



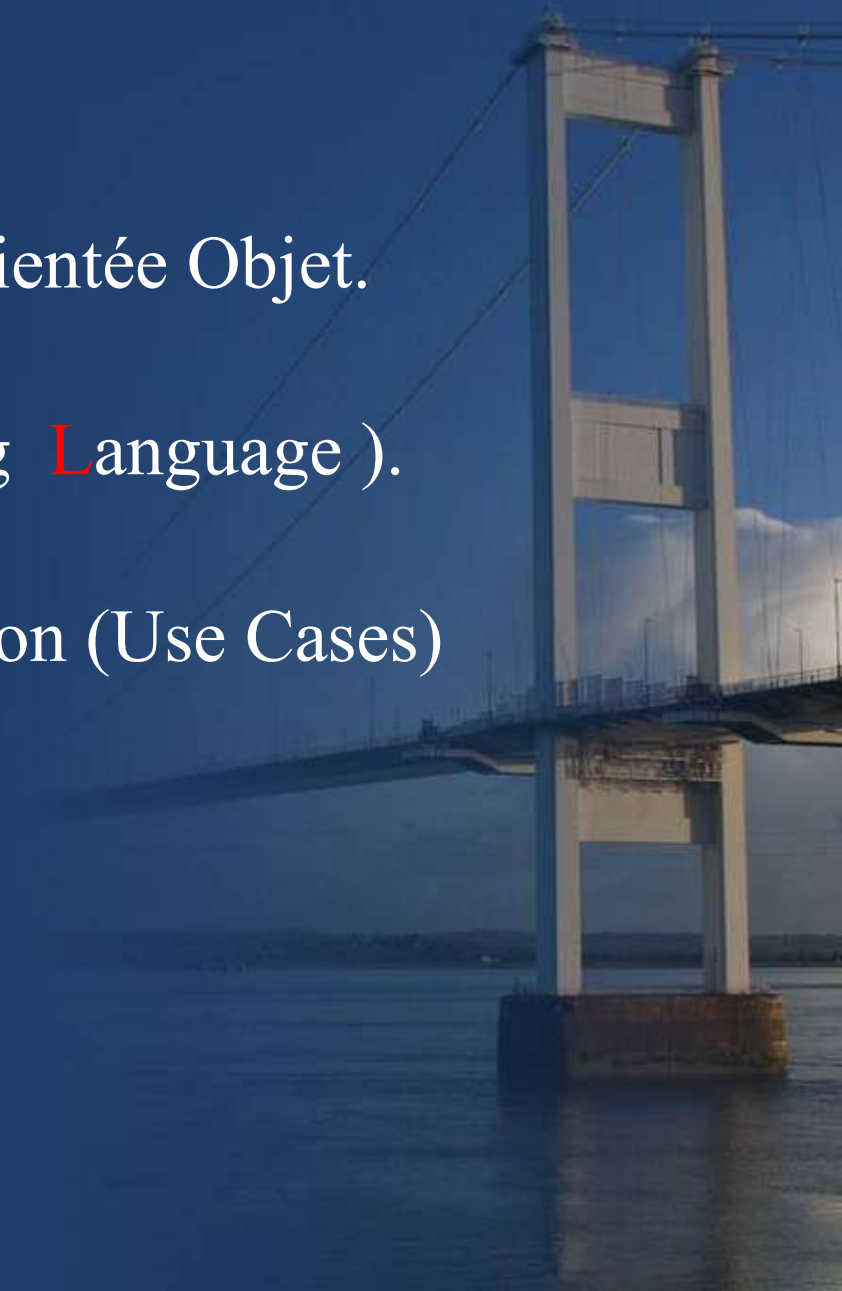
DIAGRAMME DE CAS D'UTILISATION (USE CASES)

PREPARER PAR

HERAGUEMI KAMEL EDDINE

Plan de travail

- Technique de modélisation Orientée Objet.
- UML 2.0 (**U**nified **M**odeling **L**anguage).
- Diagramme de Cas d'utilisation (Use Cases)



Technique de modélisation Orientée Objet.

✓ L'émergence des approches 'objet' (1990-1995)

- Prise de conscience de l'importance d'une approche spécifiquement objet : comment structurer un système sans centrer l'analyse uniquement sur les données ou uniquement sur les traitements (mais sur les deux) ?

- Plusieurs méthodes objet sont apparues durant cette période (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE...) !

✓ Méthodes ont émergé du lot :

- *OMT* (Object Modelling Technique), par James Rumbaugh,
- *OOD* (Object Oriented Design), par Grady Booch ;
- *OOSE* (Object Oriented Software Engineering), par Ivar Jacobson,

Ce sont les ascendants d'**UML**

UML 2.0 (Unified modeling language)

AUTEURS:

James Rumbaugh, Grady Booch et Yvar Jacobson

OBJECTIFS:

- ✓ Faciliter la communication entre les différents acteurs d'un projet
- ✓ Faciliter la communication avec la machine
- ✓ Limiter les ambiguïtés
- ✓ Construire (interpréter les diagrammes pour code)

UML 2.0 (Unified modeling language)/2

DÉFINITION:

- ✓ Un *langage* pas une méthode : UML définit des modes de représentation (diagrammes et notations) mais n'impose pas de démarche standardisée.
- ✓ Un *langage de modélisation objet* permettant de documenter dans des modèles toutes les phases du développement (analyse, conception et implantation).

UML 2.0 (Unified modeling language)/3

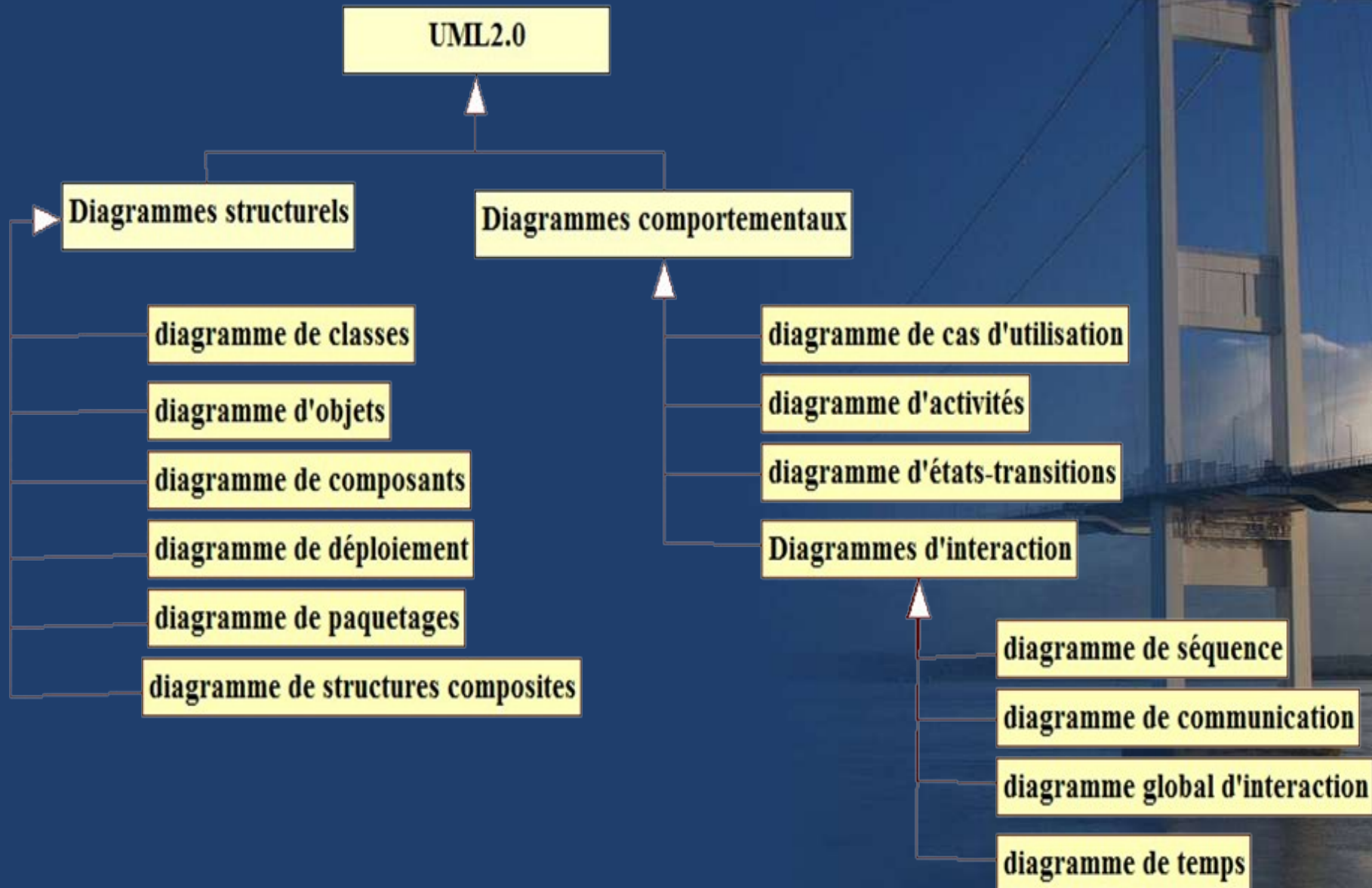


Diagramme de Cas d'utilisation

QU'EST-CE QU'UN CAS D'UTILISATION:

- ✓ Un *cas d'utilisation (Use Cases)* est un diagramme qui modélise une interaction entre le *système informatique* à développer et un *utilisateur* ou *acteur* interagissant avec le système.
- ✓ Permettent de définir les *besoins des utilisateurs* et les *fonctionnalités du système* :
 - Limitation du système,
 - Relations avec son environnement,
 - Fonctions attendues.

Acteurs et cas

QU'EST-CE QU'UN **Acteur**:

Personne ou *Système* qui interagit avec le système étudié en échangeant de l'information.

- ✓ Il possède un rôle par rapport au système,
- ✓ Il peut consulter ou modifier l'état d'un système.

❖ Il existe **4** catégories principales d'acteur:

- ❖ **Acteur PRINCIPAL**: Les personnes qu'utilisent la fonction principale du système.
- ❖ **Acteur SECONDAIRE**: Les personnes qu'effectuent des tâches administratives ou maintenance du système
- ❖ **Matériels Externes**: Les périphériques qui doit être utiliser(Ex :imprimante...)
- ❖ **Autre Systèmes**: Les systèmes avec lesquels le système doit être interagit.

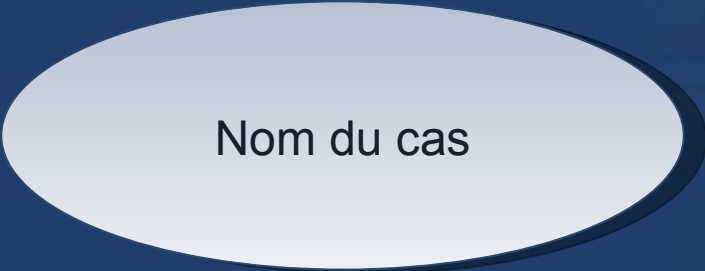


Acteur

Acteurs et cas 2

QU'EST-CE QU'UN **CAS**:

- ✓ Un cas d'utilisation représente une *fonctionnalité fournie par le système*, typiquement décrite sous la forme *Verbe*.
- ✓ Les cas d'utilisation sont représentés par une ellipse contenant leurs nom.



Nom du cas

Acteurs et cas 3

COMMENT IDENTIFIER LES CAS ?

- Chaque cas d'utilisation doit décrire les exigences fonctionnelles du système.
- Chaque cas d'utilisation correspond à une fonction du système (besoins des utilisateurs et possibilités du système).
- *Donc il faut chercher pour chaque acteur :*
 - I. Les différentes intentions métier avec lesquelles il utilise le système,
 - II. Déterminer les services fonctionnels attendus du système.

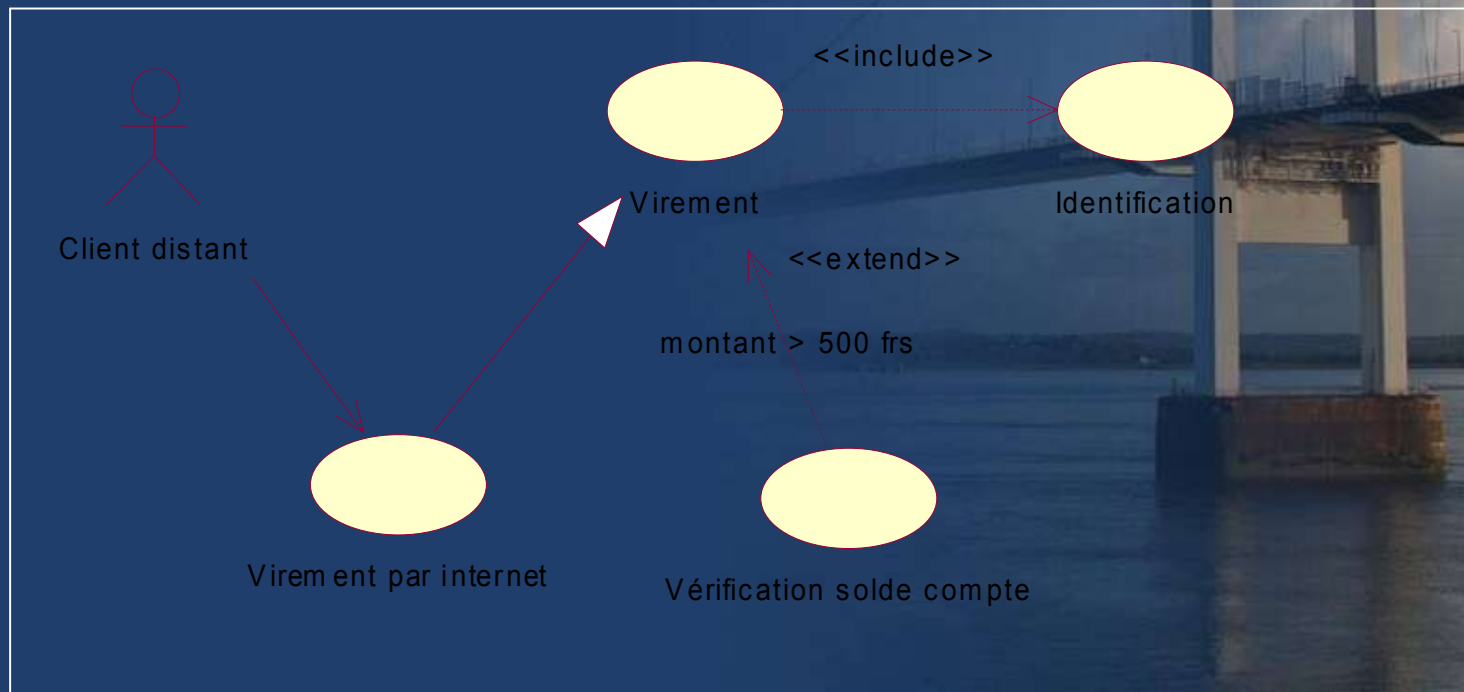
Les relations dans un diagramme cas d'utilisation

Il existe 4 relations principales :

- *La relation d'association*
- *La relation de généralisation*
- *La relation d'inclusion*
- *La relation d'extension*

Exemple

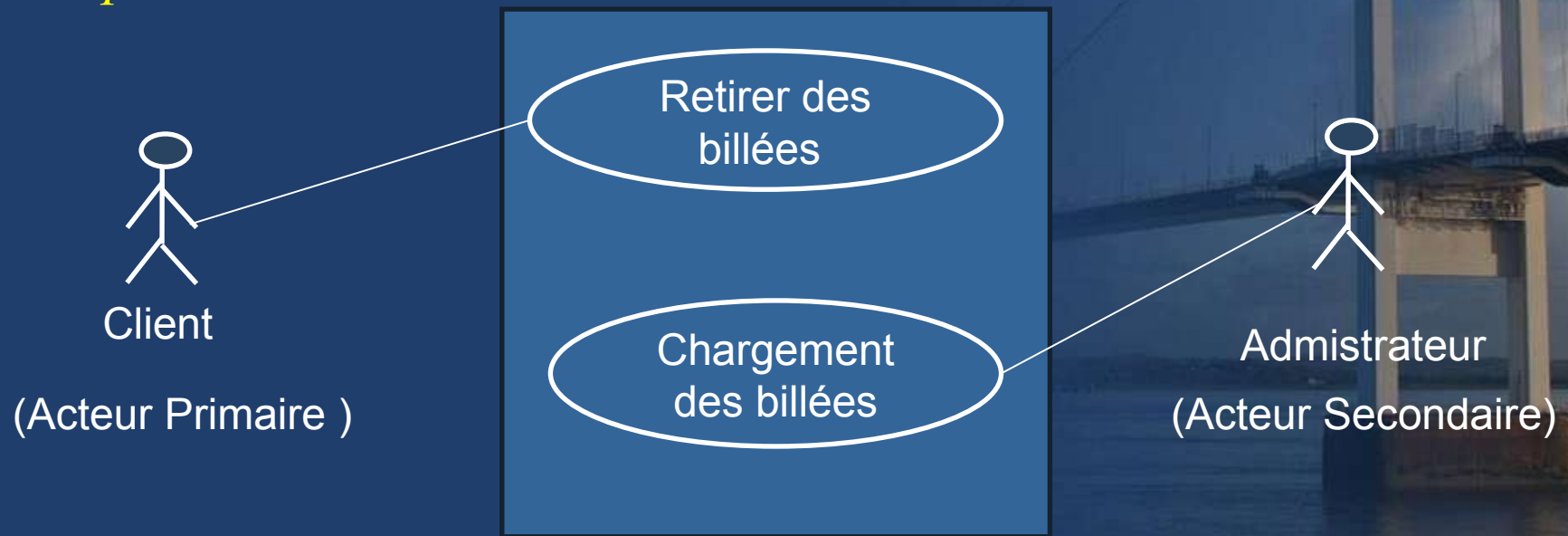
System de
gestion de
Compte
bancaire



Les relations dans un diagramme cas d'utilisation 5

✓ *Une relation d'association* : est un chemin de communication entre un acteur et un cas d'utilisation.

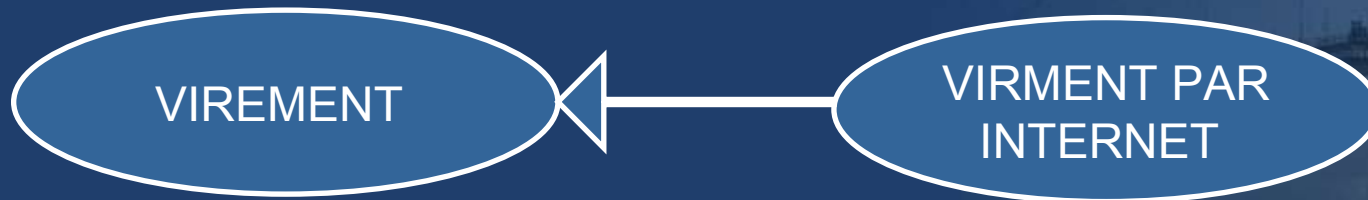
Exemple



Les relations dans un diagramme cas d'utilisation 4

✓ *Héritage (généralisation)* : le cas d'utilisation dérivé est une spécialisation du cas d'utilisation parent (même notion d'héritage entre les classes) ;

Exemple



Le cas **VIREMENT PAR INTERNET** hérite de tout les caractéristique du cas **VIREMENT**

Les relations dans un diagramme cas d'utilisation 2

La relation d'inclusion (« Include »):

un cas d'utilisation a besoin d'un autre cas d'utilisation pour réaliser sa tâche ;

Exemple



Pour qu' un utilisateur réalise le cas de VIREMENT il fait qu'il passe le cas d'IDENTIFICATION .

Donc l'opération virement utilise l'opération d'identification

Les relations dans un diagramme cas d'utilisation 3

La relation d'extension (« Extend ») : le cas source ajoute son comportement au cas destination (cible). L'extension peut être soumise à une condition.

Exemple



Description narrative des cas d'utilisation

- ✓ Comme la plupart des diagrammes UML, le diagramme des cas d'utilisation nécessite souvent une description narrative (textuelle) associée ;
- ✓ Décrire un cas d'utilisation consiste à définir son contexte, et à détailler la communication entre le cas et l'acteur ;
- ✓ Dans la plupart des cas, on peut adopter le plan suivant :
 - *Pré condition* : conditions garantissant que le cas d'utilisation peut démarrer correctement.
 - *Processus ou dialogue* : c'est la description pas à pas des échanges entre l'acteur et le cas d'utilisation.
 - *Arrêt* : liste des fins possibles du cas.
 - *Postcondition* : ensemble de conditions qui doivent être satisfaites à la fin du cas, pour garantir que le système est dans un état cohérent.

Conclusion

- Le diagramme de cas d'utilisation est plus riche que le diagramme *acteurs/flux* de *Merise*.
- En plus des acteurs et des communications, il liste les principales *fonctionnalités attendues*. Il permet de les organiser grâce aux relations d'héritage, d'inclusion et d'extension.
- Avec les *descriptions textuelles* et les *scénarios*, l'analyste dispose de moyens simples pour exprimer de manière semi-formelle les ***besoins fonctionnels et non fonctionnels du système étudié***.