

Préambule

Pour des raisons pratiques, les fonctionnalités du système présenté ci-après ont été grandement simplifiées pour se concentrer sur des problèmes de conception bien localisés. Le modèle d'analyse qui est fourni est donc uniquement un modèle préliminaire permettant d'aborder quelques problèmes classiques de conception.

Présentation du système

Le système étudié permet de gérer la sécurité d'un ou plusieurs bâtiments. Les principales caractéristiques demandées par la maîtrise d'ouvrage sont les suivantes :

- 1- Contrôle des accès au bâtiment ou à des parties du bâtiment via l'utilisation de cartes magnétiques
- 2- Contrôle des hausses de température pour détecter tout incendie
- 3- Contrôle des portes et fenêtres pour détecter toute intrusion
- 4- Indicateurs (état des zones et des détecteurs notamment) de chaque bâtiment visibles et configurables via une (ou plusieurs) console d'administration mettant en œuvre une IHM graphique et intuitive
 - a. Une ou plusieurs consoles centrales permettent de configurer l'ensemble du système et de voir en temps réel l'état des indicateurs du bâtiment
 - b. Il est possible de visualiser l'état des indicateurs du système à un instant t en se connectant via Internet

La configuration d'un bâtiment doit suivre les exigences suivantes (cf. Figure 1):

- 1- Un bâtiment est composé de zones de sécurité
- 2- Pour chaque zone de sécurité, le système permet de définir les personnes pouvant accéder à cette zone
- 3- L'accès sécurisé à une zone se fait via un point d'accès qui intègre un lecteur de carte magnétique
- 4- Chaque point d'accès est compris dans une porte. En conséquence, chaque porte peut contenir un ou deux points d'accès selon qu'elle est unidirectionnelle ou bidirectionnelle.
- 5- Chaque porte peut contenir un détecteur d'intrusion.
- 6- Le système archive les dates de passage d'une personne autorisée dans une zone.
- 7- Chaque zone peut posséder un nombre variable de détecteur de fumée et d'incendie
- 8- Toute alarme est notifiée en temps-réel aux consoles d'administration, et archivée par le système (type d'occurrence et date d'apparition)

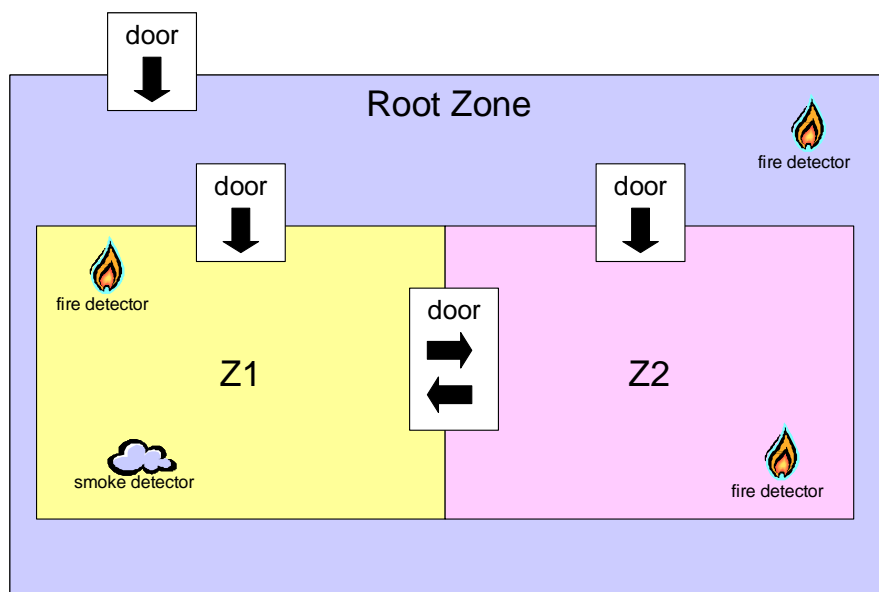


Figure 1 - Exemple de configuration de zones de sécurité dans un bâtiment

Légende

<div style="border: 1px solid black; padding: 5px; text-align: center;"> door ↓ </div>	Le contrôle d'accès est effectué dans un seul sens (un seul point d'accès dans la porte). Le passage est libre ou pas dans l'autre sens, en fonction de la configuration de la porte physique.
<div style="border: 1px solid black; padding: 5px; text-align: center;"> door ↓ ↑ </div>	Le contrôle d'accès est effectué dans les deux sens (deux points d'accès dans la porte).

L'infrastructure physique est imposée par la maîtrise d'ouvrage (cf. Figure 2). Elle se compose d'un Serveur central gérant l'ensemble d'un bâtiment, auquel sont connectés des Terminals X servant de console d'administration. L'ensemble des détecteurs et lecteurs de cartes du bâtiment peut communiquer avec le Serveur via l'utilisation de multiplexeurs. Enfin, la consultation de l'état d'un bâtiment peut être réalisée via l'utilisation de PC au travers d'un réseau WAN.

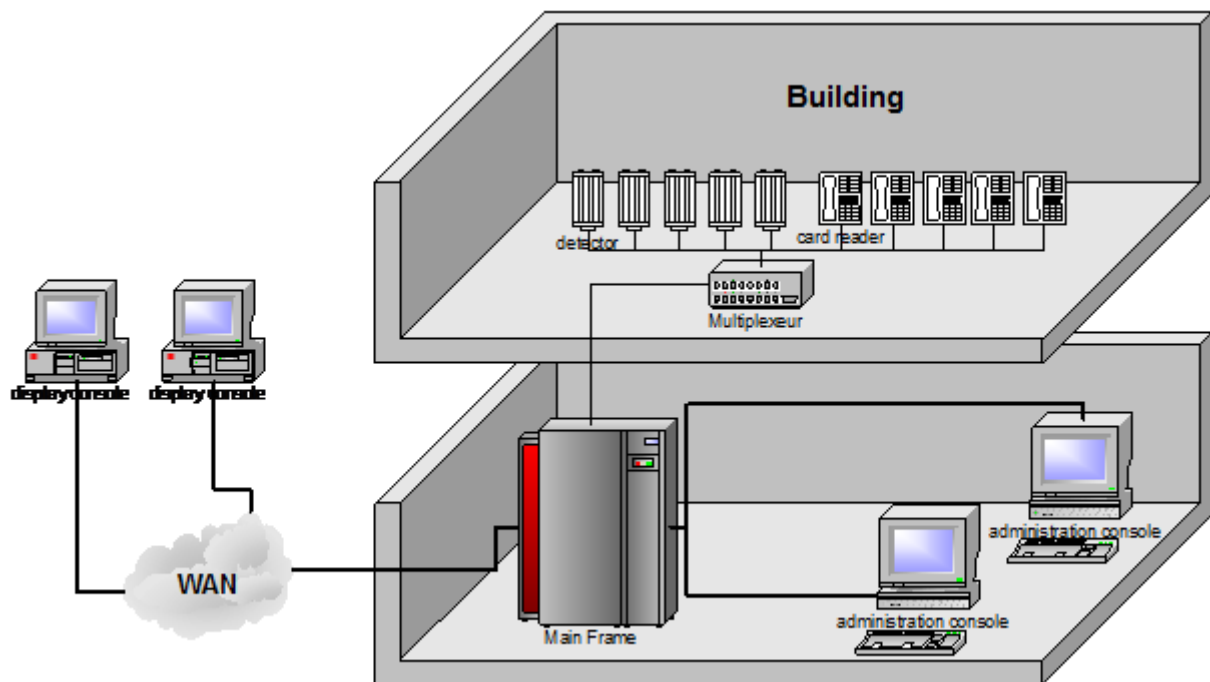


Figure 2 - Infrastructure physique du système de sécurité

Modèle d'analyse

Diagramme des classes métier du système

Le diagramme suivant (cf. Figure 3) montre les classes d'analyse fondamentales et leurs relations pour démarrer la conception du système

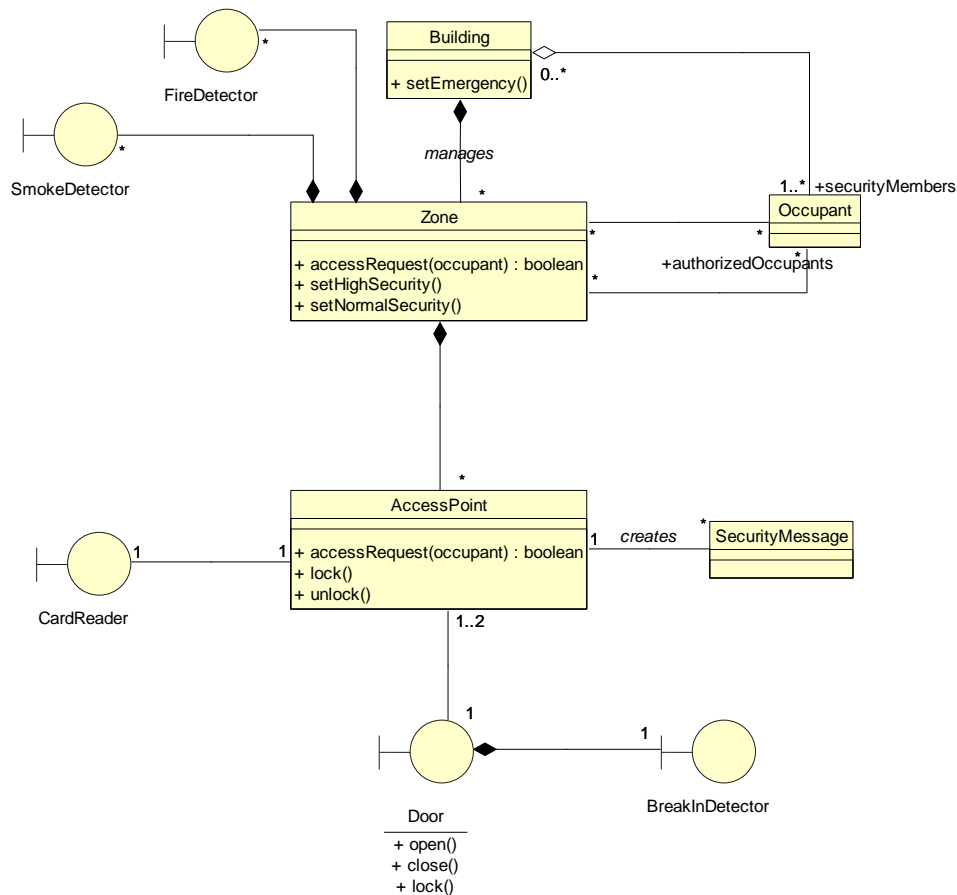


Figure 3 - Les principales classes métier du système

Rappel : définition des classes Boundary

<p>Boundary</p>	<p>Classe représentant une interface entre le système et une entité externe (personne physique ou autre système)</p>
Représentation UML	Classe stéréotypée <<boundary>>

Au sein du modèle d'analyse du système de sécurité, des classes <<Boundary>> sont utilisées pour représenter :

- Les interfaces avec les détecteurs répartis dans le bâtiment : classes FireDetector, SmokeDetector, BreakInDetector
- Les interfaces avec les lecteurs de cartes : classe CardReader
- Les interfaces avec les portes : classe Door

Comportement des points d'accès, portes, zones de sécurité et bâtiments

Comportement d'un point d'accès – classe AccessPoint :

Chaque point d'accès comporte deux états différents :

- Dans l'état « locked », la sécurité est activée sur le point d'accès et il est nécessaire de s'authentifier pour pouvoir ouvrir la porte contenant le point d'accès.
- Dans l'état « unlocked », aucune authentification n'est nécessaire pour pouvoir ouvrir la porte. Il est tout de même nécessaire d'insérer sa carte pour pouvoir déverrouiller la porte.

Le passage d'un état à un autre se fait sur réception des événements lock/unlock.

Comportement d'une porte – classe Door :

Le comportement d'une porte est donné par la Figure 4 :

- Dans l'état « Locked », la porte est verrouillée (gachette électrique activée)
- Dans l'état « Closed », la porte est déverrouillée (gachette électrique désactivé) et en attente d'ouverture
- Dans l'état « Opened », la porte est ouverte
- A la fermeture de la porte (événement « close »), le verrouillage de la porte (retour à l'état « Locked ») est réalisé uniquement si le bâtiment n'est pas en mode d'évacuation.

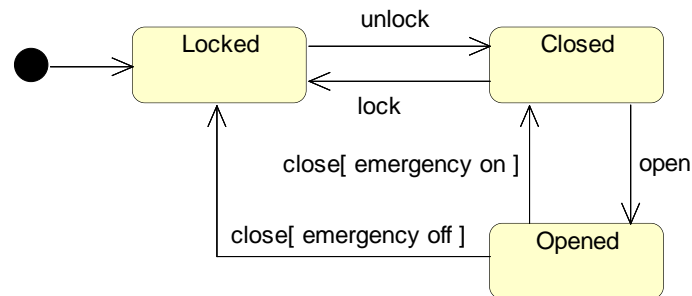


Figure 4 - Comportement d'une porte

Comportement d'une zone de sécurité – classe Zone :

Chaque zone de sécurité dans un bâtiment peut être configurée selon deux modes (cf. Figure 5) :

- Sécurité Normale (état « Normal Security Mode ») : l'accès est permis pour les membres autorisés ainsi que les membres de la sécurité
- Sécurité Haute (état « High Security Mode ») : l'accès est uniquement permis aux membres de la sécurité

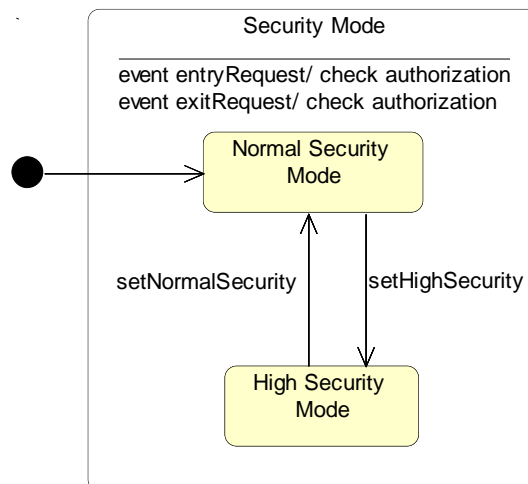


Figure 5 – Comportement d'une zone de sécurité

Comportement d'un bâtiment – classe Building :

Chaque bâtiment possède deux modes de fonctionnement (cf. Figure 6) :

- Etat « Security Mode » : la sécurité est activée dans le bâtiment
- Etat « Emergency Mode » : l'ensemble des portes du bâtiment sont déverrouillées pour permettre l'évacuation

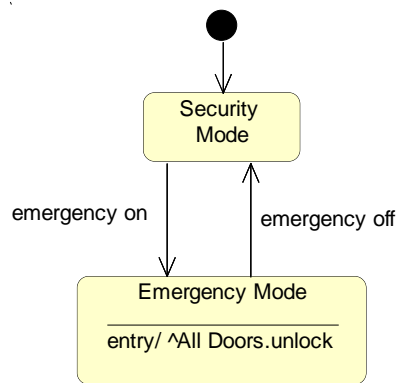


Figure 6 - Comportement d'un bâtiment

Elaboration du modèle de conception

La conception du système est réalisée par itération, en se concentrant à chaque fois sur un problème particulier à résoudre.

Durant l'étude de cas, le travail à réaliser pour chaque itération se décompose de la manière suivante :

- 1- **Identification du ou des Design Patterns à utiliser à partir de l'énoncé du problème**
- 2- **Elaboration d'un diagramme de classes, avec pour chacune les attributs et méthodes nécessaires à la résolution du problème**
- 3- **Elaboration d'un ou plusieurs diagrammes de collaboration pour montrer comment les objets interagissent pour réaliser la fonctionnalité demandée**

Itération 1 : gestion des détecteurs

Enoncé du problème :

Les détecteurs présents dans un bâtiment peuvent provenir de différents constructeurs. Par exemple, un constructeur peut fournir des détecteurs dédiés aux bâtiments militaires, tandis qu'un autre peut fournir différentes familles de détecteurs dédiés à différentes catégories de bâtiments civil (immeuble de bureaux, usine, etc.). Ceci entraîne les exigences suivantes :

Req#1 : *Chaque bâtiment est équipé par une famille cohérente de détecteurs provenant d'un fournisseur. Pour notre système de sécurité, nous désirons gérer l'utilisation de ces familles de manière transparente.*

Req#2 : *les changements d'état d'un détecteur doivent pouvoir être facilement notifiés à d'autres objets du système (des composants graphiques par exemple).*

Travail de conception :

- Proposez une solution au problème posé à base de Design Patterns et justifiez vos choix.
- Elaborez le diagramme de classes correspondant en identifiant les principales méthodes sur chaque classe introduite.
- Illustrez votre solution avec un diagramme de collaboration.
- Elaborez un découpage en packages/sous-systèmes uniquement pour la partie traitée et indiquez les dépendances.

Itération 2 : configuration des zones

Enoncé du problème :

La configuration des zones de sécurité doit pouvoir être effectuée de manière fine. Les exigences suivantes doivent être prises en compte :

Req#3 : *Deux catégories de zones doivent être considérées : les départements et les salles. Les départements peuvent contenir d'autres zones.*

Req#4 : *Au comportement générique d'une porte doivent pouvoir être ajouté dynamiquement et de manière transparente un ou plusieurs comportements supplémentaires permettant de gérer au cas par cas des niveaux de sécurité supplémentaires, comme :*

- a. *La levée d'une alarme si une porte reste ouverte plus d'un certain temps.*

b. La levée d'une alarme si une porte reste déverrouillé plus d'un certain temps et son verrouillage automatique

Travail de conception :

- Proposez une solution à base de Design Patterns et justifiez vos choix.
- Elaborez le diagramme de classes correspondant en identifiant les principales méthodes sur chaque classe introduite.
- Elaborez un diagramme de collaboration illustrant l'utilisation d'un comportement supplémentaire sur une porte.
- Mettez à jour le découpage en packages/sous-systèmes commencé durant l'itération 1 pour prendre en compte la partie traitée. Faites une séparation entre la définition abstraite du contenu d'un bâtiment et la définition concrète, qui permettra de définir plus tard d'autres classes d'éléments de configuration par spécialisation des éléments abstraits.

Note : ne traitez pas dans cette itération les autorisations d'accès à une zone.

Itération 3 : gestion des autorisations d'accès

Enoncé du problème :

Le principe d'authentification d'une personne pour accéder à une zone est le suivant (cf. diagramme d'analyse de la Figure 7) :

- Une personne insère sa carte magnétique dans le lecteur du point d'accès pour demander l'entrée dans la zone.
- Le lecteur lit la carte pour récupérer l'identifiant de la personne et exécute une requête d'entrée au système, via l'objet AccessPoint correspondant.
- L'objet AccessPoint transmet à son tour la demande à son objet Zone qui va rechercher si la personne est autorisée à pénétrer, en fonction de la date courante.
- Si l'autorisation est accordée, l'objet AccessPoint crée un message de sécurité et déverrouille la porte. La personne peut alors ouvrir celle-ci.
- Lorsque la porte est refermée, celle-ci se verrouille automatiquement, si le bâtiment n'est pas en mode évacuation.

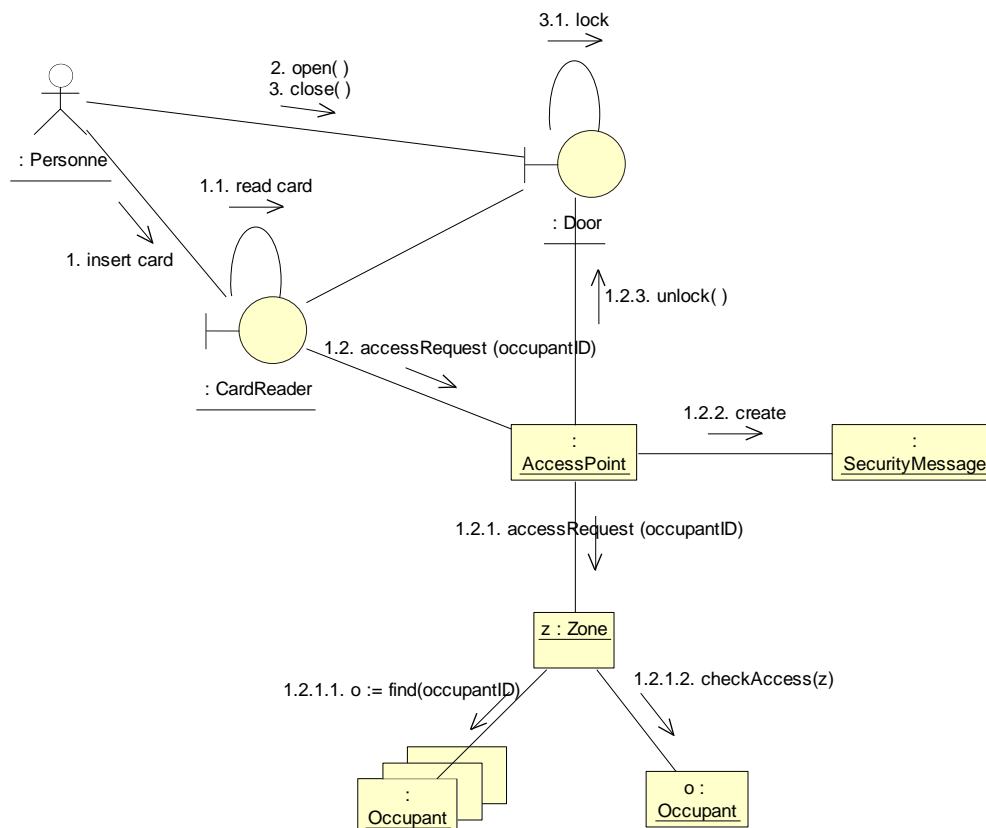


Figure 7 - Principe d'autorisation d'accès d'une personne dans une zone

La gestion et le suivi des autorisations d'accès aux zones de sécurité d'un bâtiment amènent les exigences suivantes :

Req#5 : *La configuration des autorisations d'accès pour chaque zone doit être découplée au maximum de la configuration de ces mêmes zones, de manière à pouvoir éventuellement réutiliser un système de gestion de la sécurité existant.*

Req#6 : *Chaque point d'accès est responsable de la création de messages de sécurité (cf. Figure 3 – classe SecurityMessage), permettant notamment d'archiver les demandes d'accès effectuées. Par contre, les points d'accès ne doivent pas être responsables de la manière dont ces messages seront traités par le système, notamment pour leur archivage et leur restitution.*

Req#7 : *les changements d'état d'une porte doivent pouvoir être facilement notifiés à d'autres objets du système (des composants graphiques par exemple).*

Travail de conception :

- Proposez une solution à base de Design Patterns et justifiez vos choix
- Elaborez le diagramme de classes correspondant en identifiant les principales méthodes sur chaque classe introduite
- En partant du diagramme d'analyse de la Figure 7, élaborer un diagramme de collaboration montrant le traitement d'une demande d'accès à une zone
- Proposez un découpage en packages/sous-systèmes pour la partie traitée.

Itération 4 : architecture logicielle

Proposez une architecture en couche simple décrivant notre système de sécurité. Indiquez la responsabilité de chaque couche et le sens des dépendances entre les couches. L'architecture doit prendre en compte les interfaces avec les entités externes du système.

Note : Une structuration en couche est généralement transverse au découpage en packages et sous-systèmes. Chaque couche est représentée en UML par un package stéréotypé <<layer>>.

Calcul des métriques

Calculer les métriques des packages de l'application et commentez-les.

Note : ne prenez pas en compte l'utilisation du JDK dans le calcul.