

Système de commandes OCTAVE pour la société BRICOVPC

Correction

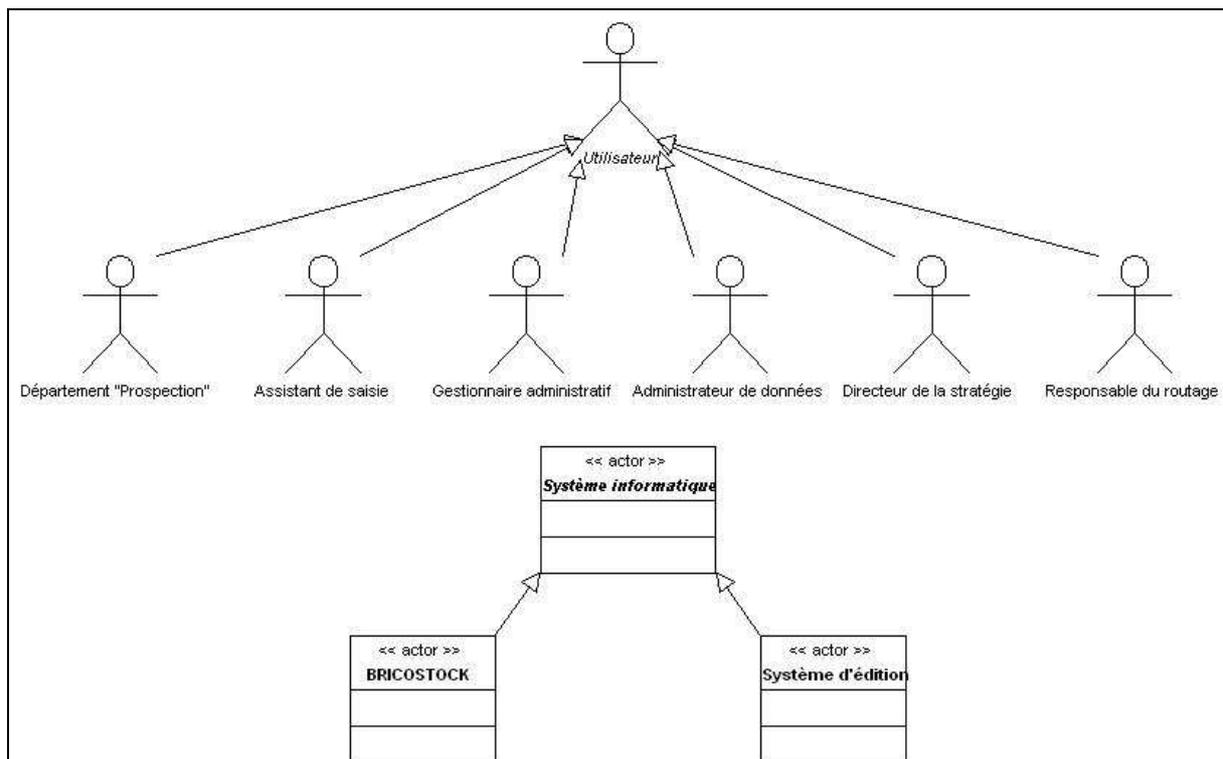
Avertissement : comme pour toute activité de modélisation, il n'existe pas UNE solution unique. Ce document présente donc une solution possible. Comparer la solution proposée avec celles trouvées par les différents groupes. Pour chaque nouvelle étape, partir de la solution proposée dans ce document afin que chaque groupe ne diverge pas trop des autres. Ce document est un document Word dans lequel on a inséré des modèles UML.

1. Architecture fonctionnelle

A. Acteurs et contexte

Les acteurs du système sont les suivants :

- **Acteurs « humain »** : Département Prospection, Assistant de saisie, Gestionnaire administratif, Administrateur des données, Directeur de la stratégie, Responsable du routage.
- **Acteurs « système »** : BRICOSTOCK, Système d'édition.



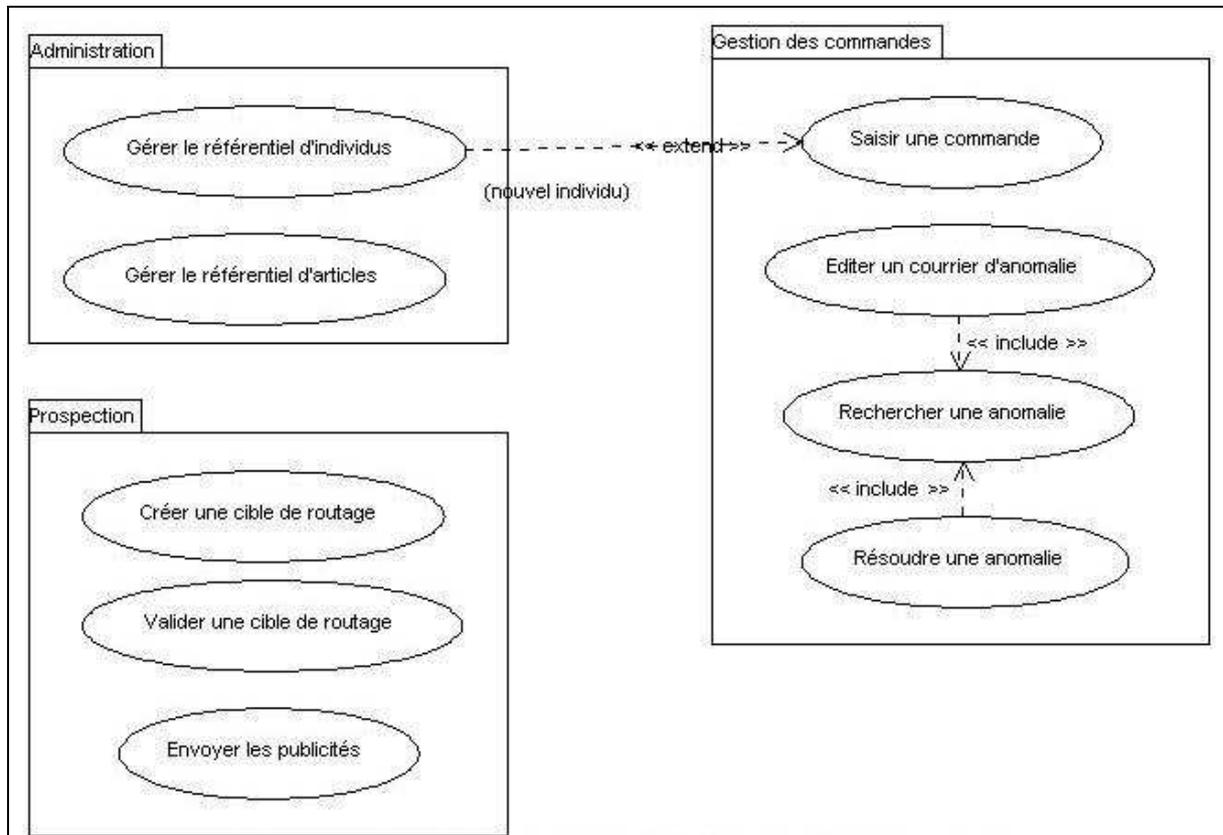
Note : dans le méta modèle UML, le concept d'acteur est en fait un stéréotype prédéfini de classe, avec une représentation graphique standard qu'il est possible de substituer à la notation textuelle du stéréotype. Sur le diagramme présenté ci-dessus, le choix a été fait de représenter les acteurs « humain » sous la forme du « stick man », et les acteurs « système » sous la forme classique d'une classe stéréotypée.

B. Cas d'utilisation

B1. Identification des cas d'utilisation

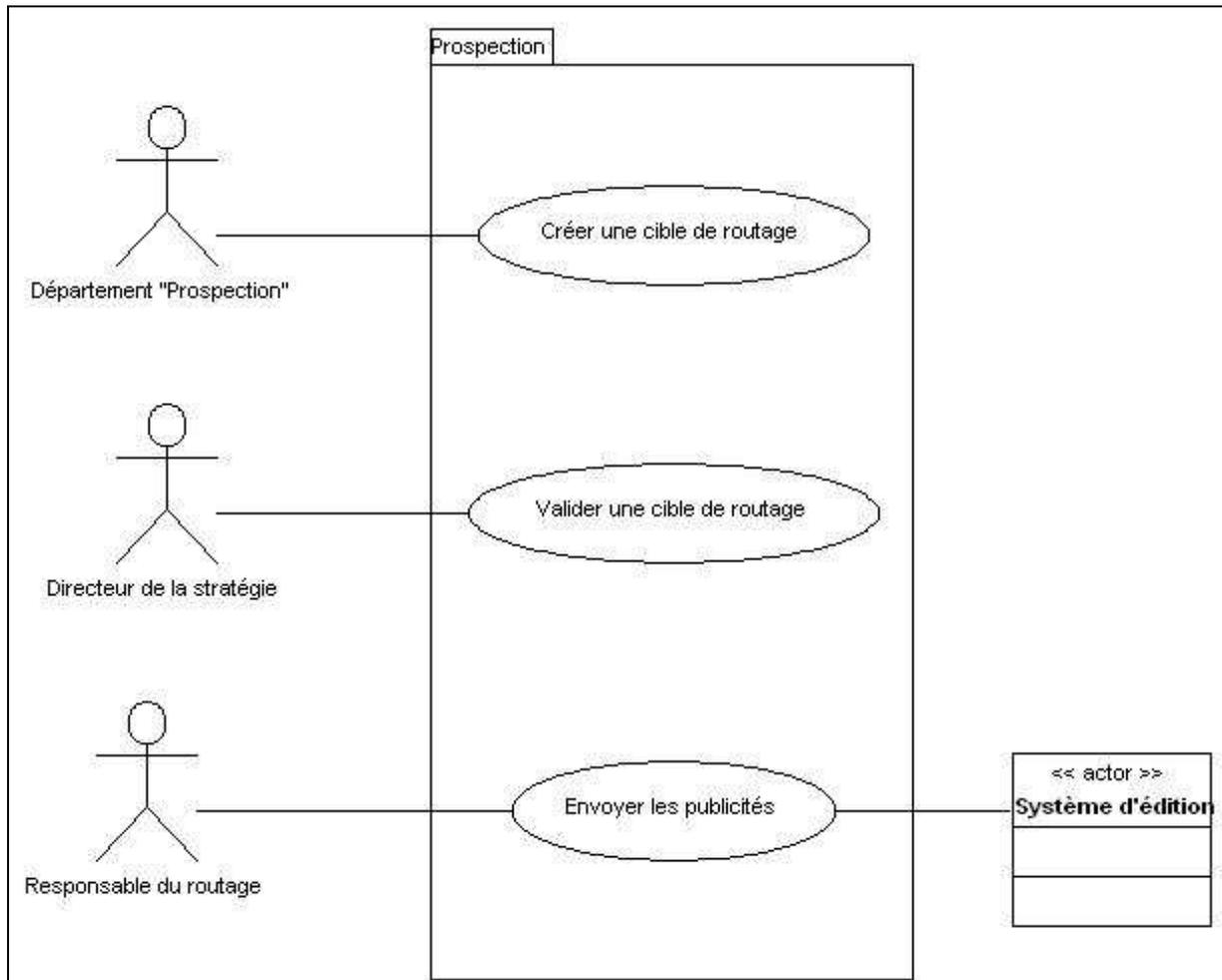
Les cas d'utilisation du système OCTAVE sont organisés en trois packages :

- Prospection,
- Gestion des commandes,
- Administration.

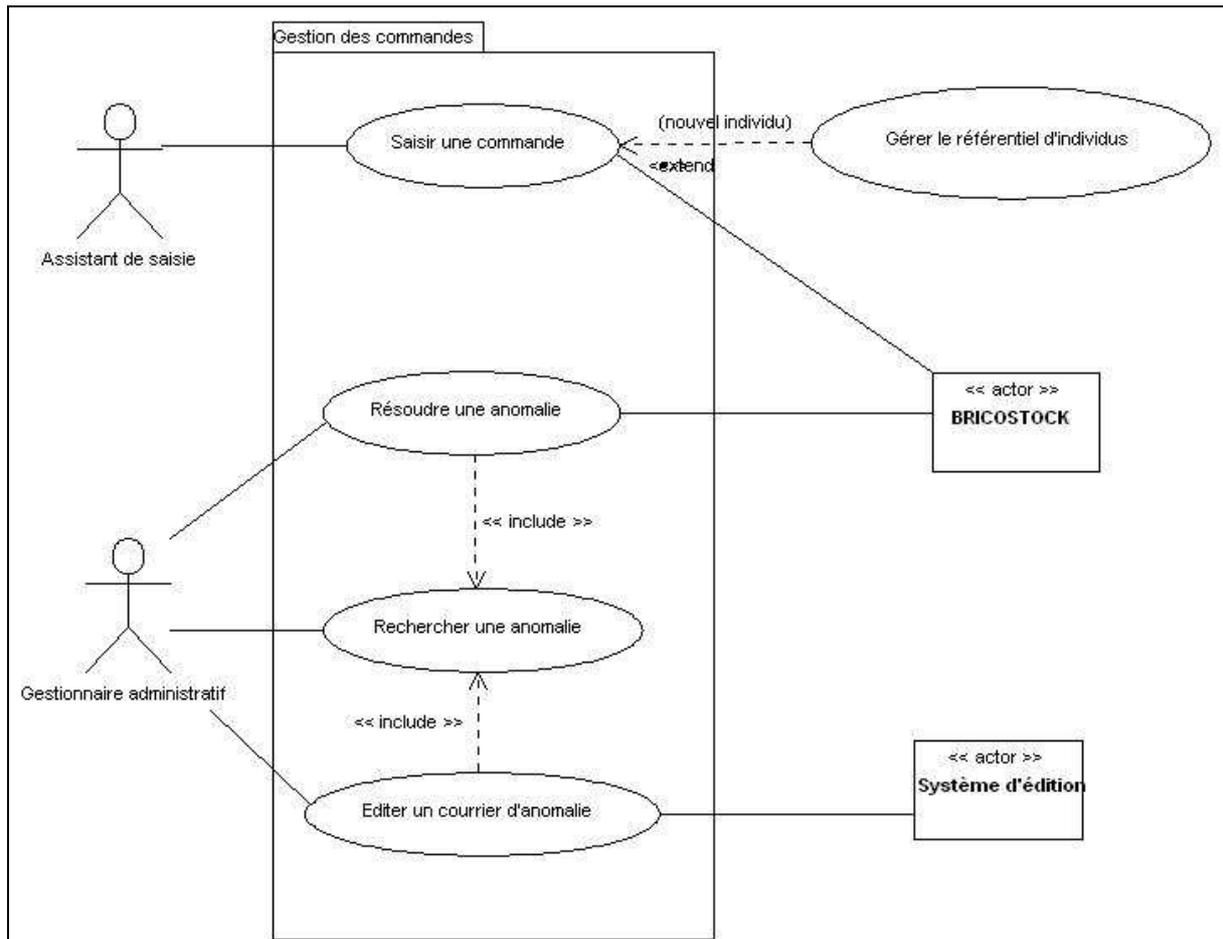


Le diagramme présenté ci-dessus fournit une vue globale de l'application. L'ensemble des cas d'utilisation y est représenté, organisé en packages.

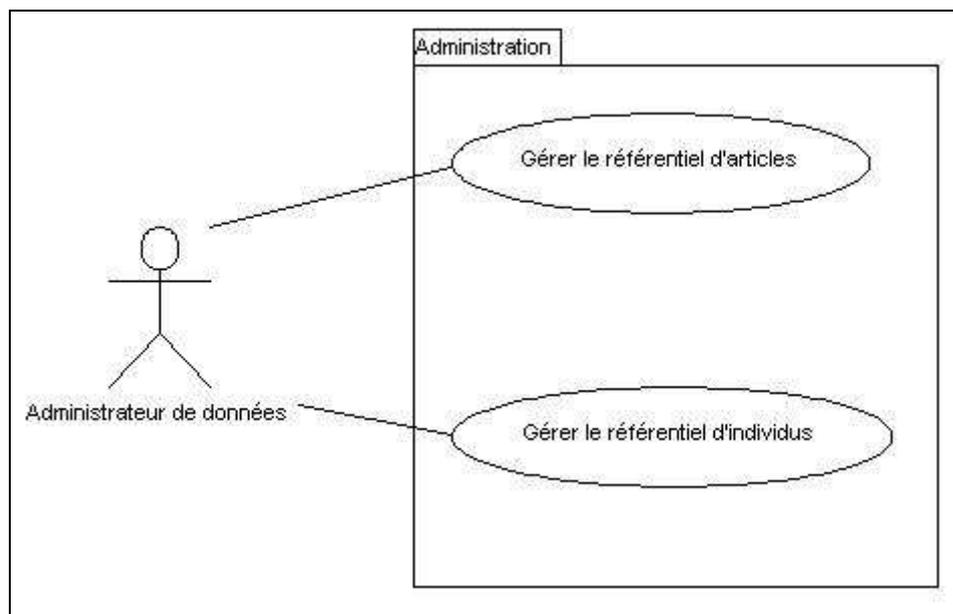
Package Prospection



Package Gestion des commandes



Package Administration



Il faut éventuellement prévoir d'autres cas d'utilisation pour faire fonctionner le système, même si ceux-ci ne sont pas cités ni décrits dans le cahier des charges. Par exemple un cas d'utilisation

de définition et gestion des habilitations, un cas d'utilisation de suppression des cibles de routage, ...

B2. Description des cas d'utilisation

Note : tous les cas d'utilisation identifiés dans les diagrammes précédents sont décrits dans une fiche de description textuelle. Cette fiche de description n'est pas normalisée par UML, mais la plupart des ouvrages traitant d'UML donnent un plan de description des cas d'utilisation semblable à celui utilisé dans la suite du document.

Titre du cas d'utilisation : Créer une cible de routage

Référence : UC-01

Objectif : permettre à un acteur de sélectionner des individus afin de leur envoyer une publicité.

Résumé : l'acteur crée la cible de routage en sélectionnant la publicité et les individus.

Acteurs : département « Prospection »

Pré-conditions : aucune.

Description des enchaînements :

Enchaînement nominal :

Ce cas d'utilisation commence lorsque l'acteur demande à créer une nouvelle cible de routage.

L'acteur saisit les critères de sélection des individus. Il renseigne tout ou partie des critères suivants : catégorie socio-professionnelle, âge, département de résidence, individu déjà client ou non.

L'acteur crée une publicité. Pour cela, il sélectionne le support : papier ou message par internet, puis saisit un titre, une description, et sélectionne les articles à inclure dans la publicité. Si la publicité est envoyée sous forme papier, alors l'acteur sélectionne la qualité du papier : standard, supérieur ou économique.

Le système vérifie que le nombre d'articles sélectionnés est compris entre un et cinq. Si ce n'est pas le cas, l'acteur est invité à modifier sa sélection d'articles.

Le système crée la nouvelle cible de routage.

Enchaînements alternatifs : aucun.

Enchaînements d'exception : aucun.

Post-conditions : une nouvelle cible de routage est créée.

Titre du cas d'utilisation : Valider une cible de routage

Référence : UC-02

Objectif : permettre à un acteur de valider une cible de routage précédemment créée (dans UC-01).

Résumé : l'acteur sélectionne une cible de routage, la consulte et la valide.

Acteurs : directeur de la stratégie

Pré-conditions : il existe une cible de routage à valider.

Description des enchaînements :

Enchaînement nominal :

Ce cas d'utilisation commence lorsque l'acteur demande à valider une cible de routage.

L'acteur sélectionne une cible de routage.

L'acteur peut consulter le détail de la cible de routage : la date de création de la cible, la liste des individus associés, les caractéristiques de la publicité, la liste des articles associés.

L'acteur valide la cible de routage sélectionnée.

Le système vérifie que la cible n'a pas déjà été validée. Si c'est le cas, le système prévient l'acteur que la cible sélectionnée ne peut être validée, et l'invite à sélectionner une autre cible.

Sinon, le système valide la cible de routage.

Enchaînements alternatifs : aucun.

Enchaînements d'exception : aucun.

Post-conditions : une cible de routage non validée à été validée.

Titre du cas d'utilisation : Envoyer les publicités

Référence : UC-03

Objectif : envoyer une publicité à un groupe d'individus, suite à la création et validation d'une cible de routage (UC-01 et UC-02).

Résumé : l'acteur sélectionne une cible de routage, la consulte et lance l'envoi de la publicité.

Acteurs : responsable du routage

Pré-conditions : il existe une cible de routage validée qui n'a pas fait l'objet d'un envoi de publicité.

Description des enchaînements :

Enchaînement nominal :

Ce cas d'utilisation commence lorsque l'acteur demande à envoyer une publicité associée à une cible de routage.

L'acteur sélectionne une cible de routage.

L'acteur peut consulter le détail de la cible de routage : la date de création de la cible, la liste des individus associés, les caractéristiques de la publicité, la liste des articles associés.

L'acteur valide la cible de routage sélectionnée.

Le système vérifie que la cible est validée, et que la cible n'a pas déjà fait l'objet d'un envoi de publicité.

Si c'est le cas : le système prévient l'acteur que la publicité ne peut être envoyée, et l'invite à sélectionner une autre cible.

Sinon :

- le système transmet au système d'édition les informations concernant l'envoi de la publicité : le support d'envoi de la publicité (papier et choix de la qualité, ou message par internet), la liste des individus et leurs caractéristiques, le titre et la description de la publicité, et le nom et le prix des articles de la publicité.
- Pour que la publicité ne puisse être envoyée plusieurs fois, le système enregistre que l'envoi de la publicité associée à la cible sélectionnée a été effectué.

Enchaînements alternatifs : aucun.

Enchaînements d'exception : aucun.

Post-conditions :

- un fichier d'envoi de publicité a été transmis au système d'édition.
- une cible validée a été marquée comme ayant fait l'objet d'un envoi de publicité.

Titre du cas d'utilisation : Saisir une commande

Référence : UC-04

Objectif : saisir une commande dans le système, à partir d'une commande reçue par courrier postal.

Résumé : l'acteur renseigne l'individu, les articles et les quantités commandées, et les informations de règlement.

Acteurs : assistant de saisie, Système « BRICOSTOCK ».

Pré-conditions : une commande a été reçue par courrier postal.

Description des enchaînements :

Enchaînement nominal :

Ce cas d'utilisation commence lorsque l'acteur demande la saisie d'une nouvelle commande.

L'acteur saisit les informations concernant l'individu qui a effectué la commande.

Le système vérifie que l'individu est connu du système. Si ce n'est pas le cas, le système propose à l'acteur de créer un nouvel individu. [**Extension : cas d'utilisation UC-09**].

L'acteur saisit la référence et la quantité de chaque article commandé.

L'acteur saisit les informations de règlement : type de règlement, et en fonction du type de règlement :

- Pour un règlement par chèque : numéro de chèque, nom de la banque émettrice du chèque, indication si la chèque est signé ou non,
- Pour un règlement par carte bancaire : numéro et date d'expiration de la carte.

Le système calcule un numéro de commande, et crée une commande en lui affectant le numéro calculé.

Le système vérifie que le montant réglé est bien égal au montant attendu. Si ce n'est pas le cas, [**Enchaînement d'exception 1**].

Si l'acteur a indiqué que le chèque n'était pas signé, ou si le numéro de carte bancaire n'est pas valide, [**Enchaînement d'exception 2**].

Le système transmet la commande au logiciel de gestion des stocks « BRICOSTOCK ».

Le système modifie les caractéristiques de l'individu qui a effectué la commande : si celui-ci n'était pas client, alors il le devient.

Enchaînements alternatifs : aucun.

Enchaînements d'exception :

Enchaînement d'exception 1 :

Le système enregistre une anomalie « Erreur sur le montant », et place la commande en attente.

Fin du cas d'utilisation.

Enchaînement d'exception 2 :

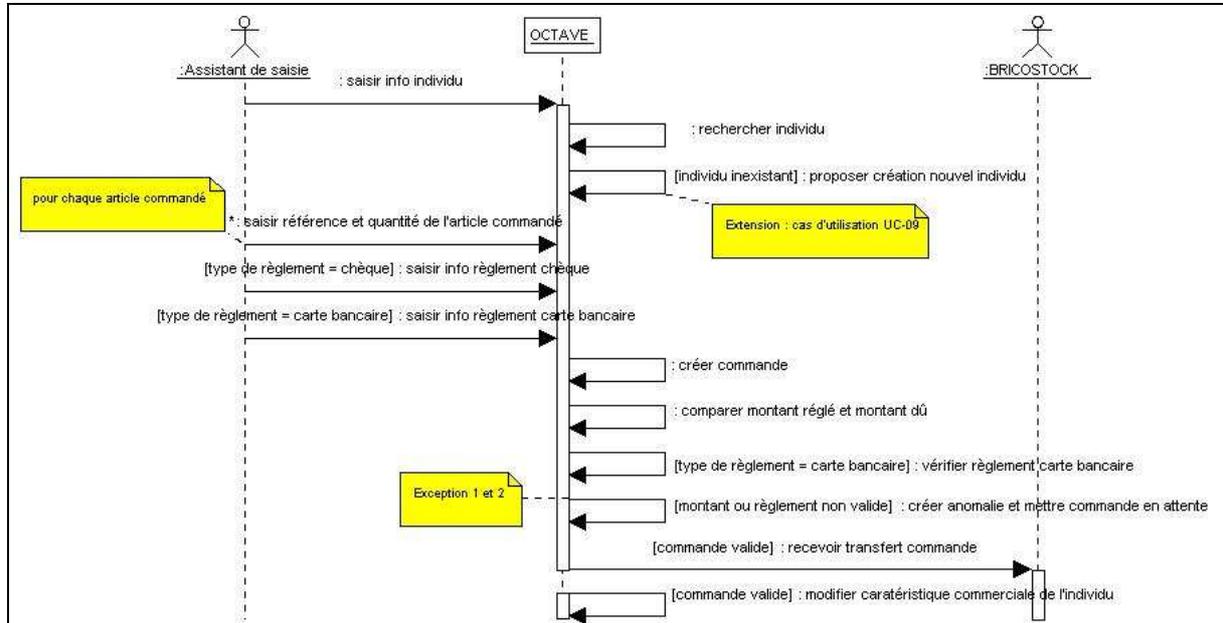
Le système enregistre une anomalie « Problème sur le moyen de paiement », et place la commande en attente.

Fin du cas d'utilisation.

Post-conditions : une nouvelle commande est créée dans le système. Cette commande a été transmise au logiciel de gestion des stocks « BRICOSTOCK ». L'individu qui a effectué la commande est devenu « client », s'il ne l'était pas déjà.

Pour les enchaînements d'exception : une nouvelle commande « en attente » est créée dans le système. Une ou deux anomalies sont associées à cette commande.

La description textuelle peut être illustrée par une description sous forme de diagramme de séquence :



Titre du cas d'utilisation : Rechercher une anomalie

Référence : UC-05

Objectif : rechercher et consulter les anomalies.

Résumé : l'acteur saisit des critères de recherche d'anomalies, et consulte la liste ou le détail des anomalies.

Acteurs : gestionnaire administratif.

Pré-conditions : une anomalie existe dans le système.

Description des enchaînements :

Enchaînement nominal :

Ce cas d'utilisation commence lorsque l'acteur demande la consultation des anomalies.

Le système propose à l'acteur une recherche multi-critères d'anomalies : par individu, par numéro de commande, ou par date de génération de l'anomalie.

L'acteur saisit les critères de recherche qu'il désire.

Le système recherche les anomalies correspondantes et les présente à l'acteur.

Enchaînements alternatifs : aucun.

Enchaînements d'exception : aucun.

Post-conditions : aucune modification dans le système.

Titre du cas d'utilisation : Editer un courrier d'anomalie

Référence : UC-06

Objectif : éditer un courrier d'anomalie pour une ou plusieurs anomalies.

Résumé : l'acteur recherche et sélectionne une anomalie, le système génère un courrier d'anomalie et le transmet au système d'édition.

Acteurs : gestionnaire administratif, système d'édition.

Pré-conditions : il existe une anomalie n'ayant pas fait l'objet d'une édition de courrier.

Description des enchaînements :

Enchaînement nominal :

Ce cas d'utilisation commence lorsque l'acteur demande l'édition d'un courrier d'anomalie.

L'acteur recherche l'anomalie pour laquelle il désire envoyer un courrier. [**Inclusion** : UC-05].

Une fois l'anomalie désirée sélectionnée, l'acteur demande l'édition d'un courrier pour cette anomalie.

Le système recherche si, pour la commande associée à l'anomalie, il existe une autre anomalie n'ayant pas fait l'objet d'un courrier.

Si tel est le cas, [**enchaînement alternatif**].

Sinon, le système crée un courrier pour l'anomalie sélectionnée.

Le système transmet les informations concernant le courrier d'anomalie au système d'édition.

Enchaînements alternatifs :

Le système propose à l'acteur d'éditer un courrier pour les deux anomalies associées à la commande.

L'acteur choisit une option parmi :

- éditer le courrier pour l'anomalie sélectionnée. Le système crée alors un courrier d'anomalie pour cette anomalie.
- éditer un courrier pour les deux anomalies associées à la même commande. Le système crée alors un courrier d'anomalie pour les deux anomalies.

Poursuite de l'enchaînement nominal.

Enchaînements d'exception : aucun.

Post-conditions : un courrier d'anomalie a été créé, les informations concernant ce courrier ont été transmises au système d'édition.

Titre du cas d'utilisation : Résoudre une anomalie

Référence : UC-07

Objectif :

Résumé : l'acteur recherche et sélectionne une anomalie, le système enregistre la résolution de l'anomalie, la commande associée devient éventuellement valide.

Acteurs : gestionnaire administratif, Système « BRICOSTOCK ».

Pré-conditions : il existe une anomalie non résolue.

Description des enchaînements :

Enchaînement nominal :

Ce cas d'utilisation commence lorsque l'acteur demande la résolution d'une anomalie.

L'acteur recherche l'anomalie qu'il désire résoudre. [**Inclusion** : UC-05].

Une fois l'anomalie désirée sélectionnée, l'acteur demande la résolution de cette anomalie.

Le système enregistre la résolution de l'anomalie, et teste si la commande associée à l'anomalie possède encore une anomalie non résolue.

S'il n'y a plus d'anomalie non résolue, alors :

- la commande devient valide,
- le système transmet la commande au logiciel de gestion des stocks « BRICOSTOCK »,
- le système modifie les caractéristiques de l'individu qui a effectué la commande : si celui-ci n'était pas client, alors il le devient.

Enchaînements alternatifs : aucun.

Enchaînements d'exception : aucun.

Post-conditions :

- une anomalie a été résolue,
- éventuellement une commande a été validée et transmise au système « BRICOSTOCK »,
- éventuellement l'individu associé à la commande est devenu client.

Titre du cas d'utilisation : Gérer le référentiel d'articles

Référence : UC-08

Objectif : mettre à jour le référentiel des articles proposés par la société.

Résumé : l'acteur crée ou modifie un article.

Acteurs : Administrateur de données.

Pré-conditions : aucune.

Description des enchaînements :

Enchaînement nominal :

Enchaînement nominal 1 : ajouter un article

Ce cas d'utilisation commence lorsque l'acteur demande à ajouter un article dans le référentiel des articles.

L'acteur saisit les informations relatives à l'article : numéro, désignation, prix de vente.

Si un article du référentiel possède déjà le numéro saisi, le système en informe l'acteur et l'invite à modifier sa saisie.

Sinon, le système crée le nouvel article dans le référentiel.

Enchaînement nominal 2 : modifier un article

Ce cas d'utilisation commence lorsque l'acteur demande à modifier un article du référentiel des articles.

L'acteur recherche et sélectionne l'article qu'il souhaite modifier.

L'acteur modifie les caractéristiques de l'article : désignation et prix de vente.

Remarque : la modification du prix de l'article ne doit pas avoir de conséquences sur les commandes en cours. Le système doit donc conserver l'historique des prix de l'article.

Enchaînements alternatifs : aucun.

Enchaînements d'exception : aucun.

Post-conditions : un nouvel article a été ajouté au référentiel des articles, ou bien un article du référentiel des articles a été modifié.

Titre du cas d'utilisation : Gérer le référentiel d'individus

Référence : UC-09

Objectif : mettre à jour le référentiel des individus et des clients de la société.

Résumé : l'acteur crée ou modifie un individu.

Acteurs : Administrateur de données.

Pré-conditions : aucune.

Description des enchaînements :

Enchaînement nominal :

Enchaînement nominal 1 : ajouter un individu

Ce cas d'utilisation commence lorsque l'acteur demande à ajouter un individu dans le référentiel des individus.

L'acteur saisit les informations relatives à l'individu : nom, prénom, date de naissance, catégorie socio-professionnelle, adresse, numéro de téléphone, adresse mail.

Le système crée le nouvel individu dans le référentiel. Ce nouvel individu a pour caractéristique commerciale « prospect ».

Enchaînement nominal 2 : modifier un individu

Ce cas d'utilisation commence lorsque l'acteur demande à modifier un individu du référentiel des individus.

L'acteur recherche et sélectionne l'individu qu'il souhaite modifier.

L'acteur modifie les caractéristiques de l'individu : nom, prénom, date de naissance, catégorie socio-professionnelle, adresse, numéro de téléphone, adresse mail.

Enchaînements alternatifs : aucun.

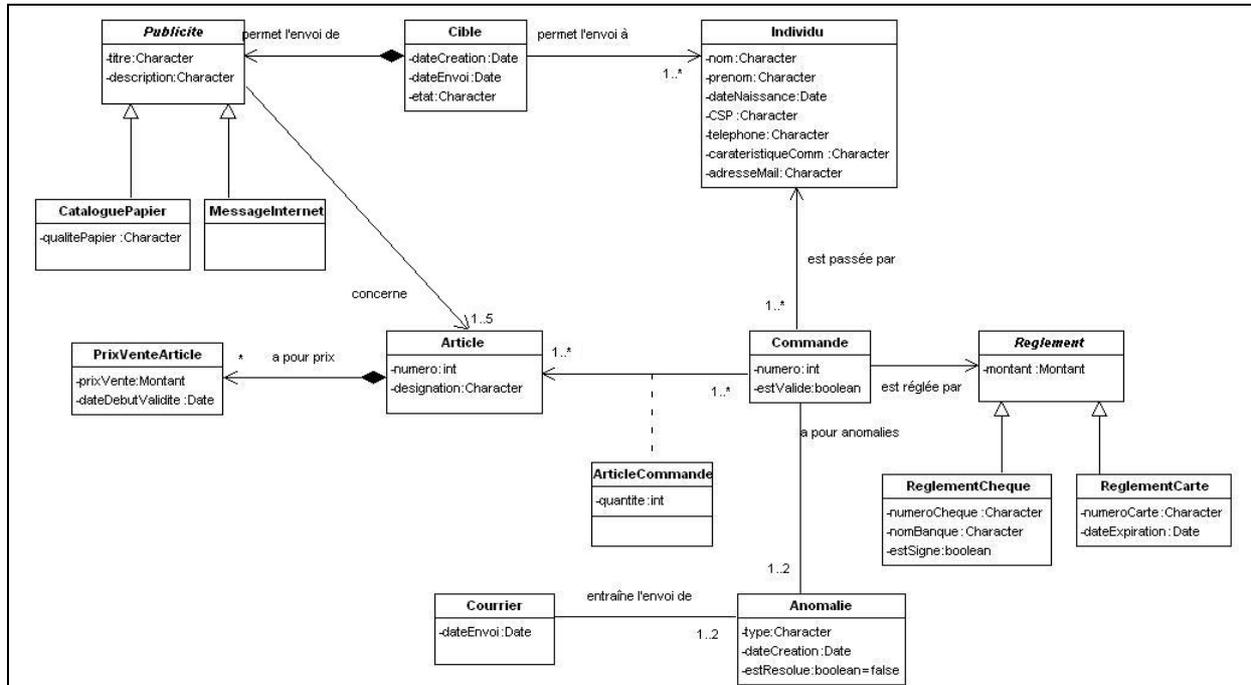
Enchaînements d'exception : aucun.

Post-conditions : un nouvel individu a été ajouté au référentiel des individus, ou bien un individu du référentiel des individus a été modifié.

2. Analyse Objet

A. Modèle statique

La première étape de l'analyse objet consiste à identifier les principales classes du système, leurs propriétés et leurs relations. Ce modèle de classes est ensuite enrichi lors de l'étude des scénarios.



La description des classes permet de comprendre plus rapidement le modèle statique UML.

Publicite : cette classe, abstraite, permet de représenter tout type de document envoyé aux individus. Chaque instance d'une des sous-classes est rattachée à une et une seule instance de la classe Cible.

CataloguePapier et **MessageInternet**: les deux sous-classes de Publicite.

Cible : cette classe représente les envois de publicité à un groupe d'individus.

Individu : cette classe contient l'ensemble des personnes physiques à qui la société envoie ou souhaite envoyer des publicités.

Commande : cette classe permet d'enregistrer certaines informations d'une commande d'articles.

Reglement : cette classe abstraite définit les informations de règlement relatives à une commande.

ReglementCheque : cette classe enregistre les règlements effectués par chèque. Il est possible d'indiquer l'absence de signature du chèque au système, qui génère alors une anomalie.

ReglementCarte : cette classe enregistre les règlements par carte bancaire.

Article : cette classe contient l'ensemble des articles proposés par la société.

PrixVenteArticle : cette classe permet de fixer le prix des articles à une date donnée. Ainsi, le prix d'un article peut varier au cours du temps, cette classe conserve alors l'historique des prix de l'article.

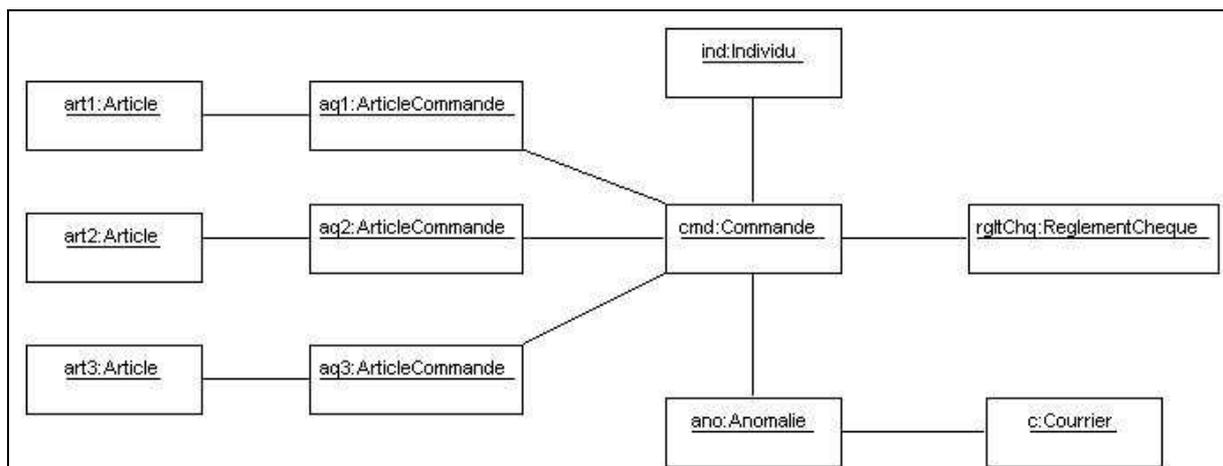
ArticleCommande : cette classe est une classe d'association entre les classes Commande et Article. Elle permet d'enregistrer, pour chaque commande, la quantité de chaque article commandé.

Anomalie : cette classe contient l'ensemble des anomalies associées aux commandes.

Courrier : cette classe permet de connaître les courriers envoyés aux individus lors d'une anomalie liée à une commande.

Le modèle de classe présenté dans le diagramme précédent peut être illustré par un diagramme d'objets. Ce type de diagrammes permet de valider le modèle statique générique à travers un exemple concret.

Le diagramme présente une commande de trois articles, réglée par chèque, et possédant une anomalie ayant fait l'objet d'un courrier.



B. Modèle dynamique

Deux types de diagrammes vont être utilisés pour identifier et définir les principales opérations de chaque classe : le diagramme de séquence et le diagramme d'état.

Diagramme de séquence objet du cas d'utilisation UC-04 « Saisir une commande » :

Le diagramme suivant décrit les interactions entre objets lors de la saisie d'une commande, dans le cas d'un règlement par carte bancaire.

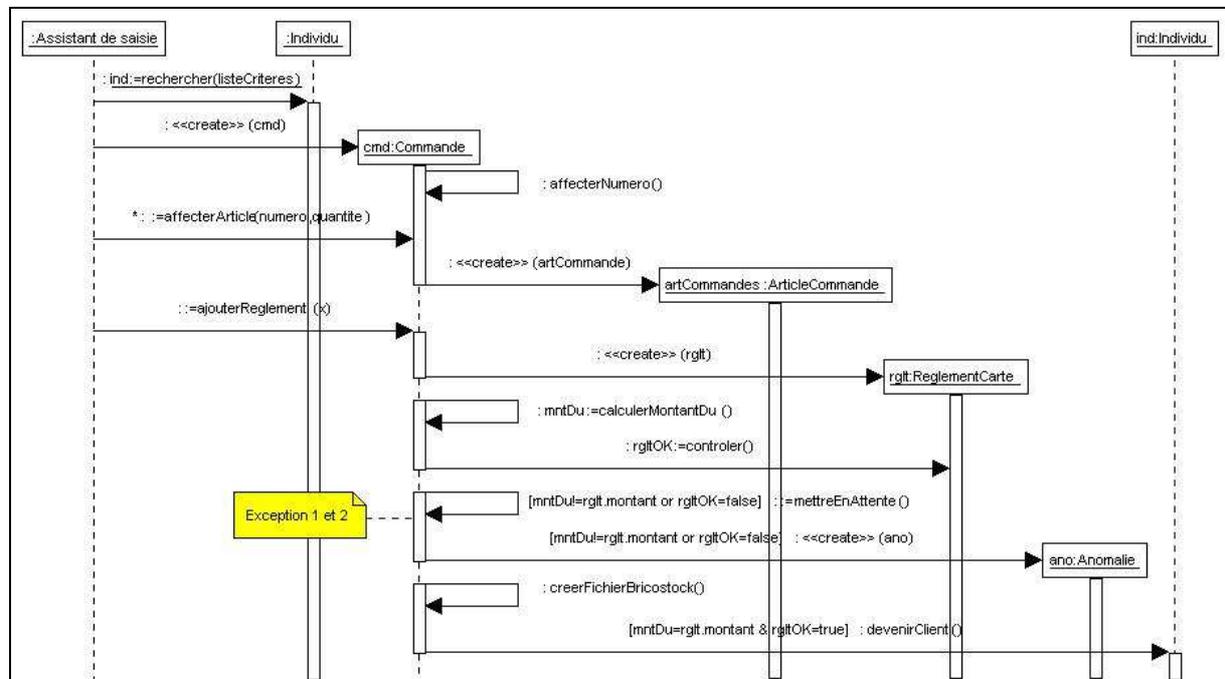
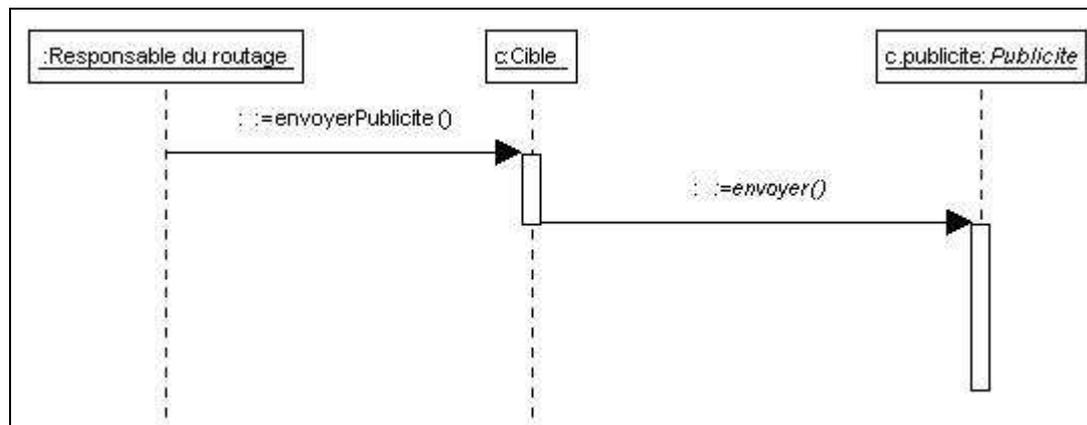


Diagramme de séquence objet du cas d'utilisation UC-03 « Envoyer les publicités » :



Ce diagramme illustre la notion de polymorphisme : lorsque le responsable de routage souhaite envoyer la publicité associée à une cible, la cible invoque l'opération « envoyer() » sur la publicité associée. Peu importe le support d'envoi de la publicité, l'opération « envoyer() » est définie sur chaque sous-classe de la classe Publicite.

Ainsi, si l'on ajoute un nouveau support d'envoi de publicités (par exemple par envoi de SMS), il faudra juste créer une nouvelle sous-classe de la classe abstraite Publicité, et créer sur cette classe une opération « envoyer() » spécifique à l'envoi de SMS.

Diagramme de séquence objet du cas d'utilisation UC-07 « Résoudre une anomalie » :

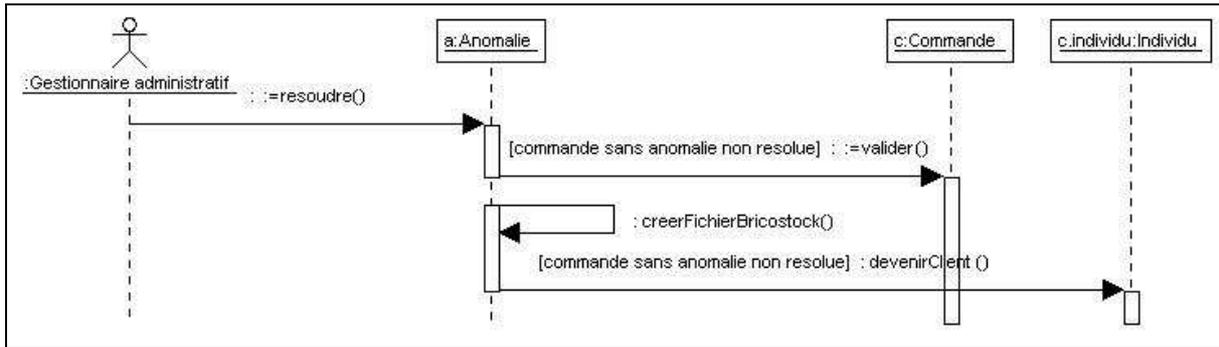
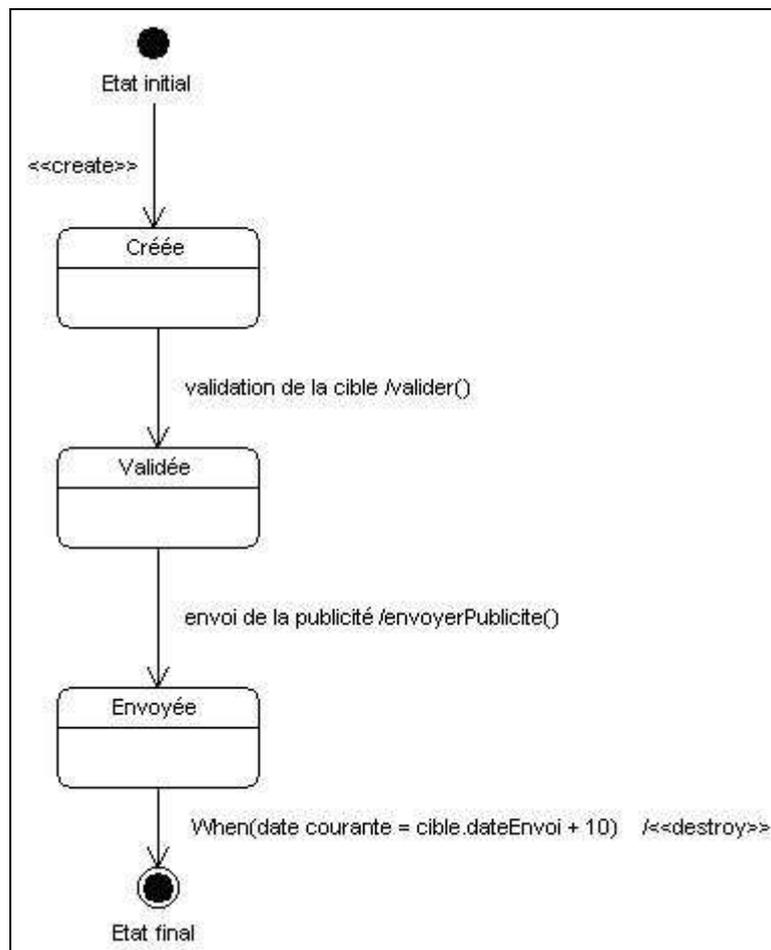


Diagramme d'état de la classe Cible :



Ce diagramme présente, pour les objets de la classe Cible, les différents états et les évènements de transition pour passer d'un état à l'autre. Dans le modèle de classe, l'état correspond à l'attribut « etat ».

Diagramme d'état de la classe *Commande* :

Bien que présentant seulement deux états, il est intéressant de représenter le diagramme d'état de la classe *Commande*. On y montre les transitions, les conditions de transition et les envois de messages aux autres classes.

L'état d'une commande est enregistré dans l'attribut « *estValide* » de la classe.

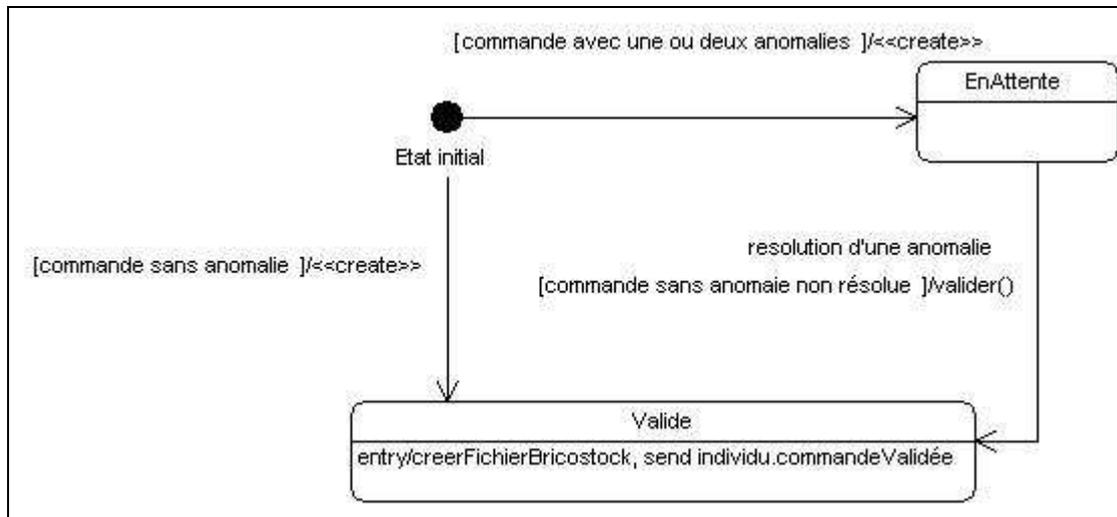
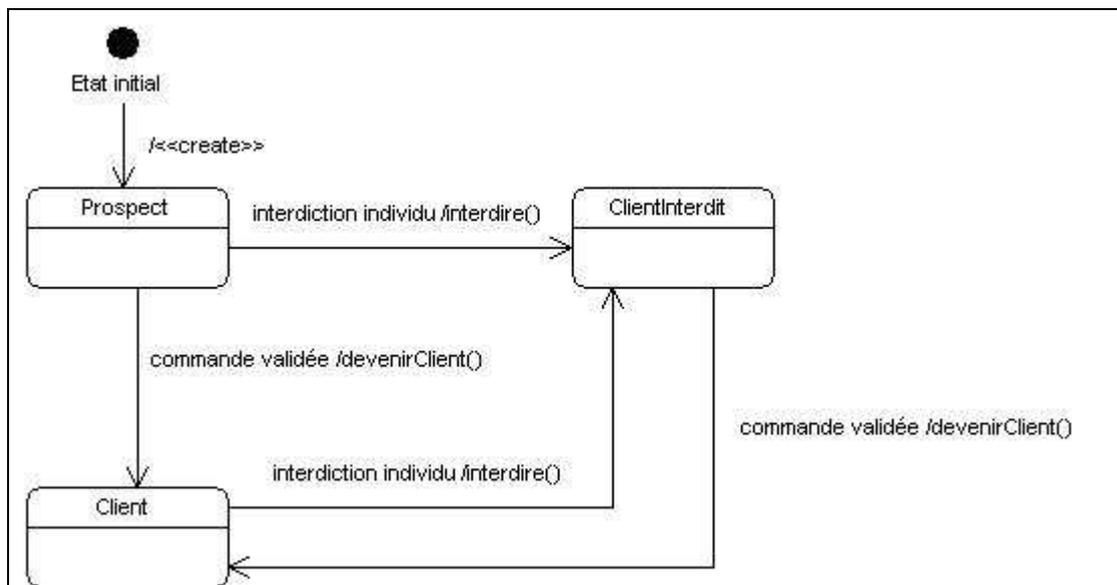


Diagramme d'état de la classe *Individu* :

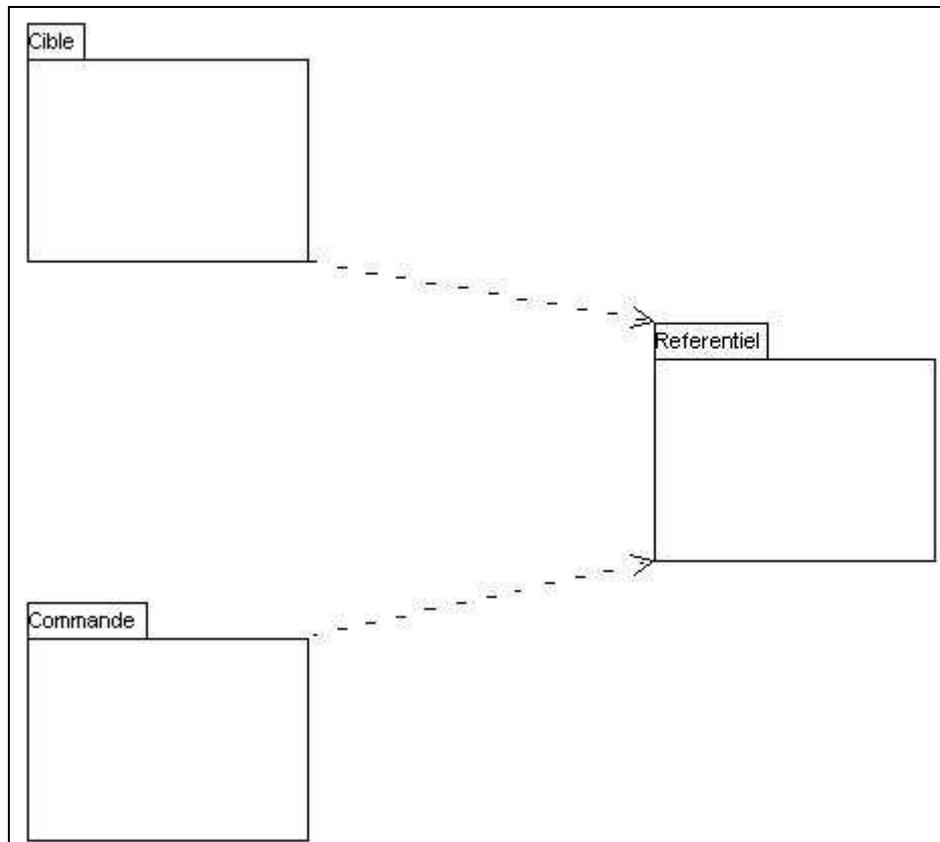
L'état d'un individu est enregistré dans l'attribut « *CaracteristiqueComm* ». Les valeurs et les conditions de transition d'un état à l'autre sont présentées sur le diagramme suivant :



C. Architecture

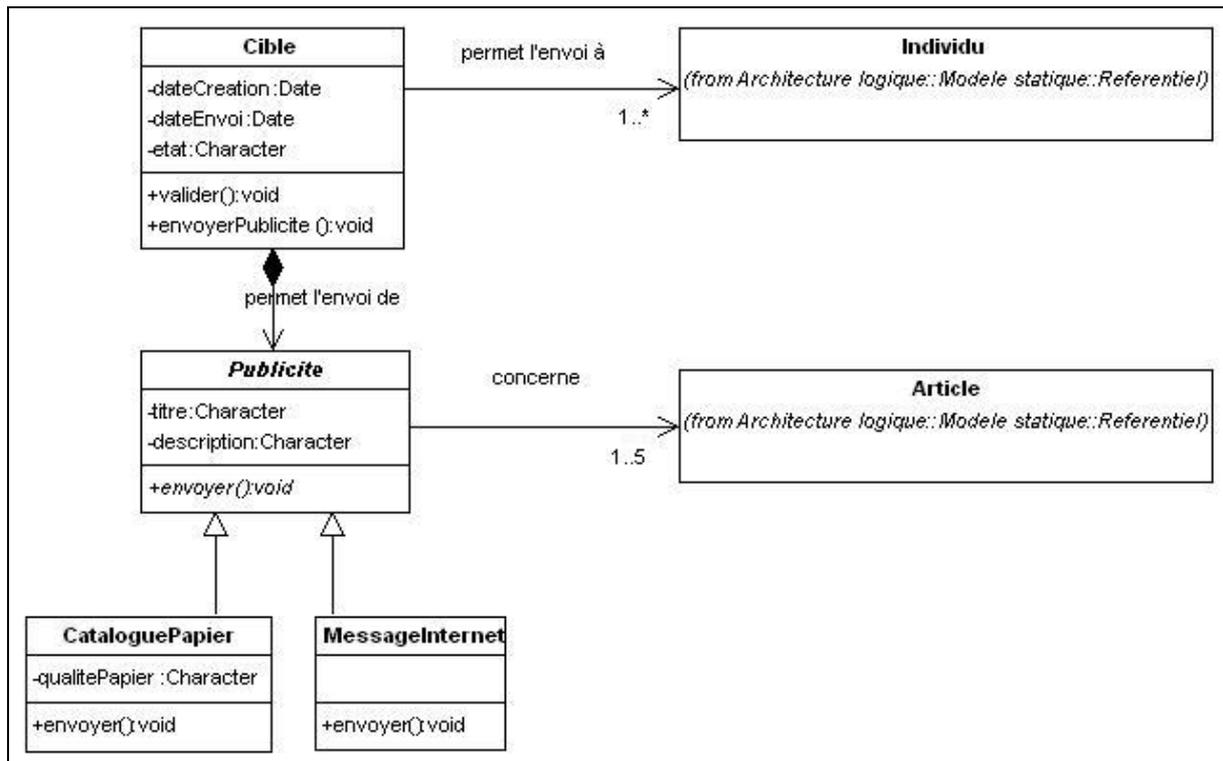
Les classes identifiées dans le paragraphe précédent sont réparties dans des « packages ». La répartition des classes se fait en minimisant les dépendances entre classes de packages différents.

L'architecture proposée est constituée de trois packages : Référentiel, Cible et Commande.

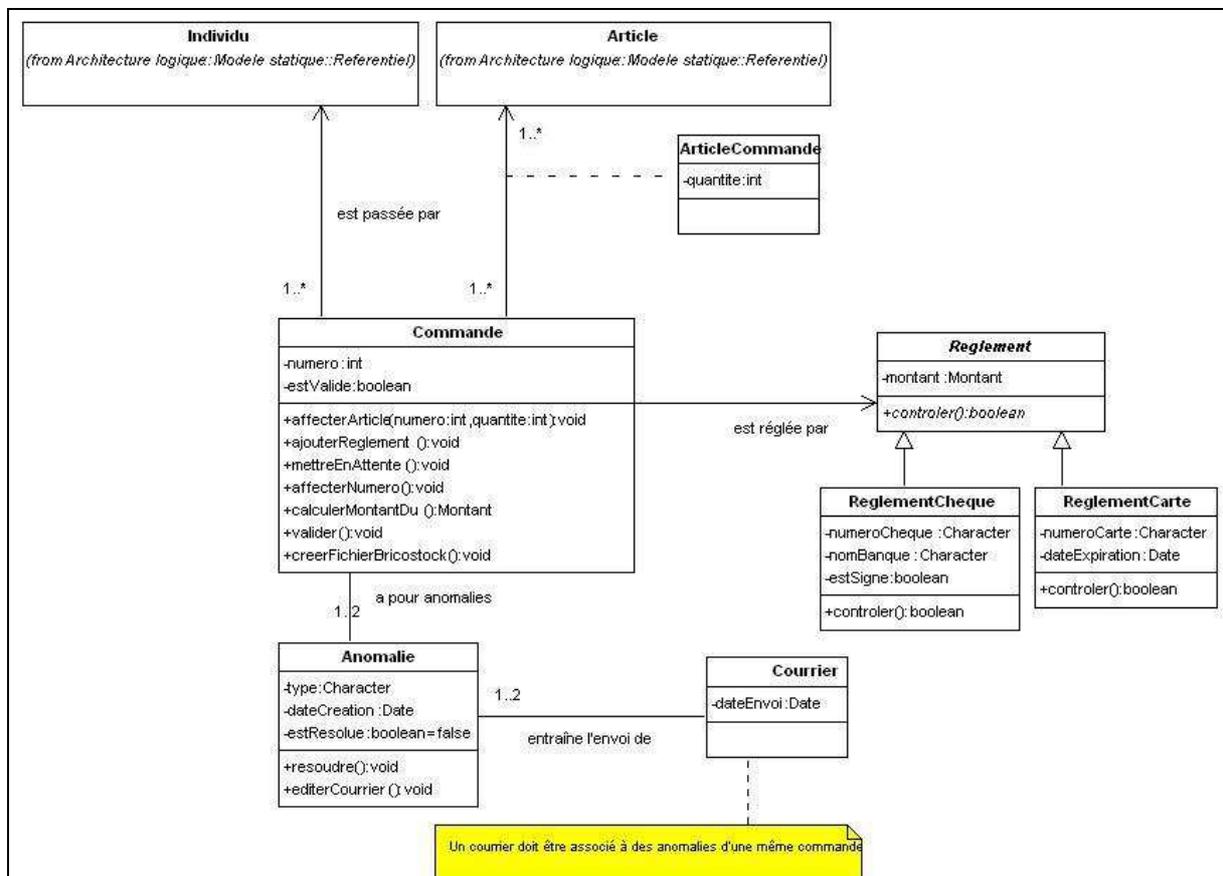


Chacun des trois packages est décrit par les diagrammes de classes qui suivent.

Package Cible



Package Commande



Package Référentiel

