







Plan du cours

-  **Taxonomie des systèmes informatiques**
-  **Systèmes temps réel**
-  **Spécificités des OS pour le temps réel**
-  **L'OS Xenomai pour le temps réel**
-  **Systèmes embarqués**
-  **Linux pour l'embarqué**
-  **Marché des OS pour le temps réel et l'embarqué**
-  **Modélisation d'applications temps réel avec UML 2.x**



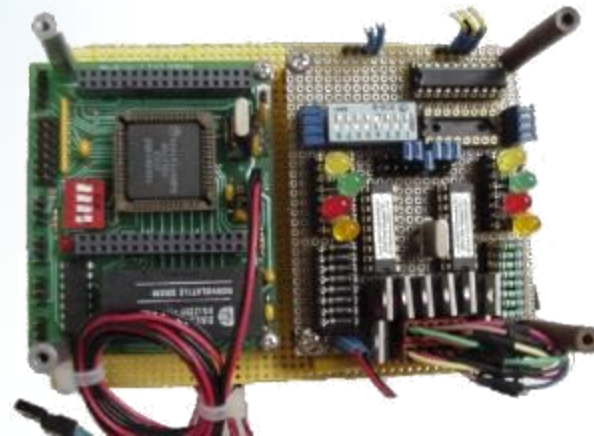
Les contraintes de modélisation pour le temps-réel

- Résultat fourni dans le **respect des contraintes temporelles** (échéances)
- Pseudo-**concurrency** (monoprocasseur) et concurrence réelle (multiprocasseur)
- **Ordonnancement** des activités **périodiques / apériodiques**
- Partage de **ressources**
- **Prédictibilité, ordonnançabilité et déterminisme**
- **Tolérance aux fautes et sûreté de fonctionnement**



Les contraintes de modélisation pour l'embarqué

- **Fiabilité**
- **Sécurité**
- **Modularité**
- **Spécificité** (support hardware)
- **Encombrement**
- **Energie**
- **Coût**





Avantages de la modélisation objet

- **Consistance des vues** du modèle
- Abstraction adaptée au domaine du problème
- **Flexibilité** accrue
- Modularité
- **Réutilisation** facilitée
- **Evolutivité** aisée
- Facilité pour le déploiement d'applications



UML 2.x : les diagrammes

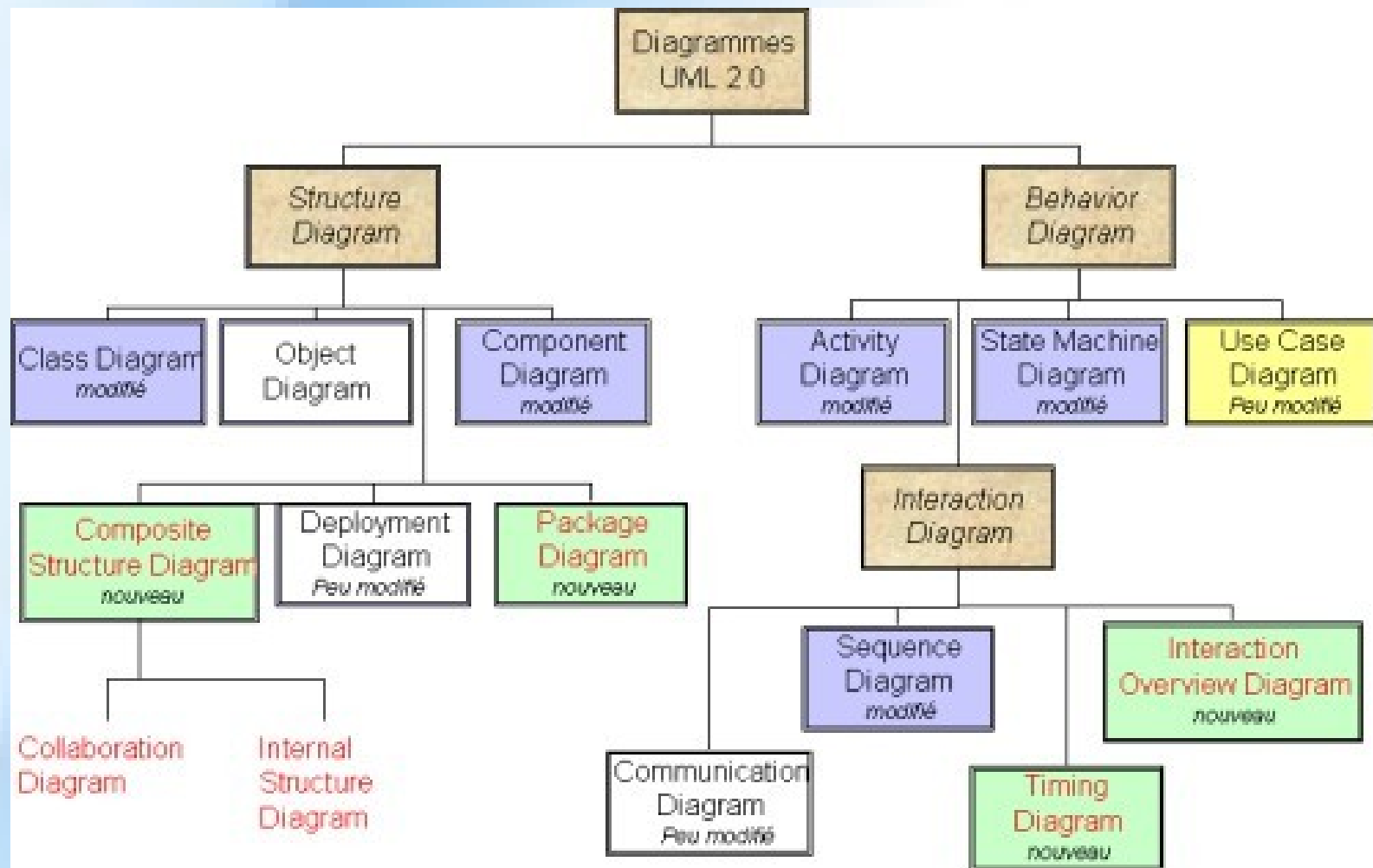
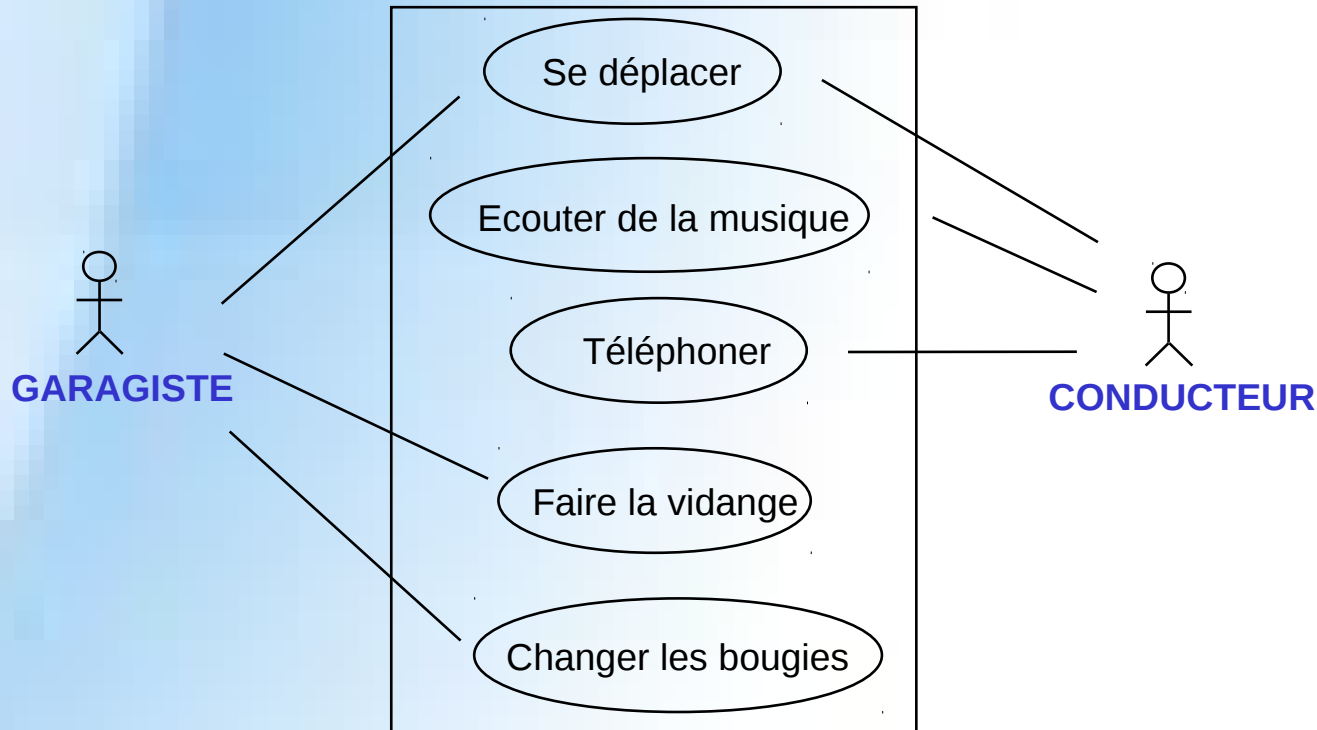


Diagramme des cas d'utilisation

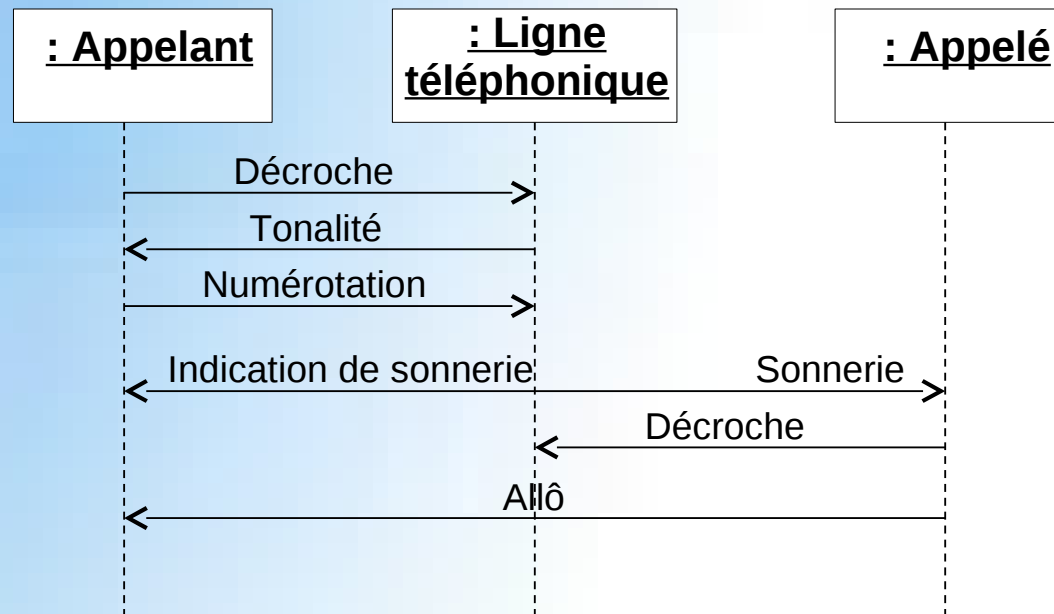


- Comment spécifier des contraintes temporelles ?

- « Lorsque l'acteur **CONDUCTEUR** envoie une commande *ClignotantDroit*, le temps de réponse du système ne doit pas excéder **10ms** »



Diagramme de séquence



- **Spécification temporelle** sur un diagramme de séquence :

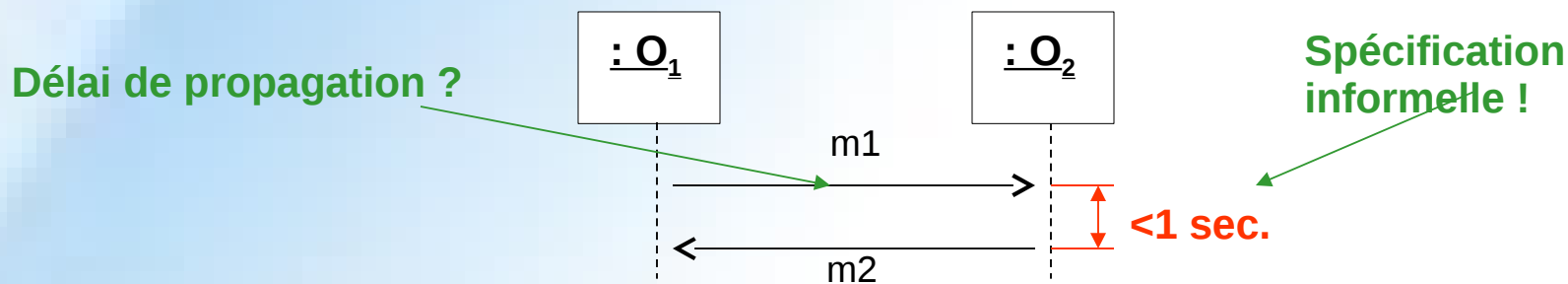


Diagramme d'activité

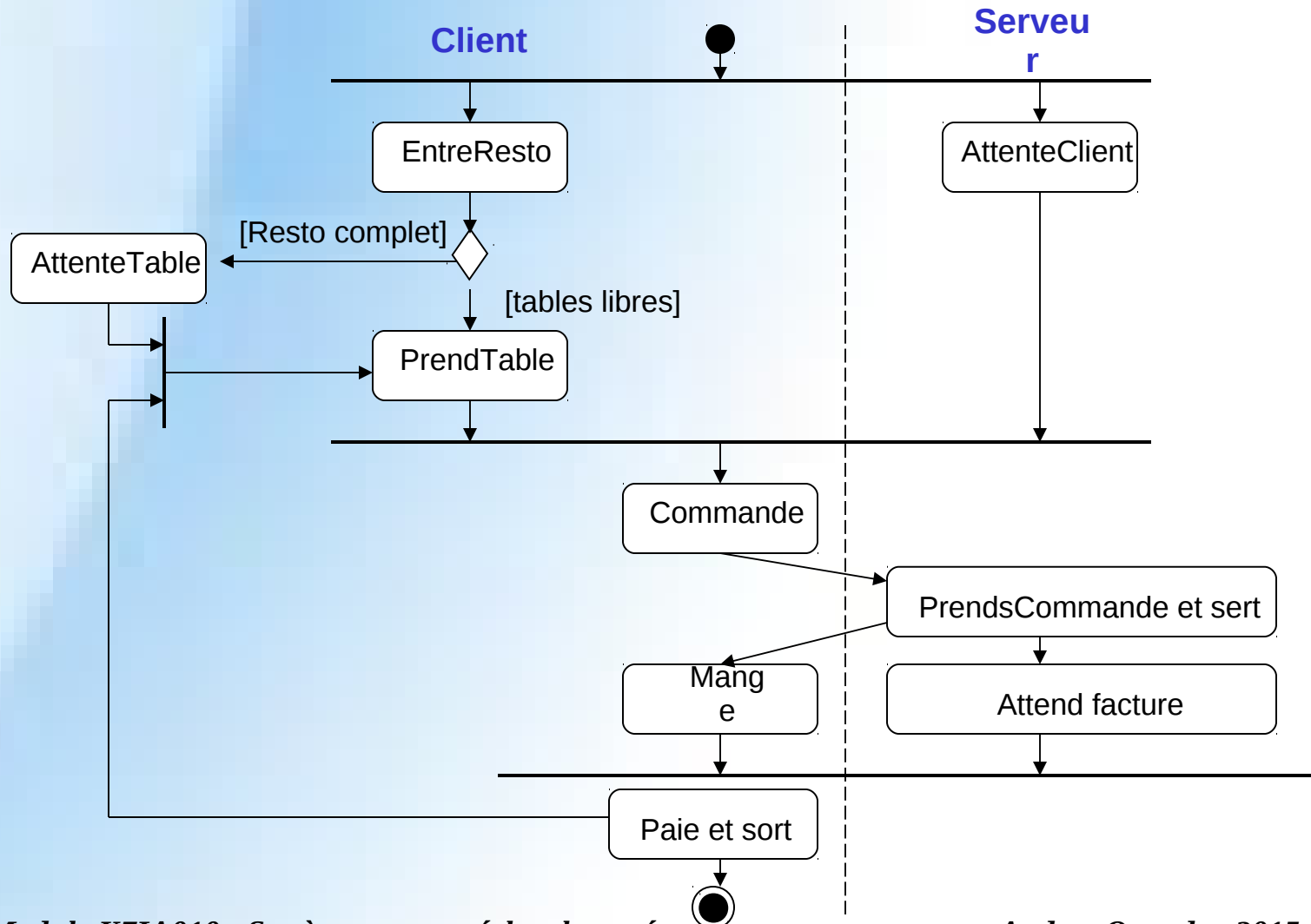
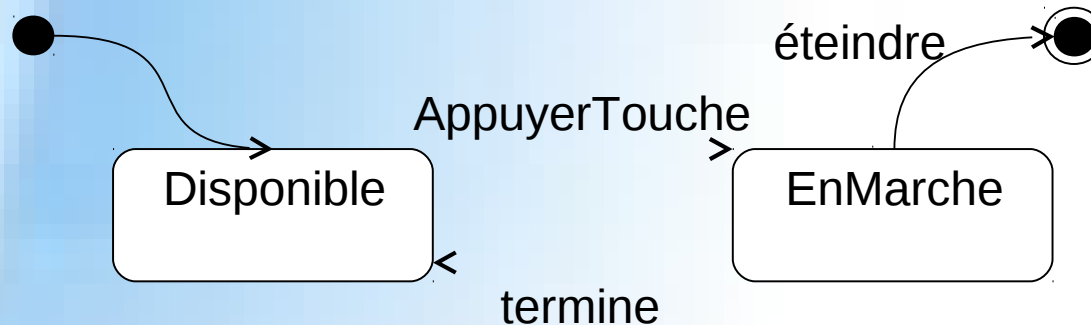


Diagramme d'état-transition

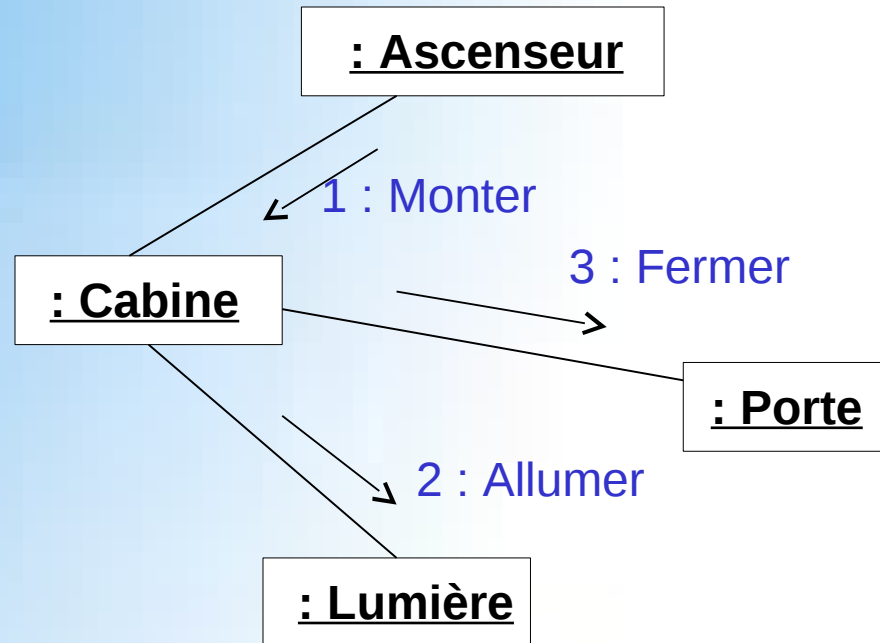


Non-déterministe car traité comme tous les autres événements !

- **Spécification temporelle** sur un diagramme d'état-transition : **TimeEvent**



Diagramme de collaboration

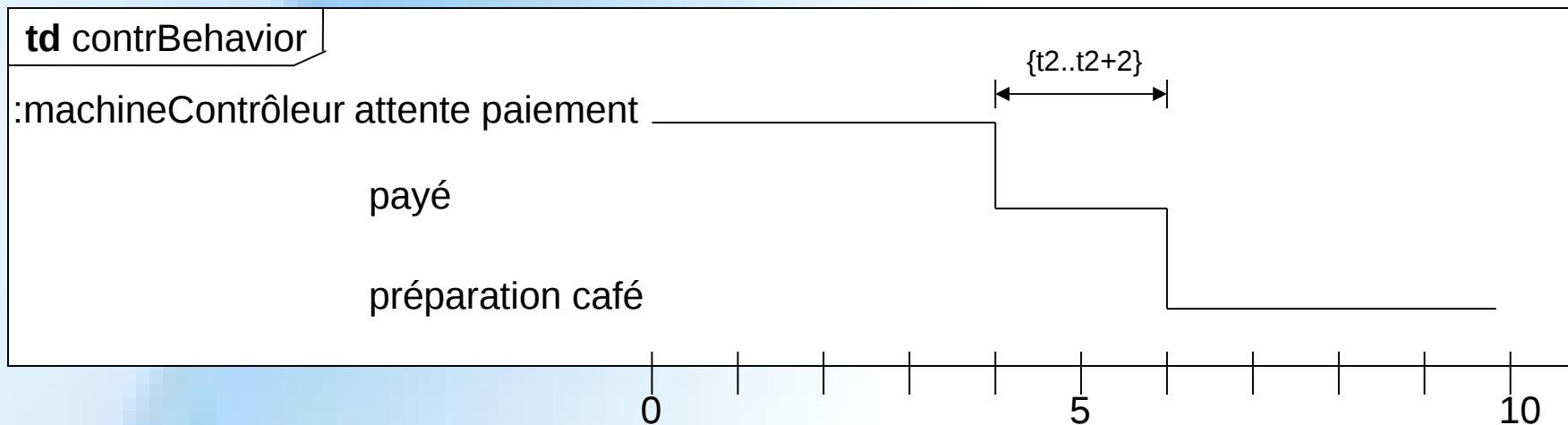


- Un objet **Ascenseur** envoie un message **Monter** à un objet **Cabine**, puis l'objet **Cabine** envoie un message **Allumer** à un objet **Lumière**, et enfin **Cabine** envoie un message **Fermer** à un objet **Porte**.

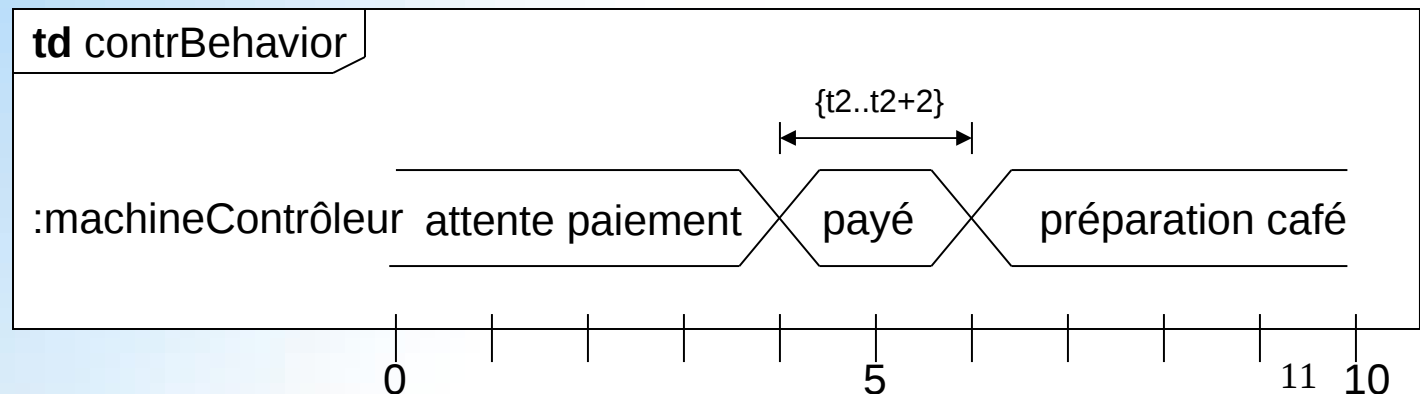


Les diagrammes de modélisation comportementale (3)

- Le diagramme de timing (nouveau diagramme) :



Vue
compacte :





Les outils du marché pour le développement des SETR (1)

- Les outils propriétaires :

- **Enterprise Architect** (UML 2.0) de chez Sparx Systems (<http://www.sparxsystems.com.au/>)
- **Rational Rose Real-Time** (UML-RT) de chez IBM (<http://www.rational.com>)
- **Poseidon for UML Professional Edition** (UML 2.0) (<http://www.gentleware.com>)
- **Real-Time Studio** (ACCORD/UML) de chez ARTiSAN (<http://www.artisansw.com>)
- **Rhapsody** (UML 2.0) de chez ILogix (<http://www.ilogix.com>)
- **Metamill** (UML 2.1) de Metamill Software (<http://www.metamill.com>)
- **UModel** (UML 2.2) de chez Altova (<http://www.altova.com/umodel.html>)
- **Visual Paradigm for UML** (UML 2.2) de chez Visual Paradigm... (<http://www.visual-paradigm.com>)



Les outils du marché pour le développement des SETR (1)

- Les outils open-source :

- **ArgoUML** (UML 1.4) (<http://argouml.tigris.org>)
- **Poseidon Community Edition** (UML 2.0 partiel) (<http://www.gentleware.com>)
- **BOUML** (UML 2.2) (<http://bouml.free.fr>)
- **EclipseUML Studio** (UML 2.2) (<http://www.uml2.org>)



Références

- B. P. Douglass, **Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks and Patterns**, Addison Wesley, 1999
- B. P. Douglass, **Real-Time UML : Developping Efficient Objects for Embedded Systems**, Addison Wesley, 1999
- J. Rumbaugh, I. Jacobson, G. Booch, **UML 2.0 Guide de Référence**, CampusPress, 2004
- C. Larman, **UML 2 et les design patterns**, Pearson Education, 2005
- P. Roques, F. Vallée, **UML 2 en action : De l'analyse des besoins à la conception J2EE**, Eyrolles, 2004
- <http://www.omg.org>