

SLR202 – Micro-projet Smart Institute
Modélisation pour un bâtiment intelligent
Sylvie Vignes – Etienne Borde

Il s'agit de modéliser le système de gestion d'un bâtiment intelligent, c'est à dire d'un bâtiment éco-énergétique et confortable qui conviendrait à un établissement d'enseignement supérieur. Le sujet s'inspire de façon très libre des idées de l'architecture du centre de recherche EDF Lab Paris-Saclay¹ et de scénarios de système autonome domotique². Le cahier des charges ci-après est imaginé. L'objectif du projet est uniquement pédagogique et n'incite pas à la contrefaçon; il vise à évaluer votre capacité à modéliser un système informatique.

Ce n'est pas une modélisation exhaustive du système de gestion qui est demandée; De l'ensemble des éléments décrits dans le cahier des charges il faut dégager une bonne compréhension de l'objectif global pour concevoir le système logiciel. Le travail de modélisation est guidé par la démarche itérative présentée en cours ; cela vous laisse des choix de modélisation structurelle ou de modélisation comportementale, à justifier selon des critères tels que : extensibilité du modèle, séparation des préoccupations, maintenabilité ... qui sont concrètement explicités.

I - Cahier des charges

On suppose qu'un architecte a conçu un bâtiment esthétique et fonctionnel, organisant sur plusieurs niveaux, diverses salles: amphithéâtres, salles de cours, salles de travaux en groupe, de salles de TP, salles de réunion, locaux techniques, laboratoires, centre documentaire, bureaux individuels ou partagés pour le personnel, et un grand Forum. Ce bâtiment doit satisfaire des exigences qui respectent des recommandations éco-énergétiques et des exigences qui concernent le confort de ses occupants.

D'un point de vue énergétique, les salles du bâtiment sont chauffées ou rafraîchies ce qui consomme de l'énergie, mais le système prend en compte d'une part l'énergie du flux solaire qui apporte un réchauffement naturel et d'autre part, les transferts d'énergie entre les salles voisines. Le confort est perçu par un occupant par la température, le taux d'humidité, la qualité de l'air et la luminosité de la pièce où il se trouve. Le bâtiment doit aussi respecter des normes de sécurité Incendie : la détection de fumée dans une pièce doit déclencher une alarme dans toutes les salles.

L'étude de ces exigences a permis de dégager les grandes lignes d'une infrastructure logicielle. Vous devrez identifier et analyser les fonctionnalités attendues par les différents acteurs pour concevoir le système de gestion du bâtiment.

L'infrastructure logicielle est composée d'un système de pilotage global de l'éco-énergie relié à des systèmes répartis dans les salles qui assurent la régulation locale du confort.

¹ <https://pulse.edf.com/fr/saclay-un-batiment-paysage>

² Article disponible sur la page <http://adadiaconescu.thereyouare.com/publications.html>

S. Frey, F. Huguet, C. Miville, David Menga, A. Diaconescu and I. Demeure

"Scenarios for an Autonomic Micro Smart Grid", 1st International Conference on Smart Grids and Green IT Systems (Smart Greens2012), Porto, Portugal, 19-20 April 2012.

Le système de pilotage global a un rôle décisionnel essentiellement sur l'aspect énergétique: il optimise la consommation électrique globale du bâtiment en fonction de la politique tarifaire de son fournisseur (tarifs forts en heures de pointe) et gère les conflits entre toutes les demandes de consommation. Un algorithme décide de la consommation autorisée par chaque salle en prenant en compte : 1) la prédiction de consommation électrique de chaque salle pour la climatisation, la ventilation et l'éclairage ; 2) la réaction thermique de l'habitat en prenant en compte le flux solaire (provenant d'une station météo extérieure) et le transfert d'énergie entre les salles voisines d'après un modèle de thermodynamique ; 3) sa capacité d'apprentissage sur l'ambiance du bâtiment à partir d'observations de l'historique des régulations locales dans chaque salle. C'est un administrateur qui met en marche/arrêt le système et le configure selon des règles propres à chaque type de salle. Lorsqu'une salle est fermée (régulation arrêtée) le système prend en compte la modification de configuration, de même lorsqu'elle est remise à disposition.

Chaque salle dispose donc d'une borne, équipement intelligent avec une IHM minimale, reliée à des capteurs et des actionneurs. Dans chaque salle, il y a des capteurs de présence d'occupant(s), de température, de luminosité, de qualité de l'air, de fumée. Il y a aussi des dispositifs matériels que la borne peut commander et dont elle reçoit des mesures de consommation énergétique : une climatisation pour chauffer ou rafraîchir, une ventilation pour ajuster la qualité de l'air, un éclairage, des volets sur les fenêtres (consommation négligeable non fournie à la borne). Ces volets peuvent servir à maîtriser l'ensoleillement de la pièce mais la luminosité doit rester acceptable.

La borne assure une régulation locale, un compromis entre la consommation énergétique et le confort des occupants qui peuvent interagir à travers l'IHM: elle reçoit la consigne de consommation du système global, prend en compte la demande de confort d'occupant(s) éventuels, agit sur des matériels et effectue un calcul pour envoyer sa prédiction de consommation.

Tous les occupants du bâtiment sont identifiés, ont un badge et appartiennent à une catégorie : enseignants, élèves, personnel administratif, personnel de maintenance, visiteur. Les occupants permanents ont la possibilité d'enregistrer le profil de leurs préférences pour la température, le taux d'humidité et la luminosité. En s'identifiant, ils peuvent interagir via les bornes selon leur catégorie et selon le type de salle. En s'identifiant, tout occupant a une autre possibilité de réglage simple : agir sur un bouton pour obtenir plus ou moins un ou deux degrés de température, à partir de l'état dans lequel il trouve la pièce. Exemple de règles : un visiteur a un badge mais pas de profil. Un étudiant ne peut régler que les salles de travail en groupe. Les enseignants et le personnel administratif peuvent régler toutes les salles sauf le Forum qui n'est réglable que par le personnel de maintenance. Seul le personnel de maintenance a le droit de marche/arrêt sur la régulation d'une salle.

Quand une salle est inoccupée, l'éclairage est coupé automatiquement et la température est baissée; Dès que la présence d'occupant(s) est détectée, l'éclairage et la consigne de température sont rétablis dans l'état précédent. Mais la ventilation reste toujours en fonctionnement. En cas de défaillance de son matériel, une salle peut se mettre automatiquement en arrêt. Certaines salles ont des caractéristiques configurées par

l'administrateur non modifiables par les occupants : par exemple, la température d'un local technique est toujours maintenue à 16°.

II - Travail à réaliser

Conseils et rappel sur la méthode de modélisation

Pour ce sujet, le diagramme de domaine initial est simple ; il représente l'infrastructure logicielle. Pour l'enrichir, suivez la méthode sur plusieurs itérations. La première itération doit aboutir à un modèle avec un « bon » niveau d'abstraction décrivant le système sans introduire les détails. Cette itération 1 doit aboutir à une modélisation structurelle de l'essentiel, autrement dit aboutir à un diagramme de classes de conception représentatif de la finalité du système. Ensuite à chaque itération, les diagrammes élaborés aux différentes étapes s'enrichissent en assurant la traçabilité entre les besoins fonctionnels. Pensez à utiliser les Design Patterns pour introduire les détails ! En appliquant la méthode, le modèle UML devra être organisé en 3 couches Modèle, Vues, Contrôleurs (MVC).

Le modèle UML final doit être clairement structuré, facilement extensible et maintenable.

A- La démarche à suivre pour la première itération, comporte 4 étapes:

1. Répertorier les acteurs et les principales fonctionnalités attendues pour modéliser les diagrammes de cas d'utilisation correspondants. On commencera par modéliser les diagrammes correspondant à la finalité opérationnelle du site et les rôles des acteurs. Finalement on en déduira le diagramme de contexte (Use Case général).
2. Modéliser le domaine d'application par un diagramme de classes (avec les attributs indispensables) modélisant les concepts essentiels ;
3. Modéliser quelques diagrammes de séquence système (DSS) correspondant aux cas d'utilisation de première importance ; on y retrouvera les principaux messages entre les acteurs et le système (cf. point 1)
4. Modéliser un premier diagramme de classes d'analyse à partir des diagrammes précédents en introduisant les stéréotypes prédéfinis (*boundary, control, entity*), ce qui revient à présenter le modèle en couche MVC. Ce diagramme doit encore rester simple en introduisant les principales classes de contrôle et les principales classes d'interface du système. Ces principales classes seront détaillées au § C.

B- La seconde itération concerne l'enrichissement du diagramme de classes.

Tout d'abord, repérer les classes ayant un comportement à état et associer à chacune une machine à état simple dans le but d'identifier ses opérations. Ensuite la démarche consiste à reprendre les points 1, 2, et 3 de l'itération précédente pour les détailler. Il y a de très nombreuses opportunités de Design Patterns (DP) à découvrir dans le sujet pour enrichir le Modèle. Trouvez au moins une opportunité de DP pour chaque grande catégorie : construction, structuration, comportement. Inclure un DP au modèle doit contribuer à améliorer le Modèle pour le rendre plus facilement extensible, maintenable malgré l'ajout de détails. Vous indiquerez clairement le problème que vous modélisez, le DP que vous choisirez, et présenterez son application (c'est à dire son adaptation et son intégration dans le diagramme de classe de conception de *SmartInstitute*).

Ainsi vous vous focaliserez sur quelques services attendus essentiels. Vous raffinerez les cas d'utilisation et les diagrammes de séquence associés (cela permet de bien comprendre l'usage d'un DP), vous écrirez les scénarii d'utilisation (cas nominaux, alternatifs et erronés). Au final vous aurez enrichi le diagramme de classes produit au §A2, structurant les « entités » que l'on retrouve au §A4 dans la couche Modèle, de façon à construire le diagramme de classes de conception, en cohérence avec les services attendus et les états de classe.

C - La troisième itération reprend le point 4 de la première itération pour développer les Contrôleurs (ou classes de contrôle) et Vues (classes d'interface) du modèle MVC sur des exemples précis et ainsi compléter le diagramme de classes de conception. Les exemples donnés ci-après sont indicatifs, vous pouvez choisir de modéliser une autre situation. Identifiez les Design Patterns qui permettent de répondre aux caractéristiques suivantes et adaptez les :

1. entre Vue et Modèle : Dès que le Modèle est modifié, on veut que la Vue soit modifiée. Par exemple, lorsque salle s'arrête automatiquement pour cause de défaillance, la configuration du système est mise à jour, ce qui est visualisé sur l'IHM de l'administrateur.
2. entre Contrôleur et Modèle : L'état du Modèle a un impact sur le Contrôleur. Par exemple, lorsqu'un occupant agit directement sur un bouton de réglage local : plus ou moins un ou deux degrés de température, à partir de l'état de dans lequel il trouve la pièce. Cela doit être répercuté sur la borne via son propre contrôleur.
3. entre Vue et Contrôleur : Les interactions de l'acteur avec le système (dans une Vue) modifient le Modèle. Le Contrôleur associé à la vue réagit aux événements extérieurs (clics sur des boutons, présentation d'un badge d'identification...) et modifie le Modèle en réponse à ces événements. Par exemple un occupant identifié demande le réglage d'une salle selon son profil de préférence. Autre situation, la manière dont un Contrôleur agit sur le modèle (son comportement) dépend de la Vue à laquelle il est associé. Par exemple, on décide de développer une application mobile qui permette d'interagir avec la borne sans avoir à se déplacer vers l'IHM. En tant que concepteur, vous souhaitez éviter les instructions conditionnelles lors du développement du contrôleur ...

Voici des exemples de caractéristiques qui doivent influencer la conception de votre modèle :

- Le sujet ne donne -volontairement- aucun détail sur les différents modèles de calcul de consommation. Le modèle devra permettre qu'ils soient facilement modifiables.
- L'implémentation de l'algorithme de décision du système central doit aussi pouvoir évoluer pour intégrer les résultats de recherche dans le domaine des systèmes autonomiques. Il faut prévoir d'affiner les règles du système lors de la mise en œuvre pour une meilleure performance. Par exemple :
 - On fera facilement évoluer les droits de réglage de salles. Soit selon la catégorie des occupants : par exemple les élèves auront les mêmes droits que les enseignants ; ou encore quand une salle est inoccupée, on fermera la ventilation.

- Les occupants ont tout d'abord un unique profil de préférences, puis on souhaite que ce profil soit différent selon l'hiver et l'été, puis selon les saisons, puis finalement selon les mois.
- L'algorithme du système central d'apprentissage se fait sur l'historique des observations. On souhaite recouper cet historique les heures de présence du personnel, connues par un autre système d'accès du personnel.

III - Travail à rendre:

Dans un zip à déposer sur le site pédagogique :

- Le modèle développé sur StarUML-V2
- Un rapport retraçant la démarche en 3 itérations intégrant les diagrammes, justifiant au fur et à mesure la conception du modèle.