son adresse IP, à partir de là il est possible de communiquer avec le système. Ce moyen nous oblige donc à utiliser les interfaces développées par les industriels à partir des logiciels de conception d'IHM fournis par les constructeurs (ex. Factory Cast).
- On peut développer des IHM via des applets Java, ceci nous permet de personnaliser nos propres interfaces mais nous oblige à utiliser la librairie Java du constructeur afin de pouvoir communiquer avec l'automate et d'héberger les pages dans un serveur web. Nous ne pouvons pas utiliser la librairie à l'extérieur car celle-ci est protégée par licence.
- On peut développer utiliser des interfaces personnelles et communiquer avec des automates sans stocker les pages dans un serveur mais en utilisant les passerelles. Ces passerelles peuvent recevoir des ordres d'actions sous forme de web services et les transmettre à l'automate. Ceci à l'avantage de ne pas se munir de la librairie Java du constructeur mais on est limité aux fonctions proposées par le web services de la passerelle ou serveur. De plus, c'est option qui présente que dans certaines gammes de produits de type passerelle et serveur industriels.

XIII.3.4.7.5 - Interface client/serveur SOAP/XML

Les services web est un ensemble d'outils qui communiquent les applications par Internet en tenant compte des contraintes suivantes :

- Communiquer entre les applications.
- Etre compatible avec le web et Internet.
- Utiliser des solutions non propriétaires basées sur des normes et des standards.
- Faciliter l'implémentation.
- Les applications doivent communiquer quel que soit :
 - Le langage dans lequel elles ont été développées.
 - Le système d'exploitation sur lequel elles s'exécutent.

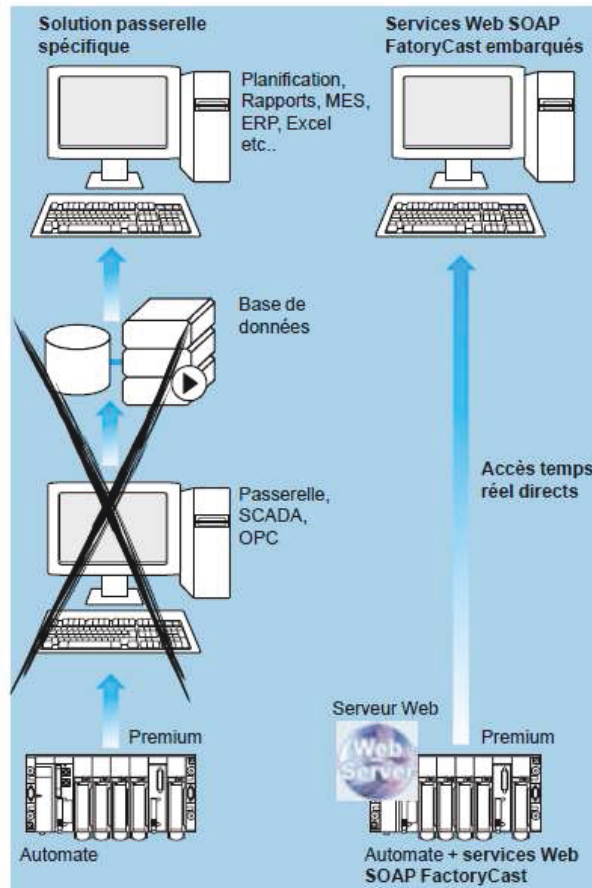
La technologie, compatible avec l'utilisation d'Internet, est basée sur un protocole standard SOAP XML (*Simple Object Access Protocol*) cohabitant avec HTTP, elle permet la communication entre les applications.

Une description standard des services et des interfaces est fournie par une application WSDL (*Web Service Description Language*) qui est une bibliothèque de fonctions d'accès aux données standardisées et auto documentées.

L'association de ces technologies est communément appelée "services web", de plus elle est indépendante des plates-formes, des langages et des systèmes d'exploitations.

Afin de disposer d'une totale interopérabilité, le logiciel Factory Cast HMI implémente les Web services de type SOAP/XML tels que :

- Fonction serveur capable de répondre aux demandes SOAP générées par toute application client.
- Fonction Client prenant l'initiative d'envoyer une demande SOAP à une application serveur SOAP.



Interface client / serveur SOAP/XML

Fig. XIII.3.4.7.5 - Schéma de fonctionnement des services web d'un coupleur automate/serveur web

XIII.3.4.7.6 - Présentation générale des Web Services

Il s'agit d'une technologie permettant à des applications de dialoguer via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent.

Pour ce faire, les Web services s'appuient sur un ensemble de protocoles Internet très répandus (XML, http), afin de communiquer. Cette communication est basée sur le principe de demandes et réponses, effectuées avec des messages XML.

Les Web services sont décrits par des documents WSDL (Web Service Description Language), qui précisent les méthodes pouvant être invoquées, leurs signatures et les points d'accès du service (URL, port).

Pour ce faire, le langage définit 4 éléments principaux qui sont :

- « **portType** » cet élément décrit le Web service, les actions qu'il peut s'exécuter et les messages qui lui sont associés. Il peut être comparé aux prototypes des fonctions dans un langage de programmation traditionnel.
- « **message** » cet élément définit les données d'une action et peut être constitué de plusieurs parties. Il peut être comparé aux arguments d'une fonction dans un langage de programmation traditionnel.
- « **types** » cet élément décrit les types de données utilisées par le Web service en utilisant des schémas XML. Il doit absolument se placer au début d'après le profil basique.
- « **binding** » cet élément définit le format des messages et le protocole de communication utilisé pour chaque Web service

Il y a plusieurs définitions des Web Services, mais elles ont toutes ces points en commun :

- les Web Services proposent aux utilisateurs du Web des fonctionnalités pratiques grâce à un protocole Web standard
- les Web Services offrent un moyen de décrire leurs interfaces suffisamment en détail pour permettre à un utilisateur de créer une application cliente capable de converser avec eux
- les Web Services sont inscrits afin que les utilisateurs potentiels puissent les trouver facilement. Ceci est possible grâce à UDDI (Universal Discovery Description and Integration)

Il existe plusieurs avantages à utiliser les Web services :

- Les Web services fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes,
- Les Web services utilisent des protocoles standards ouverts (HTTP, SOAP, UDDI)
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges
- Basés sur le protocole HTTP, les Web services peuvent fonctionner au travers de nombreux pare-feux sans nécessiter de changements sur les règles de filtrage
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les Web services existants

Les principaux inconvénients des Web services sont les suivants :

- Les normes de Web services dans certains domaines sont actuellement récentes
- Les Web services souffrent de performances faibles comparées à d'autres approches de l'informatique répartie telles que RMI, CORBA, ou DCOM
- Par l'utilisation du protocole HTTP, les Web services peuvent contourner les mesures de sécurité mises en place au travers des pare-feu

Il existe trois technologies derrière le terme Web services, le RPC-XML, le SOAP et les Web services RESTful. Mais nous allons limiter notre étude aux Web services SOAP. Même si les autres protocoles sont intéressants, les serveurs web Schneider ne communique qu'avec ce protocole via son logiciel Factory Cast.

Suite aux besoins de l'application et après analyse des différents Web services, il a été décidé d'utiliser les Web services de type SOAP. Le volume de données qui transitera restant relativement faible, les performances seront similaires à celle que nous connaissons sur les autres projets et conviennent au contexte du projet.

XIII.3.7.7 - Les Web Services de type SOAP

SOAP (Simple Object Access Protocol) est un standard développé par le W3C (World Wide Web Consortium). Ce protocole repose entièrement sur le langage de description XML. Son objectif est de définir la structure générale des messages échangés entre les composants Web services, sans pour autant en définir la structure du contenu. En ce sens, c'est un protocole peu restrictif, qui laisse aux composants Web services le soin de définir comment ils formateront le contenu du message.

Même si, dans la pratique, les messages SOAP sont utilisés dans un mécanisme de requête-réponse, cela ne constitue pas leur vocation première. En effet, même si la relation entre un client et son fournisseur suit ce mécanisme, il est important de noter que les messages SOAP échangés sont dits « one-way », car ils n'ont pas de relation entre eux. Au delà de la structure XML classique, on y trouve les blocs :

- « **Envelope** » c'est lui qui contient le message et ses différents sous-blocs. Il s'agit du bloc racine XML. Il peut contenir un attribut « encodingStyle » dont la valeur est une URL vers un fichier de type XML qui décrira les types applicables au message SOAP
- « **Header** » c'est un bloc optionnel qui contient des informations d'en-têtes sur le message. S'il est présent, ce bloc doit toujours se trouver avant le bloc « Body » à l'intérieur du bloc « Envelope »
- « **Body** » il contient le corps du message. Il doit absolument être présent de manière unique dans chaque message et être contenu dans le bloc « Envelope ». SOAP ne définit pas comment est structuré le contenu de ce bloc. Cependant, il définit le bloc « Fault » qui peut s'y trouver
- « **Fault** » ce bloc est la seule structure définie par SOAP dans le bloc « Body ». Il sert à reporter des erreurs lors du traitement du message, ou lors de son transport. Il ne peut apparaître qu'une seule fois par message. Sa présence n'est pas obligatoire

Ici, un fournisseur Web service (Service Provider) s'inscrit initialement auprès de l'annuaire UDDI (Service Broker) en se servant du protocole WSDL pour décrire ses fonctionnalités. Une fois cette étape effectuée, n'importe quel client, comme celui présenté sur le schéma (Service Requester), peut interroger l'annuaire UDDI afin de trouver le fournisseur Web service adéquate pour exécuter une action voulue.

Une fois le Web service trouvé, le client n'a plus qu'à lui envoyer la demande d'exécution de l'action à l'aide du protocole SOAP afin que celui-ci s'y emploie et envoie la réponse au client. Cela donne alors une liaison totalement indépendante et dynamique entre les clients et les fournisseurs de services Web.

XIII.4 – Environnement de développement

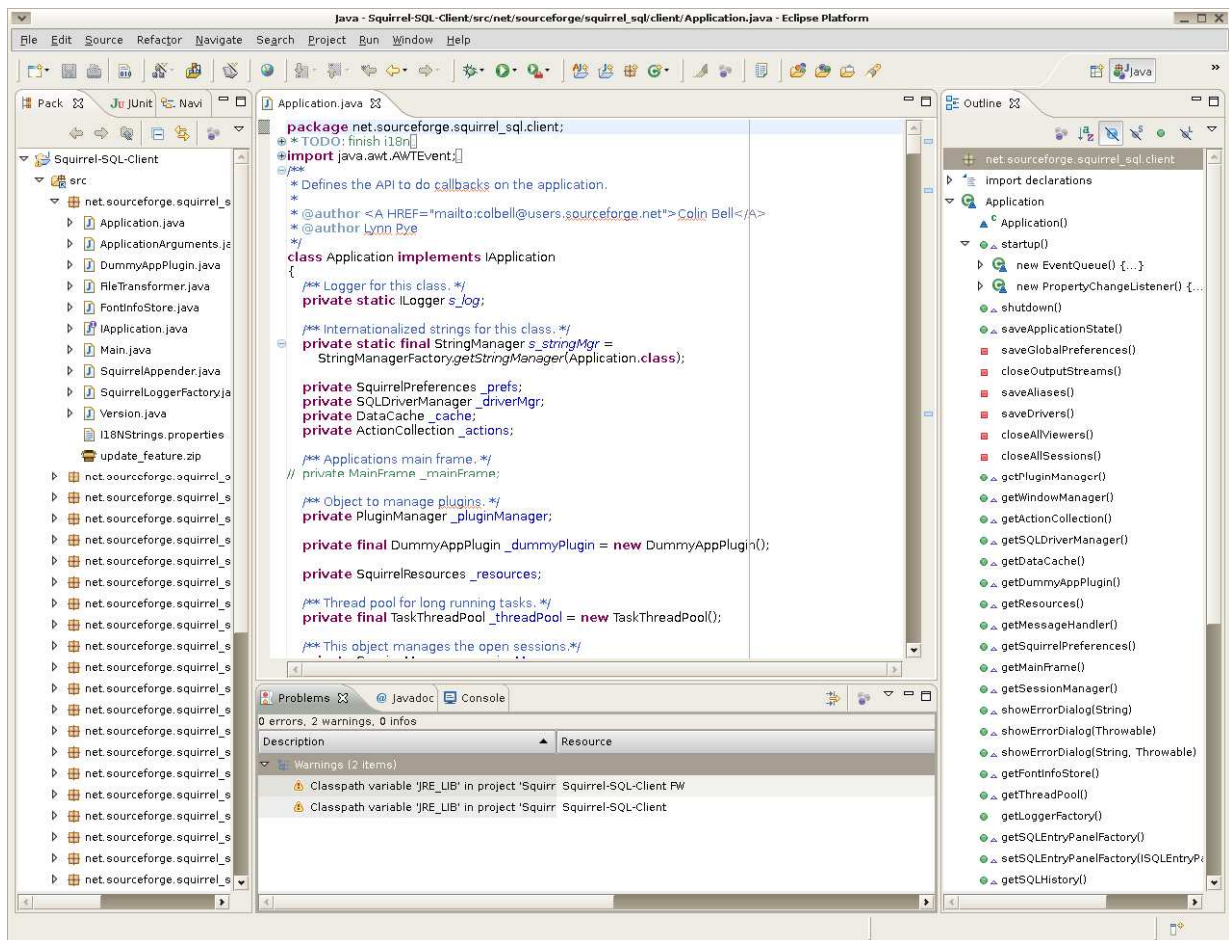
Il est inutile de rappeler l'importance de choisir les outils qui vont nous permettre de réaliser cette plateforme. Sans rentrer en détails, concernant le choix des outils que l'on va employer pour le développement, je vais faire une présentation rapide de ceux-ci.

Ces outils ont été choisis pour leurs popularités et leur performance largement reconnus dans la communauté des développeurs. Leurs utilisations sont par ailleurs libres (pas de licence à payer).

Eclipse

Eclipse est un projet de la Fondation Eclipse visant à développer tout un environnement de développement libre, extensible, universel et polyvalent.

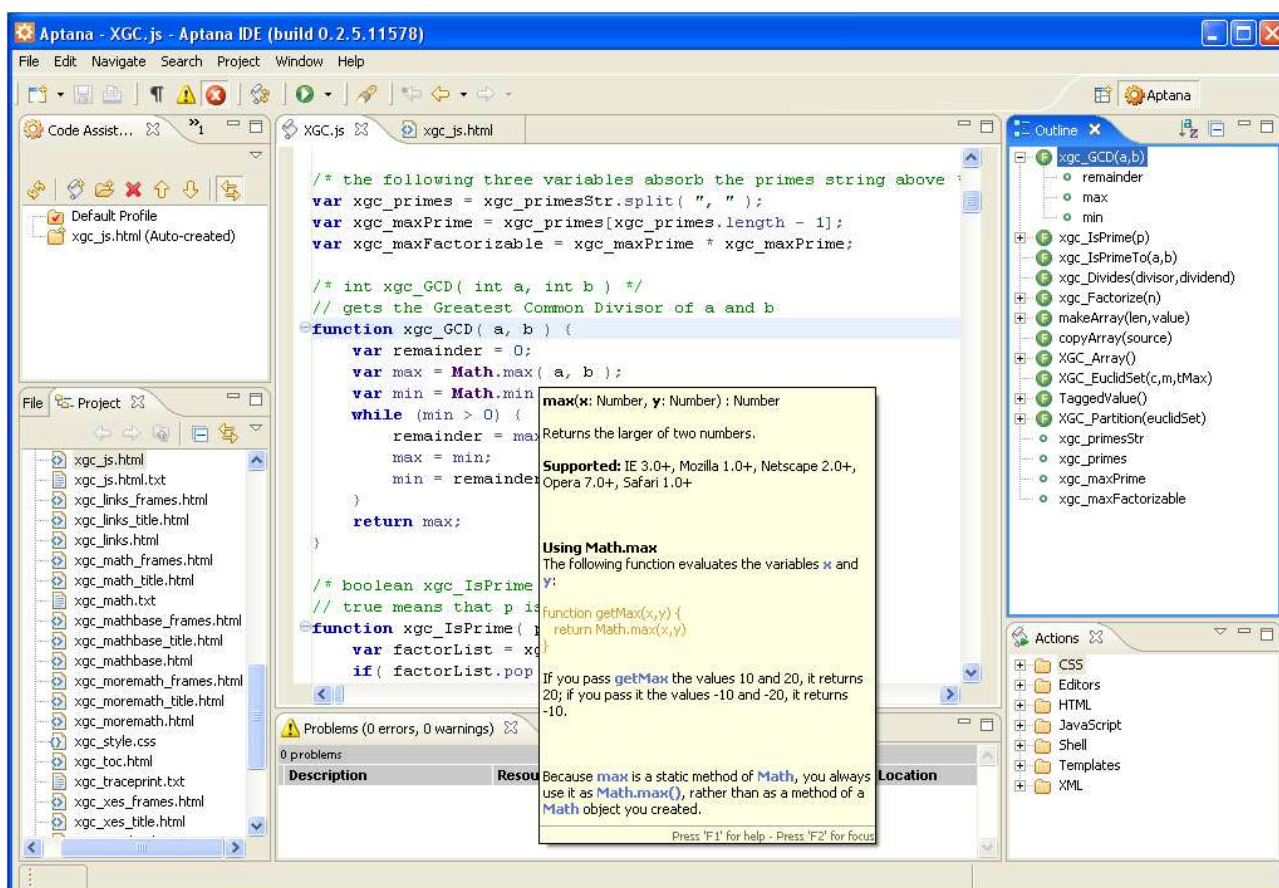
Son objectif est de produire et fournir divers outils gravitant autour de la réalisation de logiciels, englobant les activités de codage logiciel proprement dites (avec notamment un environnement de développement intégré) mais aussi de modélisation, de conception, de test, de reporting, etc. Son environnement de développement notamment vise à la généralité pour lui permettre de supporter n'importe quel langage de programmation.



Aptana

Aptana Studio est un environnement de développement intégré orienté web, multi-plateforme et open-source. Il facilite l'écriture du code en fournissant des aides à la saisie pour le JavaScript, l'HTML, les CSS, PHP et Python.

Aptana est disponible en version autonome ou bien en plugin pour son environnement d'origine : Eclipse. Même si celui-ci existe en version plugin, il est bien moins performant que sa version autonome. Pour avoir développé du javascript et Css à plusieurs reprises, je préfère utiliser la version complète et autonome car j'estime que pour chaque type de langage, un outil doit lui être associé et non pas un outil universel. Par ailleurs, sa ressemblance avec Eclipse facilitera sa prise en main.



Tortoise SVN

TortoiseSVN est un des logiciels client de SVN les plus populaires. C'est un logiciel libre distribué selon les termes de la licence GNU GPL. En s'intégrant dans l'explorateur de Windows, il offre aux utilisateurs de Windows une interface graphique permettant de réaliser la plupart des tâches qu'offre SVN en ligne de commande.



Wamp Server

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un *tray icon* (icône près de l'horloge de Windows).



Mozilla Firefox

Mozilla Firefox est un navigateur Web gratuit, développé et distribué par la Mozilla Foundation avec l'aide de centaines de bénévoles grâce aux méthodes de développement du logiciel libre/open source et à la liberté du code source.



PhpMyAdmin

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases de données, de nombreuses requêtes comme les créations de table de données, les insertions, les mises à jour, les suppressions, les modifications de structure de la base de données. Ce système est très pratique pour sauvegarder une base de données sous forme de fichier .sql et ainsi transférer facilement ses données.

De plus celui-ci accepte la formulation de requêtes SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.



XIII.5 – Architecture du système

XIII.5.1 – Type d'application

Il existe deux types d'applications différentes dont l'usage peut varier en fonction du domaine d'utilisation et des exigences techniques liées.

- Application dite « lourde »

On appelle une application lourde un logiciel installé localement sur le poste de l'utilisateur. Celle-ci a l'avantage de proposer un grand nombre de possibilités techniques, mais l'inconvénient d'être la moins portable, car dépendantes de l'environnement sur lequel elle se trouve.

Elle implique un effort supplémentaire lors du déploiement sur différents postes d'utilisateurs dû à l'installation du système et de ses composants. Elle peut rencontrer certains problèmes d'incompatibilités et de dysfonctionnement sur certains environnements trop spécifiques.

- Application dite « légère »

Celle-ci se distingue surtout de par sa portabilité. Elle ne nécessite dans un environnement donné que de l'installation d'un navigateur Internet et d'outils additionnels dans certains cas (ex : pilote 3D, pilote de lecture de flux vidéo, JavaVM). Le principal avantage réside dans le déploiement de ces applications et de leur centralisation, qui impliquent des coûts d'installation et de maintenance assez réduits. Ce type d'application ne permet pas de faire des processus en temps réel.

L'inconvénient réside dans les faibles possibilités d'action sur le poste du client qui, pour des raisons de sécurité liée à Internet, sont volontairement réduites à des actions visuelles. A ces applications légères peuvent être ajoutés des composants graphiques supplémentaires, qui font de ces applications des clients dites « riches » dont l'aspect est similaire à une application lourde, on parle ainsi de RIA « Rich Internet Application ».

Tableau comparatif des solutions :

	Exécution d'un programme extérieur	Affichage d'éléments en 3D	Affichage de flux vidéo	Portabilité	Communication (protocole http)
Application lourde	OUI	OUI	OUI	NON	OUI
Application légère	NON	OUI	OUI	OUI	OUI

Malgré les inconvénients du client léger qui ne peut exécuter des programmes sur la machine locale et notamment du fait que nous développons une application web. Notre application sera une application légère.

XIII.5.2 – Architecture du système : Modèle, Vue et Contrôleur

Toute application de qualité se doit d'avoir une bonne architecture, et ceci pour diverses raisons que je détaillerais au fur et à mesure. Ceci est davantage préconisé lorsque l'on développe à partir d'une base vierge. Après plusieurs analyses, j'ai décidé d'opter pour une architecture de type MVC.

D'une part, c'est une architecture que je connais bien pour l'avoir développé à plusieurs reprises sous Java et d'autre part, le modèle ou architecture MVC est largement reconnu dans le domaine du développement. Par ailleurs, le MVC tout comme l'orientation objet du code, semble être devenu un standard dans le développement d'applications web, avec la réputation d'être une bonne pratique de conception.

XIII.5.3 – Définition

Le Modèle-Vue-Contrôleur (MVC) est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle. Ce paradigme divise l'IHM en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. L'architecture MVC ne résout pas tous les problèmes. Elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application. Ce modèle d'architecture impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.

XIII.5.4 – Architecture Modèle/Vue/Contrôleur

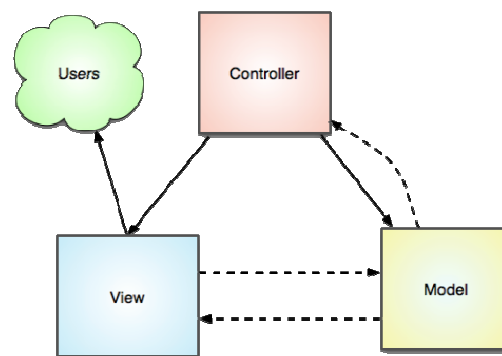


Fig.XIII.5.4.a - Schéma de principe du modèle-vue-contrôleur

Le modèle

Le modèle représente le comportement de l'application : traitement des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

Dans le cas de données importantes, le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir tous les cours d'une salle, tous les cours d'une personne ou tous les cours d'un groupe de TD.

La vue

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Plusieurs vues, partielles ou non, peuvent afficher des informations d'un même modèle. Par exemple, une application de conversion de bases a un entier comme unique donnée. Ce même entier peut être affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir la possibilité à l'utilisateur de changer de vue.

Elle peut-être conçue en HTML, ou tout autre « langage » de présentation.

Le contrôleur

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle, ce dernier avertit la vue que les données ont changé pour qu'elle se mette à jour. Certains événements de l'utilisateur ne concernent pas les données mais la vue.

Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

Par exemple, dans le cas d'une base de données gérant les emplois du temps des professeurs d'une école, une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne.

Quand un même objet contrôleur reçoit les événements de tous les composants, il lui faut déterminer quelle est l'origine de chaque événement. Ce tri des événements peut s'avérer fastidieux et peut conduire à un code pas très élégant (un énorme Switch). C'est pourquoi le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants.

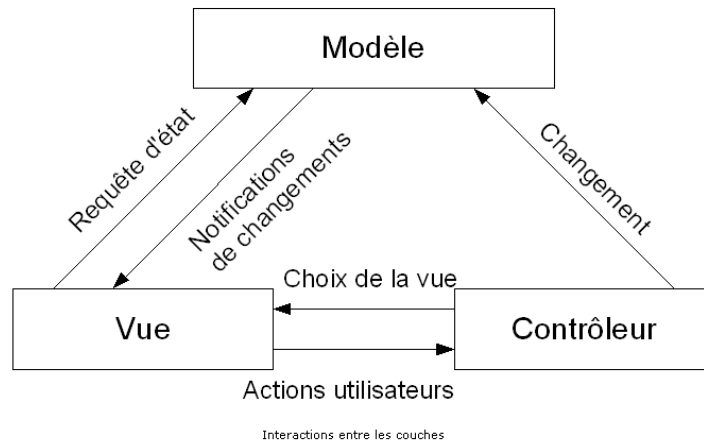


Fig.XIII.5.4.b – Echange entre modèle-vue-contrôleur

Le traitement d'une demande d'un client se déroule selon les étapes suivantes :

1. Le client fait une demande au contrôleur. Ce contrôleur voit passer toutes les demandes des clients. C'est la porte d'entrée de l'application. C'est le C du MVC.
2. Le contrôleur traite cette demande ? Pour ce faire, il peut avoir besoin de l'aide de la couche métier, ce qu'on appelle le modèle M dans la structure MVC.
3. Le contrôleur reçoit une réponse de la couche métier. La demande du client a été traitée. Celle-ci peut appeler plusieurs réponses possibles. Un exemple classique est
 - a. Une page d'erreurs si la demande n'a pu être traitée correctement
 - b. Une page de confirmation
4. Le contrôleur choisit la réponse (= vue) à envoyer au client. Celle-ci est le plus souvent une page contenant des éléments dynamiques. Le contrôleur fournit ceux-ci à la vue.
5. La vue est envoyée au client. C'est le V de MVC.

XIII.5.5 – Avantages et inconvénients du MVC

Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.

Le MVC montre ses limites dans le cadre des applications utilisant les technologies du web, bâties à partir de serveurs d'applications. Des couches supplémentaires sont alors introduites ainsi que les mécanismes d'inversion de contrôle et d'injection de dépendance.

XIII.5.6 – Différences avec l'architecture trois tiers

L'architecture trois tiers est un modèle en couches, c'est-à-dire, que chaque couche communique seulement avec ses couches adjacentes (supérieures et inférieures) et le flux de contrôle traverse le système de haut en bas; les couches supérieures contrôlent les couches inférieures, c'est-à-dire, que les couches supérieures sont toujours sources d'interaction (clients) alors que les couches inférieures ne font que répondre à des requêtes (serveurs).

Dans le modèle MVC, il est généralement admis que la vue puisse consulter directement le modèle (lecture) sans passer par le contrôleur. Par contre, elle doit nécessairement passer par le contrôleur pour effectuer une modification (écriture). Ici, le flux de contrôle est inversé par rapport au modèle en couche, le contrôleur peut alors envoyer des requêtes à toutes les vues de manière à ce qu'elles se mettent à jour.

Dans l'architecture trois tiers, si une vue modifie les données, toutes les vues concernées par la modification doivent être mises à jour, d'où l'utilité de l'utilisation du MVC au niveau de la couche de présentation. La couche de présentation permet donc d'établir des règles du type « mettre à jour les vues concernant X si Y ou Z sont modifiés ». Mais ces règles deviennent rapidement trop nombreuses et ingérables si les relations logiques sont trop élevées. Dans ce cas, un simple rafraîchissement des vues à intervalle régulier permet de surmonter aisément ce problème. Il s'agit d'ailleurs de la solution la plus répandue en architecture trois tiers, l'utilisation du MVC étant très moderne et encore marginale.

L'interface utilisateur est souvent un navigateur web mais cela pourrait être également une application autonome qui via le réseau enverrait des requêtes http au service web et mettrait en forme les résultats que celui-ci lui envoie. La **logique applicative** est constituée des scripts traitant les demandes de l'utilisateur, des classes métier et d'accès aux données. La **source de données** est souvent une base de données mais cela peut être aussi de simples fichiers plats, un annuaire LDAP, un service web distant, ...

XIII.6 – Définition des données

Lors de la rédaction des spécifications techniques, nous avons identifiés les entités ou données qui interagissent au sein de notre application. Ces interactions ont été appuyés par des schémas UML afin d'avoir une meilleure compréhension des actions possibles. Rappelons-les brièvement. Les descriptions étant brèves, se reporter au document de spécifications techniques pour consulter l'ensemble des fonctionnalités que je ne décrirais pas ci-dessous.

- **Elève** : Composant permettant de gérer les élèves de la plateforme. Notamment avec les actions « Ajout », « Suppression », « Modification », Récupération du mot de passe si perdu et de l'affichage/listage.
- **Utilisateur** : Composant permettant de gérer les administrateurs de la plateforme. Avec les actions « Ajout », « Suppression », « Modification », récupération du mot de passe si perdu et l'affichage/listage.
- **Système** : Composant permettant la gestion des systèmes de la plateforme. Avec les actions « Ajout », « Suppression », « Modification » et l'affichage/listage. Ce composant est lié avec la filière.

- **Filière** : Composant permettant la gestion des filières de la plateforme. Avec les actions « Ajout », « Suppression », « Modification » et l’affichage/listage.
- **Groupe** : Composant permettant la gestion des groupes de la plate forme. Ajout, suppression, modification, affichage/listage et gestion.
- **Groupe des élèves** : Composant lié au précédent permettant la gestion d’élèves appartenant à des groupes. Ajout d’un élève à un groupe, Suppression d’un élève d’un groupe, lister les élèves d’un groupe.
- **Séance** : Composant de gestion des séances. Ajout, suppression, modification et affichage sont disponibles.
- **TP** : Composant de gestion des TP. Ajout, suppression, modification et affichage sont disponibles.
- **Séances de TP** : Composant de gestion des Séances par rapport aux TP, ainsi on peut lier des TP à des séances.

Autres réalisations :

- **Gestion de l’authentification** de la plateforme. Il existe trois types d’utilisateurs : administrateur, élève et enseignant (pouvant être administrateur). Un système d’authentification existe pour ces trois types et permet à chacun d’entre eux d’aller sur l’espace qui leur est réservé. Des restrictions sont appliquées en fonction de l’utilisateur connecté. J’ai du passer un grand nombre d’heures sur ce système car il a été revu plusieurs fois (d’abord mono-utilisateur puis tri-utilisateur).
- **Gestion des sessions** de la plateforme. Les sessions sont liées étroitement au système d’authentification et permettent de garder certaines informations persistantes. On peut ainsi afficher le nom de l’utilisateur courant et ses informations personnelles dans l’espace prévu à cet effet. On peut ainsi connaître le statut de l’utilisateur et appliquer les restrictions de sécurité nécessaires.

A partir de ces informations, il m’est possible de pouvoir définir l’architecture de la base de données de notre application.

XIII.7 – Modélisation et conception de la base de données

Toute application, qu’elle soit, manipule des données. Les systèmes de gestion de bases de données (SGBD) jouent alors un rôle essentiel : fournir un service efficace de stockage et d’extraction des données.

Le choix du SGBD et de ses outils, le schéma de la base et la configuration de cet ensemble sont autant de sujets stratégiques qui requièrent la plus grande attention. Comprendre les notions essentielles liées aux bases de données est un point de départ nécessaire pour garantir performance, stabilité et durabilité à mon application.

Un SGBD (Systèmes de Gestion de Bases de Données), comme son nom l’indique, permet de manipuler des bases de données. Ul faut bien distinguer ces deux notions de base :

- Une *base de données* est composée d'un ensemble de données structurées. Une application peut utiliser une ou plusieurs bases de données. Physiquement, une base de données est constituée d'un ou plusieurs fichiers qui contiennent des données et des informations sur leur structure.
- Un *système de gestion de bases de données (SGBD)* est l'outil qui permet de manipuler les bases de données. C'est lui que l'on interroge pour ajouter, extraire, supprimer ou modifier des données des données dans une base.

A cause du temps qui me manquait et n'ayant pas encore à ce stade une visibilité complète de la plateforme (n'oublions pas que nous développons une plateforme expérimentale, tous les besoins de sont pas encore claires et définies). Je ne me suis pas attardé sur la conception du SGBD celui-ci ayant l'avantage d'être souple et adaptable durant la phase de développement.

Il est toutefois important de commencer par la conception de la base de données avant les tâches de développement afin d'avoir une meilleure visibilité sur le développement des IHM et du code source.

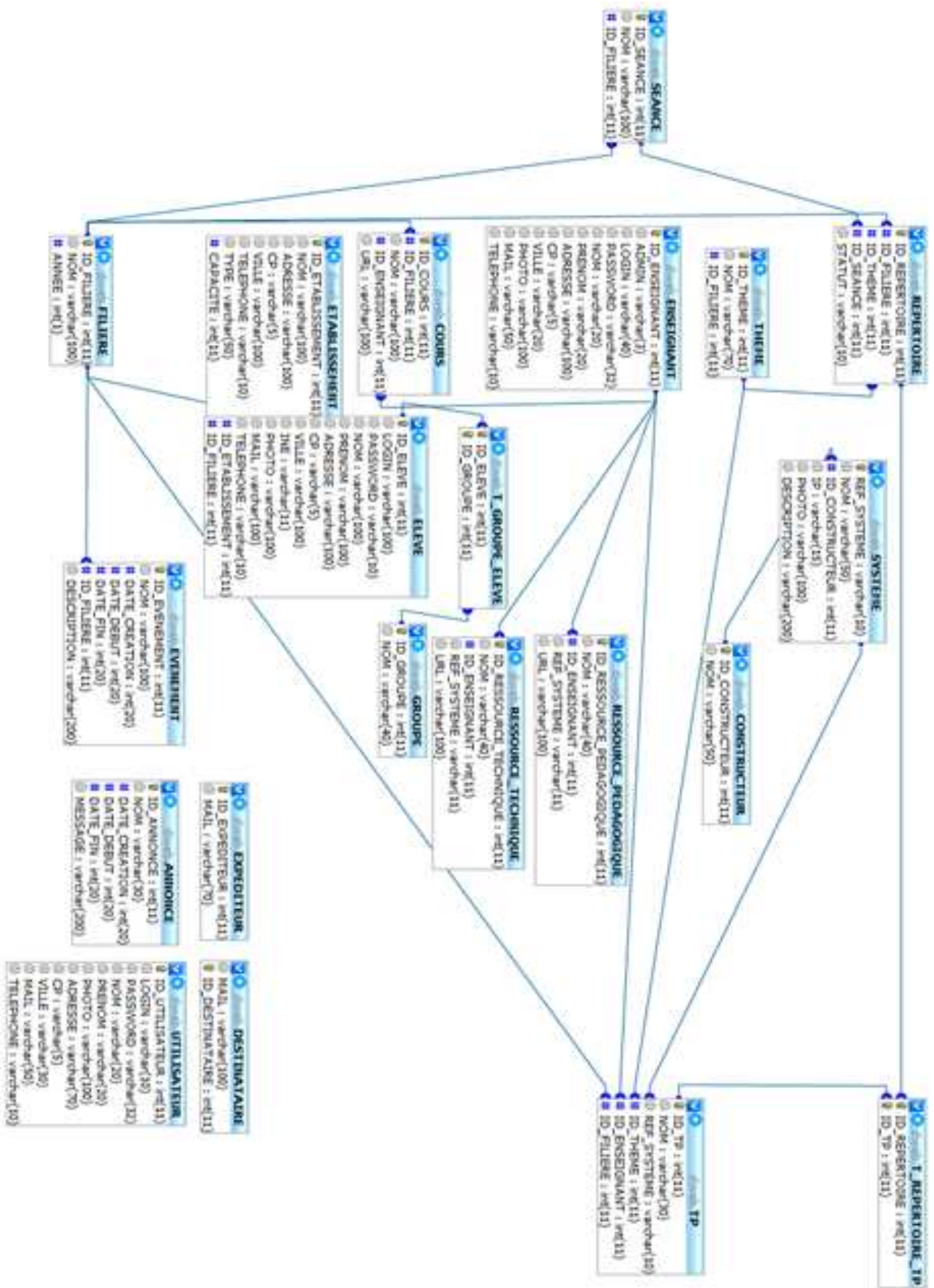


Fig. XIII.7 - Schéma de bases de données

XIII.8 – Découpage des tâches de développement

Afin de structurer le développement, il est possible de décomposer celui-ci en lots ou en chantiers, afin d'obtenir des sous-ensembles dont la complexité est plus facilement maîtrisable. Ceci est d'autant plus important à effectuer dès lors que nous avons plusieurs développeurs. Après découpage, il me sera plus aisé de procéder à sa planification.

Pour effectuer le découpage, il est possible de procéder des deux manières suivantes :

Découpage en tâches

Une autre manière (complémentaire) d'aborder le découpage d'un projet en sous-ensembles élémentaires consiste à découper le projet en sous-ensembles d'activité à fonction simple : les **tâches**.

Chaque tâche est caractérisée par des matières premières qui lui sont nécessaires : ce sont les objets entrants ou préalables (un document, une spécification, une machine mise à disposition, une norme, un opérateur formé et opérationnel, un jeu d'essai, etc.) et fournit un ou plusieurs produits résultats, ce sont les objets sortants ou **livrables** (un logiciel, une plaquette publicitaire, un support de cours de formation, une fiche technique, etc.). Les objets entrants peuvent être déjà à disposition ou bien sont les objets sortants d'une autre tâche destinée à les produire. Les objets sortants peuvent être réemployés par une autre tâche.

Cette méthode (menée par exemple par interviews des différents acteurs impliqués) permet de mettre en évidence les tâches préparatoires et de matérialiser les dépendances entre tâches (par exemple pour formaliser un réseau PERT).

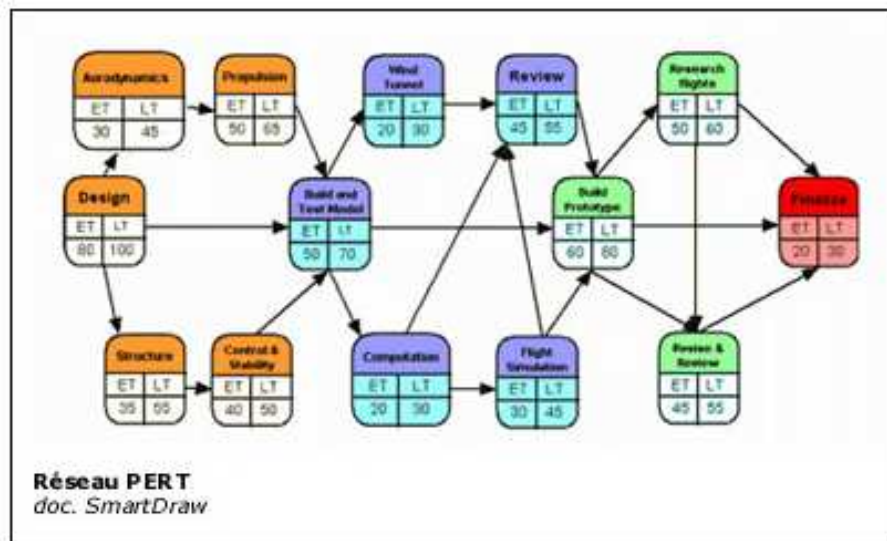


Fig. XIII.8.a – Exemple de diagramme de PERT suite à un découpage en tâches

Découpage en activités WBS

Une deuxième approche, connue sous le nom de méthode WBS (*Work Breakdown Structure*), procède à un découpage statique (analytique) en **activités** (on retrouve ici une notion voisine de celle de **tâches**) avec des entrées et des résultats identifiés ainsi qu'une responsabilité confiée à une personne nommée, et ceci jusqu'au niveau de décomposition optimal nécessaire pour :

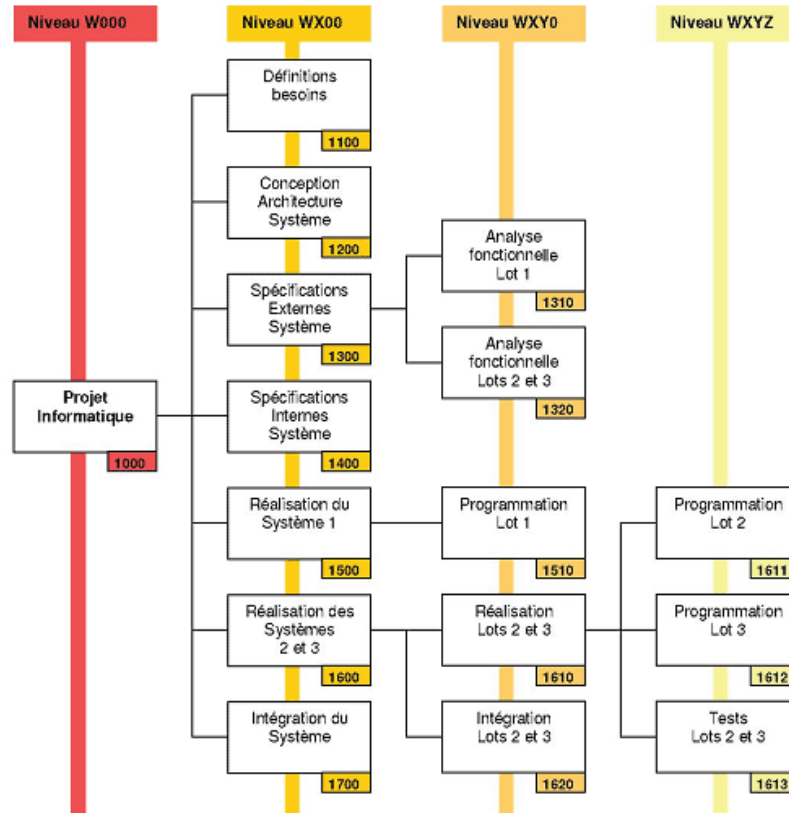
- Maîtriser la durée d'une activité,
- Connaître les ressources requises,
- Connaître le coût d'une activité.

Le Work Breakdown Structure (WBS ou OT Organigramme des Tâches) est un arbre représentant la liste structurée de tous les travaux du projet. Les travaux sont à ce stade uniquement identifiés.

Ce graphe, utilisant la notation de précedence, représente les relations entre les tâches du type « Fait partie de ». C'est une vision maîtrise d'œuvre des tâches afin de déterminer les niveaux de visibilité du projet.

Pour construire cette vision, il faut ...

- Effectuer l'inventaire exhaustif des tâches à réaliser
- Identifier les lots de travaux (LT)
- Attribuer à chaque LT un responsable unique
- Définir de façon claire les niveaux de responsabilités et leurs limites



Exemple d'Organigramme des Tâches (WBS) pour un projet informatique

Fig.XIII.8.b – Exemple d'organigramme WBS suite à un découpage en tâches

C'est cette dernière méthode que j'ai retenue afin d'effectuer mon découpage, l'ayant acquis au cours d'un module d'ingénierie des systèmes d'information, je n'avais encore jamais eu l'opportunité de le mettre en pratique. Cela me permettra de comparer ses performances par rapport au diagramme de PERT que j'ai déjà pratiqué.

Ce WBS se concentre donc sur des lots de travaux et pas sur les parties à développer comme les interfaces web de création, de suppression, de modification et de mise à jour de données. De plus, en parallèle du développement il me faut superviser la cohérence des données échangées entre le serveur principal et les serveurs embarqués des systèmes communicants.

Ce WBS a été réalisés lors des spécifications techniques avant la mise en place du réseau Electrotechnique et des serveurs. N'ayant eu le temps par la suite le mettre à jour certaines informations ont été revues par la suite. Toutefois les exigences principales sont toujours présentes.

XIII.9 – Structure initiale

XIII.9.1 – Phase de conception 1 – Structure initiale (WBS 1000)

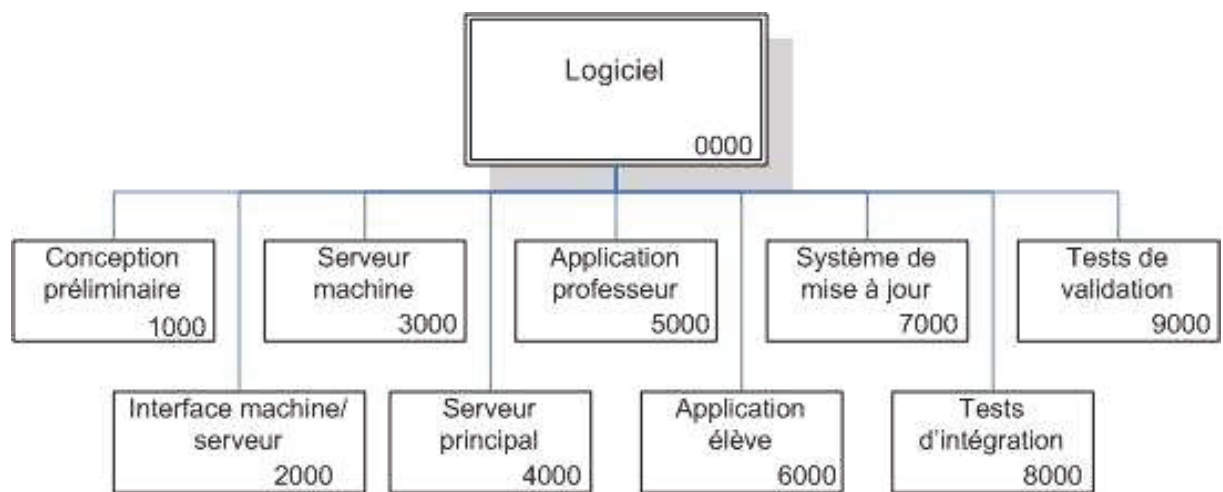


Fig.XIII.9.1 – Structure initiale WBS du projet

Commentaires :

La phase de conception 1 concerne l'ensemble des tâches de l'application de façon globale. L'entrée de l'arbre se situe donc au lot 0000 qui se trouve être l'application finale à aboutir. Celui-ci est découpé en 9 lots d'activités qui désignent les catégories auxquels les tâches appartiennent :

- **Lot 1000** : Conception préliminaire, il s'agit de la mise en place de l'architecture du système et de la conception de la base de données. Il s'agit aussi d'introduire les développeurs à l'environnement de développement et aux outils à utiliser. En ce sens, il s'agit de ma responsabilité que de les former à cela.
- **Lot 2000** : Interface machine/serveur, il s'agit de développer les interfaces qui manipuleront les données issues des serveurs web embarqués. Ce lot est à la fois difficile et essentielle sur notre développement, en effet l'échange entre notre application et les serveurs web embarqués est le noyau de l'application. Sans des données exploitables, notre application n'aura aucun sens. La plupart des cas ces données sont des fichiers structurés à la manière de XML, ou CSV voire même des variables à lire en temps réels. Pour les variables, comme nous avons une application de type légère il n'est pas possible d'avoir des lectures en temps réels, il nous est donc essentielle d'avoir des traitements d'acquisition de mesures.

- **Lot 3000** : Serveur machine, il s'agit de toutes les interfaces à développer sur notre application ayant pour but d'exploiter les données issues des serveurs web embarqués. Ces données sont fournies sous forme de fichiers et variables de lecture, il nous est donc nécessaires d'avoir des IHM de rendu graphiques et autres (graphiques, vidéos, tableaux et autres). Nous avons 3 solutions à notre disposition :
 - Nous redéveloppons les interfaces embarquées par les systèmes communicants via des applets Java,
 - Nous développons les interfaces et les stockons sur le serveur principal (interfaces personnalisées pour chaque système),
 - Nous réutilisons les interfaces déjà présentes dans les systèmes, ceux qui ont été refaites par les étudiants de BTS Electrotechnique. *Nous préconiserons pour l'instant celui-ci car le temps manquera pour aller plus loin.*
- **Lot 4000** : Serveur principal, il s'agit de l'environnement de base pour notre application, c'est-à-dire l'aspect graphique, l'authentification et la connexion à celle-ci. L'affichage des ressources et transfert des données.
- **Lot 5000** : Application professeur, il s'agit de développer les IHM de gestion d'un professeur ainsi que les fonctionnalités de connexion. Pour l'instant, je ne souhaite pas commencer par focaliser les fonctionnalités d'interactions avec les élèves (suivi des Tp, corrections, forum etc.). Le temps nous manquera pour tout faire mais je souhaite prioritairement essayer de résoudre les fonctionnalités de communication ou d'échange avec un élève qui me paraît plus important techniquement comme la visio-chat, assistance à distance etc.
- **Lot 6000** : Application élève, il s'agit de développer les IHM de gestion d'un élève ainsi que les fonctionnalités de connexion. Là aussi, le temps nous manquera pour tout faire hormis les IHM de gestion nous laisseront de côté les fonctionnalités d'interactions avec l'application (suivi de TP, assistance par l'enseignant et autres ...). Je prendrais toutefois le temps de voir la faisabilité de l'affichage 2D/3D des données qui est un élément assez conviviale pour l'exploitation de données.
- **Lot 7000** : Système de mise à jour, il s'agit de développer toutes les fonctions de mises à jour comme le transfert, l'enregistrement de données et fichiers, ce qui permettra de développer dans les systèmes tous les IHM relatives à la gestion des TP.
- **Lot 8000** : Tests d'intégration, il s'agit de vérifier que le code source de l'application et toutes les fonctions de celle-ci.
- **Lot 9000** : Tests de validation, il s'agit de vérifier toutes les fonctionnalités avant la mise en service de l'application.

Tous les lots comprennent des tâches de conception détaillée qui fait référence aux spécifications techniques ainsi que des tests unitaires pour assurer une qualité et une certaine rigueur au niveau du développement. Effectivement, les lots de tâches étant successives, il suffit qu'une soit erronée pour que cela engendre d'autres erreurs par la suite (effets de cascade).

XIII.9.2 – Phase de conception 2 (WBS 2000)

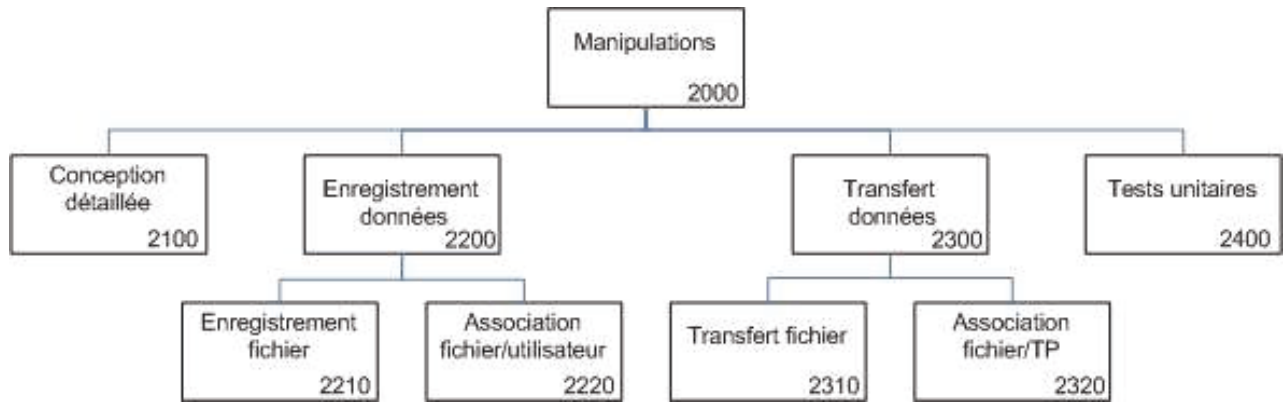


Fig.XIII.9.2 – Structure WBS Phase de conception 2

Commentaires :

Le lot 2000 comprend 7 tâches correspondant aux fonctionnalités d'exploitation et de sauvegarde de données :

- **Lot 2200** : correspond aux fonctionnalités d'acquisition et d'enregistrement de mesures ou autres résultats retournés par le système.
- **Lot 2300** : correspond aux fonctionnalités de récupération et de transfert de données sous forme de fichiers dans les serveurs embarqués pour ensuite les enregistrer en base de données (**Lot 2310**) ou les stocker sur le serveur (**Lot 2320**), on lui associe un utilisateur/élève afin de l'identifier.

La complexité dans ce développement est de savoir si l'on développe des interfaces génériques et adaptables à chaque système ou si on développe une interface par système. La contrainte étant les types de serveurs web embarqués et leurs capacités.

XIII.9.3 – Phase de conception 3 (WBS 3000)

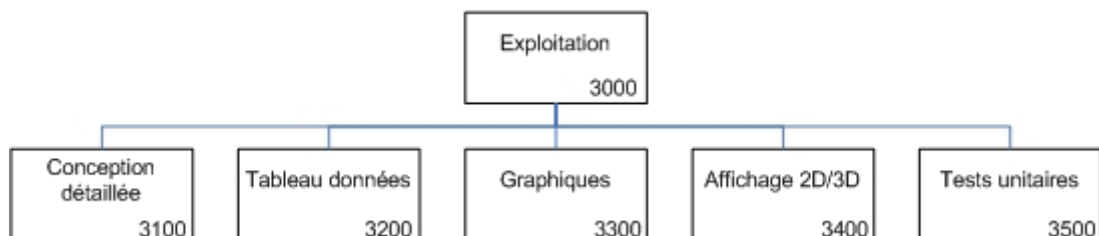


Fig.XIII.9.3 – Structure WBS Phase de conception 3

Commentaires :

Le lot 3000 comprend 5 tâches correspondant aux fonctionnalités d'exploitation de données :

- **Lot 3200** : permet de retourner les résultats issus d'une manipulation d'un système sous forme de tableaux.
- **Lot 3300** : permet de retourner les résultats issus d'une manipulation d'un système sous formes de graphiques.
- **Lot 3400** : permet de retourner les résultats issus d'une manipulation sous forme d'affichage 2D/3D, cette fonctionnalité ne sera pas développée dans un premier temps.

XIII.9.4 – Phase de conception 4 (WBS 4000)

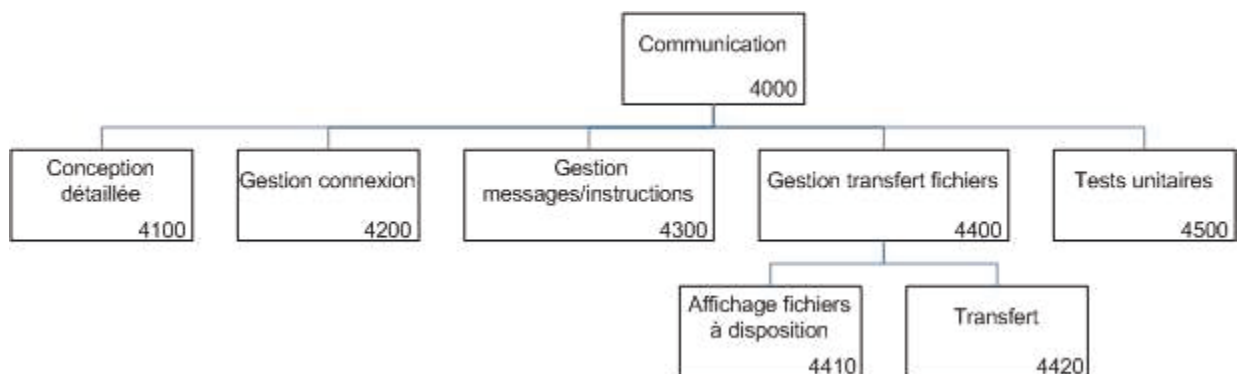


Fig.XIII.9.4 – Structure WBS Phase de conception 4

Commentaires :

Le lot 4000 comprend 7 tâches correspondant aux fonctionnalités de communication de la plateforme :

- **Lot 4200** : correspondant au module de connexion à la plateforme.
- **Lot 4300** : correspondant au module d'affichage de pages, messages et instructions de TP.
- **Lot 4400** : module de gestion documentaire pour transfert et consultation

XIII.9.5 – Phase de conception 5

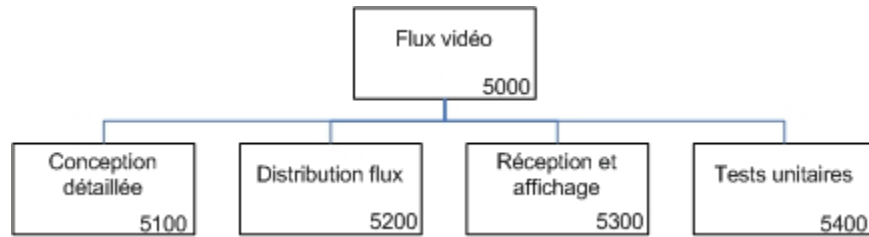


Fig.XIII.9.5 – Structure WBS Phase de conception 5

Commentaires :

Le lot 5000 comprend 4 tâches correspondant aux fonctionnalités d'acquisition vidéo :

- **Lot 5200** : Module d'analyse de flux vidéo pour encodage et décodage, ce flux sera issu de la caméra pilotable à distance.
- **Lot 5300** : Module de rendu vidéo issu de la caméra pilotable à distance.

XIII.9.6 – Phase de conception 6

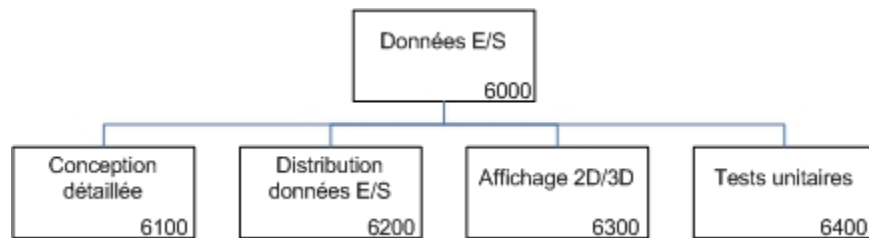


Fig.XIII.9.6 – Structure WBS Phase de conception 6

Commentaires :

Le lot 5000 comprend 4 tâches correspondant aux fonctionnalités d'affichage de données :

- **Lot 6200** : Module de distribution de données d'acquisition à la demande.
- **Lot 6300** : Affichage des données sous forme 2D/3D

XIII.9.7 – Phase de conception 7

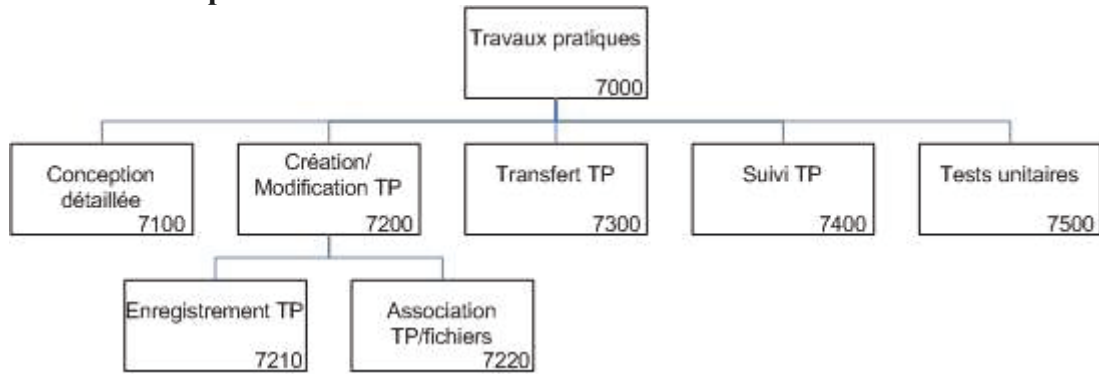


Fig.XIII.9.7 – Structure WBS Phase de conception 7

Commentaires :

Le lot 7000 comprend 7 tâches correspondant aux fonctionnalités de gestion de travaux pratiques :

- **Lot 7200** : Fonctionnalités de création, modification, suppression et lecture de TP.
- **Lot 7300** : Fonctionnalités de transfert de TP créé à partir d'un fichier Word.
- **Lot 7400** : Fonctionnalités de suivi de TP par les élèves.

XIII.10 – Répartition des tâches

Après avoir effectué le découpage des développements et avoir définis les lots. Il me paraissait évident qu'à moi seul, il m'aurait été difficile d'effectuer l'ensemble du développement de l'application. Mais j'ai pris soin d'anticiper cette lourde charge de travail en demandant de à DMS de pouvoir mettre à ma disposition des stagiaires en DUT Informatique et de m'assister dans cette phase de développement.

Après m'être entretenu avec chacun d'eux pour connaître leurs connaissances et compétences en développement, je me suis rendu compte que le niveau technique d'un BAC+2 est largement en dessous de ce que j'avais espérés, mais je ne suis pas était surpris car j'ai déjà eu l'occasion par le passé de diriger des stagiaires de 4^{ème} ou 5^{ème} année d'école d'ingénieur. De plus, par mon expérience professionnelle assez enrichi, j'ai une vision plus large et pragmatique du développement que l'on n'a pas toujours lorsqu'on est étudiant, ça sera donc bénéfique. La durée du stage est d'environ 11 semaines, 10 seront réservés pour le développement de la plateforme, je leurs ai accordé la dernière semaine pour la rédaction de leurs compte-rendu ainsi que la préparation à leurs présentation à l'oral.

Voici donc la répartition des tâches que je leurs ai affectés ainsi que les miennes :

	Semaine 1 : 05/04 - 09/04	Semaine 2 : 12/04 - 16/04	Semaine 3 : 19/04 - 23/04	Semaine 4 : 26/04 - 30/04	Semaine 5 : 03/05 - 07/05
Jean-Philippe Sechao	Lot 2200	Lot 2300	Lot 4300	Lot 3200, 3300	Lot 3200, 3300
Matthieu Flatres	Lot 4200	Lot 6000	Lot 6000	Lot 6000	Lot 6200
Romain Bouldoires	Lot 5000	Lot 5000	Lot 4400	Lot 5000	Lot 5200
	Semaine 6 : 10/05 - 14/05	Semaine 7 : 17/05 - 21/05	Semaine 8 : 24/05 - 28/05	Semaine 9 : 31/05 - 04/06	Semaine 10 : 07/06 - 11/06
Jean-Philippe Sechao	Lot 3400	Lot 7300	Lot 7400	Lot 8000	Lot 9000
Matthieu Flatres	Lot 6300	Lot 7200	Lot 7200	Lot 8000	Lot 9000
Romain Bouldoires	Lot 5200	Lot 7000	Lot 7000	Lot 8000	Lot 9000

Par précaution, je me suis réservé les tâches les plus complexes et longues à réaliser, me contentant ainsi de leur affecter des tâches de développement primaires afin d'évaluer leurs niveau de compréhension et techniques. Je serais tout de même attentif concernant la qualité du code source. La répartition des tâches est associée à un planning qui servira plus ou moins à mieux définir les temps de charges des lots de tâches, certaines tâches sont dépendantes de d'autres tâches.

Par expérience, je sais pertinemment qu'il me sera difficile de faire respecter ce planning car il y'a toujours des facteurs de temps que l'on ne maîtrise pas aussi, ai-je décidé de prendre large au niveau du temps de charge des lots de tâches.

Au fur et à mesure du développement, j'ai réalisés les diagrammes de classes correspondant aux tâches de développement de chacun. Ce diagramme de classes vient appuyer une meilleure compréhension de l'architecture de l'application que j'avais acquise lors de la réalisation d'une étude technique sur l'outil ATL utilisant une méthode de conduite de développement informatique par les modèles.

XIV – DEVELOPPEMENT

Après le découpage des activités et la répartition de celle-ci, il est grand temps de commencer la construction de l'application. Avant de commencer le développement, il reste au préalable une dernière à voir. En effet, il a faut se familiariser avec l'utilisation du PHP orienté objet car j'avais fait le souhait d'utiliser le PHP en langage objet ainsi que l'utilisation de la méthodologie de conception MVC.

En informatique, un framework peut être vu comme un outil de travail adaptable. C'est un ensemble de bibliothèques, d'outils et de conventions permettant le développement rapide d'applications. Un framework est composé de briques logicielles organisées pour être utilisées en interaction les unes avec les autres. L'utilisation d'un framework permet le partage d'un savoir-faire technologique et impose suffisamment de rigueur pour pouvoir produire une application aboutie et facile à maintenir

Il existe de nombreux Framework pour les développements sous PHP mais pour une raison personnelle, je me suis intéressé à un Framework en particulier, celui de « Zend Framework ». D'une part parce qu'il est un des seuls Framework que je n'ai pas eu l'occasion de mettre en œuvre ou d'autre part il est axé sur la méthodologie MVC, par ailleurs ZF est un framework Open Source qui se veut modulaire et modulable. Je profite donc par l'opportunité qui se présente à moi et par la liberté de choix des outils pour le mettre en œuvre. J'ai donc pris soin de me former à cet outil afin de pouvoir former les stagiaires en informatique. Je vais donc commencer par faire une présentation rapide.

XIV.1 – Introduction à Zend Framework

Le **Zend Framework** est un Framework pour PHP 5 créé en mars 2006 par Zend Technologies . Il est distribué sous la Licence BSD Modifiée. Le Zend Framework, aussi nommé ZF, a été développé dans le but de simplifier le développement Web tout en recommandant les bonnes pratiques et la conception orientée objets en offrant des outils aux développeurs. ZF permet aussi d'utiliser nativement le principe de MVC (Modèle-Vue-Contrôleur) mais ne l'oblige pas.

Le Framework Zend est une collection de composants modulaires comme les autres bibliothèques comme PEAR, Zend Framework y ajoute les bonnes pratiques issues de l'expérience et de la théorie de l'informatique objet (design pattern) utilisés de longue date avec Java.

Pourquoi l'utiliser ?

La version 5 de PHP facilite l'industrialisation des développements, ainsi des solutions ont vu le jour comme CakePHP, Symfony, Zend Framework, etc.

Un Framework ou cadriceil et parfois « cadre de travail » est un ensemble de bibliothèques et d'outils pour aider le développeur mais en plus il impose une rigueur sur la façon de programmer justifiée par les bonnes pratiques et les conventions communes. Un Framework est un gage de qualité car un code cohérent sera plus aisé à maintenir ou à faire évoluer.

Pour le développeur les bibliothèques fondamentales sont fournies et comme les connecteurs à la base de données, la persistance et la validation de données, la génération de formulaires, etc.

Initiation au Zend Framework : Le patron de conception Modèle vue contrôleur (MVC)

Zend Framework comme tous les cadriceils majeurs des langages de scripts serveurs utilise le patron de conception Modèle-vue-contrôleur qui sépare le modèle de données, l'interface utilisateur et les traitements. Ces trois parties fondamentales se nomment : le modèle, la vue et le contrôleur.

Le modèle ne s'occupe que des traitements des données, les liens avec la base de données, lecture, insertion, mise-à-jour des tuples dans une base, vérifier que les données sont bien formatées (validation). La présentation des résultats ne se fait pas dans le modèle.

La vue correspond à la présentation des résultats, par exemple un tableau HTML, PDF pour un utilisateur ou sous forme XML pour fournir un programme distant. Les événements et les actions de l'utilisateur (clic d'un bouton, liste de sélection, etc.) sont gérés dans la vue. Cette séparation permet au graphiste de travailler sans avoir à se soucier des rouages de l'application.

Le contrôleur prend en charge le déroulement du programme. La liste des actions sera dans le contrôleur. Il synchronise les événements provenant de l'utilisateur vers la base de données.

Ce patron de conception existe traditionnellement dans les langages objet comme Java comme par exemple la bibliothèque graphique SWING et est employé généralement dans les langages web comme la plateforme Adobe/Flex ou la prochaine norme W3C de formulaires Xforms. Cette orientation a été perçue très tôt par les développeurs web notamment avec la popularité du Framework Ruby on Rails. L'éditeur de PHP, Zend a pris la même orientation tout en conservant les développeurs ne souhaitant pas migrer vers MVC et permet ainsi d'utiliser ZF comme une bibliothèque classique.

Avantages et inconvénients de l'architecture MVC Zend Framework.

L'avantage premier est la modularité, les objets sont réutilisables dans une autre application. Le cloisonnement ajoute en facilité de maintenance. De plus, la modification des traitements se fait indépendamment de l'affichage (la vue). Il faut préciser que les développeurs qui sont impliqués dans l'amélioration du Framework proposent des composants issus de la théorie informatique (comme les design patterns) ou le fruit de l'expérience par de bonnes pratiques de programmation.

L'inconvénient est une phase de conception plus longue et ces ajouts de couches multiplient le nombre de fichiers sur le serveur. Cela nécessite des outils spécifiques de débogage et de

mécanismes d'amélioration des performances (cache). Le temps d'apprentissage est à prendre en compte pour les développeurs.

Zend Framework : Fonctionnement

L'apprentissage du fonctionnement du *Framework* peut être un peu ardu, car cela nécessite une connaissance de quelques concepts de programmation orientée objet. Mais la consultation de la documentation, du *tracker* de *bugs*, des blogs des développeurs permettent d'en comprendre la globalité.

Le processus d'une requête est :

- Le *bootstrap* (html/index.php) est le point d'entrée dans l'application, il s'agit de l'implémentation du design pattern contrôleur frontal et d'un *singleton*. Toutes les requêtes passent par cet objet, à sa charge d'acheminer (*dispatcher*) vers les actions (suivant une route par défaut ou définie par le concepteur). Les réponses sont collectées par cet objet.
- La requête peut d'abord être prétraitée dans un *plugin*, cela permet au concepteur d'effectuer des traitements en tout début de requête.
- La requête est routée, c'est à dire que l'on traduit l'URI d'entrée en lien vers le bon contrôleur. Le contrôleur est instancié, son constructeur *init()* est lancé.
- La fonction *preDispatch()* est lancé si elle existe juste avant de lancer l'action : par exemple *readAction(...)*.
- Le contrôleur se termine et la fonction *postDispatch()* est lancée.
- La réponse est renvoyée au visiteur.

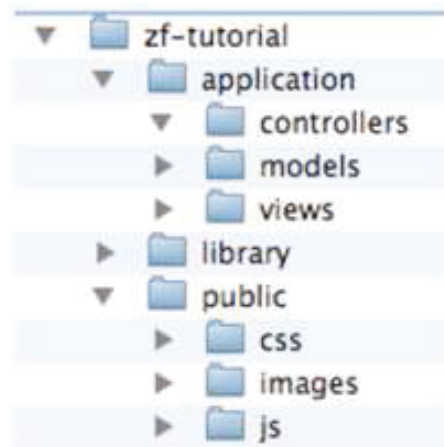
Configuration Apache

Apache doit avoir le mod_rewrite présent et activé (a2enmode rewrite) ainsi que d'autoriser les directives par .htaccess : « AllowOverride » sera à « All » dans le fichier de configuration du site.

```
<Directory /var/www/monblog/html>
    DirectoryIndex index.php
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```

XIV.2 – Représentation UML des objets métiers – Diagrammes de classes

Nous sommes maintenant prêts à mettre en place les contrôleurs de l'application. Avec Zend Framework, le contrôleur est une classe qui doit être appelée "{Nom du contrôleur}Controller". Rappelons aussi que la structure des répertoires de fichiers doit être la suivante :



Ces classes doivent être chacun dans un script appelé {Nom du contrôleur}Controller.php dans le répertoire du contrôleur spécifié. De nouveau, {Nom du contrôleur} doit commencer par une lettre majuscule et ne contenir que des minuscules par la suite. Chaque action est une fonction publique dans le contrôleur et doit être appelée {nom de l'action}Action. Dans ce cas, {nom de l'action} doit commencer par une lettre minuscule.

Les Modèles sont les parties du MVC qui s'occupe de l'objectif central de l'application (la *logique applicative*) et ainsi, dans notre cas, dialogue avec la base de données. Nous utiliserons la classe **Zend_Db_Table** qui recherche, ajoute, modifie ou supprime des enregistrements de la base de données.

Dans les diagrammes de classes qui vont suivre, je me suis penché essentiellement sur les classes Contrôleurs et Modèles. Les fichiers de Vues n'étant pas des classes et de plus n'apporte nullement d'intérêts. Je réaliserais les vues après pendant que les stagiaires développent les classes en fonction des diagrammes.

Les diagrammes de classes sont regroupés autour de 5 thèmes principaux de l'application :

- **Bureau** : Ensemble de fonctionnalités relatives à la gestion des travaux que doivent réaliser l'élève, il s'agit ni plus ni moins qu'un tableau de bord.
- **Cours & Ressources** : Ensemble des fonctionnalités relatives à la gestion des ressources documentaires, fichiers et autres formats.
- **TP & Systèmes** : Ensemble des fonctionnalités relatives aux systèmes communicants, de leurs thématiques à leurs informations générales.
- **Utilisateurs & Authentification** : Ensemble de fonctionnalités relatives à la gestion des utilisateurs et leurs authentification sur l'application.
- **Zone TP** : Ensemble de fonctionnalités relatives à l'organisation, déroulement et suivi des TP.

Il faut garder en tête que tout ce développement est susceptible d'être remise en cause si il s'avérait que les fonctionnalités ne correspondraient à ce que les enseignants souhaiteraient. La base de données ainsi que les diagrammes et le code source n'ont pour l'instant que l'objectif de fournir une plateforme prototype voire expérimentale.

Il se peut que l'on trouve des incohérences sur les diagrammes de classes, en effet, ceux ayant été faits avant la phase de développement, ils n'ont été remis à jour suite aux modifications du code source pendant le développement. Par l'utilisation de la méthodologie de développement XP, mes stagiaires et moi-même avons modifié et adapté le code source suite aux différentes remarques et avis survenus en cours de développement.

Introduction au diagramme de classe

Le **diagramme de classes** est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Sur un diagramme, on distingue aisément :

- *Le nom de la classe*, indiqué en haut du rectangle représentant une classe.
- *Les attributs*, indiqués sur la partie du milieu d'un rectangle.
- *Les méthodes*, indiquées sur la partie basse d'un rectangle.

La plupart des diagrammes de classes qui suivront n'ont pas d'indications sur les attributs. En effet, au moment du développement, il ne m'a pas été donné d'indications sur les données à conserver.

Par ex. Pour un enseignant mise à part, son nom prénom et sa matière, on ne sait pas ce que nous devons lui associer. Il en est de même pour tous les interactions avec les données.

En raison de cela et afin de simplifier au maximum la compréhension de l'architecture, j'ai fait l'impasse sur les attributs. Il sera tout à fait possible de pouvoir les ajouter ultérieurement.

Pour réaliser ces diagrammes, j'ai utilisé un logiciel assez performant, qui est PowerAMC.

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom AMC*Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque **PowerDesigner**.

Les diagrammes de classes de l'application sont partiellement conformes aux structures WBS, en raison, que les besoins ont entre temps été changés depuis la rédaction des spécifications

techniques. Il s'agira de plus de définir des priorités aux classes à développer dans un premier temps.

Description des modèles

Le modèle « Enseignant », représentant les données et actions sur un enseignant.

Les méthodes disponibles pour ce modèle sont :

- obtenirEnseignant(), permet de consulter la fiche d'un enseignant
- obtenirEnseignantParMail(), permet de récupérer le mot de passe par mail
- ajouterEnseignant()
- modifierEnseignant()
- supprimerEnseignant()
- modifierPassword()

Le modèle « Elève », représentant les données et actions sur un élève.

Les méthodes disponibles pour ce modèle sont :

- obtenirEleve(), permet de consulter la fiche d'un élève
- obtenirEleveParMail(), permet de récupérer le mot de passe par mail
- ajouterEleve()
- modifierEleve()
- supprimerEleve()
- modifierPassword()

Le modèle « Utilisateur », représentant les données et actions sur un utilisateur. Un utilisateur est une personne qui ne soit pas un élève, ni un enseignant mais qui souhaite tout de même pouvoir accéder aux fonctionnalités du site par ex. pour une formation à distance ou autres formes d'apprentissage. Les méthodes disponibles pour ce modèle sont :

- obtenirUtilisateur(), permet de consulter la fiche d'un utilisateur
- obtenirUtilisateurParMail(), permet de récupérer le mot de passe par mail
- ajouterUtilisateur()
- modifierUtilisateur()
- supprimerUtilisateur()
- modifierPassword()

Le modèle « Cours », représentant les données et actions sur un cours. Les cours seront principalement des supports sous format numérique tels que word ou pdf. Les méthodes disponibles pour ce modèle sont :

- obtenirCours(), permet d'afficher la liste des cours disponibles
- ajouterCours() permet d'ajouter un cours c'est-à-dire transférer un support électronique sur le serveur pour le rendre téléchargeable.

Le modèle « Calendrier », représentant les données et actions sur un calendrier. Il s'agira d'un modèle qui ne servira que pour la vue. Son rôle étant d'afficher un calendrier à la demande.

Le modèle « Annonce », représentant les données et actions sur une annonce. Même si le besoin a été exprimé, nous manquons toujours les informations nécessaires à son implémentation. Nous le rajoutons quand même dans le diagramme.

Le modèle « Evènement », représentant les données et actions sur un évènement. Même si le besoin a été exprimé, nous manquons toujours les informations nécessaires à son implémentation. Nous le rajoutons quand même dans le diagramme.

Le modèle « Filière », représentant les données et actions sur une filière. Il s'agira d'un modèle qui ne servira qu'à déterminer les filières auxquels appartiennent les enseignants et les supports de

cours. Notre objectif visant pour l'instant, la filière Electrotechnique, ce modèle n'est pas important pour l'instant, mais sera utile par la suite. Les méthodes disponibles pour ce modèle sont :

- obtenirFiliere(), permet d'afficher la liste des filières disponibles.
- obtenirNomFiliere()
- ajouterFiliere()
- modifierFiliere()
- supprimerFiliere()

Le modèle « RessourceTechnique », représentant les données et actions sur une ressource technique. Les ressources techniques seront principalement des supports sous format numérique tels que pdf ou word. Il est particulièrement utile pour regrouper tous les documents techniques relatives à un système afin d'éviter de perdre du temps à les chercher. Son développement n'est toutefois pas prioritaire. Les méthodes disponibles pour ce modèle sont :

- obtenirRessourceTechnique(), permet d'obtenir la liste des ressources techniques.
- ajouterRessourceTechnique(), permet d'ajouter une ressource technique sur le serveur.

Le modèle « RessourcePedagogique », représentant les données et actions sur une ressource pédagogique comme les tutoriels ou manuels. Les ressources pédagogiques seront principalement des supports sous format numérique tels que pdf ou word. Il est particulièrement utile pour regrouper tous les documents pédagogiques tous les documents relatives à la compréhension d'une théorie ou d'un principe. Son développement n'est toutefois pas prioritaire. Les méthodes disponibles pour ce modèle sont :

- obtenirRessourcePedagogique(), permet d'obtenir la liste des ressources pédagogiques
- ajouterRessourcePedagogique(), permet d'ajouter une ressource pédagogique sur le serveur.

Le modèle « Systeme », représentant les données et actions sur un système. Un système est représenté par une fiche qui contient toutes les données relatives à celles-ci, caractéristiques techniques et principe de fonctionnement. Il est utile pour appréhender et s'initier au système. Son développement n'est toutefois pas prioritaire.

Les méthodes disponibles pour ce modèle sont :

- obtenirSysteme(), permet d'obtenir la liste des systèmes.
- ajouterSysteme(),
- modifierSysteme(),
- supprimerSysteme(),

Il existe d'autres modèles mais il n'est pas nécessaire de les détailler. Ils apportent une meilleure compréhension des fonctionnalités de l'application mais ils ne sont pas indispensables. Par ailleurs, on remarque que les modèles disposent de méthodes simples principalement d'accès aux données et c'est un peu leurs rôles principaux que de servir d'intermédiaire avec la base de données, on les appelle ainsi des interfaces de mapping (ORM en anglais pour Object Relationnel Mapping).

Nous nous contenterons de ces méthodes avant d'aller plus loin. En méthodologie XP, il vaut mieux partir de bases simples et d'alimenter les classes par des méthodes dont on aura certainement l'utilité. Rappelons que ces diagrammes ne donnent pour l'instant qu'une vue générale de l'architecture pour une meilleure compréhension, elle n'est pour l'instant pas figée. Beaucoup de modifications peuvent intervenir en cours de route.

XIV.2.1 – Diagramme de classe Bureau

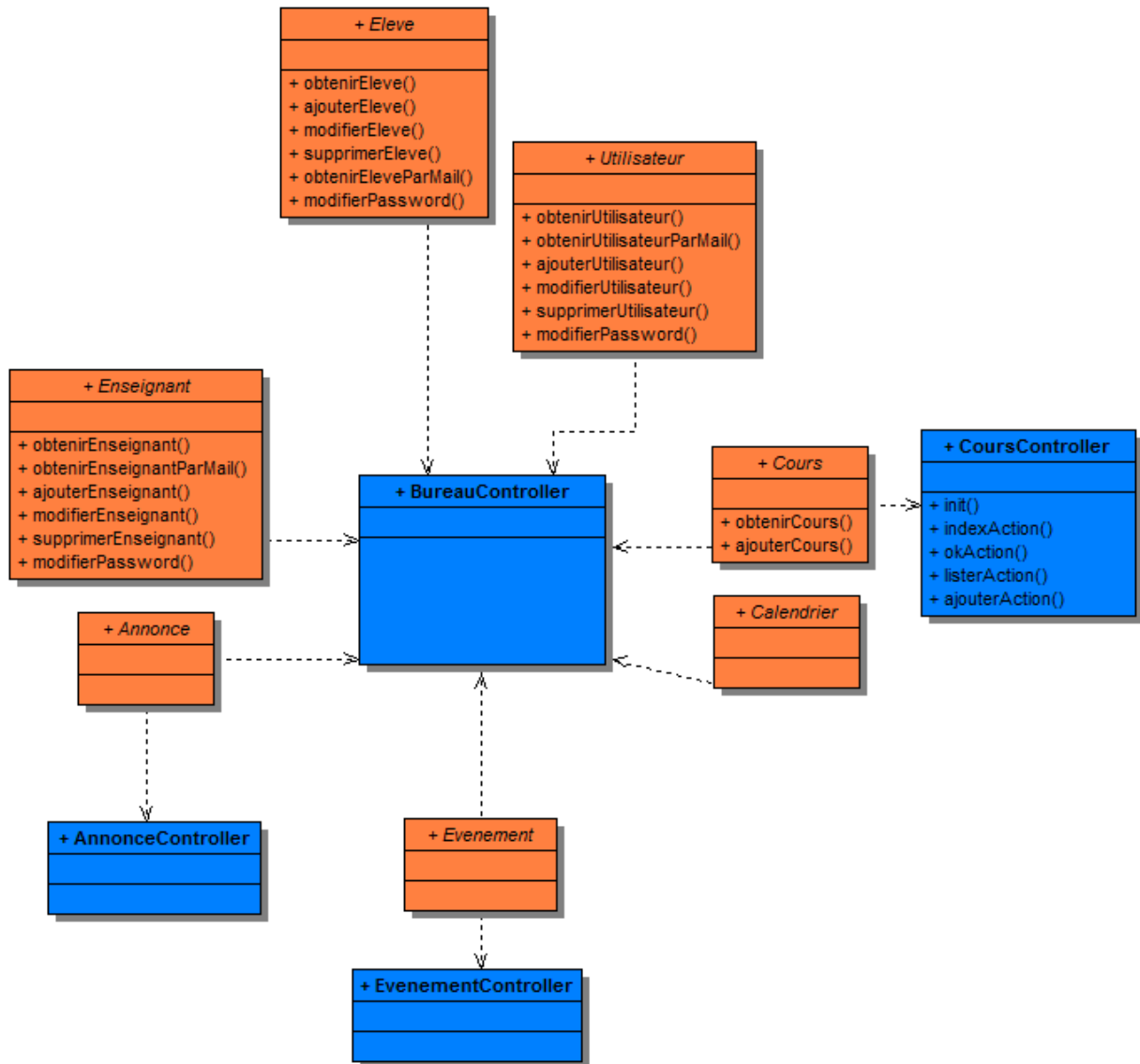


Fig. XIV.4.1 – Diagramme de classe Bureau

Le contrôleur du Bureau permet à un élève qui se connecte de voir l'état de ses tâches à réaliser, cela va de la lecture de cours à la réalisation de TP. Afin de fournir ces informations, la classe « BureauController » doit s'interfacer avec les modèles « Enseignant », « Eleve », « Utilisateur », « Cours », « Calendrier », « Evenement » et « Annonce ».

Ainsi, l'élève qui se connecte à la possibilité de réaliser les actions suivantes :

1. Il peut consulter les annonces que les autres élèves lui envoient.
2. Il peut consulter les événements que ses professeurs lui envoient.
3. Il peut consulter le calendrier des séances de TP.
4. Il peut consulter les cours sous format électronique mise à disposition.
5. Il peut modifier les informations de son profil

Ces classes n'étant pas prioritaires, elles ne seront pas développées dans un premier temps.

XIV.2.2 – Diagramme de classe Cours & Ressources

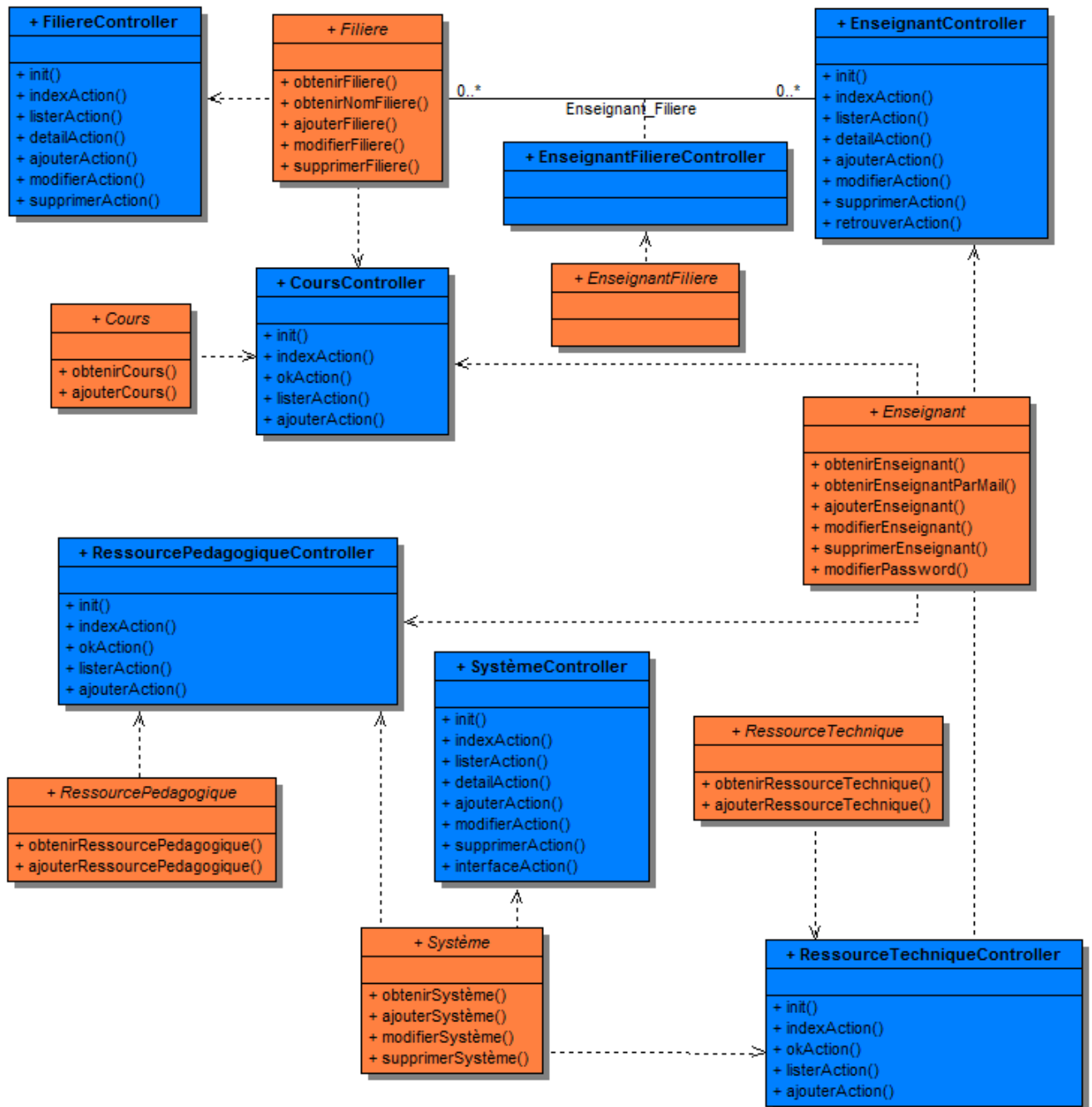


Fig. XIV.4.2 – Diagramme de classe Cours & Ressources

Le contrôleur du Cours & Ressources permet à un élève qui se connecte de l'ensemble de la ressource documentaire mise à disposition pour la réalisation des TP. Afin de fournir ces informations, la classe « CoursController » doit s'interfacer avec les modèles « Enseignant », « Filiere », « Systeme », « Cours », « RessourcePedagogique », « RessourceTechnique » et « EnseignantFiliere ».

Ainsi, l'élève qui se connecte à la possibilité de réaliser les actions suivantes :

1. Il peut consulter les cours rattaché à sa filière auquel il appartient (par ex. section Electrotechnique).

2. Il peut consulter les ressources techniques relatif à un système et mise à disposition par un enseignant.
3. Il peut consulter les ressources pédagogiques relatif à un système et mise à disposition par un enseignant.

Ces classes n'étant pas prioritaires, elles ne seront pas développées dans un premier temps.

XIV.2.3 – Diagramme de classe TP & Systèmes

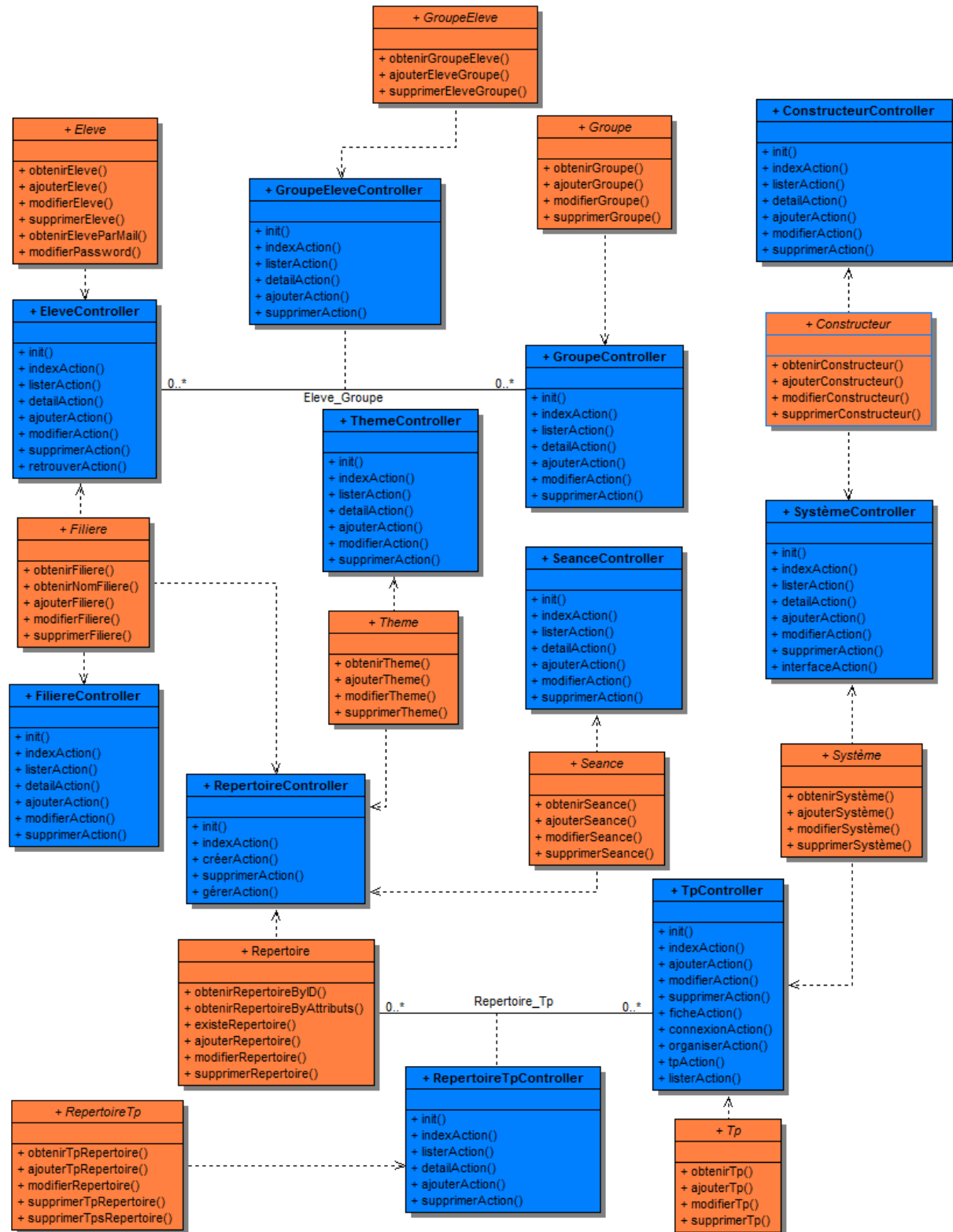


Fig. XIV.4.3 – Diagramme de classe TP & Systèmes

Le contrôleur du TP & Systèmes permet à un enseignant qui se connecte de gérer la liste des TP et les organiser par thématique. L'enseignant peut aussi attribuer la réalisation d'un TP à un élève seul ou à un groupe. Lors des séances de TP, l'enseignant peut suivre l'avancement de celui-ci par les élèves. Les TP disponibles sont rangés dans des répertoires de stockage que seuls les enseignants peuvent administrer.

Ainsi, l'enseignant qui se connecte à la possibilité de réaliser les actions suivantes :

1. Il peut gérer les TP en leurs associant des systèmes.
2. Il peut gérer les systèmes disponibles pour la réalisation de TP.
3. Il peut définir un planning des séances de TP.
4. Il peut communiquer avec les élèves.

Ces classes étant prioritaires, elles seront développées dans un premier temps.

XIV.2.4 – Diagramme de classe Utilisateurs & Authentification

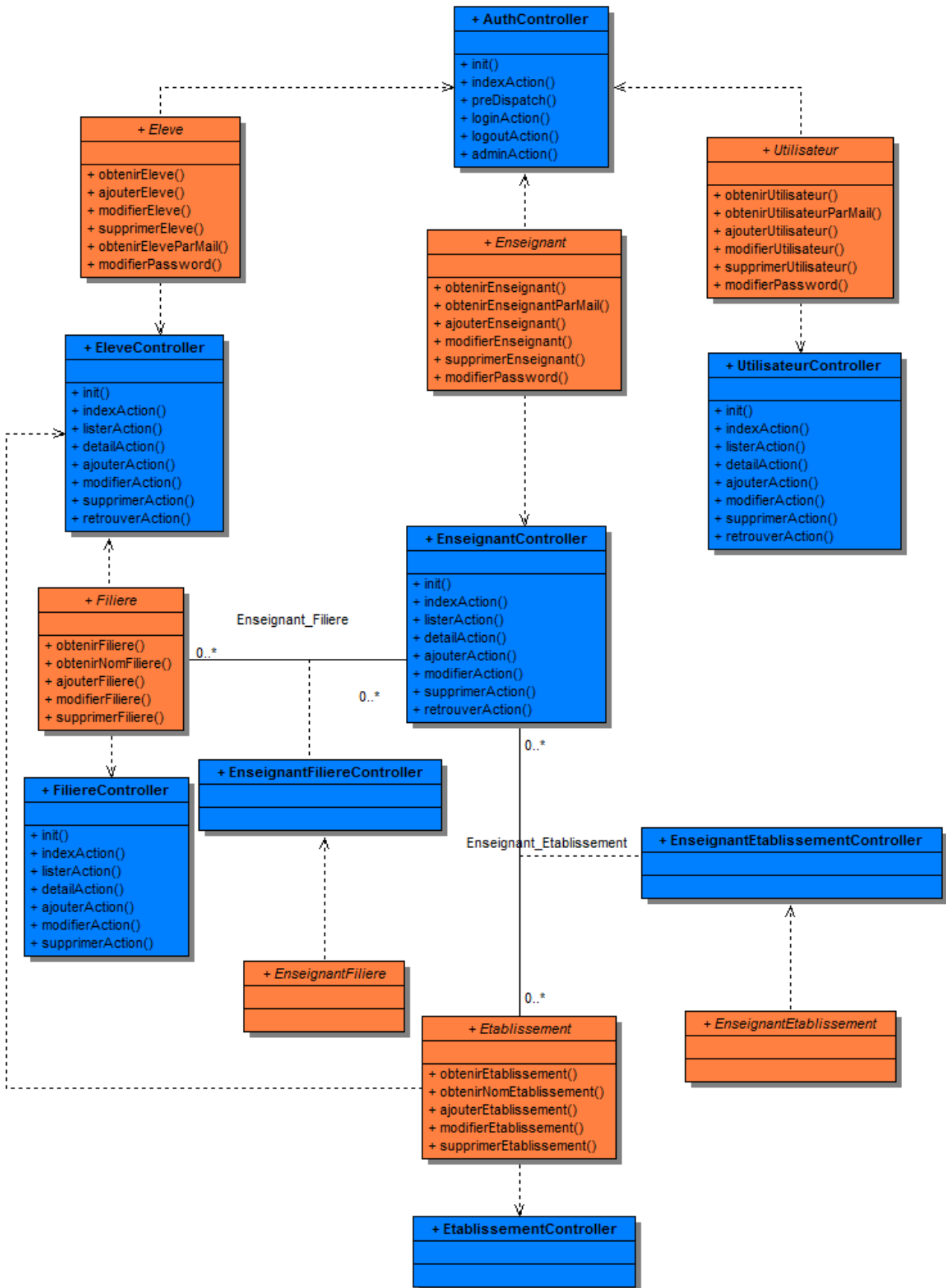


Fig. XIV.4.4 – Diagramme de classe Utilisateurs & Authentification

Le contrôleur du Zone TP permet à un élève qui se connecte de consulter la liste des TP disponibles et d'en réaliser un. L'élève pourra réaliser le TP seul ou en groupe. Les TP sont organisé par thème et par séance. Les TP disponibles sont rangés dans des répertoires de stockage que seuls les enseignants peuvent administrer.

Ainsi, l'élève qui se connecte à la possibilité de réaliser les actions suivantes :

1. Il peut consulter le planning des TP à effectuer
2. Il peut accéder à un TP et l'initialiser et démarrer le déroulement de celui-ci.
3. Il peut accéder à un système et lancer une séquence d'acquisition.
4. Il peut enregistrer et stocker son travail sur le serveur.

Ces classes étant prioritaires, elles seront développées dans un premier temps.

XIV.3 – Présentation des IHM

Avant le développement du Php, j'ai réalisé les interfaces graphiques qui viendront accueillir les fonctions de traitement en Php. Je n'ai pas eu l'ambition de réaliser des IHM performants, mais au moins assez conviviale pour que l'on ait envie de l'utiliser. Et pour les développeurs cela permet d'avoir une idée du résultat attendu.

XIV.3.1 - Interface « Travaux en cours »

Tp en cours	Date	Progression
TP Comportement des charges mécaniques 1	03/02/2011	37 %
TP Assestissement et régulation 2	16/02/2011	60 %
TP. Énergie 3	21/02/2011	45 %
TP. La réversibilité énergétique 4	03/03/2011	20 %
TP. La distribution de l'énergie électrique 5	03/04/2011	18 %

Fig.XIV.3.1 – IHM « Travaux en cours »

Il s'agit d'une interface d'accueil présentant l'état actuel des travaux de l'élève. Il se présente en 2 parties :

- La partie du haut, présente les messages destinés à l'élève et sont organisés dans 3 dossiers.
 - *Boite aux lettres*, ces messages sont ceux échangés individuellement entre élèves et enseignants.
 - *Annonces*, ces messages sont ceux envoyés par les enseignants à tous les élèves. Cela peut s'agir par exemple des dates d'examens.
 - *Evènements*, ces messages sont envoyés de façon automatique ou régulière par les enseignants.

Les messages sont datés, et sont donc consultable sur le jour sélectionné sur le calendrier.

- La partie du bas, présente l'état des TP en cours. Leurs progressions s sont représentées par une barre de pourcentage. Il sera possible de reprendre le TP si celui est disponible à son état d'avancement.

Les fonctionnalités de cette interface ne seront pas réalisés dans un premier temps car je les ai jugé cela non prioritaire.

XIV.3.2 - Interface du tableau des TP à choisir

SECTION ELECTROTECHNIQUE

Centre d'intérêts / Séances	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Cl. 1 - Mise en service d'un système d'un ouvrage ou d'un équipement	Red	Green	Red	Red	Red	Green	Yellow	Yellow	Green						
Cl. 2 - La distribution de l'énergie électrique	Red	Red	Red	Red	Green	Yellow	Yellow	Green							
Cl. 3 - Comportement des charges mécaniques	Red	Red	Red	Green	Green	Yellow	Yellow	Green							
Cl. 4 - Les différents procédés de transformation de l'énergie	Red	Red	Red	Green	Green	Yellow	Yellow	Green	Green						
Cl. 5 - La réversibilité énergétique	Red	Red	Red	Red	Green	Green									
Cl. 6 - Dimensionnement énergétique dans les systèmes autonomes et embarqués	Red	Red	Green	Green	Red	Green	Green								
Cl. 7 - La qualité de l'énergie électrique	Red	Red	Red	Green	Red	Green	Yellow	Yellow	Green						
Cl. 8 - La gestion des coûts	Red	Red	Red	Red	Green	Red	Green	Green							
Cl. 9 - Assèchement et régulation	Red	Red	Red	Yellow	Green	Green	Green	Yellow	Green						
Cl. 10 - Les équipements communicants	Red	Red	Red	Red	Red	Red	Yellow	Yellow	Green	Green					

■ TP non disponible

■ TP à venir

■ TP disponible

Fig.XIV.3.2 – IHM « Sélection des TP »

Il s'agit d'une interface présentant les séances de TP que l'on peut réaliser. Les séances sont organisées par centre d'intérêts et planifiées par semaines. Les séances sont représentés par des codes couleurs correspondant

- **Code Rouge** : Aux séances qui ne contiennent pas de TP à réaliser avec des systèmes communicants ou à des semaines où il n'est pas nécessaire d'utiliser un système.
- **Code Jaune** : Aux séances dont les TP ne sont pas disponibles immédiatement mais prochainement. Il est possible de réaliser des TP de semaines précédentes afin de mieux acquérir une théorie ou d'avancer les TP des prochaines semaines si l'élève est en avance sur son travail.
- **Code Vert** : Aux séances dont les TP sont disponibles.

Si l'on clique sur l'icône « Document » dans l'une des cellules du tableau, cela nous redirige vers la liste des systèmes associés aux TP.

XIV.3.3 - Interface du tableau des systèmes disponibles pour la réalisation de TP

Cl. 1 - Mise en service d'un système d'un ouvrage ou d'un équipement

Travaux pratiques Séance 10

Etat	Titre	Theme	Systeme	Groupe	
Libre	TP sur la régulation de température	Physique	GTI - Bain régulateur		
Libre	TP sur la partie mécanique	Mécanique	GTI - Ascenseur		
Occupé	TP sur le fonctionnement électrique	Electrotechnique	DMS - Centrale de pompage piscine	Groupe 4 (Pivetta, Depardiau, Clavier)	
Occupé	TP sur la partie opérative	Automatisme	DMS - Systeme Photovoltaïque	Groupe 2 (Naudin, Lefevre, Jauris)	
Libre	TP sur les moyens de communication	Automatisme	SPEN - Micro-centrale		

Fig.XIV.3.3 – IHM « Systèmes disponibles »

Il s'agit d'une interface présentant les TP associés à la séance sélectionnée dans l'interface précédente. La liste indique les TP qui sont disponibles et ceux qui sont occupés par des élèves. Ainsi les TP occupés ne sont pas accessibles tant que le groupe d'élèves n'ont pas validé et finaliser leurs TP. Une information permet ainsi d'indiquer les élèves qui utilisent actuellement le système.

L'accès à un système est sur un simple clic sur l'icône de mécanisme situé à droite, dès lors que cette action est effectuée, ce système est mis en état Occupé tant que l'élève n'aura pas validé ou quitter son TP. De même cette action renvoie d'abord vers la fiche détaillée du TP.

XIV.3.4 - Interface de présentation du TP

	TP de consolidation	1
REMISE EN ETAT D'UN SYSTEME DE DISTRIBUTION		
<p>Ce qu'il vous faut :</p> <ul style="list-style-type: none"> > La revue technique. > L'outillage spécifique au modèle. > Une clé dynamométrique est nécessaire. 		
<p>Ce qu'il faut savoir :</p> <ul style="list-style-type: none"> > Une distribution mal calée entraîne en générale la casse du moteur, c'est donc une operation tres delicate. > N'hésitez pas a demandez si vous etes pas sur de vous. 		
	<p>RESSOURCES PEDAGOGIQUES :</p> <ul style="list-style-type: none"> Nom : document pédagogique 1.xls Nom : document pédagogique 2.pdf 	
	<p>RESSOURCES TECHNIQUES :</p> <ul style="list-style-type: none"> Nom : document technique 1.doc Nom : document technique 2.xls Nom : document technique 3.pdf 	



Fig.XIV.3.4 – IHM « Présentation du TP »

Il s'agit d'une interface qui présente rapidement les éléments à disposition (documents, ressources numériques), ainsi que les informations à connaître avant la réalisation du TP. Cette interface intermédiaire permet déjà à l'élève de gagner du temps en préparant d'abord les outils nécessaires pour ce TP. Cela permet aussi de voir par l'élève s'il est préparé ou non à réaliser le TP. Auquel cas, il revient vers la liste des TP pour en choisir une qui lui correspondrait.

XIV.3.5 - Interface de visualisation du système

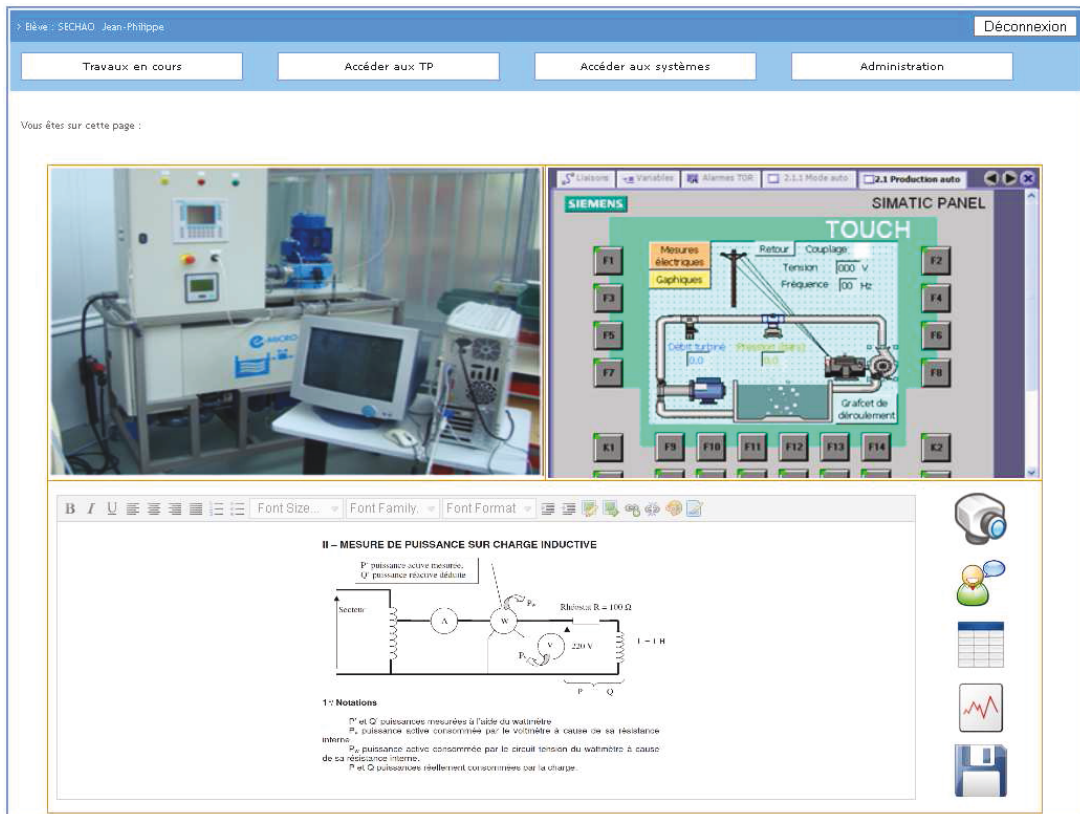


Fig.XIV.3.5 – IHM « Visualisation du système »

Une fois connecté sur un système, nous accédons à une interface séparée en 3 parties.

Nous trouvons en haut à gauche l'écran de retour vidéo du système, grâce à la webcam pilotable et zoomable. A côté de celui-ci se trouve l'IHM embarqué dans le serveur web du système, il permet de prendre en main le système à distance et de visualiser les actions mécaniques de celui-ci.

A cet écran est intégré un éditeur de texte HTML qui permet d'afficher un TP réalisé sous format HTML et de pouvoir y saisir des réponses.

Une barre d'icônes est située à droite de l'éditeur de texte celui-ci contient 5 images correspondant aux actions suivantes :

- Afficher l'écran de la caméra.
- Lancer un chat audio et vidéo avec un enseignant situé à proximité du système.
- Afficher les données d'acquisition sous la forme d'un tableau.
- Afficher les données d'acquisition sous la forme d'un graphique.
- Sauvegarder le travail.

XIV.3.6 - Interface de visualisation des données issues d'une manipulation sous formes graphiques



Fig.XIV.3.6 – IHM « Visualisation des données »

Exemple d’affiche des données sous forme de graphiques (Utilisation de PHP Chart). Il sera possible de pouvoir récupérer ces images pour les imprimer.

XIV.3.7 - Interface de visualisation des données issues d'une manipulation sous forme de tableau

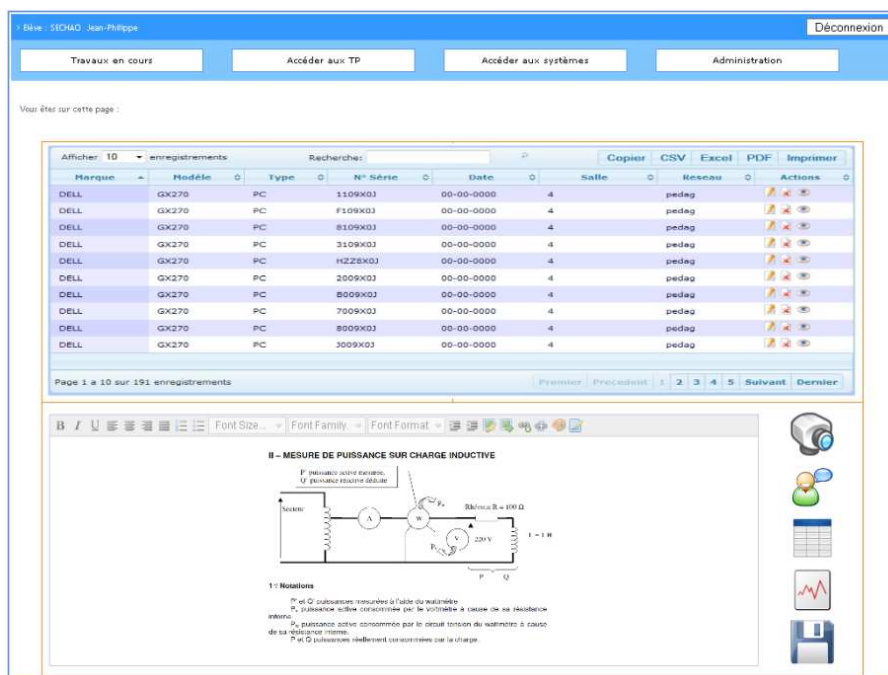


Fig.XIV.3.7 – IHM « Visualisation des données »

Exemple d'affiche des données sous forme de tableau (Utilisation de PHP DataTable). Il sera possible de pouvoir extraire ces données sous différents formats (XML, CSV⁵⁸ notamment).

XIV.4 – Intégration sous Zend Framework

XIV.4.1 - Organisation de l'arborescence

Pour préparer les fichiers sources de l'application, j'ai structuré les dossiers de la manière suivante en respectant les règles de Zend Framework. J'y ai inclus les premiers fichiers nécessaires au démarrage du développement. L'objectif est d'isoler physiquement les différentes logiques (vues, contrôleur, modèles, zone publique, bibliothèques).

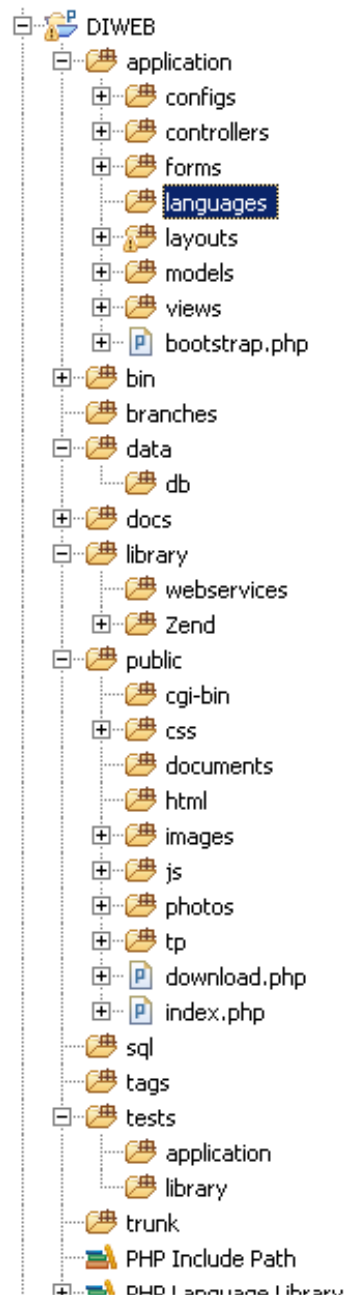


Fig.XIV.4.1 – Arborescence des sources du système

⁵⁸ Comma-separated values : Format de fichiers où les données sont séparées par un caractère spécial

Ce squelette de fichiers propre à MVC, organise les fichiers de la manière suivante :

XIV.4.2 - Le dossier application

Dans ce dossier on retrouve 8 dossiers qui ont tous une fonction bien déterminée.

- **configs**
Dans ce dossier, nous allons mettre notre fichier *application.ini* qui représentera la configuration de notre application. On aura donc à l'adapter pour correspondre à la version de MySQL utilisée, la configurer pour le MVC et définir *l'include path* (pour inclure le Zend Framework à notre projet).
- **forms**
On retrouvera dans ce dossier tous les formulaires de notre application.
- **languages**
Comme son nom l'indique, ici on mettra les fichiers de langue pour avoir une application en anglais et en français !
- **layouts**
Ce sont les fichiers qui vont permettre de définir le style commun à certaines pages, par exemple vous ne voulez pas répéter la structure HTML du header de votre site, c'est ici ! Mais nous y reviendrons par la suite.
- **controllers**
Les contrôleurs sont des classes PHP qui permettent d'effectuer des actions dès lors que l'utilisateur appelle une certaine page et ce contrôleur envoie les informations à une vue correspondante qui affiche le résultat au visiteur.
- **models/DbTable**
Les modèles sont les classes PHP qui permettent de faire des requêtes SQL sur la base de données sans avoir à écrire cette même requête.
- **views**
Ce sont les templates de notre application.
- **plugins**
Les plugins de notre application.

Le dossier library

Ce dossier est à utiliser pour ajouter des composants ou étendre des composants de Zend qui ne vous conviennent pas pour votre application. Par exemple vous voulez ajouter une fonctionnalité à un composant, il faut le mettre dans ce dossier, mais nous y reviendront le moment venu !

Le dossier public

C'est le point d'entrée de notre application. Il contient donc les styles CSS, les images de notre application ainsi que les fichiers JavaScripts.

XIV.4.3 - Mise en place des fichiers de démarrage

Les fichiers de démarrage sont très importants pour que notre application fonctionne correctement. Il est nécessaire de les préparer pour que notre application fonctionne.

Le fichier index.php

Ce fichier doit être situé dans le dossier public/ de notre application qui représente notre point d'entrée et doit contenir ces lignes de code :

```
defined('APPLICATION_PATH')
    || define('APPLICATION_PATH', realpath(dirname(__FILE__) . '/../application'));

defined('LIBRARY_PATH')
    || define('LIBRARY_PATH', realpath(dirname(__FILE__) . '/../library'));

defined('ZEND_PATH')
    || define('ZEND_PATH', realpath(dirname(__FILE__) . '/../../frameworks'));

defined('APPLICATION_ENV')
    || define('APPLICATION_ENV', (getenv('APPLICATION_ENV') ? getenv('APPLICATION_ENV') : 'production'));

// On modifie l'include path de PHP
set_include_path(implode(PATH_SEPARATOR, array(
    realpath(ZEND_PATH),
    get_include_path(),
)));

// On a besoin de Zend Application pour lancer notre application
require_once 'Zend/Application.php';
// On lance la session
require_once 'Zend/Session.php';
Zend_Session::start();

// On crée l'application, on lance le bootstrap et on lance l'application !
$application = new Zend_Application(
    APPLICATION_ENV,
    APPLICATION_PATH . '/configs/application.ini'
);
$application->bootstrap()
    ->run();
```

Le composant `Zend_Application` permet de lancer notre application sans se soucier de grand-chose, en effet, vous avez à lui passer l'environnement de l'application ainsi que le fichier de configuration.

Le fichier `application.ini`

Nous ajoutons quelques lignes à `application/configs/application.ini` (ZF utilise PDO pour l'accès aux données).

```
database.adapter = PDO_MYSQL
database.params.host = localhost
database.params.username = root
database.params.password =
database.params.dbname = diweb
```

Puis nous exploitons ces informations au démarrage (dans `index.php`)

```
-- dans index.php
```

```
// Chargement automatique de Zend_Db_Adapter_Pdo_Mysql, et instanciation.
$config = new Zend_Config_Ini('./application/configs/application.ini', 'production');
```

```
$db = Zend_Db::factory($config->database->adapter,array(
    'host'    => $config->database->params->host,
    'username' => $config->database->params->username,
    'password' => $config->database->params->password,
    'dbname'  => $config->database->params->dbname,
    )
);
```

```
// Placons la connexion dans un registre global à l'application
$registry = Zend_Registry::getInstance();
$registry->set('db', $db);
```

```
// En faire la connexion par défaut
Zend_Db_Table::setDefaultAdapter($db);
```

On remarquera la souplesse d'utilisation de `Zend_Config_Ini` : les clés du fichier de configuration sont traduites en propriétés public, ce qui nous permet de consulter une valeur associée à une clé par : `instance_Zend_Config_Ini->clé`.

Nous passerons les arguments par l'URL. Le mode `rewrite` étant activé sur le serveur HTTP, les arguments peuvent être inscrits dans une expression de chemin :

```
http://localhost/diweb/eleve/voir/id/1
```

Équivalent à :

```
http://localhost /diweb/eleve/voir.php?id=1
```

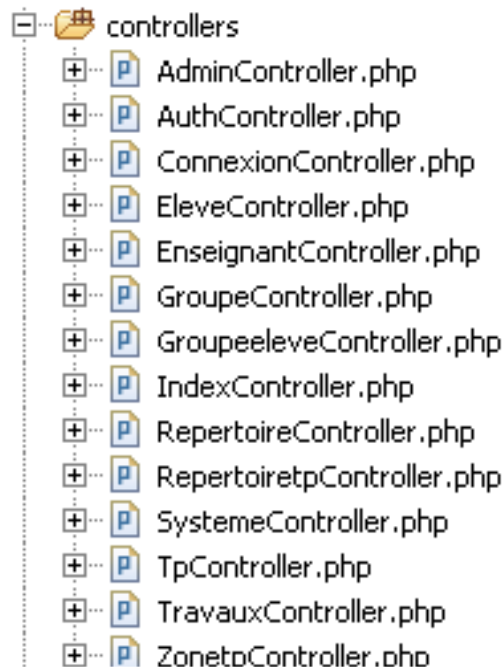
Sauf que l'expression de chemin est d'une apparence plus stable, donc plus facile à gérer pour les moteurs de recherche, entre autres.

La syntaxe générale de passage d'arguments : `.../nom1/val1/nom2/val2/.../nomN/valN`

XIV.4.4 – Ajout des fichiers de type Controller

Nous allons concevoir les contrôleurs de l'application. Nous allons prendre pour exemple le cas d'utilisation d'un élève. Nous allons donc mettre en place un **contrôleur de cas d'utilisation**. Ce contrôleur sera accessible par l'url : <http://localhost/diweb/eleve> par exemple pour le contrôleur *EleveController.php*

Par défaut, ZF fait correspondre à cette ressource un script PHP bien particulier. Hormis le fait que le nom de ce script est basé sur le nom de la ressource (ici Country) suivi du mot Controller (on remarquera la capitalisation des termes), ce script, qui est en fait une **classe** héritant de **Zend_Controller_Action**, sera placé dans la branche Controller :



J'ai ensuite développé le script *EleveController.php* de la manière suivante :

```
1 <?php
2
3 require_once 'Zend/Controller/Action.php';
4
5 /**
6  * @name      : EleveController
7  * @package   :
8  * @property  : Gestion des Elèves
9  * @author   : Jean-Philippe SECHAO
10 * @version  : v.1.0
11 * @date     : 05/2010.
12 */
13 class EleveController extends Zend_Controller_Action {
14
15     * @method   : init
16     public function init () {
17         //Initilisation de vue et contrôle d'accès
18         $this->initView();
19         $this->view->baseUrl = $this->_request->getBaseUrl();
20         $session = Zend_Auth::getInstance();
21         $this->view->eleve = $session->getIdentity();
22     }
23
24     * @method   : indexAction
25     public function indexAction () {
26         // TODO Auto-generated EleveController::indexAction() default action
27     }
28
29     * @method   : listerAction
30     public function listerAction () {
31         //
32         $this->view->title = "Liste des élèves";
33         // Liste des Elèves
34         $eleve = new Eleve();
35         $this->view->eleveArray = $eleve->fetchAll();
36     }
37
38     * @method   : detailAction
39     public function detailAction () {
40         //
41         $eleve = new Eleve();
42         $id = $this->_getParam('id', 0);
43         $this->view->title = "Fiche de l'élève";
44         $student = $eleve->obtenirEleve($id);
45         $this->view->eleveDetail = $student;
46     }
47 }
```

Présentons rapidement ce script PHP

- C'est une *classe* qui **hérite** de `Zend_Controller_Action`, une classe de framework.
- *init*: appelé une seule fois, à l'instanciation de la classe (à l'image d'un constructeur)
- *preDispatch*: appelé avant un appel automatique à une *méthode d'action*
- *xxxAction* : des méthodes d'action
- *postDispatch*: appelé après un appel automatique à une *méthode d'action*

Méthode d'Action

Une *Méthode d'Action* est une méthode d'une sous-classe de `Zend_Controller_Action` identifiée par un nom se terminant par **Action**. Par exemple : `ajouterAction`, `supprimerAction`.

Qu'est-ce qu'une « Action » ? C'est le « service » associé à une ressource. Par exemple, dans la requête suivante :

http://localhost/diweb/eleve/lister

La ressource demandée est de lister les élèves. Dans un contexte ZF de base, le contrôleur `EleveController.php` (qui sera automatiquement chargé, suivi d'une instanciation) devra contenir la méthode `listerAction`, car c'est cette dernière qui sera appelée par le framework.

L'action par défaut : En absence de nom de service l'action « `indexAction` » sera invoquée. Par exemple, dans la requête suivante :

http://localhost/diweb

la méthode `indexAction` de l'objet, instance de `EleveController` (ou `IndexController` par défaut), sera appelée.

Il en va de même avec la ressource. En absence de nom de ressource, c'est la ressource « `index` » qui sera sollicitée. Par exemple, dans la requête suivante :

http://localhost/diweb

Sera interprété comme:

http://localhost/diweb/index/index

Cela suppose donc que le contrôleur `IndexController` et sa méthode `indexAction` existent !

Ajout des modèles de données :

Dans une application de gestion typique (interface avec un SGBDR⁵⁹), il n'est pas rare d'avoir un modèle qui colle aux tables concernées par les responsabilités de l'application. Dans ce contexte, il paraît alors logique de charger le modèle des responsabilités d'accès aux données : `create`, `retrieve`, `update`, `delete` (CRUD⁶⁰).

Il est commode alors de faire **hériter** les classes du modèle par une classe prenant en charge les interactions typiques avec un SGBDR. ZF propose la classe `Zend_Db_Table_Abstract` comme parent de base des classes du modèle.

Nous allons voir maintenant comment lier une classe à cette table.

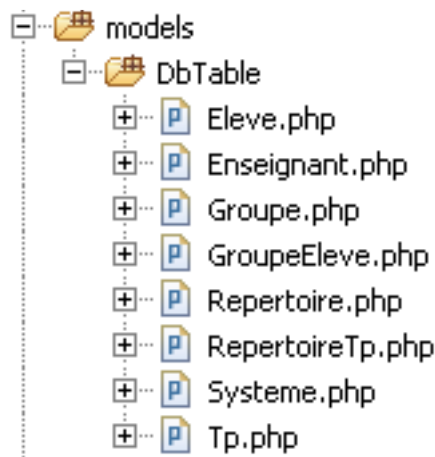
⁵⁹ *Systèmes de gestion de base de données relationnelle*

⁶⁰ *Create, Read, Update, Delete*

XIV.4.5 – Utilisation de Zend Db Table, Zend Db Row

Nous avons avec la base de données (le SGBDR), deux types de structure à gérer : La table et ses lignes. ZF met à notre disposition deux super-classes `Zend_Db_Table_Abstract` et `Zend_Db_Table_Row_Abstract`

Concevons une classe modèle représentant une instance d'un élève : `Eleve.php`. Les classes du modèle seront placées dans le dossier « `models /DbTable` ».



Les modèles contenu dans le dossier `DbTable` sont pour la plupart utilisés pour des opération CRUD, les modèles contenu dans le dossier `models` sont des fonction spécifiques du modèle par ex. `CalculerAge()` du modèle `Eleve`.

Opérations CRUD

Les opérations de base SQL sur les données (interrogation, création, modification, suppression) sont assistées par des méthodes héritées de `Zend_Db_Table_Abstract`. En effet, ces responsabilités sont conceptuellement soit de niveau « table », soit de niveau « objet métier ('tuple') ».

En voici quelques exemples :

Sélectionner une ligne, un objet métier

Recherche le pays d'id = 1

```
// Fonction de niveau 'table'  
$eleve = new Eleve();  
$secoles = $eleve->find(1)->current();
```

La méthode `find` prend comme argument un identifiant (un ou plusieurs arguments selon le cas), et retourne un `Rowset`.

La méthode `current` d'un `Rowset` retourne l'élément courant sous la forme d'un objet `Row` (ou `NULL` s'il n'existe pas. Dans notre cas c'est le premier et unique `CountryRow`.

J'ai ensuite développé le script `Eleve.php` de la manière suivante :

```

1 <?php
2
3
4 * @name      : class Eleve[]
12 class Eleve extends Zend_Db_Table_Abstract (
13
14     protected $_name = 'eleve';
15
16
17 * @method      : obtenirEleve[]
23 public function obtenirEleve($id) {
24     //
25     $id = (int)$id;
26     $row = $this->fetchRow('id = ' . $id);
27     //
28     if (!$row) {
29         throw new Exception("Count not find row $id");
30     }
31     return $row->toArray();
32 }
33
34
35 * @method      : ajouterEleve[]
52 public function ajouterEleve($password, $nom, $prenom, $adresse, $cp, $ville, $mail, $telephone) {
53     //
54     $data = array(
55         'login'     => strtolower($prenom.".".$nom),
56         'password'  => md5($password),
57         'nom'       => $nom,
58         'prenom'   => $prenom,
59         'adresse'  => $adresse,
60         'cp'       => $cp,
61         'ville'    => $ville,
62         'mail'     => $mail,
63         'telephone' => $telephone
64     );
65     //
66     $this->insert($data);
67 }
68
69
70 * @method      : modifierEleve[]
87 public function modifierEleve($id, $nom, $prenom, $adresse, $cp, $ville, $mail, $telephone) {
88     //
89     $data = array(
90         'nom'       => $nom,
91         'prenom'   => $prenom,
92         'adresse'  => $adresse,
93         'cp'       => $cp,
94         'ville'    => $ville,
95         'mail'     => $mail,
96         'telephone' => $telephone
97     );
98     //

```

Ajout des vues :

Comme on a pu le constater, ZF, par défaut, s'attend à ce que le développeur ait conçu des vues pour chaque méthode d'action.

Diweb

```

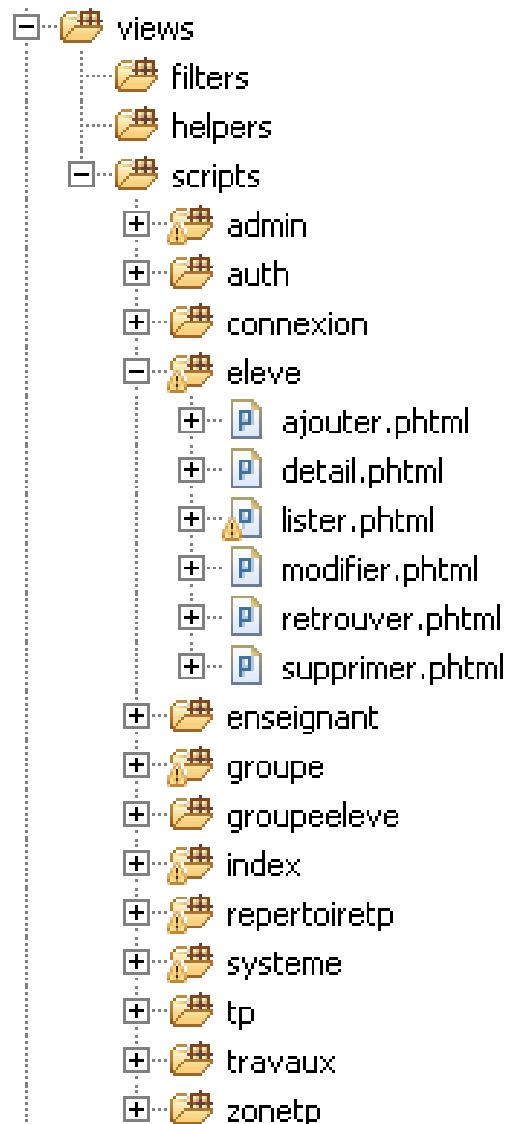
|-- application
| |-- controllers
| | |-- EleveController.php
| |
| |-- models
| |-- views
| | |-- scripts
| | | |-- eleve
| | | |-- index.phtml
|-- library
| |-- Zend (racine des librairies de ZF)
|-- public
| |-- styles
| |-- images
|-- js

```

Les vues sont stockées dans la branche racine/application/views/scripts/xxx/yyy, ou xxx est le nom de la ressource sollicitée (en référence au contrôleur), et yyy le nom de la vue en relation, de la forme **nomDuService.phtml**. Le nom du service étant la partie gauche du nom de la méthode d'action. Par exemple :

http://localhost/diweb/eleve/ajouter

la méthode ajouterAction de l'objet, instance de EleveController, sera appelée, puis la vue application/views/scripts/eleve/ajouter.phtml sera traitée et retournée au client web. Bien entendu, toute cette logique de traitement est entièrement prise en charge par le framework.



Exemple de script contenu dans les fichiers de vues :

```
1 <table>
2   <tr>
3     <td colspan="3">&nbsp;</td>
4   </tr>
5   <?php $eleve = $this->eleveDetail; ?>
6   <tr>
7     <td>Nom :</td>
8     <td>&nbsp;</td>
9     <td><?php echo $this->escape($eleve['nom']);?></td>
10  </tr>
11  <tr>
12    <td>&nbsp;</td>
13  </tr>
14  <tr>
15    <td>Prenom :</td>
16    <td>&nbsp;</td>
17    <td><?php echo $this->escape($eleve['prenom']);?></td>
18  </tr>
19  <tr>
20    <td>&nbsp;</td>
21  </tr>
22  <tr>
23    <td>Adresse :</td>
24    <td>&nbsp;</td>
25    <td><?php echo $this->escape($eleve['adresse']);?></td>
26  </tr>
27  <tr>
28    <td>&nbsp;</td>
29  </tr>
30  <tr>
31    <td>Code Postal :</td>
32    <td>&nbsp;</td>
33    <td><?php echo $this->escape($eleve['cp']);?></td>
34  </tr>
35  <tr>
36    <td>&nbsp;</td>
37  </tr>
38  <tr>
39    <td>Ville :</td>
40    <td>&nbsp;</td>
41    <td><?php echo $this->escape($eleve['ville']);?></td>
42  </tr>
43  <tr>
44    <td>&nbsp;</td>
45  </tr>
46  <tr>
47    <td>Mail :</td>
48    <td>&nbsp;</td>
49    <td><?php echo $this->escape($eleve['mail']);?></td>
50  </tr>
```

XIV.4.6 – Contrôleur et Vues

Contrôleurs et Vues sont liés par contrat. Le contrôleur passe des données à la vue, cette dernière doit en connaître la nature, c'est à dire le « nom » et le « type associé ». Le contrôleur passera un objet de type "RowSet" (ensemble de lignes) d'objets de type "Row" (ligne).

Les objets de type [Zend_Db_Table_Row](#), sont automatiquement spécialisés par ZF en fonction du modèle. Le script de la vue associé à l'action est : *views/eleve/index.phtml*. En effet, les attributs automatiques de notre modèle sont directement accessibles (non privés).

```
<table>
<?php foreach ($this->countries as $country) : ?>
  <tr>
    <td> <?php echo $this->escape($country->id) ?> </td>
    <td> <?php echo $this->escape($country->Name) ?> </td>
    <td> <?php echo $this->escape($country->Ville) ?> </td>
    <td> <?php echo $this->escape($country->Etablissement) ?> </td>
  </tr>
<?php endforeach; ?>
</table>
```

Ce script utilise la méthode « escape » pour échapper les caractères (Par défaut, la méthode `escape()` utilise la fonction PHP `htmlspecialchars()` pour l'échappement).

A noter la syntaxe à la « endif », qui favorise la lecture (les accolades ouvrantes et fermantes sont parfois difficiles à repérer dans la vue).

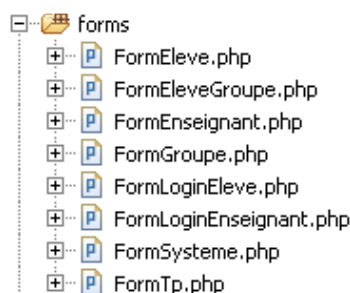
Le contrat qui lie la vue et le contrôleur est simple : La vue considère qu'un contrôleur lui a mis à disposition, sous l'appellation **eleves**, une collection d'objets de type `Eleve`.

Ajout des formulaires et validateurs :

Comme tout framework qui se respecte, ZF propose des composants dédiées aux formulaires des applications Web. ZF propose l'API `Zend_Form`, associé à `Zend_Validate`, `Zend_Filter` pour la création et la gestion des formulaires (présentation, validation, retour au client). Un formulaire est un peu plus qu'une vue, il permet à l'utilisateur de transmettre de multiples informations issues de listes déroulantes, cases à cocher, champs d'édition de texte...

De plus, les données d'un formulaire doivent être filtrées (pour se prémunir d'actions dangereuses) et validées (pour assurer la cohérence des données), ceci pouvant entraîner des aller/retour entre le serveur et le client. Ce type de comportement est pré-programmé par ZF, et pour en hériter, nous devons faire ... hériter nos formulaires de `Zend_Form`.

Tout d'abord, je crée mes formulaires dans le dossier *forms* :



La classe *FormEleve* hérite de *Zend_Form*. Le formulaire *EleveForm.php* est à placer dans */application/forms*.

Quelques explications :

- Le constructeur se charge de la structure du formulaire, élément par élément.
- Des contrôles HTML de type `input='text'` sont créés via `Zend_Form_Element_Text`
- Les champs de saisis requis sont spécifiés (`required`)
- A chacun des contrôles d'entrée sont associés filtres, validateurs et vue

```
1 <?php
2
3 * Enter description here ...
4
5 class FormEleve extends Zend_Form {
6
7     * Enter description here ...
8
9     public function __construct($options = null) {
10
11         parent::__construct($options);
12         $this->setName('Eleve');
13
14         //
15         $id = new Zend_Form_Element_Hidden('id');
16
17         // Input "Password"
18         $password = new Zend_Form_Element_Password('password');
19         $password->setLabel('Mot de passe :');
20         $password->setRequired(true);
21         $password->addFilter('StripTags');
22         $password->addFilter('StringTrim');
23         $password->addValidator('regex', false, array('[0-9]*'));
24         $password->getValidator('regex')->setMessage("Le mot de passe doit contenir au moins un chiffre !", Zend_V
25         $password->addValidator('StringLength', false, array('MIN' => 5, 'MAX' => 10));
26         $password->getValidator('StringLength')->setMessage("Le mot de passe est trop court !", Zend_Validate_Stri
27         $password->getValidator('StringLength')->setMessage("Le mot de passe est trop long !", Zend_Validate_Strir
28
29         // Input "Password validation"
30         $password_bis = new Zend_Form_Element_Password('password_bis');
31         $password_bis->setLabel('Mot de passe :');
32         $password_bis->setRequired(true);
33         $password_bis->addFilter('StripTags');
34         $password_bis->addFilter('StringTrim');
35         $password_bis->addValidator('regex', false, array('[0-9]*'));
36         $password_bis->getValidator('regex')->setMessage("Le mot de passe doit contenir au moins un chiffre !", Zend_V
37         $password_bis->addValidator('StringLength', false, array('MIN' => 5, 'MAX' => 10));
38         $password_bis->getValidator('StringLength')->setMessage("Le mot de passe est trop court !", Zend_Validate_Stri
39         $password_bis->getValidator('StringLength')->setMessage("Le mot de passe est trop long !", Zend_Validate_Strir
40
41         // Input Name
42         $nom = new Zend_Form_Element_Text('nom');
43         $nom->setLabel('Nom :');
44         $nom->setRequired(true);
45         $nom->addFilter('StripTags');
46         $nom->addFilter('StringTrim');
47         $nom->addValidator('NotEmpty');
48
49         // Input Firstname
50         $prenom = new Zend_Form_Element_Text('prenom');
51         $prenom->setLabel('Prenom :');
52         $prenom->setRequired(true);
53         $prenom->addFilter('StripTags');
54         $prenom->addFilter('StringTrim');
55         $prenom->addValidator('NotEmpty');
```

```
class Zend_Form_Element_Password extends Zend_Form_Element_Xhtml
{
    /**
     * Use formPassword view helper by default
     * @var string
     */
    public $helper = 'formPassword';

    /**
     * Whether or not to render the password
     * @var bool
     */
}
```

XIV.4.7 – Qu'est-ce qu'un filtre ?

Un filtre est une **fonction de transformation** d'une donnée d'entrée, soumise par un client, en une donnée *conforme* pour le système : simplification, ajout de code d'échappement...

Il existe de nombreux filtres prêts à l'emploi. Un filtre est un objet, instance d'une classe qui implémente *Zend_Filter_Interface*.

Par exemple :

```
$eleve = new Zend_Form_Element_Text('Nom');
$eleve->setLabel('Nom élève');
$eleve->setRequired(true);
$filter = new Zend_Filter_StripTags();
$eleve->addFilter($filter);
```

Le filtre *StripTags* supprimera toute balise présente dans prénom (afin de se prémunir d'injection de code HTML).

Les filtres peuvent être chaînés :

```
...
$filter = new Zend_Filter_StripTags();
$nom->addFilter($filter);
$nom->addFilter(new Zend_Filter_StringToUpper());
...
```

Dans ce cas, les filtres sont appliqués par ordre d'insertion dans le composant. Ici, les balises seront supprimées avant la passage en majuscule.

XIV.4.8 – Qu'est-ce qu'un validateur ?

Un validateur est une **fonction booléenne** qui reçoit une donnée d'entrée, et retourne vrai si son argument est *conforme* aux règles de gestion prises en charge par le validateur, retourne faux sinon.

De plus, le validateur peut fournir des informations sur les règles non respectées, comme une chaîne vide, ou trop courte ou ne respectant pas une certaine syntaxe, etc.

Il existe de nombreux validateurs prêts à l'emploi. Un validateur est un objet, instance d'une classe qui implémente *Zend_Validate_Interface*.

Par exemple :

```
$eleve = new Zend_Form_Element_Text('Nom_Eleve');
$eleve->setLabel('Nom : ');
$eleve->setRequired(true);
$eleve->addValidator(new Zend_Validate_NotEmpty());
```

Les validateurs peuvent être chaînés :

```
...
$eleve->addValidator(new Zend_Validate_StringLength(2));
$eleve->addValidator(new Zend_Validate_Alnum());
// ou $headstate->addValidator('Alnum'); // le framework se chargeant de la correspondance
```

Dans ce cas, les validations sont appliquées par ordre d'insertion dans le composant. Ici, le test de la longueur minimale est réalisé en premier, puis, *quelque soit le résultat* le seconde test sera lui aussi évalué (caractères alphanumériques uniquement).

XIV.5 – Ajout des bibliothèques javascript et répertoire web

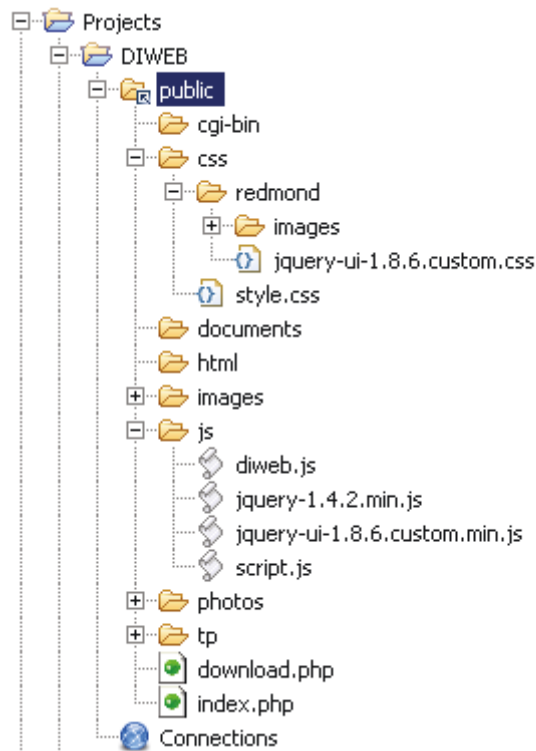
Afin de rendre le site plus convivial, de lui ajouter des effets sympas ou des fonctionnalités utiles, on y ajoute souvent du javascript. Mais le Javascript est un langage un peu compliqué, et déroutant car sa syntaxe dépend du navigateur. Avec la librairie JQuery, on s'affranchit de tous ces problèmes, et bien plus.

Une librairie est un fichier — ou plusieurs — que charge la page web pour avoir accès aux nouvelles fonctionnalités qu'elle offre. Dans le cas de JQuery, c'est un fichier unique de 25ko environ.

En anglais, on peut entendre le terme *framework*.

Dans la structure de Zend on peut ajouter la librairie de JQuery dans le répertoire **/public/js/**. Il nous suffira par la suite de le configurer dans le fichier `index.php`.

Arborescence des répertoires dans /public :



Dans ce répertoire public, nous retrouvons 8 dossiers ayant les rôles suivants :

- « *cgi-bin* », emplacement des scripts CGI ou des exécutables.
- « *css* », emplacement des feuilles de style.
- « *documents* », emplacement des documents au format électronique (pdf, word, excel ...)
- « *html* », emplacement des documents au format html.
- « *images* », emplacements des images du site.
- « *js* », emplacements contenant les fichiers et bibliothèques javascript.
- « *photos* », emplacement contenant des photos uploadé ou chargé sur le répertoire.
- « *tp* », emplacement des fiches de TP de l'application.

Exemple de codes sources javascript.

```
function setContentInnerHTML(divId, popup_url, optionsUrl) {
    var texte = "";
    if (texte = file('index.php?popup_url=' + popup_url + '&' + optionsUrl)) {
        document.getElementById(divId).innerHTML = texte;
    }
}

function changeStateOfComponent(component, type, id) {
    file('index.php?popup_url=popup_changeStateComponent&type=' + type + '&id='
        + id + '&state=' + component.checked);
}

function ajaxCmd(popup_url,options) {
    if (popup_url == 'delete') {
        if (confirm('Etes-vous sûr de vouloir supprimer cet objet ?' )) {
            return file('index.php?popup_url='+popup_url+'&' + options);
        }
    }
    else {
        return file('index.php?popup_url='+popup_url+'&' + options);
    }
}

function addIdToHiddenField(id, value, hiddenFieldName) {
    var hiddenField = document.getElementById(hiddenFieldName);
    var str = hiddenField.value;
    var reg = new RegExp("\\[" + id + ",[^\\]]+\\]", "g");
    if (str.match(reg)) {
        str = str.replace(reg, "[" + id + "," + value + "]");
        hiddenField.value = str;
    } else {
        hiddenField.value = hiddenField.value + "[" + id + "," + value + "]"
            + '|';
    }
}

function addIdFromCBToHiddenField(id, value, hiddenFieldName) {
    var hiddenField = document.getElementById(hiddenFieldName);
    var str = hiddenField.value;
    var reg = new RegExp("\\[" + id + "\\]\\|", "g");
    if (str.match(reg)) {
        if (value) {
            str = str.replace(reg, "[" + id + "]" | "");
        }
    }
}
```

Les variables en JavaScript n'ont pas de type défini, et n'importe quelle valeur peut être stockée dans n'importe quelle variable. Les variables peuvent être déclarées avec **var** .

Ces variables ont une portée lexicale et une fois la variable déclarée, on peut y accéder depuis la fonction où elle a été déclarée. Les variables déclarées en dehors d'une fonction et les variables utilisées sans avoir été déclarées en utilisant var, sont globales (peuvent être utilisées par tout le programme).

Exemple de codes sources css

```
/* ###DIV### */
div#corp {
    height: 600px;
}

div#left {
    float: left;
    width: 318px;
    margin: 0px;
    padding: 0px;
}

div#contenu-left {
    padding-top: 20px;
    padding-bottom: 20px;
    padding-left: 63px;
    padding-right: 25px;
}

div#title {
    height: 26px;
    width: 216px;
    margin-bottom: 10px;
    padding-top: 5px;
    padding-left: 10px;
    font-family: "Times New Roman", Times, Verdana;
    font-size: 13px;
    color: #000000;
    font-weight : bold;
    font-style: italic;
}

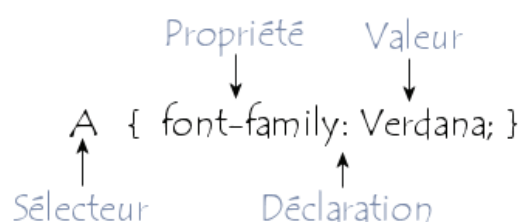
div#contenu-menu {
    padding-bottom: 0px;
    padding-left: 10px;
    padding-bottom: 18px;
}

div#menu {
    margin-left: 10px;
}
```

La définition d'un style se fait à l'aide de règles en texte simple permettant de décrire l'aspect des éléments de la page. Une règle CSS est caractérisée par deux principaux éléments :

- Un **sélecteur de balises**, permettant de préciser à quelles balises du document le style s'applique;
- Une **déclaration** de style, définie entre accolades, permettant de préciser le style à appliquer aux balises sélectionnées. La déclaration est elle-même constituée :
 - d'une ou plusieurs **propriété(s)**, suivie du caractère « : » (double point),
 - d'une ou plusieurs **valeur(s)** associée(s) à chaque propriété, entourée de guillemets s'il s'agit de plusieurs mots ou séparés par des virgules s'il y en a plusieurs, suivie d'un point virgule.

La syntaxe est ainsi la suivante :



XIV.6 – Méthodes de récupération des données

Les modules ETZ 510 intègrent de base un serveur web qui permet d'effectuer un diagnostic à distance à l'aide de services prédéfinis.

Le serveur web est limité au diagnostic de premier niveau. Les pages standard associées (non modifiables) sont prêts à l'utilisation et ne nécessitent aucune configuration utilisateur. On peut toutefois y ajouter des pages créées par l'utilisateur, le serveur web disposant d'une capacité de 8 Mo de mémoire disponible.

La mise au point des pages Web peut se faire en mode connecté ou en mode simulation, il permet de tester le fonctionnement de l'application Web sans module (donc sans connexion physique à un automate) simplifiant le travail de mise au point.

Les pages web utilisateurs créées avec l'éditeur graphique intégré du module ETZ 510 (Factory Cast) se basent sur les technologies web (HTML et Java), elles permettent l'accès en temps réel aux variables des automates grâce à la bibliothèque d'objet graphiques (Javabeans).

Ainsi, le module offre 2 possibilités pour la récupération de données en temps réel. Soit par l'archivage de données via le service *Data Logging* soit par *services web SOAP/XML*.

XIV.6.1 – Archivage de données

- Ce service permet l'archivage des données de l'application : événements, alarmes, données de processus, état des équipements, mesures, etc...
- Les données sont consignées dans des fichiers CSV au format ASCII⁶¹, qui sont stockés localement sur la carte mémoire SD du module.

XIV.6.2 – Services Web SOAP/XML

Les modules réseau Ethernet intègrent un serveur de données au standard SOAP/XML qui permet l'interopérabilité directe entre les équipements d'automatisme et les applications de niveau informatique (MES, ERP, SAP, application .Net, ...).

Ces services Web embarqués dans les automates sont conformes aux normes de services Web **W3C** (*World Wide Web Consortium*). Ils offrent des moyens de communication ouverts et normalisés grâce auxquels les périphériques du niveau commande peuvent interagir directement avec les applications de gestion informatique en utilisant un protocole SOAP non propriétaire.

Les services Web SOAP/XML reposent sur les normes suivantes :

- **SOAP** (*Single Object Access Protocol*), le protocole d'échange exécuté via le canal HTTP (*HyperText Transfer Protocol*),
- **WSDL** (*Web Services Description Language*), le langage de description de services Web au format XML,
- **XML** (*eXtensible Markup Language*), la norme universelle pour l'échange de données.

⁶¹ *American Standard Code for Information Interchange* : norme de codage de caractères en informatique

XIV.6.3 – Services Web ModbusXMLDa : interface de serveur SOAP

L'implémentation des services **ModbusXMLDa ()** dans les serveurs Web des équipements d'automatisme permet à l'informaticien de créer facilement sa propre application qui accèdera à l'information souhaitée directement dans l'automate et en temps réel.

L'initiative des échanges appartient à l'application client SOAP (le serveur répond à ces requêtes). Les échanges d'information se font au format standard XML en réponse à une requête utilisant le protocole SOAP.

XIV.7 – Tests unitaires

Lorsqu'on parvient à structurer correctement un projet web, il devient possible de le tester. Les tests, au même titre que la documentation, font partie du cycle de développement de l'application web. Le recours à la programmation orientée objet et à la séparation des fonctionnalités dans un programme permettent d'organiser et d'industrialiser des procédures de tests.

Les tests ont pour objectif notamment de simplifier la maintenance de la manière suivante :

- Tester les attentes souhaitées
- Tester les comportements décrits et identifiés dans l'application
- Tester nous informe quand des changements cassent du code ou des comportements existants.

Pour notre application nous ferons appel au framework **PHPUnit** couplé à Zend Framework. Afin de nous familiariser avec ceux-ci, ces tests se porteront notamment sur les couches bases de données, qui sont une étape fondamentale dans le développement d'applications.

Il nous permet d'assurer que toutes les opérations de données se comportent correctement, sans perte de données. Je vais décrire un moyen simple de tester la couche base de données de notre application, en utilisant des classes de bases de données de PHPUnit et de Zend Framework.

J'ai écrit une classe de cas de tests de base de données qui va me permettre de vérifier que chaque opération de lecture récupère les données correctes et que chaque opération d'écriture affecte la base de données comme prévu.

Les données de l'application peuvent résider sur plusieurs systèmes de stockage différents. Cependant, nous pouvons identifier quatre fonctions génériques : *Create*, *Read* (ou *Retrieve*), *Update* et *Delete*, habituellement appelé opérations CRUD. Elles correspondent à :

- *Create* (*Créer*)
 - Créer une instance du modèle dans la base de données (créer un nouvel enregistrement)
 - Vérifiez que la base de données est dans l'état prévu (vérifier tous les contenus de bases de données)
- *Read* (*Lire*)
 - Récupérer une instance du modèle à partir de la base de données (lecture d'un enregistrement)
 - Vérifiez que les données retournées correspondent au résultat attendu
- *Update* (*Mise à jour*)
 - Mise à jour une instance existante de notre modèle dans la base de données (mise à jour un enregistrement)
 - Vérifiez que la base de données est dans l'état prévu (vérifier tous les contenus de bases de données)
- *Delete* (*Supprimer*)
 - Supprimer une instance existante de notre modèle de la base de données (supprimer un enregistrement)
 - Vérifiez que la base de données est dans l'état prévu (vérifier tous les contenus de bases de données)

Un modèle d'objet de domaine (par ex. Elève ou Enseignant) encapsule une certaine logique définie et nécessite généralement quelques informations pour être stockées dans des propriétés (variables). Souvent, certaines de ces informations sont stockées dans la couche de persistance d'un schéma qui diffère de la manière dont l'objet est représenté.

Il peut être stocké dans une ou plusieurs tables ou des documents ou des fichiers, selon le logiciel et les relations. Un objet mappeur de données (Data Mapping) encapsule la logique de stockage de ces informations, c'est donc cet objet que nous voulons tester.

Pour cet exemple, nous testons les intégrités des données et non pas la logique métier et fonctionnelles. Je ne teste pas de valeurs spécifiques comme les chaînes vides, les valeurs NULL, la longueur de mot de passe, validité de l'email ou d'autres éléments qui appartiennent au logique métier implémenté dans les modèles de domaine. Ni les tests de fragmentation des données, la réplication ou d'autres caractéristiques spécifiques de stockage car ils appartiennent au modèle de base de données.

En PHP, les tests seront écrits en utilisant le Framework Zend de tests unitaires, un cadre qui permettra à notre application de tester des unités isolées de fonctionnalités telles qu'une seule classe ou une méthode. Nous utiliserons la librairie PHPUnit, qui offre l'avantage d'être entièrement compatible avec PHP.

XIV.8 – Difficultés rencontrés de mise en œuvre

Les difficultés rencontrées que j'ai rencontrées tout au long du stage ont été de deux ordres. Des difficultés techniques et des difficultés d'organisation.

XIV.8.1 – Difficultés d'organisation

Dans ce projet, j'ai rencontré des difficultés d'organisation. En effet, comme je l'ai déjà dit, je devais travailler seul, mes tâches étaient principalement secondaires et autonome par rapport aux autres tâches. Il y'a eu de nombreuses tâches secondaires, imprévus et imprévisibles. Je n'ai pas réussi à travailler régulièrement pour mon stage. Par ailleurs, je ne l'ai pas indiqué sur le planning mais plusieurs évènements sont survenus au cours duquel il m'a fallu m'adapter et qui a engendré du retard sur le développement de l'application parmi lesquels on peut citer :

- Le déménagement et l'emménagement de la société DMS à Saint –Martin du Touch, les stagiaires et moi- même avons participé aux rangements des matériels dans les cartons et ainsi qu'à l'installation dans les nouveaux locaux. Cela a duré environ 1 semaine.
- A plusieurs reprises, nous avons des soucis de connexion dans les locaux du Lycée ce qui m'a valu de me déplacer systématiquement à chaque reprise pour relancer le serveur.
- Nous avons été privés de connexion internet pendant 9 jours après avoir déménagé ce qui nous a empêchés de pouvoir nous connecter sur le serveur.
- Les stagiaires étaient parfois sollicités pour effectuer d'autres types de tâches que le développement informatique.

D'autres difficultés d'organisation sont liées au cadre du déroulement du stage :

L'une d'elles rencontrée lors du stage est due à l'indisponibilité de mon tuteur de stage. Cette indisponibilité s'analyse sous deux angles :

- D'une part une indisponibilité résultant du hasard du calendrier, mon stage correspondant à une période difficile pour l'entreprise, la plupart des collaborateurs de l'entreprise étaient indisponibles pour m'aider dans mes travaux ainsi que mon tuteur.
- D'autre part l'indisponibilité est due à une pression du travail des dossiers importants à traiter en priorité comme l'organisation et le changement de directeur. Le temps que mon tuteur avait à m'accorder relevait plus de la politesse que du temps nécessaire à l'avancement du projet. Ce fut une expérience éprouvante et travailler dans ces conditions était difficile.

Une autre difficulté auquel j'ai été confronté est que j'avais eu pour cadre des stagiaires qui eux-mêmes sont en train d'apprendre et donc n'ayant pas tous les rudiments nécessaires.

Le dernier problème auquel j'ai été confronté est celui du milieu de l'enseignement au sein d'un établissement scolaire. En effet, les enseignants se sont parfois montrés réticents dans le développement du projet, n'approuvant guère ce genre de méthodes de travail des entreprises et du privé.

XIV.8.2 – Difficultés techniques

La réalisation de 1 à Z d'un logiciel n'est parfois pas très aisée. J'ai été confronté à plusieurs problèmes techniques d'ordre informatique et industriel. Pour commencer, concernant les matériels informatiques, DMS ne disposait pas d'un matériel informatique convenable à mettre à ma disposition. Ce qui a été un frein dans un premier temps puisque je ne pouvais commencer le développement. C'est donc le Lycée Déodat de Séverac qui m'a généreusement fourni un poste informatique convenable pour effectuer la partie du développement informatique ainsi que un serveur web.

Par la suite, j'ai été confrontés à un domaine qui m'était totalement inconnu, à savoir celui de l'industrie, il m'a fallu dans un premier temps les notions et techniques employés dans l'informatique industriels. Prise de connaissances sur les automates et les systèmes par ex., apprentissage des protocoles de communication. Cette prise de connaissance s'est avéré nécessaire mais difficile au démarrage du projet.

La mise d'un place d'un tel réseau au sein d'un établissement qui sont sur protégés et sécurisé s'est montré très difficile. J'ai été contraint à plusieurs reprises d'intervenir sur la configuration interne du réseau par l'installation de baies et de routeurs. Par la suite, le temps d'attente pour l'ouverture d'un tunnel qui relie directement le réseau interne de la zone Electrotechnique au réseau extérieur s'est montré très long à mettre en place par la mission TICE.

C'est ainsi que durant le développement de l'application plusieurs points ont été relevés :

- Problème de communication avec les systèmes, le réseau du Lycée ne dispose que de peu de débit pour l'ensemble de l'établissement à savoir 2MO/sec, il est passé à 4Mo par la suite. C'est donc insuffisant pour charger des applets qui présentent des interfaces riches et lourdes.
- Temps de réponse du système. Les automates n'ont pas été conçus pour être sollicité de manière importante dès que la demande devient intense, on sent donc le temps augmenté en fonction de la demande.

On peut évoquer des difficultés de conception. Il m'a fallu imaginer une interface la plus claire possible, essayé d'être cohérent dans l'ensemble des menus et interfaces. Mais pour autant je n'avais pas eu de retours de la part des utilisateurs d'autant plus que certaines fonctionnalités que je devais interfacer étaient plutôt complexes. Les réunions que j'avais eues ne concernaient pas le développement des IHM.

Quand aux difficultés de programmation, elles sont venues principalement de l'implémentation de librairie intéressante mais complexe à mettre en œuvre. Là, j'ai perdu un temps précieux. Une autre difficulté a été de transmettre des données entre serveurs et automates. A l'entreprise, personne n'avait l'expérience de DLL gérant des fonctions si complexes. Finalement, en tâtonnant, j'ai pu trouver une solution déjà présente mais qui se révélera insuffisante pour la suite.

Il est très difficile, dans les faits, d'évaluer le nombre d'heures effectives consacrées au développement de l'application. Toujours est-il que, quelles qu'en soient les raisons plus ou moins valables, il reste encore une bonne partie du travail à faire. Je ne saurais évaluer combien de temps prendra encore la fin du développement mais il m'a été demandé d'anticiper ce développement en transmettant mes connaissances à un autre stagiaire Cnam.

CONCLUSION

Ce projet aura été l'occasion pour moi d'appréhender la gestion de projet et ses difficultés. Il a fallu gérer le planning, le travail des développeurs (stagiaires) et les séances de travail avec les enseignants, tout en développant une partie de la solution.

Ce jonglage entre le rôle de chef de projet et de développeur a été particulièrement éprouvant, car il fallut à la fois avancer sur les tâches que l'on doit réaliser, et s'assurer que l'ensemble du projet se déroule correctement.

Une autre difficulté, amenée à mon sens par le principe même de fonctionnement de la méthode Agile Scrum, est la gestion des demandes des utilisateurs finaux. En effet, cette méthode a l'énorme avantage de mettre ceux-ci au cœur du développement du logiciel. Ce qui permet de réaliser une solution vraiment proche de ses attentes grâce au travail réalisé autour du prototype. On se rend immédiatement compte des incompréhensions qu'il y'aurait pu avoir durant les phases précédentes. Mais durant le projet, beaucoup n'étaient pas disposés à me faire un retour sur le travail que je réalisais.

A ce jour la solution est en toujours cours de déploiement dans le Lycée. Le choix de déployer la solution au fur et mesure a été pris, pour assurer une introduction efficace aux enseignants de l'établissement. En effet le changement, avec l'arrivée d'un nouvel outil peut en inquiéter certains. Les personnes qui ont commencé à travailler dans le projet montrent un réel enthousiasme. Elles font également ressortir certaines fonctionnalités qui ne sont pas présente mais qui avaient été discutées lors de la phase de construction.

Elles n'avaient pas pu être intégrées au périmètre car elles auraient trop pénalisé le projet en termes de coûts et de délais. Ces remarques confortent DMS Education dans la perspective de les intégrer le plus rapidement possible dans une nouvelle version de la solution.

Ce stage se termine donc sur une réalisation inachevée. Il faudra tout de même que je finalise DIWEB afin de pouvoir le présenter prochainement lors de ma soutenance du mémoire. C'est sans doute le point le plus frustrant. Je constate tout de même que d'autres tâches que j'ai dû réaliser, comme l'installation du réseau informatique et qui ne rentre pas directement dans le cadre du diplôme, ont été réalisées avec succès. C'est une forme de consolation.

J'ose enfin espérer que les difficultés d'organisation me seront profitables pour l'avenir. Je me rends compte à quel point il peut être difficile de dialoguer avec des personnes, qui n'ont aucun cursus informatique, sur des points techniques avancées. La réalisation de ce projet m'a permis de vérifier que la présentation de l'idée est presque aussi importante que l'idée elle-même. Avant de développer une solution, les utilisateurs finaux ainsi que leurs besoins doivent être connus et bien identifiés. A l'inverse des projets en entreprise, où le besoin est exprimé par le client, je devais comprendre le besoin moi-même que souhaitent les utilisateurs soit les enseignants. Dans mes prochaines activités, je ne manquerais pas de mettre à profit cette compréhension des attentes de l'utilisateur final.

Enfin, ce mémoire est pour moi un faire-valoir de mes compétences en architecture logicielle et en informatique en général dans un processus de spécification de systèmes.

TABLE DES ANNEXES

Annexe 1 : Images du Lycée Déodat de Séverac

Annexe 2 : Systèmes communicants présents au sein de la zone Electrotechnique

Annexe 3 : Configuration du module ETZ510

Annexe 4 : Exemple de configuration web avec le système industriel Tapiris

Annexe 5 : Le protocole Modbus

Annexe 6 : Ressource logiciel industriels

Annexe 7 : IHM du système industriel SPEN

Annexe 8 : Applet de connexion automate

Annexe 9 : Méthodologie de développement

Annexe 1 : Images du Lycée Déodat de Séverac



Photo A1.1.1 – Logo du Lycée Déodat de Séverac



Photo A1.1.2 – Photo de l'enceinte du Lycée Déodat de Séverac



Photo A1.1.3 – Un étudiant effectuant un câblage lors d'un TP

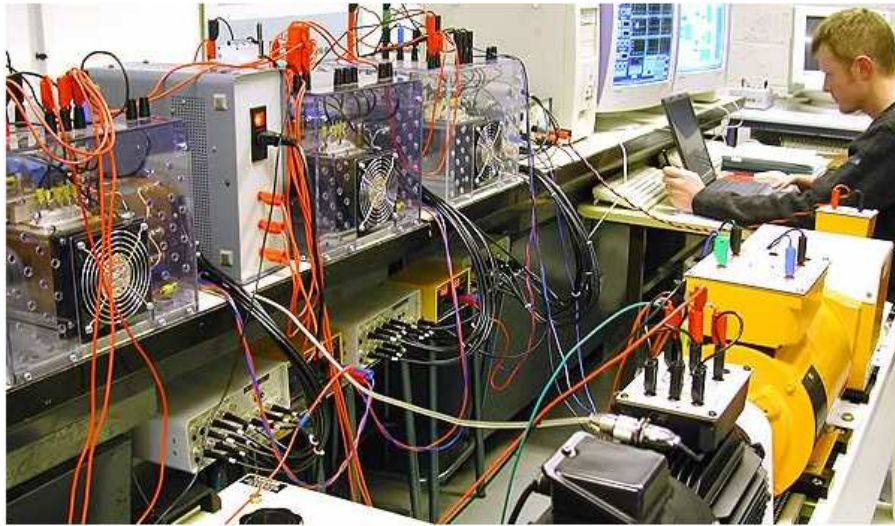


Photo A1.1.4 – Un étudiant effectuant des tests de puissance sur un moteur asynchrone

A1.2 – Divers équipements installés dans la zone Electrotechnique



Photo A1.2.1 – Une caméra IP pivotable accessible depuis Internet



Photo A1.2.2 – Un système industriel communicable avec un PC



Photo A1.2.3 – Une vue d'ensemble de l'atelier



Photo A1.2.4 – Une vue sur les bureaux des étudiants

Annexe 2 : Systèmes communicants présents au sein de la zone Electrotechnique

A2.1 – Tapiris

Fonctionnement générale :

L'équipement pédagogique Tapiris est module de tri postal. Les colis sont représentés par des blocs de mousse équipés d'une étiquette qui comporte un code barre. Ces colis sont ensuite aiguillés vers trois destinations possibles : deux destinations perpendiculaires au convoyeur et une à la fin de celui-ci. Cet équipement est constitué de deux sous-ensembles.

Il est basé sur le concept AS-i et met en œuvre des capteurs et des actionneurs « unifié » c'est-à-dire que chaque élément possède une adresse lui permettant la communication. Cet ensemble permet de comparer et d'apprécier ce concept par rapport à des solutions de câblage traditionnel pour le contrôle d'une application industrielle.



A2.2 – Centrale de pompage piscine

Fonctionnement générale :

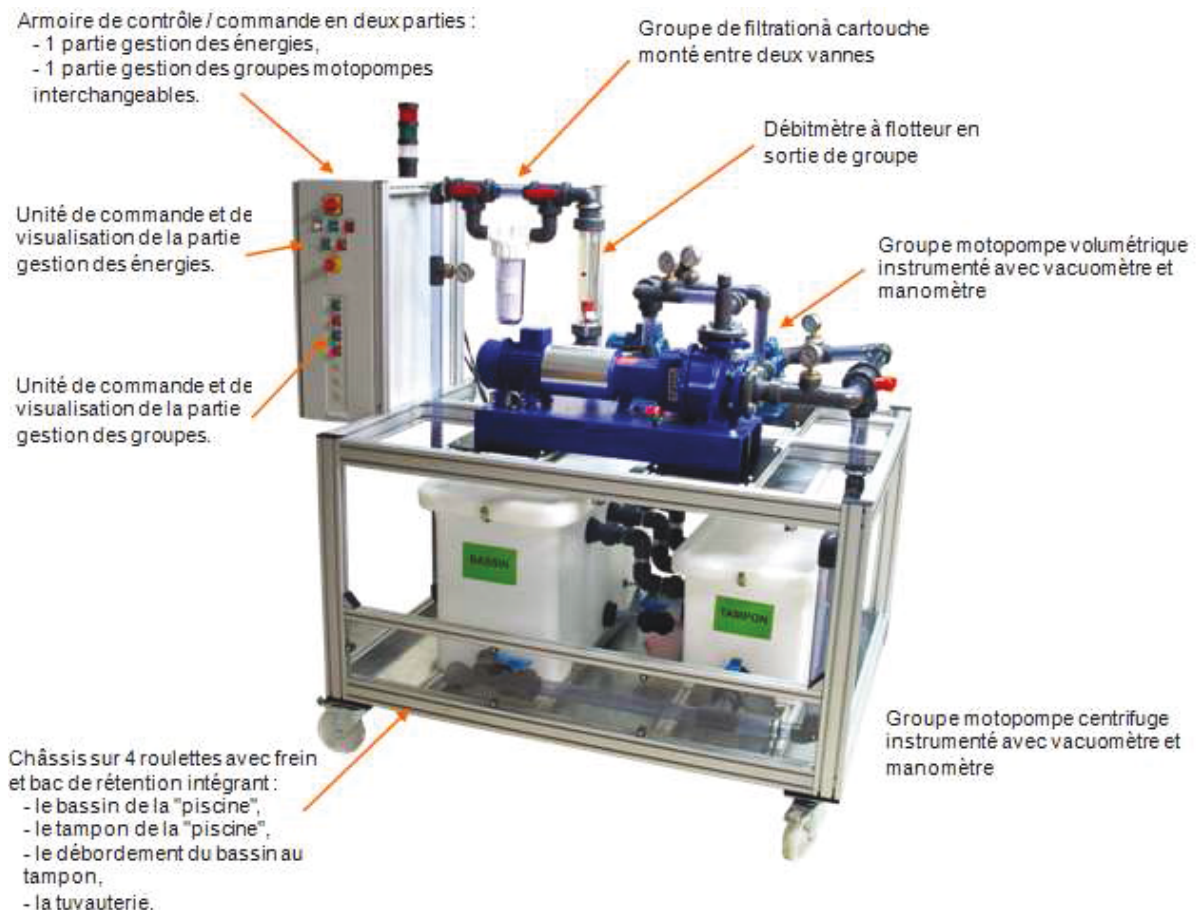
La centrale de pompage piscine est un système didactisé issu d'une véritable piscine municipale à débordement. Ce système met en œuvre l'ensemble des équipements et composants qui permettent d'assurer la circulation de l'eau et de concourir à sa qualité, appelé hydraulité d'une piscine.

Parmi les grands types d'hydraulité rencontrés dans les piscines, le système didactique permet de répondre à l'hydraulité mixte et l'hydraulité inversée.

Le système didactique permet :

- De mener des études fonctionnelles et structurelles,
- D'analyser les équipements de puissance et commande,
- De préparer, de localiser et de diagnostiquer les pannes,
- De mener des opérations de maintenance corrective sur des pompes : dépose, repose, échange, réglage et de réparation par démontage remontage,
- De mener des opérations sur maintenance préventive et améliorative

Le système didactique est composé d'une partie opérative originale puisqu'elle permet de recevoir indépendamment ou simultanément un groupe motopompe centrifuge (à étanchéité d'arbre par presse-étoupe ou par garniture mécanique) et/ou un groupe motopompe volumétrique. La figure ci-dessous présente les différents constituants du système.



A2.3 – Micro-centrale SPEN

Fonctionnement générale :

Le principe de fonctionnement d'une petite centrale hydroélectrique consiste à transformer l'énergie potentielle d'une chute d'eau en énergie mécanique grâce à une turbine, puis en énergie électrique au moyen d'une génératrice. La puissance installée de la centrale est fonction du débit d'eau turbine et de la hauteur de chute.



A2.4 – Surpressic

Fonctionnement générale :

Ce système se comporte comme une station de pompage et de distribution de l'eau. Il fonctionne sur le même principe que les stations d'épuration, d'irrigation ou d'alimentation en eau des réseaux urbains. Ce système permet de traiter les thèmes d'applications tels que :

- Commande de moteurs,
- Détection de défauts,
- Régulation de niveaux,
- Choix mode de fonctionnement,
- Sécurités.



A2.5 – TGBT

Fonctionnement générale :

Le système TGBT est un tableau général Basse Tension d'une entreprise équipée d'un circuit de secours alimenté par un groupe électrogène. La supervision et l'automate WAGO présent dans le TGBT gèrent la commutation entre la source normale et la source secours ainsi que le délestage des différents départs motorisés.

Une centrale de mesure avec afficheur relève en permanence les informations relatives à la consommation des équipements alimentés par le TGBT.



A2.6 – GBT

Fonctionnement générale :

L'automate GTB a pour vocation de gérer un lot technique donné comme le chauffage, la climatisation, l'éclairage ou encore la gestion des points d'accès comme nous le faisons dans notre projet.



A2.7 – L'irio

Fonctionnement générale :

L'iRio est un automate de télégestion industriel serveur web qui ouvre notre réseau pour le contrôle et la commande de parties opératives quelconques. Dans la zone, celui-ci est utilisé pour la gestion de capteurs tels que la sonde de température.



A2.8 – Bain régulateur de température

Fonctionnement générale :

Le bain régulateur étant un système qui demande un temps de mesures très long (chauffe d'un volume d'eau),

A2.9 – Equipement photovoltaïque

Fonctionnement générale :

Il s'agit d'une centrale de production située sur le toit de l'établissement scolaire, elle est composé de 9 panneaux. Le tout est connecté dans l'atelier des terminales.

A2.10 – Superviseur Redlion G308C

Fonctionnement générale :

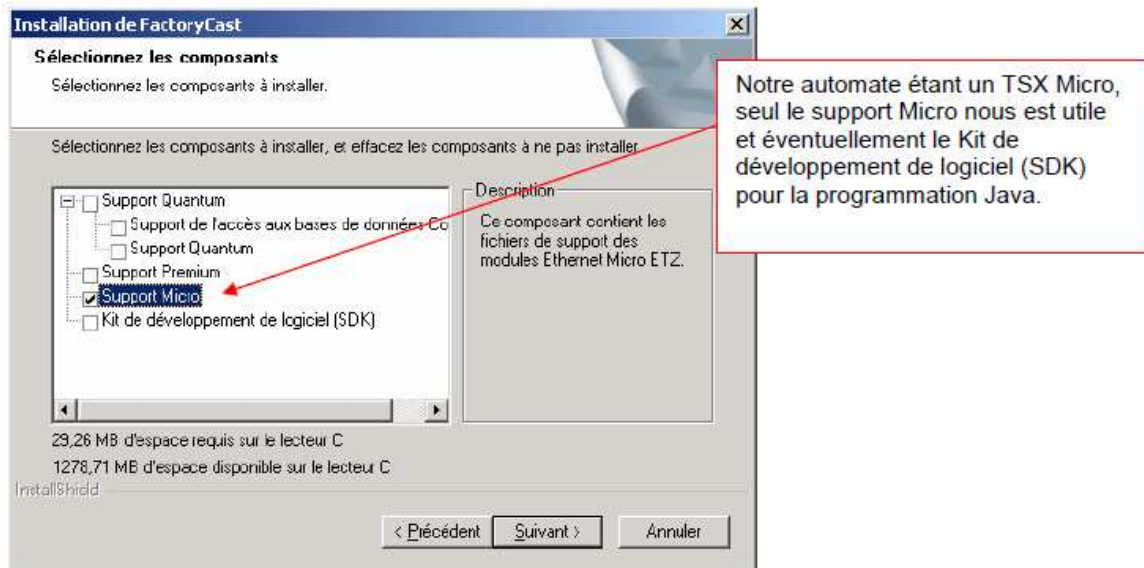
Cette console permet de superviser certains des systèmes. Elle est dotée d'un écran LCD couleur tactile.



Annexe 3 : Configuration du module ETZ510

A3.1 – Installation du logiciel de configuration de Factory Cast

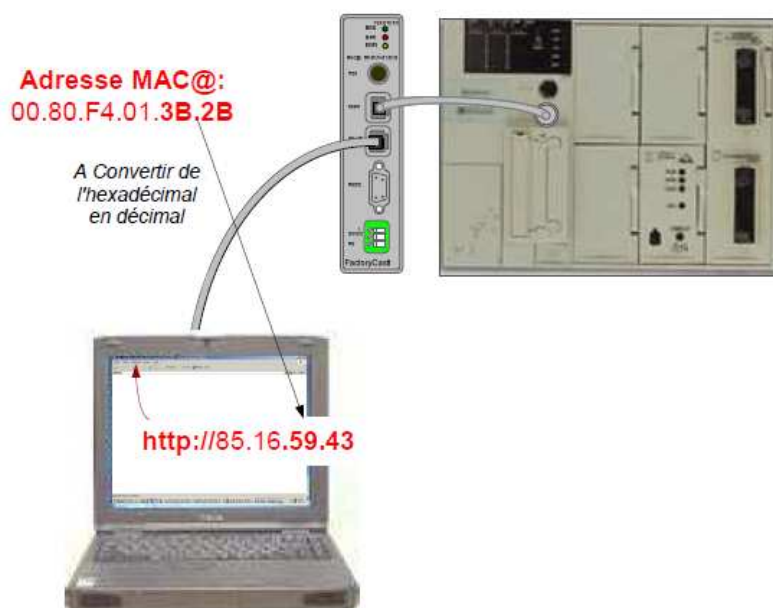
A partir du CD-Rom « Factorycast For Quantum, Premium and Micro PLCs, exécutez le fichier « \DISK1\setup.exe ». Après le choix de la langue et de l'emplacement, choisissez les composants à installer :



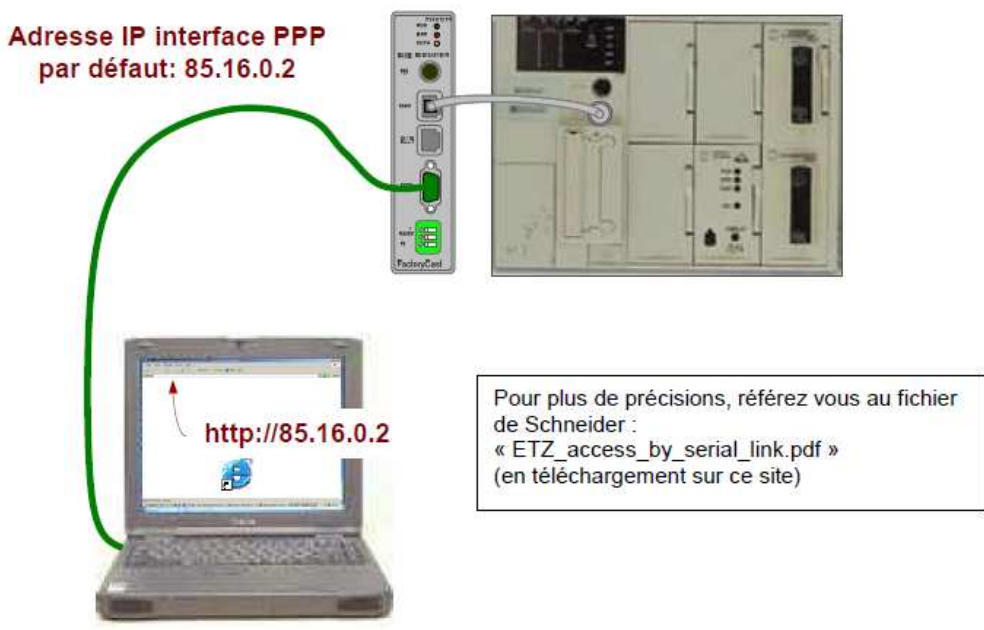
A3.2 – Connaître l'adresse IP du module ETZ 510

Par liaison Ethernet

A la livraison, le coupleur Ethernet a pour adresse IP : 85.16.XX.YY avec XX et YY les deux derniers octets de son adresse MAC (notée en face avant).



A3.3 – Configuration par liaison série RS 232

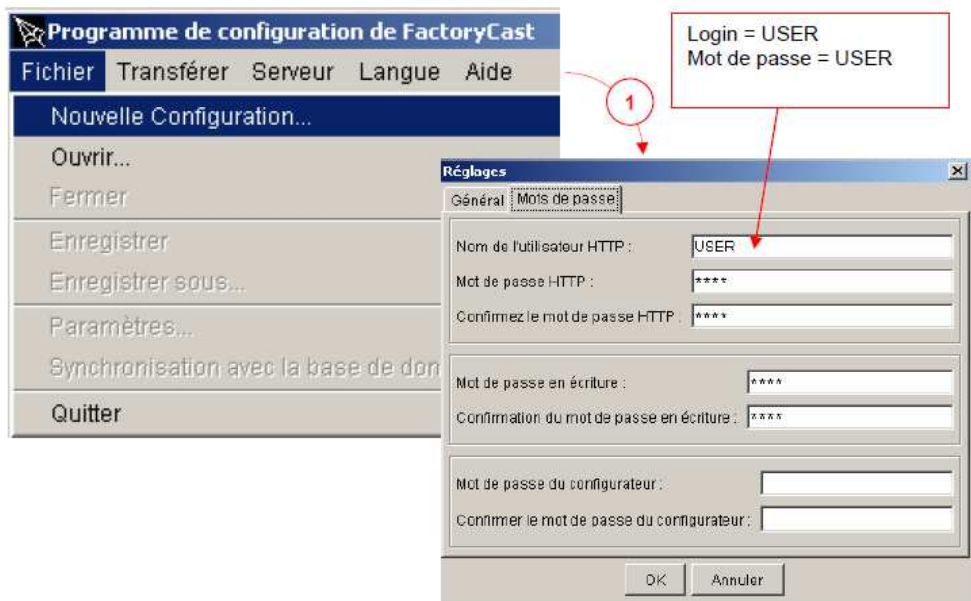


A3.4 – Installer une nouvelle configuration du serveur web

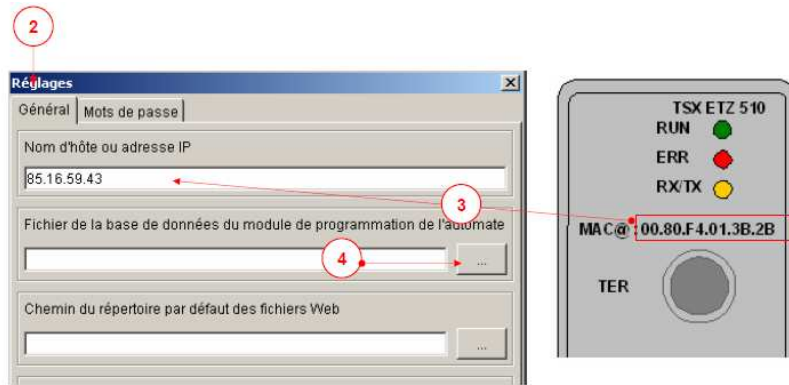
Lancez le programme de configuration FactoryCast :

Démarrer > Programmes > Schneider Electric > FactoryCast > Programme de Configuration

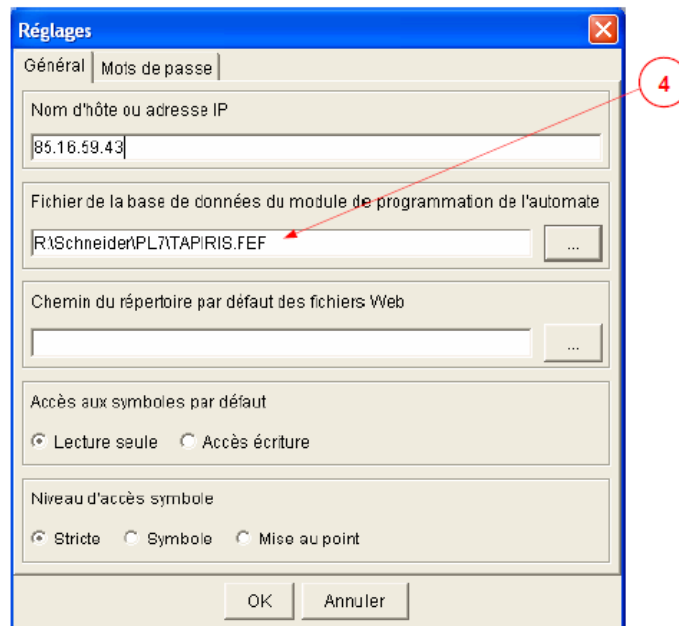
Cliquez sur Fichier > Nouvelle Configuration, puis, dans l'onglet « mot de passe » entrez les identifiants de l'utilisateur autorisé à modifier la configuration du serveur web.



Précisez l'adresse IP de l'automate concerné



Le fichier de la base de données du module de programmation de l'automate (*.FEF) est fourni par Schneider car il a été créé avec la version PRO de PL7. Cette base de données contient la liaison entre les variables automates et leur symbole.

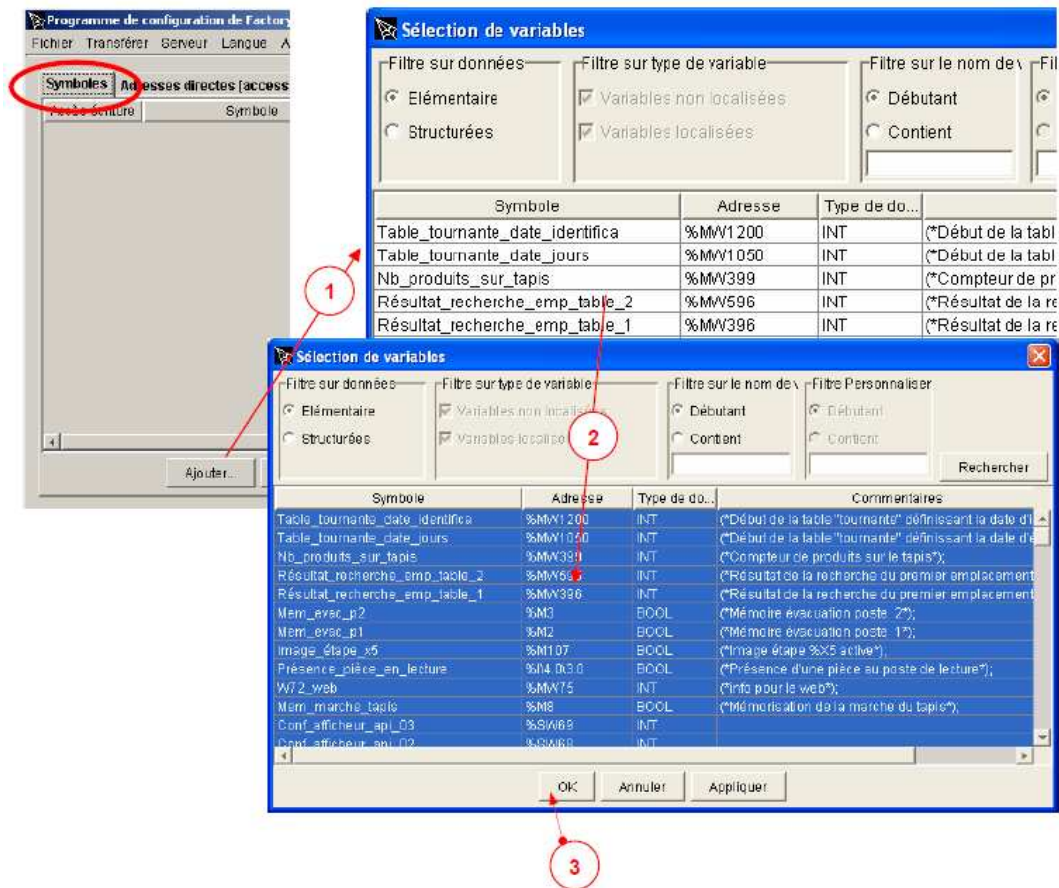


A3.5 – Construire et transférer un espace de noms

Un espace nom est une base de données Web qui contient des symboles et/ou des adresses avec leur droit d'accès à partir du web (lecture seule ou écriture). Cet espace nom trouve ces informations à partir du fichier ***.FEF**.

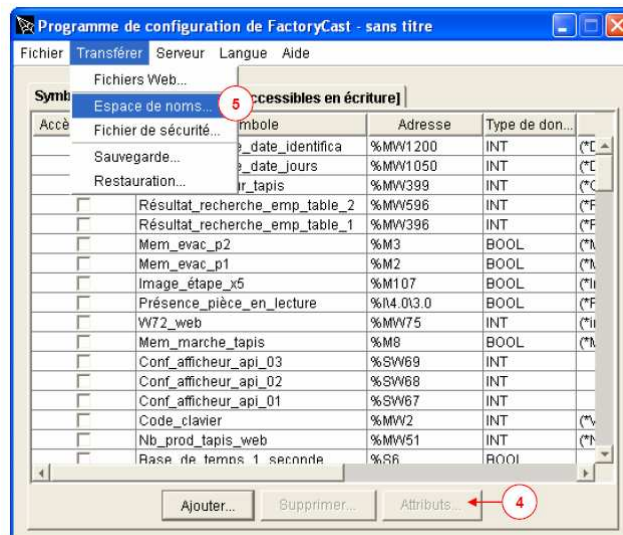
Si vous souhaitez afficher des symboles dans l'éditeur de données, l'éditeur graphique ou le viewer d'alarmes, vous devez les inclure dans l'espace nom. L'éditeur de données ne permet pas l'affichage des symboles qui ne figurent pas dans l'espace nom.

Pour importer des symboles d'une base de données Concept, PL7 ou Unity Pro dans votre espace nom, suivez les étapes suivantes :



*

Étapes 1 à 3 : Sélectionnez tous les symboles et cliquez sur OK. Tous les symboles héritent du droit « lecture seule ». Pour autoriser l'écriture (étape 4), sélectionnez un symbole et cliquez sur « Attributs ».

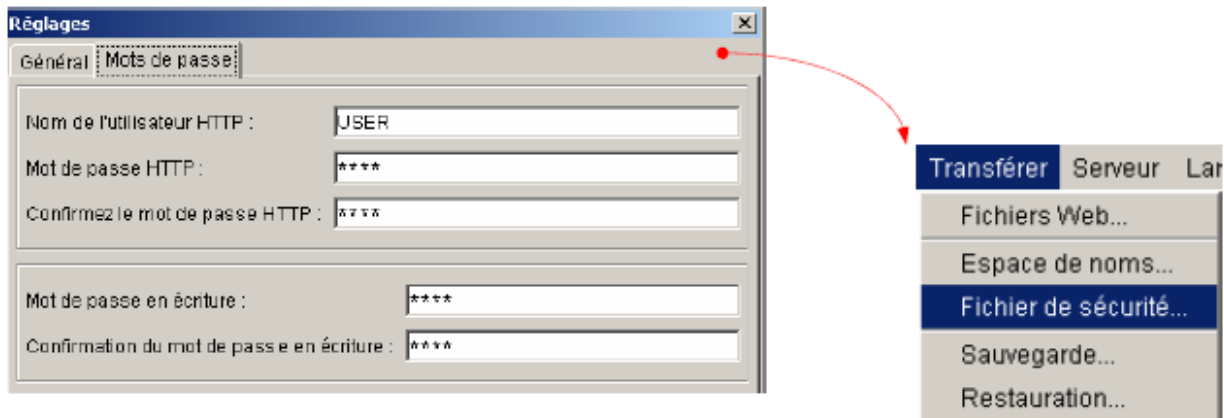


Pour envoyer cet espace de noms sur le serveur web de l'automate, cliquez sur Transférer > Espace de noms (étape 5) et donnez l'adresse IP de l'automate (étape 6).



A3.5 – Transférer le fichier de sécurité

Pour enregistrer un nouveau login / mot de passe sur le serveur web :



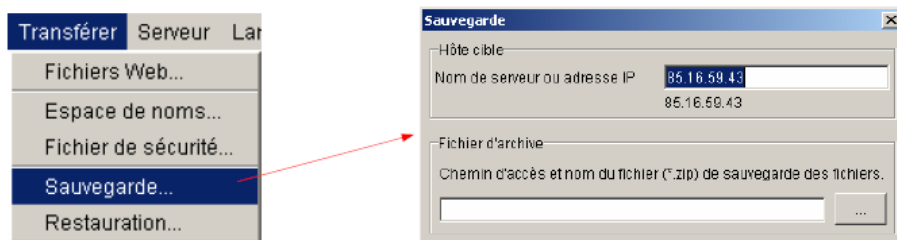
A3.6 – Sauvegarde de l'espace de noms et de la sécurité

Pour générer un fichier *.cfg de sauvegarde de l'espace de noms et de la sécurité, cliquez sur :

Fichier > Enregistrer sous et choisissez le nom du fichier de sauvegarde.

A3.7 – Sauvegarde de toute la configuration du serveur web embarqué

Vous pouvez sauvegarder toutes les informations de configuration du serveur web de l'automate dans un fichier *.zip. Cliquez sur Transférer > Sauvegarde et donnez l'adresse IP de l'automate ainsi que l'emplacement du fichier de sauvegarde.

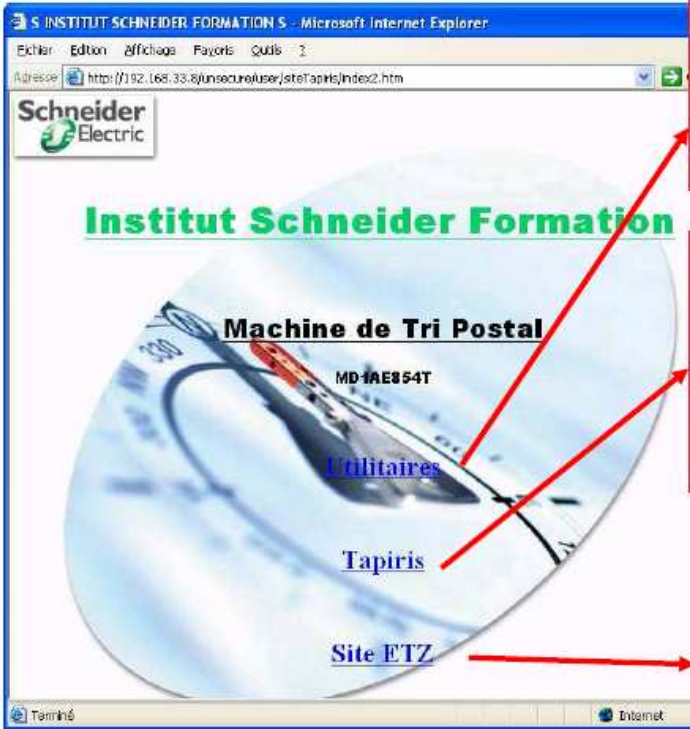


Annexe 4 : Exemple de configuration web avec le système industriel Tapiris

A4.1 – Présentation

Le site web hébergé sur le coupleur ETZ permet de visualiser et configure le coupleur ETZ. De plus, il permet de paramétrer la maquette Tapiris et de suivre son fonctionnement. Il est possible d'ajouter des pages web personnelles sur le serveur (applet pour un synoptique de fonctionnement en liaison avec le Tapiris, flux vidéo pour la visualisation à distance de la production, etc ...).

A4.2 – Page d'accueil



Utilitaires : Accès direct à certaines pages de configuration de « Site ETZ »

- Diagnostic systeme
- Editeur de données
- Editeur graphique
- Configuration en ligne

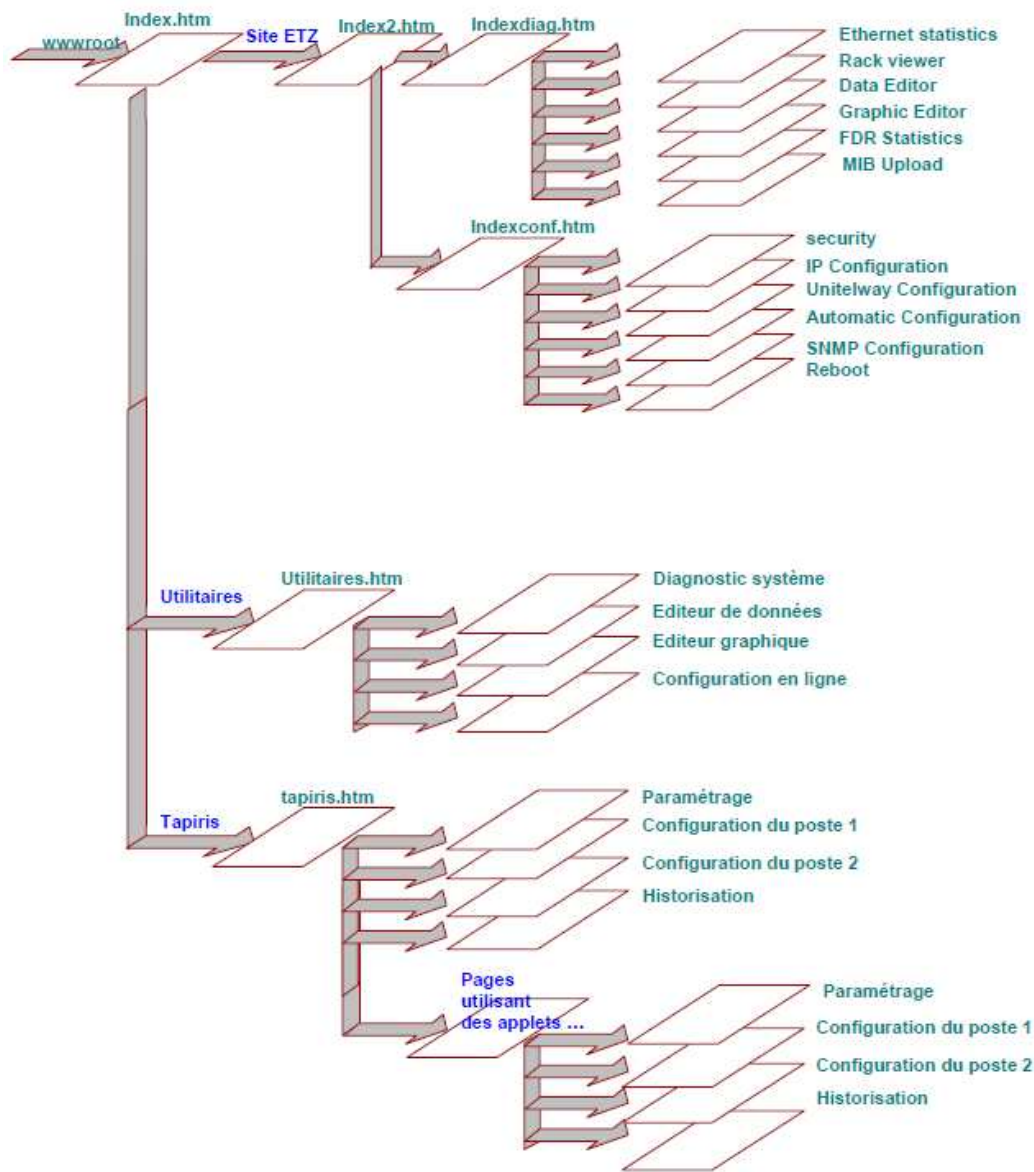
Tapiris : Paramétrage et visualisation de la maquette Tapiris

- Paramétrage
- Configuration du poste 1
- Configuration du poste 2
- Historisation
- Pages utilisant des applets Java utilisateur

Site ETZ : Pages web standard de gestion du coupleur ETZ et de l'automate TSX Micro

- Configuration
- Diagnostic

A4.3 – Arborescence des IHM du Tapiris

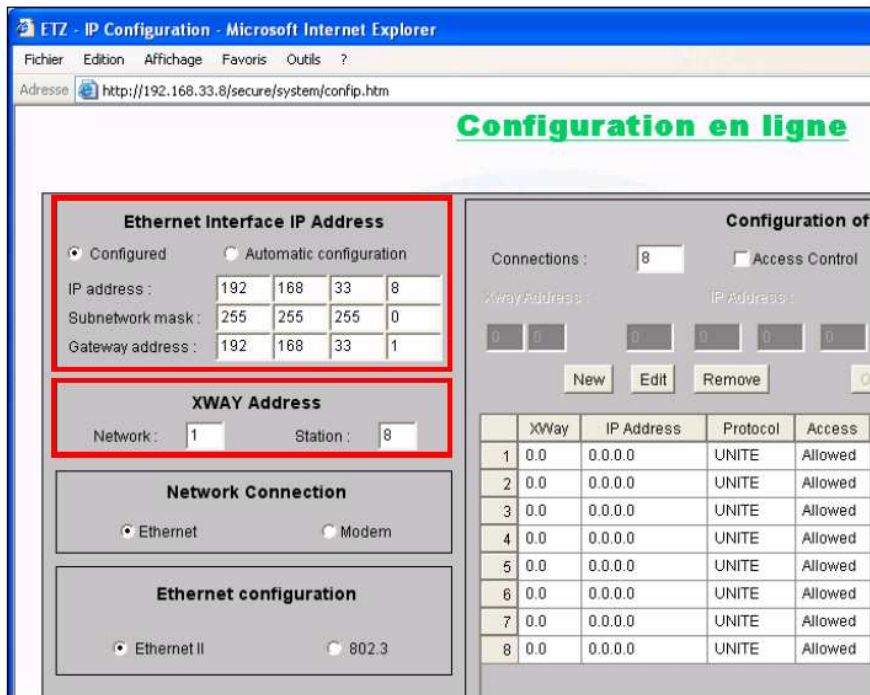


A4.4 – Configuration de l'adresse IP de l'automate

Pour changer les adresses IP et XIP afin d'intégrer l'automate dans notre réseau, il faut aller sur la page suivante :

Accueil > Utilitaires > Configuration en ligne Ou Accueil > Site ETZ > Configuration > Ip Configuration

Après authentification (USER / USER par défaut), on peut modifier les champs Ethernet et XWAY pour s'adapter à notre réseau.



A4.5 – Editeur de données

L'éditeur de données est une applet Java qui vous permet de visualiser dynamiquement des données automatées. Les deux éditeurs sont mis à jour de façon dynamique avec les données d'exploitation de l'automate.

Pour lancer l'éditeur de données : *Accueil > Utilitaires > Editeur de données*



L'éditeur de données permet d'afficher les données de la façon suivante :

- **Nom de variable** : Les noms des symboles (variables) PL7. Les seuls symboles (variables) pouvant être utilisés dans l'éditeur de données sont ceux qui figurent dans l'espace nom du serveur intégré. Si vous essayez de saisir un symbole (variable) absent de l'espace nom, le message « variable non trouvée » s'affiche.
- **Adresse** : la colonne Adresse contient les adresses directes ainsi que les adresses des symboles (variables) PL7. Vous pouvez afficher n'importe quelle adresse directe en saisissant sa référence dans ce champ. Vous n'avez pas besoin de l'inclure dans l'espace nom.
- **Type de données** : Le champ Type de données contient le type de données du symbole (variable) ou de l'adresse directe. Les types de données du symbole (variable) s'affichent automatiquement quand le symbole (variable) est localisé. Les types de données des adresses directes doivent être configurés par l'utilisateur dans une liste déroulante.
- **Valeur** : La colonne Valeur sera remplie avec la valeur du symbole (variable) ou l'adresse directe. Le champ est mis à jour de façon dynamique.*
- **Format** : Le champ Format contient le type de format de l'affichage de la valeur du symbole (variable) ou de l'adresse directe.
- **Etat** : La colonne Etat contient les messages d'état des communications avec le symbole (variable) ou l'adresse directe. Si les communications sont normales, le message d'état est « OK ». S'il y'a un problème de communication avec le symbole (variable) ou l'adresse directe, la colonne **Etat** affiche un message d'erreur décrivant le problème.

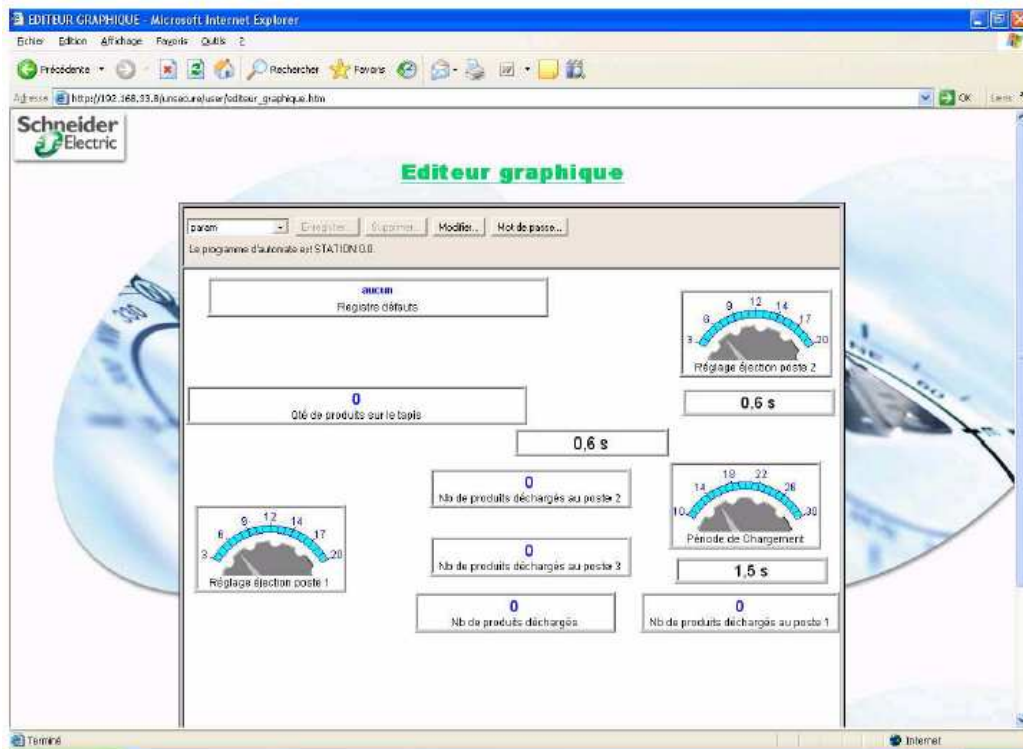
A4.6 – Editeur graphique

L'éditeur graphique est une applet Java qui vous permet de créer des affichages graphiques dynamiques en utilisant un ensemble d'objets graphiques prédéfinis.

L'éditeur graphique est à la fois :

- L'éditeur qui permet de créer et modifier les affichages et l'environnement d'exécution qui permet d'obtenir les affichages dynamiques des données d'exécution de l'automate.

Pour lancer l'éditeur graphique : *Accueil > Utilitaires > Editeur graphique*

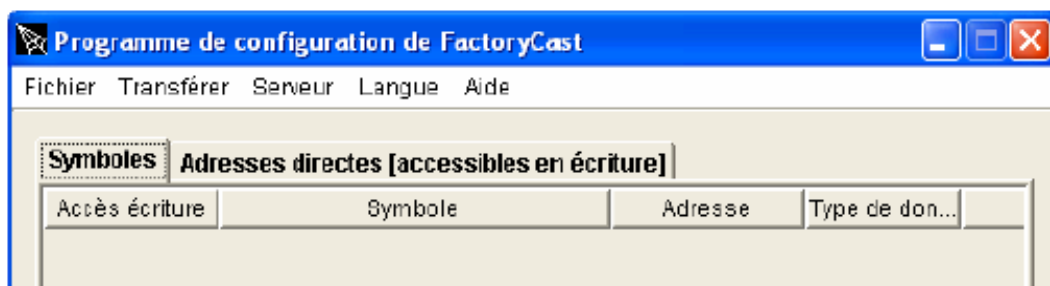


A4.7 – Ajout de nouvelles pages web sur le serveur

Le coupleur ETZ 510 contient un serveur FTP qui permet de transférer des fichiers HTML et JAVA (applets) sur le serveur web embarqué. Le client FTP est accessible via le logiciel de configuration Factory Cast.

Lancez le programme de configuration de FactoryCast via : *Démarrer > Tous les programmes > Schneider Electric > FactoryCast > Programme de configuration de FactoryCast*

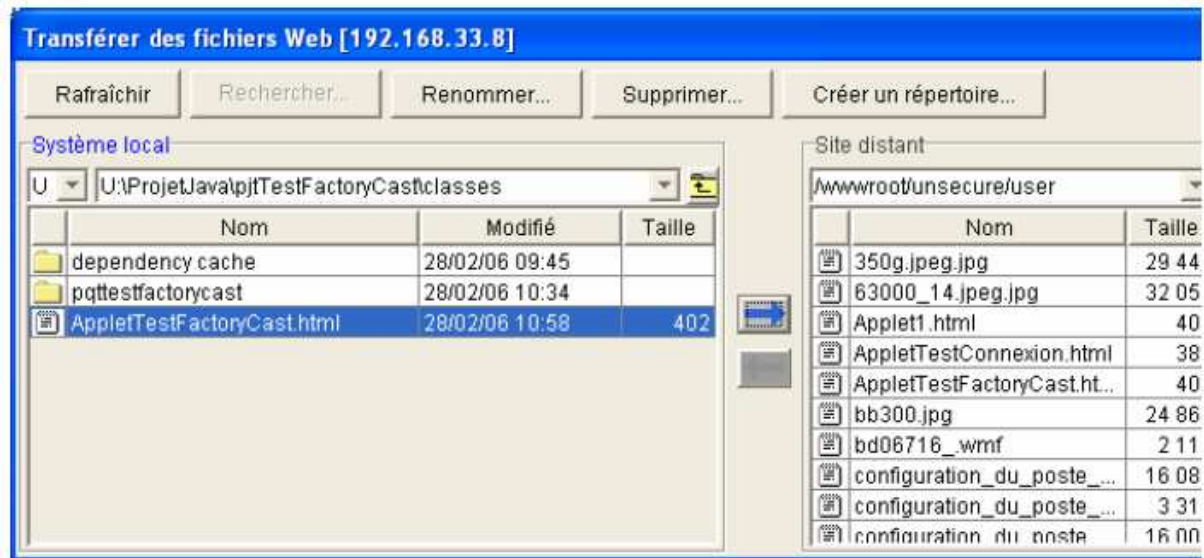
La fenêtre suivante s'ouvre :



Cliquez ensuite sur Transférer > Fichiers Web ...



Renseignez ensuite le champ « Nom de serveur ou adresse IP » avec l'adresse IP de l'automate. Cliquez sur OK.



Annexe 5 : Le protocole Modbus

A5.1 – Principes de base

Modbus est un protocole utilisé pour des réseaux d'automates programmables du constructeur Schneider-Télemécanique pour sa gamme des Modicon. Il fonctionne sur le mode maître/esclave. Il est constitué de trames contenant l'adresse de l'automate concerné, la fonction à traiter (écriture, lecture), la donnée et le code de vérification d'erreur appelé contrôle de redondance cyclique sur 16 bits ou CRC16.

Les trames sont de 2 types :

- Mode RTU (Remote Terminal Unit) : les données sont sur 8 bits.
- Mode ASCII : les données sont sur 7 bits (les trames sont donc visible en hexadécimal et il faut deux caractères pour représenter un octet).

Le protocole Modbus (marque déposée par Modicon) est un protocole de dialogue basé sur une structure hiérarchisée entre un maître et plusieurs esclaves.

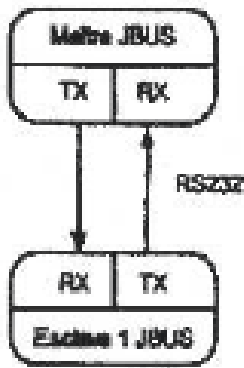
Couche physique, le protocole Modbus peut être implémenté :

- Sur une liaison série asynchrone de type RS-422 ou RS-485 ou TTY (boucle de courant), avec des débits et sur des distances variables ;
- Via TCP/IP sur Ethernet ; on parle alors de Modbus TCP/IP ;
- Via Modbus Plus. Modbus Plus est un réseau à passage de jetons à 1 MB/s, pouvant transporter les trames Modbus et d'autres services propre à ce réseau.

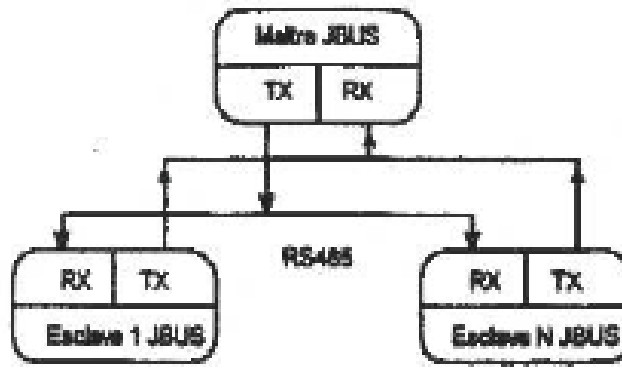
Une liaison multipoints de type RS-485 relie maître et esclaves via une paire différentielle qui permet un débit élevé (jusqu'à 10 méga-bits par seconde) sur une distance importante (jusqu'à 1200 m). Elle ne dispose que de 2 bornes qui alternativement passent les données dans un sens puis dans l'autre (half duplex).

MODBUS définit un réseau de communication logique ayant un seul **MAITRE** et un ou plusieurs **ESCLAVES**. Un réseau simple ou à des connexions multiples est possible. Les deux types de réseaux de communications sont illustrés sur le schéma ci-dessous

Liaison série simple



Liaison série à connexions multiples

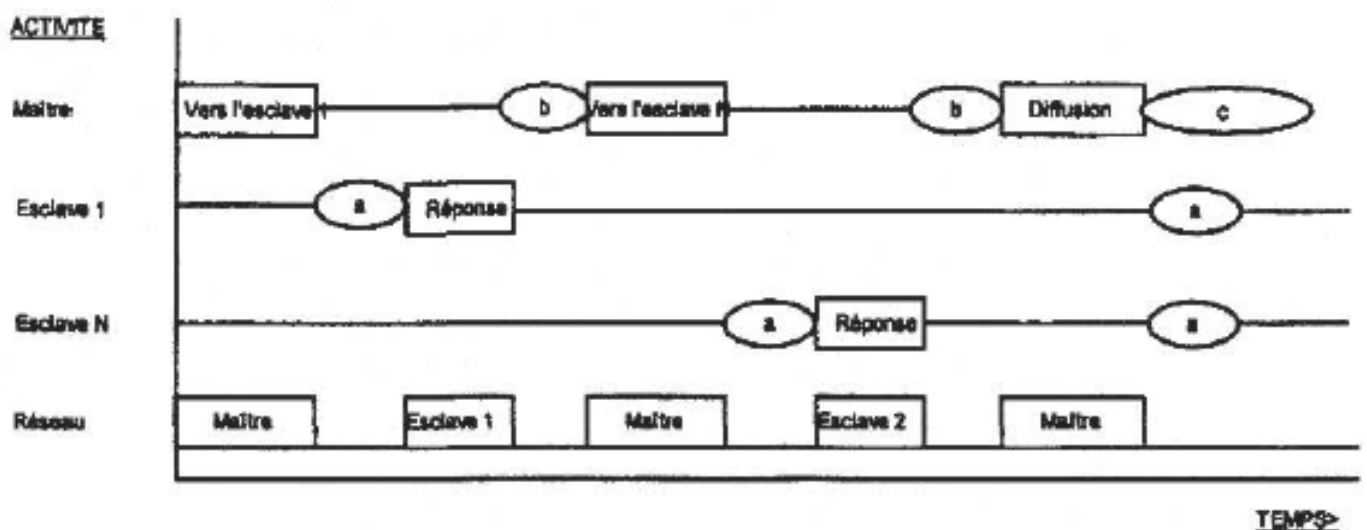


Principe de liaison Maître/Esclave

- Une transaction-type est composée d'une demande envoyée par le maître, suivie d'une réponse de l'esclave.
- Chaque esclave possède une adresse de périphérique unique sur 8 bits.
- L'envoi d'une communication avec l'adresse de périphérique '0' est une communication de diffusion qui envoie des informations à tous les périphériques du réseau. Chacun effectue l'action demandée mais sans envoyer de réponse.
- L'envoi d'une communication avec une adresse de périphérique unique déclenche une réponse du seul périphérique possédant cette adresse. Ce périphérique recherche les erreurs, effectue la tâche demandée puis répond en donnant sa propre adresse, les données et un total de contrôle.

A5.2 – Activité-type d'une ligne de transmission

Cette figure illustre la suite-type des événements sur une ligne de transmission Modbus.



Événements sur une ligne de transmission Modbus

Période « a » : temps de traitement nécessaire à l'esclave pour exécuter la commande et élaborer une réponse.

Période « b » : temps de traitement nécessaire au maître pour analyser la réponse de l'esclave et formuler la commande.

Période « c » : temps d'attente calculé par le maître pour que les esclaves exécutent l'opération.

Pour les fonctions d'échange de paramètres (écriture ou lecture de n bits ou de n mots), le temps d'attente est de 20 millisecondes par paramètre transmis (125 mots maxi).

A5.3 – Syntaxe de la trame du message

Le schéma qui suit montre l'ordre définissant la syntaxe de la trame du message utilisée par JBUS et MODBUS :

Début de la trame	Adresse du périphérique	Code fonction	Données accompagnant la fonction	Données de détection d'erreur	Fin de la trame
	1 octet	1 octet	n octets	2 octets	

Le début de la trame est une période d'inactivité égale à au moins 3.5 fois la durée de transmission d'un caractère unique. Par exemple, à 9600 bauds, un caractère comportant un bit de départ, un bit d'arrêt et 8 bits de données a besoin d'un début de trame de 3,5 millisecondes. Cette période est la fin de transmission implicite d'une transmission antérieure.

- L'adresse du périphérique est un seul octet (8bits) propre à chaque périphérique de réseau.
- Les codes fonction sont une instruction à un seul octet destinée à l'esclave et décrivant l'action à exécuter.
- Le segment de données d'un message dépend du code fonction et le nombre d'octets varie en conséquence.
- En règle générale, le segment de données contient une adresse de paramètres et le nombre de paramètres à lire ou écrire.
- Le contrôle de redondance cyclique est un code de détection d'erreur qui a une longueur de 2 octets.
- Le segment Fin de la transmission est une période d'inactivité égale à 3,5 fois la durée de transmission d'un seul caractère. Le segment Fin de la transmission à la fin d'un message indique à l'appareil récepteur que la transmission suivante sera un nouveau message et par conséquent un caractère d'adresse de périphérique.

A5.4 – Le "Contrôle de Redondance Cyclique"

Le contrôle de redondance cyclique (CRC) est un code de détection d'erreur qui a une longueur de deux octets (16 bits). Après construction d'un message (données uniquement, absence de bits de départ, d'arrêt et de parité), l'appareil émetteur calcule un code CRC et l'ajoute à la fin du message. Un appareil récepteur calcule un code CRC à partir du message qu'il a reçu. Si ce code CRC est différent du code émis, il y'a eu une erreur de communication.

Traduction de l'organigramme en pseudo-code :

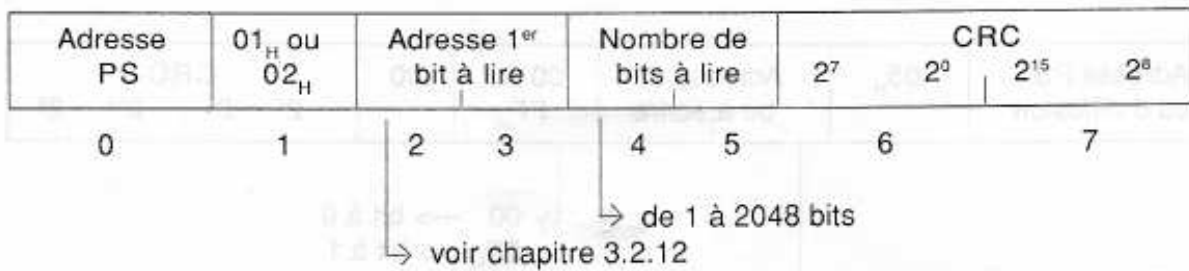
```
DEBUT
  CRC = FFFFh
  OCTET SUIVANT = premier octet de la trame
  REPETER
    CRC = CRC ⊕ OCTET SUIVANT décalé de 8 bits à gauche
    POUR CPT VARIANT DE 1 A 8
      FAIRE
        CRC = CRC décalé d'un bit à droite
        SI BIT DECALE := 1 ALORS
          FAIRE
            CRC = CRC ⊕ A001h
          FIN SI
        FIN POUR
    OCTET SUIVANT = octet suivant dans la trame
  TANT QU'IL RESTE DES OCTETS DANS LA TRAME
  FIN TANTQUE
FIN
```

Le symbole \oplus indique une opération 'OU exclusif'.

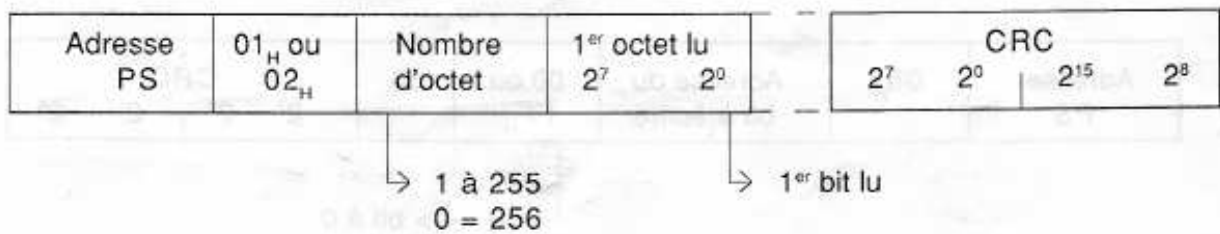
Algorithme du CRC sur le protocole Modbus

Fonction 1 ou 2

Demande de lecture bits (8 octets)

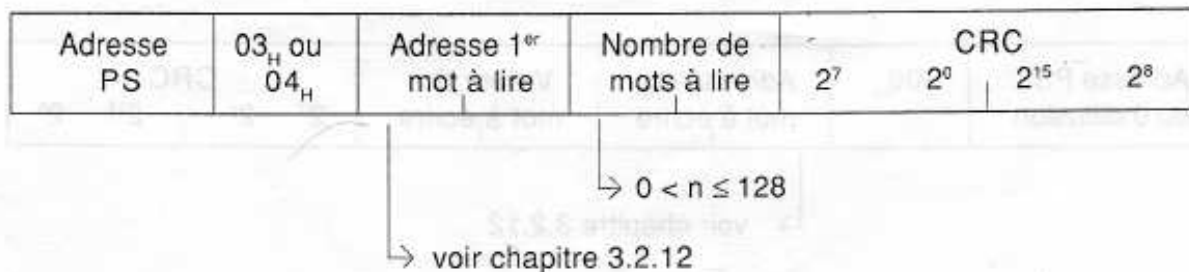


Réponse à une demande de lecture bits (de 6 à 261 octets)

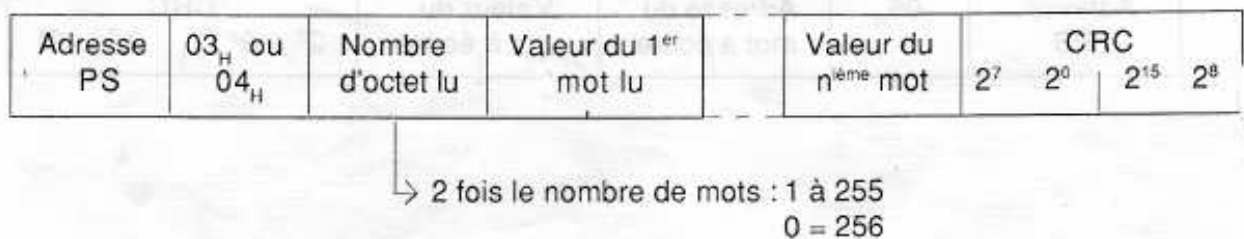


Fonction 3 ou 4

Demande de lecture mots (8 octets)



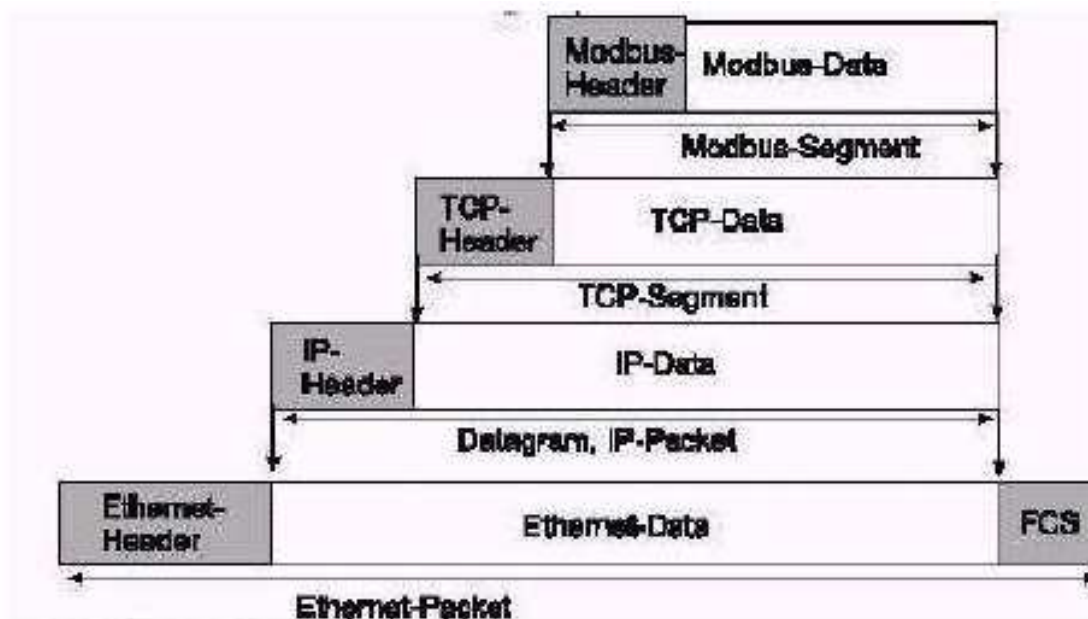
Réponse à une demande de lecture mots (de 7 à 261 octets)



Exemples de trames Modbus

A5.6 – Trame MODBUS portée par TCP/IP

Figure 16 : place de la trame Modbus dans une communication Ethernet. Représentation avec le modèle en couches OSI (Protocole Open Modbus).



Positionnement de la trame Modbus dans une communication Ethernet

Seule la trame Modbus doit être gérée car les autres « paquets » tel que TCP/IP sont directement gérés par Winsock.

Ce protocole fait la communication au **port 502** de l'esclave

A5.7 – Format de la trame Modbus porté par TCP/IP

Figure 17 :

<i>Transaction Identifier</i>	<i>Protocol Identifier</i>	<i>Length</i>	<i>Unit Identifier</i>	<i>Code de la fonction</i>	<i>Adresse de départ</i>	<i>Nombre de bits ou mots lus</i>
2 octets	2 octets	2 octets	1 octet	1 octet	2 octets	2 octets
Identificateur de la demande	0 car protocole Modbus	Nombres d'octets qui suivent	Adresse de l'esclave (non utilisé car l'adresse IP permet d'arriver sur l'esclave voulu).	Voir protocole Modbus (lecture ou écriture de bits ou de mots, etc.)	De la lecture dans l'esclave interrogé.	Dans l'esclave interrogé

Format de la trame Modbus porté par TCP/IP

A5.8 – Exemple de demande de lecture de huit bits :

Trame de la demande : lecture de 8 bits à partir de l'adresse 0

Figure 18 : les octets de 0 à 6 ne changent pas quelle que soit la demande effectuée, l'octet 7 indique la fonction exécutée et les quatre derniers différents en fonction de la demande. En effet, ces derniers permettent de sélectionner l'adresse du premier bit à récupérer et le nombre de bits à récupérer.

Byte	Field name	Example
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0006
Byte 6	unit identifier	0x01 not used
Byte 7	MODBUS function code	0x01
Byte 8, 9	reference number	0x0000
Byte 10, 11	Bit count	0x0008

Trame réponse :

Pour ce qui est de la réponse, les 7 premiers octets sont identiques à la demande. Le septième octet correspond à la fonction Modbus à laquelle il répond, le huitième octet indique le nombre d'octets renvoyés et le ou les derniers octets correspondant aux états demandés.

Figure 19

Byte	Field name	Example
.....		
Byte 7	MODBUS function code	0x01
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

Remarque : quand une trame Modbus est « portée » par TCP/IP, elle n'a plus besoin de son CRC car son contrôle est assuré par le protocole Ethernet.

Annexe 6 : Ressource logicielle industriels

- Schneider-Telemecanique :
 - **Unity Pro S** – Logiciel de programmation Automate
 - Pour les systèmes Centrale de pompage et équipement Photovoltaïque.
 - **PL7 Pro** – Logiciel de programmation Automate
 - Pour le système Surpressic, Unity Pro S ne fonctionnant pas sur celui-ci.
 - **Win Designer** – Logiciel de programmation Web
 - Pour les systèmes Surpressic, Centrale de pompage et Equipement Photovoltaïque, le logiciel doit être compatible TSX ETZ 510
 - **Factory Cast** – Logiciel de programmation Web
 - Pour les systèmes Surpressic, Centrale de pompage et Equipement Photovoltaïque, le logiciel doit être compatible TSX ETZ 510

- Siemens :
 - **Win CC Flexible** – Logiciel de programmation Web Siemens
 - **Step 7** – Logiciel de programmation Automate Siemens
 - Pour le système SPEN

- Superviseur RedLion G308C :
 - **Crimson 2.0** – Logiciel de programmation d'interface et de configuration du superviseur.

- Logiciel de développement :
 - **NVU** – Logiciel de programmation HTML.
 - **GIMP** – Logiciel de création graphique
 - **Notepad++** - Editeur de langages informatique multi-plateformes
 - **Open Office** – Logiciel Bureautique libre

- Caméra IP :
 - **Axis IP Utility** – Logiciel de configuration de la camera IP

- Télécontrôle industriel iRio :
 - **X Flow iRio** – Logiciel de configuration de l'iRio

- Bain régulateur :
 - **iTools Engineering 2.07** – Logiciel de configuration du régulateur de procédé INVENSYS Eurotherm série 3500 réf. 3504
 - **Simulateur du bain régulateur** – Exécutable sous MatLab Simulink

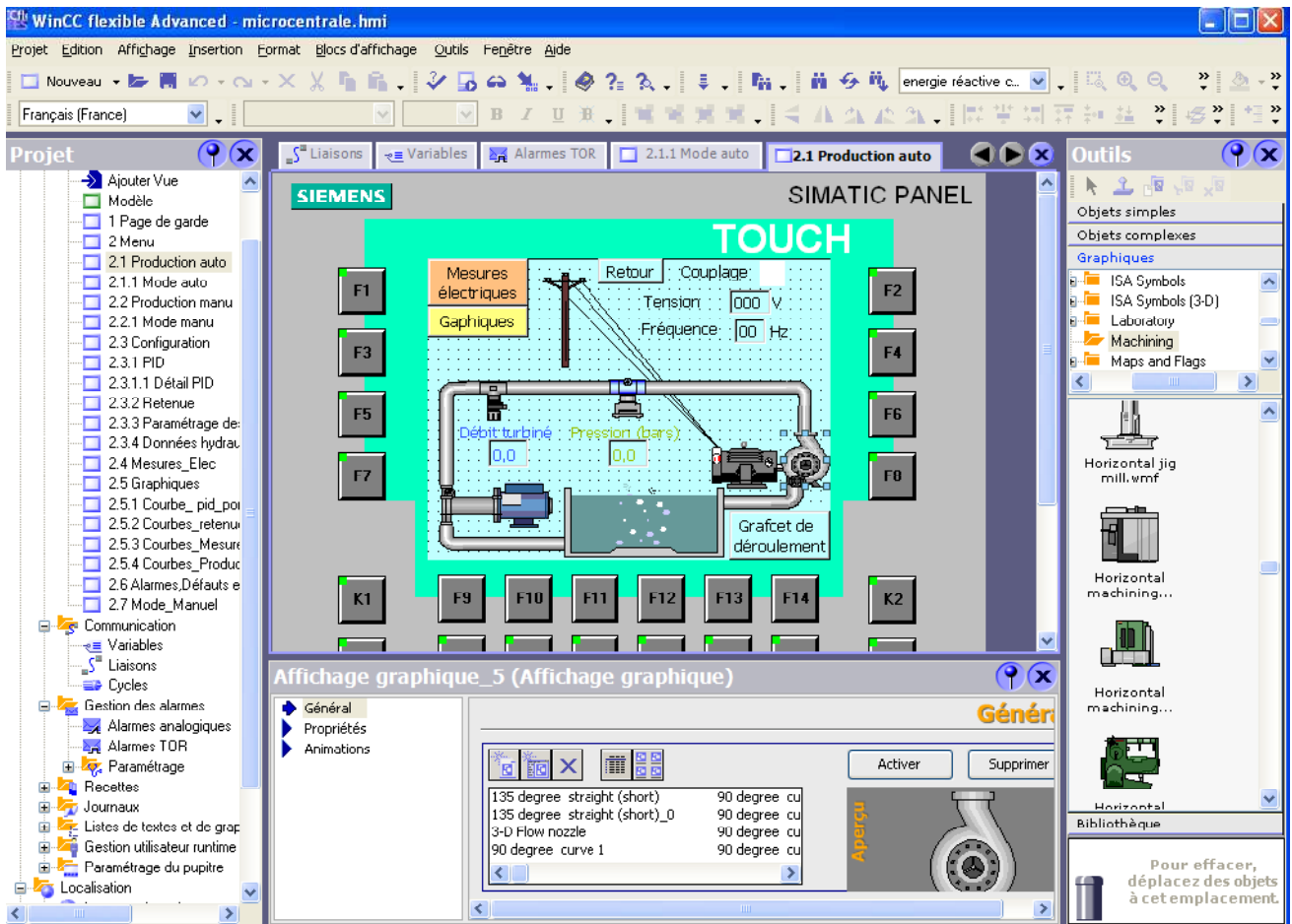
- Terminal Kep France 4156F :
 - **Easy Builder** – Logiciel de développement d'interface de commande à distance.
 - **Emulateur MMI 6000/8000** – Logiciel de simulation de superviseur MMI 6000/8000
 - **Logiciels de supervision** – téléchargeable sur le site de kepfrance (logiciel serveur, logiciel d'impression, logiciel FTP ...)

- TGBT :
 - **Wago Ethernet Setting** – Logiciel de configuration IP
 - **CoDeSys V2.3** – Logiciel de programmation automate Wago

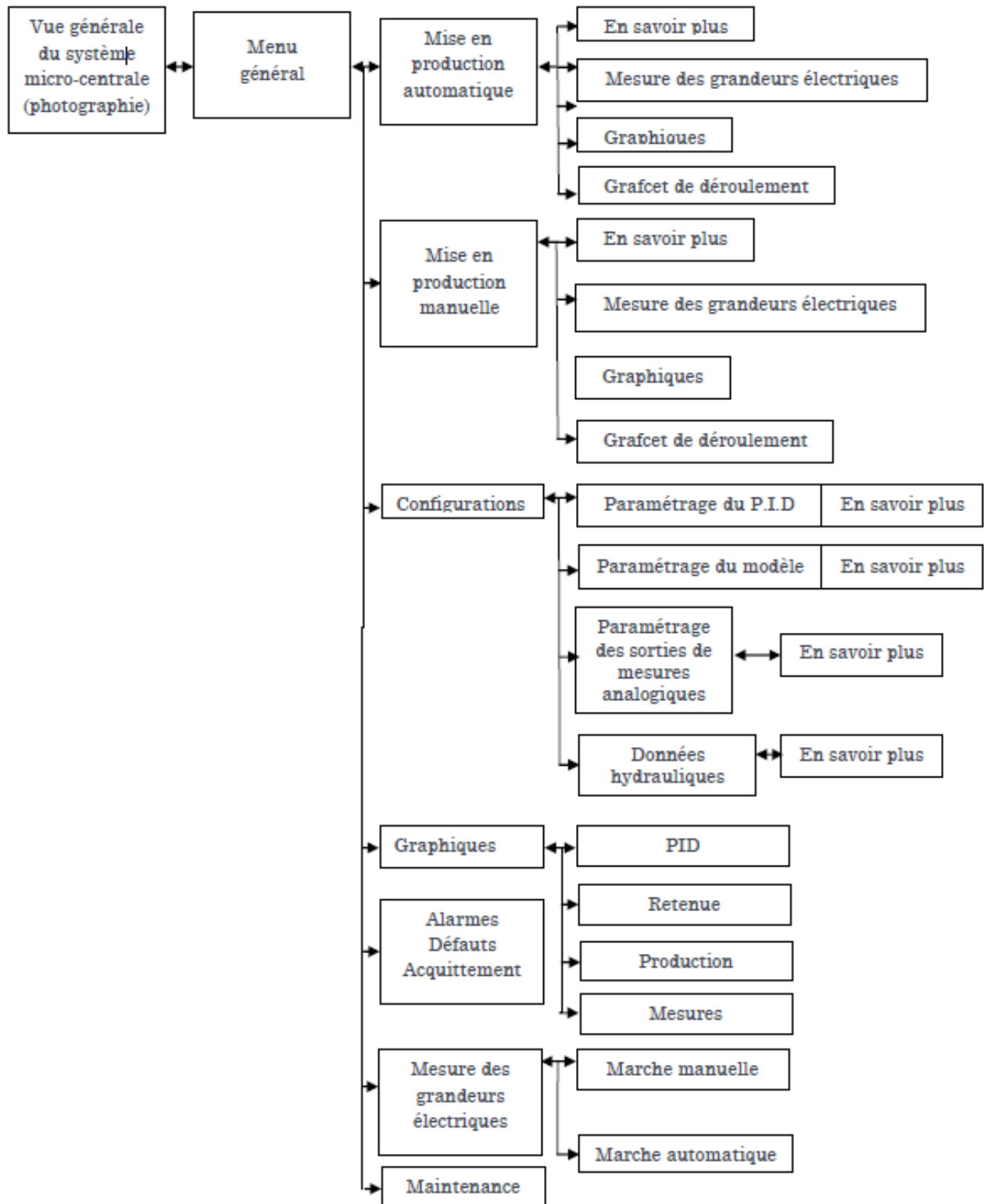
- GBT :
 - **PG5** – Logiciel de programmation

Annexe 7 : IHM du système industriel SPEN

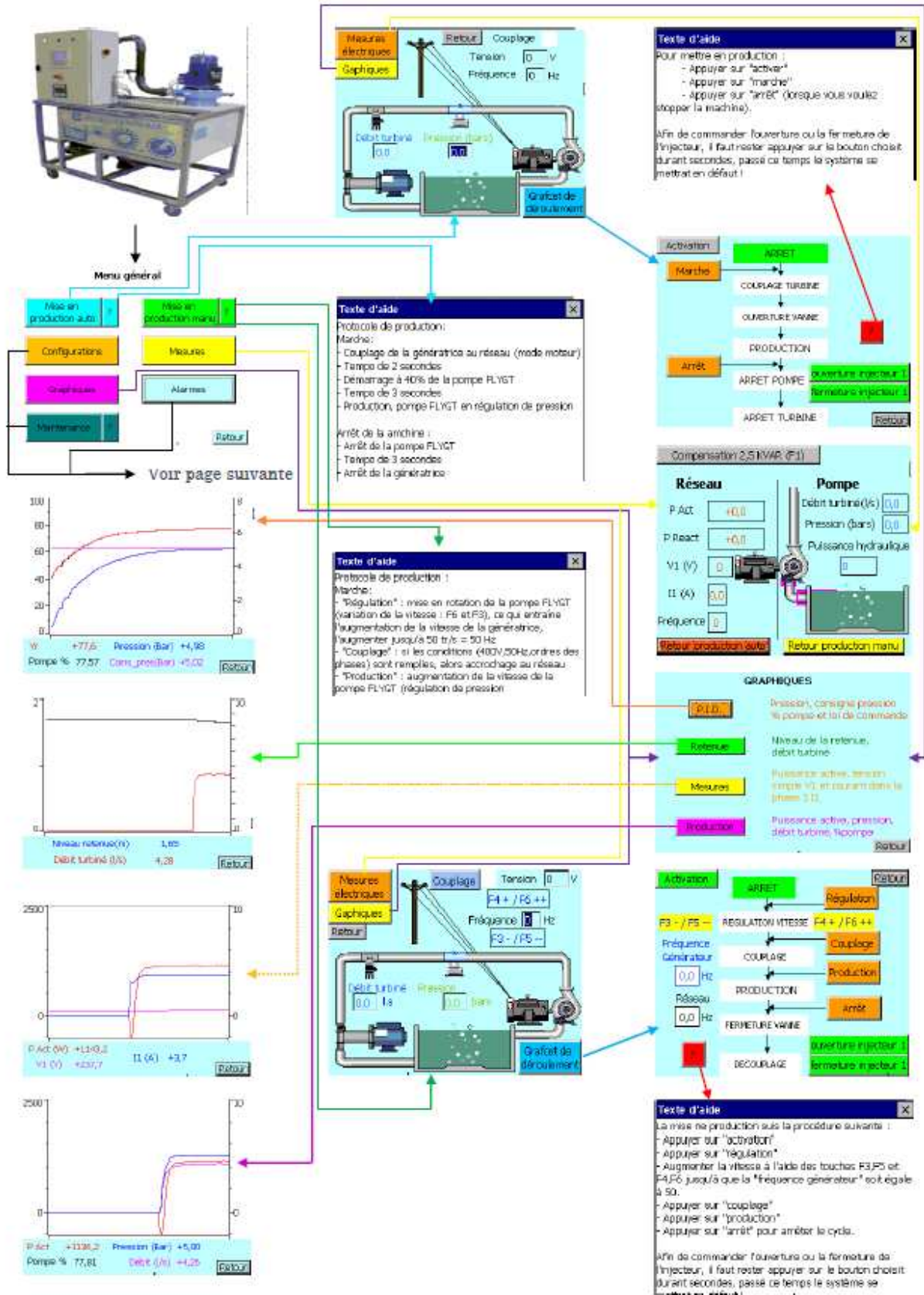
A7.1 – Micro-centrale SPEN



XII.7.2 – Organigramme du nouveau menu de la SPEN



XII.7.3 – Cheminement des nouvelles interfaces de la SPEN



Annexe 8 : Applet de connexion automate

A8.1 – Mise en œuvre d'une applet Java avec la bibliothèque

Intégration de la bibliothèque Schneider dans JBuilder 5.

A8.1.1 – Présentation

Je vais d'abord réaliser une applet utilisant les packages Schneider et la transférer vers le serveur web embarqué. Cette applet réalisera simplement une connexion avec l'automate. Cette partie du document est importante à détailler dans le cas où à la lecture de celui-ci, d'autres enseignants souhaitent essayer de mettre en œuvre des applets Java.

A8.1.2 – Matériels et logiciels nécessaires

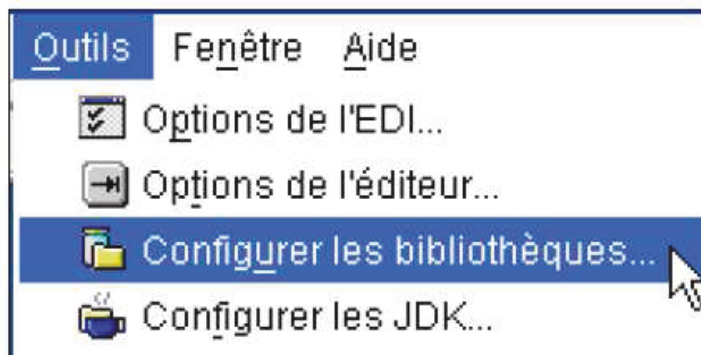
Il vous faudra un ordinateur qui possède :

- Le logiciel **JBuilder** version 5,
- Le logiciel **FactoryCast** avec le SDK,
- Le package Schneider **SAComm.jar** (malheureusement nous ne disposons pas de documentation sur les classes de ce package)
- Un navigateur Web (Firefox)
- Une machine virtuelle JAVA 1.4 ou >.

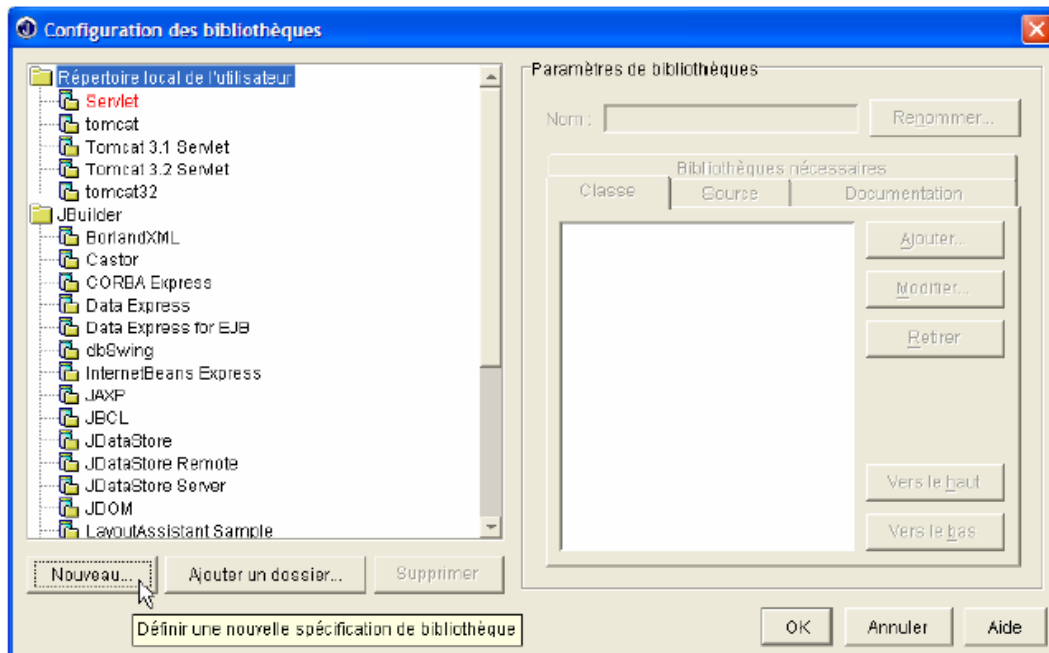
A8.1.3 – Création d'une bibliothèque JBuilder contenant SAComm.jar

Nous allons modifier l'environnement de développement de JBuilder en créant une nouvelle bibliothèque disponible pour tous les futurs projets.

Dans la barre d'outils de JBuilder, cliquez sur **Outils > Configurer les bibliothèques ...**



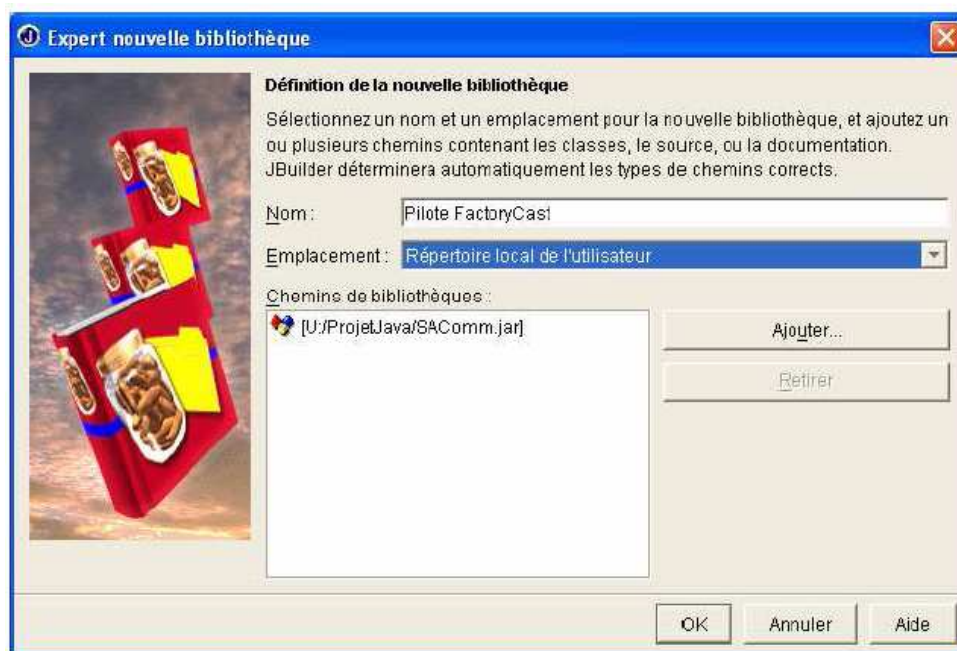
Dans la nouvelle fenêtre qui s'affiche, cliquez sur **Nouveau ...** en bas à gauche :



Dans le champ **Nom** : entrez un nom pour votre bibliothèque, ici *Pilote FactoryCast*.

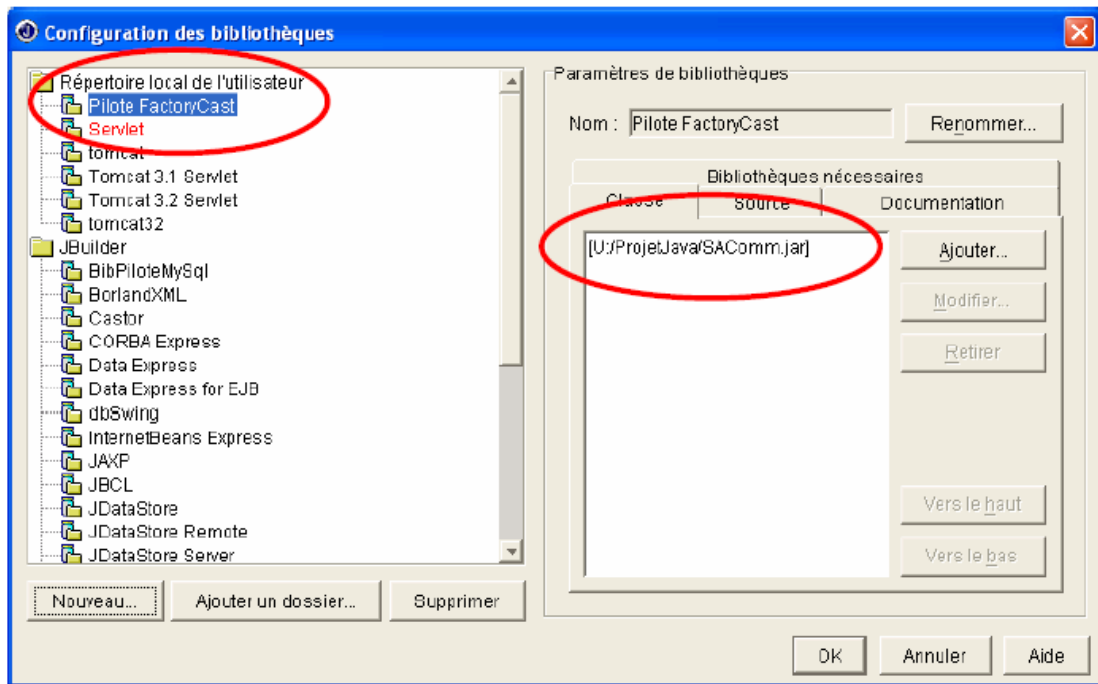
Dans le champ **Emplacement** : choisissez *Répertoire local de l'utilisateur* ou *JBuilder* (accessible à TOUS les utilisateurs de JBuilder).

Cliquez sur **Ajouter...** et choisissez le fichier « **SAComm.jar** »



Enfin cliquez sur **Ok**.

Dans la fenêtre de **Configuration des bibliothèques** vous devrez retrouver la nouvelle bibliothèque créée :



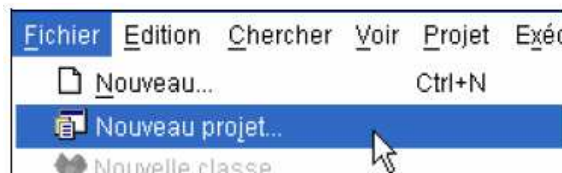
Cliquez sur **OK**.

La bibliothèque est désormais incorporée à JBuilder.

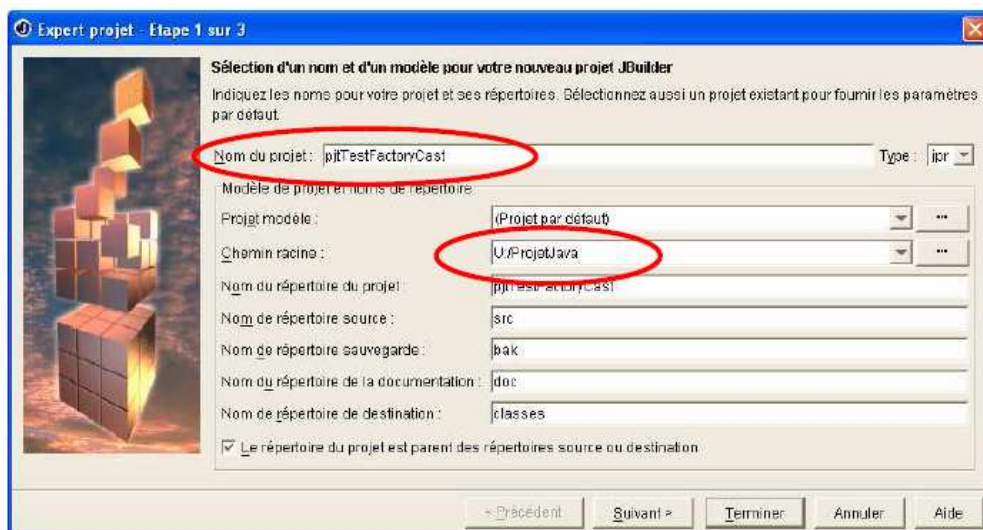
A8.2.4 – Création et configuration du nouveau projet

Notre projet va devoir inclure cette bibliothèque pour accéder à l'automate.

Cliquez sur **Fichier > Nouveau projet ...**

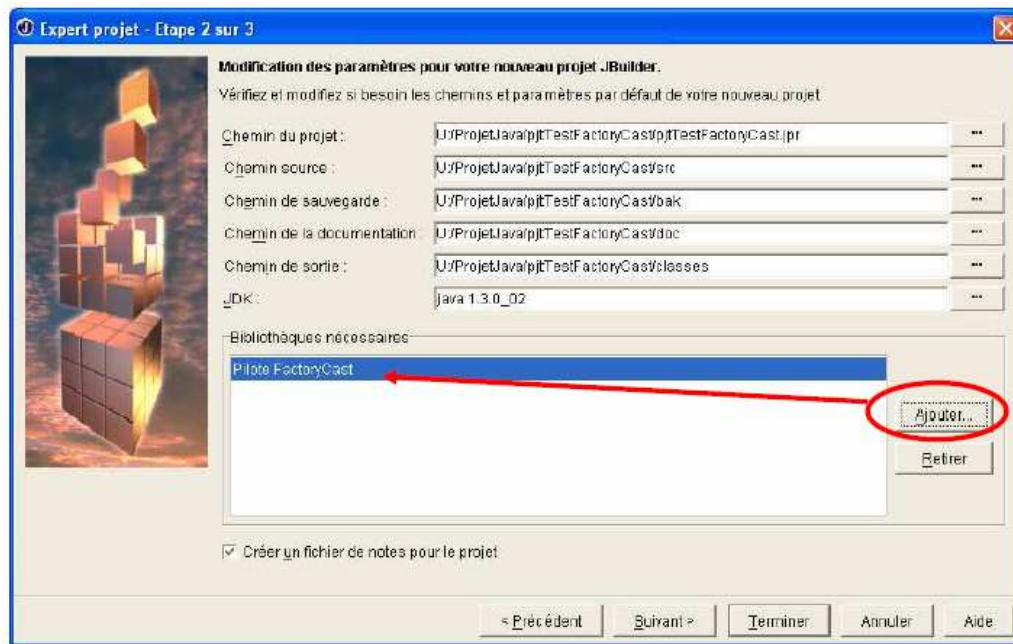


Donnez un nom au projet, par exemple `ProjetTestFactoryCast`, et renseignez également le champ « chemin racine ».



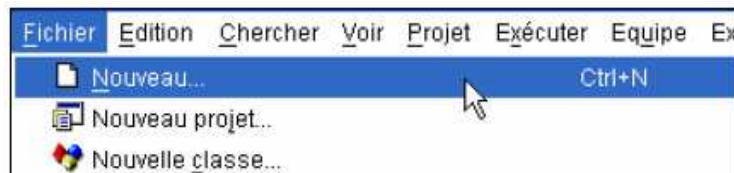
Cliquez sur **Suivant** :

Dans l'étape numéro deux, cliquez sur **Ajouter...** et sélectionnez la bibliothèque Pilote FactoryCast.

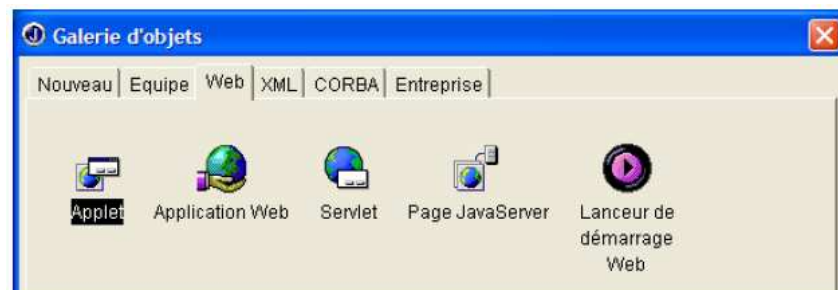


Cliquez sur **Terminer** :

A présent, cliquer sur **Fichier > Nouveau...**

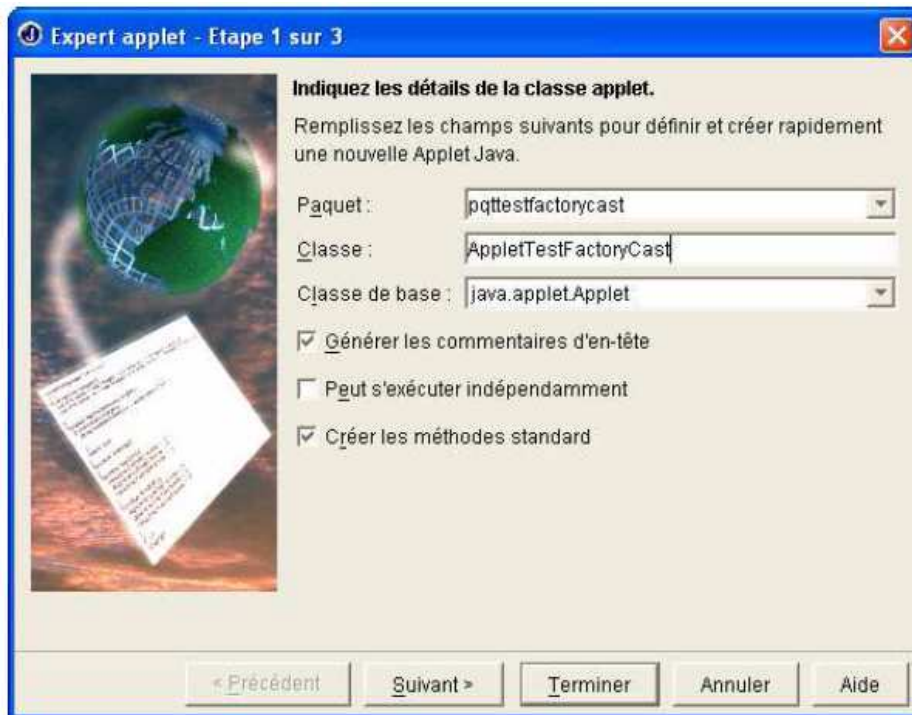


Dans l'onglet **Web**, sélectionnez **Applet** et cliquez sur **Ok**.



Nommez le paquet **pqtestfactorycast**, et la classe **AppletTestFactoryCast**.

Cochez les options « Générer les commentaires d'en-tête » et « Créer les méthodes standard ».



Cliquez sur **Terminer** :

Nous pouvons maintenant construire notre interface pour tester la connexion avec l'automate.

A8.3 – Test de connexion à l'automate

Après la configuration de l'éditeur, j'ai réalisé sous Jbulider un applet Java contenant seulement deux actions afin de tester dans un premier temps 2 variables de l'automate programmable.

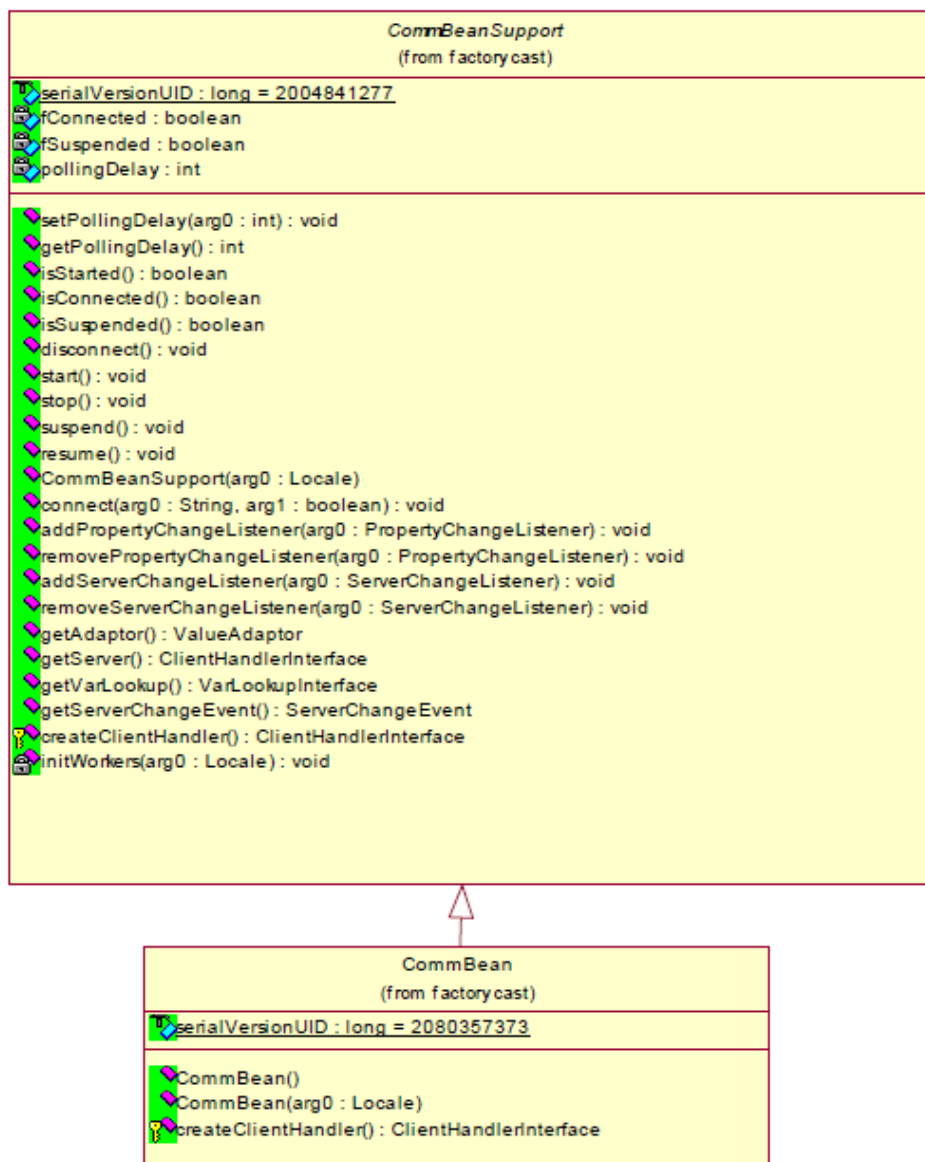
A8.3.1 – IHM servant au test

Cette IHM très simple servira à tester la connexion à l'automate.



A8.3.2 – Codage de l'applet

Le programme doit utiliser le package Factorycast fourni par le constructeur.



```
// import de la Bibliothèques FactoryCast
import com.schneiderautomation.factorycast.* ;
```

Il faut ensuite définir un objet de communication avec l'automate :

```
// Objet de communication avec l'automate
private CommBean commFactoryCast ;
// Création de l'objet de communication avec l'automate
commFactoryCast = new CommBean () ;
```


Il faut ensuite se connecter à l'automate et lancer la communication :

```
//Récupération de l'adresse IP de l'automate
adresseIPAutomate = new String("192.168.33.8") ;

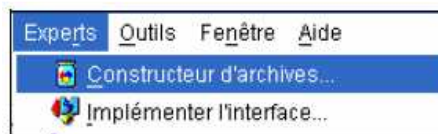
try {
    // On tente une connexion avec l'automate
    commFactoryCast.connect(adresseIPAutomate, false);
    // Si c'est correcte, on lance la communication
    commFactoryCast.start();
} catch (Exception ex) {
    // Gestion de l'erreur
    System.out.println("Erreur :" + ex.getMessage());
}
```

La déconnexion s'effectue comme suit :

```
try {
    commFactoryCast.disconnect() ;
} catch(Exception e) {
    // Gestion de l'erreur
    System.out.println("Erreur :" + e.getMessage());
}
```

A8.3.3 – Construction de l'archive de démarrage Web (.jar)

Cliquez sur **Experts > Constructeur d'archives...**

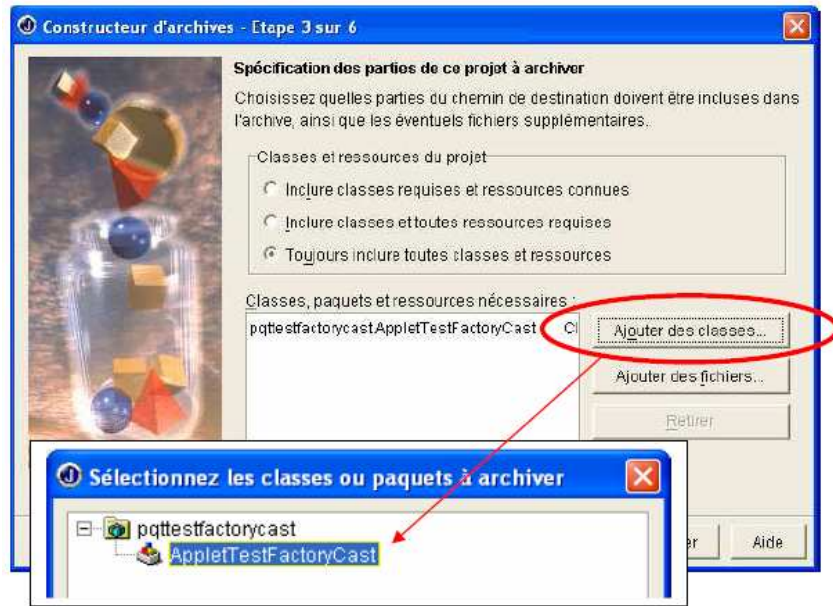


Sélectionnez « **Applet de démarrage Web** » comme Type d'archive.



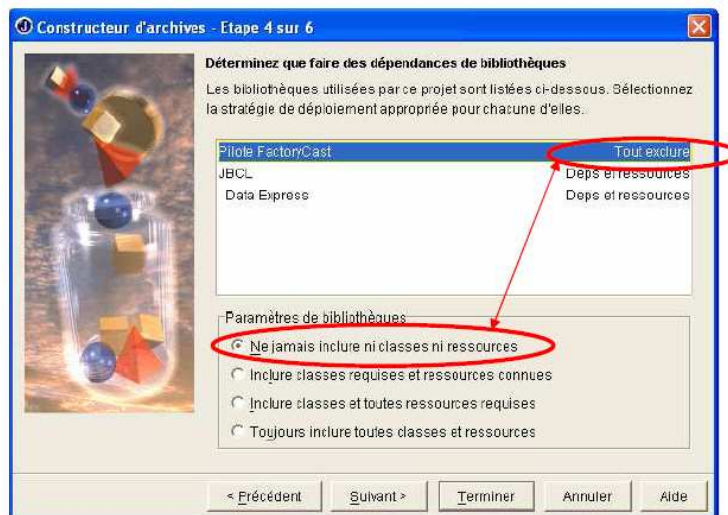
Cliquez sur **Suivant**.

A l'étape 3 / 5, cliquez sur « **Ajouter des classes** »



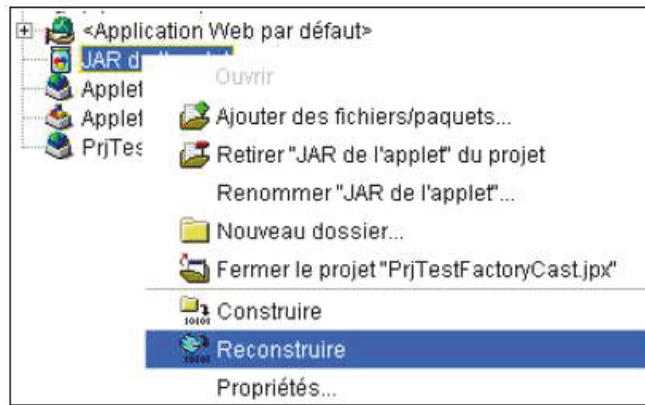
A l'étape numéro 4 / 5, sélectionnez « **Ne jamais inclure ni classes ni ressources** » pour le pilote FactoryCast.

En effet il est inutile d'intégrer les classes et les ressources de la bibliothèque Schneider dans cette archive car on utilisera plutôt l'archive SAcomm.jar.



Cliquez sur **Terminer** :

Pour compiler le projet et générer le fichier **JAR de l'applet** faites un clic droit sur JAR de l'applet et cliquez sur **Reconstruire** :



Maintenant nous allons modifier le code source du fichier **AppletTestFactoryCast.html** :

```
<html>
<head><title>Page de test HTML</title></head>
<body>
Page de Test de connexion <br>
<applet
  codebase = "/classes"
  archive = "SAComm.jar,pjtTestFactoryCast.jar"
  code = "pqttestfactorycast.AppletTestFactoryCast.class"
  name = "AppletTest"
  width = "400"
  height = "300"
  hspace = "0"
  vspace = "0"
  align = "top"
>
</applet>
</body>
</html>
```

Les archives étant stockées sur l'automate dans le répertoire **/wwwroot/classes/** et les pages HTML étant stockées dans le répertoire **/wwwroot/unsecure/user/** il est obligatoire de donner à la propriété **codebase** la valeur **"/classes"**.

Notre applet utilisant 2 archives (**SAComm.jar**, **pjtTestFactoryCast.jar**) il est obligatoire de les spécifier dans la propriété **archive**.

Il faut également donner le nom complet de la classe (paquet.class) dans la propriété **code**.

Enregistrez la modification.

A8.3.4 – Déploiement de l'applet sur l'automate

Il existe deux solutions pour transférer l'applet sur l'automate :

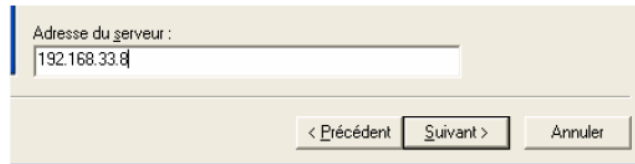
- Via un logiciel FTP classique
- Via le programme de configuration FactoryCast

A8.3.5 – Utilisation de WS_FTP Professional

Choisissez **fichier > connecter > assistant connexion** et renseignez le champ nom du site (par exemple ici : SiteWebEmb).

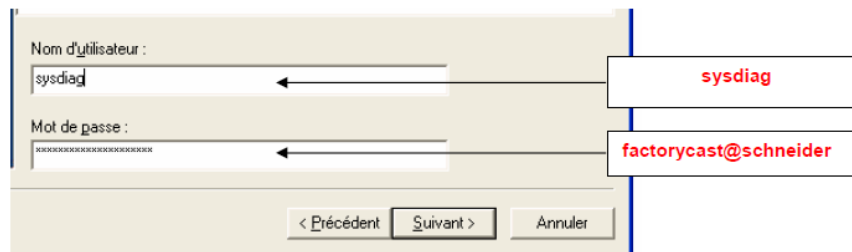
Cliquez sur **Suivant**, dans la fenêtre qui s'affiche, sélectionnez **FTP**.

Cliquez sur **Suivant**, dans la fenêtre qui s'affiche, renseignez le champ « **Adresse du serveur** »



A screenshot of a wizard window showing the 'Adresse du serveur' field. The text '192.168.33.8' is entered in the field. Below the field are three buttons: '< Précédent', 'Suivant >', and 'Annuler'.

Cliquez sur **Suivant**, dans la fenêtre qui s'affiche, renseignez les champs « **Nom d'utilisateur** » et « **Mot de passe** » comme ci-dessous :



A screenshot of a wizard window showing the 'Nom d'utilisateur' and 'Mot de passe' fields. The text 'sysdiag' is entered in the 'Nom d'utilisateur' field, and 'factorycast@schneider' is entered in the 'Mot de passe' field. Arrows point from the text boxes to the respective fields. Below the fields are three buttons: '< Précédent', 'Suivant >', and 'Annuler'.

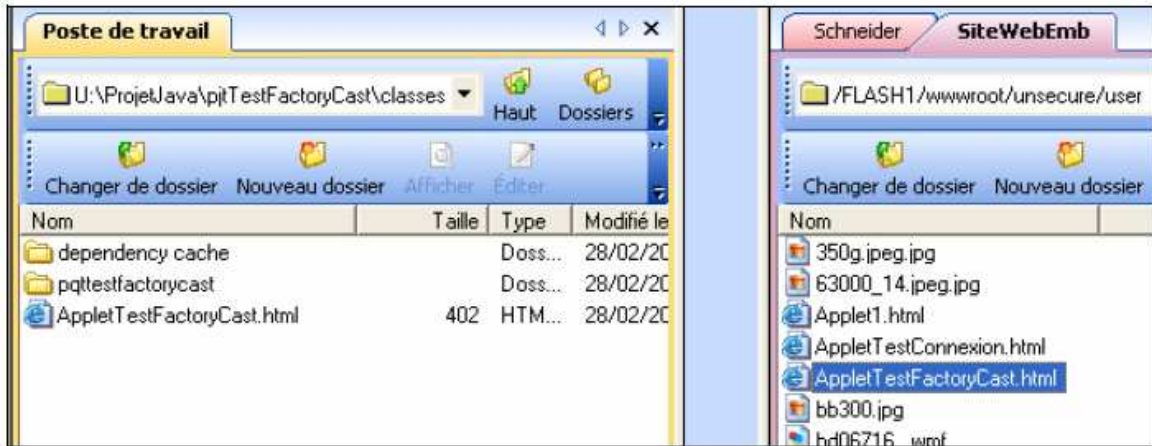
Cliquez sur **Suivant**, vérifiez ensuite les informations qui s'affichent dans la fenêtre



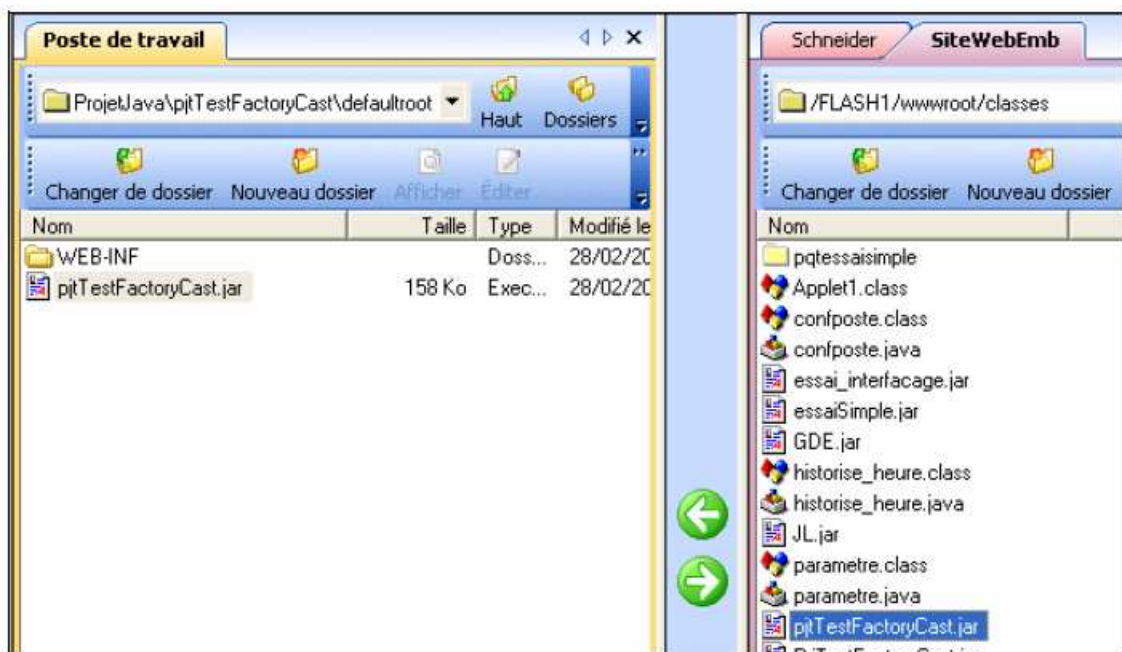
A screenshot of the 'Assistant Connexion' window. The title bar says 'Assistant Connexion'. The main area is titled 'Terminer'. It contains the following text: 'Si les informations suivantes sont correctes, cliquez sur Terminer pour créer le site.' and 'Si vous ne voulez pas vous connecter à ce serveur FTP, désélectionnez l'option Connecter à ce site.' Below this is a list of information: 'Nom de connexion : SiteWebEm', 'Adresse du serveur : 192.168.33.8', 'Nom d'utilisateur : sysdiag', and 'Mot de passe : * * * * *'. At the bottom, there is a checkbox labeled 'Connecter à ce site' which is checked, and a button labeled 'Avancé...'. Below the checkbox are three buttons: '< Précédent', 'Terminer', and 'Annuler'.

Validez par le bouton **Terminer**, la connexion est établie.

Copiez maintenant le fichier **AppletTestFactoryCast.html** situé dans le répertoire **classes** du répertoire de sauvegarde du projet dans le répertoire **/wwwroot/uncsure/user/** du serveur.



Copiez maintenant le fichier **pjtTestFactoryCast.jar** situé dans le répertoire **defaultroot** du répertoire de sauvegarde du projet dans le répertoire **/wwwroot/classes/** du serveur.

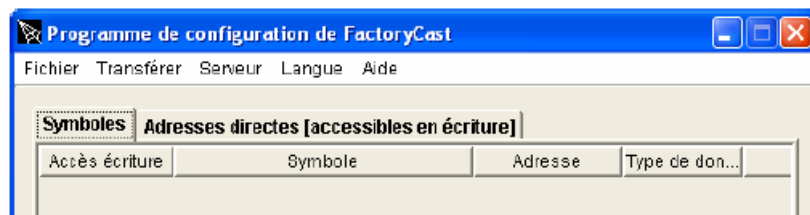


A8.3.6 – Utilisation du programme de configuration de Factory Cast

Lancez le programme de configuration de FactoryCast via :

Démarrer>Tous les programmes>Schneider Electric>FactoryCast>Programme de configuration de FactoryCast

La fenêtre suivante :



Cliquez ensuite sur **Transférer>Fichiers Web ...**



Renseignez ensuite le champ « **Nom de serveur ou adresse IP** » avec l'adresse IP de l'automate.

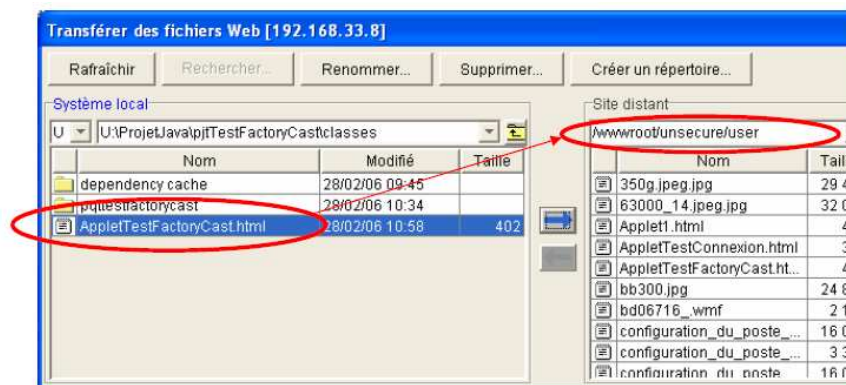


Cliquez sur OK.

Dans la fenêtre qui s'affiche, sous le panel « **Systèmes local** » placez-vous dans le sous-répertoire **classes** du répertoire de votre projet.

Sous le panel site distant, placez-vous sous **/wwwroot/unsecure/user**.

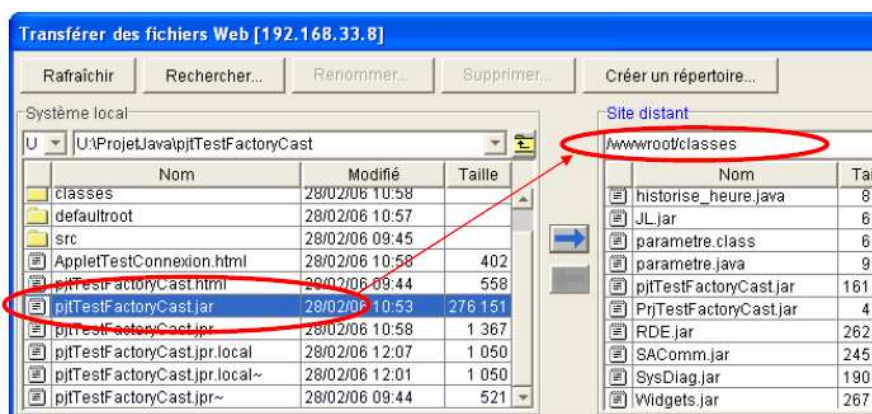
Sélectionnez ensuite votre page HTML (ici : **AppletTestFactoryCast.html**) dans le panel « **Système local** » et cliquez sur la flèche de transfert :



Puis, sous le panel « **Système local** » placez vous dans le répertoire de votre projet.

Sous le panel site distant, placez-vous sous **/wwwroot/classes**.

Sélectionnez ensuite votre archive (ici : **pjtTestFactoryCast.jar**) dans le panel « **Système local** » et cliquez sur la flèche de transfert :



A8.3.7 – Exécution de l'applet

Maintenant on peut visualiser la page HTML contenant notre applet dans un navigateur web à l'adresse suivante : **<http://192.168.33.8/unsecure/user/AppletTestFactoryCast.html>**

Par ailleurs, le logiciel d'édition d'interface graphique pour serveur web embarqué de Schneider : **FactoryCast** n'est rien d'autre qu'un constructeur de classe Java. En effet, les fichiers qu'il génère sont des extensions *.java qui sont ensuite compilés et exécuté par le navigateur web.

Annexe 9 : Méthodologie de développement

A9.2 – Méthode Agile

Les méthodes Agiles sont des procédures de conception de logiciel qui se veulent plus pragmatiques que les méthodes traditionnelles. En impliquant au maximum le demandeur (client), ces méthodes permettent une grande réactivité à ses demandes, visent la satisfaction réelle du besoin du client, et non des termes du contrat de développement.

On peut ainsi dire que les méthodes Agile sont définies comme des « Méthodes itératives à planification souple qui leur permettent de s'adapter à la fois aux changements de contexte et de spécifications du projet ». C'est pour cette raison qu'elles m'ont d'abord semblées intéressantes à mettre en place.

Valeurs :

Dans ce but, elles prônent 4 valeurs fondamentales :

- **L'équipe** (« Personnes et interaction plutôt que processus et outils ») : Dans l'optique agile, l'équipe est bien plus importante que les outils (« structurants ou de contrôle) ou les procédures de fonctionnement. Il est préférable d'avoir une équipe soudée et qui communique composée de développeurs (éventuellement à niveaux variables) plutôt qu'une équipe composée d'experts fonctionnant chacun de manière isolée. La communication est une notion fondamentale.
- **L'application** (« Logiciel fonctionnel plutôt que documentation complète ») : Il est vital que l'application fonctionne. Le reste, et notamment la documentation technique, est une aide précieuse mais non un but en soi. Une documentation précieuse est utile comme moyen de communication. La documentation représente une charge de travail importante, mais peut pourtant être néfaste si elle n'est pas à jour. Il est préférable de commenter abondamment le code lui-même, et surtout de transférer les compétences au sein de l'équipe (on en revient à l'importance de la communication).
- **La collaboration** (« Collaboration avec le client plutôt que négociation de contrat ») : Le client doit être impliqué dans le développement. On ne peut se contenter de négocier un contrat au début du projet, puis de négliger les demandes du client. Le client doit collaborer avec l'équipe et fournir un feed-back continu sur l'adaptation du logiciel à ses attentes.
- **L'acceptation du changement** (« Réagir au changement plutôt que suivre un plan ») : La planification initiale et la structure du logiciel doivent être flexibles afin de permettre l'évolution de la demande du client tout au long du projet. Les premiers *releases* du logiciel vont souvent provoquer des demandes d'évolution.

Parmi ces méthodes, eXtreme Programming (XP) est apparu depuis quelques années comme une méthodologie adaptée aux petits projets. Elle utilise un processus itératif permettant de toujours rester en accord avec les besoins du client. L'accent mis sur les technologies et le codage permet également de garder le contact avec les contraintes de la réalisation.

Enfin sa simplicité de mise en œuvre en fait une méthodologie appréciée dans le contexte de développement dont le cycle de vie est court.

A9.3 – Méthodes de développement

A9.3.1 – Méthodologie XP (eXtreme Programming)

L'extreme programming (XP) est une méthode de développement « agile », principalement destinée à des petites équipes confrontées à un environnement mal connu ou changeant. On peut aussi parler d'un ensemble de pratiques destinées à organiser le travail d'une équipe de développement. Ces pratiques se focalisent sur la construction proprement dite au logiciel, en aval des phases préparatoires d'études d'opportunité ou de faisabilité.

La méthode **XP** (eXtreme Programming) est très axée sur la partie construction de l'application. Une de ses originalités réside dans l'approche de planification qui se matérialise sous la forme d'un jeu intitulé **Planning game** et qui implique simultanément les utilisateurs et les développeurs.

On notera aussi des techniques particulières liées à la production du code comme la programmation en binôme (Pair programming), l'appropriation collective du code, la Refactorisation (refactoring) et l'Intégration continue. La méthode RAD préconise dans ce sens des revues de codes personnelles, puis collectives et l'intégration avant chaque Focus (ou Show).

Par contre, le RAD limite la programmation en binôme aux parties les plus stratégiques.

L'ensemble des pratiques organisées autour des principes suivants :

- **Le client (maîtrise d'ouvrage) pilote lui-même le projet**, et ce de très près grâce à des cycles itératifs extrêmement courts (1 ou 2 semaines).
- **L'équipe livre très tôt dans le projet** une première version du logiciel, et les livraisons de nouvelles versions s'enchaînent ensuite à un rythme soutenu pour obtenir un feedback maximal sur l'avancement des développements.
- **L'équipe s'organise elle-même pour atteindre ses objectifs**, en favorisant une collaboration maximale entre ses membres.
- **L'équipe met en place des tests automatiques** pour toutes les fonctionnalités qu'elle développe, ce qui garantit au produit un niveau de robustesse très élevé.
- **Les développeurs améliorent sans cesse la structure interne du logiciel** pour que les évolutions y restent faciles et rapides.

Elle pousse à l'extrême des principes simples.

Pratiques extrêmes

Les principes de cette méthode ne sont pas nouveaux : ils existent dans l'industrie du logiciel depuis des dizaines d'années et dans les méthodes de management depuis encore plus longtemps.

L'originalité de la méthode est de les pousser à l'extrême :

- Puisque la revue de code est une bonne pratique, elle sera faite en permanence (par un binôme) ;
- Puisque les tests sont utiles, ils seront fait systématiquement avant chaque implémentation ;
- Puisque la conception est importante, elle sera faite tout au long du projet (refactoring) ;
- Puisque la simplicité permet d'avancer plus vite, nous choisirons toujours la solution la plus simple ;
- Puisque la compréhension est importants, nous définirons et ferons évoluer ensemble des métaphores ;
- Puisque l'intégration des modifications est cruciale, nous l'effectuerons plusieurs fois par jour ;
- Puisque les besoins évoluent vite, nous ferons des cycles de développement très rapides pour nous adapter au changement.

Valeurs

L'Extreme Programming repose sur 4 valeurs fondamentales :

La communication

C'est le moyen fondamental pour éviter les problèmes. Les pratiques que préconise l'XP imposent une communication intense. Les tests, la programmation en binôme et le jeu du planning obligent les développeurs, les décideurs et les clients à communiquer. Si un manque apparait malgré tout, un *coach* se charge de l'identifier et de remettre ces personnes en contact.

La simplicité

La façon la plus simple d'arriver au résultat est la meilleure. Anticiper les extensions futures est une perte de temps. Une application simple sera plus facile à faire évoluer.

Le feedback

Le retour d'information est primordial pour le programmeur et le client. Les tests unitaires indiquent si le code fonctionne. Les tests fonctionnels donnent l'avancement du projet. Les livraisons fréquentes permettent de tester les fonctionnalités rapidement.

Le courage

Certains changements demandent beaucoup de courage. Il faut parfois changer l'architecture d'un projet, jeter du code pour en produire un meilleur ou essayer une nouvelle technique. Le courage permet de sortir d'une situation inadaptée. C'est difficile, mais la simplicité, le *feedback* et la communication rendent ces tâches accessibles.

Néanmoins, les inconvénients majeurs d'XP sont les suivants :

- Il ne couvre pas les phases en amont et en aval au développement (capture des besoins, support, maintenance, tests d'intégration),
- Il élude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et à défaire,
- Assez flou dans la mise en œuvre, il ne définit pas clairement les intervenants et les documents livrables

BIBLIOGRAPHIE

- [1] « *L'utilisation pédagogique et l'enseignement des TIC dans les automatismes* », B. Riera, F. Gellot, O. Dubois, J.P. Chemla, S. Triki, EDP Sciences 2005.
- [2] « *Pilotage à distance de systèmes industriels didactisés et évaluation de la pratique des étudiants* », S. Mias, P. Maussion, IUFM Midi-Pyrénées 2008.
- [3] « *Conception d'une chaîne d'acquisition virtuelle à but didactique* », F. Robert et C Boey, p357, CETSIS 2003, Toulouse.
- [4] « *Développement d'une chaîne d'acquisition virtuelle suivant un formalisme à trois niveaux* », A. Vander Biest, C. Boey, A. Nonclercq, F. Robert, CETSIS 2005, Nancy.
- [5] « *L'Instrumentation Virtuelle comme outil pédagogique* », B. Philippe, CETSIS 2005, Nancy.
- [6] « *Mesures automatiques en électrotechnique* », E. David, A. Dworak, F. Louboutin et G. Stevenard, p115, CETSIS 2003, Toulouse.
- [6] « *Régulation de vitesse pour MCC : choix d'un contrôle à distance* », A. Tauvel, E. Flety, CETSIS 2005, Nancy.
- [7] « *TD/TP d'automatique à distance : comment assurer une interactivité avec l'enseignant ?* », D. Theilliol et J.C. Ponsart, p401, CETSIS 2003, Toulouse.
- [8] « *Un centre de ressources virtuel partagé* », T. Zimmer, D. Geoffroy, M. Billaud et Y. Danto, p443, CETSIS 2003, Toulouse.
- [9] « *Contrôle et Supervision à Distance : Réseaux Industriels, Internet et Sécurité* », E. Bajic, O. Bellot, O. Douchin, Ch. Saunal, CETSIS 2005, Nancy.

LISTE DES FIGURES

Fig. II.1.2 – Organigramme de l’entreprise DMS	14
Fig. IV.2.1.2 Connexions simulateur de PO, superviseur, API.....	28
Fig. VI.1 – Schéma de fonctionnement global.....	34
Fig. VII.1 – Architecture générale du réseau du Lycée.....	40
Fig. VII.2.a – Architecture du réseau informatique de la zone Electrotechnique (Projet 2008)	42
Fig. VII.2.b – Architecture du réseau informatique de la zone Electrotechnique (Projet 2010)	43
Fig. VII.9.2.1.a – Automate et coupleur de communication.....	53
Fig. VII.9.2.1. – Description du coupleur.....	53
Fig. VII.9.2.2 – Interface de configuration pour la communication avec un système.....	53
Fig. VII.9.2.3.a – Interface de configuration pour la communication avec un système.....	54
Fig. VII.9.2.3.b – Use case « Pilotage Superviseur ».....	54
Fig. VII.9.2.3.c – Diagramme de séquence de la « Connexion, de la lecture et la déconnexion	55
Fig.8.1 – Schéma de principe de la plateforme.....	58
Fig. 8.2 – Schéma des verrous technologiques.....	60
Fig. 8.3. – Schéma de synthèse de fonctionnement.....	61
Fig. 8.4. – Planning prévisionnel.....	62
Fig. 9.5.1 – Diagramme de cas d’utilisation Application – Elève.....	71
Fig. 9.5.2 – Diagramme de cas d’utilisation Application – Professeur.....	72
Fig. 9.5.3 – Diagramme de cas d’utilisation IHM-Serveur.....	73
Fig. 9.5.4 – Diagramme de cas d’utilisation Serveur-Machine.....	74
Fig. 9.5.5 – Diagramme de cas d’utilisation Serveur Principal.....	75
Fig. 9.6.1 – Diagramme de composants Application-Elève.....	76
Fig. 9.6.2 – Diagramme de composants Application-Professeur.....	77
Fig. 9.5.3 – Diagramme de composants Serveur-Machine.....	77
Fig. 9.5.4 – Diagramme de composants Serveur Principal.....	78
Fig. 10.1.a – Schéma de l’architecture logicielle de la plateforme.....	80
Fig. 10.1.b – Schéma d’architecture matérielle.....	81
Fig. 10.2.a – Plan d’aménagement de la zone Electrotechnique.....	82
Fig. 10.2.b – Schéma du réseau local de l’atelier BTS Electrotechnique.....	83
Fig. 10.2.a – Schéma du réseau local de l’atelier BTS Electrotechnique avec les adresses IP.	84
Fig. 10.2.b – Récapitulatifs des adresses IP distribuées.....	84
Fig. 10.3.a – Ouverture de la connexion vers l’extérieur.....	90
Fig. 10.3.1 – Principe de la DMZ	94
Fig. 10.4 – Schéma du réseau de connexion DMS-Lycée Déodat de Séverac.....	95
Fig.XIII.2 – Planning globale détaillé.....	122
Fig. XIII.3.4.7.5 - Schéma de fonctionnement des services web d’un coupleur automate/serveur web.....	133
Fig.XIII.5.4 - Schéma de principe du modèle-vue-contrôleur.....	140
Fig.XIII.5.4.b – Echange entre modèle-vue-contrôleur.....	142
Fig. XIII.7 - Schéma de bases de données.....	146
Fig. XIII.8.a – Exemple de diagramme de PERT suite à un découpage en tâches.....	147
Fig.XIII.8.b – Exemple d’organigramme WBS suite à un découpage en tâches.....	148
Fig.XIII.9.1 – Structure initiale WBS du projet.....	149
Fig.XIII.9.2 – Structure WBS Phase de conception 2	151
Fig.XIII.9.3 – Structure WBS Phase de conception 3.....	151
Fig.XIII.9.5 – Structure WBS Phase de conception 5.....	152
Fig.XIII.9.6 – Structure WBS Phase de conception 6.....	153
Fig.XIII.9.7 – Structure WBS Phase de conception 7.....	153
Fig. XIV.4.1 – Diagramme de classe Bureau.....	162
Fig. XIV.4.2 – Diagramme de classe Cours & Ressources.....	163
Fig. XIV.4.3 – Diagramme de classe TP & Systèmes.....	164

Fig. XIV.4.4 – Diagramme de classe Utilisateurs & Authentification.....	166
Fig. XIV.4.5 – Diagramme de classe Zone TP	167
Fig.XIV.3.1 – IHM « Travaux en cours ».....	169
Fig.XIV.3.2 – IHM « Sélection des TP ».....	170
Fig.XIV.3.3 – IHM « Systèmes disponibles ».....	171
Fig.XIV.3.4 – IHM « Présentation du TP »	171
Fig.XIV.3.5 – IHM « Visualisation du système ».....	172
Fig.XIV.3.6 – IHM « Visualisation des données».....	173
Fig.XIV.3.7 – IHM « Visualisation des données».....	173
Fig.XIV.4.1 – Arborescence des sources du système 184.....	174

Développement d'une plateforme d'apprentissage et de pilotage de systèmes industriels pour la réalisation de travaux pratiques.

Mémoire d'Ingénieur C.N.A.M, Toulouse 2011

RESUME

L'utilisation d'outils pédagogiques fait partie depuis longtemps du système éducatif. Tableaux, craies, manuels, compas, etc. sont devenus des outils tellement familiers que leur remise en cause serait, selon certains, impossible.

Au cours du 20^{ème} siècle, le système éducatif s'approprie, avec plus ou moins de volonté, les médias et innovations technologiques développés par notre société. Ainsi, les termes de ressources numériques, multimédias, plateforme, etc. sont apparus, entraînant un bouleversement des méthodes éducatives, pas toujours bien compris par les enseignants.

D'ailleurs, il y'a encore peu de temps elles étaient notamment réservées à des écoles d'ingénieurs, dans les établissements d'études supérieurs ou laboratoire de recherche, ils sont maintenant présents de plus en plus dans les établissements scolaires secondaires tels que les Lycées.

Nous verrons dans cette présentation que les évolutions pédagogiques, notamment en terme de ressources, ont toujours tentés de suivre les évolutions technologiques mais que ce rapport pédagogie/technologie est tributaire de paramètres bien plus compliqués en notamment l'apprentissage et l'enseignement.

Mots clés : pédagogie, apprentissage, éducation, formation, enseignement à distance.

SUMMARY

The use of teaching tools has long been part of the education system. Tables, chalk, textbooks, compass, etc... Tools have become so familiar that their challenge would be, some say, impossible.

During the 20th century, the education system takes, with varying degrees of willingness, media and technological innovations developed by our company. Thus, the terms of digital, multimedia platform, etc. appeared, causing a disruption of educational methods, not always well understood by teachers.

Besides, there is only a short time they were especially reserved for engineering schools, in institutions of higher education or research laboratory, they are now increasingly present in schools such as secondary high schools.

We will see in this presentation that the educational developments, particularly in terms of resources, are always tempted to follow technological developments but the report pedagogy / technology depends on more complicated parameters such as learning and teaching.

Keywords: teaching, learning, education, training and education.