

Département Bâtiment – BAT4

Projet : Immotique - Séance 1 : Définition du Projet - Rappel et synthèse des acquis avant projet

Jean-Yves Tigli – tigli@polytech.unice.fr

Durée : 3h

1. Introduction au Projet :

Ce projet a pour objectif d'illustrer les possibilités offertes par l'informatique pour surveiller un bâtiment et faire des traitements automatiques à partir :

- d'informations collectées dans le bâtiment
- d'informations collectées auprès de fournisseurs de données ouvertes (open data)

Les traitements peuvent alors utiliser des outils adaptés avec lesquels des non informaticiens ont l'habitude d'interagir. Citons par exemple :

- Matlab / Scilab
- Excel
- Base de données (Access, MySQL ..)
- etc.

Les cours suivis en Informatique durant l'année BAT4 vous ont préparé à ce type de Projet. Nous faisons néanmoins quelques rappels techniques ci-dessous, en particulier sur l'utilisation d'Excel (pour le traitement de données) et de Service Web (pour la récupération de données) depuis une interface graphique C#.

Cependant, en premier lieu, il convient d'établir le sujet de projet sur lequel vous comptez travailler.

Voici donc différents défis qui vous sont proposés, à vous de choisir le votre et de le traiter avec votre propre approche.

1. Amélioration de la Gestion énergétique d'un bâtiment grâce à la collecte d'information en temps réel
2. Amélioration de la Gestion des fluides d'un bâtiment grâce à la collecte d'information en temps réel
3. Suivi du vieillissement d'un bâtiment grâce à la collecte d'information en temps réel
4. Systèmes d'Alerte pour la prévention des catastrophes naturelles
5. etc.

A titre d'exemple :

L'intégration des prévisions météorologiques dans la gestion d'un bâtiment permettrait :

- de gérer l'arrosage automatique
- de gérer la marche, l'arrêt, et le niveau de climatisation en prenant en compte l'inertie du système de climatisation.
- de gérer la marche, l'arrêt, et le niveau de chauffage en prenant en compte l'inertie du système de chauffage.

La collecte d'informations capteur pour la gestion d'un bâtiment permettrait :

- de mettre en place un système d'alerte pour des consommations anormales en fluides grâce à des capteurs de mesure de flux (électriques, hydrauliques, ...).
- de mettre en place un système de détection de mouvements sismiques grâce à des capteurs de mouvement et vibration

Livrables 1 et 2 du projet (à rendre à l'issu de la première séance) :

Chaque étudiant travaille sur son projet. Plusieurs projets peuvent appartenir à un même défi mais chacun doit apporter son lot de spécificités et d'originalités.

La connaissance du domaine du bâtiment étant au cœur de votre métier.

La capacité à mettre en oeuvre cette connaissance doit faire partie de vos compétences professionnelles.

C'est pour cette raison que la définition même de votre projet pour répondre à un des défis présentés ci-dessus, sera évalué en premier lieu.

Pour cela vous devez, durant cette première séance de projet :

1. Rédigez un texte d'une page maximum détaillant le défi que vous relevez et le projet que vous proposez (Il s'agit des spécifications du projet):
 - Les objectifs du projet et ce qu'il apporte de spécifique en réponse au défi.
 - Les données que vous aller récupérer et traiter
 - Les traitements que vous allez appliquer à ces données (y compris des équations mathématiques s'il y a lieu)
 - Les données que vous aller produire
2. Rédigez un texte d'une page maximum détaillant le défi que vous relevez et le projet que vous proposez (Il s'agit des spécifications logicielles du projet):
 - Les capteurs et/ou bibliothèques logicielles et/ou Services Web que vous allez utiliser pour récupérer des données. Une partie de ces données comme celles qui devraient venir de capteurs du bâtiment, pourront être pré-stockées dans un fichier Excel qui fera donc office de DataCenter. Vous devrez néanmoins fournir des données réalistes en fournissant les références et caractéristiques du ou des capteurs utilisés. Le contenu du fichier Excel respectera les caractéristiques de ces données.
 - Le format des données que vous aller récupérer et traiter
 - Les bibliothèques logicielles que vous allez utiliser pour traiter ces données (si vous voulez bénéficier des rappels techniques ci-dessous, alors choisissez Excel)
 - Les formats des données que vous aller produire et/ou actionneur que vous allez mettre en oeuvre
 - Les fichiers et/ou bibliothèques logicielles et/ou Services Web que vous allez utiliser pour visualiser et/ou stocker les données produites.
 - La description de l'interface graphique que vous allez mettre en place avec votre programme sous la forme d'un mockup (vous pouvez faire un dessin à la main ! Le tout est de prévoir les widgets comme les boutons, les textboxes, les labels, ..., que vous allez mettre dans votre programme)

L'organisation générale du projet sera donc la suivante :

séance 1 : définition et spécifications du projet	séance 2 : Correction des spécifications et Développement du projet	séance 3 : Développement du projet et rendu
--	--	--

2. Rappel Environnement de développement d'une interface graphique

Un projet C# .Net qui permet de créer une interface graphique est un projet "Application Windows Forms".

Le développement d'un tel projet se fait en trois temps :

1. Instanciation et placement des widgets (comme boutons, textboxes, ...) avec la souris (drag and drop) depuis la boîte à outils vers la fenêtre du projet.
2. Les propriétés initialisées par défaut pour chaque widget sont modifiables pour changer leur valeur initiale.
3. Le comportement de l'application graphique se développe au travers l'écriture du contenu des événements des widgets, par exemple :

```
private void button1_Click(object sender, EventArgs e)
{
    TextBox1.Text = ("Bonjour");
}
```

Un développement interactif d'Interface Graphique pourra illustrer ces points en cours.

3. Rappel : Créer un fichier Excel en C# sous .Net

Nous allons créer un projet "Application Windows Forms" nommé *CreationFichierExcel* pour créer un fichier Excel depuis une programme en C# sous .Net

Nous devons pour cela utiliser la référence à une des bibliothèques "COM" : Microsoft Excel 12.0 Object Library dans le projet.

Ainsi nous pourrons utiliser des classes qui nous permettrons de créer et manipuler une application Excel, en particulier :

- L'Application : **Excel.Application** avec par exemple *Excel.Application xlApp* ;
- Le Classeur : **Excel.Workbook** avec par exemple *Excel.Workbook xlWorkBook* ;
- Les Feuilles : **Excel.Worksheet** avec par exemple *Excel.Worksheet xlWorkSheet* ;
- Les Cellules : avec par exemple *xlWorkSheet.Cells*;

Ajoutons maintenant un bouton que l'on nommera "Creation" et qui affichera une étiquette "Création d'un fichier Excel".

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;

namespace CreationFichierExcel
{
```

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void Creation_Click(object sender, EventArgs e)
    {
        Excel.Application xlApp ;
        Excel.Workbook xlWorkBook ;
        Excel.Worksheet xlWorkSheet ;

        xlApp = new Excel.Application();
        xlWorkBook = xlApp.Workbooks.Add();

        xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
        xlWorkSheet.Cells[1, 1] = "8"; // ligne 1 colonne A
        xlWorkSheet.Cells[1, 2] = "6"; // ligne 1 colonne B
        xlWorkSheet.Cells[1, 3] = "=A1+B1";

        xlWorkBook.SaveAs("c:\\csharp-Excel.xls", Excel.XlFileFormat.xlWorkbookNormal);
        xlWorkBook.Close(true);
        xlApp.Quit();

        MessageBox.Show("Le fichier c:\\csharp-Excel.xls vient d'être créé");
    }
}

```

4. Rappel : Lire un fichier Excel en C# sous .Net

Nous allons créer un projet "Application Windows Forms" nommé *LireFichierExcel* pour lire un fichier Excel depuis une programme en C# sous .Net

Nous devons pour cela utiliser la référence à une des bibliothèques "COM" : Microsoft Excel 12.0 Object Library dans le projet.

Ajoutons maintenant un bouton que l'on nommera "LectureFichier" et qui affichera une étiquette "Lecture d'un fichier Excel".

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;

namespace LireFichierExcel
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

```

```
    }  
  
    private void LectureFichier_Click(object sender, EventArgs e)  
    {  
        Excel.Application xlApp ;  
        Excel.Workbook xlWorkBook ;  
        Excel.Worksheet xlWorkSheet ;  
        Excel.Range range ;  
  
        double str;  
        int rCnt = 0;  
        int cCnt = 0;  
  
        xlApp = new Excel.Application();  
        xlWorkBook = xlApp.Workbooks.Open("c:\\csharp-Excel.xls");  
        xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);  
  
        range = xlWorkSheet.UsedRange;  
  
        for (rCnt = 1; rCnt <= range.Rows.Count; rCnt++)  
        {  
            for (cCnt = 1; cCnt <= range.Columns.Count; cCnt++)  
            {  
                str = (double)(range.Cells[rCnt, cCnt] as Excel.Range).Value2 ;  
                MessageBox.Show(Convert.ToString(str));  
            }  
        }  
  
        xlWorkBook.Close(true, null, null);  
        xlApp.Quit();  
    }  
}
```

5. Rappel : Consommation d'un Service Web sous C#

Catalogues de Services Web :

- <http://www.websvicex.net>
- <http://www.xmethods.com>
-

Si nous prenons comme exemple les données et prévisions météorologiques, nous pouvons trouver nombre de fournisseur de données, américaines, européennes, françaises, gratuites, payantes sur abonnement ... accessibles via des services Web.

Par exemple :

PREVIMETEO : <http://www.previmeteo.com/services-pro/xml-web-services.php>

METEOMEDIA : http://media.meteomedia.com/web_datafeeds.php

6. Mise en œuvre de Matlab sous C#

Vous pourrez trouver un exemple sur <http://www.codeproject.com/Articles/5468/ways-of-integrating-MATLAB-with-the-NET>, solution 1.

Comme pour Excel, l'utilisation de Matlab passe par le chargement d'une bibliothèque "COM" dans le projet, comme "Matlab Application (Version 7.10) Type Library".

En premier lieu, il faut créer un objet qui permet d'accéder à une application Matlab comme pour Excel, puis utiliser l'API (les méthodes) proposées par la bibliothèque chargée.