HAL
open science

Développement et utilisation d'applications complémentaires dans une approche SCADA : retour sur expérience avec des étudiants Bac+2 et Bac+5

Pascal Vrignat, Jean-François Millet, Florent Duculty, Stéphane Begot, Manuel Avila

To cite this version:

Pascal Vrignat, Jean-François Millet, Florent Duculty, Stéphane Begot, Manuel Avila. Développement et utilisation d'applications complémentaires dans une approche SCADA : retour sur expérience avec des étudiants Bac+2 et Bac+5. 2014. hal-01079048

HAL Id: hal-01079048

https://hal.science/hal-01079048v1

Submitted on 31 Oct 2014

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

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

Développement et utilisation d'applications complémentaires dans une approche SCADA : retour sur expérience avec des étudiants Bac+2 et Bac+5

Pascal Vrignat, Jean-François Millet, Florent Duculty, Stéphane Begot, Manuel Avila.
pascal.vrignat@univ-orleans.fr
Laboratoire PRISME – Université d'Orléans
IUT de l'Indre, 2 Avenue François Mitterrand 36000 CHATEAUROUX

RESUME : Depuis quelques années, une des thématiques de recherche développée à l'IUT de l'Indre en collaboration avec des partenaires industriels locaux concerne l'estimation du niveau de dégradation d'un processus à l'aide d'un Modèle de Markov Caché (MMC) à partir des informations issues du service de maintenance. Pour cela, l'accès aux données associées à un processus en ligne, à un instant « souhaité », et la sélection des informations les plus pertinentes dans un volume toujours plus important sont nécessaires et utiles pour le développement d'une entreprise. Dans le cadre de la collecte des informations à partir d'un réseau informatique industriel multi-couches et de l'utilisation d'outils logiciels appropriés, nous avons mis en place au sein de nos formations un module intitulé « Supervision ». Ce module aborde entre autres, les problèmes rencontrés autour d'un serveur OPC et la connexion au processus à surveiller par différents utilisateurs à partir de technologies diverses. Intervenant également dans d'autres composantes de l'Université d'Orléans dont Polytech', cet enseignement est proposé à des étudiants de dernière année de cycle d'Ingénieur de la spécialité « Productique ».

Mots clés : simulation de processus, SCADA, OPC, multi-clients, SmartPhone, tablette.

1 INTRODUCTION

Dans le monde exigeant du contrôle-commande de processus, l'accès à des données de processus et/ou d'usine en ligne et en temps « *souhaité* » est crucial. La croissance d'une entreprise se traduit par une croissance parallèle du volume des données issues du processus et de la demande de traduction en informations pertinentes pour les équipes d'exploitation et de direction [8]. Souvent, les personnes qui ont en charge un système d'information doivent s'attacher à connecter et récupérer les informations pertinentes de leur usine via leur système informatique. Lorsque cela fonctionne, les personnes concernées ne peuvent souvent pas utiliser efficacement et rapidement les données brutes issues de la production (essentielles pour le contrôle des coûts d'exploitation (fig. 1 (a)).

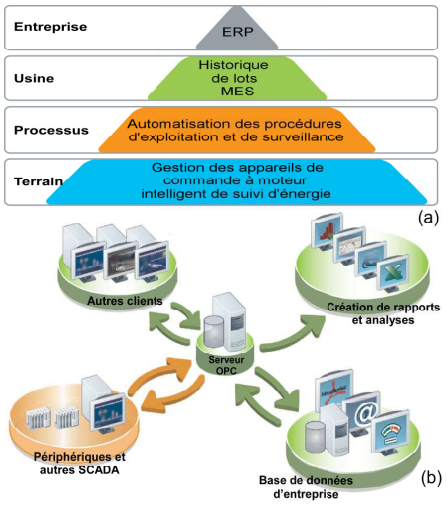

fig. 1 : Architecture pyramidale dans le Concept CIM[1]

[1] CIM : Computer Integrated Manufacturing

Ce travail tente de montrer un ensemble de verrous techniques devant être levé pour répondre d'une façon ludique au contexte soulevé dans ce préambule.
Les résultats partiels présentés dans cet article peuvent faire l'objet de travaux en étude et réalisation et/ou en travaux dirigés sur ordinateur, tablette et SmartPhone. Les postes de travail peuvent être dupliqués sans aucune difficulté. Dans le cadre de nos formations universitaires, cette « *alchimie pédagogique* » n'est aujourd'hui pas simple à mettre en œuvre, compte-tenu des heures allouées pour tels ou tels enseignements en informatique industrielle associés à des plates-formes de développement de plus en plus complexes à appréhender avant d'écrire une simple ligne de code !
Le travail présenté consiste à mettre en place une structure opérationnelle montrant les fondements d'une architecture industrielle de communication (fig. 1). Il apporte un grand nombre de complément d'information par rapport aux articles présentés lors du colloque CETSIS 2013 ([3], [6]). A partir des informations émanant du processus de fabrication et mises à disposition dans un serveur OPC[2], il faudra donc développer et utiliser plusieurs clients dans une démarche SCADA[3]. Nous montrons dans cet article plusieurs solutions ra-

[2] OPC [5] OPCFOUNDATION., https://opcfoundation.org/. : OPC est similaire à DDE (Dynamic Data Exchange) dans l'objectif de faire communiquer de façon transparente différents systèmes ou applications. Dans ses performances, OPC surclasse de loin DDE (contrôle de la qualité des échanges, gestion des erreurs de communication…) qui n'a jamais connu de véritable essor dans le monde industriel. De plus, OPC permet de gérer de façon simple des architectures réseau "*Client-Serveur*" grâce à des mécanismes natifs dans Windows 95/98/NT : OLE (Object Linking and Embedding), COM (Component Objet Model), DCOM (Distributed Component Objet Model).

[3] SCADA : Supervisory Control and Data Acquisition. L'objectif d'un environnement SCADA est de mener une conduite réactive d'un processus. Un environnement SCADA comprend 3 sous-ensembles : la fonction commande, la fonction surveillance, la fonction supervision.

pides à implémenter pour effectuer un large panel de tests. Un client final sous EXCEL permet dans un tableau multi-onglets, de suivre On line, un certain nombre de contenus de variables (Items). L'application EXCEL pourra servir d'Interface Homme Machine avec ses avantages et ses inconvénients. Ce choix fera également l'objet d'une réflexion à mener avec les étudiants sur les points tels que : stabilité de Windows, du pack Office, robustesse de la solution, sécurité… Une consultation des variables avec des objets portables (tablette et Smartphone) est également possible et présentée.

2 ENVIRONNEMENT ARCHITECTURAL

L'environnement architectural de travail est présenté fig. 2 et reprend le concept présenté fig. 1(b). Le processus est piloté par une partie commande sur laquelle est implantée une application d'automatisme répondant par exemple à une recette de fabrication (partie commande simulée). Un certain nombre de logiciels doit être installé sur l'ordinateur qui sera relié au réseau LAN[4] du département de formation (fig. 2).

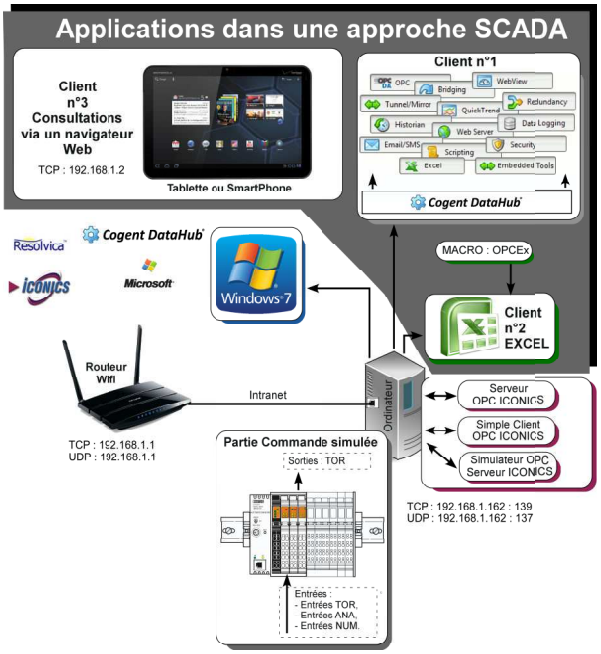

fig. 2 : Architecture du travail réalisé

Différents serveurs OPC peuvent être déployés et utilisés (CODRA Composer OPCServer, Matrikon OPC, Resolvica OPC, SV OPC...). Ils font l'objet de comparatifs dans différentes revues consacrées à l'informatique industrielle. Notre choix s'est porté sur une solution proposée par Iconics [2], société internationale qui commercialise des environnements de développement répondant aux problématiques de la fig. 1 (a) avec différents « *packages* ». Pour pouvoir lier les différents clients au serveur OPC, il faut installer le logiciel Cogent DataHub [1] et une Macro sous EXCEL. Deux orientations peuvent être abordées. La première consiste à écrire en Visual Basic sous EXCEL, une Macro capable de faire le lien avec le serveur OPC ou le logiciel Cogent DataHub. La deuxième consiste à utiliser une Macro déjà opérationnelle commercialisée ou disponible en version Beta par la société Resolvica (OPCEx) [4].

2.1 Définition des variables (Items)

Le Tableau 1 précise l'ensemble des variables (Items) qu'il faudra déclarer dans l'environnement : Simulator OPC Serveur Configuration de chez Iconics.
On compte au total, trois entrées de type TOR[5], deux entrées de type NUM[6], une entrée de type Text[7] et deux sorties de type TOR. Le projet sous l'environnement Simulator OPC Serveur étant validé, il devra faire l'objet d'une sauvegarde (fichier.mdb). Par exemple, un fichier machine1.mdb pourra être appelé pour récupérer l'intégralité des Items de l'application en cours.
Un travail peut être également mené (Import, Export, édition dans un WordPad, ouverture dans EXCEL…) concernant l'utilisation d'un fichier au format .CSV[8], standard bien connu en matière d'échange de données encapsulées dans un fichier unique. Un exemple de bribe de résultats est donné ci-dessous :

#Folders1;
"LocationPath","Name"
"\\Address Space","Numeric"
"\\Address Space","Logical"
"\\Address Space","Textual"
#Folders2;
"LocationPath","Name"
#Folders3;
"LocationPath","Name"
#DataItems4;
"LocationPath", "Name", "Domain", "DataType", "NumericSimulationSignal", "LogicalSimulationSignal", "TextualSimulationSignal", "AccessRights", "MessagePrefix", "GenerateAlarms", "LimitAlarmDefinition", "DigitalAlarmDefinition", "EUUnits", "Description", "CloseLabel", "OpenLabel", "DefDisplay", "FgColor", "BkColor", "Blink", "BMPFile", "SoundFile", "HTMLFile", "AVIFile", "Monitor"
#DataItems3;
"LocationPath", "Name", "Domain", "DataType", "NumericSimulationSignal", "LogicalSimulationSignal", "TextualSimulationSignal","AccessRights", "MessagePrefix", "GenerateAlarms", "LimitAlarmDefinition", "DigitalAlarmDefinition", "EUUnits", "Description", "CloseLabel","OpenLabel", "DefDisplay", "FgColor", "BkColor","Blink", "BMPFile", "SoundFile","HTMLFile", "AVIFile", "Monitor"
#DataItems2;
"LocationPath", "Name", "Domain", "DataType", "NumericSimulationSignal", "LogicalSimulationSignal", "TextualSimulationSignal", "AccessRights", "MessagePrefix","
…

Le Tableau 1 présente la déclaration des variables à traiter (Items) provenant du processus et gérées par la partie commande (fig. 2). La fig. 3 montre

[4] LAN : Local Area Network

[5] TOR : Tout Ou Rien
[6] NUM : Numérique
[7] Text : Alphanumérique
[8] CSV : Comma Separated Values

l'environnement permettant de déclarer l'ensemble des Items en liaison avec les variables issues de la partie commande simulée.

Description des variables à traiter (Items)						
Entrées						
Nom	Description	Type	Info initiale	Data_Type	Signal	Accès
FIN_DE_COURSE_HAUT	Fin_de_course_élévateur_en_position_haute	TOR	1 (NF)	Logical	True	Lecture-Ecriture
FIN_DE_COURSE_MEDIAN	Fin_de_course_élévateur_en_position_médiane	TOR	0 (NO)	Logical	False	Lecture-Ecriture
FIN_DE_COURSE_BAS	Fin_de_course_élévateur_en_position_basse	TOR	1 (NF)	Logical	True	Lecture-Ecriture
INFORMATION_CODEUR	Information_Codeur incrémental	NUM	0	I4	100*Random	Lecture-Ecriture
TEMP_BOITIER	Lecture_capteur_température_boitier	NUM (Interprétation / ANA)	0	R8	Ramp(T/10)	Lecture-Ecriture
CODE_ACCES_SYSTEME	Code_secret	Textual	A text		A"text"	Lecture-Ecriture
Sorties						
Nom	Description	Type	Info initiale	Data_Type	Signal	Accès
PROCESSUS_EN_RUN	Voyant_signalisation_processus_en_Run	TOR	0 (NO)	Logical	False	Lecture-Ecriture
ALARME_STOP	Alarme_processus_en_Stop	TOR	0 (NO)	Logical	Logical Square [T/10]	Lecture-Ecriture

Tableau 1 : Déclaration des entrées et des sorties

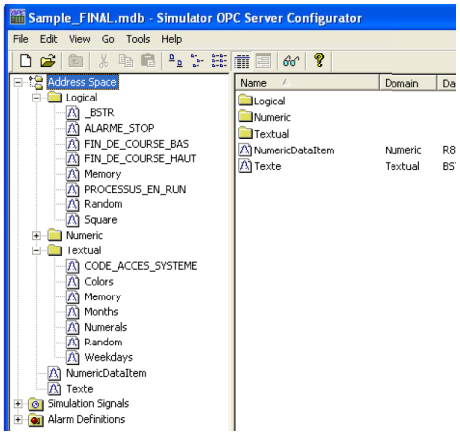

fig. 3 : Déclaration des variables (Items)

2.2 Simulation & émulation des contenus des variables (Items)

Toutes les variables étant déclarées, le premier client peut être connecté : Simple Client OPC. Comme le montre la fig. 5, l'ordinateur utilisé peut accueillir plusieurs clients OPC (CODRA, INAT, Matrikon, MATLAB… ; Tests que nous avons pu effectuer pour valider des services associés et souvent identiques).

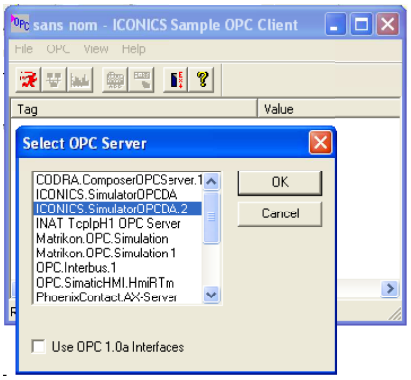

fig. 4 : Connexion au serveur OPC

Une fois le simulateur connecté, l'environnement de travail permet d'appeler l'ensemble des Items (fig. 5). Le contenu interne (value) de chaque Item est capable d'évoluer en fonction de la spécificité de cette dernière ou en fonction d'un forçage possible et parfaitement contrôlable. Cette partie du travail est la plus simple à aborder avec des étudiants.

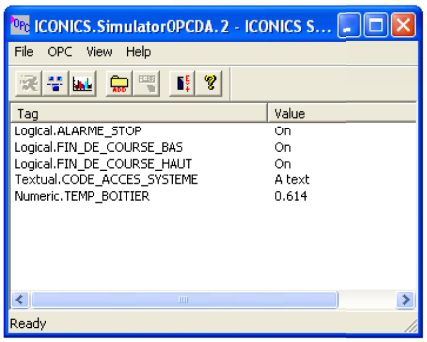

fig. 5 : Animation des variables et de leur contenu

3 DEVELOPPEMENT ET UTILISATION DES DIFFERENTS CLIENTS

L'objectif de cet article n'est pas de présenter l'ensemble des résultats issus de nos différents tests. Les résultats significatifs pour les trois clients sont présentés ci-dessous. Ils apportent de nombreux éclaircissements sur les outils disponibles sur étagère et ce que nous pouvons en faire pédagogiquement avec nos étudiants. La tâche Iconics.SimulatorOPCDA.2 doit être active pour l'ensemble des différents essais.

3.1 Client n°1

Aujourd'hui, OPC est devenu le protocole de choix dans un environnement de contrôle-commande de processus. Comme nous l'avons déjà présenté dans cet article, il existe un grand nombre de serveurs OPC permettant de disposer d'une connectivité avec différents automates programmables industriels, PC industriels… Très souvent, les fabricants de matériel d'automation proposent une suite logicielle OPC compatible avec leur gamme de produits. Cela permet ensuite aux éditeurs de logiciels de supervision de créer des applications « client OPC » pour accéder facilement à des données en temps réel provenant d'un processus de fabrication, d'un système, d'une machine… L'éditeur Cogent Real-Times Systems [7] propose une suite logicielle permettant de disposer de très nombreux services (fig. 6).

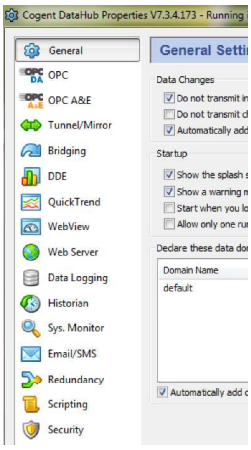

fig. 6 : Les services proposés par Cogent DataHub

L'environnement logiciel permet la connexion au serveur OPC actif. Le contenu des différents Items est récupérable de manière transparente. Les informations peuvent être archivées à la demande. La fig. 7 montre un traitement possible sur l'Item : TEMP_BOITIER.

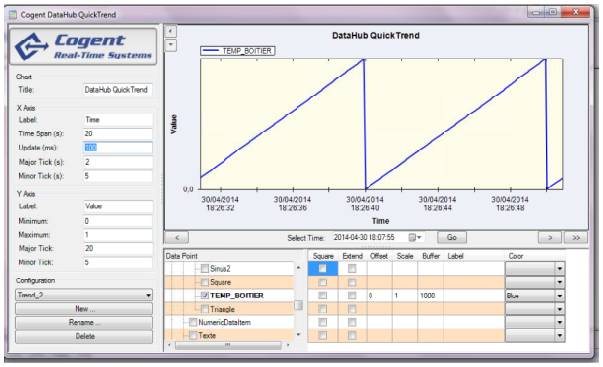

fig. 7 : Résultat avec la fonction : QuickTrend

Une autre fonctionnalité très utile est de pouvoir disposer d'un serveur Web à partir des Items actifs dans le serveur OPC. La fig. 8 présente la récupération de deux Items à partir d'un navigateur Internet (CODE_ACCES_SYSTEM (Texte) et TEMP_BOITIER (NumericDataItem)).

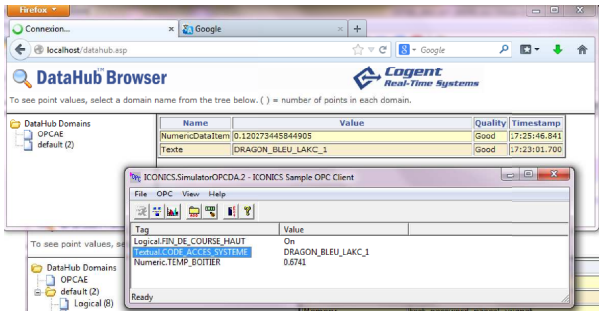

fig. 8 : Résultat avec la fonction : Serveur Web

L'environnement permet également de créer une base de mail ou de SMS en fonction des valeurs possibles pour l'ensemble des Items. Un message via une boite mail et via un SMS est acheminé si par exemple l'Item CODE_ACCES_SYSTEME prend la valeur : DRAGON_BLEU_LAKC_1.

3.2 Client n°2

L'objectif de cette deuxième partie consiste à faire fonctionner une application EXCEL permettant d'animer, de tester, de traiter les Items et leur contenu.
Le développement de l'application finale respecte l'architecture de la fig. 9. Des boutons créés avec des Macros permettent l'accès aux différents onglets.
La MACRO OPCEx3 étant installée, nous pouvons à l'aide du bouton : Add or Edit OPC Item, déclarer le serveur OPC actif et rajouter à la demande, des Items dans un onglet et donc une cellule en particulier du tableur (fig. 10).

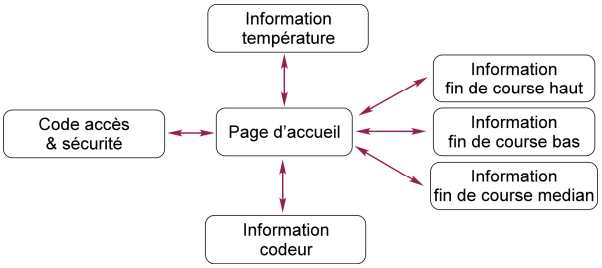

fig. 9 : Organisation des onglets et de leurs liens

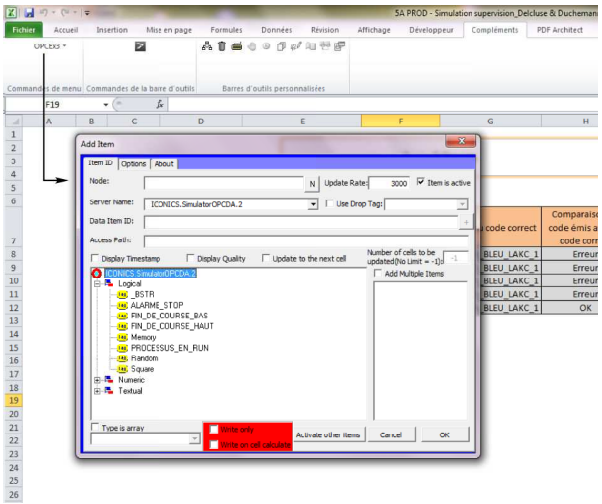

fig. 10 : Environnement EXCEL avec OPCEx3 (Macro)

Cette partie montre que l'environnement EXCEL peut répondre à de nombreuses fonctions méconnues par les étudiants. Chaque Item de type TOR est associé à son contenu et devra faire l'objet de traitements particuliers dans la page rattachée à l'onglet (fig. 12). Un travail complémentaire peut être rajouté sur un Item relevant d'une variable TOR, comme par exemple, une informa-

tion concernant le terme TRS[9]. Ce calcul peut être effectué, connaissant le temps d'ouverture de la machine, le temps de présence de l'opérateur, l'information RUN/STOP (processus), le tout horodaté. La fig. 11 présente la page d'accueil créée dans le tableur.

fig. 11 : Page d'accueil respectant les obligations (fig. 9)

Retour information fin de course haut

Heure	Horadatage	Etat de fin de course haut	Rafraichissement	Bool	Compteur position haute
15:24:40	1/05/2014 15:24:40	VRAI	GOOD	1	5
15:30:00	1/05/2014 15:30:00	FAUX	GOOD	0	
15:30:08	1/05/2014 15:30:08	VRAI	GOOD	1	
15:30:12	1/05/2014 15:30:12	FAUX	GOOD	0	
15:30:14	1/05/2014 15:30:14	VRAI	GOOD	1	
15:30:18	1/05/2014 15:30:18	FAUX	GOOD	0	
15:30:22	1/05/2014 15:30:22	VRAI	GOOD	1	
15:30:26	1/05/2014 15:30:26	FAUX	GOOD	0	
15:30:30	1/05/2014 15:30:30	VRAI	GOOD	1	
15:30:34	1/05/2014 15:30:34	FAUX	GOOD	0	

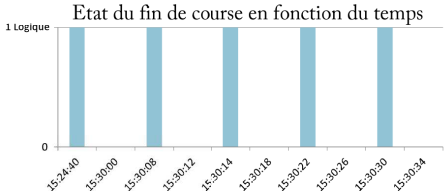

fig. 12 : Résultat concernant l'Item : ETAT_FIN_DE_COURSE_HAUT

Le traitement du CODE_ACCES_SYSTEME nécessite un peu plus de traitement sous EXCEL (fig. 13). En effet, en ayant fixé au préalable un code connu et arbitrairement choisi (DRAGON_BLEU_LACK_1), les cellules de la colonne (Comparaison du code…) permettront d'affirmer avec un test affiché, que le code saisi dans l'Item : CODE_ACCES_SYSTEME est correct ou non (VRAI, FAUX). Ce code peut être changé en fonction des préconisations connues et utilisées en matière d'usage de code secret ([8]). Par exemple, la phrase « Elle a deux vélos », devient « LA2VLO » : nouveau code. Le résultat du test VRAI ou FAUX sera issu d'une opération : =EXACT(C5;E5). Le résultat d'une cellule associée à la colonne (Ouverture de la liaison…) sera le résultat d'un test :
Gx=SI(ESTERREUR(TROUVE("FAUX";F5)); "Code OK, ouverture système";"A ressaisir"). Les couleurs associées au texte peuvent faire également l'objet d'un traitement automatique à partir des déclarations de fonctions spéciales.

[9] TRS : Taux de Rendement Synthétique

Accès & sécurité

Heure	Horodatage	Code accès système	Rafraichissement	Valeur du code correct	Comparaison du code émis avec le code correct	Ouverture de la liaison de communication
15:21:26	1/5/2014 15:21:26	A text	GOOD	DRAGON_BLEU_LAKC_1	Erreur	A ressaisir
15:22:15	1/5/2014 15:22:15	LrDJeNou	GOOD	DRAGON_BLEU_LAKC_1	Erreur	A ressaisir
15:22:47	1/5/2014 15:22:47	Kc55l	GOOD	DRAGON_BLEU_LAKC_1	Erreur	A ressaisir
15:23:25	1/5/2014 15:23:25	Bblack	GOOD	DRAGON_BLEU_LAKC_1	Erreur	A ressaisir
15:24:26	1/5/2014 15:24:26	DRAGON_BLEU_LAKC_1	GOOD	DRAGON_BLEU_LAKC_1	OK	Code OK, ouverture système

fig. 13 : Résultat concernant l'Item : CODE_ACCES_SYSTEME

Le traitement concernant l'Item : TEMP_BOITIER a fait également l'objet de traitements particuliers sous EXCEL (fig. 14). A l'issue d'une mise à l'échelle, la colonne (Seuil de détection d'alarme), devra faire l'objet d'un test utilisant la fonction SI. Les résultats affichés de la colonne (Seuil de détection d'alarme BINAIRE), devront avoir subi un traitement préalable avec la fonction :
Fx=SI(ESTERREUR(TROUVE("OK";F5));"1";"0").
Une colonne identique à la précédente est rajoutée en ne recopiant que la valeur numérique affichée en vue de tracer cette fonction. Dans cette partie de travail, la notion de vumètre est abordée affichant des résultats « *au fil de l'eau* ». On retrouve une fonctionnalité vue précédemment (fig. 7).

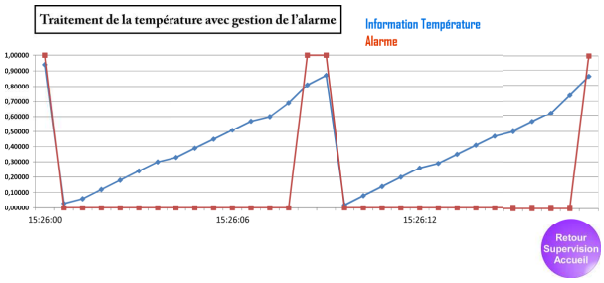

fig. 14 : Résultat concernant l'Item : TEMP_BOITIER

3.3 Client n°3

Ce dernier client présenté est dans l'actualité depuis quelques mois avec de nombreux services appelés « ScadaMobile ». Effectivement, la demande des industriels est importante afin de pouvoir disposer de nombreuses informations via leurs technologies portables (tablette, Smartphone). La fig. 2 montre la mise en place d'un routeur Wifi. Cette fonction administrable permet de rediriger l'adresse IP de l'ordinateur et donc des services potentiellement accessibles. Les auteurs de l'article [3] présentent une application hébergée sur un Smartphone et capable de piloter et contrôler certains dispositifs électriques d'une maison à partir d'un automate programmable industriel.

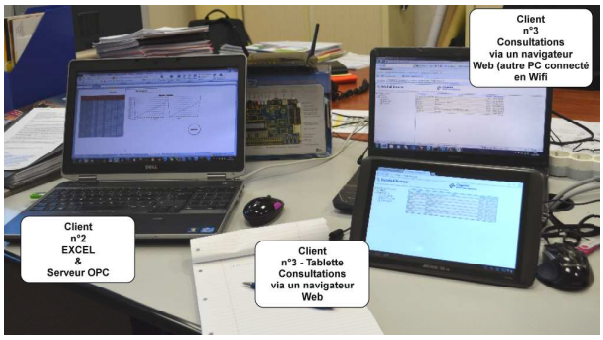

photo. 1 : Poste de tests

Dans notre cas, la tablette est capable de récupérer dynamiquement l'intégralité des Items et de leur contenu via une connexion Web (photo. 1). Nous pouvons retrouver sur la tablette, les mêmes fonctionnalités présentées dans la fig. 8. Le client n°1 est également capable de construire une base de données. Un développement sous Ajax[10], peut permettre de créer une application Web sur mesure pour le poste client n°3.

4 RETOUR SUR EXPERIENCE DANS LE CADRE DE CE TRAVAIL AVEC DES ETUDIANTS

Ce travail est mené avec des étudiants de 2ème année DUT[11] (Génie Electrique et Informatique Industrielle) dans un module nommé « Supervision ». La stratégie pédagogique consiste à transférer la partie travaux dirigés (temps alloué : 7h30) vers un travail individuel sur machine. Le retour est très positif car les étudiants peuvent travailler à leur rythme puisque la salle d'informatique est en accès libre. Ils découvrent des fonctionnalités complétant des compétences qu'ils utiliseront ensuite dans la partie travaux pratiques consacrée au module. Cette partie intègre cette fois-ci, du matériel d'automation avec des connexions en Full-IP. Une sensibilisation aux problématiques liées aux environnements SCADA vient d'être menée cette année, sur un module d'enseignement auprès d'étudiants en dernière année d'école d'ingénieur par apprentissage (Réseau POLYTECH : Université d'Orléans, spécialité productique). Les étudiants ont pu travailler sur leur ordinateur personnel dans la mesure où les environnements logiciels sont gratuits, ou limités dans leur utilisation (redémarrer les services après 1 heure de connexions). La tâche d'enseignement pouvait sembler a priori ardue, puisque ces étudiants n'ont pas de prédispositions particulières pour l'informatique industrielle. Le travail a été mené de manière très correcte après environ 8 heures d'activité. Les étudiants ont été très satisfaits par cette sensibilisation. En effet, ils ont pu « toucher du doigt », la nécessité obligatoire aujourd'hui de faire remonter des informations issues du terrain. Ce travail leur a permis d'imaginer des applications concrètes à mettre en place dans leur futur quotidien professionnel en production.

[10] Ajax : Asynchronous JavaScript and XML
[11] DUT : Diplôme Universitaire de Technologie

5 CONCLUSION ET PERSPECTIVES

Cet article tente de présenter d'une manière ludique, non coûteuse et efficace, différents aspects fondamentaux utilisés depuis quelques années sur les environnements SCADA. Cette première approche à vocation pédagogique dispose de plusieurs avantages : montrer et démontrer les spécificités d'un client-serveur OPC ; montrer que le logiciel EXCEL peut réaliser assez simplement des animations et donc une possible application d'Interface Homme-Machine. L'interopérabilité entre les différents matériels nous a également permis de valider un certain nombre de services avec une technologie tablette ou Smartphone. Les perspectives de travaux futurs sont nombreuses : greffer un système réel à notre démarche SCADA, développer des applications spécifiques pour tablette ou Smartphone.

Bibliographie

[1] COGENT DATAHUB., *http://www.cogentdatahub.com/*.

[2] ICONICS., *http://www.iconics.com/*.

[3] Jean-Paul CHEMLA and Bernard RIERA., *Programmation objet pour une domotique réalisée avec un API*, Actes du 10ème Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes (CESTIS-EEA'13), Caen (2013), pp. 78-80.

[4] OPCEX., *http://www.resolvica.com/opcexoverview.htm*.

[5] OPCFOUNDATION., https://opcfoundation.org/.

[6] Pascal VRIGNAT, Florent DUCULTY, Jean-François MILLET, Stéphane BEGOT and Manuel AVILA., *Solution Bluetooth : utilisation d'une architecture logicielle dans le cadre d'une mise en place d'une solution multi-clients à partir d'un serveur OPC*, Actes du 10ème Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes (CESTIS-EEA'13), Caen (2013), pp. 210-215.

[7] C. R.T. SYSTEMS, http://www.cogentdatahub.com/index.html.

[8] Pascal VRIGNAT., *Contrôle Commande et Supervision Son évolution : ses enjeux, les nouvelles tendances et nouveaux services*, Rapport de Master, Université d'Orléans, 2007.