

Formalisation des diagrammes états-transitions UML concurrents

Étienne André, Mohamed Mahdi Benmoussa, Christine Choppy
LIPN, UMR, CNRS 7030 – Université Paris 13, Sorbonne Paris Cité

MSR 2013

1

Vérification de système complexe

- ❑ Système critique et complexe
- ❑ Pas de défaillance ?
- ❑ **Nécessité de validation avant la mise en œuvre**

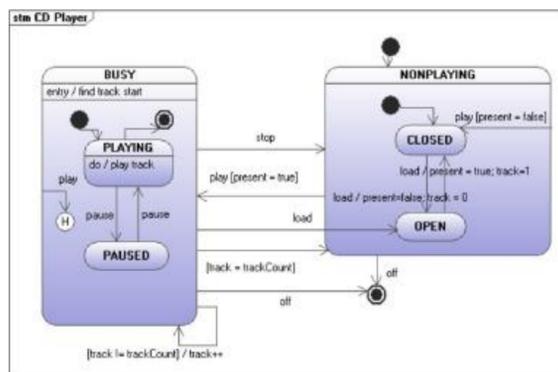


Centrale nucléaire de Chooz » par MOSSOT, Wikicommons, cc-bys

2

Diagrammes états-transitions UML

- ❑ Systèmes composés d'états et de transitions
- ❑ Très utilisés dans l'industrie
- ❑ Définis par l'OMG [1] (Version 2.4)
- ❑ Sémantique exprimée informellement



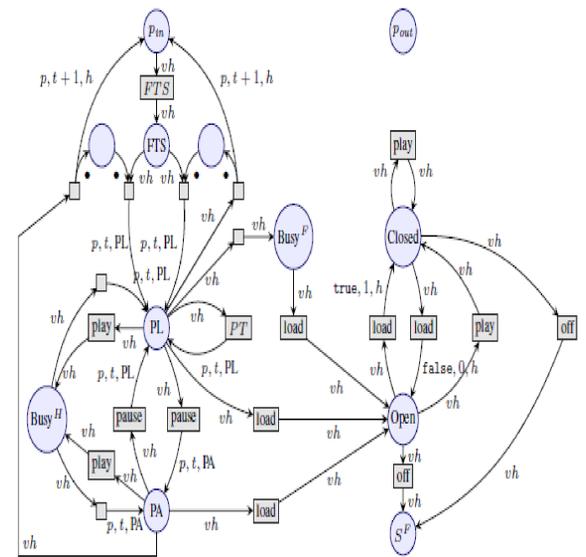
Lecteur de CD [3]

- ❑ **Système complexe → Modélisation complexe → Difficulté de l'analyse**

3

Formalisation des diagrammes états-transitions UML

- ❑ **Proposition** : traduction vers les réseaux de Petri colorés



Réseau de Petri coloré du lecteur de CD

- ❑ **Travaux de l'article [2]**
- ❑ Basé sur des algorithmes de traduction
- ❑ Traduction restreinte au cas non concurrent

4

Extension au cas concurrent

- ❑ Extension des travaux de l'article [2]
- ❑ Introduction de la **concurrency**
 - ❑ Prise en compte des **pseudo-états Fork et Join**
 - ❑ Prise en compte des **régions des états composites**
- ❑ **Implémentation** d'une partie de la traduction

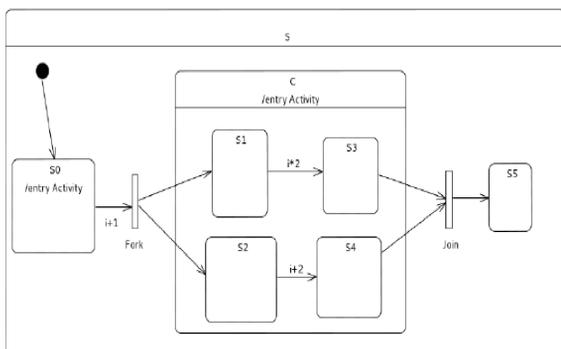


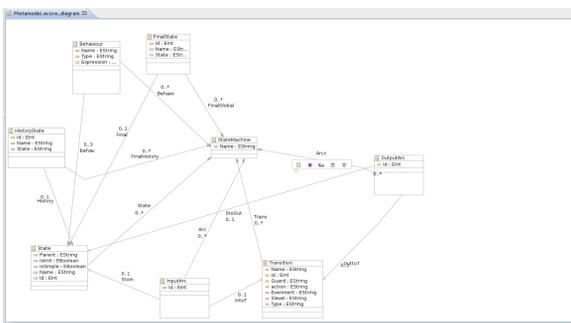
Diagramme états-transitions concurrent

- ❑ **Difficulté de l'adaptation**
 - ❑ Ambiguïté de la sémantique d'UML
 - ❑ Détection des cas non pris en compte par l'article [2]

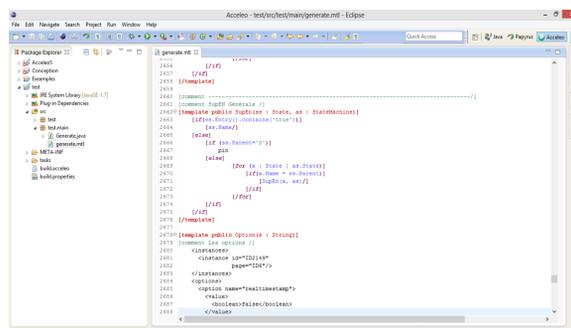
5

Implémentation avec Aceleo

- ❑ Basée sur la **transformation de modèles**
- ❑ **EMF** pour la modélisation



- ❑ **Aceleo** pour la transformation
 - ❑ Transformation modèle-texte
 - ❑ Difficulté de l'adapter pour la traduction



6

Perspectives

- ❑ Prise en compte de plus d'éléments syntaxiques (événements différés, aspects temporels)
- ❑ Trouver un meilleur outil que Aceleo pour l'implémentation
- ❑ Comparaison de la sémantique de traduction avec d'autres sémantiques

7

Références bibliographiques

- [1] **OMG. Unified Modeling Language Superstructure, Version 2.4, 2011.**
- [2] **É. André, C. Choppy, and K. Klai. Formalizing non-concurrent UML state machines using colored Petri nets. ACM SIGSOFT Software Engineering Notes 37(4) : 1-3, 2012.**
- [3] **Shaojie Zhang and Yang Liu. An automatic approach to model checking UML state machines. In SSIRI-C'10, pages 16. IEEE,**