

heig-vd MSE MASTER OF SCIENCE
IN ENGINEERING

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

**Modélisation d'applications industrielles avec
UML**

ACOO Analyse, Conception et développement Orientés Objet de logiciels de commande

heig-vd MSE MASTER OF SCIENCE
IN ENGINEERING

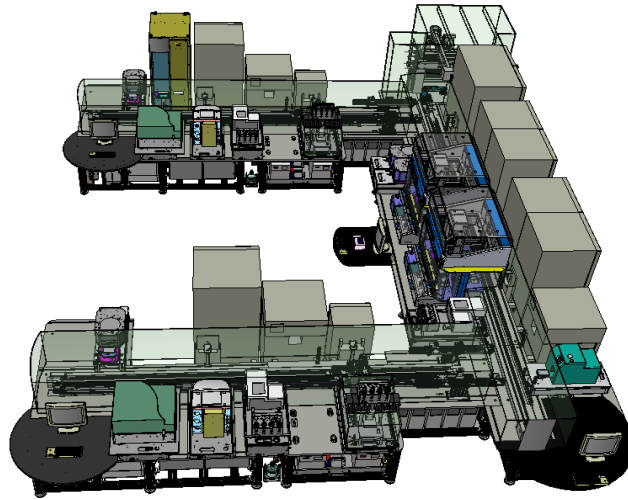
Thèmes

- Motivations à l'origine d'UML.
- Introduction au formalisme UML.
 - Modélisation de structure.
 - Modélisation de comportement.
- Principes de conception orientées objet.

Modélisation d'applications industrielles avec UML

La complexité logicielle

Exemple d'application industrielle



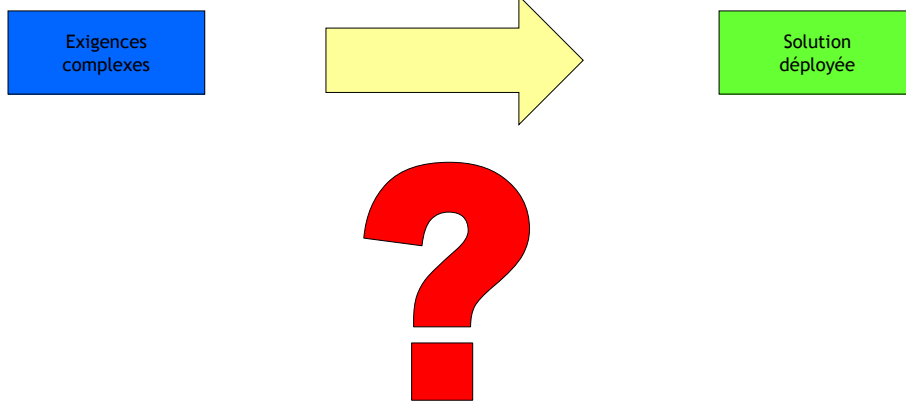
Modélisation d'applications industrielles avec UML

Nature complexe du logiciel

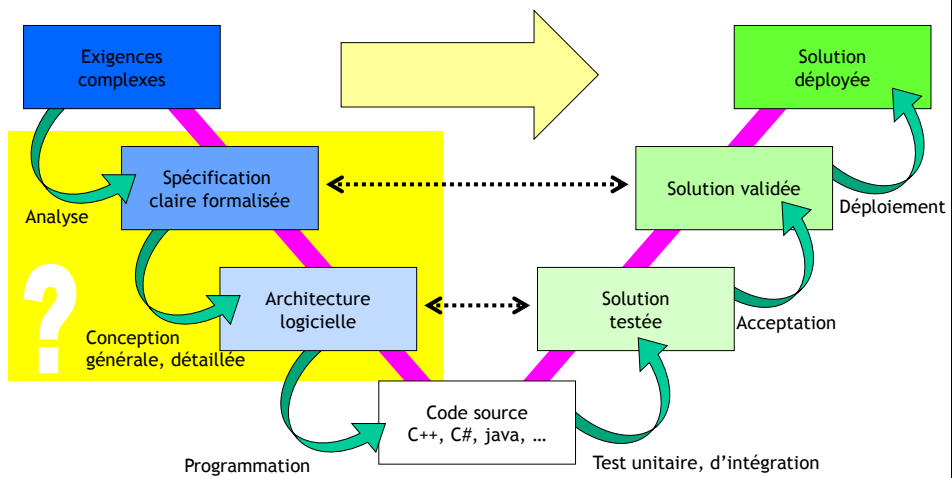
- Complexité des problèmes à traiter
 - Myriades d'exigences fonctionnelles.
 - Exigences non fonctionnelles, souvent contradictoires.
 - Difficultés de communication
- Complexité de la réalisation
 - Difficultés techniques de la programmation.
 - Développement en parallèle à plusieurs.
 - Besoin de créer une solution simple d'emploi.
- Divergence des systèmes discrets
 - petite erreur => conséquences énormes

Modélisation d'applications industrielles avec UML

Objectif du développement logiciel



Rôle de l'analyse et conception Dans le processus de développement



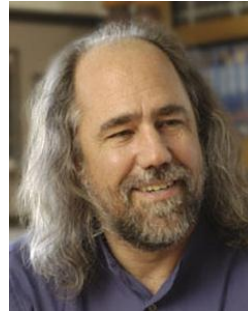
Quelques fondateurs de la modélisation orientée objet



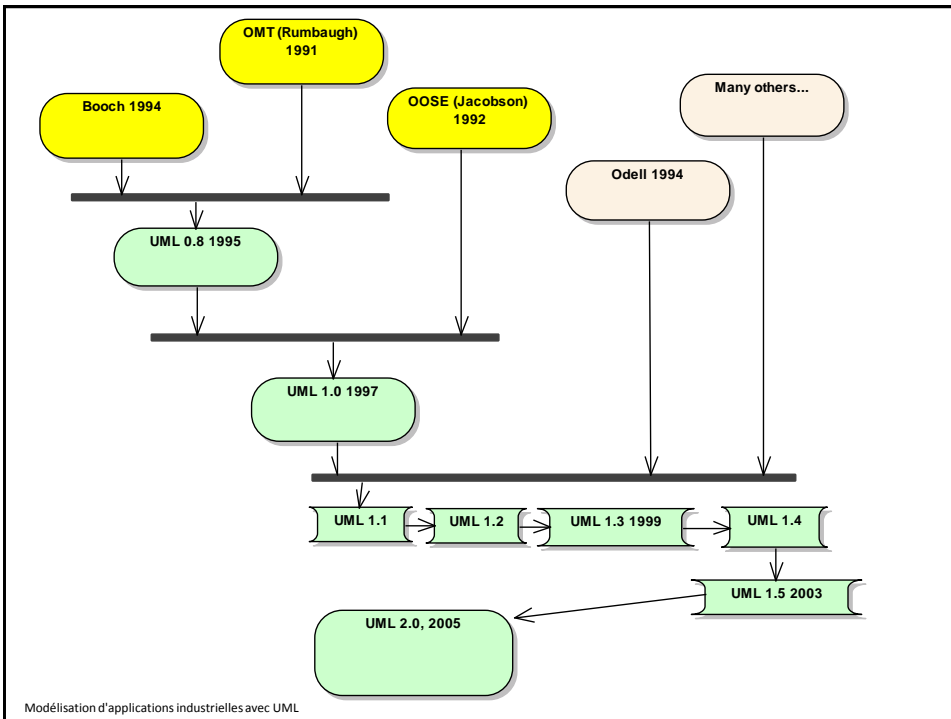
Ivar Jacobson
OOSE



James Rumbaugh
OMT



Grady Booch
Booch

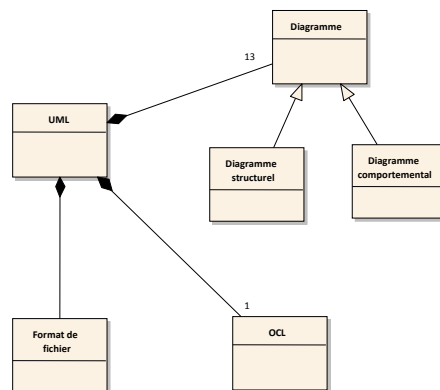


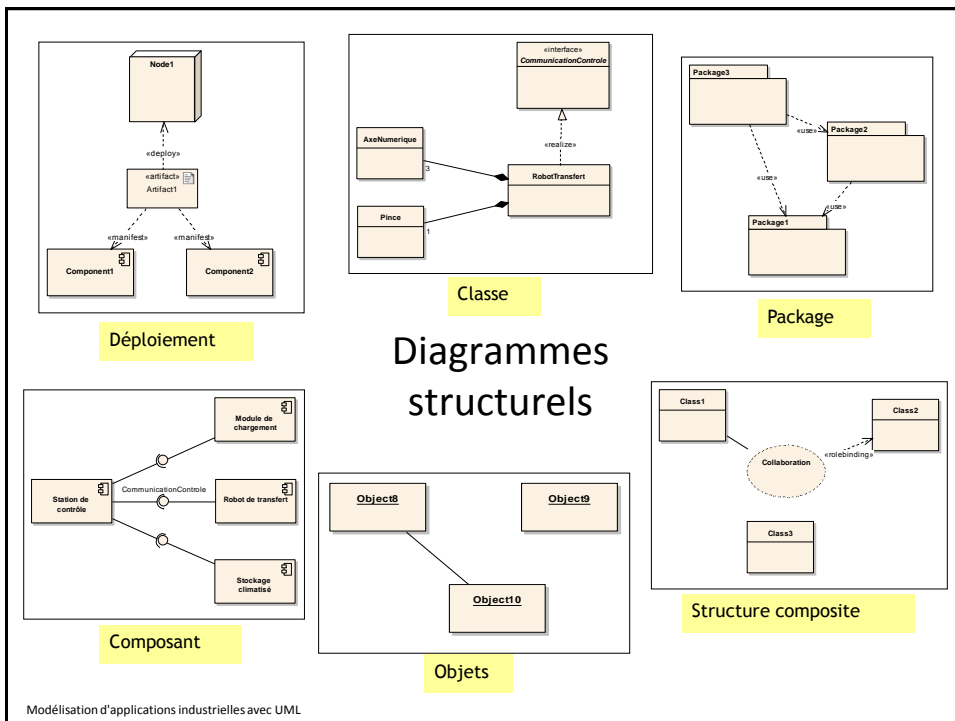
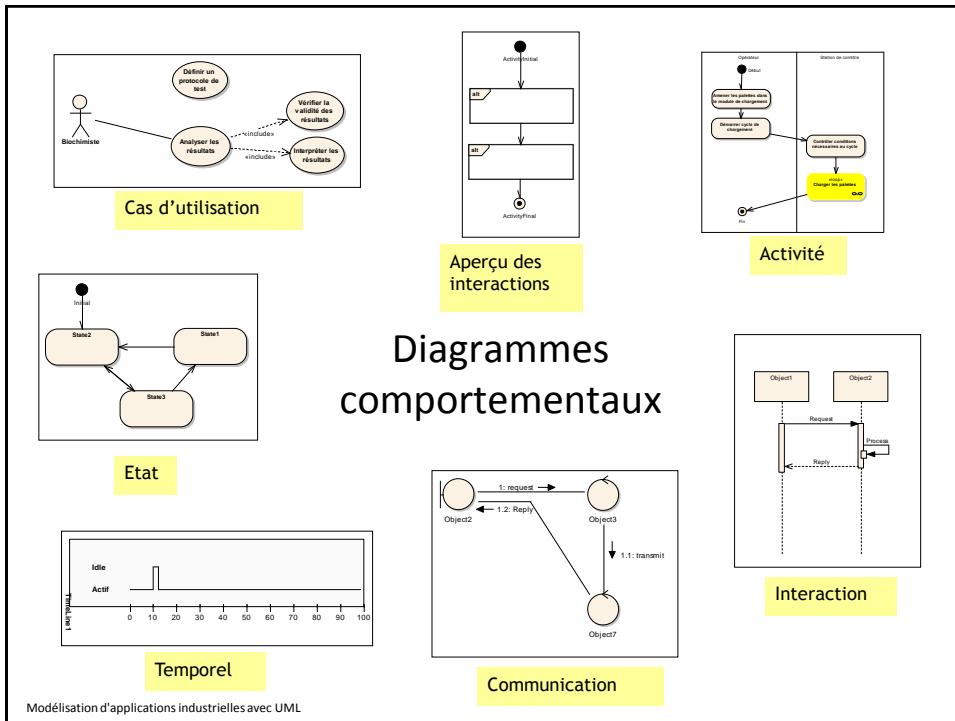
Qu'est ce qu'UML

- UML : Unified Modeling Language
 - Langage de Modélisation Unifié.
 - Appliqué à l'analyse et à la conception des logiciels.
 - Forte coloration orientée objet.
 - Langage essentiellement graphique.
 - Facile à lire et à comprendre.
- En clair
 - UML: norme qui définit les diagrammes et les conventions à utiliser lors de la construction de modèles décrivant la structure et le comportement d'un logiciel.
 - Les modèles sont des diagrammes constitués d'éléments graphiques et de texte.
 - UML n'est pas une méthode, mais un langage.

Le contenu d'UML ... en UML

- 13 types de diagrammes
- Classés en 2 catégories
 - 7 diagrammes comportementaux
 - 6 diagrammes structurels
- Un langage de spécification de contraintes
 - OCL : Object Constraint Language
- Une spécification de format de fichier.





UML, un langage pour

- Visualiser
 - chaque symbole graphique a une sémantique.
- Spécifier
 - de manière précise et complète, sans ambiguïté.
- Construire
 - les classes, les relations SQL peuvent être générées automatiquement
- Documenter
 - les différents diagrammes, notes, contraintes, exigences seront présentés dans un document.

Quelques diagrammes très utilisés

- Analyse de la fonctionnalité
 - Cas d'utilisation
 - Diagrammes d'activité
- Conception de la structure
 - Diagramme de classes
- Conception du comportement
 - Diagramme de séquence

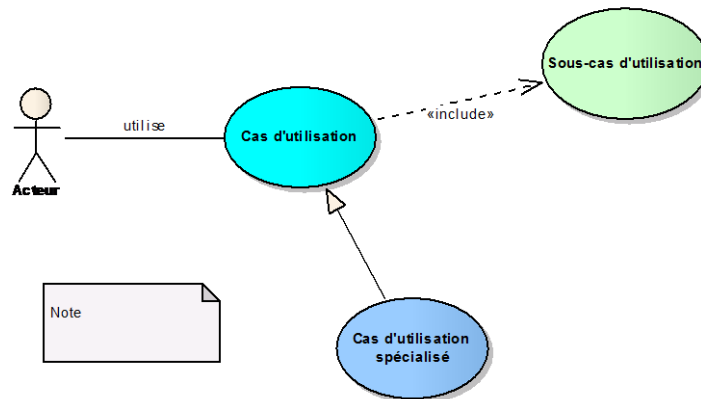
Cas d'utilisation

Principes

- Définition
 - Description d'un cheminement complet d'utilisation du logiciel.
 - Aboutit à un résultat concret pour l'utilisateur.
- Utilisation
 - Analyse de la fonctionnalité.
 - Identifier les façons d'utiliser le système.

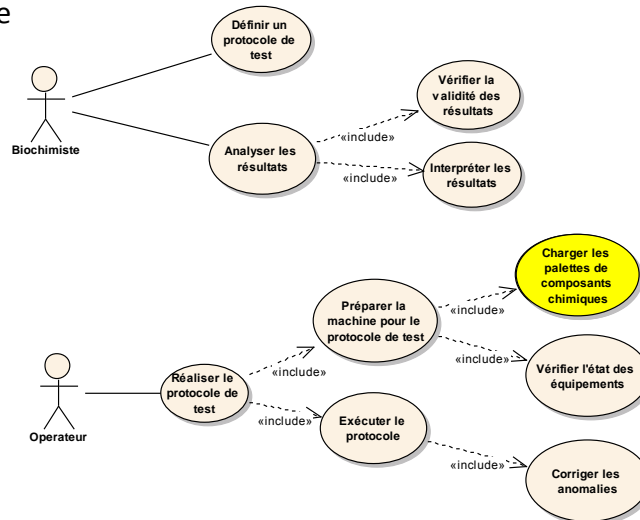
Cas d'utilisation

Symboles



Cas d'utilisation

Exemple



Diagrammes d'activités

Principes

- Définition
 - Description d'un flot d'activité.
 - Utilisation des lignes d'eau (swimlane) pour structurer.
 - Met en évidence les interactions pour réaliser une tâche.
- Utilisation
 - Analyse de la fonctionnalité.
 - Très utilisé pour expliciter un cas d'utilisation.

heig-vd MSE MASTER OF SCIENCE IN ENGINEERING

Diagrammes d'activités

Symboles

The diagram illustrates the symbols used in UML Activity Diagrams across three swimlanes: Acteur, SousSystème1, and SousSystème2. In the Acteur swimlane, there is a 'Début' (Start) symbol (a solid black circle) and a 'Fin' (End) symbol (a circle with a smaller solid black circle inside). In the SousSystème1 swimlane, there is an 'Action1' symbol (a rounded rectangle) and an 'Action3 Parallèle' symbol (a rounded rectangle with a thick horizontal bar at the top). In the SousSystème2 swimlane, there is an 'Action2 Parallèle' symbol (a rounded rectangle with a thick horizontal bar at the top). Arrows show the flow from 'Début' to 'Action1', from 'Action1' to a fork node (a thick horizontal bar), which then splits into two parallel paths leading to 'Action3 Parallèle' and 'Action2 Parallèle'. These two paths merge at a join node (a thick horizontal bar), which then leads to 'Fin'.

Modélisation d'applications industrielles avec UML

heig-vd MSE MASTER OF SCIENCE IN ENGINEERING

Diagramme d'activités

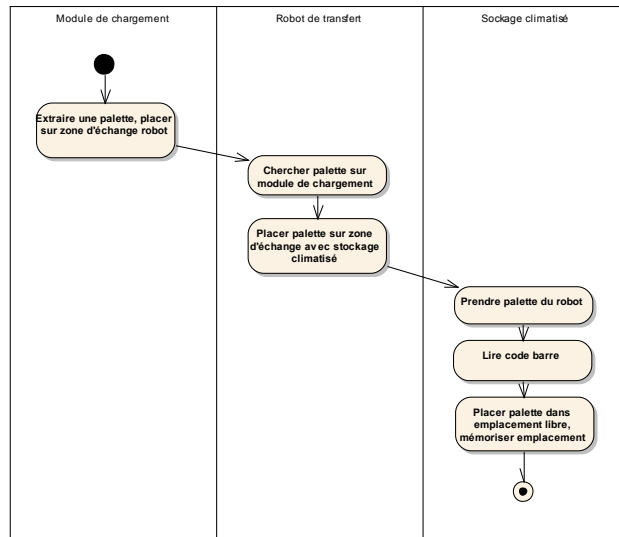
Exemple

The diagram shows an example of an activity diagram with two swimlanes: Opérateur and Station de contrôle. In the Opérateur swimlane, the flow starts at 'Début' (Start), goes to 'Amener les palettes dans le module de chargement', then to 'Démarrer cycle de chargement'. An arrow from 'Démarrer cycle de chargement' points to 'Contrôler conditions nécessaires au cycle' in the Station de contrôle swimlane. From there, the flow goes to a loop symbol (a rounded rectangle with a yellow background and an infinity symbol) labeled '«loop» Charger les palettes'. An arrow from the loop points back to 'Fin' (End) in the Opérateur swimlane.

Modélisation d'applications industrielles avec UML

Diagramme d'activités

Exemple



Modélisation d'applications industrielles avec UML

Diagrammes de classes

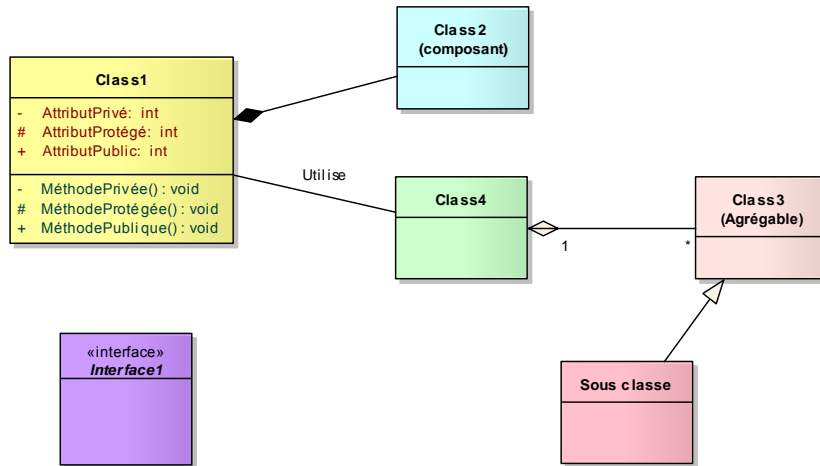
Principes

- Définition
 - Vue structurelle du logiciel
 - Représentation de classes et de leurs relations
 - Proche de l'implémentation
- Utilisation
 - Très utilisé pour la conception de la structure du logiciel
 - Préférer des vues simples explicitant quelques aspects du logiciel
 - Outil de travail itératif

Modélisation d'applications industrielles avec UML

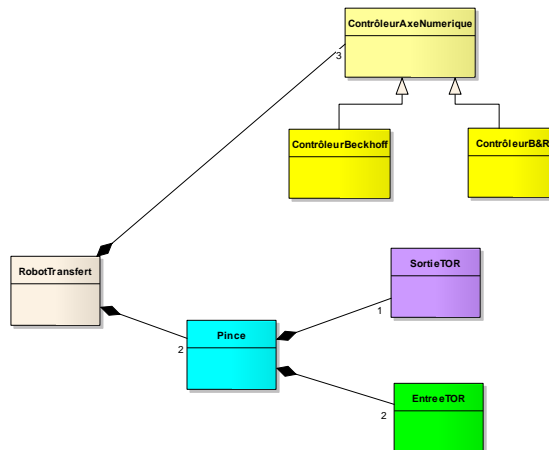
Diagrammes de classes

Symboles



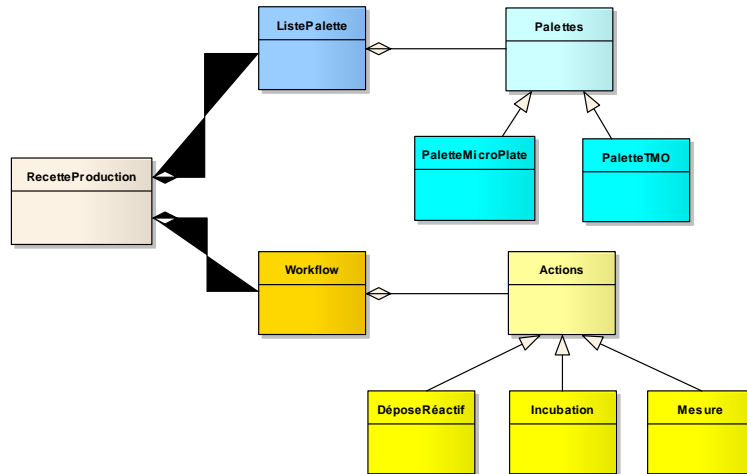
Diagrammes de classes

Exemple – structure du contrôle de la machine



Diagrammes de classes

Exemple – définition des séquences



Modélisation d'applications industrielles avec UML

Diagrammes de séquences

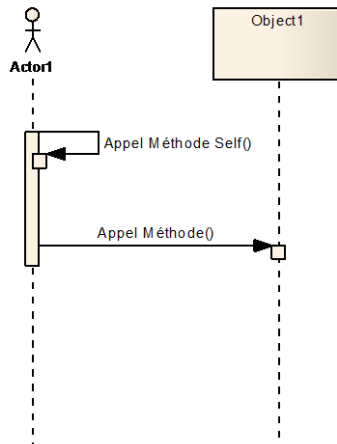
Principes

- Définition
 - Vue comportementale du logiciel
 - Représentation des appels de méthodes entre objets
 - Proche de l'implémentation
- Utilisation
 - Très utilisé pour décrire un comportement du logiciel.
 - Préférer des vues simples explicitant une séquence particulière.
 - Ne l'utiliser que lorsqu'il y a un vrai travail d'analyse de séquence.
 - On ne documente pas tous les enchaînements d'appels !

Modélisation d'applications industrielles avec UML

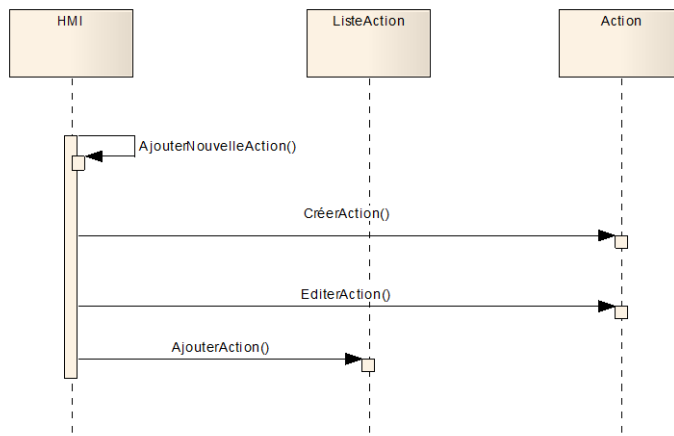
Diagrammes de séquences

Symboles



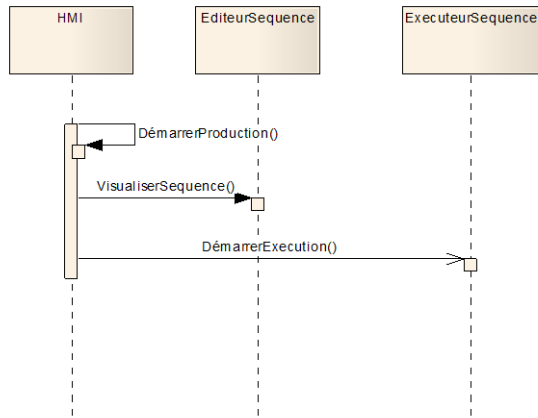
Diagrammes de séquences

Exemple - Edition



Diagrammes de séquences

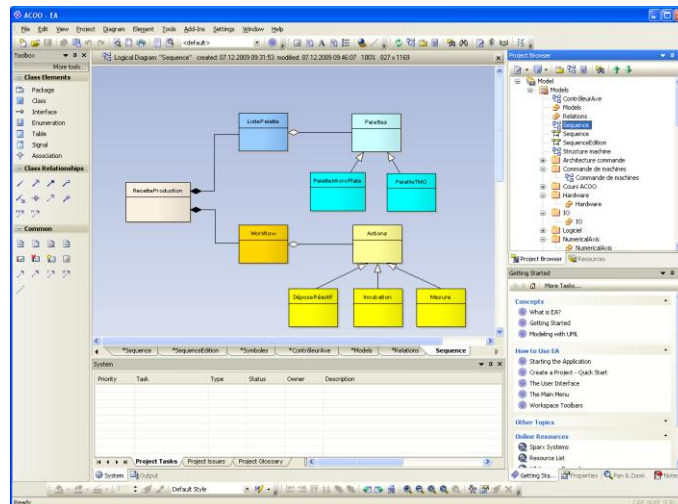
Exemple



Modélisation d'applications industrielles avec UML

Outils UML pour la conception orientée objet

Illustration avec EnterpriseArchitect



Modélisation d'applications industrielles avec UML

Pourquoi modéliser ?

- **Modèle**
 - Vue simplifiée de la réalité.
 - Permet de comprendre synthétiquement le système à développer.
- **Le modèle permet de**
 - Visualiser le système
 - Spécifier la structure et le comportement du système.
 - comme il est ou comme il devrait être.
 - ce qu'il fait ou ce qu'il devrait faire.
 - Valider le modèle vis à vis des clients
 - Fournir un guide pour la construction du système.
 - Documenter le système et les décisions prises.

Principes de conception orientée objet

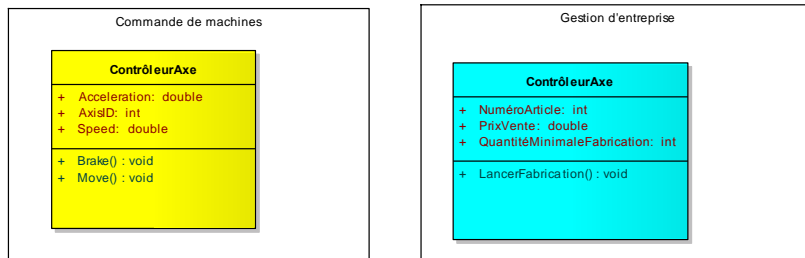
Organiser la connaissance

- **Systèmes complexes**
 - Trop d'information pour pouvoir les mobiliser simultanément.
 - Nécessité de structurer pour rendre compréhensible.
- **Plusieurs axes d'organisation de la connaissance**
 - Abstraction
 - Encapsulation
 - Modularité
 - Hiérarchie

Principes de conception orientée objet

Abstraction

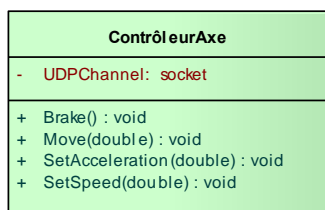
- Principe
 - Vision simplifiée d'un objet ou d'une réalité d'un domaine.
 - Focalisée sur les caractéristiques importantes pour l'application visée.
- Exemple
 - Système de commande d'axe numérique



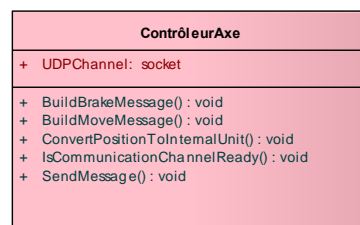
Principes de conception orientée objet

Encapsulation

- Principe
 - Rendre un objet utilisable sans connaissance de sa structure interne.
 - Par la seule connaissance de son interface.
 - Meilleure indépendance entre les objets d'un système complexe.
- Exemple



Contre-exemple



Il s'agit de méthodes d'implémentation. Elles doivent être private.

Principes de conception orientée objet

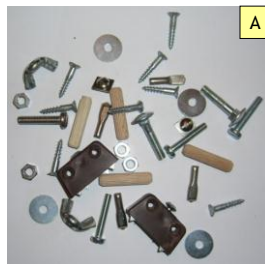
Modularité

- Principe
 - Regrouper dans des paquetages des ensembles d'abstractions.
 - Donne une vue d'ensemble et une cohérence à l'ensemble.
 - Le découpage doit être compréhensible.

Principes de conception orientée objet

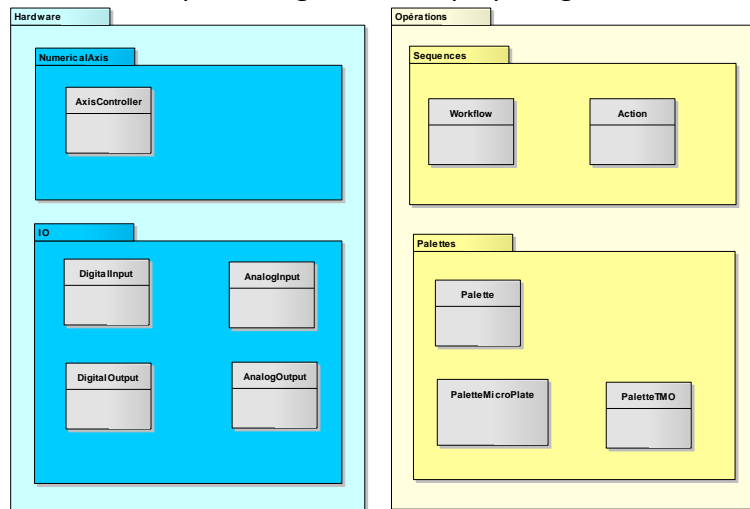
Modularité

Quelle est la
meilleure
organisation ?



Principes de conception orientée objet

Modularité – Exemple – Diagramme de paquetages



Modélisation d'applications industrielles avec UML

Principes de conception orientée objet

Hierarchie

- Définition
 - Une hiérarchie est un classement arborescent d'abstractions.
 - Selon une logique ou une cohérence compréhensible.
 - Permet d'appréhender beaucoup de concepts par la racine de l'arbre.
- Hiérarchies principales
 - Composition des objets
 - Fait partie de...
 - Classification des objets
 - Est un...

Modélisation d'applications industrielles avec UML

Modèle orienté objet

Notion de classe

- Classification
 - Approche naturelle pour organiser la connaissance.
 - Nos capacités cognitives maîtrisent très bien ce processus.
 - Offre une vision synthétique d'une collection d'objets.
 - Produit un découpage hiérarchique.
- Notion de classe
 - Ensemble d'objets qui partagent le même comportement.
 - Partagent donc le même code source !
 - On retrouve ainsi la notion de classe des langages OO.
- Notion de sous classe
 - Sous ensemble d'objet ayant toutes les caractéristiques de la classe parent.
 - Ont en commun des caractéristiques plus spécialisées.

Modèle orienté objet

Notion de classe



Modèle orienté objet
Notion de classe



Modèle orienté objet
Notion de classe



Modèle orienté objet

Notion de classe



Modèle orienté objet

Notion de classe

- La classification dépend du point de vue duquel on se place



Modèle orienté objet

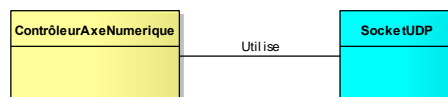
Notion d'objet

- Qu'est ce qu'un objet
 - Un élément tangible ou visible.
 - Quelque chose qui peut être appréhendé intellectuellement.
 - Un élément sur lequel on peut souhaiter agir.
- 3 Caractéristiques
 - Identité
 - C++: c'est l'adresse en mémoire de l'objet.
 - Etat
 - C++: c'est la réunion des valeurs des champs de l'objet.
 - Comportement
 - C++: c'est l'ensemble des méthodes de la classe de l'objet.

Relations entre classes

Lien simple : relation d'utilisation

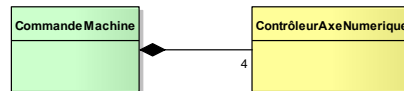
- Définition
 - Un objet utilise les services d'un autre.
 - Concrètement, il appelle ses méthodes.
- Test de validité
 - Un objet de ClasseA utilise un objet de ClasseB
- Exemple
 - L'objet « Contrôleur d'axe numérique » utilise l'objet « socket UDP » pour communiquer.



Relations entre classes

Composition

- Définition
 - Un objet est composé de sous-objets.
 - Concrètement, le sous objet fait partie des champs de son contenant.
- Test de validité
 - Le sous-objet a exactement la même durée de vie que le contenant.
 - Un objet de ClasseA *est composé* d'un objet de ClasseB.
- Exemple
 - Le « système de commande de la machine » *est composé* de 4 « contrôleurs d'axe numérique », de 20 « entrées TOR » et de 10 « sorties TOR ».



Modélisation d'applications industrielles avec UML

Relations entre classes

Agrégation

- Définition
 - Un objet peut contenir des sous-objets.
 - Concrètement, un objet contient des références vers des sous-objets.
- Test de validité
 - Le sous-objet peut avoir une durée de vie différente du contenant.
 - Un objet de ClasseA *peut contenir* un objet de ClasseB.
- Exemple
 - Un « workflow » contient des « actions ».

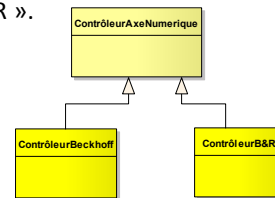


Modélisation d'applications industrielles avec UML

Relations entre classes

Héritage (généralisation / spécialisation)

- Définition
 - Une sous-classe est une spécialisation de sa classe parent.
 - Une classe est une généralisation de ses sous classes.
- Test de validité
 - Un objet de la sous classe est un objet de la classe.
- Exemples
 - Une « opération de fraisage » est une « opération d'usinage ».
 - Une « sortie TOR Beckhoff » est une « sortie TOR ».



Relations entre classes

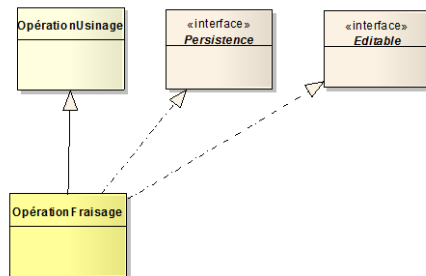
Héritage d'interface

- Une classe présente souvent plusieurs aspects orthogonaux
 - Une opération de fraisage est une opération d'usinage.
 - Une opération de fraisage est un objet persistant.
 - Une opération de fraisage est un objet éditable.
- L'héritage multiple d'implémentation
 - Est supporté en C++, mais fortement déconseillé.
 - N'est pas supporté par de nombreux autres langages.
- Solution : la notion d'interface
 - classe ne contenant que des méthodes virtuelles pures, sans implémentation.
 - Traduit conceptuellement « supporte le service de »
 - Les aspects non métiers sont usuellement traduits par l'héritage d'interface.

Relations entre classes

Héritage d'interface - illustration

- Seule la classe OpérationUsinage apporte une implémentation.
- Les interfaces Persistence et Editable n'apportent que des spécifications de méthodes.



Approche orientée objet

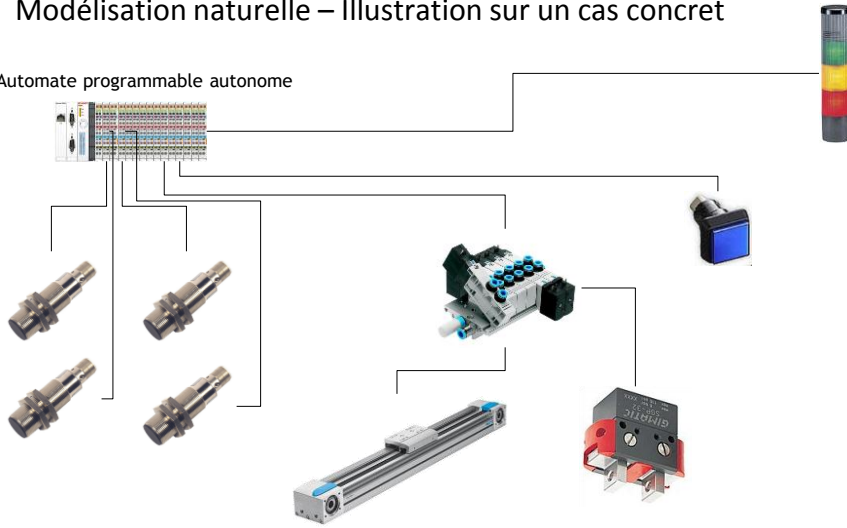
Modélisation naturelle

- Principe essentiel
 - Calquer le modèle objet sur la réalité.
- Justifications
 - Facilité
 - La structure du logiciel découle naturellement de la structure du monde réel.
 - Stabilité
 - La nature profonde de la réalité est assez stable.
 - Donc le logiciel sera également stable dans le temps.
 - Pas de refonte complète pour la prise en compte de nouvelles fonctionnalités
 - Compréhension
 - La structure du logiciel sera aisément compréhensible à partir de la connaissance du métier.

Approche orientée objet

Modélisation naturelle – Illustration sur un cas concret

Automate programmable autonome



Approche orientée objet

Modélisation naturelle

Approche orientée objet

Démarche intellectuelle – travailler sur la langue naturelle

- Reconnaître classes-objets et méthodes
 - Classes et objets : noms communs
 - Méthodes : verbes
- Avec l'interface homme-machine, l'opérateur édite une séquence en ajoutant des actions, pour lesquelles il doit saisir les paramètres d'usage.
- L'exécution d'une séquence d'actions consiste à verrouiller les portes de sécurité, à mettre sous tension les axes numériques, puis à exécuter les actions l'une après l'autre.

Approche orientée objet

Démarche intellectuelle – travailler sur la langue naturelle

- Reconnaître les relations entre classes
 - Les compositions
 - est composé de....
 - La machine est composée de 4 axes numériques et d'une pince.
 - Les agrégations
 - peut contenir des...
 - Une séquence peut contenir de nombreuses opérations.
 - Les généralisations :
 - Est un type de...
 - Le *fraisage* est un type d'opération réalisable sur ces machines.
 - Les spécialisations
 - Plusieurs types de ...
 - Il y a plusieurs types d'utilisateurs : *opérateur, technicien de maintenance*.

Comment procéder ?

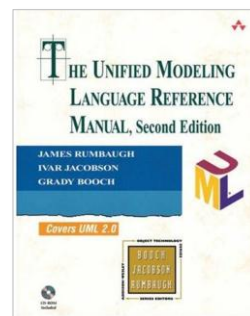
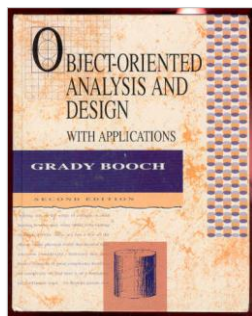
L'analyse et la conception

- Phase d'analyse : clarifier les objectifs
 - Clarifier exhaustivement les exigences à remplir.
 - Formaliser le comportement attendu du logiciel.
 - Le faire valider par le client.
- Phase de conception : clarifier la structure
 - Définir la structure du logiciel permettant de couvrir les exigences.
 - La valider par rapport aux exigences à remplir.
- Démarche générale
 - Abstraction : créer des vues simples à comprendre.
 - Décomposition hiérarchique vers plus de détail.
 - Utilisation d'un langage facile à comprendre par le client.

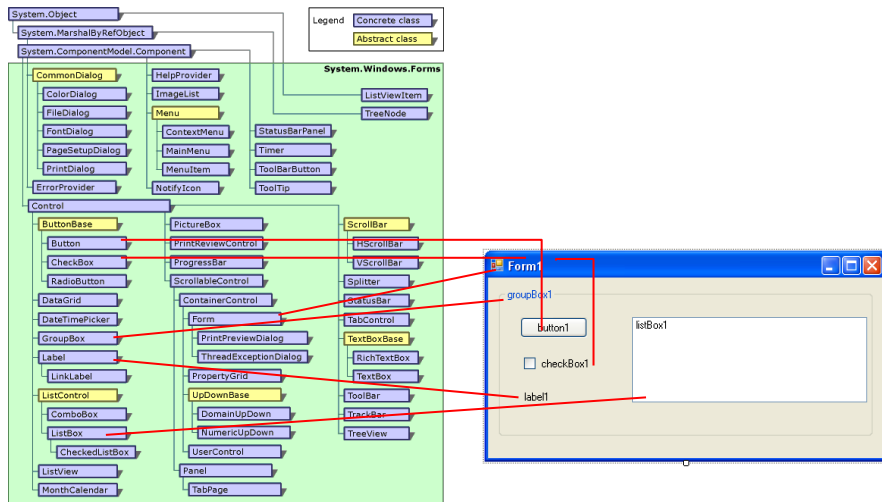
En savoir plus...

sur UML et la conception orientée objet

www.uml.org

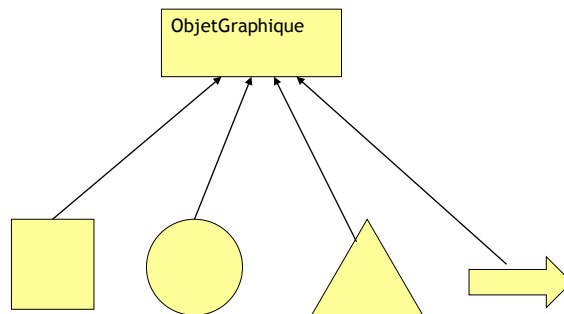


Application répandue de l'orienté objet Les Frameworks pour le développement d'interfaces graphiques



Modélisation d'applications industrielles avec UML

Application répandue Les logiciels graphiques vectoriels



Modélisation d'applications industrielles avec UML

heig-vd

MSE MASTER OF SCIENCE
IN ENGINEERING

Qu'avons-nous appris ?

Modélisation d'applications industrielles avec UML

heig-vd

MSE MASTER OF SCIENCE
IN ENGINEERING

Vos questions

Modélisation d'applications industrielles avec UML