Revue-IRS

Revue Internationale de la Recherche Scientifique
(Revue-IRS)
ISSN: 2958-8413
Vol. 1, No. 4, Août 2023

Conception d'un système informatique des gestion des espaces dans les cimetières de la ville de Lubumbashi

LUFULUABO KENDA HILAIRE

Résumé : Le présent article se concentre sur le problème de gestion des espaces dans un cimetière dans la ville de Lubumbashi étant donné que cette ville possède quatre sites d'inhumation et une grandes masse de la population, nous y sommes penché en y utilisant la méthode descriptive et analytique basé sur le langage de modélisation UML dans sa version 2.0 selon la démarche 2TUP pour concevoir un système informatique de gestion des espaces qui doit aider l'administration de tutelle de la mairie de ville de Lubumbashi de trouver les fonctionnalité telles que : l'attribution d'un espace, l'attribution d'une concession, l'accord des visites, la recherche des espaces, la sécurité avec gestion des profits d'utilisateur.

Mots-clés : Conception, système informatique, gestion des espaces dans les cimetières.

Digital Object Identifier (DOI): https://doi.org/10.5281/zenodo.8253604

1. Introduction

Dès l'âge de la pierre à nos jours, l'esprit perfectionniste de l'être humain n'a cessé de lui permettre d'améliorer sa vie quotidienne. Le passage de la mécanique aux domaines de l'informatique, de l'électronique, d'automatique et de domotique a révolutionné la vie journalière de l'être humain. Les nouvelles technologies de l'information et de communication illustrent ce phénomène. Aujourd'hui, vu l'intérêt croissant de vouloir gagner en temps en collectant, traitant et distribuant l'information et pas mal d'autres raisons, ont poussé petites, moyennes, grandes entreprises et services d'administration publique à chercher des solutions informatiques capables de répondre à leurs besoins. Dans ce cadre s'inscrit notre travail de fin d'études qui consiste à réaliser un système informatique de gestion des espaces dans un cimetière sur mesure du bureau tutelle de l'administration générale pour la ville de Lubumbashi. Le bureau tutelle de l'administration générale de la mairie de la ville de Lubumbashi compte quatre sites d'inhumation repartis sur cette ville. Les différents sites d'inhumation reçoivent au jour le jour des corps à inhumer ainsi, face à cette multiplication des tombes pour chaque site d'inhumation ce bureau connait un problème d'urbanisation de ces sites dans le sens qu'il a difficile à connaitre quel espace a été occupé à une date précise et combien d'argent a été versé pour une période donnée, après combien de temps un site peut être rempli pour permettre à ce bureau une prévoyance de déménagement et avoir une cartographie mise à jour qui peut permettre que tel corps a été inhumé dans l'espace indiqué. C'est ainsi qu'en réponse au problème qui guide notre recherche,

nous pensons que la conception d'un système informatique de l'urbanisation pour tous les sites serait la meilleure solution au problème de gestion des espaces dans un cimetière. Pour atteindre cette solution nous recourons à la méthode descriptive et analytique basé sur le langage de modélisation UML 2.0 dans la démarche 2TUP, en utilisant la technique d'observation participative, la technique de l'analyse documentaire ainsi que l'interview. C'est ainsi que cet article est axé sur la gestion des espaces dans un cimetière en contrôlant les services d'inhumation depuis la livraison du permis d'inhumation jusqu'à l'inhumation, en prévoyant le déménagement du site depuis l'enregistrement du premier corps inhumée dans un site par le service d'inhumation jusqu'au dernier corps. Hormis l'introduction et la conclusion cet article présente deux points sur lesquels développé le sujet abordé. Dont le premier point concerne l'analyse fonctionnelle et le concerne la conception du système informatique de gestion des espaces dans un cimetière dans la ville de Lubumbashi.

2. Analyse fonctionnelle

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire nous dit Pascal Roques.

1.1 Détermination des cas d'utilisation

Un cas d'utilisation est une entité cohérente d'une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation est donc une modélisation d'un service rendu par le système, sans imposer le mode de réalisation de ce service. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera. Le cas d'utilisation met en valeur les interactions métier entre les acteurs et le système. Le tableau ci-dessous montre les cas d'utilisation retenus dans le système Gestion des espaces.

Tableau 1 : tableau des cas d'utilisation

Cas d'utilisation	Acteurs	Messages émis/reçus par les acteurs
RecevoirActeDecès	Secrétaire / Section inhumation	Emet : validation espace Reçoit : demande espace, prix espace, liste espaces
RecevoirDemandeConcession	Secrétaire / section inhumation	Emet : validation concession Création concession Reçoit : prix concession, demande concession, Liste concession
RecevoirDemandeVisite	Secrétaire / section inhumation	Emet : validation espace Reçoit : demande visite, prix visite
GererEspace	Secrétaire / section inhumation	Emet : création espace, rechercher espace Reçoit : espace trouvé Espace créé Concession trouvée Concession créée
RechercherEspace	Police cimetière	Emet : affichage formulaire Reçoit : identités défunt, confirmation concession occupée

InhumationDefunt	Fossoyeur	Emet : validation inhumation Reçoit : confirmation inhumation

2.2 Diagramme de cas d'utilisation

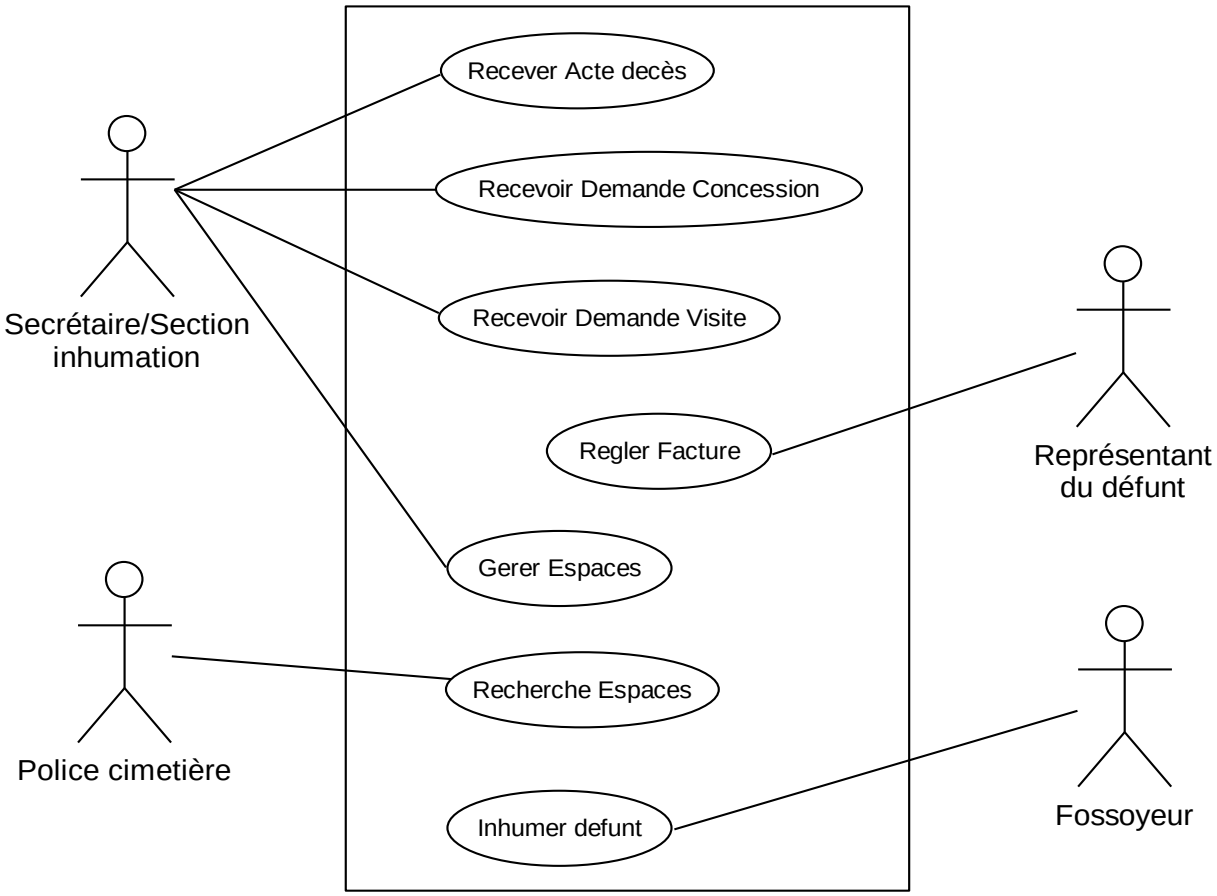

Figure 1 : Diagramme de cas d'utilisation

2.3 Identification des classes candidates

L'identification de classes candidates est fonction des concepts connus, c'est-à-dire, ceux qui sont utilisés couramment dans la gestion des espaces (les objets métier). Nous ajoutons dans un second temps des concepts applicatifs, liés à l'informatisation. A partir de la description de chaque cas d'utilisation, il y a des classes participantes qui en découlent, soit par les concepts connus ou les concepts applicatifs. Nous formalisons ensuite ces concepts métier sous forme de classes et d'associations rassemblées dans un diagramme statique pour chaque cas d'utilisation. Ces diagrammes de classes candidates, appelés « diagramme préliminaire », n'ont pas d'objectif exhaustif. Ils servent uniquement à démarrer la découverte des classes du modèle d'analyse pour la partie de l'application. Voici les diagrammes de classes candidates par cas d'utilisation.

– Diagramme des classes candidates du cas d'utilisation « RecevoirActeDecès »

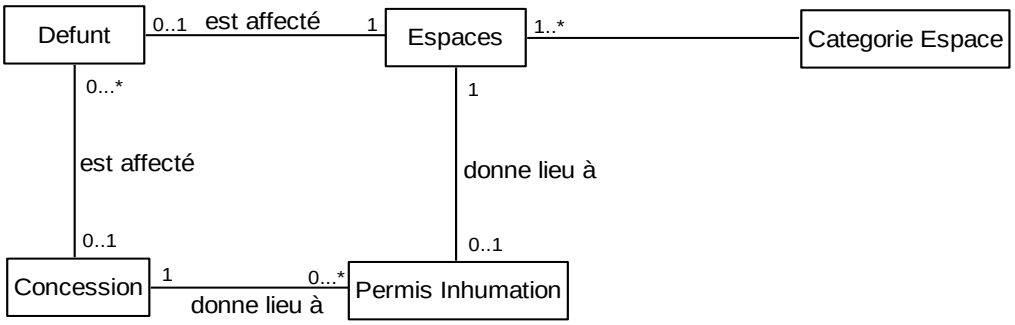

Figure 2 : Diagramme de classes candidates de cas d'utilisation « RecevoirActeDécès »

– Diagramme des classes candidates du cas d'utilisation « RecevoirDemandeConcession »

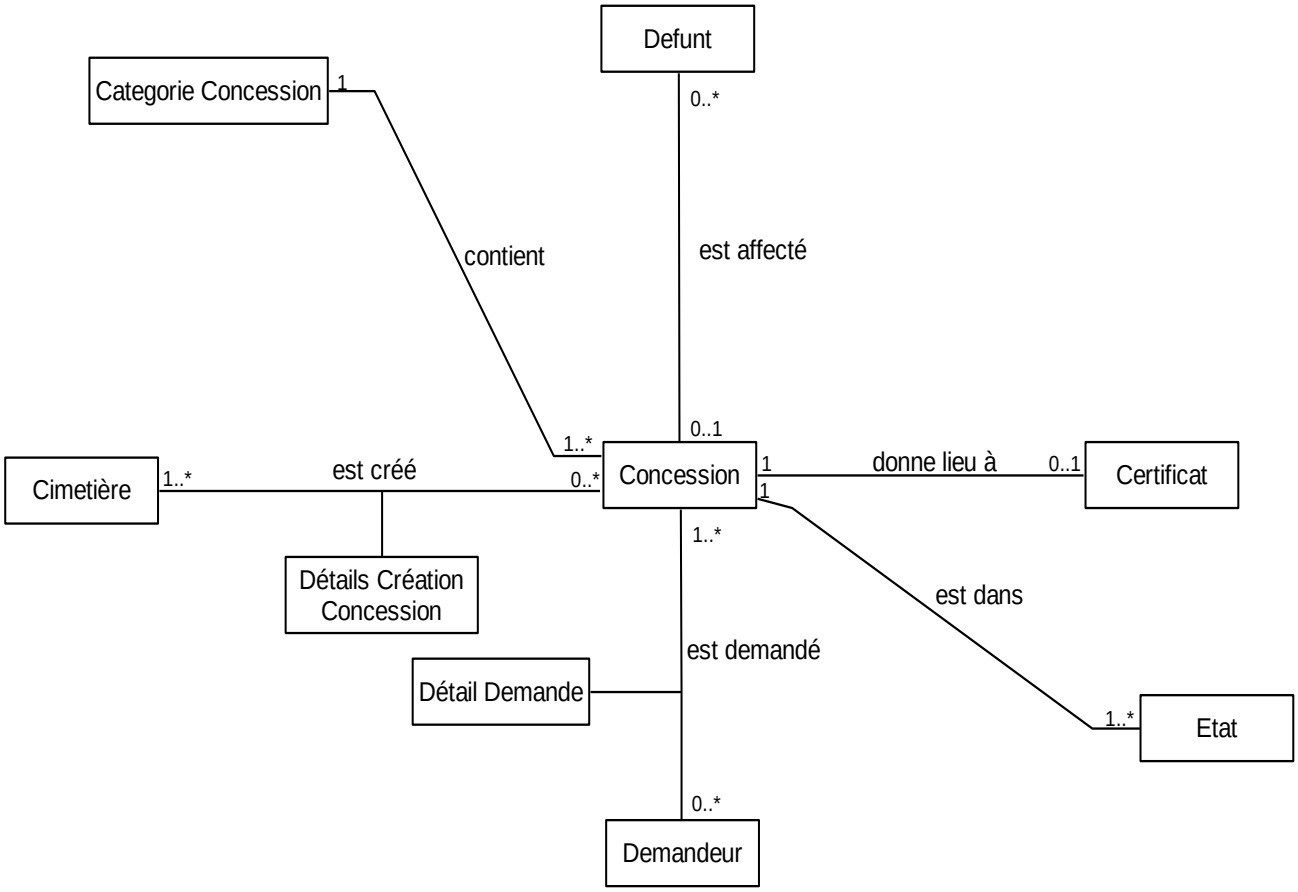

Figure 3 : Diagramme de classes candidates de cas d'utilisation « RecevoirDemandeConcession »

– Diagramme des classes candidates du cas d'utilisation « RecevoirDemandeVisite »

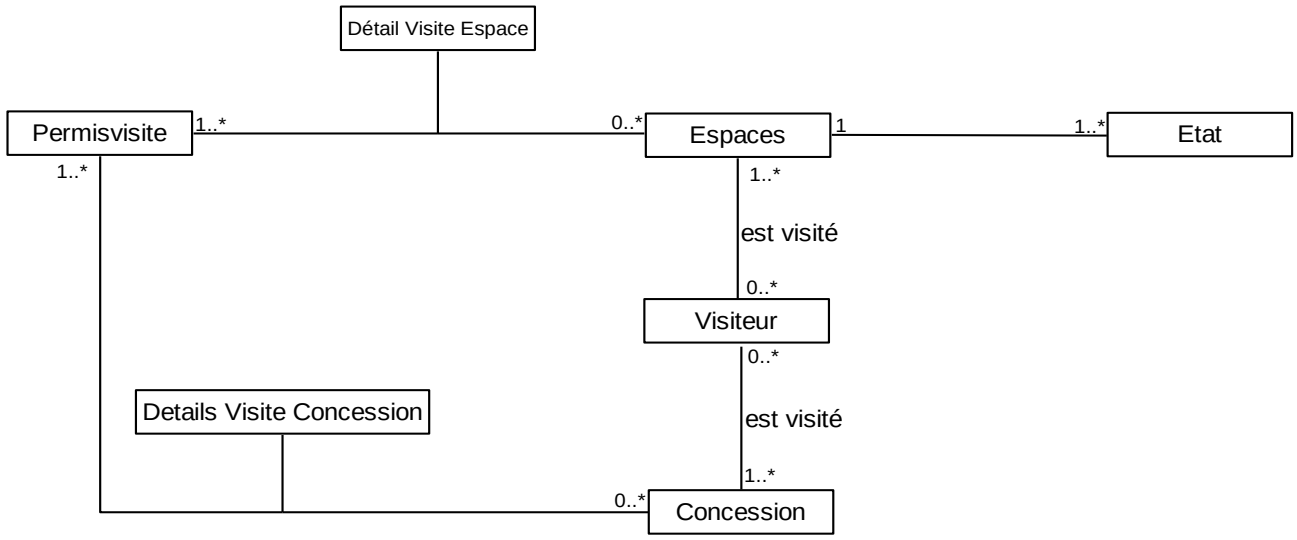

Figure 4 : diagramme de classes candidates de cas d'utilisation « RecevoirDemandeVisite»

– Diagramme des classes candidates du cas d'utilisation « ReglerFacture »

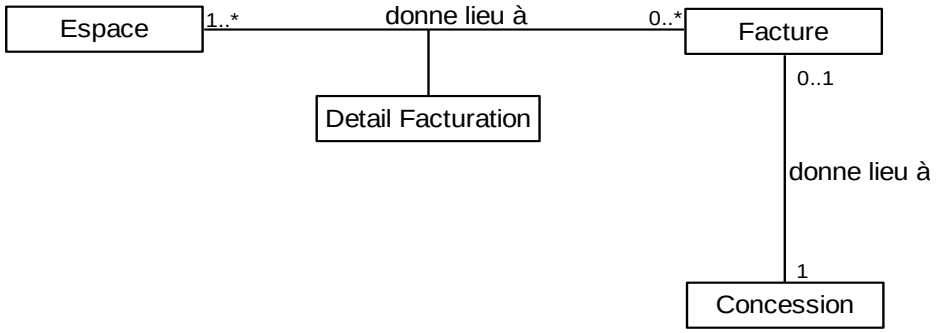

Figure 5 : Diagramme de classes candidates de cas d'utilisation « ReglerFacture»

– Diagramme des classes candidates du cas d'utilisation « GererEspaces »

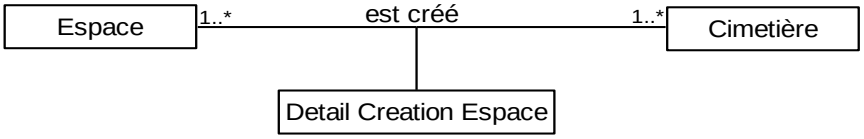

Figure 6 : Diagramme de classes candidates de cas d'utilisation « GererEspaces»

– Diagramme des classes candidates du cas d'utilisation « InhumerDefunt »

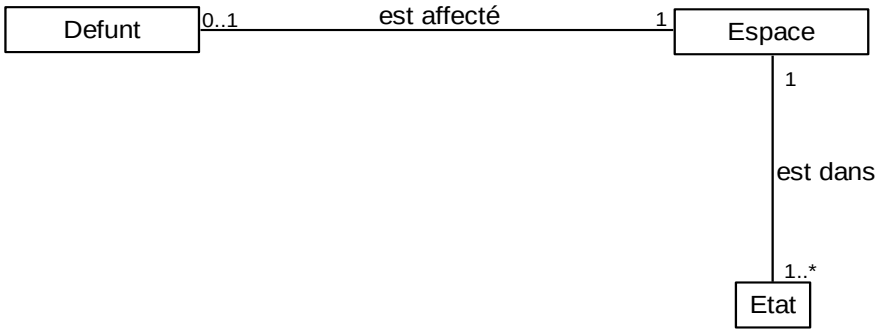

Figure 7 : Diagramme de classes candidates de cas d'utilisation « InhumerDefunt»

2.4 Capture de besoins technique

Par rapport l'analyse fonctionnelle qui permet de voir les besoins fonctionnels, les besoins techniques viennent en complément. Ils concernent les contraintes autres que la description métier et la description applicative. La spécification technique est nécessaire pour la conception d'architecture. Cette capture des besoins intéresse deux aspects : l'aspect logiciel et l'aspect matériel. Les matériels à utiliser doivent être définis suivant la structure logicielle prévue.

1.4.1 Spécification du point de vue matériel

Dans la spécification des méthodes et techniques de développement, nous avons fait des choix en rapport avec la configuration logicielle et matérielle à utiliser pour le cas d'étude. Cette configuration se base sur la sécurité du système, son utilisation, l'accès aux données. Pour cet article l'architecture à deux niveaux (local et section Inhumation) est celle qui est favorable. Les attentes techniques du système de gestion des espaces, selon l'architecture choisie, nous donne les nœuds suivants :

- un poste qui fera office du serveur de base de données ;
- un poste qui fera office du serveur d'applications ;
- 3 postes de travail pour les acteurs internes au système ;
- Une imprimante pour toutes impressions ;
- Un serveur web pour tous les représentants du défunt et aux 3 postes de se connecter via internet.

Voici le diagramme de déploiement de la gestion des espaces.

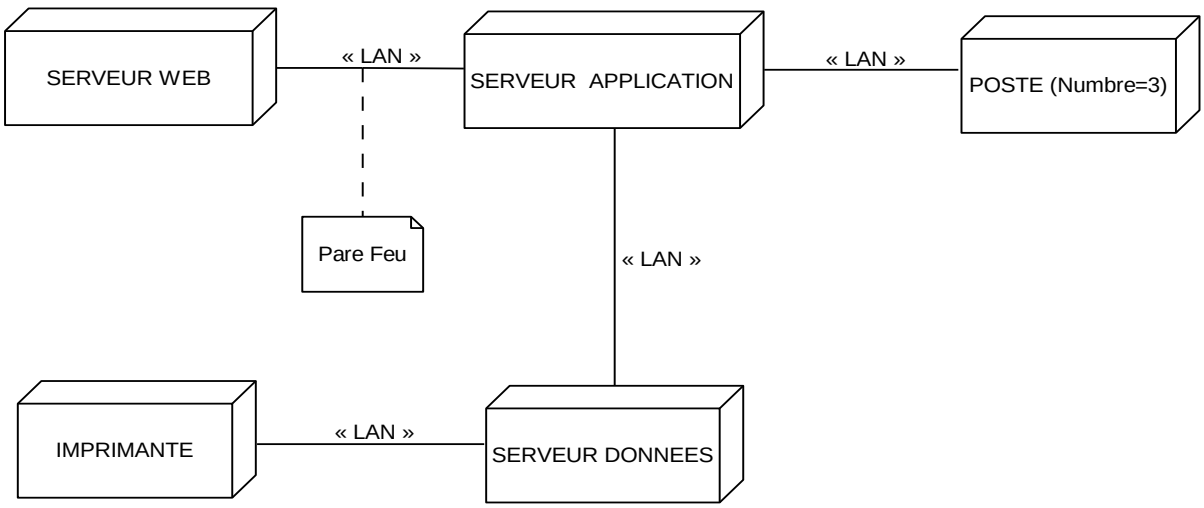

Figure 8 : Configuration matérielle Gestion des Espaces

Voici la configuration réseau de la gestion des espaces

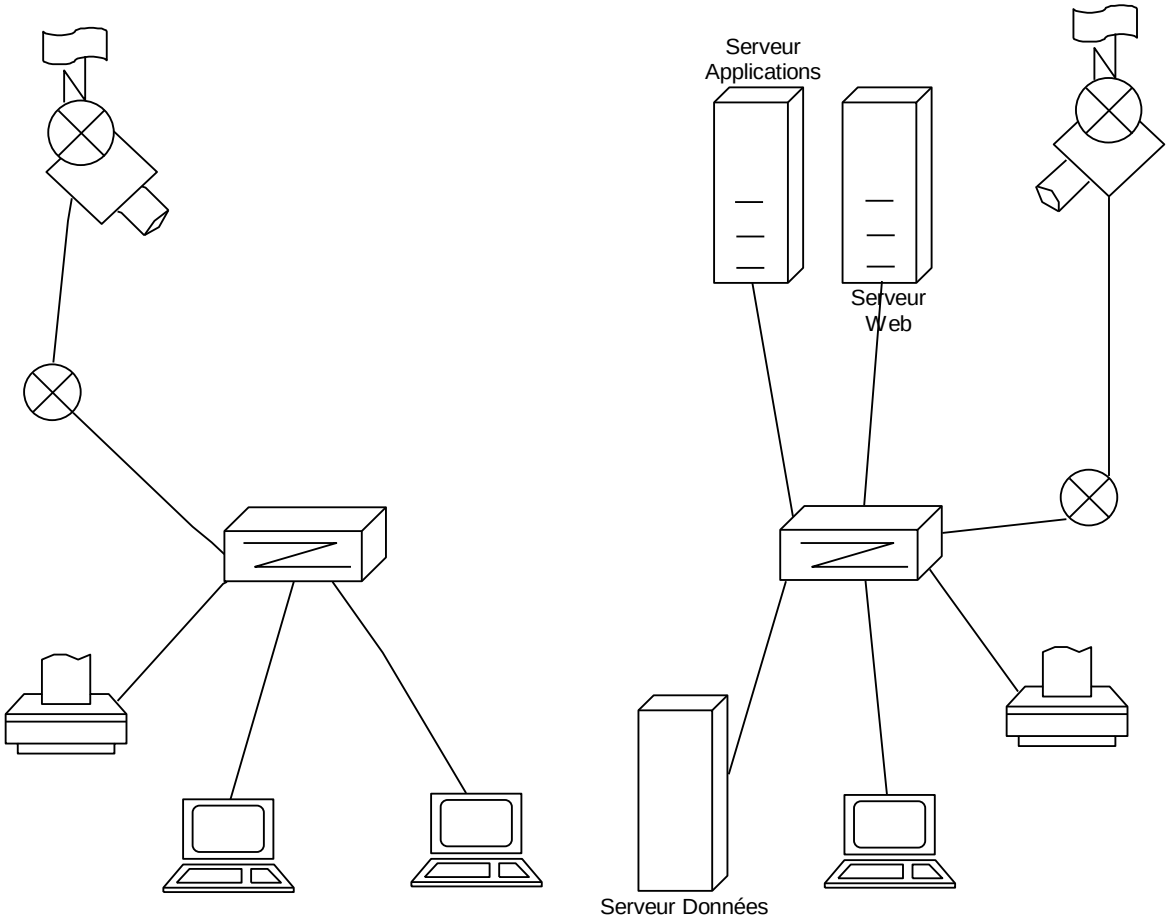

Figure 9 : Diagramme de la configuration du réseau de la gestion des espaces

1.4.2 Identification des cas d'utilisation technique

Les cas d'utilisation retenus pour la gestion des espaces ainsi que leurs exploitants sont :

- **L'utilisateur**
  - Utiliserobjets
  - ConsulterAide
  - S'authentifier
- **Administrateur système**
  - GérerSécurité

Ainsi, voici le diagramme des cas d'utilisation technique :

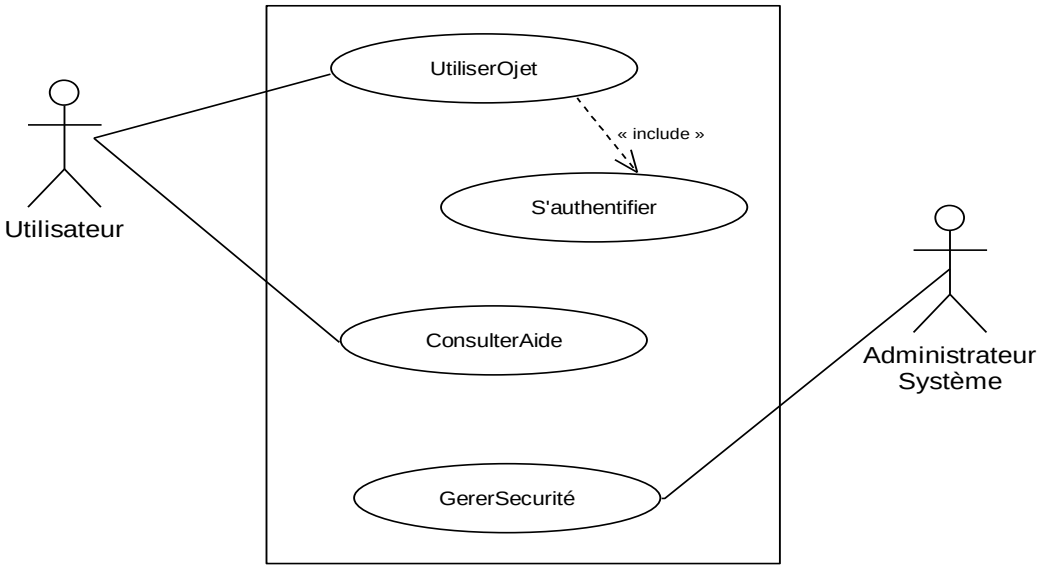

Figure 10 : Diagramme de cas d'utilisation technique

3. Conception du système informatique

Dans ce point nous continuons sur la branche gauche de notre modèle en Y du processus UP, c'est-à-dire les besoins fonctionnels ainsi il succède à la capture de besoins fonctionnels. A cet effet nous faisons le développement du modèle statique, le développement du modèle dynamique et la conception générique.

2.1 Développement du modèle statique

Cette partie du point nous permet de faire la construction du diagramme de classes en se basant sur les diagrammes des classes candidates présentés selon le cas d'utilisation et nous permet aussi d'optimiser ces classes candidates. D'après Rogues, les classes identifiées lors de l'étude des cas d'utilisation sont maintenant réétudiées, d'une manière détaillée afin de voir s'il faut en ajouter ou en supprimer certaines. Cet ajout ou suppression de classe se base sur les critères suivants :

- L'élimination des classes redondantes ;
- L'élimination des classes vagues ;
- L'élimination des classes à la place d'attribut groupe d'objets ;
- L'élimination des classes de conception ;
- La subdivision des classes ayant trop de responsabilités.

Après avoir réétudié les classes, nous faisons l'étude des associations et nous ajoutons les opérations. Pour notre cas au lieu de reprendre tous les diagrammes des classes candidates nous présentons ici, le diagramme de classe du modèle statique.

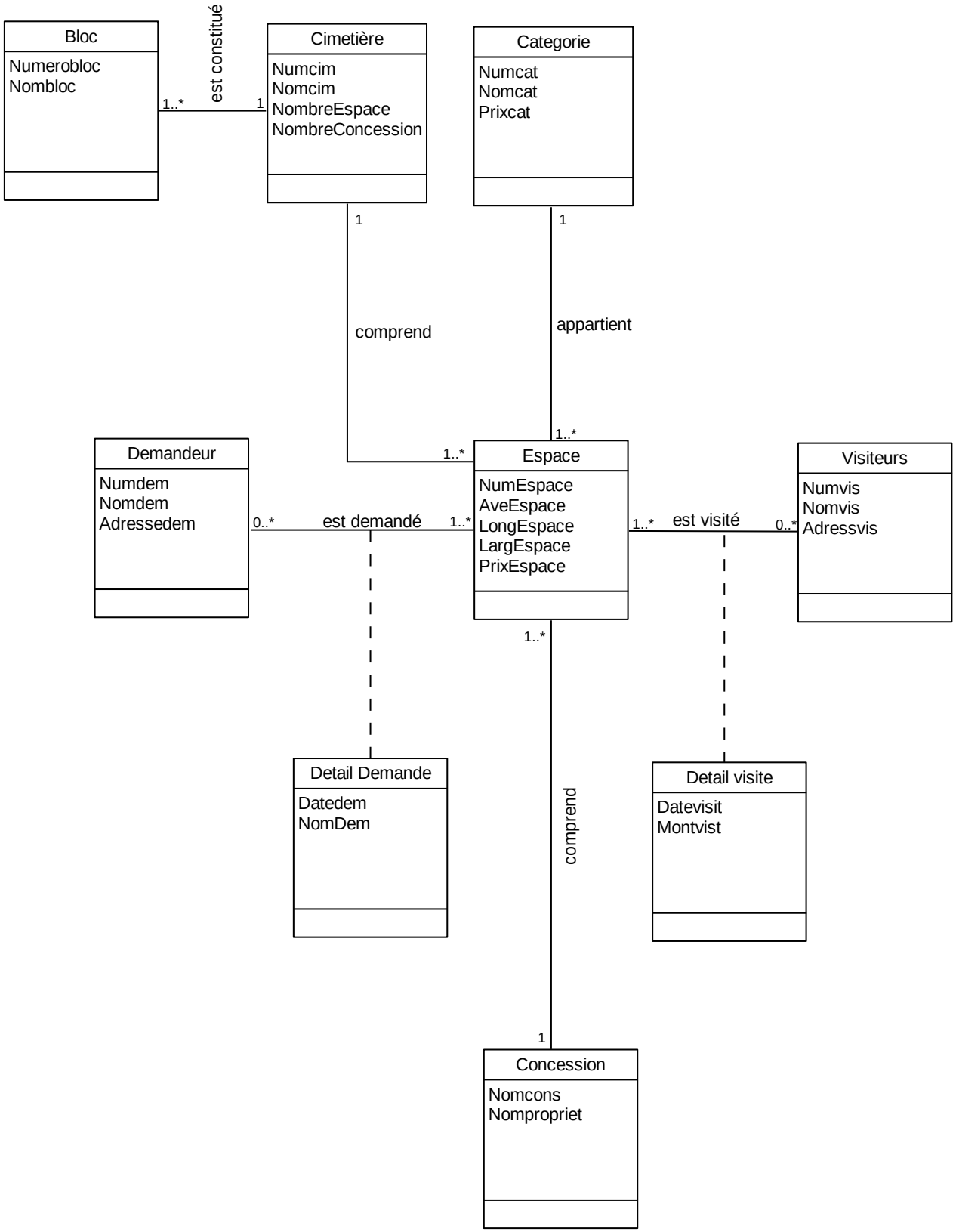

Figure 11 : Diagramme de classes du domaine

3.2 Développement du modèle dynamique

Dans cette partie du point, nous prenons les cas d'utilisation obtenus lors de l'étape de détermination des besoins fonctionnels où un scénario représente une séquence d'interactions entre le système et ses acteurs. Le système était

considéré comme une boite noire. Maintenant, nous considérons le modèle statique et nous remplaçons le système par une collaboration d'objets pour chaque cas d'utilisation.

3.2.1 Identification et formalisme des scénarios

Un scénario décrit une exécution particulière d'un cas d'utilisation du début à la fin. Il correspond à une sélection d'enchainements du cas d'utilisation.

Ainsi, on a les scénarios suivants pour les cas d'utilisation :

* Scénario pour le cas d'utilisation « RecevoirActeDécès »
  - Attribution d'un espace.
* Scénario pour le cas d'utilisation « RecevoirDemandeConcession »
  - Attribution d'une concession.
* Scénario pour le cas d'utilisation « RecevoirDemandevisite »
  - Accorder une visite.
* Scénario pour le cas d'utilisation « ReglerFacture »
  - Accorder une facture
* Scénario pour le cas d'utilisation « GererEspace »
  - Création d'un espace.
  - Rechercher d'un espace.
  - Annuler création d'un espace.
* Scénario pour le cas d'utilisation « Inhumerdefunt »
  - Affectation du defunt.

3.2.2 Formalisme des scénarios

Etant donné que les cas d'utilisation ont décrit ces scénarios, il s'avère intéressant pour nous de présenter seulement les diagrammes de séquence.

- **Scénario :** ***Attribution d'un espace***

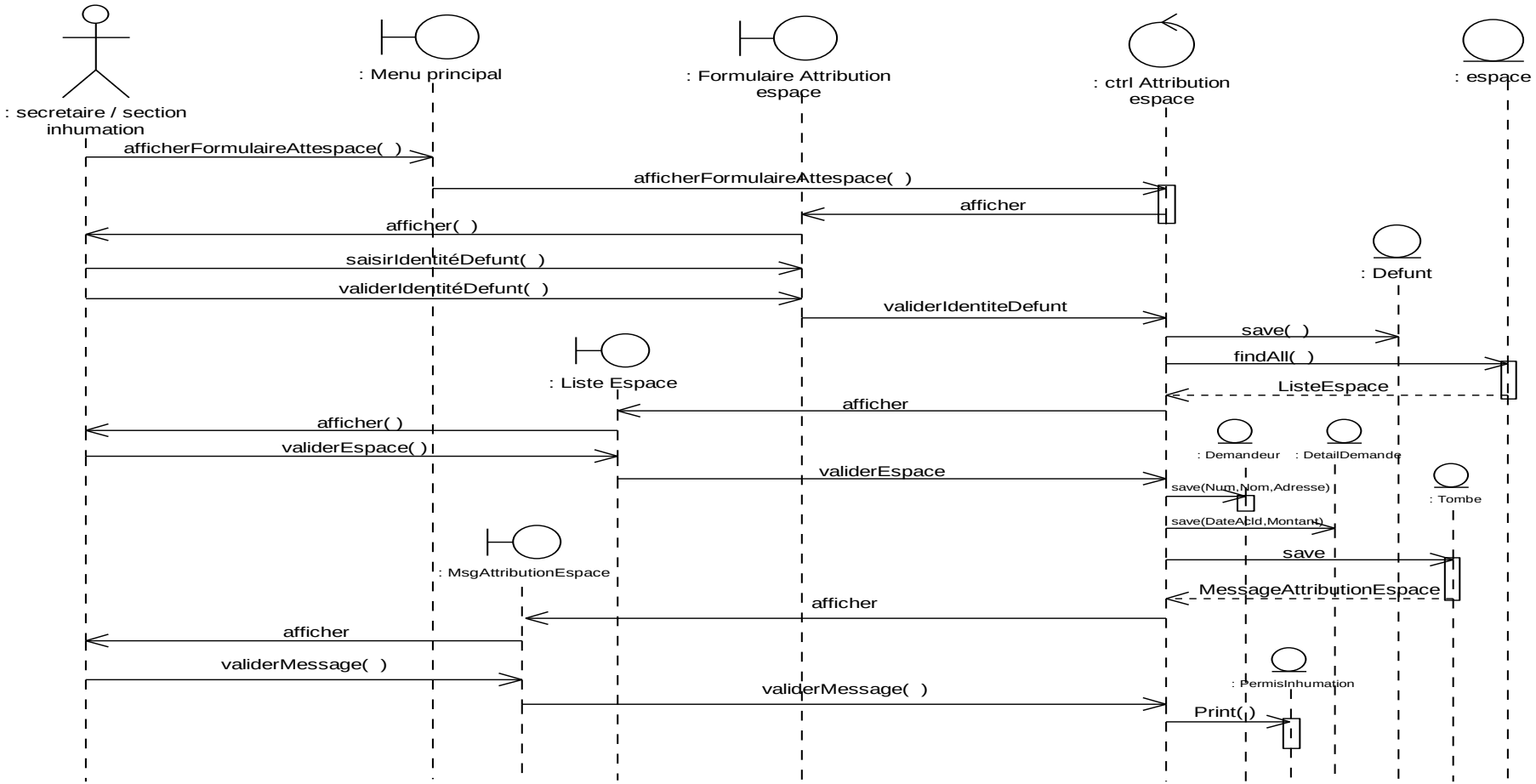

Figure 12 : Diagramme de conception des objets du scénario attribution d'un espace

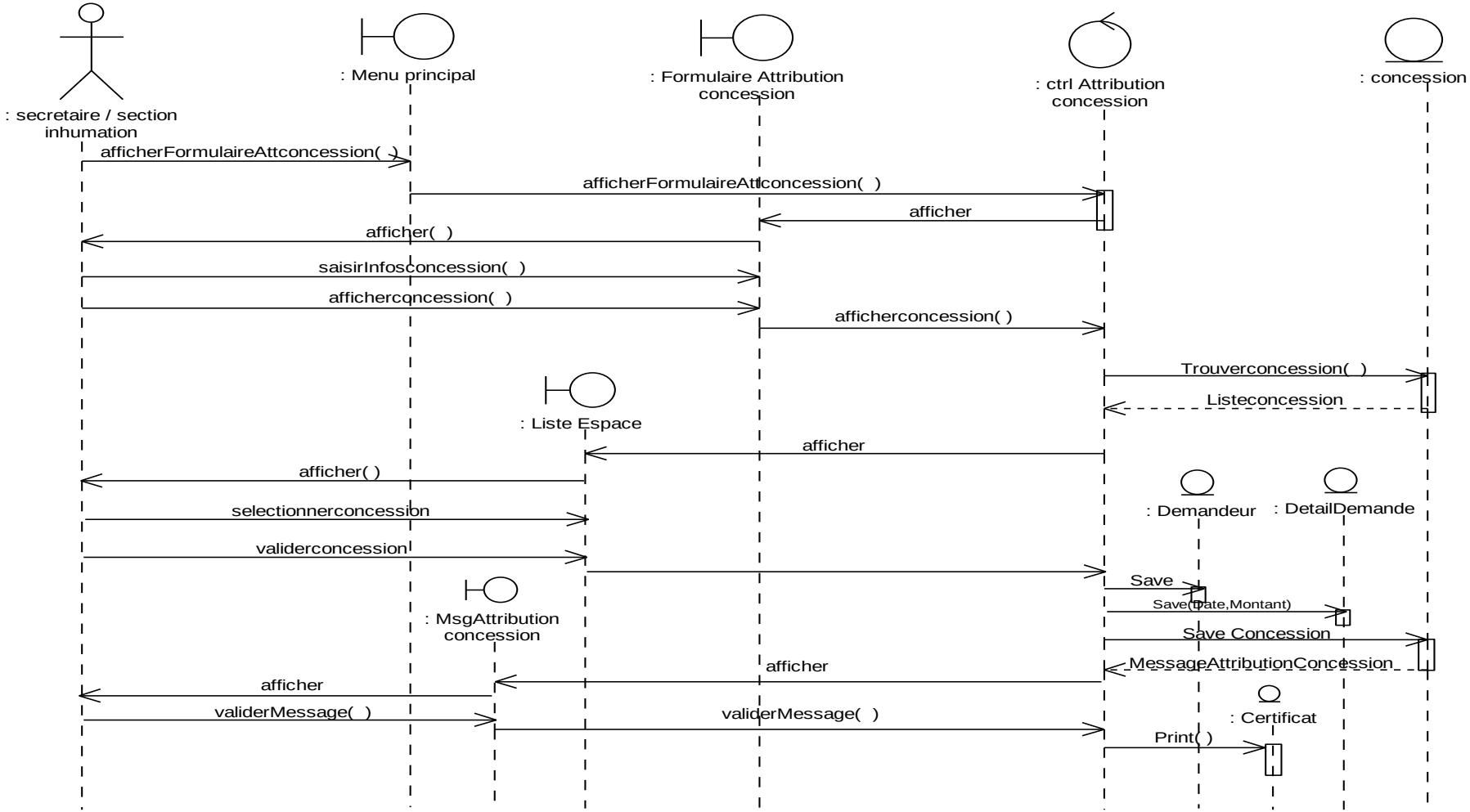

Figure 13 : Diagramme de conception des objets du scénario attribution d'une concession

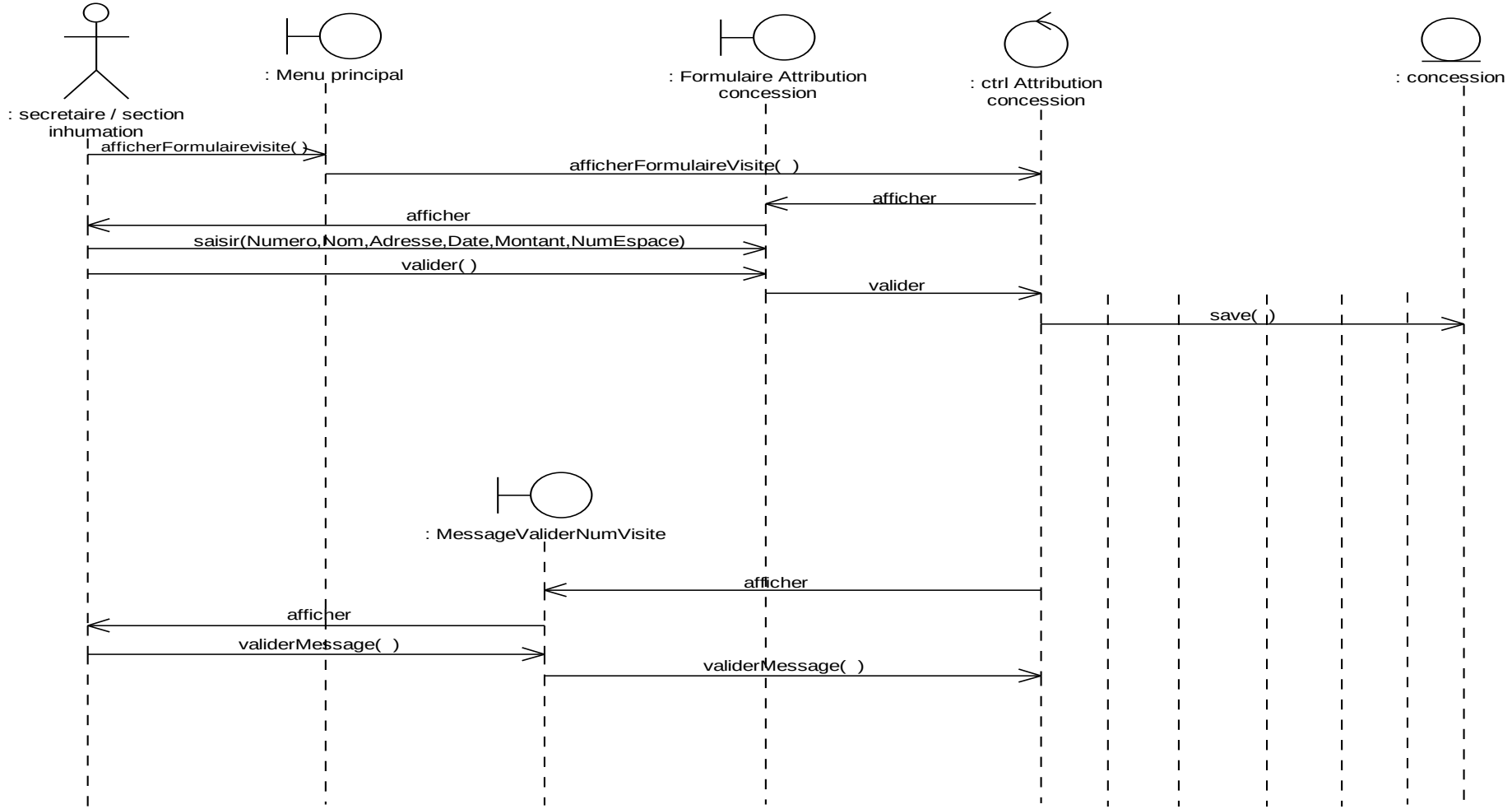

Figure 14 : Diagramme de conception des objets du scénario accorder visite

Revue Internationale de la Recherche Scientifique (Revue-IRS) - ISSN : 2958-8413

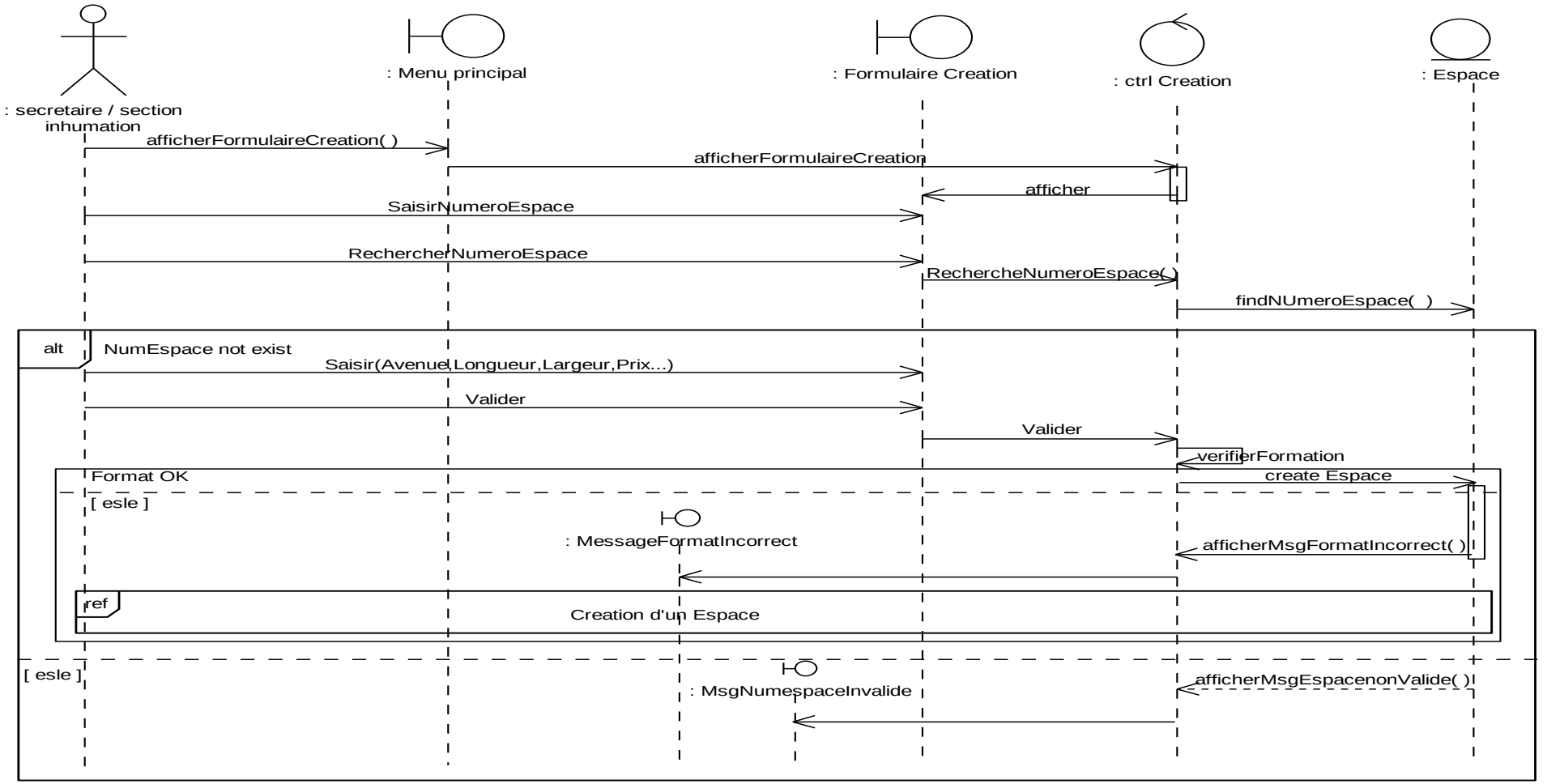

Figure 15 : Diagramme de conception des objets du scénario création espace

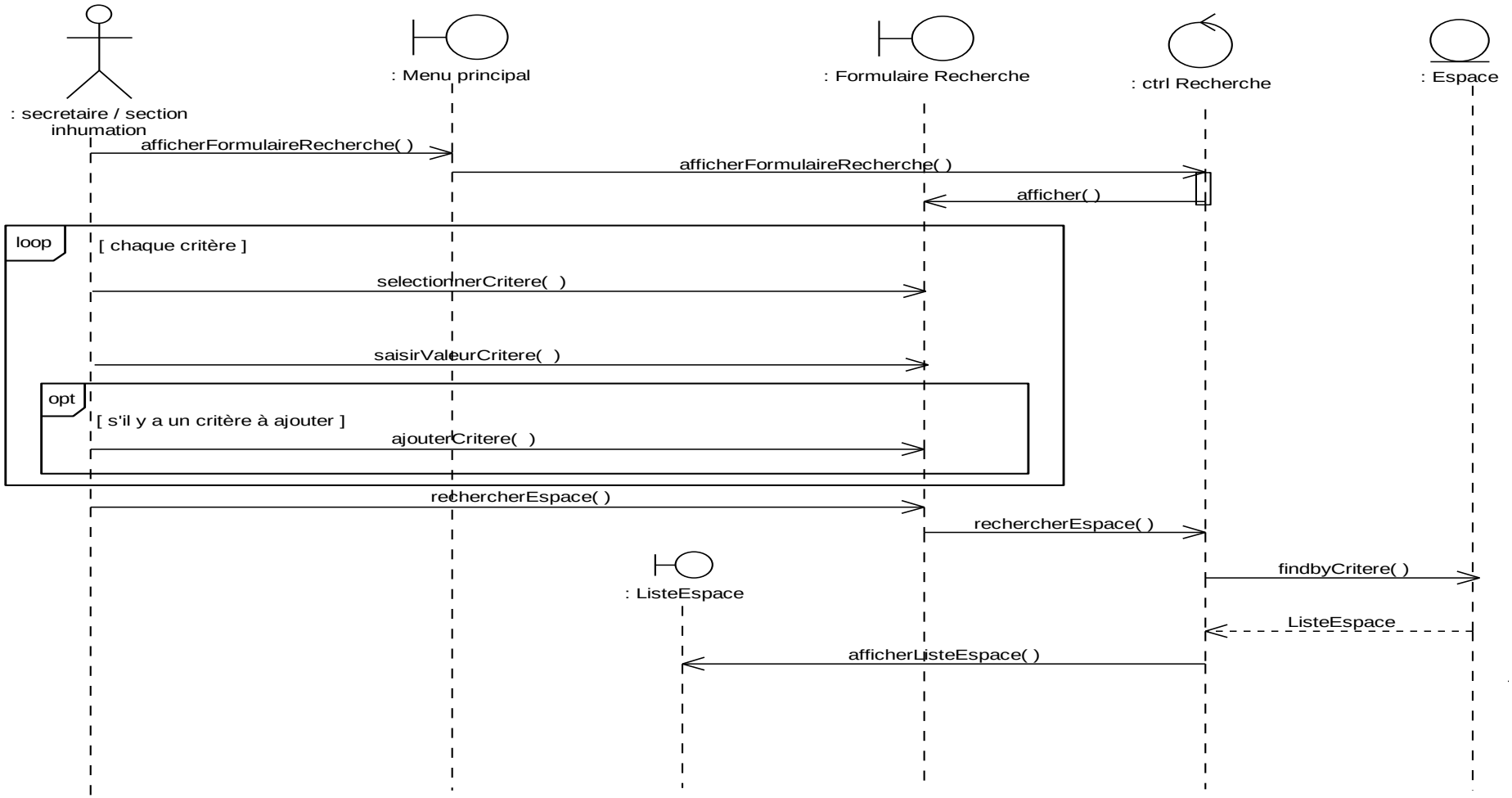

Figure 16 : Diagramme de conception des objets du scénario Rechercher espace

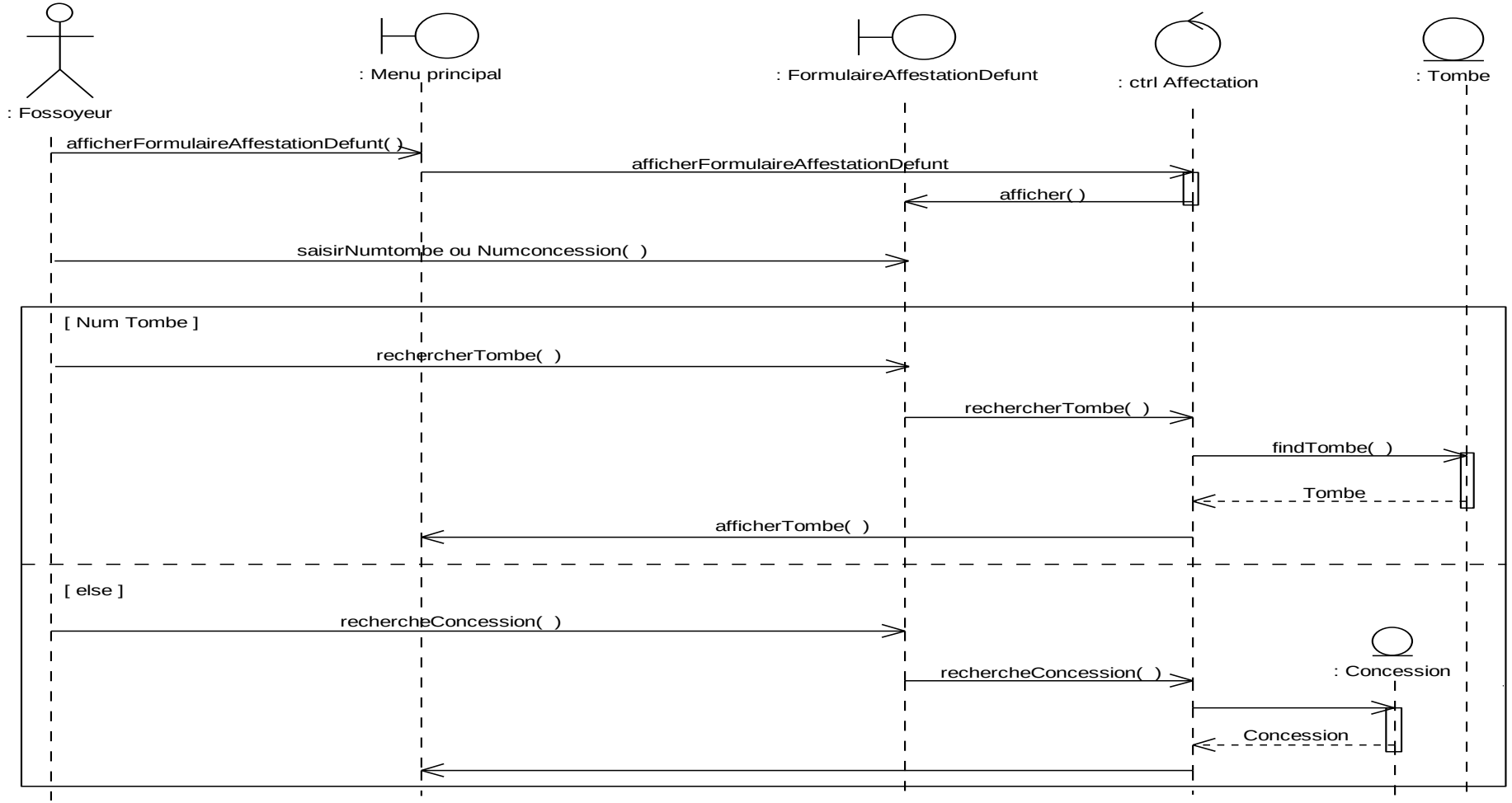

Figure 17 : Diagramme de conception des objets du scénario Affectation Défunt

4. Conception du modèle de déploiement

Le déploiement d'une solution client / serveur se construit sur la définition des postes de travail. Un poste de travail représente un ou plusieurs acteurs pouvant être localisés sur une machine d'un type particulier et remplissant une fonction identifiée dans l'entreprise. Il ne représente pas forcement une machine physique, mais peut consister en plusieurs machines, à condition qu'elles donnent lieu au même type de déploiement.

Ainsi, les postes de travail de la gestion des espaces sont répartis de la manière suivante :

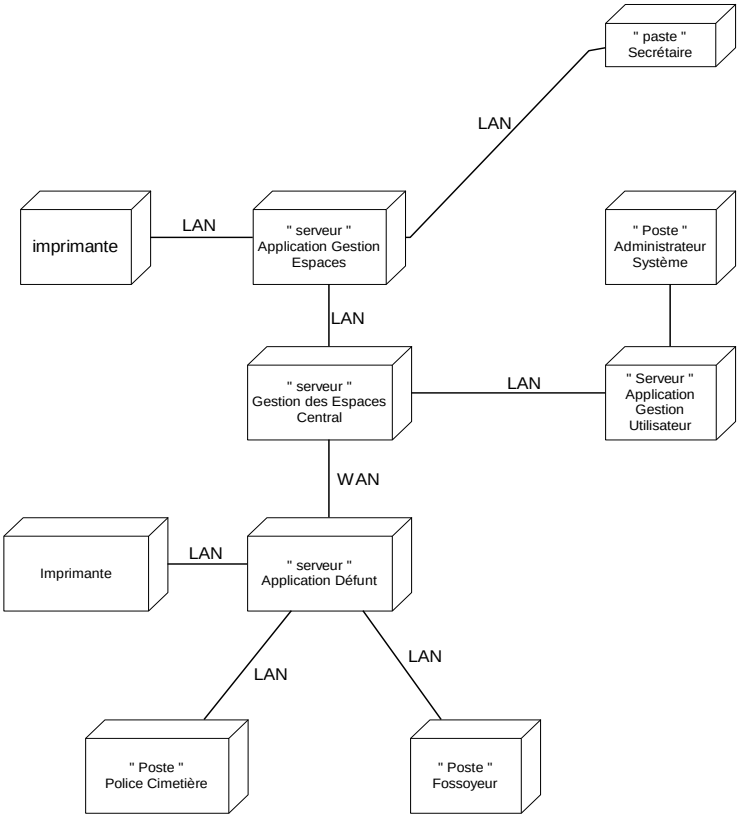

Figure 18 : Modèle de déploiement Gestion des Espaces

Les postes de travail s'articulent autour du serveur : Gestion des espaces via leurs serveurs d'application. Un serveur d'application pour utilisateur pour permettre à l'administrateur système de gérer les utilisateurs, un serveur d'application Gestion espaces pour permettre au secrétaire de gérer les espaces, un serveur d'application défunt pour permettre au Fossoyeur et au Policier du cimetière de gérer l'inhumation et la recherche des défunts. Le modèle d'exploitation commence à phase de la conception générique. A partir du modèle de déploiement obtenu, il est possible de le compléter en fonction des machines et des postes de travail, tout en intégrant les applications installées sur les postes de travail, les composants métiers déployés sur les serveurs et les instances des bases de données implantées sur des serveurs afin d'optimiser la distribution et éventuellement énumérer les interfaces utilisateurs. Dans notre cas de Gestion des espaces nous avons le diagramme d'exploitation suivant :

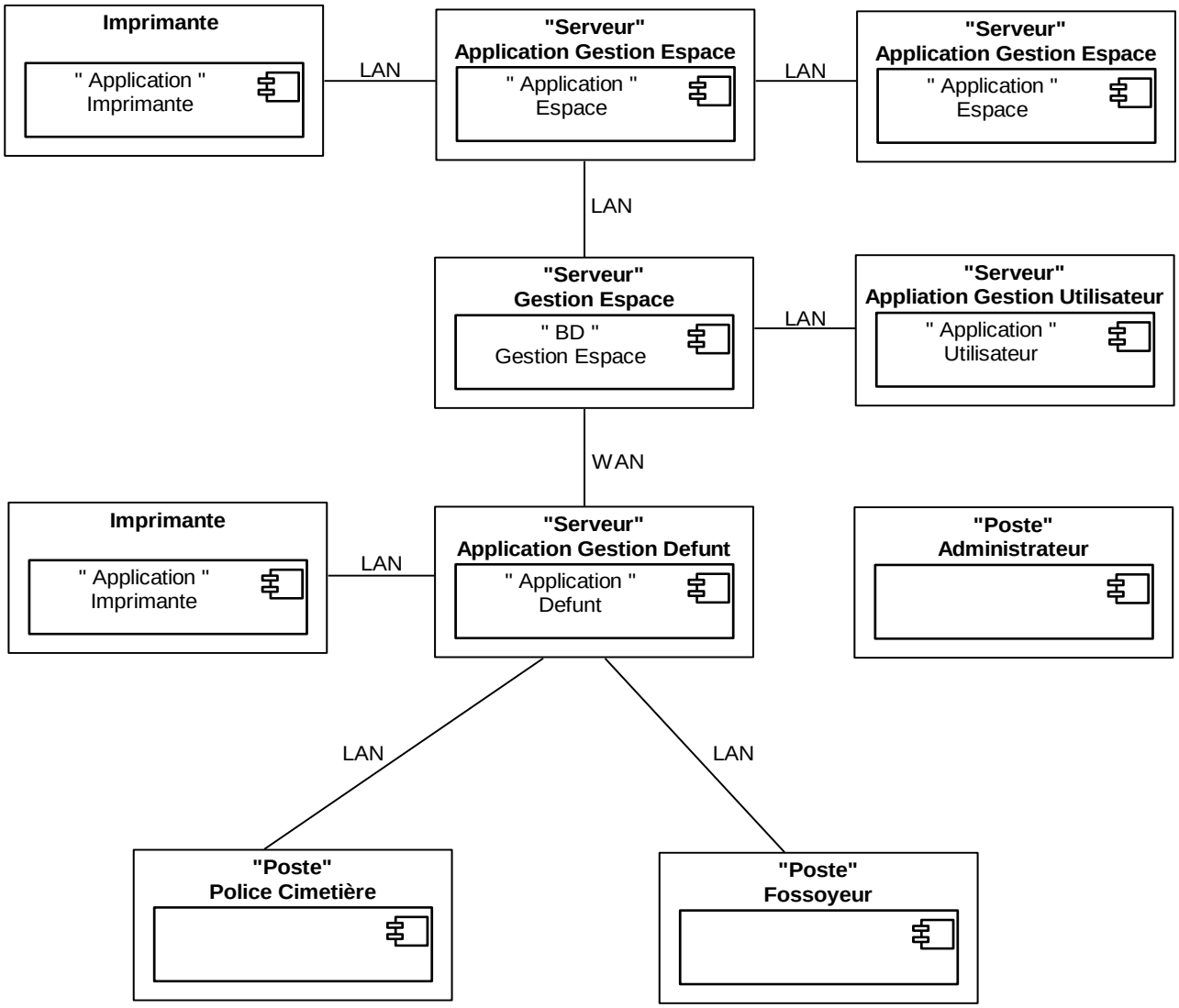

Figure 19 : Définition des applications dans le modèle de déploiement

Les différents composants de la gestion ont été répartis selon les fonctions de chaque utilisateur, en tenant compte de la définition du poste dans le diagramme de déploiement détaillé déjà établi. Ainsi nous avons une première ébauche des interfaces définie de la manière suivante selon le regroupement de responsabilité sur le diagramme suivant :

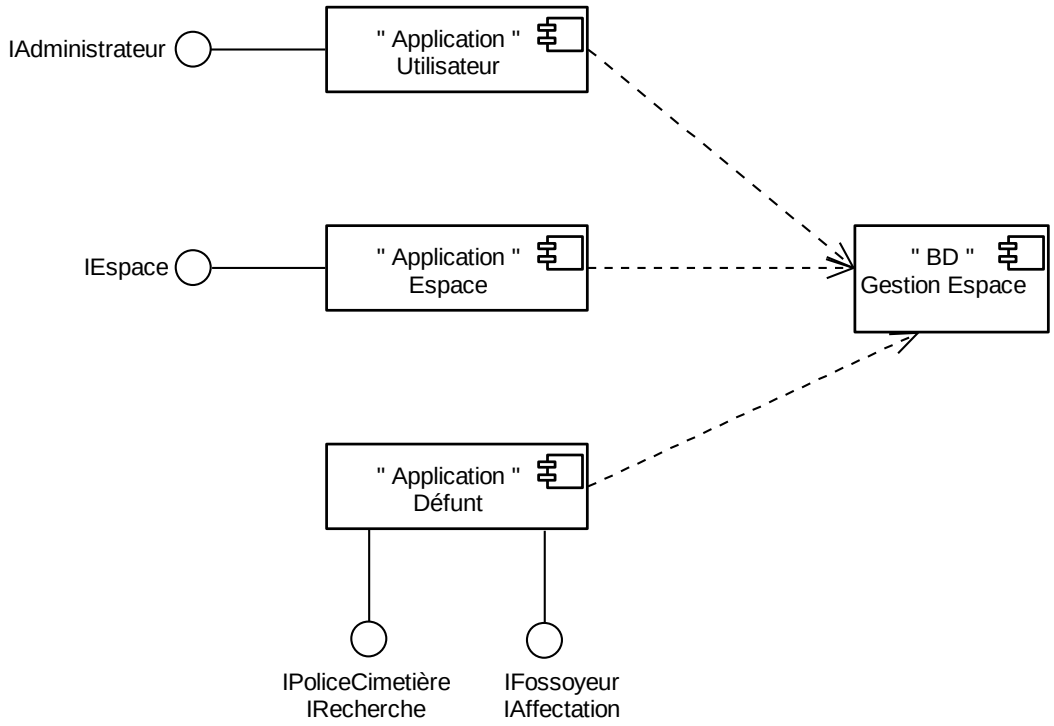

Figure 20 : Diagramme de consolidation de l'application des Espaces

Tableau 2 : Illustration de diagramme de consolidation de l'application

Composant	Interface	Description
Utilisateur	/Administrateur	Création, suppression, rechercher modification utilisateurs
Espace	/Espace	Création, Affectation, visite, Demande espace
Défunt	/ RechercheDefunt	Recherche d'un espace
	/ AffetationDefunt	Affectation d'un défunt dans un espace.

5. Développement du modèle logique de conception

Le développement du modèle logique de conception développe les classes nécessaires au codage. Une catégorie de conception est une catégorie qui organise des classes techniques et contribue à la réutilisation et à l'optimisation d'un système à développer. C'est ici que le découpage du système en sous-système peut se réaliser. Il s'agit d'étudier chaque package (composant) indépendamment des autres, par rapport aux 5 couches logiques. Pour notre cas, les composants de la Gestion des espaces aurons, par exemple, pour la couche présentons les catégories suivantes :

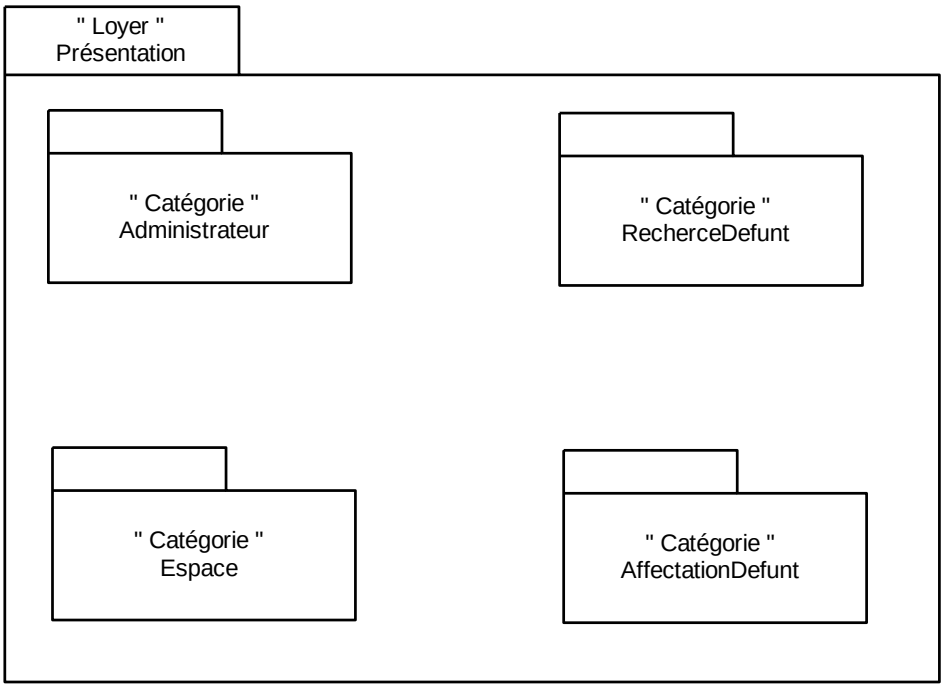

Figure 21 : Identification des catégories de la couche présentation

6. Création des interfaces des catégories

Les interfaces de catégories se construisent à partir des interfaces déjà identifiées des composants d'exploitation. Leur définition nécessite d'être précise et doit prendre en compte des opérations identifiées dans le modèle d'analyse. Ces opérations sont réparties sur les couches applications, métier ou accès aux données selon leur spécialité. Le tableau ci-dessous donne quelques opérations d'analyse de gestion des espaces.

Tableau 3 : tableau d'opération d'analyse de gestion des espaces

Opération d'analyse	Description	Positionnement
Sélectionner Espace	Sélectionner un espace déclenche l'affichage des prix de cet espace	C'est un service applicatif du composant Espace qui entre dans la définition de l'interface IEspace.
Recherche Espace	Rechercher un espace Cette opération est déclenchée par la saisie du numéro de l'espace	C'est un service applicatif du composant Espace qui entre dans la définition de l'interface IEspace.

Les opérations de la couche métier du composant Espace sont : creation Espace, trouver Espace et les opérations de la couche applicative sont rechercher Espace, selectionner Espace, Attribuer Espace, Accorder visite Espace.

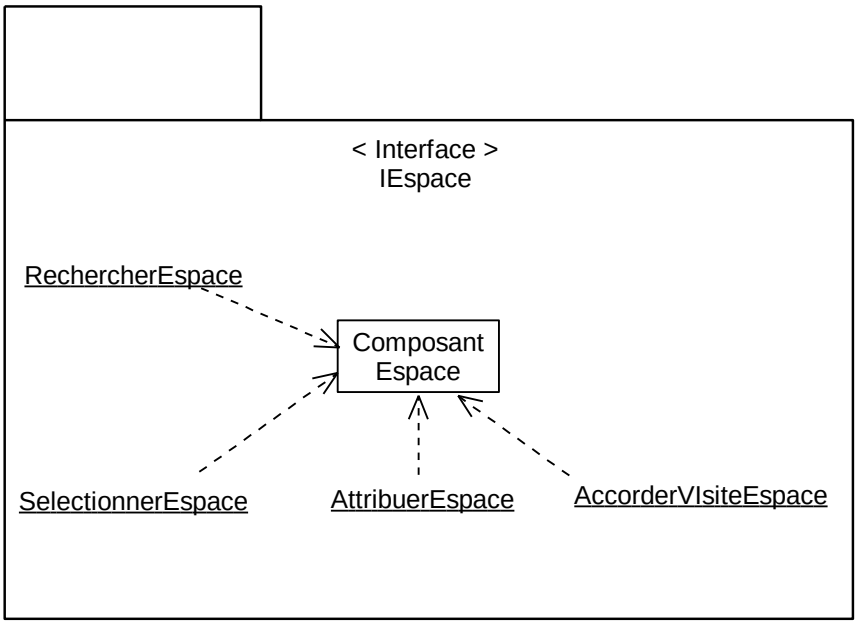

Figure 22 : Identification des catégories de l'interface Espace

7. Conception détaillée

Selon Pascal Roques et Franck Valée définissent la conception détaillée comme étant une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. Ils poursuivent en disant que le modèle logique particulièrement important dans la mesure où c'est en conception détaillée que l'on génère le plus gros volume d'informations. Nous construisons les classes, les vues d'IHM, les interfaces, les tables et les méthodes qui sont donnés en image « prête à coder » de la solution. Et en dernier lieu, nous devons préciser le contenu des sous-systèmes de manière à compléter la configuration logicielle. Toutes les questions relatives à l'agencement et aux détails de la solution doivent être modélisées ainsi que les interrogations restantes concernant la bonne utilisation des langages et des outils de développement. Le niveau d'abstraction visé par l'étape de conception détaillée est d'avoir une idée la plus précise possible pour la fabrication et l'assemblage des sous-systèmes de configuration logicielle. La conception précède la phase de codage.

8. Conception détaillée du modèle logique

Le microprocessus de conception logique concerne la définition des classes à implémenter. C'est donc une activité centrée sur le modèle logique, qui combine les diagrammes UML suivants :

- Principalement les diagrammes des classes pour préciser la structure des classes de développement ;
- Mais aussi les diagrammes d'interactions pour préciser les communications entre objets ;
- Et les diagrammes d'activités pour exprimer les algorithmes des méthodes.

Ce microprocessus est une itération des étapes suivantes :

- La conception des classes ;
- La conception des associations ;
- La conception des attributs ;
- La conception des opérations ;
- La validation du modèle.

Appliquer ces étapes à la gestion des espaces, nous avons le diagramme de classes détaillées suivant :

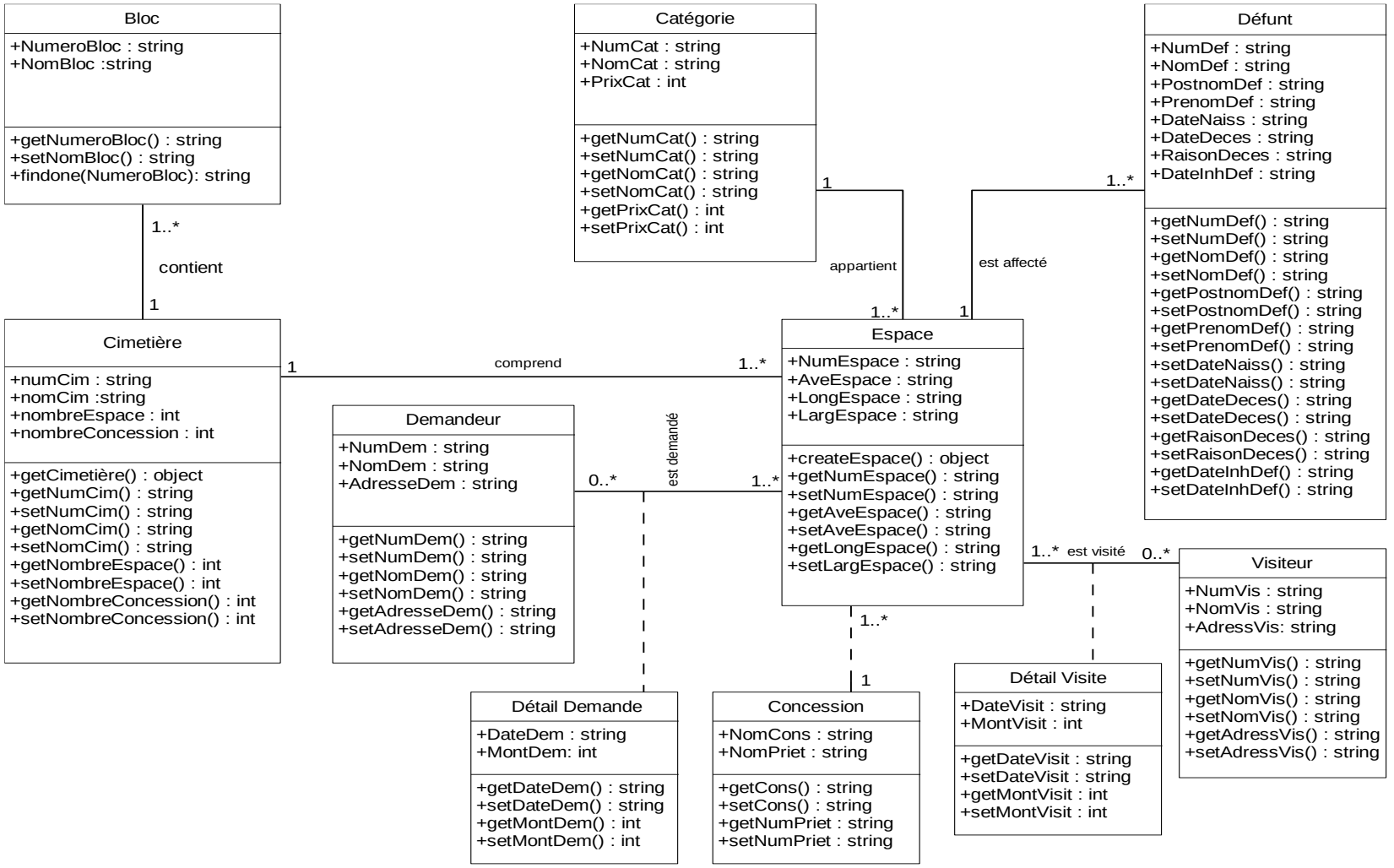

Figure 23 : Diagramme des classes de conception

9. Conception de la couche présentation

Selon Franck Valée, la première étape de conception d'une IHM concerne la définition visuelle des fenêtres ou des pages. L'existence d'un modèle d'analyse permet d'influencer cette conception : à partir d'un diagramme de classes, un générateur de code pourrait générer des fenêtres ou des pages pour l'affichage et la saisie de chaque élément du modèle.

10. Conception de la couche persistance

La réalisation d'un stockage retenu. Dans tous les cas, la réalisation d'un modèle objet facilite la maintenance des données stockées. Selon Pascal Roques, il existe aujourd'hui plusieurs modèles de stockage possibles.

- Le système de fichiers est le moyen le plus rudimentaire de stockage. Cependant, il ne permet que de lire ou d'écrire une instance par des moyens externes à l'application et il n'a aucune capacité à administrer ou à établir des requêtes complexes sur les données.
- La base des données relationnelles ou SGBDR est un moyen déjà plus sophistiqué. Il existe aujourd'hui une large gamme de SGBDR répondant à des besoins de volume, de distribution et d'exploitations différentes. Le SGBDR permet d'administrer les données et d'y accéder par des requêtes complexes.
- La base de données XML ou SGBDX est un concept émergeant qui répond au besoin croissant de stocker des documents XML sans risque d'altération de ces derniers.

Ainsi, cette conception de la couche persistance revient à la conception du mode de stockage de données consistant à étudier sous quelle forme les instances sont sauvegardées sur un support physique. Pour ce faire, en ce qui concerne notre cas d'étude, nous retenons le mode de stockage le plus répandu, le SGBDR. Vu que les modèles développés dans la conception sont des modèles objets d'où, il faut utiliser des principes de rapprochement objet relationnel.

Le passage du Modèle Objet au Modèle Relationnel

L'utilisation d'un système de Gestion des Bases de Données Relationnelles (SGBDR) impose un changement de représentation entre la structure des classes et la structure des données relationnelles, les deux structures ayant des analogies des équivalences. Une classe définit une structure de donnée à laquelle souscrivent des instances, elle correspond donc à une table du modèle relationnel : chaque attribut donne lieu à une colonne, chaque instance stocke ses données dans une ligne et son id sert de clé primaire. Selon le type d'association il y a création de(s) donnée(s) étrangère(s) dans certaines relations, fusion de certaines classes en une relation. Certains attributs de type complexe ne correspondent à aucun des types de SQL, on rencontre fréquemment ce cas pour les attributs représentant une structure de données. Un type complexe peut être conçu :

- Soit avec plusieurs colonnes, chacune correspondant à un champ de la structure ;
- Soit avec une table spécifique dotée d'une clé étrangère pour relier les instances aux valeurs de leur attribut complexe.

UML définit spécifiquement un stéréotype table pour représenter la table d'un schéma relationnel. En appliquant les règles de transformation à la gestion des espaces, nous avons les relations suivantes :

Bloc(numeroBloc, NomBloc, numcim#)
Cimetiere(numcim, nomcim, nombreespace, nombrecoucession)
Categorie(Numcat, nomcat, prixcat)
Espace(Numespace, aveespace, longespace, largespace, numcim#, numcat#, numcons#)
Defunt(numdef, Nomdef, postnomdef, prenomdef, datenaiss, datedeces, raisondeces, dateInDef, numespace#)
Demandeur(Numdem, Nomdem, Adressedem)
DetailDemande(Numdem#, Numespace#, DateDem, MontDem)
Concession(NomCons, Nomprop)
Visiteur(Numvis, NomVis, AdressVis)
DetailVis(NumVis#, NumEspace#, DateVis, MontVis)

11. Conception de la couche applicative

Le rôle de la couche de l'application consiste à piloter le processus d'interaction entre l'utilisateur et le système, il s'agit généralement de mettre en œuvre toutes les règles nécessaires au maintien d'une application cohérente et à l'optimisation des échanges client / serveur et / ou des requêtes http.

D'une manière claire, les mécanismes d'une application assurent :

- L'existence de différentes fenêtres ou pages synchronisées sur les mêmes données ;
- La cohérence entre les objets distribués et les multiples façons de les représenter au travers des IHM ;
- L'optimisation des chargements pour un déploiement en client léger ou sur le poste client pour un déploiement client / serveur ;
- La mise en œuvre des concepts applicatifs : typiquement les sorties des rapports sur imprimantes.

10.1 Dans la couche présentation

- L'interface transforme les évènements de l'utilisateur en action ;
- Le contrôleur centralise l'action déclenchée depuis l'IHM, il vérifie si toutes les règles sont respectées et ensuite il crée la commande correspondante à l'action.

10.1 Dans la couche métier

- L'invocateur exécute la commande ;
- Il restitue la réponse au contrôleur qui l'envoie à l'interface.

Ces différents états, pour chaque diagramme de séquences de tous les scénarios donnent des algorithmes. Ainsi, nous formalisons ces algorithmes par des diagrammes d'activités par colonne des responsabilités. Voici un diagramme illustrant les responsabilités et l'imbrication des déclenchements entre les commandes applicatives du cas de l'attribution d'un espace.

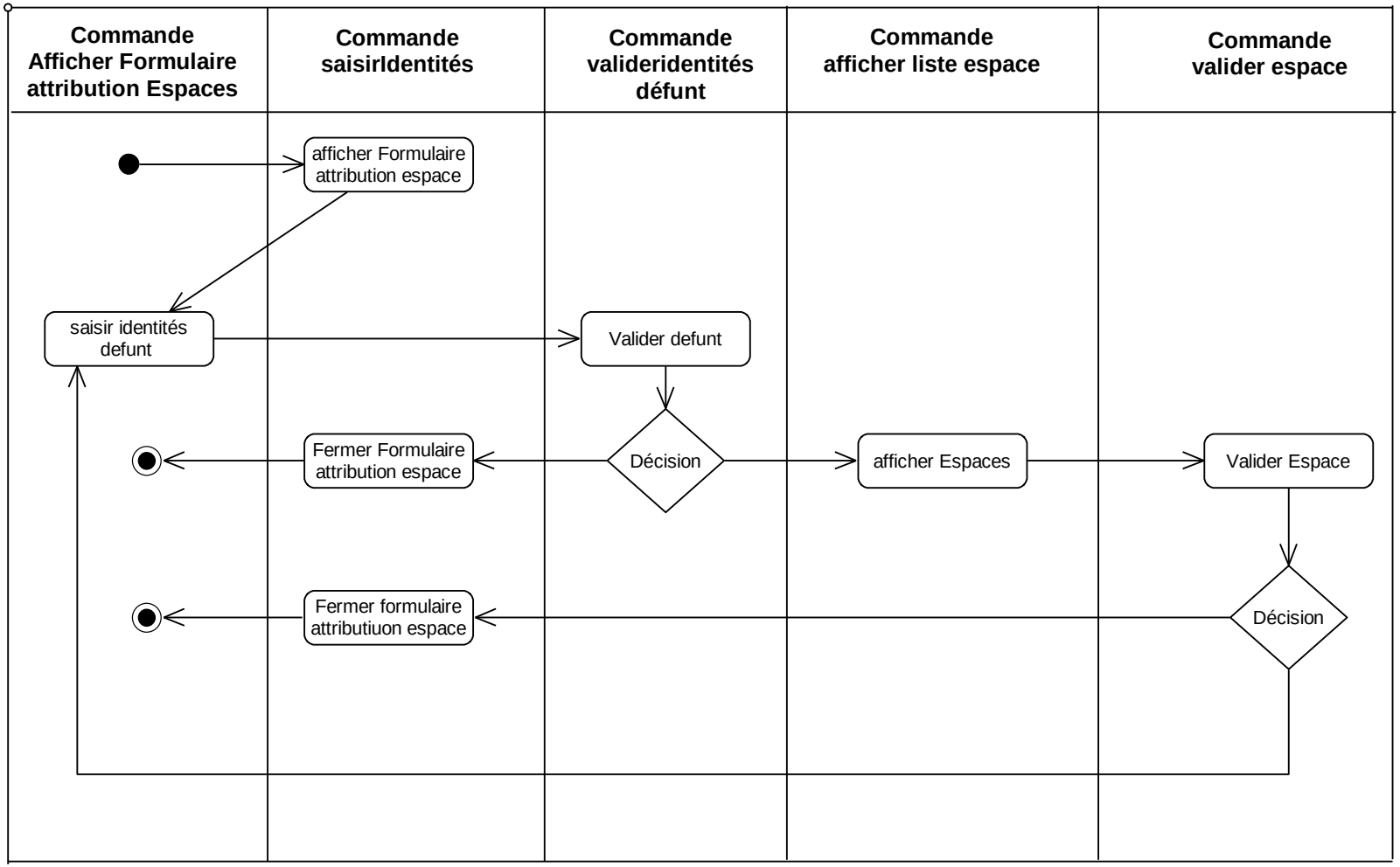

Figure 24 : Algorithme du scénario AttributionEspace

12. Conception de la couche métier distribuée

Selon Pascal Rogues la conception de la couche métier consiste à identifier les objets entités et sessions qu'il convient de développer au vu des classes et des opérations métier à distribuer. Le concept d'objets distribués pose également le problème de la distribution des lieux d'un ensemble d'objets. Pour cela, deux techniques s'opposent :

– Soit les objets sont distribués unitairement et la demande d'un objet lié ou de sa référence déclenche une nouvelle demande de transaction. Cette pratique convient bien à l'édition d'objets métier.

– Soit un ensemble complet d'objets liés et préalablement chargé dans le contexte mémoire de la couche présentation.

Cette technique s'utilise de préférence pour réaliser la présentation des éléments composants. La distribution du métier nous pousse à étudier quel type d'architecture client / serveur nous devons choisi pour la Gestion des Espaces afin de mieux gérer les transactions entre le client et le serveur.
Les 4 modèles de répartition d'une architecture clients/serveurs sont :

– Client/serveur de présentation : c'est dit d'un système qui utilise la présentation au niveau du client, et le code applicatif ainsi que les données sur le serveur.

– Client/serveur de données : c'est un système qui utilise la présentation et le code applicatif au niveau du client et au niveau du serveur, il n'y a que les données.

– Client/serveur de procédures : est un système informatique qui utilise la présentation et une partie du code applicatif au niveau client et le niveau serveur utilise l'autre partie du code applicatif et les données.

– Client/serveur système réparti : c'est un système qui utilise la présentation, une partie du code applicatif et une partie des données au niveau client et au niveau serveur il utilise l'autre partie du code applicatif et toutes les données.

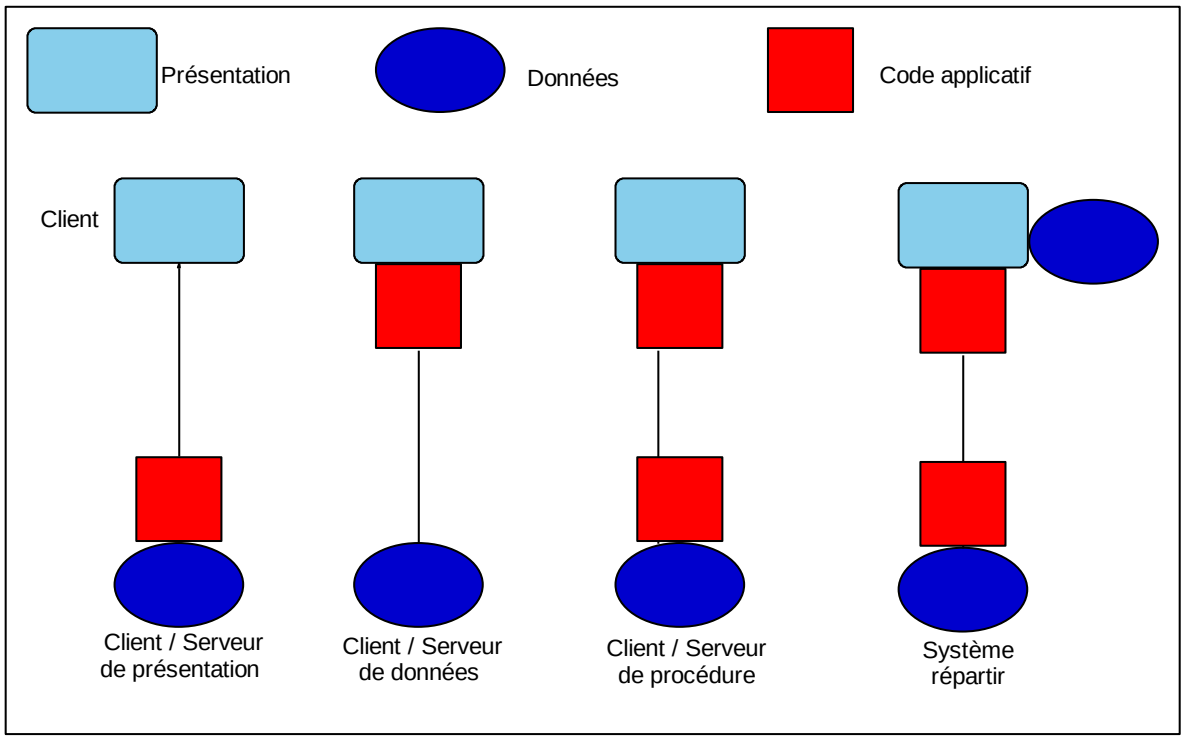

Figure 25 : Architecture Client / Serveur

Ainsi, pour notre cas d'étude nous optons pour le Client/serveur système réparti car la ville de Lubumbashi compte plusieurs cimetières contrôlés par un seul bureau à la mairie.

13. Conclusion

En conclusion, la gestion des espaces dans un cimetière au bureau tutelle de l'administration générale de la mairie de la ville de Lubumbashi a souvent des problèmes d'occupations des espaces, d'aménagement et de déménagement suite au manque d'urbanisation des cimetières. Face à cette réalité, nous avons fait l'analyse du système afin de voir dans quelle mesure l'outil informatique peut apporter une solution à ce problème. De cette analyse métier, nous avons abouti à la conception d'un système informatique (application) de gestion des espaces dans un cimetière. Ce système permet d'attribuer un espace, une concession, elle permet de suivre le service d'inhumation depuis le dépôt de l'acte de décès jusqu'à l'enterrement et permet aussi de suivre les visites dans les cimetières. Ce système doit être utilisé sous une architecture client / serveur des données, seules les données sont partagées entre les utilisateurs du système (secrétaire / section inhumation, administrateur système, la police du cimetière, le fossoyeur). Dans le cadre où le système informatique devrait être utilisé dans tous les cimetières de la ville de Lubumbashi la meilleure solution serait un système client / serveur réparti avec la géolocalisation associée que nous n'avons pas pu mettre en place suit aux coûts que cela nous demandait mais que l'on peut envisager faire ou mettre en place. Ainsi, partant d'un système réparti, chaque utilisateur aura à traiter et stocker ses données à son niveau quitte à consulter le serveur pour appeler les données qui ne sont pas dans son domaine d'action et selon les responsabilités.

Les fonctionnalités qu'offrirait le logiciel sont :

- L'attribution d'un espace ;
- L'attribution d'une concession ;
- L'accord des visites ;
- La recherche des espaces ;
- La sécurité avec gestion des profits d'utilisateur.

14. REFERENCES

1. Laurent Audibert, *cours UML2.0*, Institut Universitaire de technologie de Villetaneuse, département Informatique.
2. Pascal Rogue et Franck Vallée, *UML2 en action : De l'analyse des besoins à la conception*, éd. Eyrolles, 2007.
3. Pernet, J., & Tabeaud, M. (2001). LES CIMETIÈRES FRANCILIENS: DES PROBLÈMES DE GESTION DIFFÉRENTS SELON LA TAILLE DES COMMUNES. *La mort en Ile-de-France*, *19*, 21.
4. Assonsi, S. O. M. A. (2022). Les cimetières, des oubliés dans la gouvernance urbaine en Afrique: une analyse à travers la commune urbaine de Ouagadougou au Burkina Faso. *Bulletin de la Société Géographique de Liège*.
5. SIMBU, A. V., ZUIZANA, C. D., PUELA, F. P., NZUZI, F. L., KOMANDA, J. A., MUKOKO, Y. A., & UMBA-di-MBUDI, C. N. (2022). IMPACT DE LA MAUVAISE GESTION DU CIMETIERE MBENSEKE FUTI DANS LA VILLE DE KINSHASA. *Revue Internationale du Chercheur*, *3*(3).
6. Benkouachi, N. E., & Alatou, D. (2017). Le sig et la gestion des espaces verts de la ville d'elkhroub. *Sciences & Technologie. C, Biotechnologies*, 17-24.