

Aspects dynamique d'UML

Olivier Caron

Polytech Lille
Avenue Paul Langevin Cité Scientifique Lille 1
59655 Villeneuve d'Ascq cedex

<http://ocaron.plil.fr>
Olivier.Caron@polytech-lille.fr



L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes
- Le diagramme de classes n'est donc pas suffisant pour se définir !

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes
- Le diagramme de classes n'est donc pas suffisant pour se définir !
- La notation UML intègre d'autres diagrammes permettant de formaliser (comprendre) la dynamique du système.

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes
- Le diagramme de classes n'est donc pas suffisant pour se définir !
- La notation UML intègre d'autres diagrammes permettant de formaliser (comprendre) la dynamique du système.
 - Les diagrammes d'objets

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes
- Le diagramme de classes n'est donc pas suffisant pour se définir !
- La notation UML intègre d'autres diagrammes permettant de formaliser (comprendre) la dynamique du système.
 - Les diagrammes d'objets
 - Les diagrammes de collaborations

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes
- Le diagramme de classes n'est donc pas suffisant pour se définir !
- La notation UML intègre d'autres diagrammes permettant de formaliser (comprendre) la dynamique du système.
 - Les diagrammes d'objets
 - Les diagrammes de collaborations
 - Les diagrammes de séquences

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes
- Le diagramme de classes n'est donc pas suffisant pour se définir !
- La notation UML intègre d'autres diagrammes permettant de formaliser (comprendre) la dynamique du système.
 - Les diagrammes d'objets
 - Les diagrammes de collaborations
 - Les diagrammes de séquences
 - Les diagrammes d'états

L'aspect dynamique des systèmes

- La prise en compte des **traitements** favorise ou permet l'identification d'informations à gérer
- Peut donc faire évoluer le diagramme de classes
- Le diagramme de classes n'est donc pas suffisant pour se définir !
- La notation UML intègre d'autres diagrammes permettant de formaliser (comprendre) la dynamique du système.
 - Les diagrammes d'objets
 - Les diagrammes de collaborations
 - Les diagrammes de séquences
 - Les diagrammes d'états
 - Les cas d'utilisation

Les diagrammes d'objets

- Décrit des *objets* et des *liens* entre les objets.

Les diagrammes d'objets

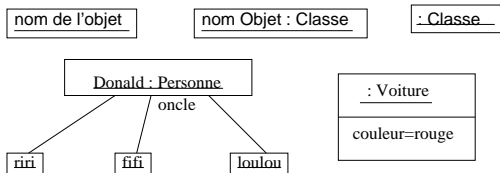
- Décrit des *objets* et des *liens* entre les objets.
- Distinction entre *classes* et *associations*

Les diagrammes d'objets

- Décrit des *objets* et des *liens* entre les objets.
- Distinction entre *classes* et *associations*
- Utilisé pour faciliter la représentation d'objets complexes

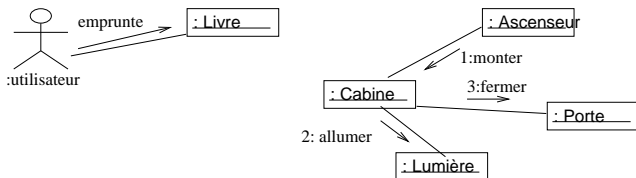
Les diagrammes d'objets

- Décrit des *objets* et des *liens* entre les objets.
- Distinction entre *classes* et *associations*
- Utilisé pour faciliter la représentation d'objets complexes
- Peut être considéré comme un diagramme de représentation statique



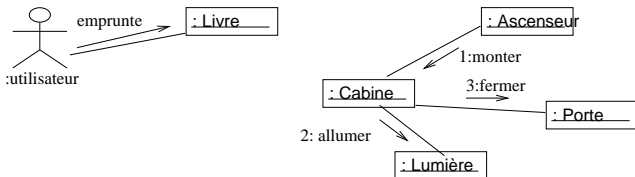
Les diagrammes de collaboration

- Extension du diagramme d'objets



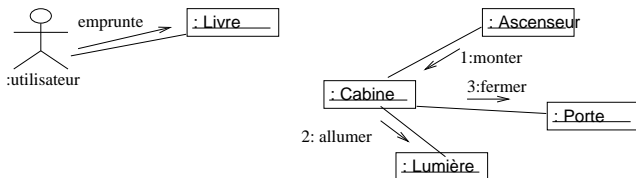
Les diagrammes de collaboration

- Extension du diagramme d'objets
- Interaction entre les objets (envoi de message)



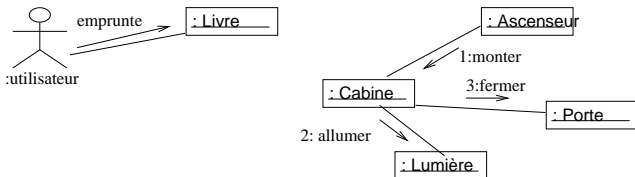
Les diagrammes de collaboration

- Extension du diagramme d'objets
- Interaction entre les objets (envoi de message)
- Structuration spatiale



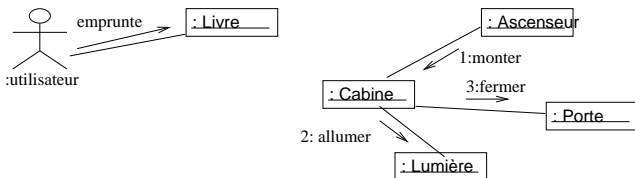
Les diagrammes de collaboration

- Extension du diagramme d'objets
- Interaction entre les objets (envoi de message)
- Structuration spatiale
- Définition d'utilisateurs



Les diagrammes de collaboration

- Extension du diagramme d'objets
- Interaction entre les objets (envoi de message)
- Structuration spatiale
- Définition d'utilisateurs
- Le temps n'est pas représenté de manière implicite



Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.

Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages

Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :

Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :
 - Description générale des messages

Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :
 - Description générale des messages
 - Pas de détail de synchronisation

Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :
 - Description générale des messages
 - Pas de détail de synchronisation
 - Notion de scénarios

Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :
 - Description générale des messages
 - Pas de détail de synchronisation
 - Notion de scénarios
 - Pas de distinction flots de contrôle et flots de données

Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :
 - Description générale des messages
 - Pas de détail de synchronisation
 - Notion de scénarios
 - Pas de distinction flots de contrôle et flots de données
- L'utilisation pour l'implantation (usage informatique) :

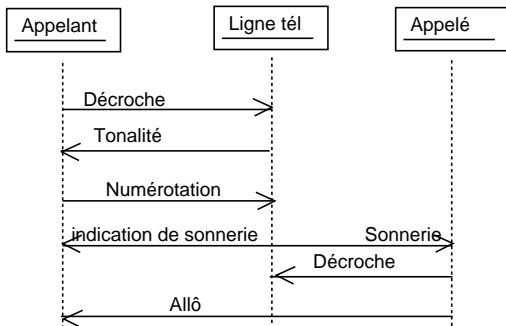
Les diagrammes de séquence

- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :
 - Description générale des messages
 - Pas de détail de synchronisation
 - Notion de scénarios
 - Pas de distinction flots de contrôle et flots de données
- L'utilisation pour l'implantation (usage informatique) :
 - Un message correspond à une méthode ou une procédure

Les diagrammes de séquence

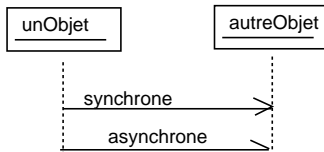
- Interactions entre objets selon un point de vue temporel.
- Echange des messages
- L'utilisation au niveau modélisation :
 - Description générale des messages
 - Pas de détail de synchronisation
 - Notion de scénarios
 - Pas de distinction flots de contrôle et flots de données
- L'utilisation pour l'implantation (usage informatique) :
 - Un message correspond à une méthode ou une procédure
 - Prise en compte des détails de synchronisation

Un exemple de diagramme de sequence



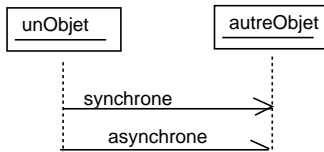
Les types d'envoi de message

- Deux grandes catégories :



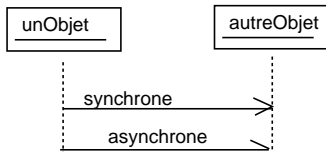
Les types d'envoi de message

- Deux grandes catégories :
 - Les envois synchrones : l'émetteur est bloqué et attend que l'appelé ait fini de traiter le message.



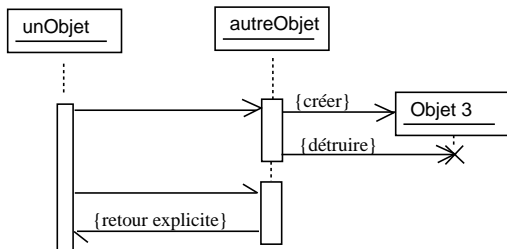
Les types d'envoi de message

- Deux grandes catégories :
 - Les envois synchrones : l'émetteur est bloqué et attend que l'appelé ait fini de traiter le message.
 - Les envois asynchrones pour lesquels l'émetteur n'est pas bloqué et peut continuer son exécution (demi-flèche).



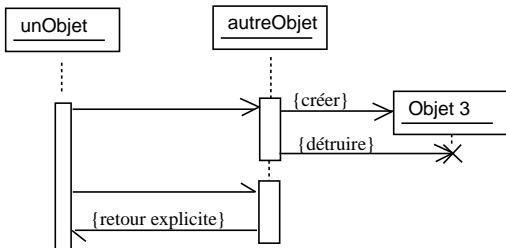
Les diagrammes de séquence (suite...)

- Description de création d'instances, de destruction



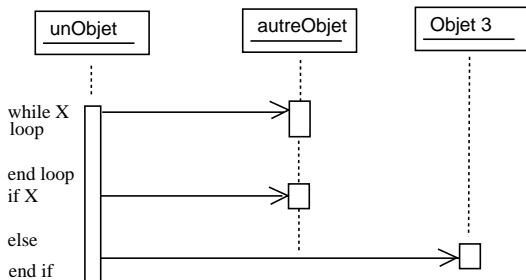
Les diagrammes de séquence (suite...)

- Description de création d'instances, de destruction
- Description des périodes d'activité



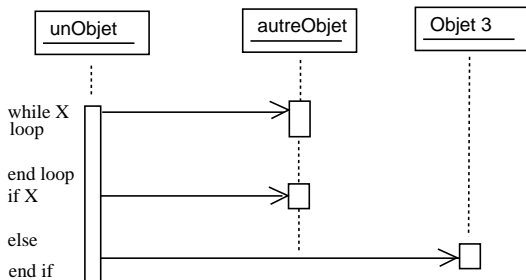
Les diagrammes de séquence (... et fin)

- Description des structures de contrôle



Les diagrammes de séquence (... et fin)

- Description des structures de contrôle



- Préférable d'avoir plusieurs scénarios que de 'if' imbriqués !

Les diagrammes d'états

- Automates d'états finis

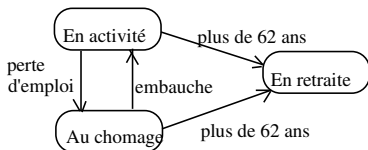
Les diagrammes d'états

- Automates d'états finis
- Comment des événements influent sur des états (liés à des objets) ?

Les diagrammes d'états

- Automates d'états finis
- Comment des événements influent sur des états (liés à des objets) ?
- Un exemple (très simple !) :

● Etat initial Etat ● Etat final



Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)

Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)
- Décrire sous forme d'actions, le **comportement** d'un système du point de vue d'un **utilisateur**.

Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)
- Décrire sous forme d'actions, le **comportement** d'un système du point de vue d'un **utilisateur**.
- Technique qui favorise la détermination des besoins.

Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)
- Décrire sous forme d'actions, le **comportement** d'un système du point de vue d'un **utilisateur**.
- Technique qui favorise la détermination des besoins.
- Permet de comprendre, structurer et déterminer les besoins.

Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)
- Décrire sous forme d'actions, le **comportement** d'un système du point de vue d'un **utilisateur**.
- Technique qui favorise la détermination des besoins.
- Permet de comprendre, structurer et déterminer les besoins.
- Les besoins sont souvent exprimées avec des expressions du genre :

Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)
- Décrire sous forme d'actions, le **comportement** d'un système du point de vue d'un **utilisateur**.
- Technique qui favorise la détermination des besoins.
- Permet de comprendre, structurer et déterminer les besoins.
- Les besoins sont souvent exprimées avec des expressions du genre :
 - Le système devra faire. . .

Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)
- Décrire sous forme d'actions, le **comportement** d'un système du point de vue d'un **utilisateur**.
- Technique qui favorise la détermination des besoins.
- Permet de comprendre, structurer et déterminer les besoins.
- Les besoins sont souvent exprimées avec des expressions du genre :
 - Le système devra faire. . .
 - Le système devrait faire. . .

Les cas d'utilisation (use Cases)

- Formalisé par Ivar Jacobson (1992)
- Décrire sous forme d'actions, le **comportement** d'un système du point de vue d'un **utilisateur**.
- Technique qui favorise la détermination des besoins.
- Permet de comprendre, structurer et déterminer les besoins.
- Les besoins sont souvent exprimées avec des expressions du genre :
 - Le système devra faire. . .
 - Le système devrait faire. . .
 - Il faut absolument que. . .

Les acteurs des cas d'utilisation (1/2)

- Les différents types d'acteurs

Illustration avec un distributeur de billets :

Les acteurs des cas d'utilisation (1/2)

- Les différents types d'acteurs

Illustration avec un distributeur de billets :

Les acteurs principaux Ce sont les personnes qui utilisent les fonctions principales du système.

Exemple : les clients

Les acteurs des cas d'utilisation (1/2)

- Les différents types d'acteurs

Illustration avec un distributeur de billets :

Les acteurs principaux Ce sont les personnes qui utilisent les fonctions principales du système.

Exemple : les clients

Les acteurs secondaires Ce sont les personnes qui effectuent des tâches administratives ou de maintenance.

Exemple : la personne qui recharge en billets le distributeur

Les acteurs des cas d'utilisation (2/2)

- Les différents types d'acteurs :

Les acteurs des cas d'utilisation (2/2)

- Les différents types d'acteurs :
 - Le matériel externe Les dispositifs matériels nécessaires au fonctionnement.
Exemple : l'imprimante (pas uniquement !)

Les acteurs des cas d'utilisation (2/2)

- Les différents types d'acteurs :

Le matériel externe Les dispositifs matériels nécessaires au fonctionnement.

Exemple : l'imprimante (pas uniquement !)

Les autres systèmes Les systèmes à faire interagir

Exemple : le groupement bancaire

Les relations entre cas d'utilisation

La relation de communication la participation de l'acteur est signalée par une flèche. Le sens de la flèche indique l'initiateur de l'interaction.

Les relations entre cas d'utilisation

La relation de communication la participation de l'acteur est signalée par une flèche. Le sens de la flèche indique l'initiateur de l'interaction.

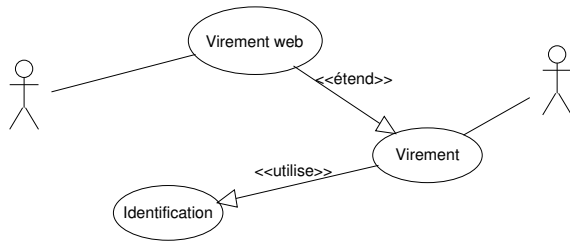
La relation d'initialisation signifie qu'un cas d'utilisation **utilise** un autre cas d'utilisation.

Les relations entre cas d'utilisation

- La relation de communication la participation de l'acteur est signalée par une flèche. Le sens de la flèche indique l'initiateur de l'interaction.
- La relation d'initialisation signifie qu'un cas d'utilisation **utilise** un autre cas d'utilisation.
- La relation d'extension signifie que le cas d'utilisation étend le comportement du cas d'initialisation d'extension.

Exemple de cas d'utilisation

- Les cas d'utilisation sont définis par un nom dans une ellipse



Transition vers les objets (classes)

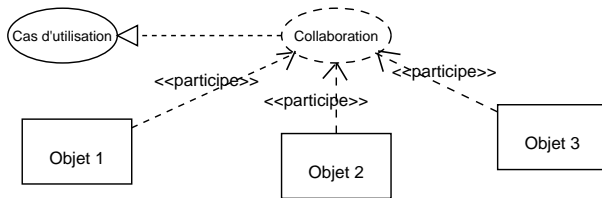
- Passage de collaboration à cas d'utilisation

Transition vers les objets (classes)

- Passage de collaboration à cas d'utilisation
- Identification des objets

Transition vers les objets (classes)

- Passage de collaboration à cas d'utilisation
- Identification des objets
- Description générique :



Les diagrammes de composants

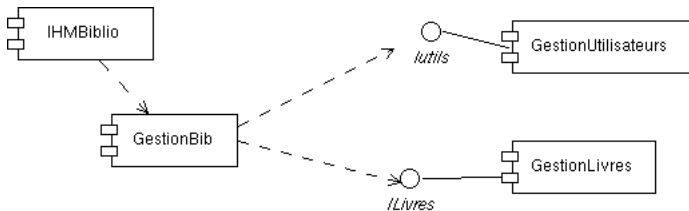
- correspond à du binaire en général

Les diagrammes de composants

- correspond à du binaire en général
- implémente des interfaces UML

Les diagrammes de composants

- correspond à du binaire en général
- implémente des interfaces UML
- relation de dépendance avec d'autres composants.



Les diagrammes de déploiement

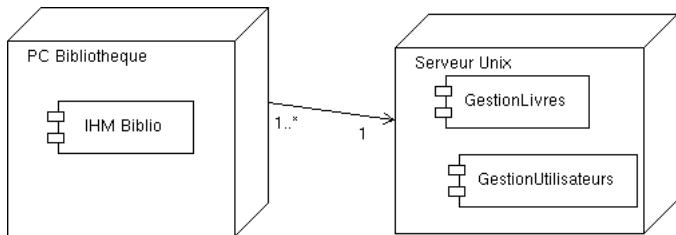
- Définition de **types** de noeud (Node)

Les diagrammes de déploiement

- Définition de **types** de noeud (Node)
- noeud = zone ou des composants seront déployées

Les diagrammes de déploiement

- Définition de **types** de noeud (Node)
- noeud = zone ou des composants seront déployées
- Association entre noeuds ?

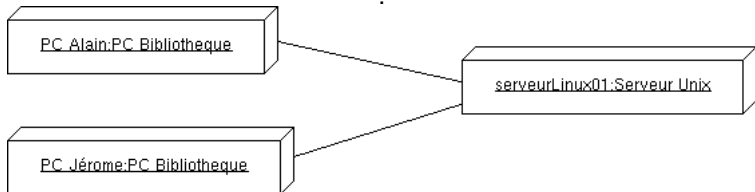


Les diagrammes d'instance de déploiement

- Décrit une configuration possible

Les diagrammes d'instance de déploiement

- Décrit une configuration possible
- conforme au diagramme de déploiement



Conclusion

- **Introduction à UML**

Conclusion

- **Introduction** à UML
- Pas nécessaire d'utiliser tous les diagrammes !

Conclusion

- **Introduction** à UML
- Pas nécessaire d'utiliser tous les diagrammes !
- Distinction de la notion de notation Objet et méthodologie

Conclusion

- **Introduction** à UML
- Pas nécessaire d'utiliser tous les diagrammes !
- Distinction de la notion de notation Objet et méthodologie
- Difficulté à faire le lien entre les différents diagrammes

Conclusion

- **Introduction** à UML
- Pas nécessaire d'utiliser tous les diagrammes !
- Distinction de la notion de notation Objet et méthodologie
- Difficulté à faire le lien entre les différents diagrammes
- Intérêt grandissant selon la complexité du projet

Conclusion

- **Introduction** à UML
- Pas nécessaire d'utiliser tous les diagrammes !
- Distinction de la notion de notation Objet et méthodologie
- Difficulté à faire le lien entre les différents diagrammes
- Intérêt grandissant selon la complexité du projet
- Directives pour la génération de logiciels