
TUTORIEL DE UWE

Traduction du tutoriel du [site officiel](#)

Images du logiciel MagicDraw

traduit et mis en page par Ludovic DUBOIS

ludovic.dubois89 (at) gmail.com

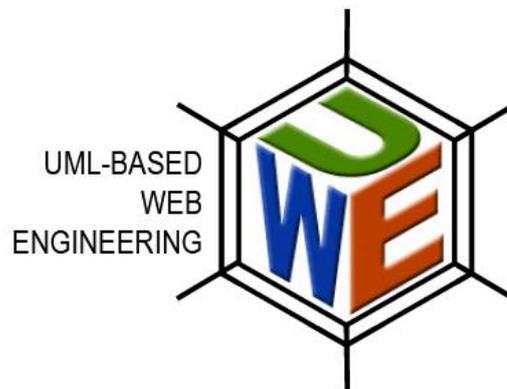


TABLE DES MATIÈRES

1	INTRODUCTION	3
2	EXEMPLE DU CARNET D'ADRESSE	4
2.1	Diagramme de Cas d'Utilisation	4
2.2	Diagramme de Classe	5
2.3	Modèle de Navigation	6
2.4	Modèle de Présentation	10
2.5	Modèle des Processus	13
2.5.1	Structure des Processus	13
2.5.2	Schéma des processus	14
3	TABLE DES FIGURES	18

1 INTRODUCTION

Ce tutoriel est une traduction du [tutoriel en anglais](#), qui se trouve sur le site officiel de UWE.

Toutes les images sont issues du logiciel [MagicDraw](#) avec l'extension [MagicUWE](#). Pour plus d'informations [voir la documentation](#). Les images sont les mêmes que sur le site officiel, donc les images sont en anglais.

UWE est une méthode d'ingénierie Web pour la POO¹ basé sur UML², elle est utilisée pour les spécifications des applications Web ([plus d'informations sur UWE](#)). UWE est une extension de UML, d'ailleurs les diagrammes reposent directement sur une approche UML avec les diagrammes de cas d'utilisation et de classe.

Pour ce tutoriel nous allons prendre un exemple simple, le cas d'un carnet d'adresse.

PS : Merci à Nora KOCH pour ses encouragements et ses conseils.

¹Programmation orientée objet

²Unified Modeling Language, Langage de Modélisation Unifié

2 EXEMPLE DU CARNET D'ADRESSE

2.1 Diagramme de Cas d'Utilisation

Créer un [new Project dans MagicDraw](#) .

Nous allons commencer par créer un diagramme de cas d'utilisation.

L'utilisateur peut chercher et supprimer des contacts dans le carnet d'adresse. De plus, les contacts peuvent être créés et modifiés, mais ces derniers changements peuvent être sauvegardés ou annulés. Le diagramme ci-dessous n'est pas exhaustif et peut être complété.

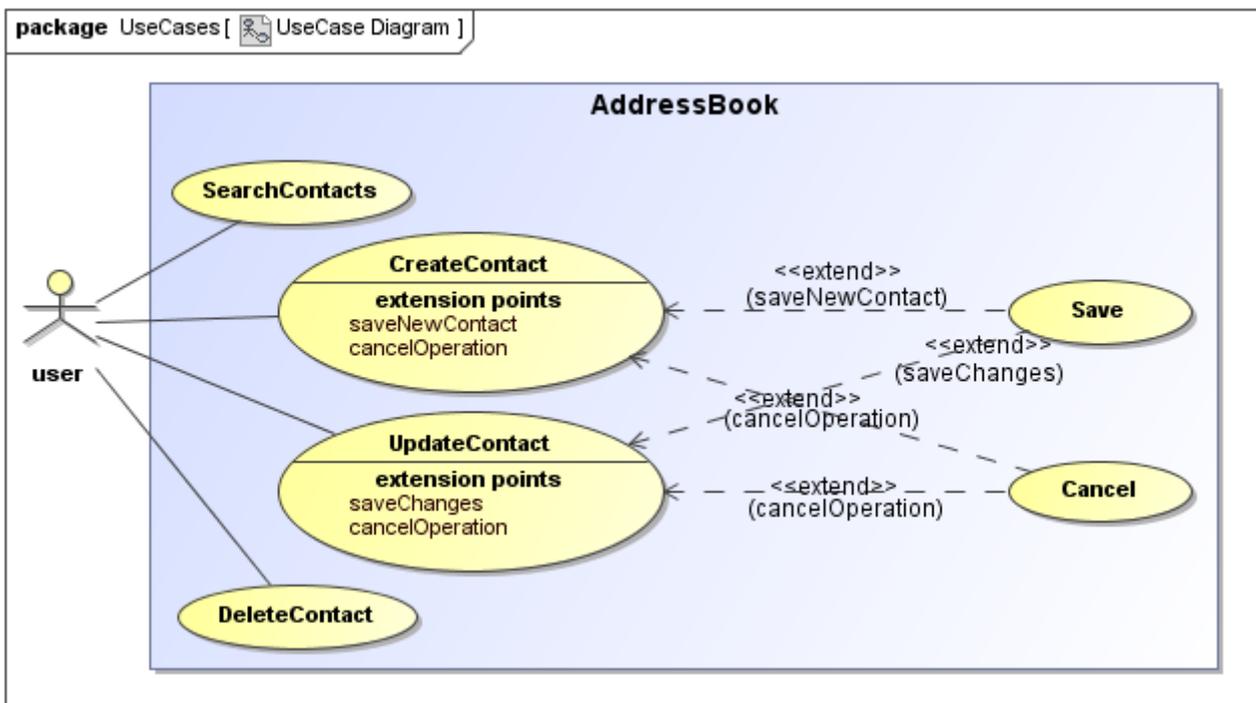


FIG. 1 – Diagramme de Cas d'Utilisaton

2.2 Diagramme de Classe

Créer un [new Content Diagram](#) . C'est un diagramme de classe UML classique, il représente les classes que nous avons besoin pour cet exemple. Nous avons besoin d'une classe `AddressBook` qui est composée de `Contact`s. Chaque `Contact` possède un nom et une adresse mail, deux numéros de téléphone et deux adresses. Le nom et l'adresse sont représentés par une chaîne de caractères (`String`). Les numéros de téléphone et les adresses sont représentés par des classes avec leurs attributs respectifs comme le montre le diagramme ci-dessous :

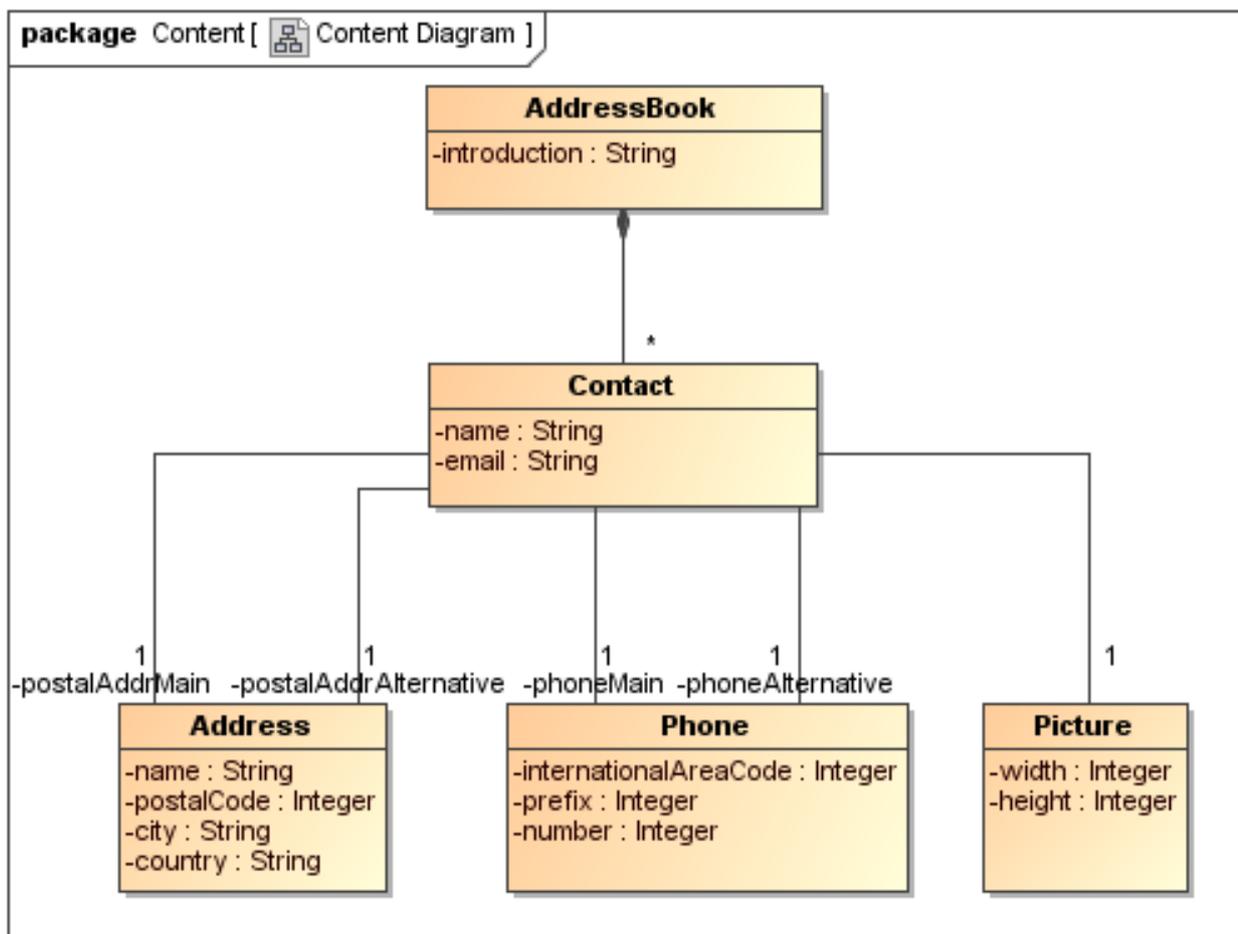


FIG. 2 – Diagramme de Classe

Nous avons ajouté un attribut « introduction » dans le carnet d'adresse car nous voulons garder un champ texte sur la page principale du site web.

2.3 Modèle de Navigation

Pour une application web, il est intéressant de connaître comment les pages sont reliées entre elles. Cela signifie que nous avons besoin d'un diagramme qui comporte des noeuds et des liens.

Les noeuds sont des unités navigables connectées par des liens. Ils sont affichés sur différentes pages ou sur une même page. UWE propose des stéréotypes différents, par exemple pour les noeuds et pour les liens les stéréotypes sont respectivement «navigationClass» et «navigationLink». Nous avons généré le diagramme ci-dessous avec l'action [Content to Navigation - Transformation](#), qui se base sur le diagramme de classe précédent.

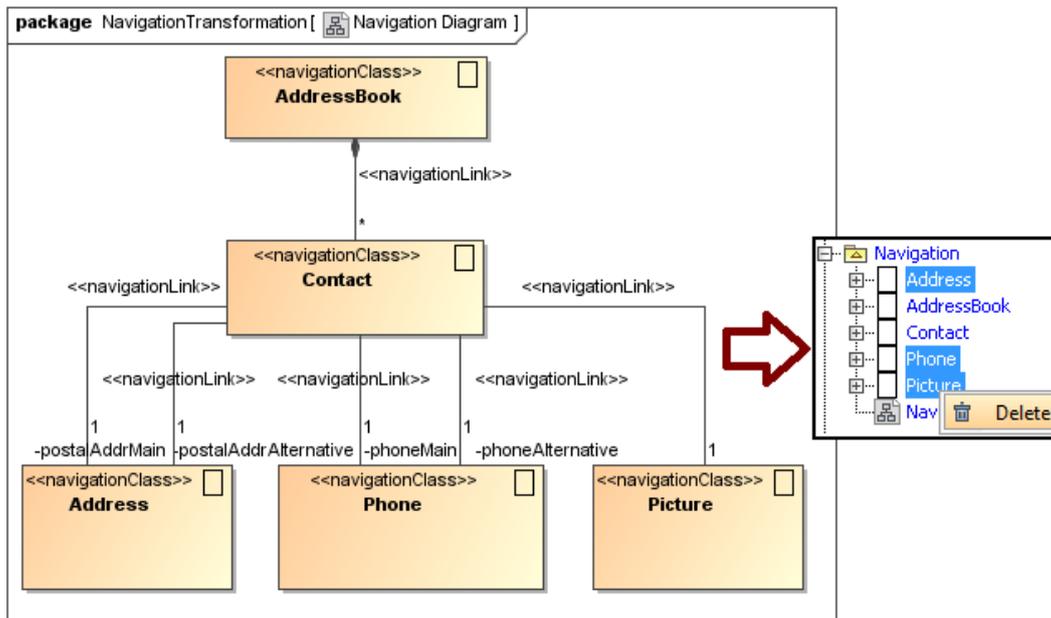


FIG. 3 – Modèle de Navigation - après transformation

Voulez-vous vraiment modéliser la navigation du contact vers Adress, Phone et Picture?. Non, car ses informations ne sont pas pertinentes à la navigation. Supprimer les de l'arborescence.

AddressBook doit être notre page d'accueil, c'est pour cela qu'il est inscrit `{isHome}`.

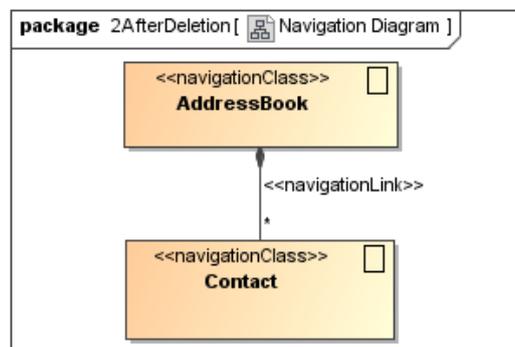


FIG. 4 – Modèle de Navigation - propriété {isHome}

Pouvez vous imaginer un site sur les carnets d'adresses avec tous les contacts dans la même page? Ce n'est pas ce que nous voulons.

Ce que nous voulons c'est une application où nous avons accès aux opérations décrites dans le diagramme de cas d'utilisation. Pour cette raison nous avons besoin d'une page d'accueil avec plusieurs noeuds :

1. ContactList : nous voulons atteindre chaque contact avec un lien à partir de l'accueil.
2. Search(Contact) : recherche pour un contact
3. ContactCreation : créer un nouveau contact et l'afficher après.

Dans UWE, vous pouvez utiliser des «menu», si vous voulez naviguer dans des classes différentes.

[Insérez en un nommé MainMenu](#) :

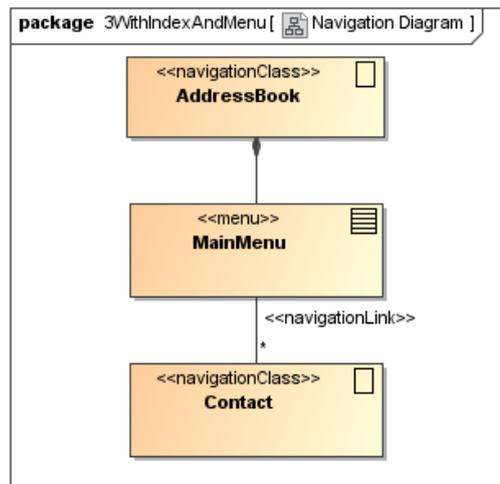


FIG. 5 – Modèle de Navigation - Ajout d'un meun MainMenu

1. De la même manière on peut insérer ContactList. Le stéréotype « index » est utilisé pour lister des objets d'un même type.

Nous devons ajouter deux autres classes avec [MagicUWE toolbar](#) :

2. La classe pour la recherche doit avoir le stéréotype «query». Une recherche exécute du code, donc nous avons besoin de l'association «processLink».
3. ContactCreation est aussi une procédure, mais elle n'est pas prédéfini, par conséquent nous utiliserons le stéréotype «processClass» (nous modéliserons l'action associée plus tard).

Si un nouveau contact est créé, il sera bien d'afficher sa fiche par la suite, et dans le cas de la recherche on voudrais l'affichage du résultat via ContactList. Nous utiliserons le stéréotype «processLink» pour ces associations «dirigées» (pour interdire le «retour» dans la navigation et ainsi éviter de dupliquer les contacts lors d'une création par exemple).

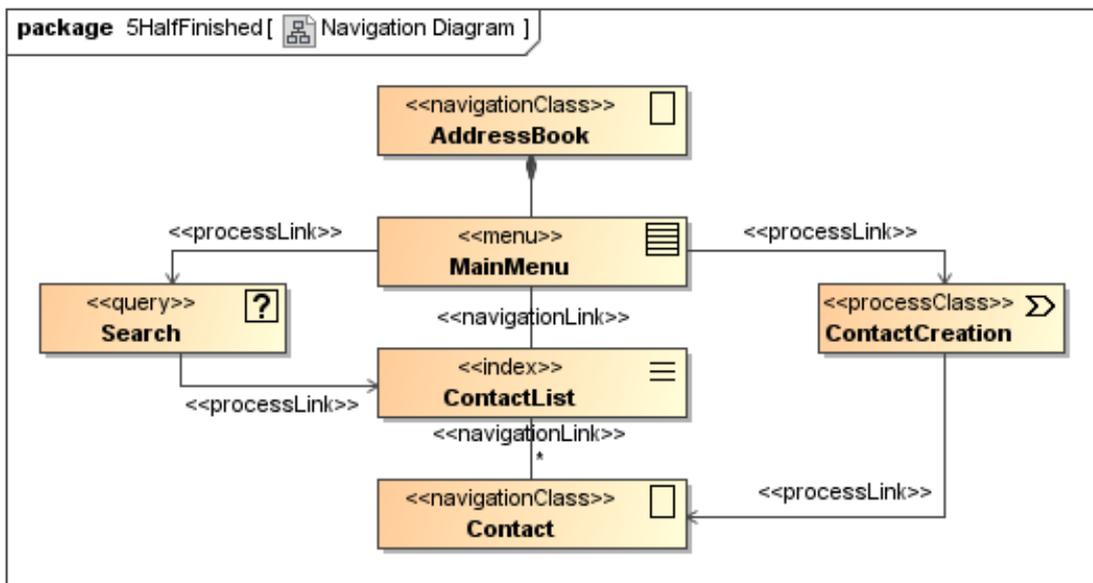


FIG. 6 – Modèle de Navigation - Ajout de la recherche et de la création d'un contact

Icônes et noms des stéréotypes	
navigationClass	menu
index	query
guidedTour	processClass
externalNode	

FIG. 7 – Modèle de Navigation - stéréotypes

Dans ce tutoriel, seulement quelques aspects des stéréotypes ont été utilisés. Pour plus d'informations : [User Guide and Reference](#) pour tous les stéréotypes de UWE.

Pour compléter notre modèle de navigation, nous devons ajouter les fonctionnalités ContactDelete et ContactUpdate (comme modélisé sur le diagramme de cas d'utilisation). Ces deux classes sont accessibles par Contact, donc nous avons besoin d'un menu (et nous l'appellerons ContactMenu, il sera présent sur chaque page d'un contact).

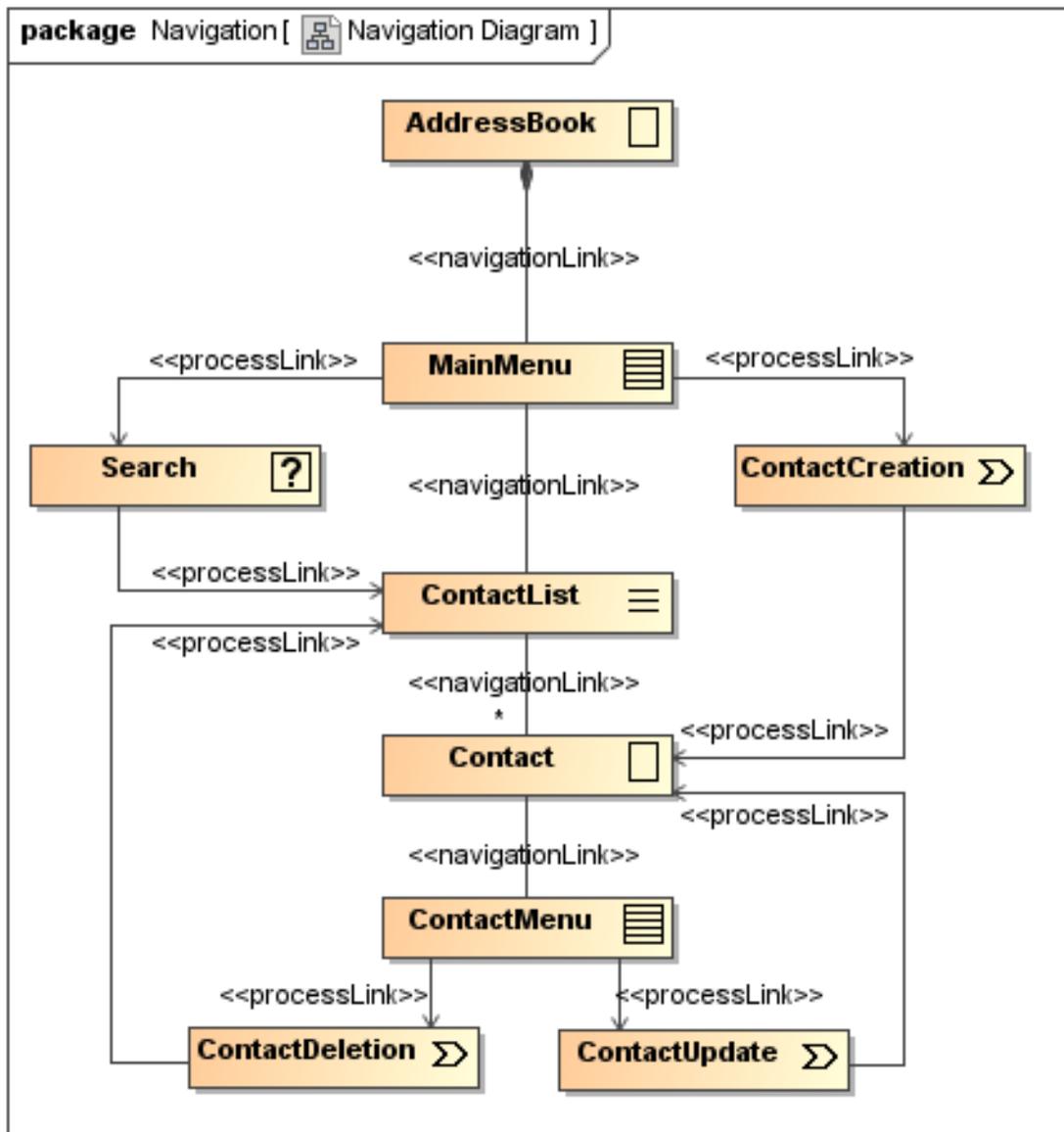


FIG. 8 – Modèle de Navigation - finale

2.4 Modèle de Présentation

Le modèle de navigation ne représente pas la relation entre : les classes de navigation, les classes dites « processus » ou « procédurale » et les pages du site internet. Nous pouvons utiliser un diagramme de présentation pour combler ce manque d'informations.

Ajouter une classe [«presentationPage»](#) et les [propriétés avec les stéréotypes UWE](#) , qui représentent les éléments comme il vont apparaître sur la page. Les propriétés peuvent être « imbriquées », par exemple chaque contact («presentationClass»-propriété), on retrouve des textes et des boutons différents. C'est à dire, que pour chaque contact, les champs pour son email, pour ses téléphones et pour ses adresses seront affichées.

Souvenons nous de l'attribut « introduction » dans notre diagramme de classe et ajoutons la page d'accueil du carnet d'adresses qui contient le texte d'introduction (stéréotype «text»). Ensuite, ajoutons un formulaire avec un champ de saisie pour la recherche par critère et un bouton pour lancer la recherche si nécessaire. Le nombre variable de contact peut être affiché et sera modélisé avec « * »-multiplicity.

Le stéréotype «presentationAlternatives» autorise à modéliser une RIA (Rich Internet Application). Nous pouvons toujours rester à une page (ici, c'est la page AddressBook) et le formulaire de modification/création des contacts, qui est l'intérieur de la propriété « MainAlternatives », sont affichés en fondu enchaînés.

Modèle de Présentation - stéréotypes	
 presentationGroup	 presentationPage
 text	 textInput
 anchor	 fileUpload
 button	 image
 inputForm	 customComponent
 presentationAlternatives	 selection

FIG. 9 – Modèle de Présentation - stéréotypes

Dans les diagrammes suivants, les stéréotypes sont seulement représentés par des icônes. Vous pouvez configurer MagicDraw pour afficher les deux, nom et l'icône du stéréotype.

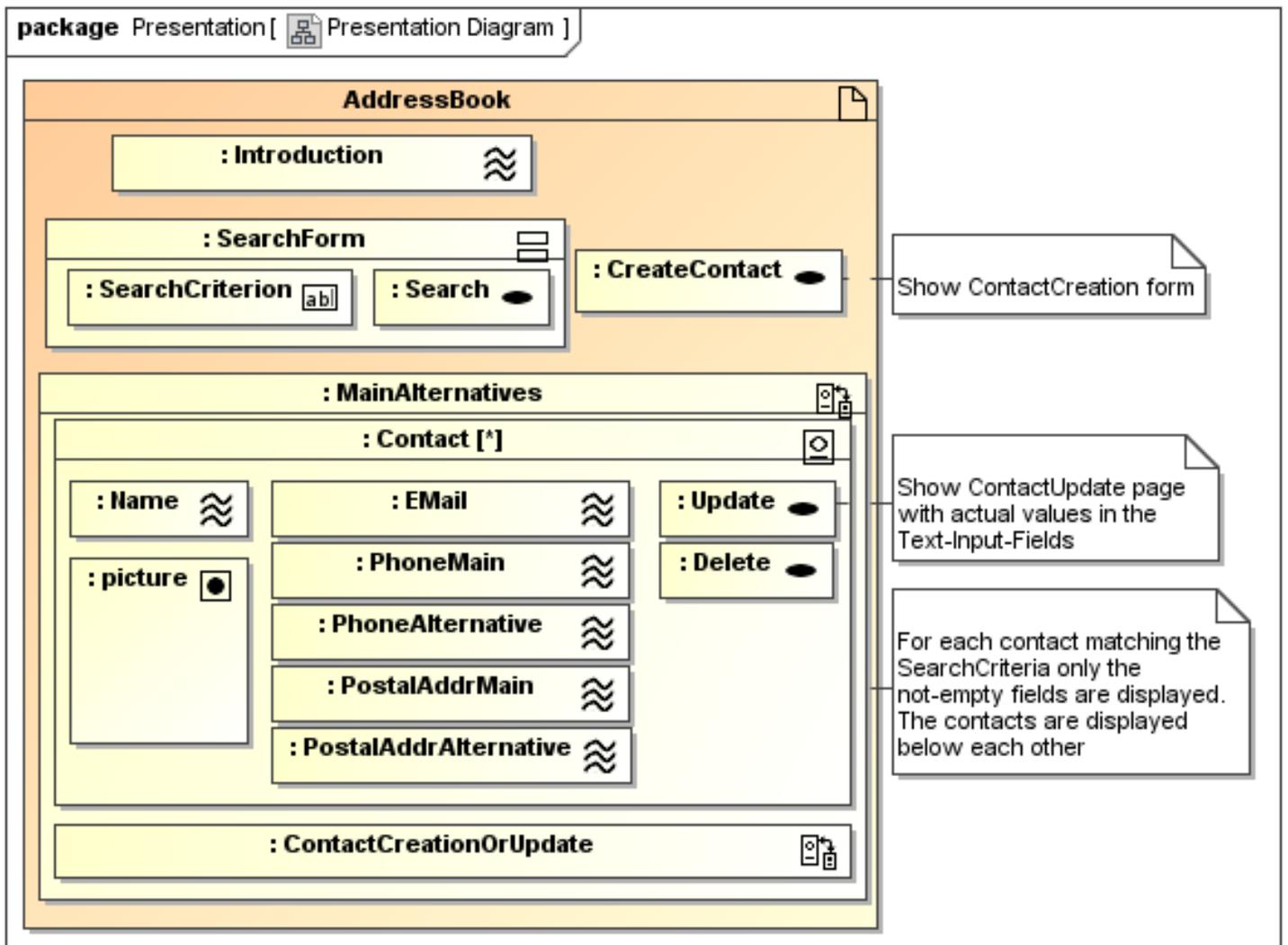


FIG. 10 – Modèle de Présentation - Page principale

Message, Confirmation and ValidationError sont modélisées ici, mais pas du tout dans le diagramme de Navigation. Ceux sont de simple boîte de dialogue qui affichent des messages ou des questions.

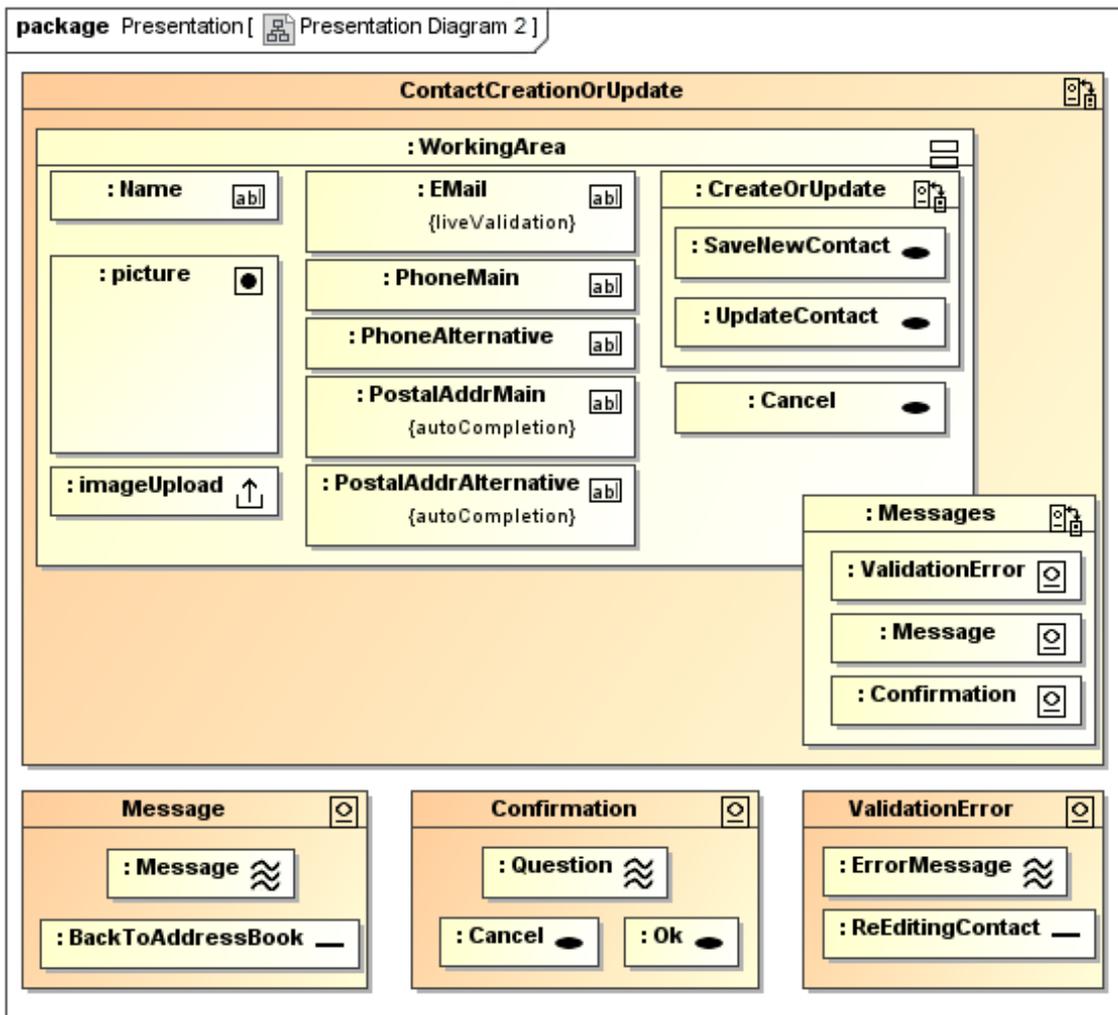


FIG. 11 – Modèle de Présentation - Autres pages

Les formulaires ContactCreation et ContactUpdate sont similaires, seulement le nom de la page et les boutons de validations ("ok"-buttons) sont différents de leurs propre fonctionnement. Par conséquence, le stéréotype « presentationAlternatives » sera encore utilisé pour éviter la modélisation multiple de l'ensemble du contenu des ces deux formulaires. Mais est-ce une solution dans le cas d'une création de contact(ContactCreation), avant que l'image soit téléchargée? - Peut-être que plus tard vous voudrais implémenter insertion d'une image par défaut...

Les valeurs indiquant autoCompletion et liveValidation sont utilisées pour spécifier que les champs des adresses ont une fonctionnalité d'"auto completion " et le format de l'adresse mail est vérifié automatiquement.

2.5 Modèle des Processus

Jusqu'à maintenant nous avons réalisé divers diagrammes, mais nous n'avons pas encore vu comment les actions des classes « processus » sont liées. Le modèle des processus comprend :

- Structure des Processus : la description des relations entre les différentes classes « processus »
- Schéma des processus : la spécification des activités connectées entre chaque «processClass»

2.5.1 Structure des Processus

Dans le but de décrire les relations entre les différentes classes « processus », nous créons un diagramme de classe (en utilisant [Navigation to Process Structure Transformation](#)). Ensuite, nous avons un diagramme de classe avec trois classes (entourées en rouge) comme ci-dessous :

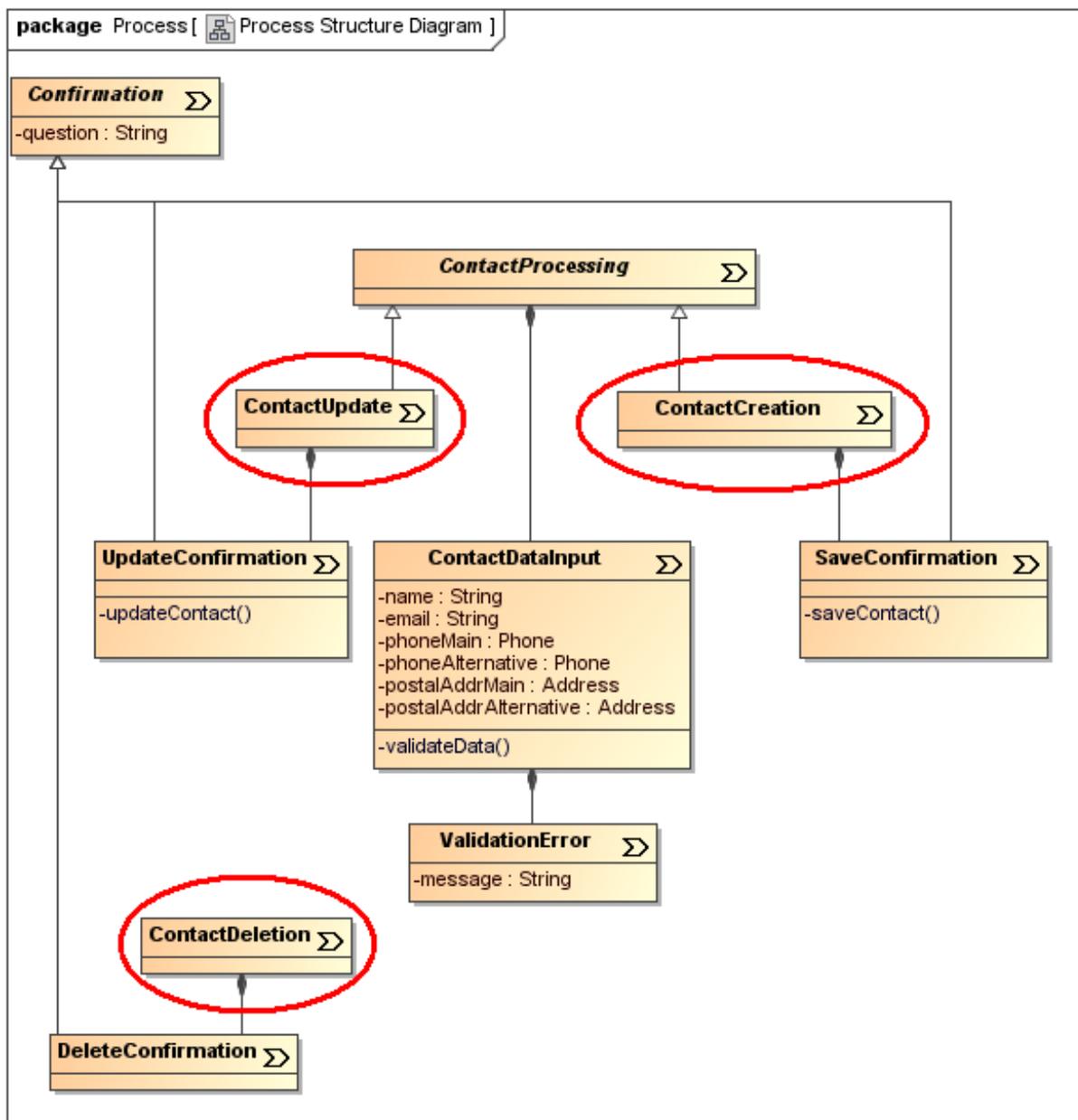


FIG. 12 – Modèle des Processus - Structure des Processus

Comme vous pouvez le voir, nous avons ajouté d'autres classes, les trois opérations (Update, Save and Delete) incluent une Confirmation (présentée dans le diagramme de présentation) avec une question. Cela signifie que si un utilisateur veut supprimer un contact, un message apparaîtra; si l'utilisateur appuie sur « ok », le contact sera supprimé. ContactCreation et ContactUpdate sont très similaires, ils héritent tous les deux d'une classe abstraite ContactProcessing, qui assure que les champs de saisie seront les attributs de ContactDataInput. Si il est possible de valider les données de l'utilisateur, nous n'avons pas besoin de prendre le cas d'une erreur de validation (ValidationError) dans l'affichage de la page de confirmation. Pour les détails des activités, passez à l'étape suivante.

2.5.2 Schéma des processus

Schéma des processus (Process Flow or Workflow) est un diagramme d'activité, il décrit le comportement d'une classe « processus », on va retrouver les événements qui se produiront quand l'utilisateur va naviguer dans une classe « processus ». (comme ContactCreation dans notre exemple).

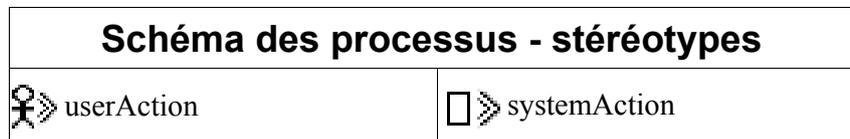


FIG. 13 – Schéma des processus - stéréotypes

Maintenant, nous sélectionnons notre diagramme de navigation et exécutons [Navigation to Process Flows Transformation](#) . Maintenant nous avons trois diagrammes d'activités vides :

- ContactCreation
- ContactDeletion
- ContactUpdate

Le stéréotype «user Action» est utilisé pour indiquer les interactions avec l'utilisateur sur une page d'initialisation d'une procédure ou répondant à un besoin d'information.

Voici la fin du tutoriel UWE, car les trois diagrammes qui suivent sont des diagrammes standard d'UML.

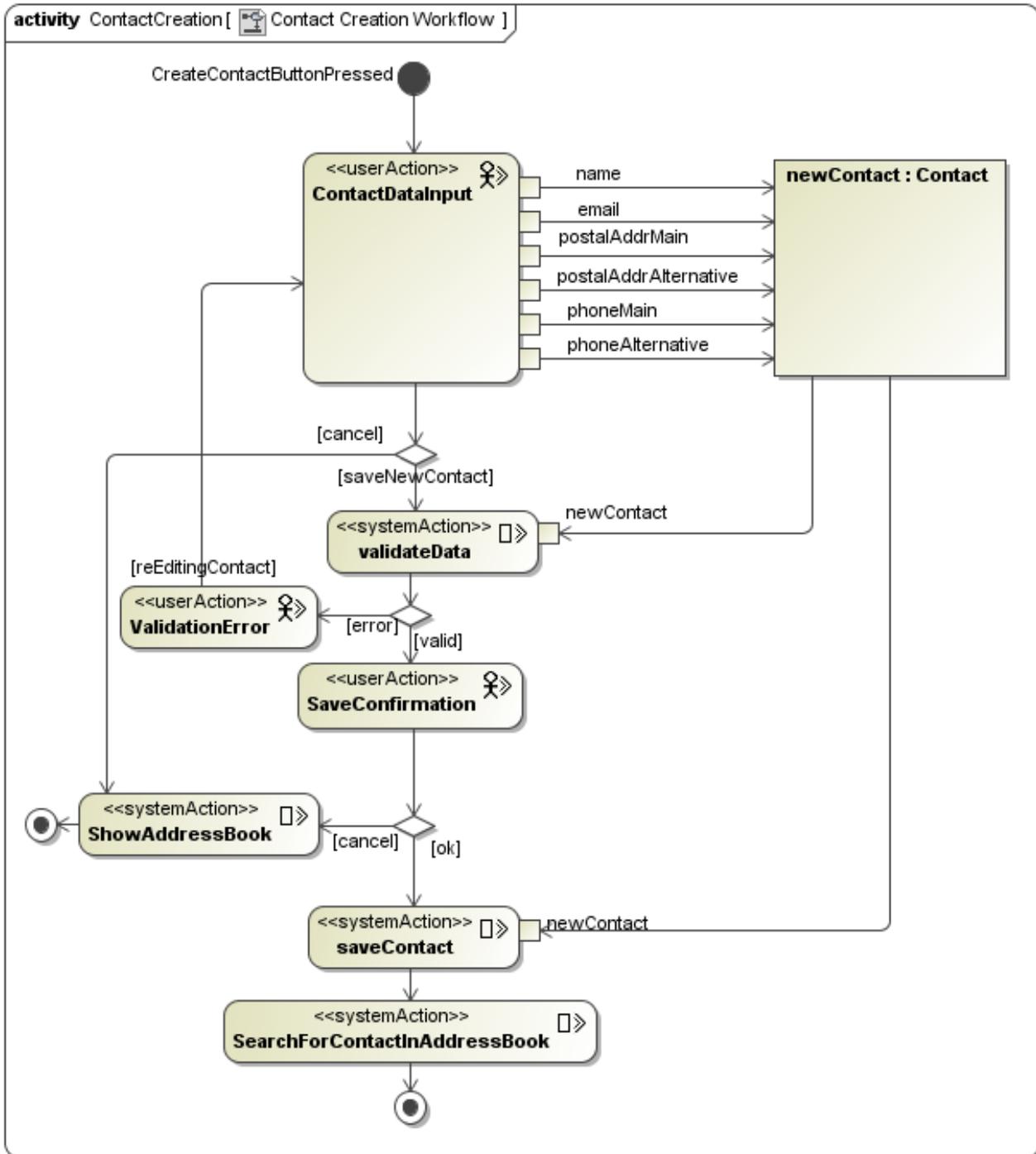


FIG. 14 – Modèle des Processus - Schéma du processus de la création d'un contact

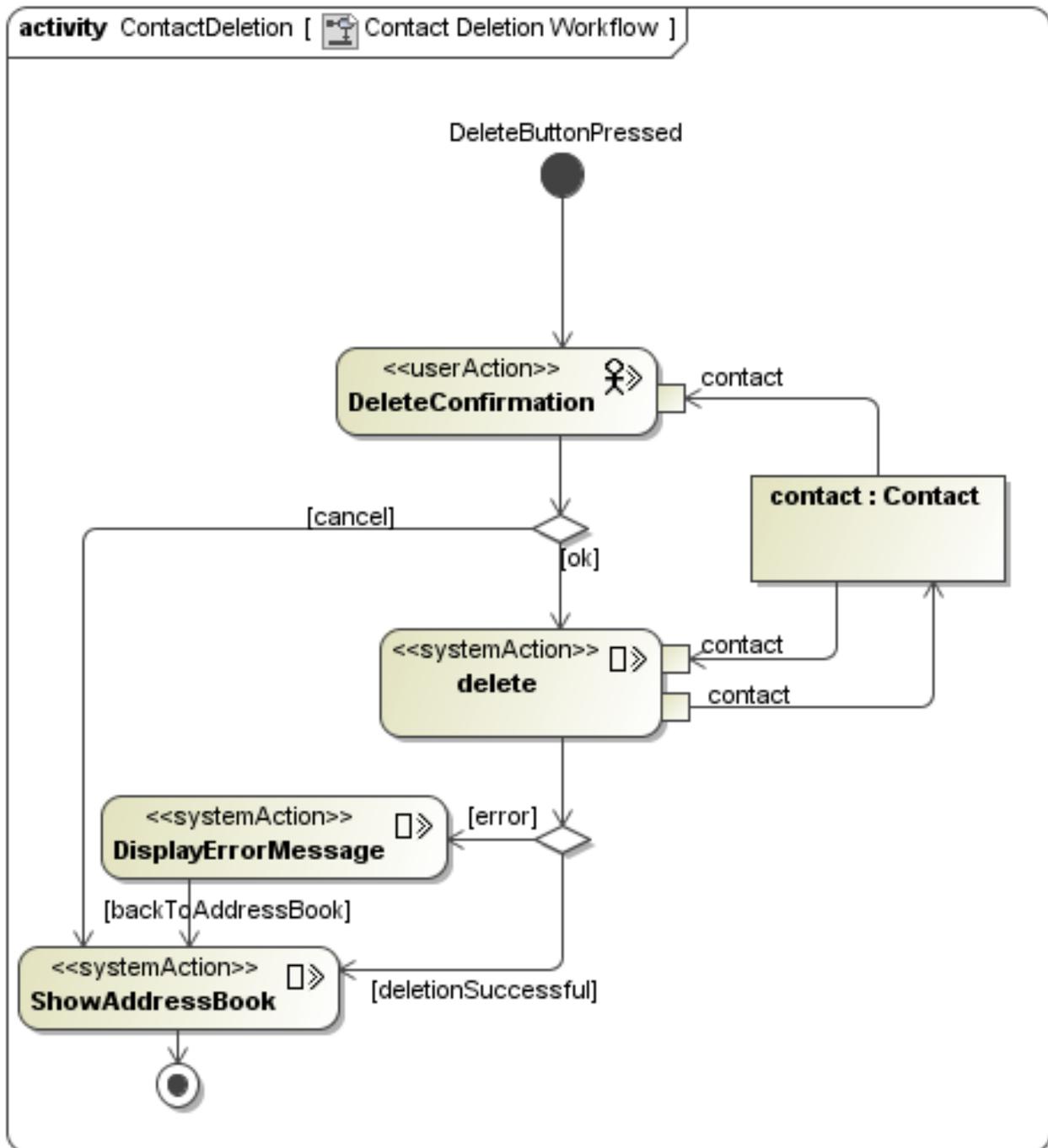


FIG. 15 – Modèle des Processus - Schéma du processus de la suppression d'un contact

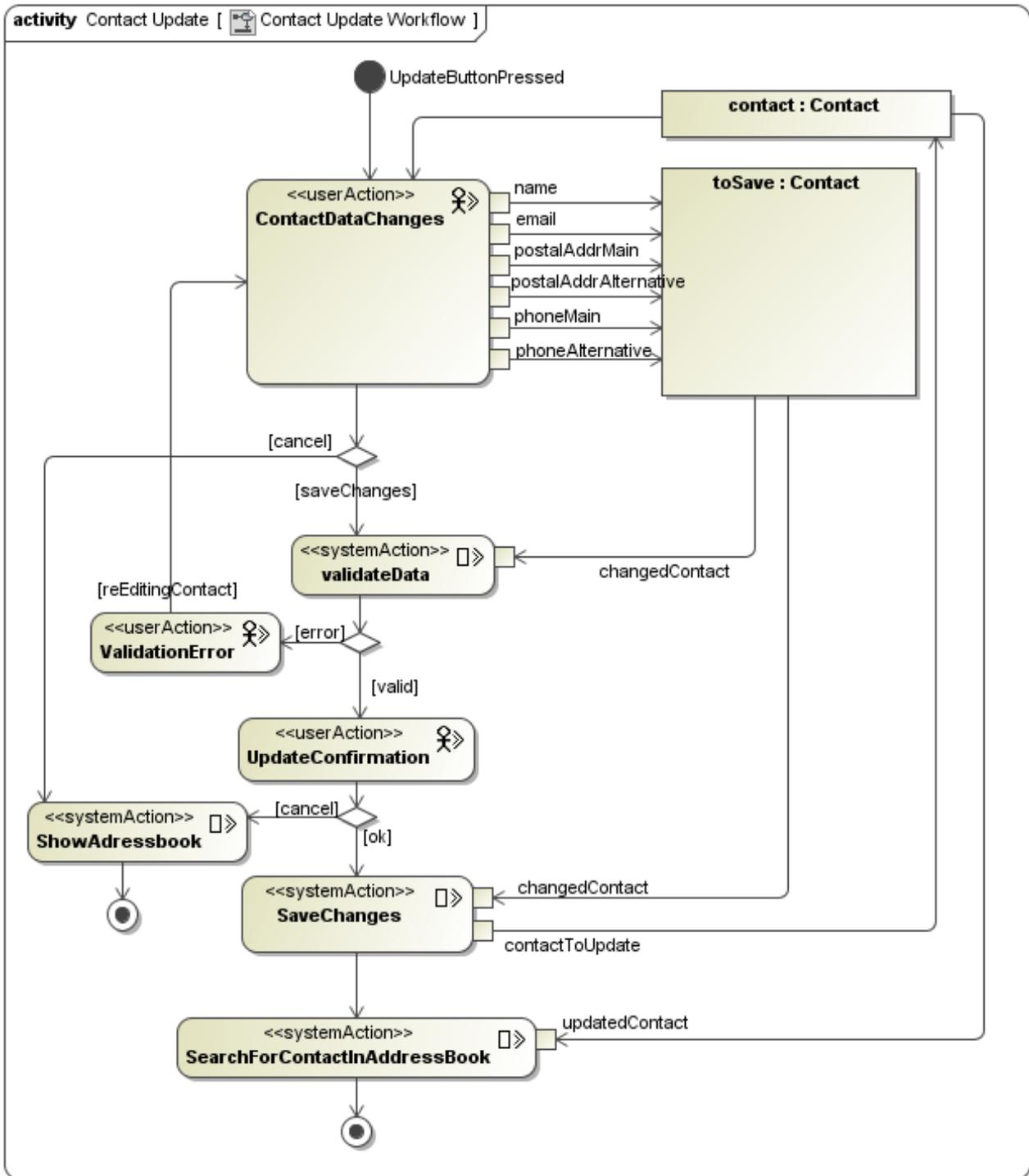


FIG. 16 – Modèle des Processus - Schéma du processus de la modification d'un contact

Vous pouvez voir d'autre exemple : [exemple section](#) .

3 TABLE DES FIGURES

1	Diagramme de Cas d'Utilisaton	4
2	Diagramme de Classe	5
3	Modèle de Navigation - après transformation	6
4	Modèle de Navigation - propriété {isHome}	6
5	Modèle de Navigation - Ajout d'un meun MainMenu	7
6	Modèle de Navigation - Ajout de la recherche et de la création d'un contact	8
7	Modèle de Navigation - stéréotypes	8
8	Modèle de Navigation - finale	9
9	Modèle de Présentation - stéréotypes	10
10	Modèle de Présentation - Page principale	11
11	Modèle de Présentation - Autres pages	12
12	Modèle des Processus - Structure des Processus	13
13	Schéma des processus - stéréotypes	14
14	Modèle des Processus - Schéma du processus de la création d'un contact	15
15	Modèle des Processus - Schéma du processus de la suppression d'un contact	16
16	Modèle des Processus - Schéma du processus de la modification d'un contact	17