

Guide de notation UML



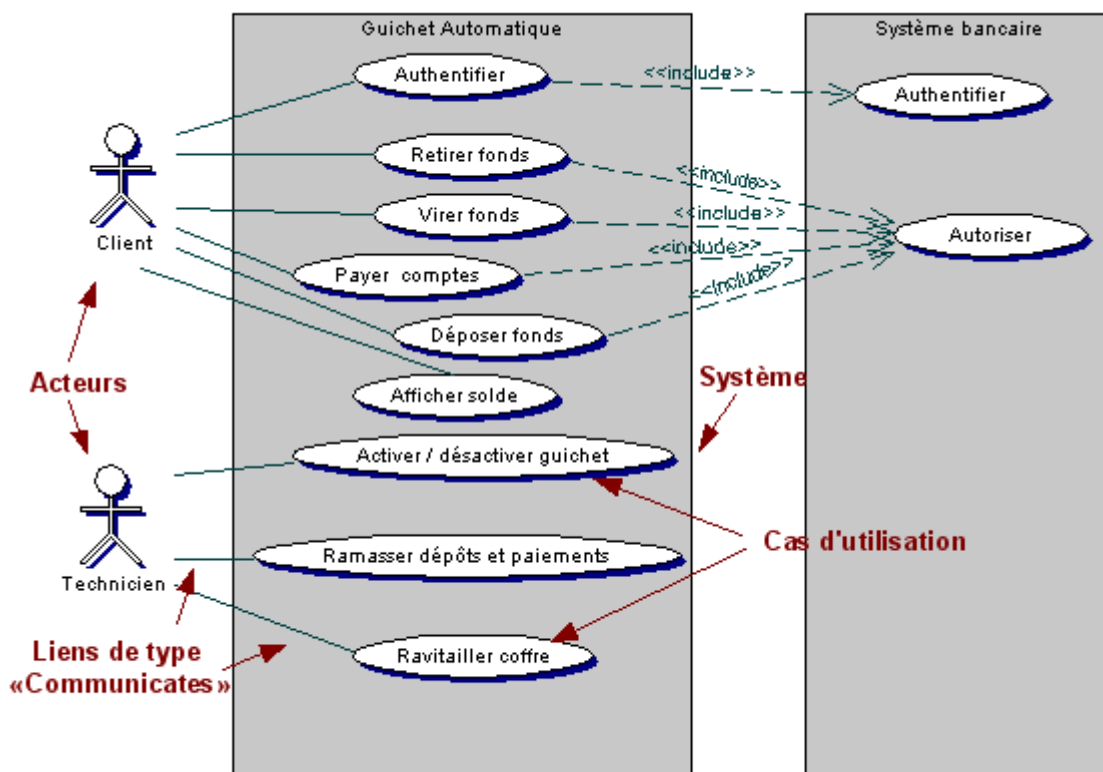
PREAMBULE	3
DIAGRAMME DE CAS D'UTILISATION.....	3
DIAGRAMME DE CLASSES	5
DIAGRAMME DE CLASSE SIMPLE	5
DIAGRAMME DE CLASSE AVANCE.....	6
DIAGRAMME DE PAQUETAGE	7
DIAGRAMME D'INTERACTIONS.....	8
DIAGRAMME D'ACTIVITES.....	8
DIAGRAMME DE SEQUENCE.....	9
DIAGRAMME DE COLLABORATION	10
DIAGRAMME D'ETATS TRANSITIONS	11
AUTRES DIAGRAMMES UML	12
DIAGRAMME DE ROBUSTESSE	12
DIAGRAMME DE COMPOSANTS	13
DIAGRAMME DE DEPLOIEMENT	14

Guide de notation UML

Préambule

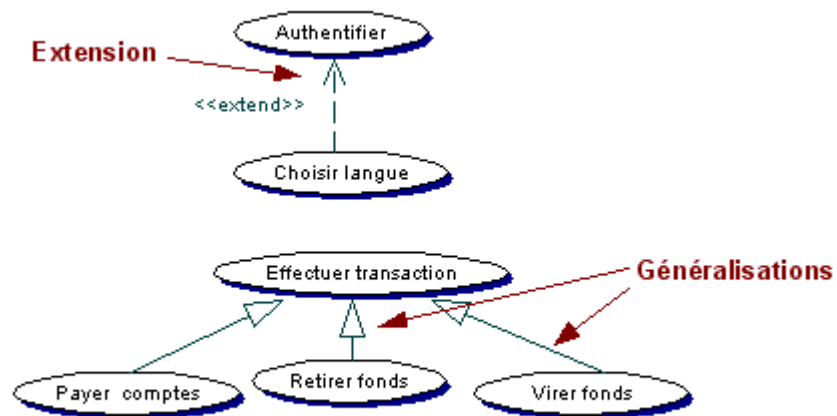
Ce document se veut une synthèse des principaux éléments de la notation UML. Chaque diagramme UML y est présenté avec un exemple et des notes succinctes. Le document ne doit pas être utilisé comme une référence pour la norme UML, pour ce besoin.

Diagramme de cas d'utilisation



Notes :

- Les liens de type `<< include >>` signifient qu'un cas d'utilisation réfère à (pointe vers) un autre
- Les noms des cas d'utilisation sont exprimés sous forme d'action
- Un acteur peut "communiquer" avec plusieurs cas

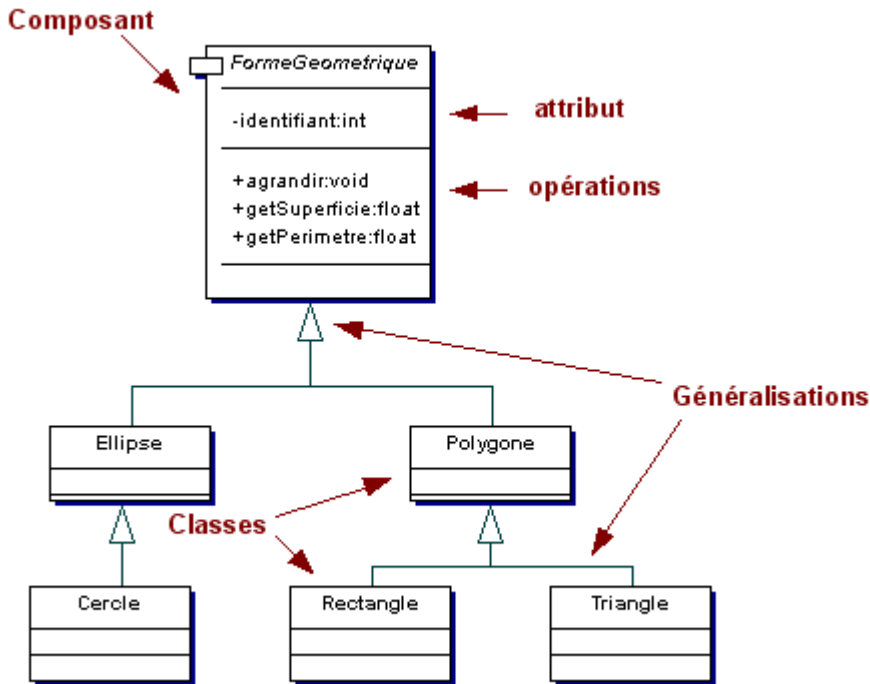


Notes :

- Les liens de type << extend>> signifient qu'un cas d'utilisation est l'extension d'un autre. Le cas qui est extensionne connaît le cas extensionner et non l'inverse. Aussi on pointe la flèche vers le cas extensionné
- Les liens de généralisation servent à indiquer la nature générique d'un cas d'utilisation. Dans l'exemple, "Effectuer transaction" est un cas générique, une généralisation des cas "Payer comptes", "Retirer fonds" et "Virer fonds"

Diagramme de classes

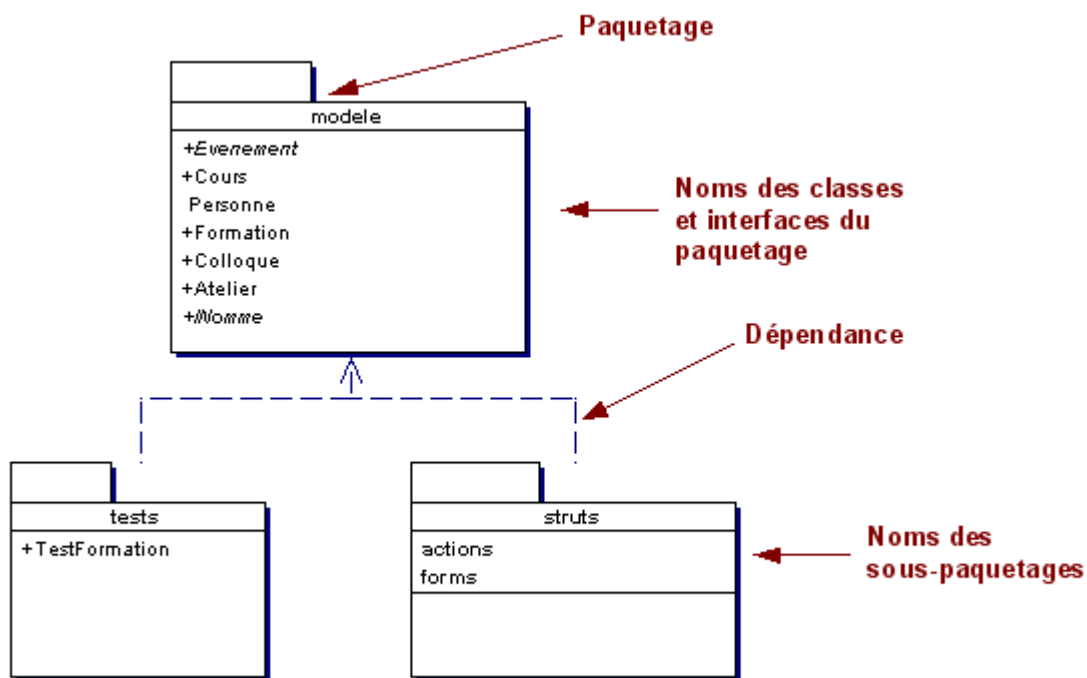
Diagramme de classe simple



Notes :

- Une classe est représentée par un rectangle avec le nom de la classe dans le haut de celui-ci
- *FormeGeometrique* est une classe abstraite et apparaît donc en italiques
- *FormeGeometrique* est un composant (Bean) ce qui est représenté avec un onglet à gauche du nom du composant
- *FormeGeometrique* a un attribut privé (-) nommé *identifiant* de type *int*
- *FormeGeometrique* a trois opérations (méthodes) publiques. L'opération *agrandir* retourne *void* alors que les deux autres retournent un objet de type *float*.
- *Ellipse* et *Polygone* sont des sous-classes de *FormeGeometrique* en vertu d'une relation de généralisation.

Diagramme de paquetage

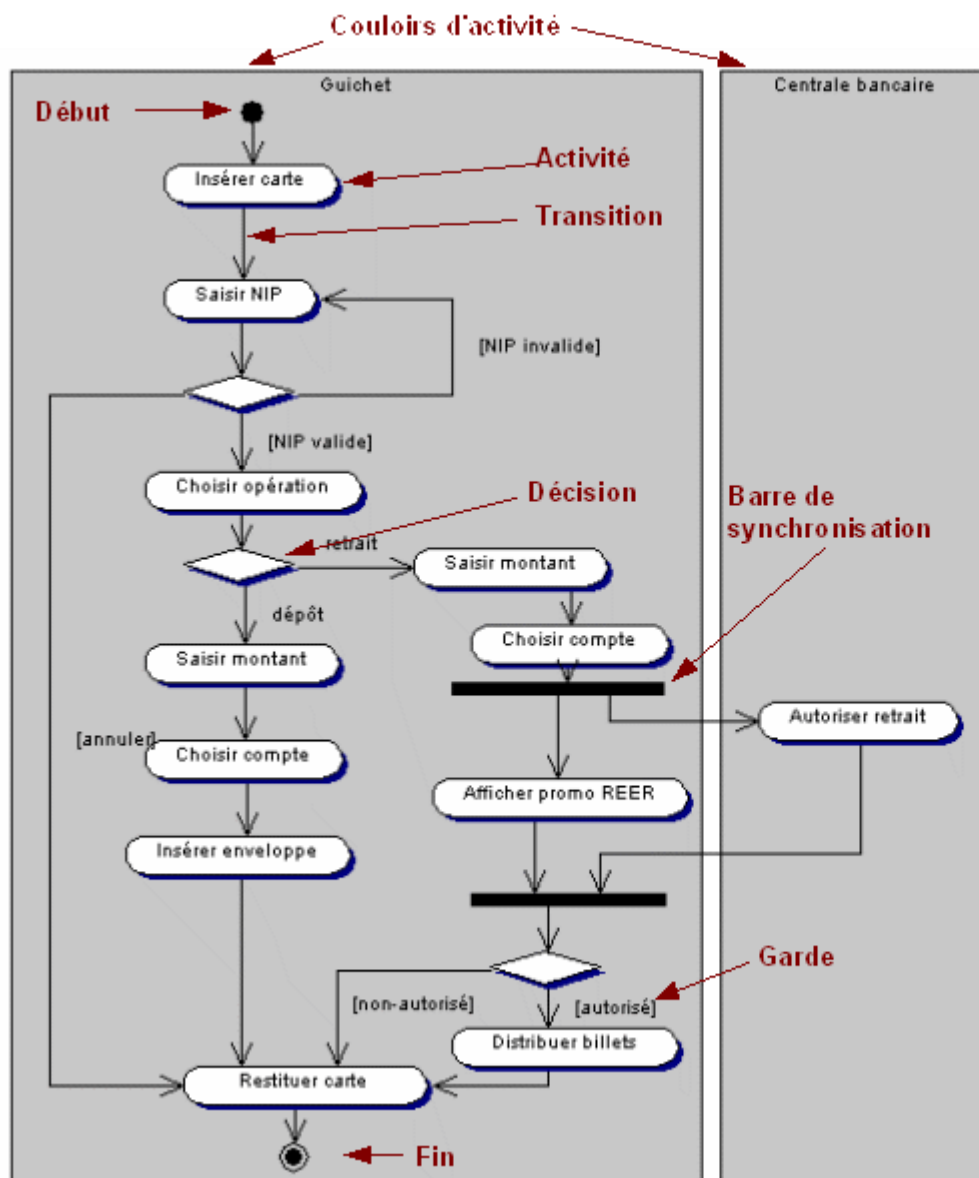


Notes :

- Dans cette figure, les noms des paquetage n'apparaissent pas au long. Ainsi, le nom complet pour le paquetage `modele` pourrait être `gouv.mrn.formation.modele`
- Le paquetage contient une listes des noms de classes, d'interfaces et de sous-paquetage
- Le lien de dépendance indique que des classes ou interfaces du paquetage tests par exemple, contiennent des références à des composants du paquetage `modele`.

Diagramme d'interactions

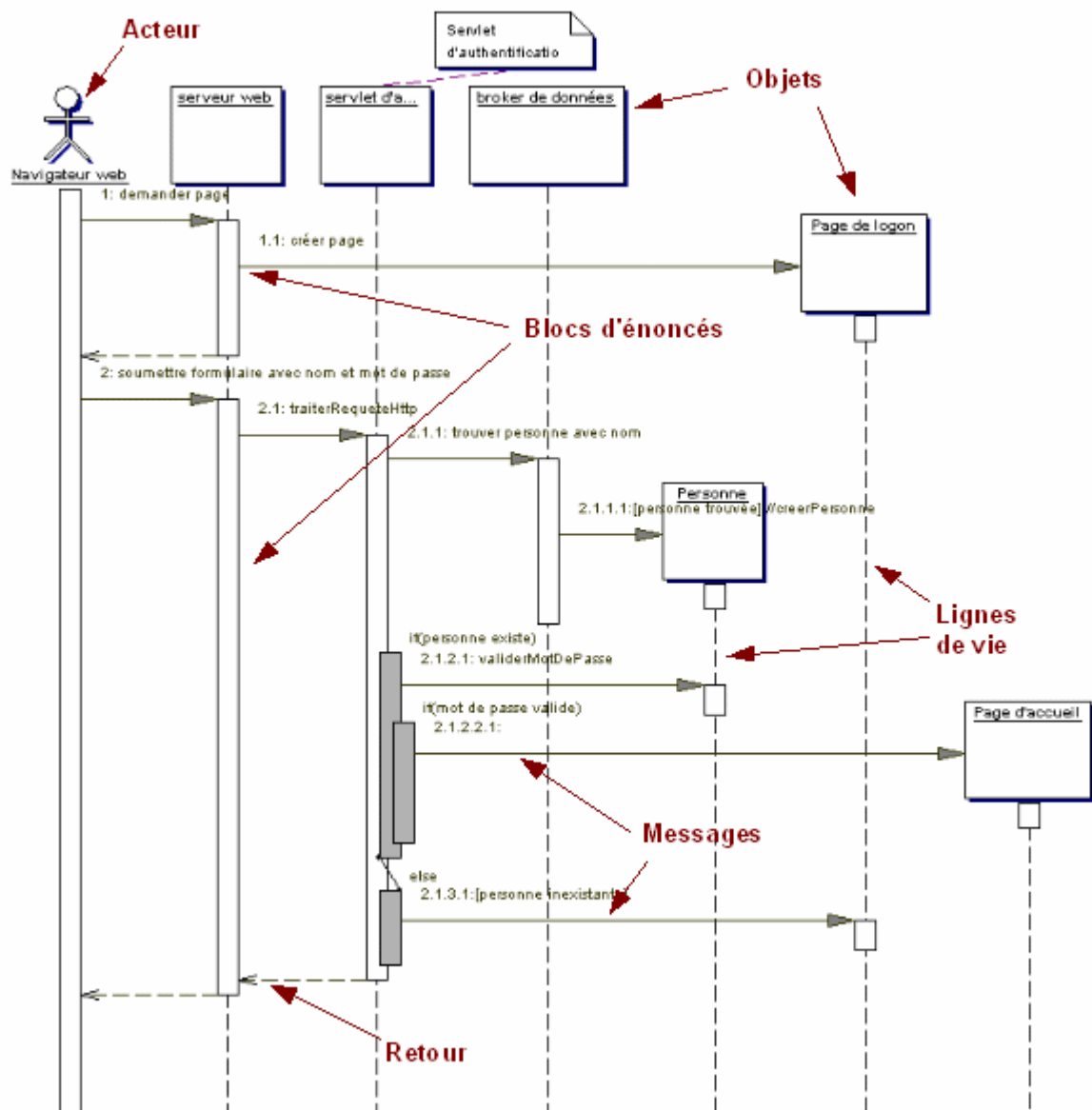
Diagramme d'activités



Notes :

- Dans un diagramme d'activité, on peut placer les activités dans des couloirs qui représentent des systèmes
- Les flèches représentent des transitions
- Une barre de synchronisation sert lorsque deux ou plus de deux activités s'effectuent en parallèle. La barre sert également à indiquer où les activités sont resynchronisées
- Une condition de garde indique le critère utilisé pour suivre une transition ou une autre.

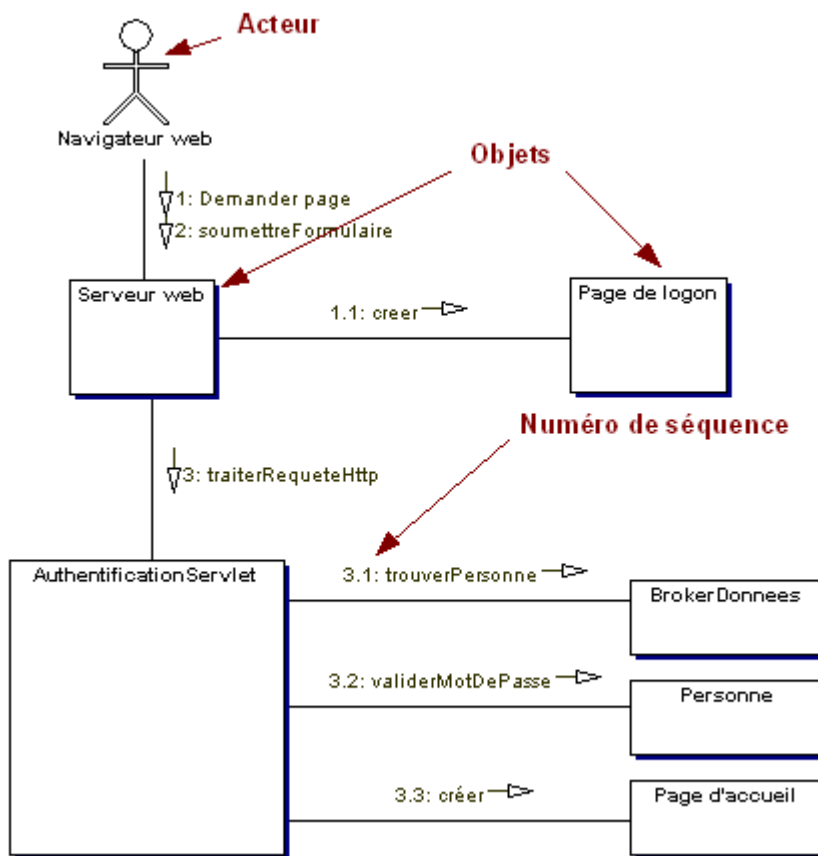
Diagramme de séquence



Notes :

- Une séquence est amorcée par un acteur
- Un message (indiqué par une flèche pleine) est envoyé à un objet
- Un bloc d'énoncés est un regroupement logique de messages et est généralement soumis à une clause
- Un message peut servir à construire un objet. La ligne de vie de cet objet est indiquée par une ligne verticale brisée
- Le retour d'un message est facultatif et indiqué par une flèche brisée

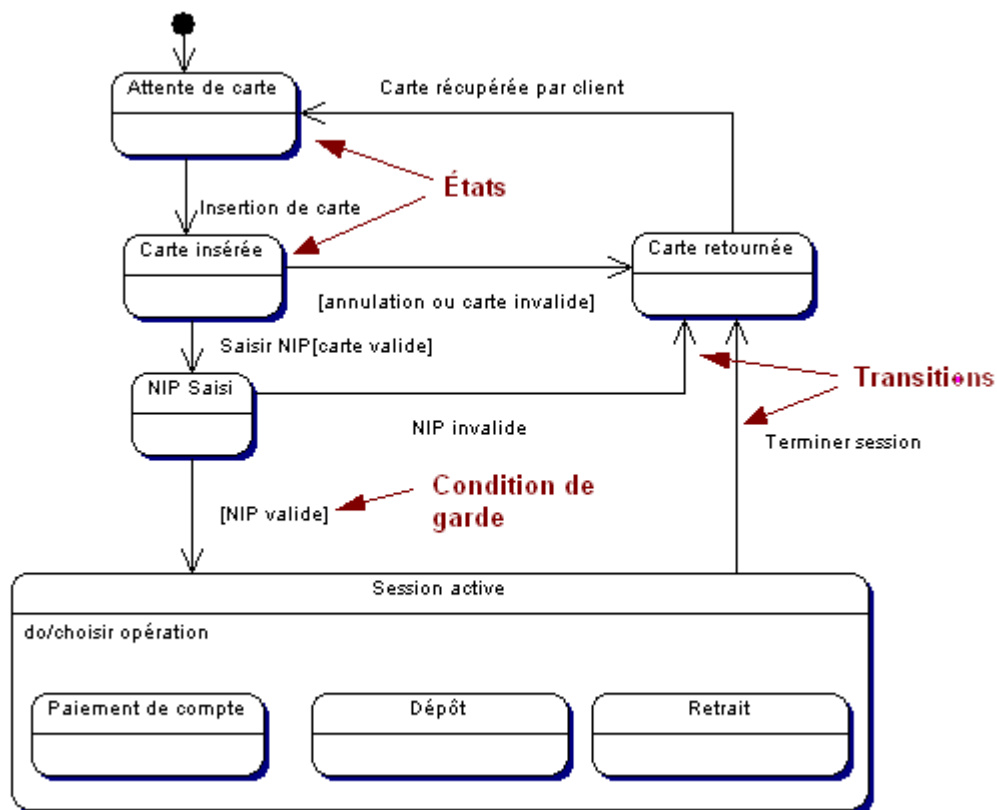
Diagramme de collaboration



Notes :

- Dans un diagramme de collaboration, la séquence des messages est indiquée par leur numéro de séquence

Diagramme d'états transitions

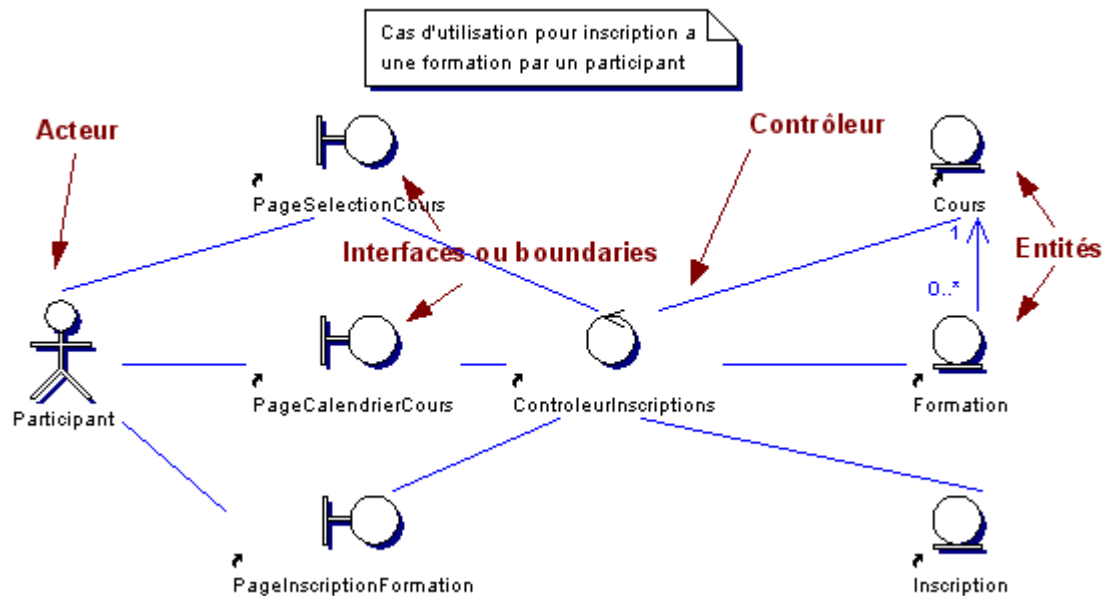


Notes :

- Les états sont représentés par des losanges
- Les transitions (flèches) sont soumises à des conditions de garde
- Les opérations "Paiement de compte", "Dépôt" et "Retrait" s'effectuent à l'intérieur de l'état "Session active"

Autres diagrammes UML

Diagramme de robustesse



Notes :

- Un acteur s'adresse à une interface
- Une interface doit passer par un contrôleur pour agir sur des entités

Diagramme de composants

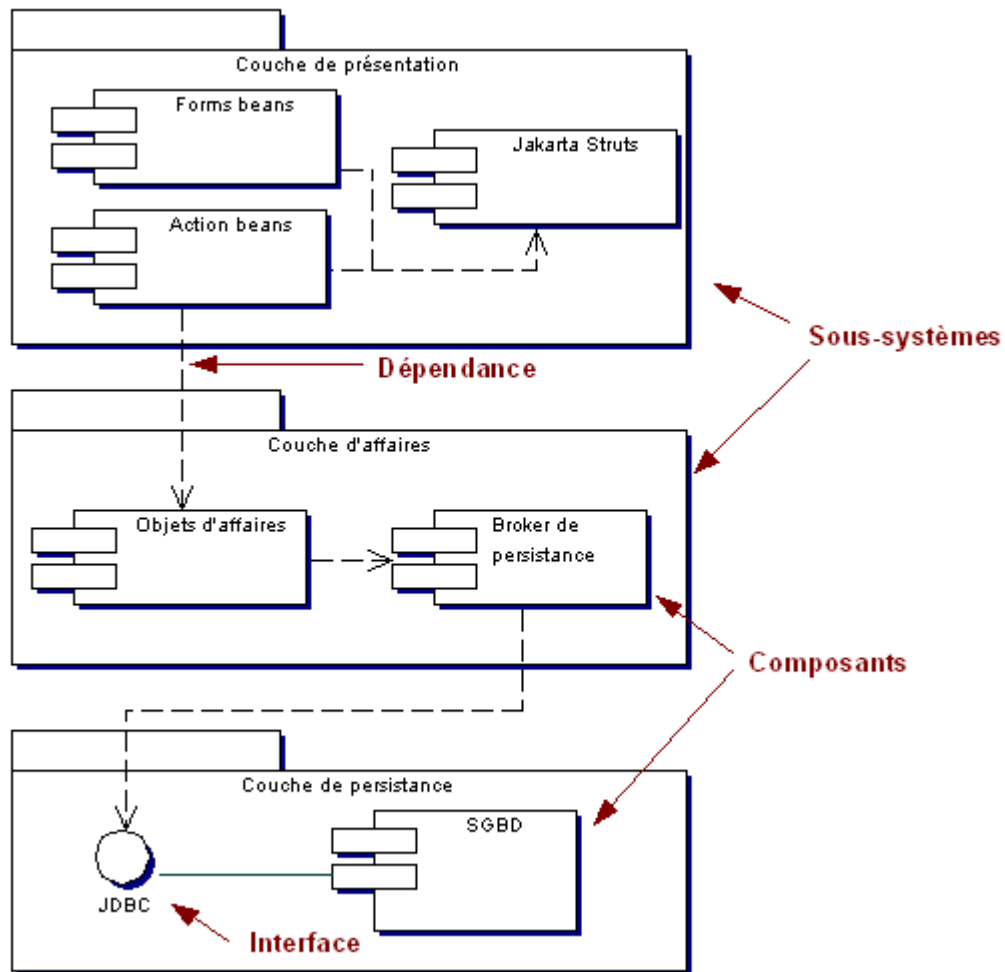


Diagramme de déploiement

