

Introduction :

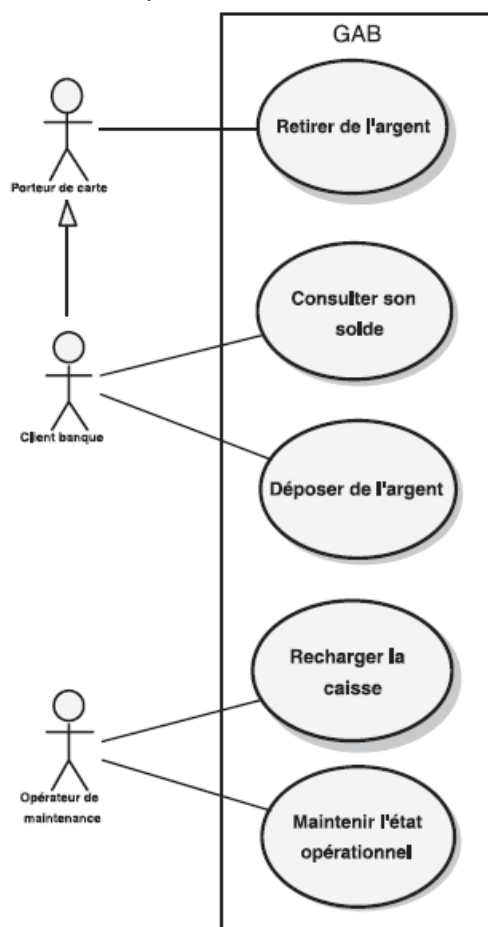
Rôles :

- Ils décrivent les **fonctionnalités du système** du point de vue d'un utilisateur. Les **Cas d'Utilisation** (ou Use Cases) recentrent l'expression des besoins sur les utilisateurs. Les cas d'utilisation sont donc très utiles pour représenter ce que doit **faire** un système par rapport à son environnement (utilisateurs + autres systèmes).

Intérêts :

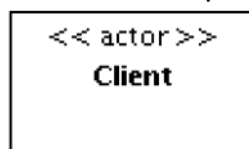
- Permettent de délimiter les frontières du système étudié.
- Utilisés par les utilisateurs du système pour exprimer leurs attentes et leurs besoins.

Ex : un Guichet Automatique de Banque



- Les acteurs (petit bonhomme) : personnes physiques ou autres systèmes qui interagissent avec le système étudié
- Les cas d'utilisation (ellipse) : une fonctionnalité du système que peut réaliser un acteur

Remarque : les acteurs peuvent également être représentés par un classeur



Les relations :

Relation entre acteur et cas d'utilisation : association ; un acteur utilise un cas d'utilisation ; symbole graphique :

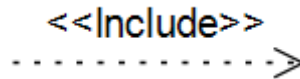


Relation entre acteurs : généralisation /spécialisation ; un acteur « hérite » des possibilités d'un autre acteur ; symbole graphique :

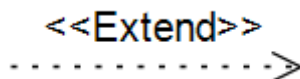


Relations entre cas d'utilisation :

- une relation d'inclusion, formalisée par le mot-clé `<<include>>` : le cas d'utilisation contient un autre cas d'utilisation ; symbole graphique :



- une relation d'extension, formalisée par le mot-clé `<<extend>>` : le cas d'utilisation étend (précise) les objectifs (comportements) d'un autre cas d'utilisation ; symbole graphique :



- une relation de généralisation/spécialisation ; un cas d'utilisation « hérite » d'un autre cas d'utilisation ; symbole graphique



Différence entre une relation de type Include et de type Extend :

- Include : utilise ou inclus un autre cas d'utilisation (obligatoire)
- Extend : étend un autre cas d'utilisation (optionnel)

