
MODÉLISATION DES EXIGENCES FONCTIONNELLES

DIAGRAMME DE CAS D'UTILISATION



OBJECTIFS

- Objectifs
 - Modéliser les fonctionnalités de haut niveau du système du point de vue des parties prenantes, appelés acteurs qui effectuent des tâches définies comme cas d'utilisation.
- Formalisme
 - Acteur
 - Cas d'utilisation
 - Association
 - Dépendance

ACTEUR

- Rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié.
- Un acteur participe à au moins un cas d'utilisation



Opérateur Maintenance

CAS D'UTILISATION

- Un cas d'utilisation définit un service attendu du système par un acteur, il représente un ensemble de séquences possibles d'actions réalisées par le système et produisant un résultat observable et mesurable afin de satisfaire les objectifs d'un utilisateur particulier du système.



Retrait d'argent

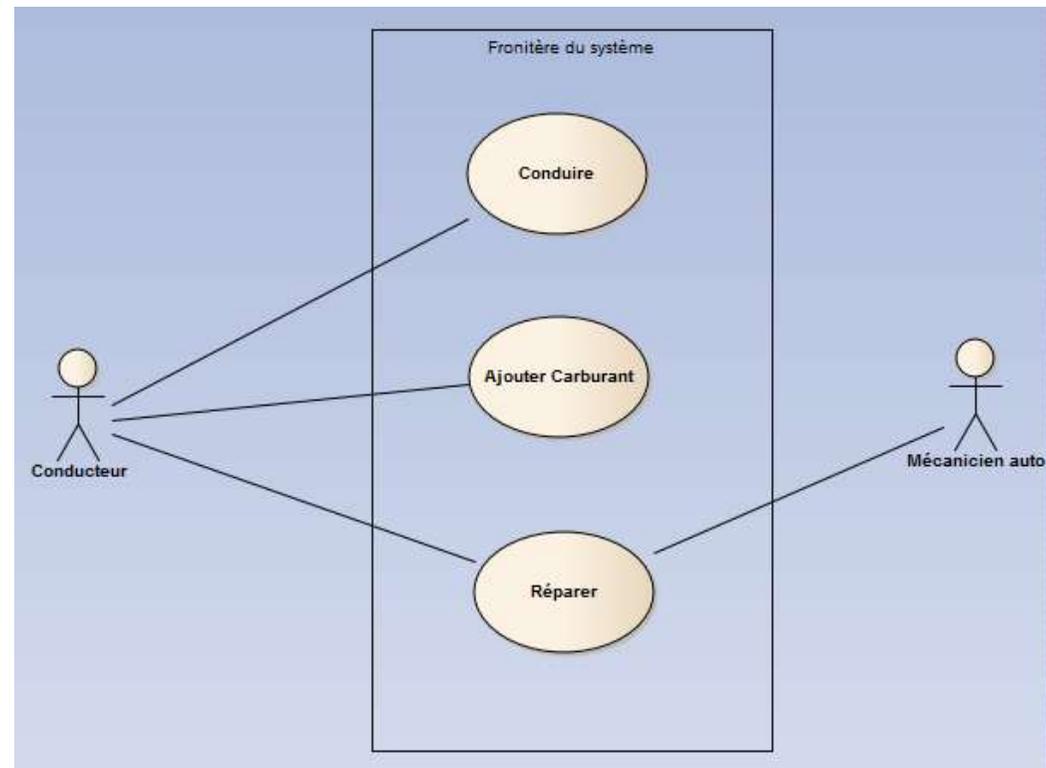
ASSOCIATIONS

- C'est une relation définie entre un acteur et un cas d'utilisation.
- Un cas d'utilisation doit être relié au moins un acteur.



ACTEUR PRINCIPAL ET ACTEURS SECONDAIRES

- L'acteur principal est celui pour qui le cas d'utilisation est réalisé.
- Les acteurs secondaires contribuent au fonctionnement du cas d'utilisation.
- Les acteurs secondaires sont souvent sollicités pour des informations complémentaires.
- Les acteurs principaux figurent en général à gauche des cas d'utilisation et les acteurs secondaires à droite.



SPÉCIFICATION D'UN CAS D'UTILISATION

- Un cas d'utilisation doit spécifier un flot complet d'événements, deux formats peuvent être utilisés :
 - Un ou plusieurs diagrammes de séquence
 - Un format textuel appelé spécification du cas d'utilisation.
- La description textuelle n'est pas normalisée par UML.

DESCRIPTION TEXTUELLE

- La description est composée de quatre rubriques :
 - Sommaire d'identification (rubrique obligatoire)
 - titre, objectif, résumé, dates, version, responsable, acteurs
 - Description des scénarii (rubrique obligatoire)
 - Décrit les enchaînements nominal et alternatifs, les enchaînements exceptionnels, les exceptions mais aussi les pré conditions et les post conditions.
 - Contraintes non-fonctionnelles (rubrique optionnelle)
 - fréquence, volumétrie, disponibilité, fiabilité, intégrité, confidentialité, performances...

DESCRIPTION TEXTUELLE

- préconditions (l'état du système pour que le cas d'utilisation puisse démarrer) et les
- Préconditions: Spécifie des contraintes qui doivent être vérifiées pour que le cas d'utilisation puisse démarrer.
- Post-conditions : ce qui a changé dans l'état du système à la fin du cas d'utilisation.

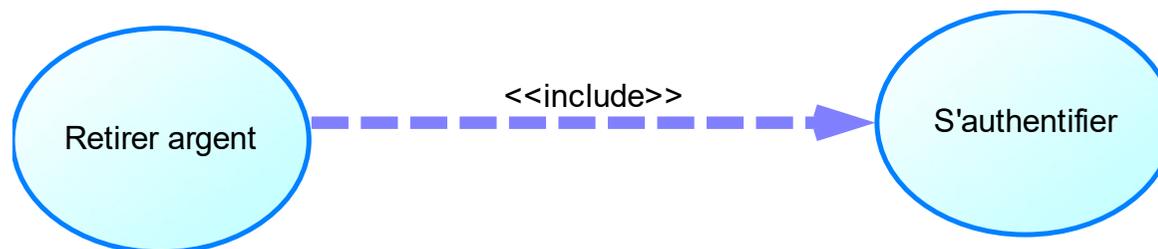
AVANTAGES DES SPÉCIFICATIONS TEXTUELLES DES CAS D'UTILISATION

- Simple à utiliser, aucun outil CASE n'est nécessaire
- aucune connaissance de la méthode utilisée n'est requise.
- aucune formation n'est nécessaire.
- Portabilité
- Expression des UC dans un langage compréhensible par le client.

RELATIONS ENTRE CAS D'UTILISATION

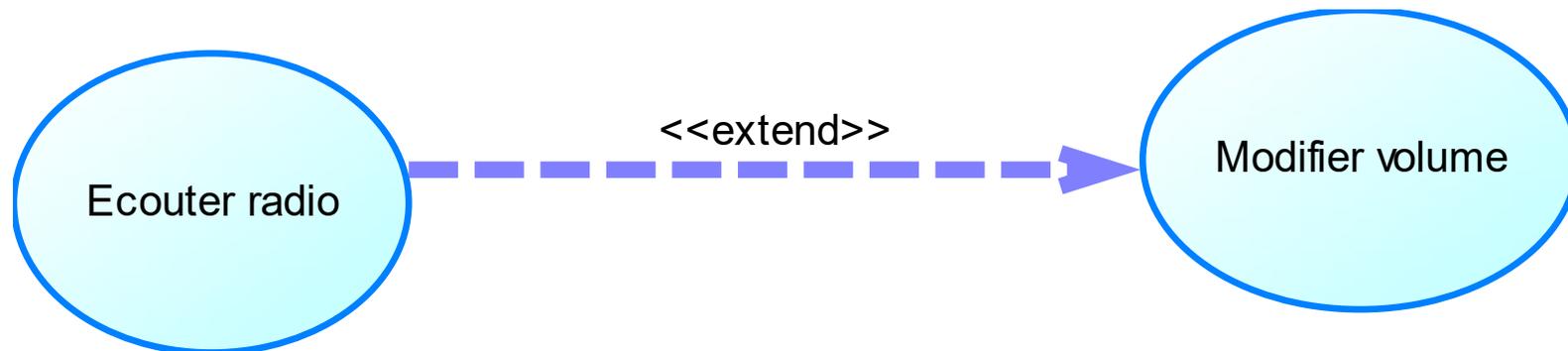
DÉPENDANCE <<INCLUDE >>

- Include: Le terme <<include>> est une dépendance stéréotypée dans laquelle le cas d'utilisation source incorpore de façon explicite le comportement du cas d'utilisation cible à un emplacement spécifié dans le cas d'utilisation source.
- La dépendance <<include>> est utilisée généralement pour placer un comportement commun à plusieurs cas d'utilisation dans son propre cas d'utilisation.



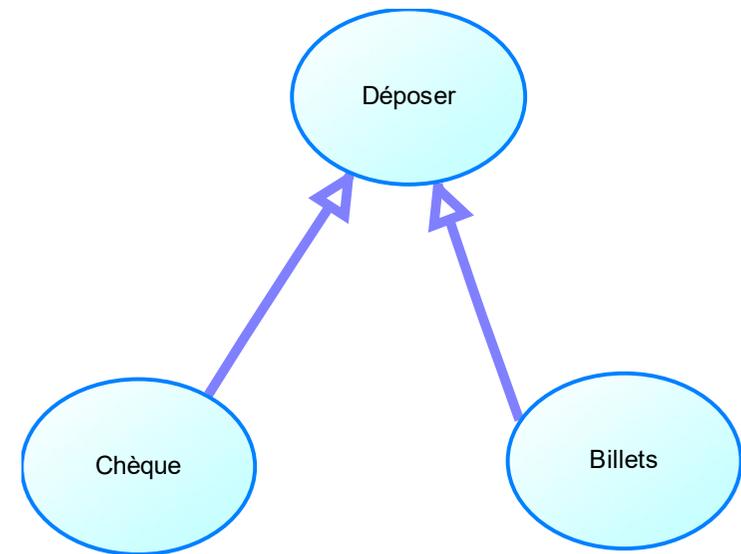
RELATIONS ENTRE CAS D'UTILISATION DÉPENDANCE <<EXTEND>>

- le cas d'utilisation de base incorpore de façon optionnelle un autre cas d'utilisation, à un endroit spécifié



RELATIONS ENTRE CAS D'UTILISATION GÉNÉRALISATION/SPÉCIALISATION

- des cas d'utilisation héritent la description de leur parent commun. Chacun d'entre eux comprend des interactions spécifiques supplémentaires.



DÉPENDANCES ENTRE CAS D'UTILISATION

Différence entre <<include>> et <<extend>>

	<<include>>	<<extend>>
L'UC est optionnel	non	Oui
L'UC de base est incomplet sans cet UC	oui	non
L'exécution de l'UC est conditionnel	non	oui
L'UC modifie le comportement de l'UC de base	non	oui