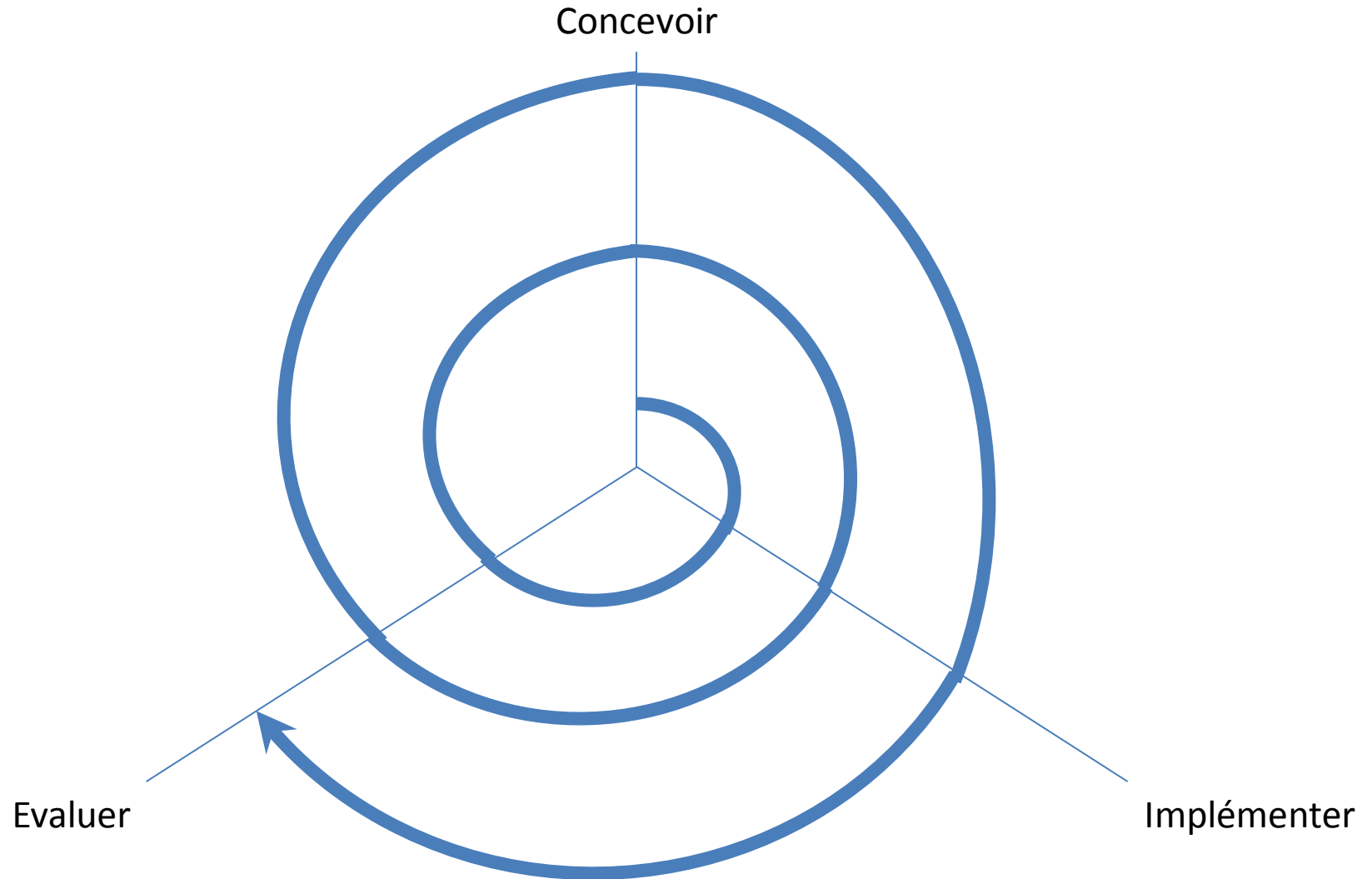


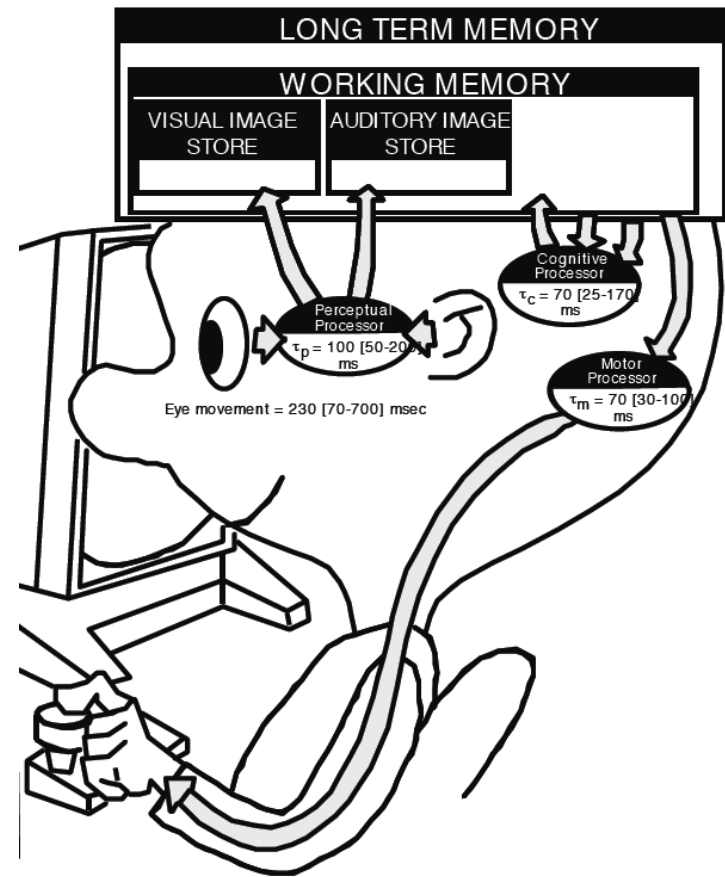
Résumé

Résumé : une méthodologie



Une façon de comprendre les utilisateurs

- Modèle simulé d'utilisateur
- Peut être utilisé pour évaluation (pas vu en cours)
- Dédution de « lois » empiriques sur les capacités des utilisateurs (ex : loi de Fitts)



Liste de critères d'utilisabilité

- Aide la conception en fournissant des réponses aux problèmes courants d'utilisabilité
 - Pas besoin de faire des expériences utilisateurs sur ce qui est évident
 - Peut être utilisé pour des évaluations heuristiques (par des experts)
- Homogénéité
 - Atteignabilité
 - Curabilité
 - Contrôle utilisateur
 - Multiplicité du rendu
 - Adaptabilité
 - Adaptativité
 - Plasticité
 - Migrabilité de tâche
 - Multimodalité
 - Observabilité
 - Réciprocité
 - Réflexivité
 - Insistance
 - Honnêteté
 - Honnêteté sociale
 - Prévisibilité
 - Guidage
 - Viscosité
 - Compatibilité
 - Flexibilité
 - Concision
 - Traitement des erreurs
 - Aide

Conception centrée utilisateur

- Une méthodologie
 - « science (logos) de la méthode »
- Identification de méthodes (analyse utilisateurs, des tâches, du domaine) et d'outils (personas, scénarios, croquis, storyboards)

Design patterns

- Une bonne solution à un problème courant
 - Réutilisable mais pas complète (pas une solution copiée-collée)
 - Lié à la notion de Best Practice
- Modèle de conception pour la conception de systèmes interactifs
- Librairie de modèles

Evaluations

- Evaluation formative
 - Méthode qualitative
 - Penser à haute voix
 - Observer les incidents critiques
 - Prototypes papier, simulation en magicien d'Oz, système réel
- Evaluation sur le terrain
- Expérience contrôlée
 - Méthode quantitative (chiffres et statistiques)
 - Hypothèse
 - Résultats scientifiques, prouvés

Et maintenant ?

- Tout ceci forme (une partie) des spécifications
- Implémenter le système réel
- Communiquer notre travail aux développeurs
 - Une image vaut mieux qu'un long discours
 - UML

Unified Modeling Language (UML)

Unified Modeling Language

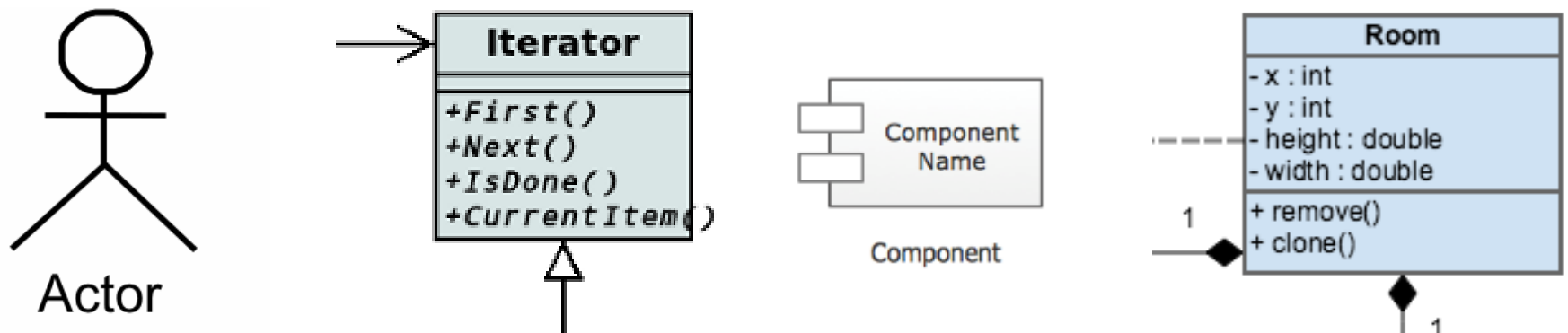
- UML est un langage, pas une méthodologie
 - Ca ne nous dit pas quoi faire, mais comment communiquer ce qu'on a fait
 - MERISE, au contraire, est une méthodologie
 - Ainsi que OMT (Rumbaugh), SSADM, LSDM, OOSA (Shlaer–Mellor), Booch, IE/IEM, Bachman, Coad–Yourdon, OOSE (Jacobson)...
- Créé par le Object Management Group et publié en 1997 (UML 1.0)
- Fréquemment critiqué
 - Exemple : article de 2005, “Is UML past its sell-by date?” http://www.theregister.co.uk/2005/02/04/tired_uml/

Unified Modeling Language

- Plusieurs diagrammes qui représentent (modélisent) tous les aspects d'un système informatique dans son environnement
- A destination des développeurs, mais aussi des décideurs (business users, MOA,...) ou mêmes des utilisateurs (en théorie)

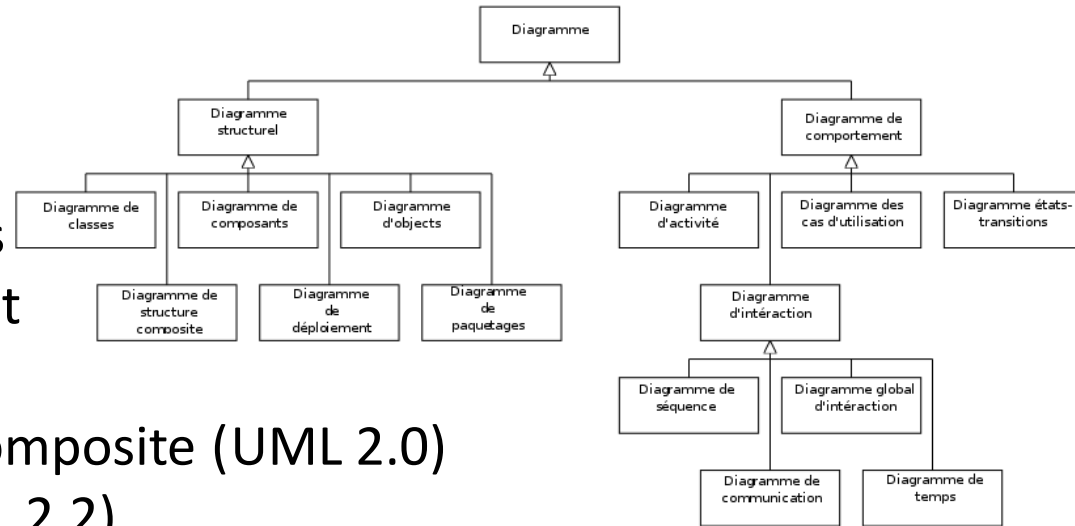
Unified Modeling Language

- Basé sur des modèles conceptuels
 - Un modèle (représentation) qui est fait de concepts (entités) et de leurs relations



Unified Modeling Language

- Diagramme de classes
- Diagramme d'objets
- Diagramme de composants
- Diagramme de déploiement
- Diagramme des packages
- Diagramme de structure composite (UML 2.0)
- Diagramme de profils (UML 2.2)
- Diagramme de cas d'utilisation
- Diagramme d'états-transitions
- Diagramme d'activité
- Diagramme de séquence
- Diagramme de communication (UML 2.0)
- Diagramme global d'interaction (UML 2.0)
- Diagramme de temps (UML 2.3)



Modélisation structurelle

- Représente les aspects statiques du système
 - Diagramme de classes
 - Le plus courant
 - Diagramme d'objets
 - Diagramme de composants
 - Diagramme de déploiement
 - Diagramme des packages
- Composants : tous les composants du systèmes sont inclus dans un de ces diagrammes
- Assemblage : plan pour assembler le système (comme des Légo)
- Ne décrit pas le comportement dynamique du système

Modélisation comportementale

- Représente les aspects dynamiques du système
 - Diagramme de cas d'utilisation
 - Diagramme d'états-transitions
 - Diagramme d'activité
 - Diagramme de séquence
- Fait références aux composants de la modélisation structurelle (pas de nouveaux composants)
- Décrit la circulation des informations et du contrôle dans le système

Modélisation architecturale

- Les aspects globaux de la modélisation structurelle et de la modélisation comportementale
- Concerne l'ensemble du système, vue de loin (« big picture »)
- Diagramme des packages
- Diagramme de cas d'utilisation

Diagramme de cas d'utilisation

- Décrit les fonctionnalités du système à sa bordure
- Interaction système – environnement (tout sauf le système)
 - Utilisateurs humains
 - Autres systèmes
- Chaque cas d'utilisation représente une unité **discrète** et **indivisible** d'interaction entre le système et l'environnement

Diagramme de cas d'utilisation

- Chaque cas d'utilisation comprend
 - Description générale (texte)
 - Exigences (*requirements*)
 - Contraintes
 - Pré-conditions
 - Post-conditions
 - Invariants
 - Scénarios (textes, au pluriel)
 - Une représentation dans le diagramme
 - Un ovale par cas d'utilisation, avec un nom

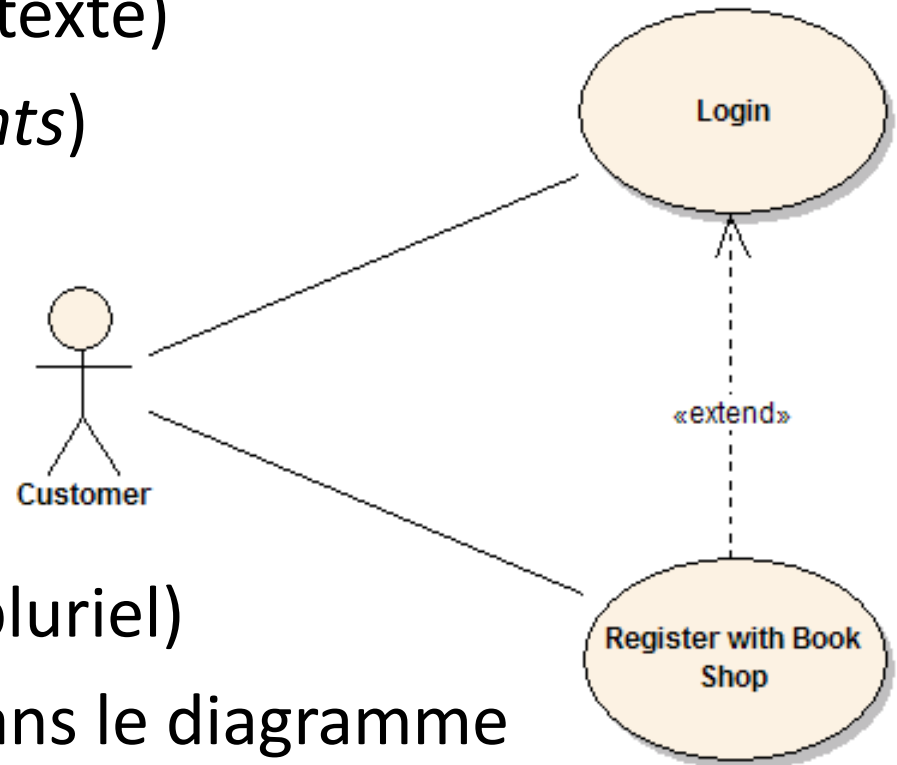


Diagramme de cas d'utilisation

- Acteur : représente l'élément extérieur qui utilise ou interagit avec le système
 - Utilisateur humain
 - Autre système
- L'acteur a un **but**
- Il utilise ou interagit avec le système pour effectuer une unité de **travail** qui le rapproche de ce but
- Un même acteur peut avoir accès à plusieurs cas d'utilisation
 - Les cas d'utilisation auxquels il a accès déterminent son rôle (utilisateur, administrateur, etc.)

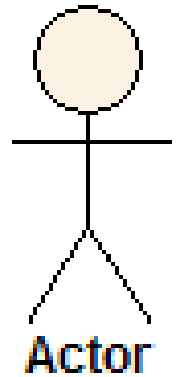


Diagramme de cas d'utilisation

- Include
 - Un cas d'utilisation peut inclure la fonctionnalité d'un autre cas d'utilisation
 - Fonctionnalité normale dans un cas normal
 - Chaque utilisation déclenche le/les cas inclus

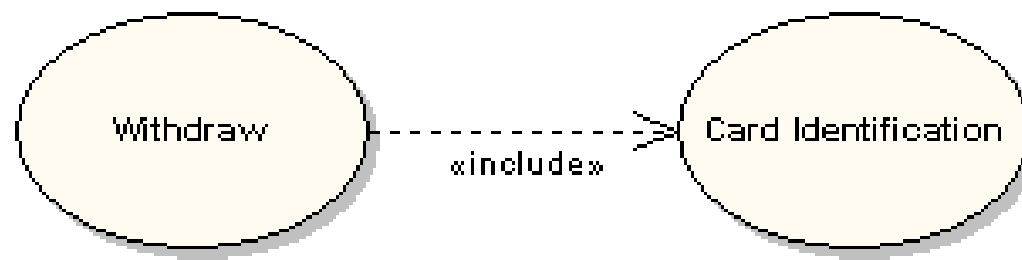


Diagramme de cas d'utilisation

- Extend
 - Un cas d'utilisation peut étendre le comportement un autre cas d'utilisation
 - Cas spécifique dans des circonstances exceptionnelles
 - Le cas de base peut être remplacé par un des cas qui l'étend
 - Un point d'extention peut être spécifié

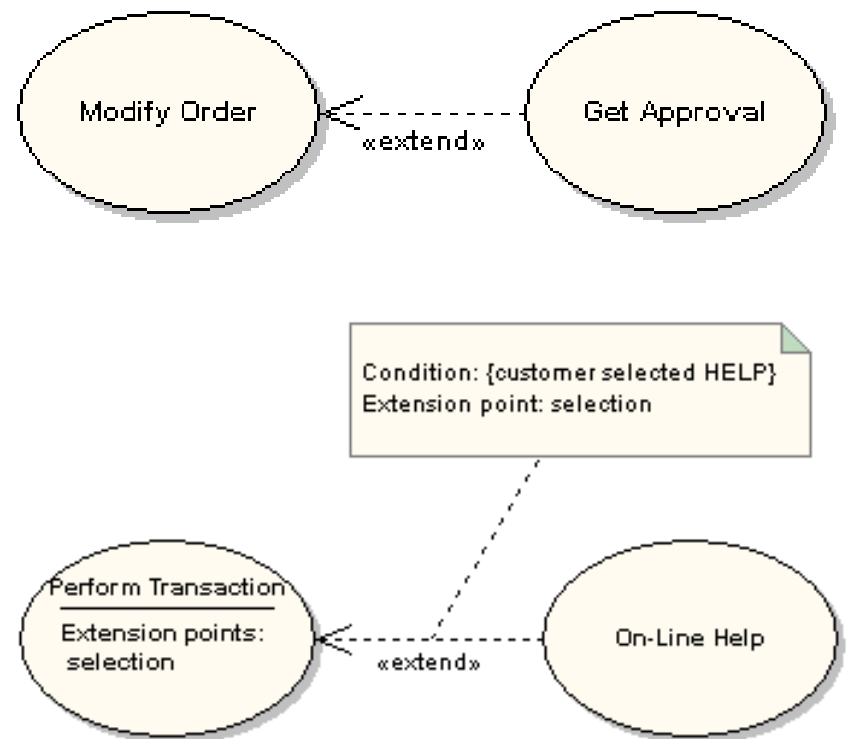


Diagramme de cas d'utilisation

- Les acteurs peuvent également généraliser un autre acteur
- Les connecteurs d'utilisation (entre un acteur et un cas d'utilisation) peuvent optionnellement avoir une multiplicité

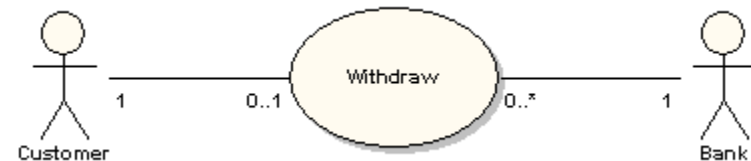


Diagramme de séquence

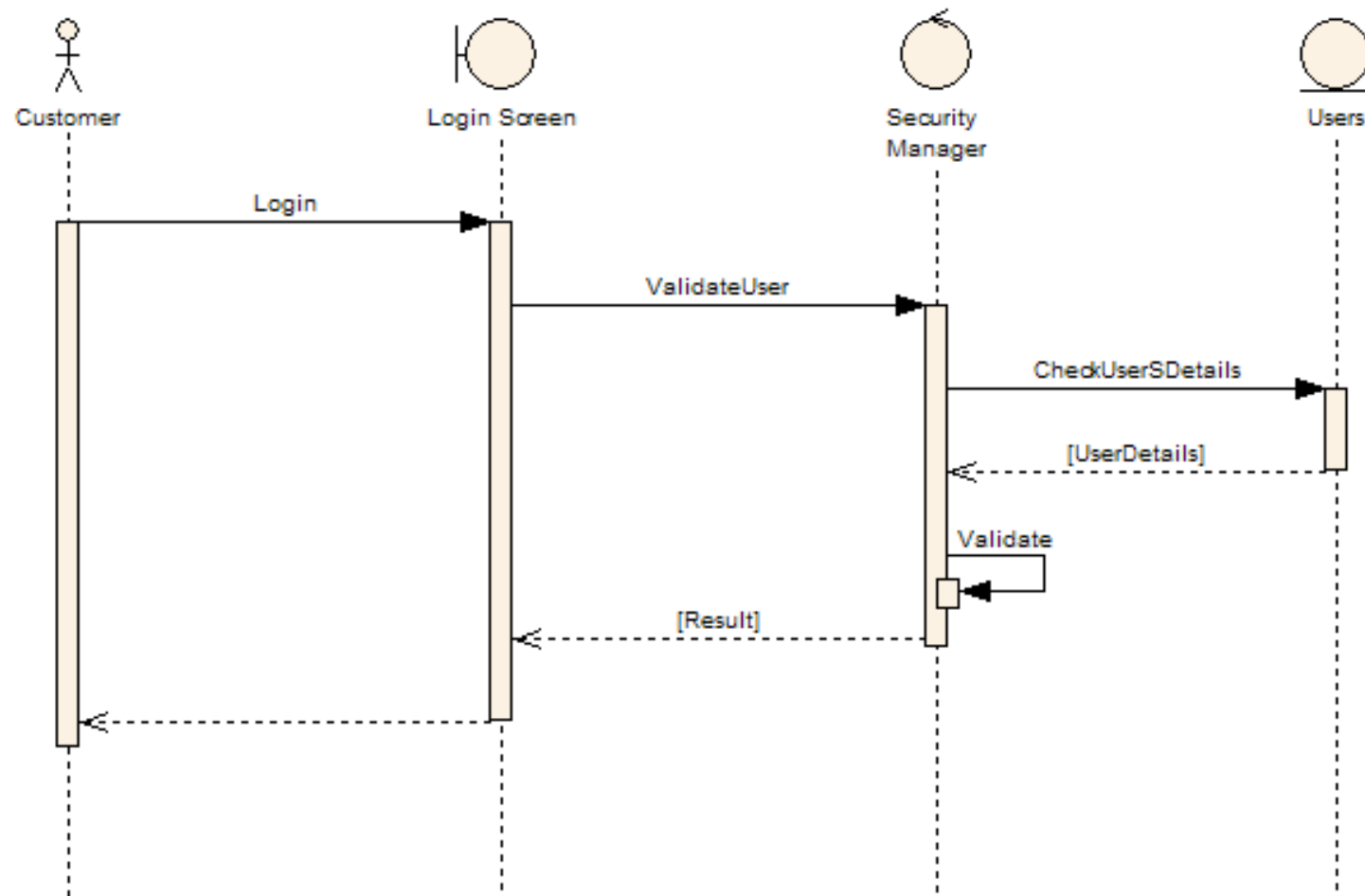


Diagramme de séquence

- Représentation graphique des interactions entre les objets du système, dans le temps
- Un diagramme de séquence pour scénario de chaque cas d'utilisation
- Les noms des objets (sauf le premier) correspondent à des noms de classes du diagramme de classes
 - Différentes icônes en fonctions du type
- Les noms des opérations correspondent à des méthodes de ces classes dans le diagramme de classes

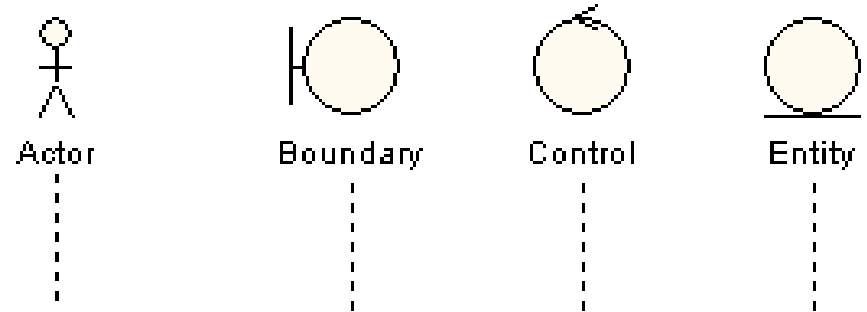


Diagramme de séquence

- Lignes de vie
 - Gestion de la création et de la destruction d'objets
- Gestion des algorithmes simples
 - Pas fait pour spécifier de la logique procédurale

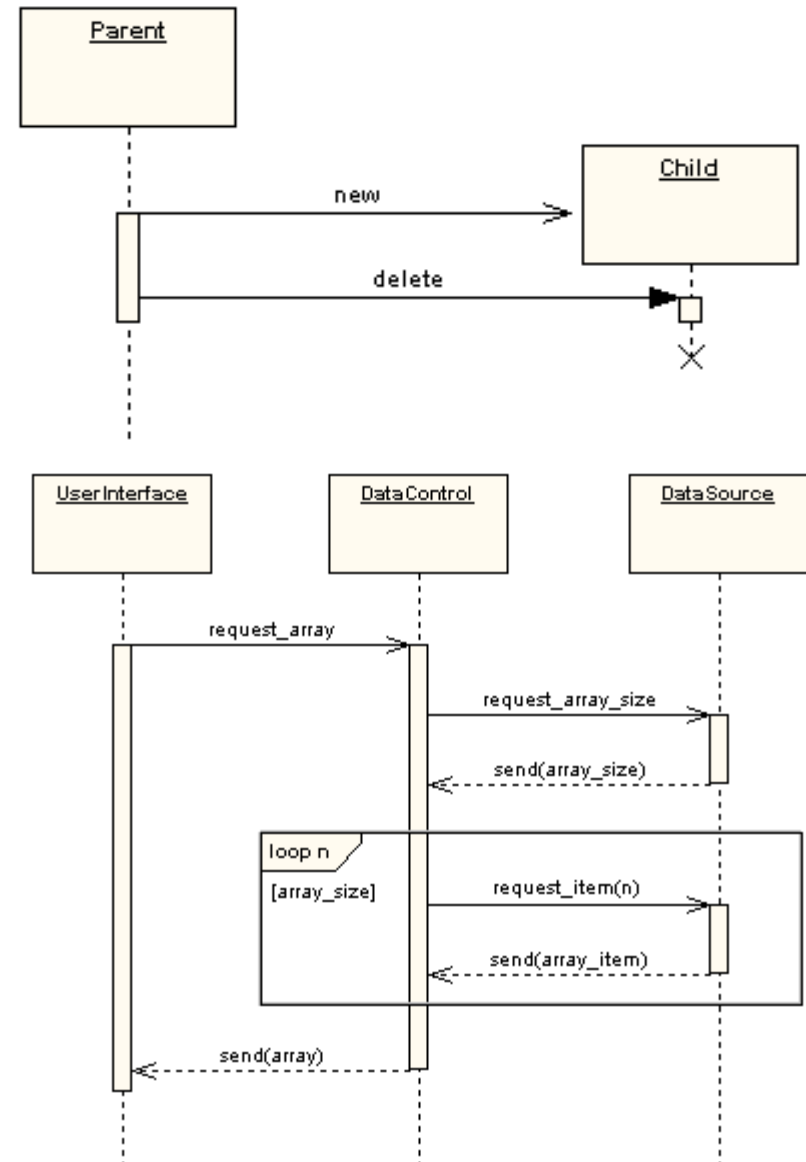


Diagramme d'activité

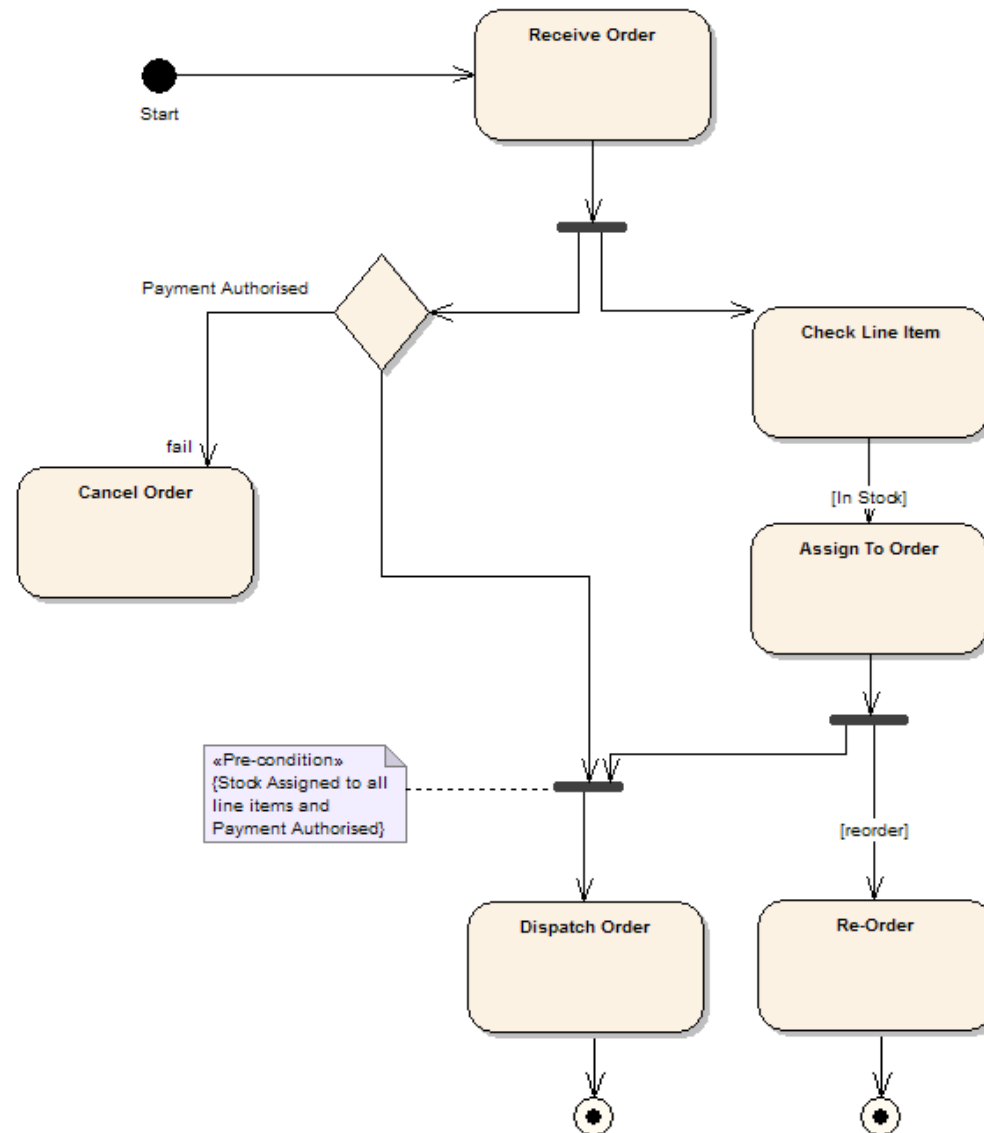

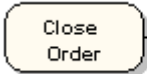
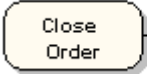
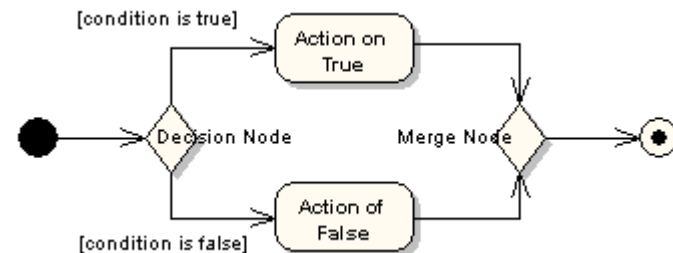


Diagramme d'activité

- Nœud initial ● → 
- Nœud final
 - De toute l'activité  → ●
 - D'une branche seulement  → ⊗

- Décision (if then else)



- Processus parallèles

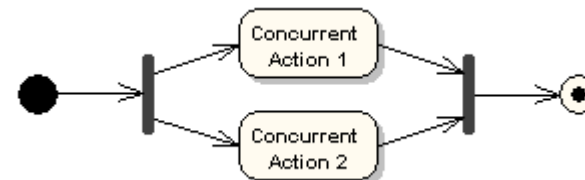
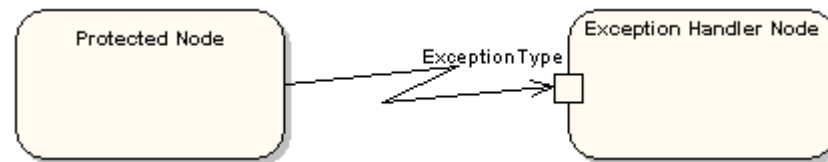
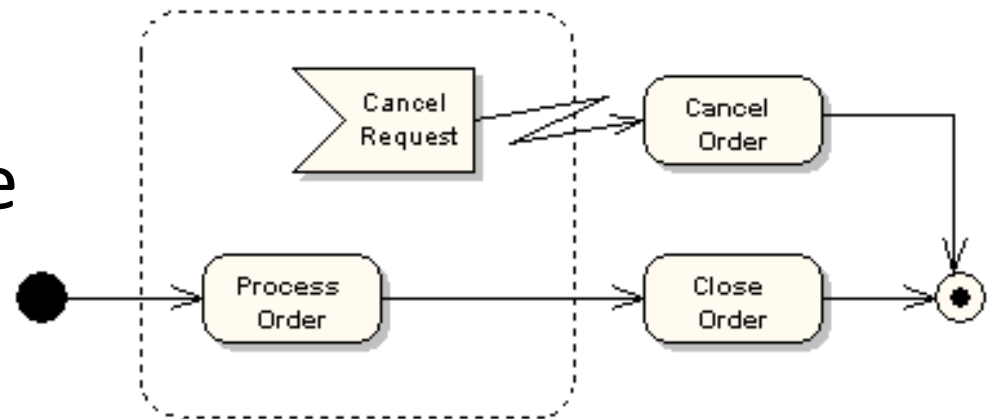


Diagramme d'activité

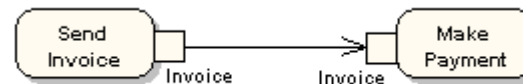
- Exceptions



- Activité interruptible



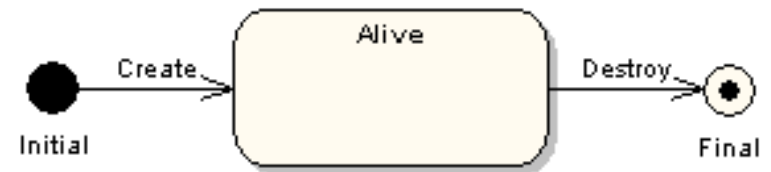
- Transfert d'objets



– Référence au diagramme des objets

Diagramme d'états-transitions

- Concerne un seul objet
- Etat de l'objet pendant sa ligne de vie (de sa création à sa destruction)



- Exemple : une porte

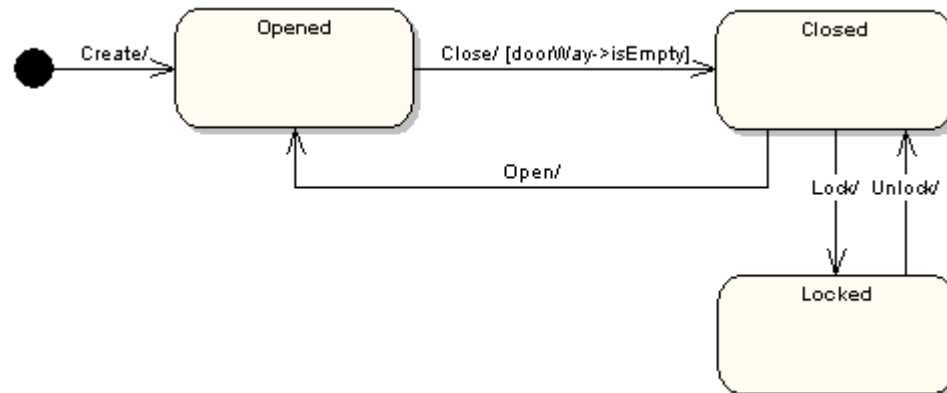


Diagramme d'états-transitions

- Chaque transition à un déclencheur (trigger)
 - Evènement externe qui peut faire changer l'objet d'état
 - Peut être un timer, pour tester une garde toutes les X millisecondes. Dans ce cas, des fois pas écrit explicitement
- Et optionnellement une garde (guard)
 - Condition booléenne qui restreint le trigger
 - Si la garde est fausse, le trigger est ignoré
- Optionnellement des effets
 - Actions qui seront exécutées sur **cet** objet

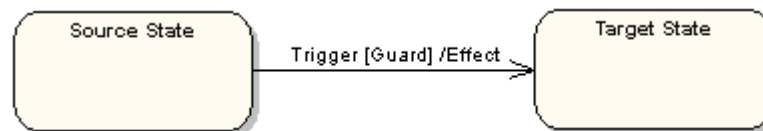
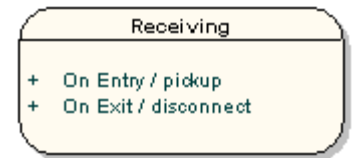


Diagramme d'états-transitions

- Un état à un nom et optionnellement des actions



- On Entry : équivalent à mettre cette action sur toutes les transitions qui vont vers cet état
- On Exit : équivalent à mettre cette action sur toutes les transitions qui partent de cet état
- Non officiel : action Loop exécutée en permanence tant que l'objet est dans cet état
 - Equivalent à une auto-transition avec un timer et une action

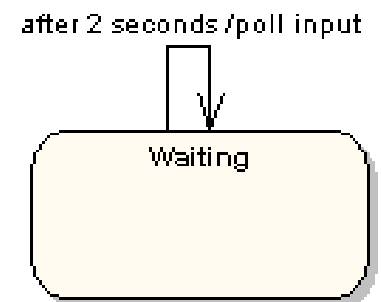


Diagramme d'états-transitions

- Différence états et transitions
- Une transition est instantanée
 - Elle n'a pas de durée
 - On ne peut pas rester dans une transition
- Un état
 - Toujours dans un seul état à la fois : l'état actif
 - Caractéristique de l'objet à un instant t : on observe l'objet, il est dans un état

Diagramme d'états-transitions

- Etats composés
- Un sous diagramme d'états-transitions, concernant le **même objet**
- Les transitions qui arrivent sur le sous diagramme sont équivalentes à arriver sur à son point initial
- Les transitions qui partent du sous diagramme sont équivalentes à être dupliquées sur tous les sous-états

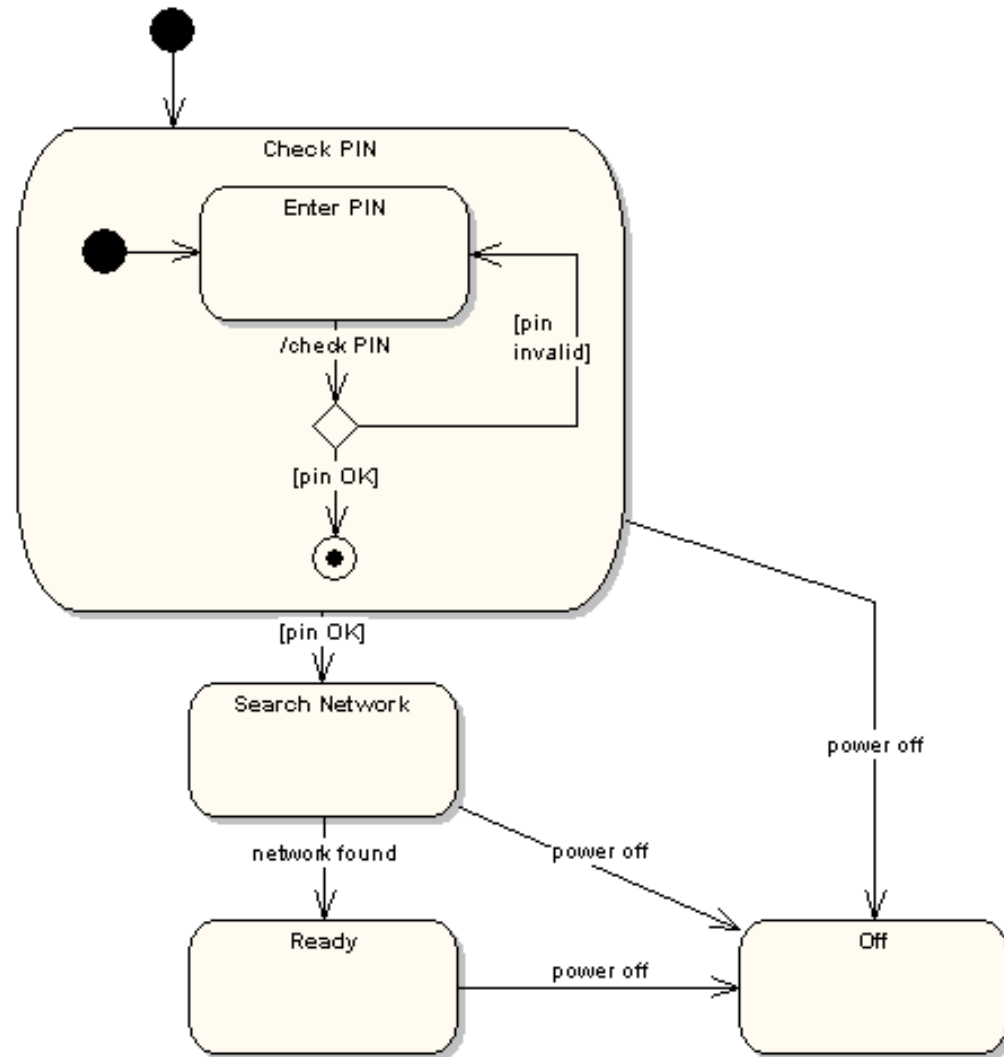


Diagramme d'états-transitions

- Etats composés : possibilité d'entrées multiples (nommées) et de sorties multiples (nommées)

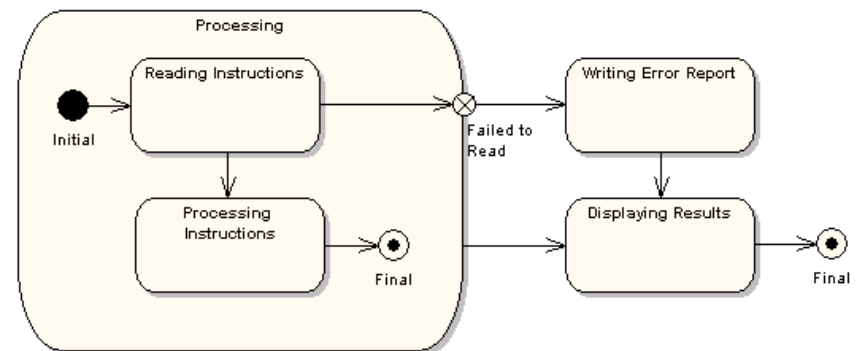
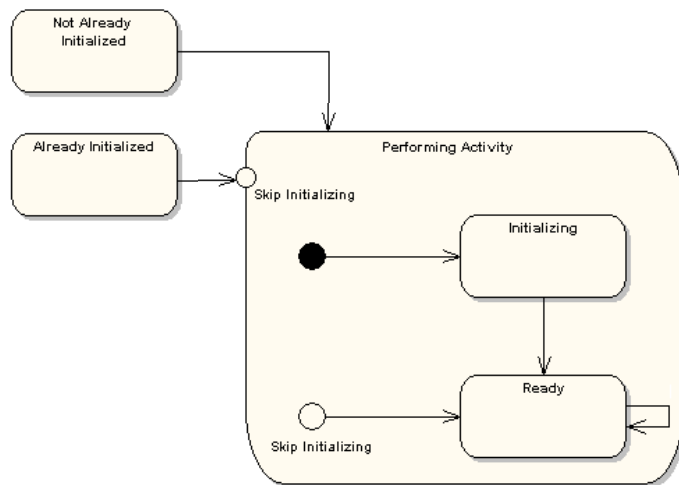


Diagramme d'états-transitions

- Pseudo-état de choix
 - L'état suivant dépend de l'état précédant (pas de l'état actuel)
 - Attention, c'est quand même un état (durée)
- Régions concurrentes
 - L'objet se divise en deux (e.g. threads) et chaque partie à son propre état

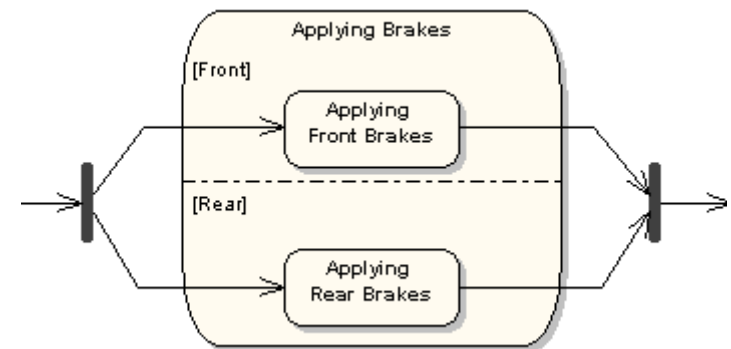
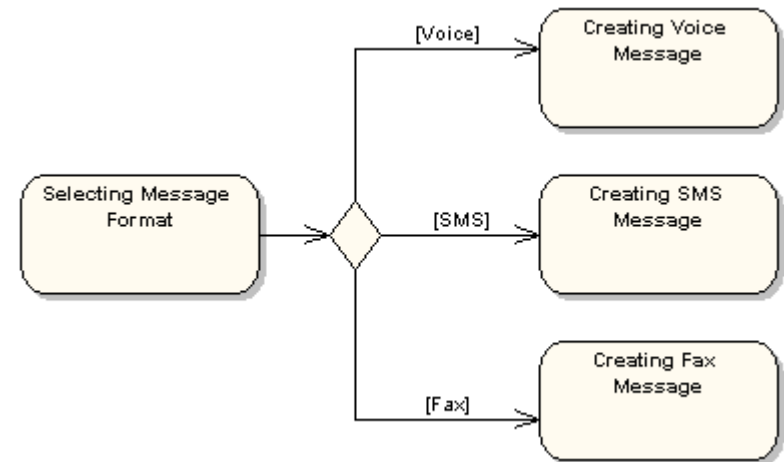


Diagramme de classes et diagramme d'objets

- Le diagramme d'objets est un exemple du diagramme de classes sur un cas spécifique
 - Nom_Objet :Nom_Classe
 - Les champs ont des valeurs
 - Multiplicités apparentes avec plusieurs objets de la même classe

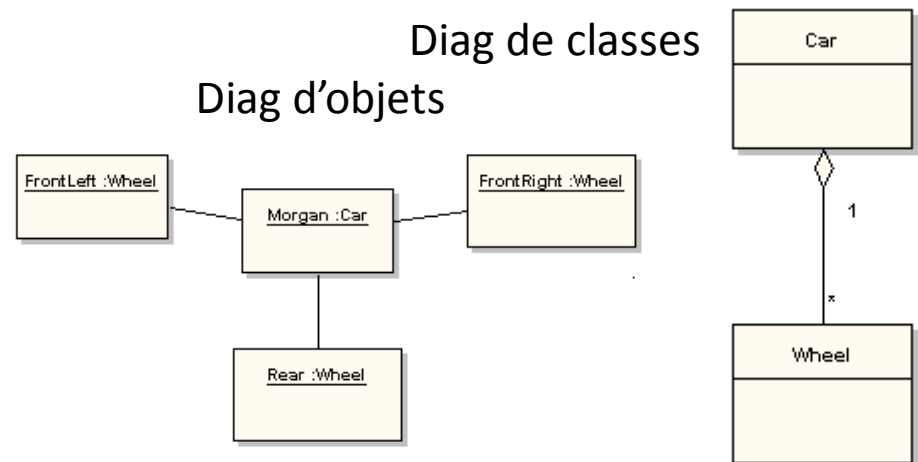
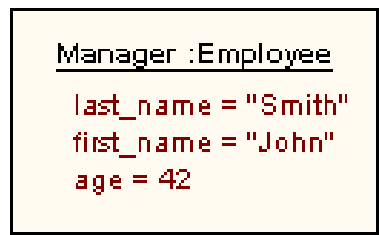
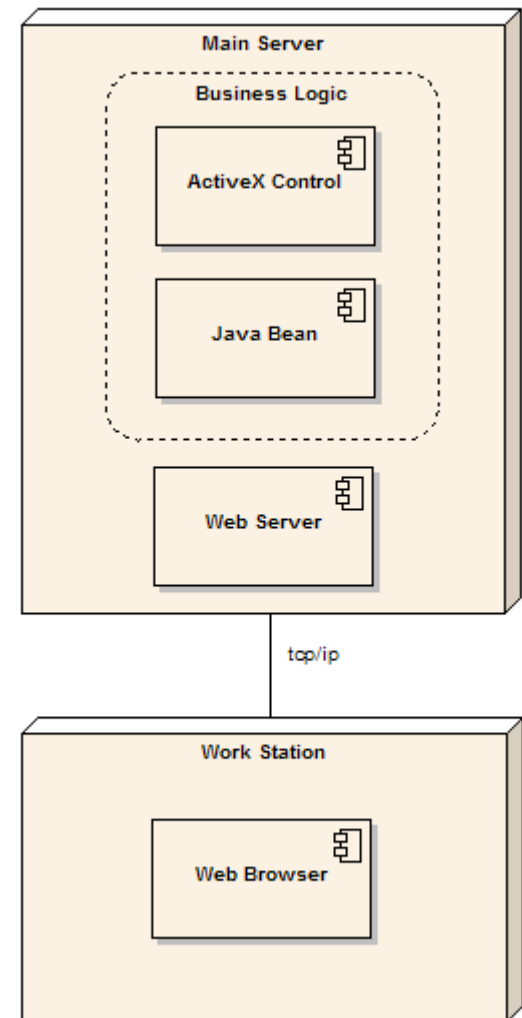
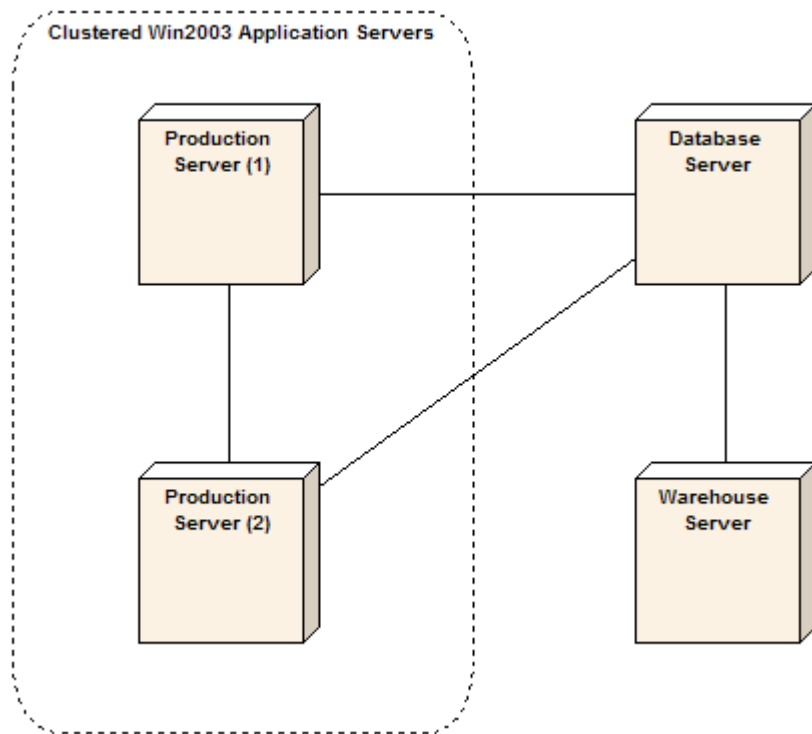


Diagramme de déploiement

- Description physique du système
 - Serveurs, capacités réseau, etc.
 - Plus ou moins détaillé



Développeur n'est
peut-être pas la
seule voie pour
vous...

(Question subsidiaire : ces
données statistiques sont-elles
représentatives de l'ensemble du
marché de l'emploi français ?)



The Hottest Skills of 2015 on LinkedIn France

1	Cloud and Distributed Computing	↑	0	14	Database Management and Software	↑	NR
2	Statistical Analysis and Data Mining	↓	-1	15	Microsoft Application Development	↑	NR
3	SEO/SEM Marketing	↑	1	16	Software Revision Control Systems	↓	-1
4	Corporate Law and Governance	↑	5	17	Social Media Marketing	↓	-4
5	Software QA and User Testing	↑	19	18	Economics	↓	-4
6	Mobile Development	↑	10	19	Public Policy and International Relations	↓	-7
7	User Interface Design	↑	NR	20	Recruiting	↓	-3
8	Compliance and Employment Law	↑	10	21	Foreign Language Translation	↑	2
9	Data Engineering and Data Warehousing	↑	NR	22	Web Architecture & Development Frameworks	↑	NR
10	Retail Store Operations	↑	NR	23	Restaurants and Catering	↑	NR
11	Mechanical and Aerospace Engineering	↓	-3	24	Computer Graphics and Animation	↓	-14
12	IBM Mainframes and Systems	↓	-7	25	Data Presentation	↑	NR
13	Perl/Python/Ruby	↑	8				

* NR (Not recorded in 2014)



The Hottest Skills of 2015 on LinkedIn China

1

Network and Information Security

2

Statistical Analysis and Data Mining

3

Cloud and Distributed Computing

4

Virtualization

5

Public Policy and International Relations

6

Mac, Linux and Unix Systems

7

Corporate Law and Governance

8

Storage Systems and Management

9

Retail Store Operations

10

Channel Marketing

11

Web Architecture and Development Frameworks

12

Business Intelligence

13

Middleware and Integration Software

14

Data Engineering and Data Warehousing

15

Social Media Marketing

16

PR and Communications

17

Digital and Online Marketing

18

Economics

19

Database Management and Software

20

Writing and Publishing

21

Microsoft Application Development

22

User Interface Design

23

TV and Video Production

24

Brand Strategy and Management

25

Computer Graphics and Animation

Autres voies

- Consultant en utilisabilité
 - Sociétés spécialisées dans l'utilisabilité
 - SSII informatique
- Spécialiste utilisabilité dans une grande entreprise
- **Double casquette dans une PME/start-up**
- Si vous avez aimé plus spécialement la partie expé contrôlée/stats, faites un M2R et une thèse de doctorat en IHM !

Examen

- Semaine du 21 mars ou du 28 mars
 - Vérifier l’affichage au secrétariat et/ou sur le Moodle
 - Vérifier la salle au secrétariat et/ou sur le Moodle
 - Vérifier le jour et l’heure au secrétariat et/ou sur le Moodle
- Penser à amener : papier (plusieurs feuilles), stylo, typex, crayon à papier, gomme, règle
 - **Aucun échange de matériel avec ses voisins pendant l’examen**
- Feuille de note autorisée pendant l’examen

Feuille de note autorisée pendant l'examen

- Une feuille A4 (taille normale)
- Recto-verso
- Manuscrite (pas à l'ordinateur)
- Au stylo (pas au crayon à papier)
- Non photocopiée
- Lisible sans instrument (loupe, microscope...)
- Non partageable avec ses voisins pendant l'examen (mais plusieurs personnes peuvent recopier la même **avant** l'examen (et à la main))