

Cycle de vie de logiciel

CMMI, vérification, test



Références

- ▶ CMMI <http://www.sei.cmu.edu/cmmi/>
- ▶ IEEE (1998). *IEEE standard for software test documentation*. New York: IEEE. ISBN 0-7381-1443-X.
- ▶ Software Testing by Jiantao Pan, Carnegie Mellon University
- ▶ Model based testing with **Labelled Transition Systems**, J. Tretmans, Nijmegen, University



plan

- ▶ CMMI
- ▶ Cycle de vie
- ▶ Vérification
- ▶ Tests

3

Capability Maturity Model Integration (CMMI)

- ▶ Wikipedia : Le **Capability Maturity Model Integration** est une approche interdisciplinaire d'[ingénierie des systèmes](#) couvrant les compétences et processus techniques et managériaux.
- ▶ C'est un modèle de référence qui permet d'évaluer le niveau d'organisation globale d'une entreprise [IT](#) relativement à une métrique d'utilisation de bonnes pratiques, en [génie logiciel](#) et [management](#).
- ▶ Origine des US

4

Capability Maturity Model Integration (CMMI)

- ▶ Entreprises certifiées:
 - Demande de certification, effectuée par un organisme agréé CMMI

- ▶ CMMi modèle pas une norme ?
 - Souvent appelé norme mais c'est un modèle
 - Norme ISO 9001 = norme qualité au sens général
 - CMMI est un modèle satisfaisant la norme ISO 9001

5

Capability Maturity Model Integration (CMMI)

- ▶ Pourquoi faire ?
 - Le développement "marchait" bien avant ? Pas tant que ça
 - => dvp, debuggage, dvp, debuggage, envoi au client => pb de fonctionnement, suivi de bug
 - Pour les systèmes critiques cette approche est des plus mauvaises ! => validation formelle depuis les années 70 ! Mais pas spécialement dans les entreprises
 - Aujourd'hui, les entreprises s'intéressent de plus en plus à la qualité logicielle (ATOS-ORIGIN (CMMI3), impôts, boeing, intel, ...
 - CMMI mais aussi d'autres modèles : ITIL Cobit,...

6

Capability Maturity Model Integration (CMMI)

- ▶ Normes qualité logicielle difficile à mettre en place
 - difficile de changer les habitudes
 - Lourd et coûteux (test de logiciel/système > 40% du coût pour les syst. Critiques)
 - Vers des nouveaux métiers informatiques ! Qualité logicielle

- ▶ Mais certaines entreprises d'Inde sont CMMI 5

7

Capability Maturity Model Integration (CMMI)

- ▶ CMMI proposé par le SEI (software engineering institute), version 1.2 aujourd'hui

- ▶ Plusieurs modèles CMMI
 - acquisition
 - Développement, doc de 573 pages

- ▶ CMMI est donc avant tout un référentiel d'évaluation
 - de la capacité à gérer et terminer un projet correctement,
 - proposant nombre de bonnes pratiques liées à la gestion, au développement et à la maintenance d'applications et de systèmes.

8

Capability Maturity Model Integration (CMMI)

- ▶ Ces bonnes pratiques sont regroupées en 24 processus, eux-mêmes regroupés en 4 types :
 - *Process Management*
 - (définir, mesurer, contrôler les processus) (méthode de fonctionnement dans l'entreprise)
 - *Project Management*
 - *Engineering*
 - *Support*
- et 5 niveaux de maturité.

9

Capability Maturity Model Integration (CMMI)

- ▶ 5 niveaux de maturation:
- ▶ un projet est supposé monter de niveau au fur et à mesure du temps. Plus le niveau est grand, moins il y a de risques et meilleure est la performance du projet.
- ▶ Pour instaurer dans les organisations et les projets l'importance de disposer de spécifications clairement établies et définies, même si elles peuvent être révisées en cours de route.
- ▶ CMMI a également établi l'intérêt de la relecture par un pair (*peer review*), qui est aujourd'hui l'un des points fondamentaux de l'Extreme Programming, ainsi que du contrôle de version, également très répandu aujourd'hui.

10

Capability Maturity Model Integration (CMMI)



11

Capability Maturity Model Integration (CMMI)

- ▶ Niveaux de maturité:
- ▶ **Niveau 1:**
- ▶ Le niveau le plus bas montre que l'organisation n'est pas prête, et le projet pas stable.
- ▶ Ce dernier dépend d'une poignée de personnes, qui ne font pas appel à des processus éprouvés.
- ▶ Il se peut cependant que le projet aboutisse, mais en dépassant certainement le budget et le temps alloués. Le projet ne construit pas sur les succès passés.

12

Capability Maturity Model Integration (CMMI)

- ▶ Niveaux de maturité:
- ▶ **Niveau 2:**
- ▶ Le projet construit sur ce qui a été appris précédemment, en faisant appel à une certaine discipline et à une gestion de projet basique.
- ▶ De fait, le projet est géré selon les plans, avec étapes-clés et vérification des coûts et des fonctionnalités.

13

Capability Maturity Model Integration (CMMI)

- ▶ Niveaux de maturité:
- ▶ **Niveau 3:**
- ▶ Ce n'est plus le projet qui dispose d'une bonne discipline, mais l'ensemble de l'organisation, de manière cohérente. Tous les projets s'en trouvent améliorés.
- ▶ Vérification ,test, gestion de risque,...

14

Capability Maturity Model Integration (CMMI)

- ▶ Niveaux de maturité:
- ▶ **Niveau 4:**
- ▶ Les efforts de mesure et de gestion autorisent un contrôle sans effort du développement, avec capacité d'ajuster et adapter des projets précis sans troubler les autres. Les performances des processus sont prévisibles en quantité comme en qualité.



15

Capability Maturity Model Integration (CMMI)

- ▶ Niveaux de maturité:
- ▶ **Niveau 5:**
- ▶ Les processus sont constamment améliorés de manière incrémentale et innovante. Les objectifs sont revus en permanence pour rester proches des besoins du marché. Les évolutions sont anticipées et gérées de bout en bout.



16

Capability Maturity Model Integration (CMMI)

Name	Abbr	ML	CL1	CL2	CL3	CL4	CL5
Requirements Management	REQM	2	Target Profile 2				
Project Planning	PP	2					
Project Monitoring and Control	PMC	2					
Supplier Agreement Management	SAM	2					
Measurement and Analysis	MA	2					
Process and Product Quality Assurance	PPQA	2					
Configuration Management	CM	2					
Requirements Development	RD	3	Target Profile 3				
Technical Solution	TS	3					
Product Integration	PI	3					
Verification	VER	3					
Validation	VAL	3					
Organizational Process Focus	OPF	3					
Organizational Process Definition +IPPD	OPD +IPPD	3					
Organizational Training	OT	3					
Integrated Project Management +IPPD	IPM +IPPD	3					
Risk Management	RSKM	3					
Decision Analysis and Resolution	DAR	3					
Organizational Process Performance	OPP	4	Target Profile 4				
Quantitative Project Management	QPM	4					
Organizational Innovation and Deployment	OID	5	Target Profile 5				
Causal Analysis and Resolution	CAR	5					

Figure 3.5: Target Profiles and Equivalent Staging

17

Capability Maturity Model Integration (CMMI)

- ▶ Dans CMMI, on trouve:
 - ▶ Des gestions de planning, de projets, le monitoring et le contrôle de projet, la gestion des risques, la gestion du client,
 - ▶ le cycle de vie du logiciel,....
 - ▶ Voyons pour la partie gestion de projet:

18

Capability Maturity Model Integration (CMMI)

- ▶ Cf CMMI doc:
- ▶ The Basic Project Management process areas address the activities related to establishing and maintaining the project plan, establishing and maintaining commitments, monitoring progress against the plan, taking corrective action, and managing supplier agreements

19

Capability Maturity Model Integration (CMMI)

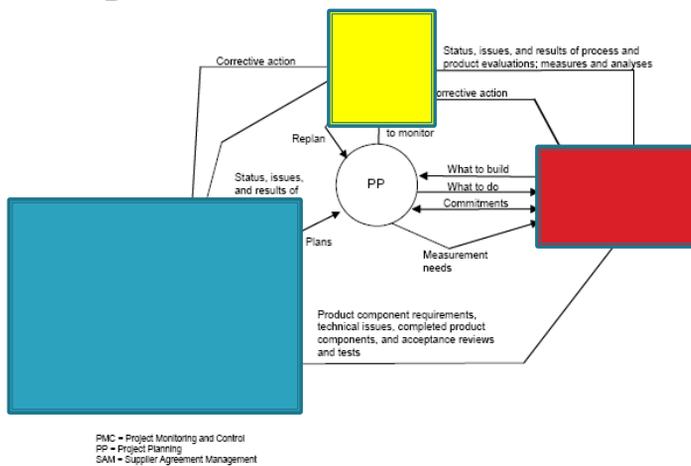


Figure 4.3: Basic Project Management Process Areas

20

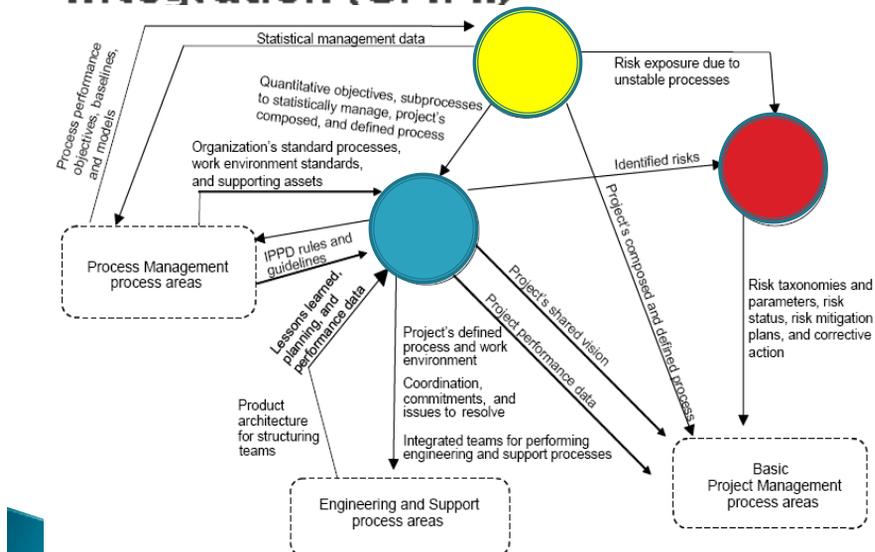
Capability Maturity Model Integration (CMMI)

- CF CMMI doc:

The Advanced Project Management process areas address activities such as establishing a defined process that is tailored from the organization's set of standard processes, establishing the project work environment from the organization's work environment standards, coordinating and collaborating with relevant stakeholders, managing risk, forming and sustaining integrated teams for the conduct of projects, and quantitatively managing the project's defined process

21

Capability Maturity Model Integration (CMMI)



22

Capability Maturity Model Integration (CMMI)

IPM+IPPD = Integrated Project Management (with the IPPD addition)

IPPD (integrated product and process development)=>

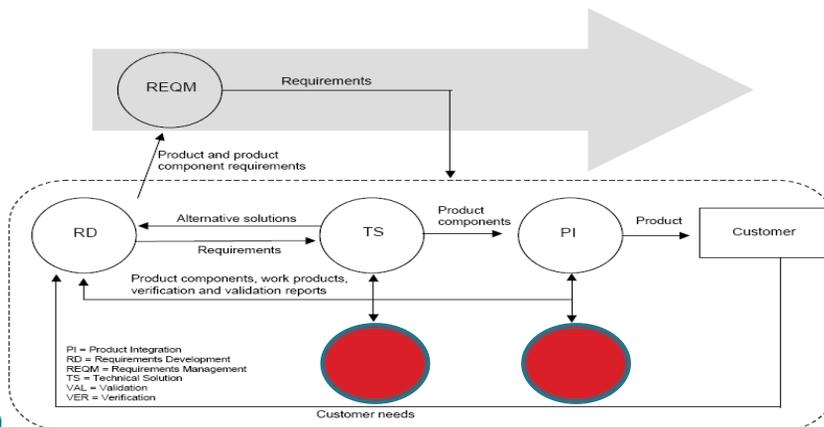
QPM = Quantitative Project Management

RSKM = Risk Management

23

Capability Maturity Model Integration (CMMI)

Engineering process area



24

Dossier d'affaire



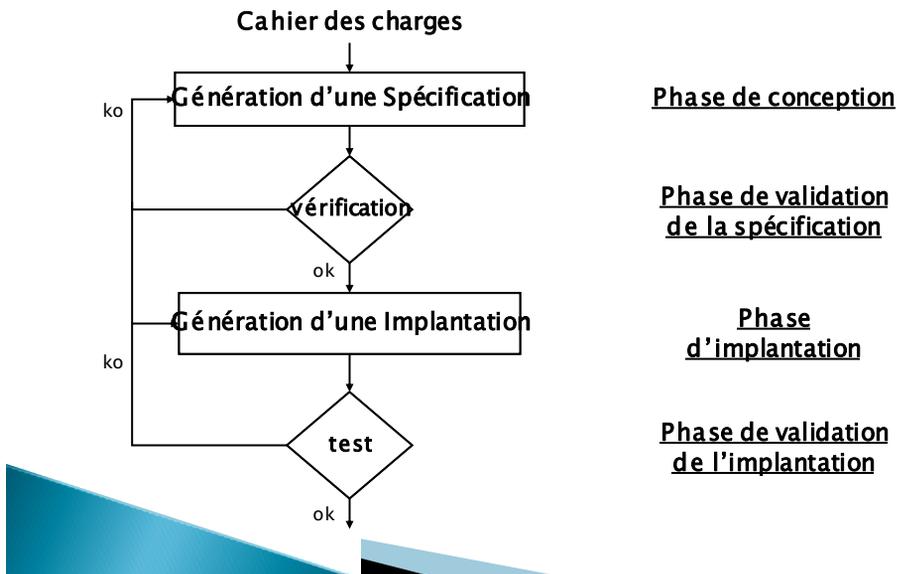
- PQP : plan qualité projet = référentiel (outils, ...)
- PdT : plan de test
- Plan d'avancement projet (remplis par chef de projet => indicateurs)

25

Cycle de vie, specification, test

Cahier des charges

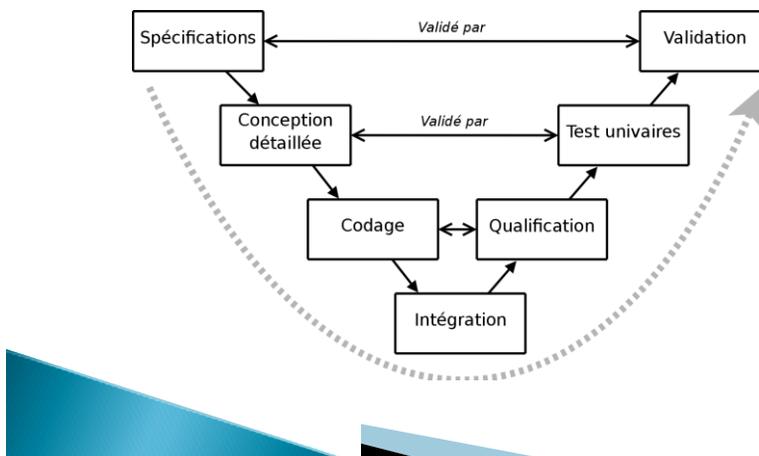
... engineering process area peut s'écrire aussi



27

...Ou encore

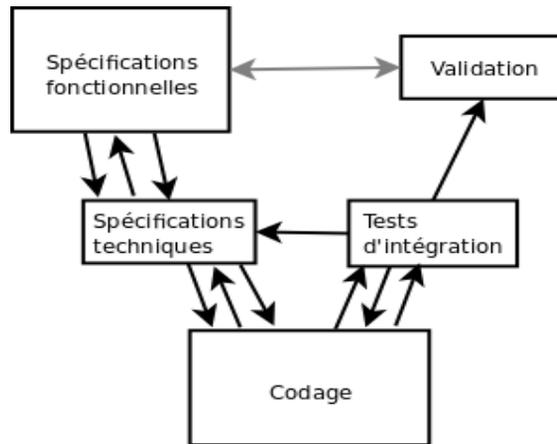
Cycle en V



28

...Ou encore

Cycle en V pas assez souple, la réalité ?



29

...Ou encore

Méthodes agiles

correspondance continue entre les dev. et le client

Ex de méthodes:

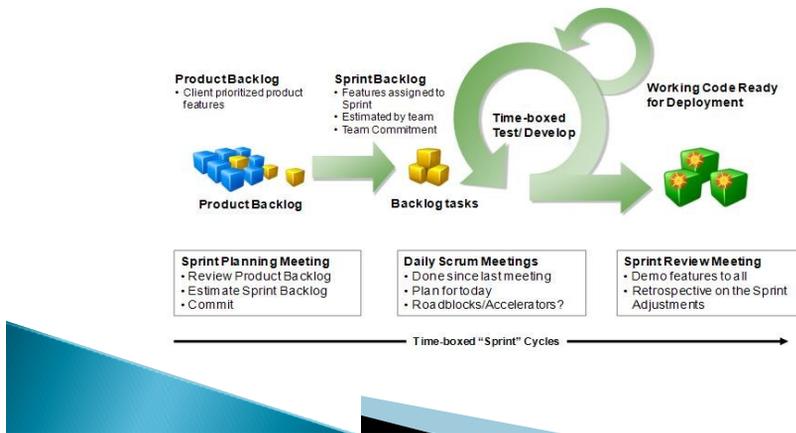
- XP (extreme programming) : cycles rapides de développement , recherche de scenarios, découpage en taches, attribution des taches, phases de tests et on recommence tant que le client a des scénarios à demander

30

...Ou encore

Méthodes agiles

SCRUM :



31

Cahier des charges

- ▶ Un « **cahier des charges** » est un document contractuel décrivant ce qui est attendu du **maître d'œuvre** par le **maître d'ouvrage**.
- ▶ Il s'agit donc d'un document décrivant de la façon la plus précise possible, avec un vocabulaire simple, les besoins auxquels le **maître d'œuvre** doit répondre.
- ▶ Pas de technique
- ▶ Elements pour juger de la taille du projet et sa complexité
- ▶ Contenu peut être modifié sur la base d'un accord
- ▶ Doit aussi contenir un planning

32

Cahier des charges

▶ A) Introduction

▶ 1 – Contexte

Décrire brièvement l'environnement dans lequel s'inscrit le projet (stratégie, enjeux, domaine, etc.)

▶ 2 – Historique

Il est possible de donner un bref historique du contexte dans lequel s'inscrit le projet si cela est pertinent. Gardez à l'esprit que ceci n'est qu'un exemple de cahier de charge. Vous ne devez pas tout appliquer à la lettre.

33

Cahier des charges

B) Description de la demande

▶ 1 – Les objectifs

Définir les résultats que le projet doit atteindre.

Méthode : Un énoncé d'objectif doit comporter un verbe d'action à l'infinitif et un objet.

Exemple : Diffuser un corpus de connaissance assimilable par toute personne de niveau Bac + 4.

▶ 2 – Produit du projet

Proposer une description générale de ce produit.

▶ 3 – Les fonctions du produit

Lister et justifier les principales fonctionnalités du produit.

▶ 4 – Critères d'acceptabilité et de réception

Formuler des indicateurs précis qui permettent de mesurer si les objectifs de performance du produit sont atteints.

Exemple : Le produit doit répondre à la norme XX0.

34

Cahier des charges

C) Contraintes

- ▶ **1 – Contraintes de coûts**
Spécifier le budget alloué au projet.
- ▶ **2 – Contrainte de délais**
Spécifier la date de livraison du produit et les éventuelles échéances intermédiaires.
- ▶ **3 – Autres contraintes**
Spécifier les éventuelles autres contraintes à prendre en compte dans le cadre du projet (normes techniques, clauses juridiques, etc.)

35

Cahier des charges

D) Déroulement du projet

- ▶ **1 – Planification**
Représenter l'articulation des grandes phases du projet et des principaux jalons.
- ▶ **2 – Ressources**
Lister les ressources humaines et matérielles que le client peut mettre à la disposition du prestataire.

36

Cahier des charges

E) Authentification

Date et signature du chef de projet et du maître d'ouvrage.

F) Annexes

Lister et joindre au cahier des charges les éventuels documents que le client peut mettre à disposition.

37

spécifications

- ▶ Du cahier des charges => on en déduit une spécification détaillée (avec pl. étapes)
- ▶ “ce que vous appelez l’analyse”

- ▶ Plusieurs modèles de spécifications (tous sont formels)
 - Automates, automates temporisés
 - Réseau de petri
 - Lds, lotos et les dérivés stockastiques et temporisés
 - UML, SYSML et les dérivés, merise,...
 - BPEL, (services)

38

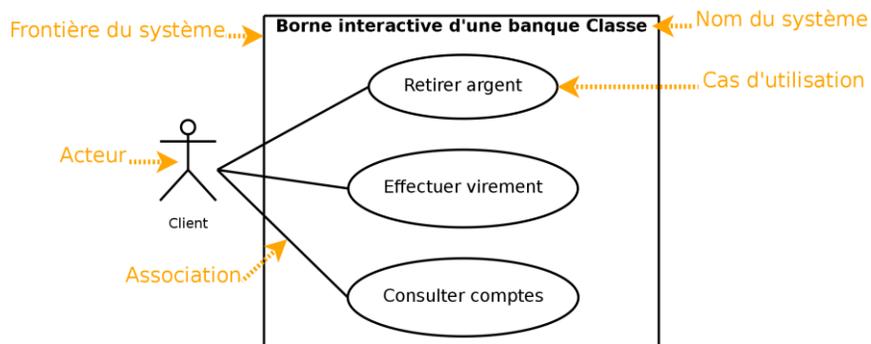
UML

- ▶ **La modélisation UML**
- ▶ UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux **vues**.
- ▶ **Les vues statiques**, c'est-à-dire représentant le système physiquement
 - ▶ diagrammes d'objets
 - ▶ diagrammes de classes
 - ▶ diagrammes de cas d'utilisation
 - ▶ diagrammes de composants
 - ▶ diagrammes de déploiement
- ▶ **Les vues dynamiques**, montrant le fonctionnement du système
 - ▶ diagrammes de séquence
 - ▶ diagrammes de collaboration
 - ▶ diagrammes d'états-transitions
 - ▶ diagrammes d'activités

39

UML

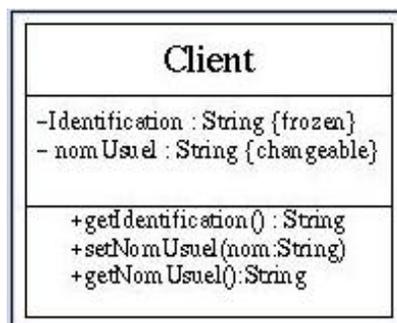
1. Cas d'usages: décrit de façon fonctionnelle le système



40

UML

2. diagramme de classes

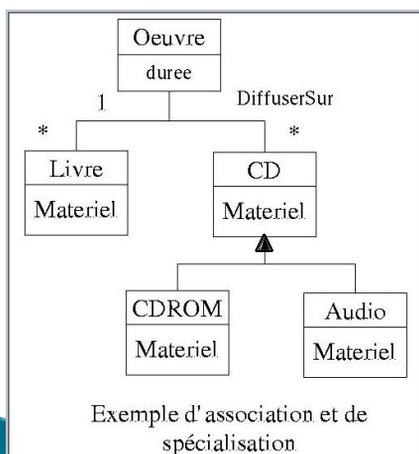


+ public
- Private
protected

41

UML

2. diagramme de classes

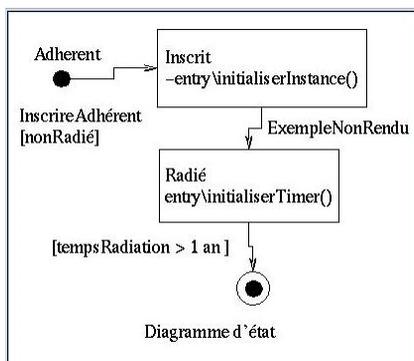


- ▶ Héritage "est une sorte de"
- ◆ Composition (classe ne peut pas exister par elle même)
- association
- ◇ Aggrégation
Une classe regroupe d'autres (instance,...)

42

UML

- ▶ 3. diagramme d'états: montre les états d'un objet au cours du temps

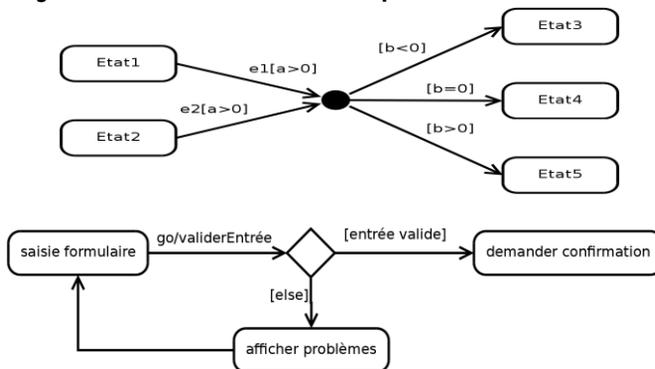


Point de choix
Decision (losange)

43

UML

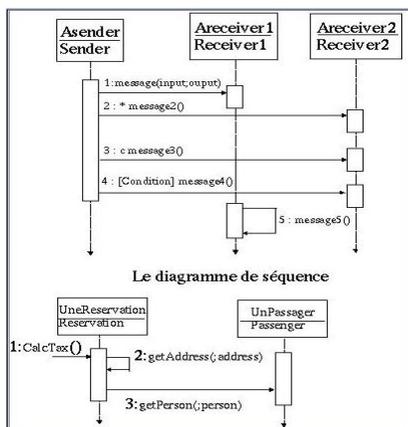
- ▶ 3. diagramme d'états: montre les états d'un objet au cours du temps



44

UML

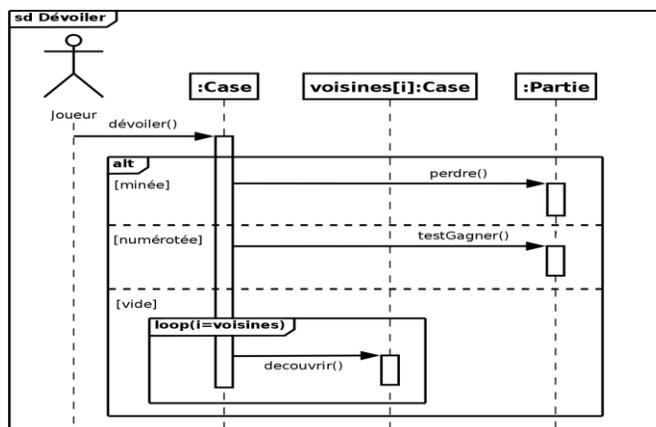
- ▶ 4. diagramme de séquence : relations entre objets dans le temps



45

UML

- ▶ 4. diagramme de séquence : opérateurs (alt, loop, ...)



46

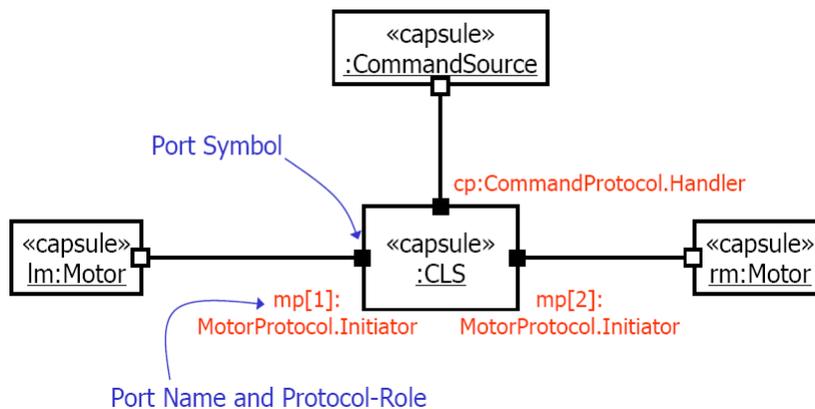
UML

▶ Beaucoup d'extensions:

- SYSML: Systems modeling langages
 - Langage plus léger
 - Permet de modéliser des performances, des contraintes mécaniques,...
- UML-RT (real time)
 - Ajout de concept comme la capsule (objet actif)
 - Ports
 - Collaboration de diagrammes,...

47

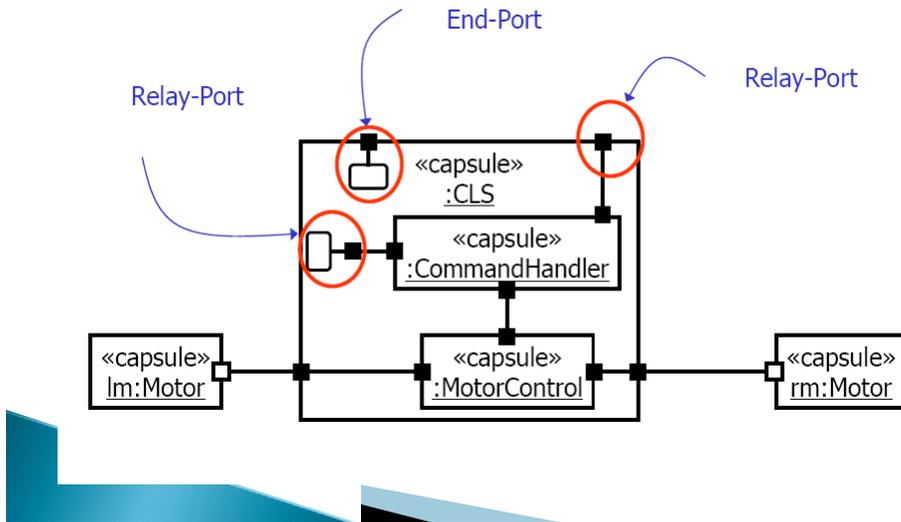
UML-RT



48

UML-RT

Ports et connecteurs



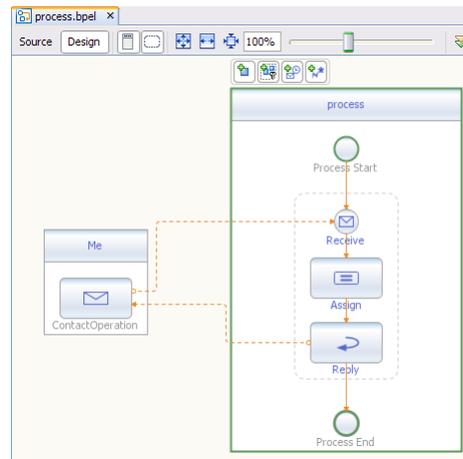
49

WS-BPEL

- ▶ Langage de modélisation d'architectures orientée services (SOA)
- ▶ Processus abstrait pour modéliser l'envoi/ l'attente de messages (orchestration)
- ▶ Utilisé avec les web services
- ▶ => donne un fichier XML directement utilisable par des moteurs de gestion des processus métiers, ces derniers se chargeant d'appliquer les règles du fichier en question.
 - On met en place les services web, on donne ce fichier qui indique comment interagissent les SW et ... c'est tout

50

WS-BPEL



51

Les automates

- ▶ Automates: très (très) utilisés pour modéliser les états d'un système d'un objet d'un composant.
- ▶ Utilisés dans la plupart des langages actuels (UML,...)

52

Les automates: FSM

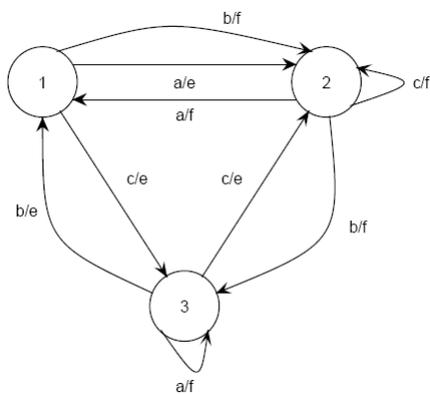
Définition 2.1.2 (FSM) Une FSM A est un quintuplet $\langle s_0, S, I, O, T \rangle$ où :

- $S_0 \in S$ est l'état initial,
- S est un ensemble non vide d'états,
- I est l'ensemble des entrées,
- O est l'ensemble des sorties,
- T est un ensemble non vide de transitions tel que $T \in S \times I \times O \times S$. Le tuple $\langle s, i, o, s' \rangle$ représente une transition qui part de l'état s et va vers l'état s' . Celle-ci est étiquetée par une entrée et une sortie qui symbolisent le fait que pour passer à l'état s' , il est nécessaire d'envoyer l'interaction i et de recevoir l'interaction o . Une telle transition est communément notée par $s \xrightarrow{i/o} s'$.



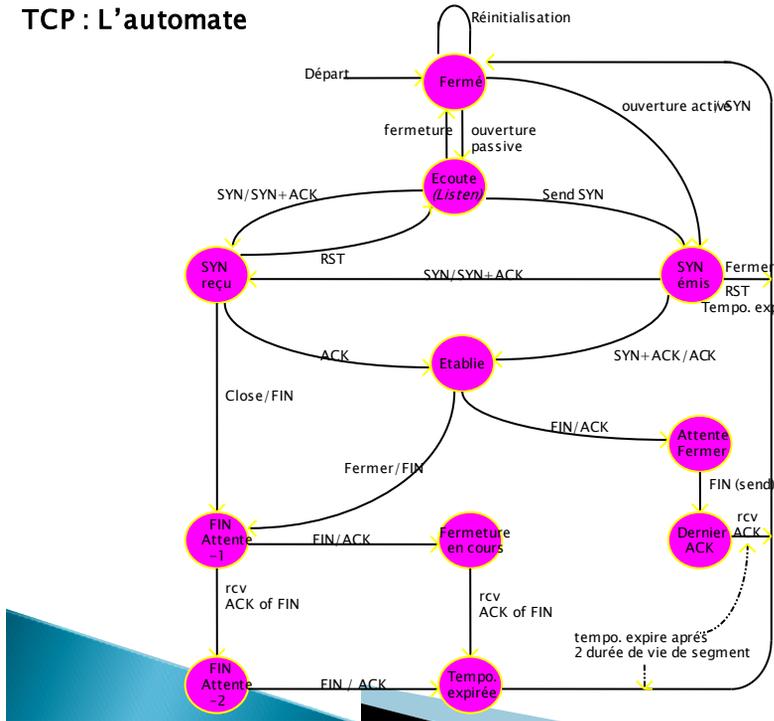
53

Les automates: FSM



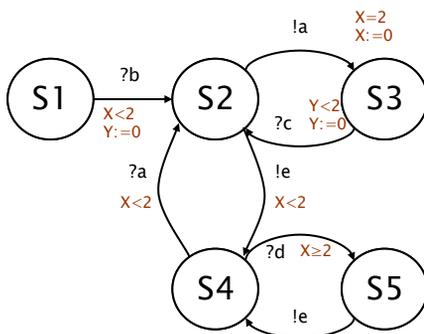
54

TCP : L'automate



55

Les automates: automates temporisés



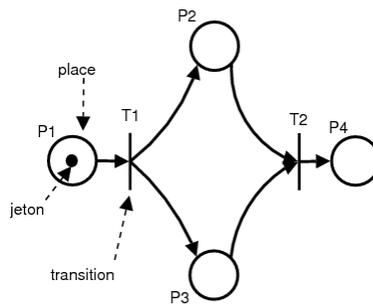
- Automate temporisé d'Alur et Dill composé :
- D'états, de transitions, de symboles,
- D'horloges réinitialisables,
- De contraintes temporelles

• Inconvénient : nombre d'états, du point de vue comportemental et temporel, infinis

56

Les RdP (réseaux de Petri)

- ▶ Un réseau de Pétri est un graphe représentant les relations entre trois ensembles d'éléments :
 - places
 - transitions
 - arcs



57

Beaucoup d'autres langages...

- ▶ SDL (specification and description language)
 - Proche de UML-RT par beaucoup d'aspects
- ▶ LOTOS
 - Pour syst. Distribués et parallèles
 - Notion de rendez vous et d'actions
 - Complexe
- ▶ Logiques,...
- ▶ Estelle, BPML, langage B, VRML (réalité virtuelle),

58

Et d'outils

- ▶ Beacoups d'outils permettent de décrire des modèles
 - Uml: netbeans, eclipse, argouml,...
- ▶ Rappel:
- ▶ VOUS POUVEZ GENERER AUTOMATIQUEMENT LES CLASSES ET PARFOIS DU CODE
- ▶ Parfois vous pouvez pratiquer directement la phase de vérification et générer automatiquement les tests

59

vérification

Preuve, model checking