AVATAR

Un profil SysML temps réel outillé

Ludovic Apvrille, Pierre de Saqui-Sannes

ludovic.apvrille@telecom-paristech.fr
pdss@isae.fr

SysML France, 6 décembre 2010

Agenda

- **De TURTLE à AVATAR**
- **Le langage AVATAR : syntaxe et sémantique**
- **Vérification logique et temporelle de modèles AVATAR**
- **Vérification orientée *security* de modèles AVATAR**
- **Positionnement et perspectives**

De TURTLE à AVATAR

Capitaliser sur plus de 10 ans d'expérience des profils UML temps réel

- **TURTLE**
  - Analyse et conception axées vérification de systèmes temps réel
  - Validation d'architectures de communication et implantation répartie
- **DIPLODOCUS**
  - Systems-on-Chip (SoC)
  - Exploration d'architectures (*design space exploration*)
- **Outil open source TTool (TURTLE Toolkit)**
  - Première version en 2003 !
  - Editeurs
  - Générateurs de code
    - Preuve: "partenaires" : UPPAAL, CADP, RTL, …
    - Simulation: SystemC / C++
    - Execution: C-POSIX, Java, middleware
  - Assistant méthodologique

Pourquoi un profil SysML temps réel ?

- **TURTLE : un avant goût de SysML ?**
  - Modélisations abstraites orientées PIM (Platform Independent Modeling)
  - Architecture de boites avec ports n'exploitant pas l'héritage
  - Comportements : la partie "contrôle" prime sur la partie "données"
- **Limitations rencontrées avec TURTLE**
  - Le diagramme d'exigences de SysML est très utile mais ne fait pas partie d'UML
  - Bien que la modélisation soit orientée vérification, les propriétés à vérifier ne font pas partie du modèle
  - La syntaxe de TURTLE doit être revisitée pour se conformer à UML 2.x
- **Modélisation orientée système**
  - Vérification de propriétés
    - Safety
    - Security

AVATAR : carte d'identité

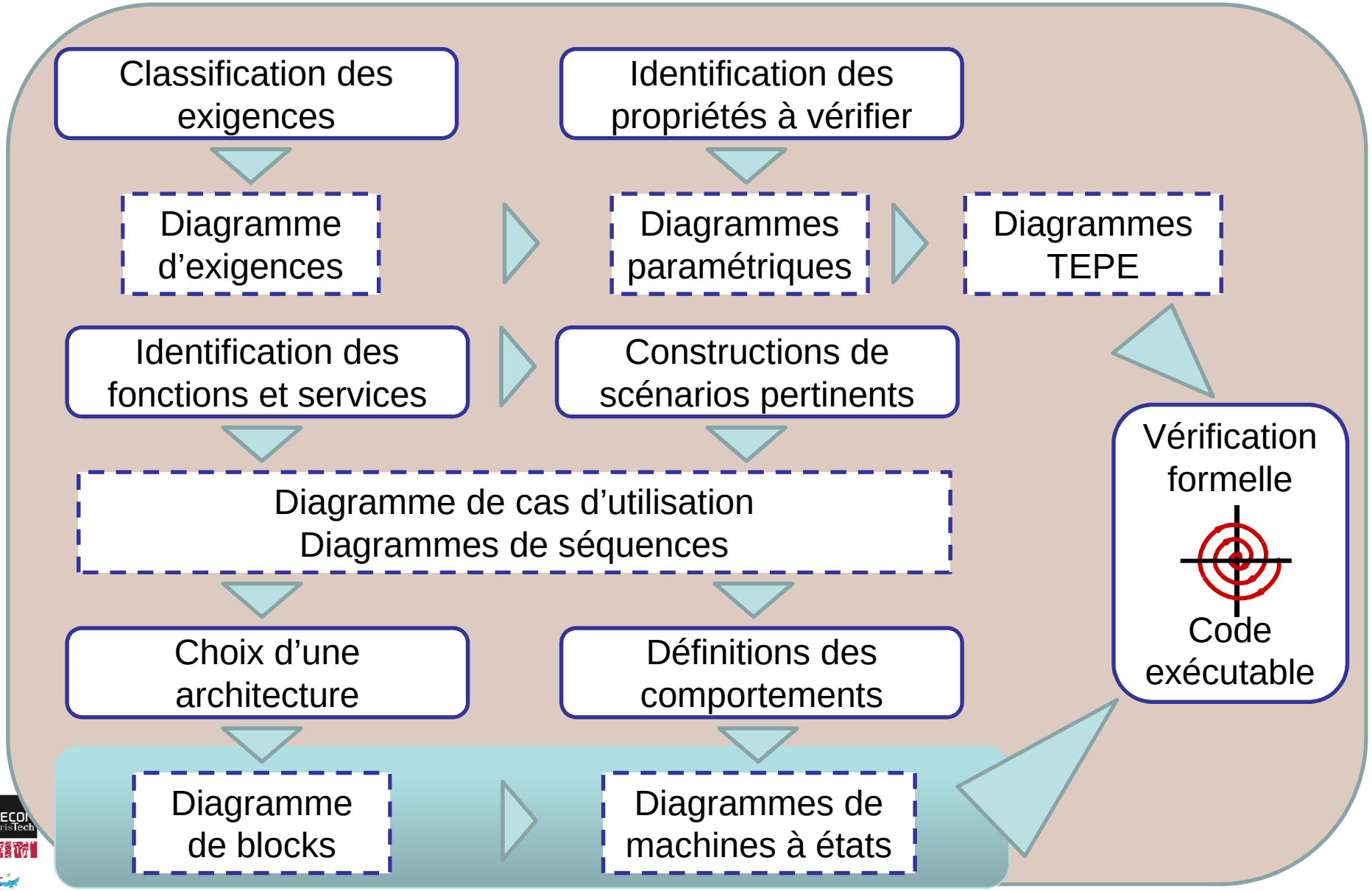
- **AVATAR = Automated Verification of reAl Time softwARe**
- **SysML temps réel orienté vérification de modèles**
- **Syntaxe : conforme au méta-modèle SysML**
- **Sémantique**
  - Modèle de preuve
    - Safety : traduction vers automates temporisés communicants (UPPAAL)
    - Security : traduction vers pi-calculus (ProVerif)
  - Modèle d'exécution : traduction en C/POSIX
- **Outil open-source TTool**

Le langage AVATAR

Positionnement du langage AVATAR par rapport à SysML

- **Conformité stricte à la norme SysML**
  - Diagrammes d'exigences
  - Diagrammes de cas d'utilisation et de séquences
  - Diagrammes de blocks et de machines à états
- **Restrictions**
  - Pas de signaux continus
- **Extensions au méta-modèle UML**
  - IOD (Interaction Overview Diagram) pour structurer les scénarios
  - Machines à états temporisées
  - Diagrammes paramétriques
    - TEPE (TEmporal Property Expression language) : langage d'expression des propriétés logiques et temporelles
  - Expression de propriétés de sécurité
    - Pragmas sur les diagrammes de blocks

Processus associé à AVATAR

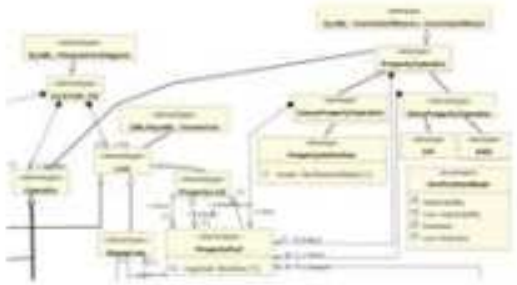


Le langage TEPE d'expression de propriétés temporelles (1/2)

- **Facilité d'utilisation**
  - Intégration dans SysML (Diagrammes paramétriques)
  - Description de haut niveau
  - Plus intuitif que CTL / LTL

- **Pouvoir d'expression**
  - Relation entre les attributs et les signaux des blocks
  - Modélisation de relations logiques et temporelles
  - Défini formellement
    - Traduction vers des observateurs + formules CTL

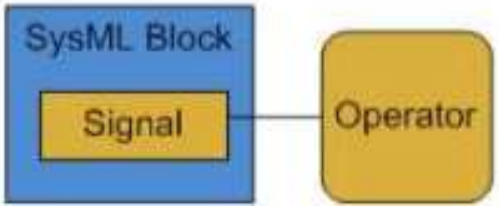
Le langage TEPE d'expression de propriétés temporelles (2/2)



SysML Metamodel

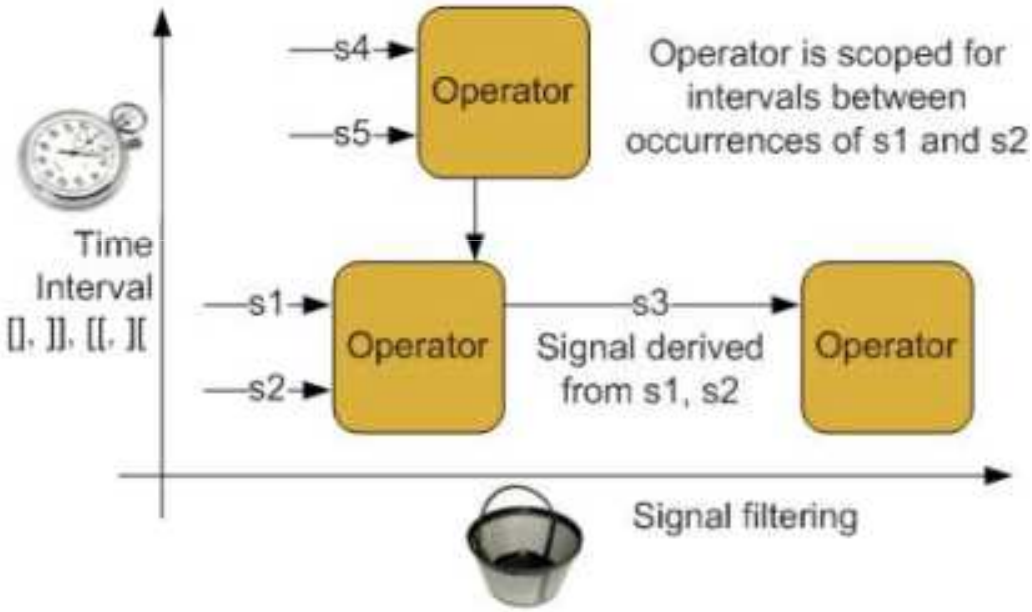
+

Direct reference to the model



+

Verification Concept



= TEPE

Comment construire un diagramme TEPE ?

1. **Blocks représentés avec attributs et signaux**

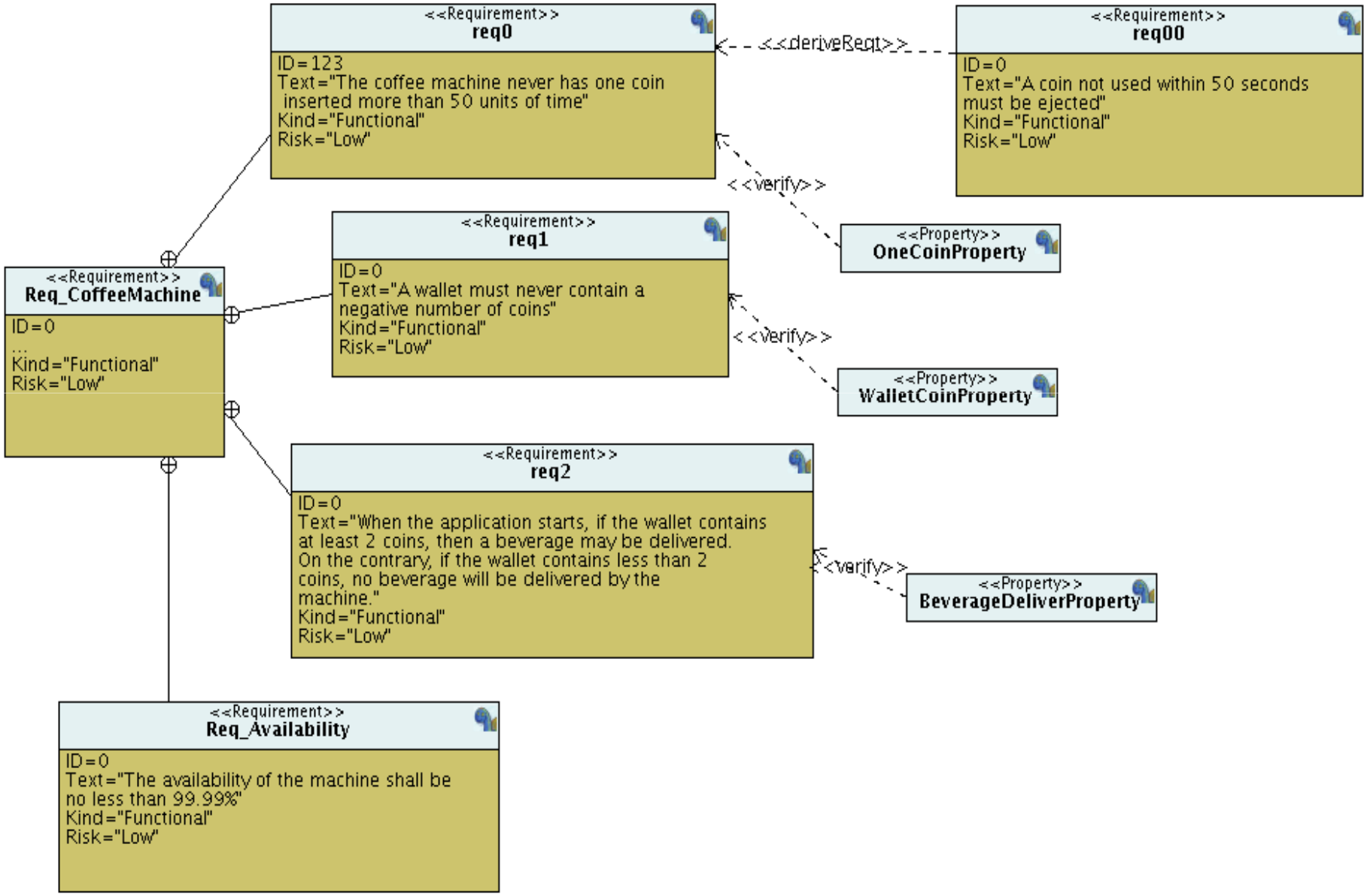
2. **Définition des éléments dérivés (équations, signaux composites)**

3. **Utilisation d'opérateurs temporels et logiques pour mettre en relation les attributs et les signaux -> Propriétés**

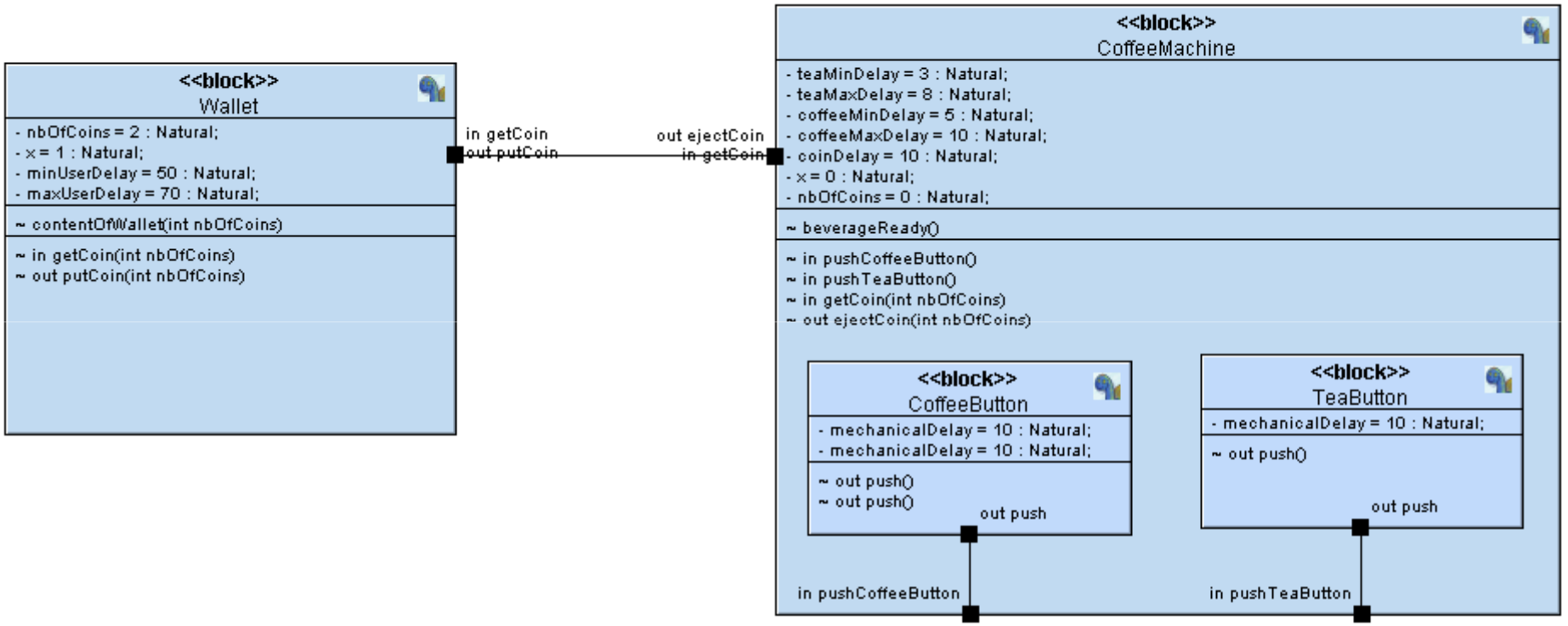
4. **Propriétés mises en relation (AND, OR, etc.)**

5. **Propriétés taguées avec un type (reachability, liveness)**

Machine à café : diagramme d'exigences

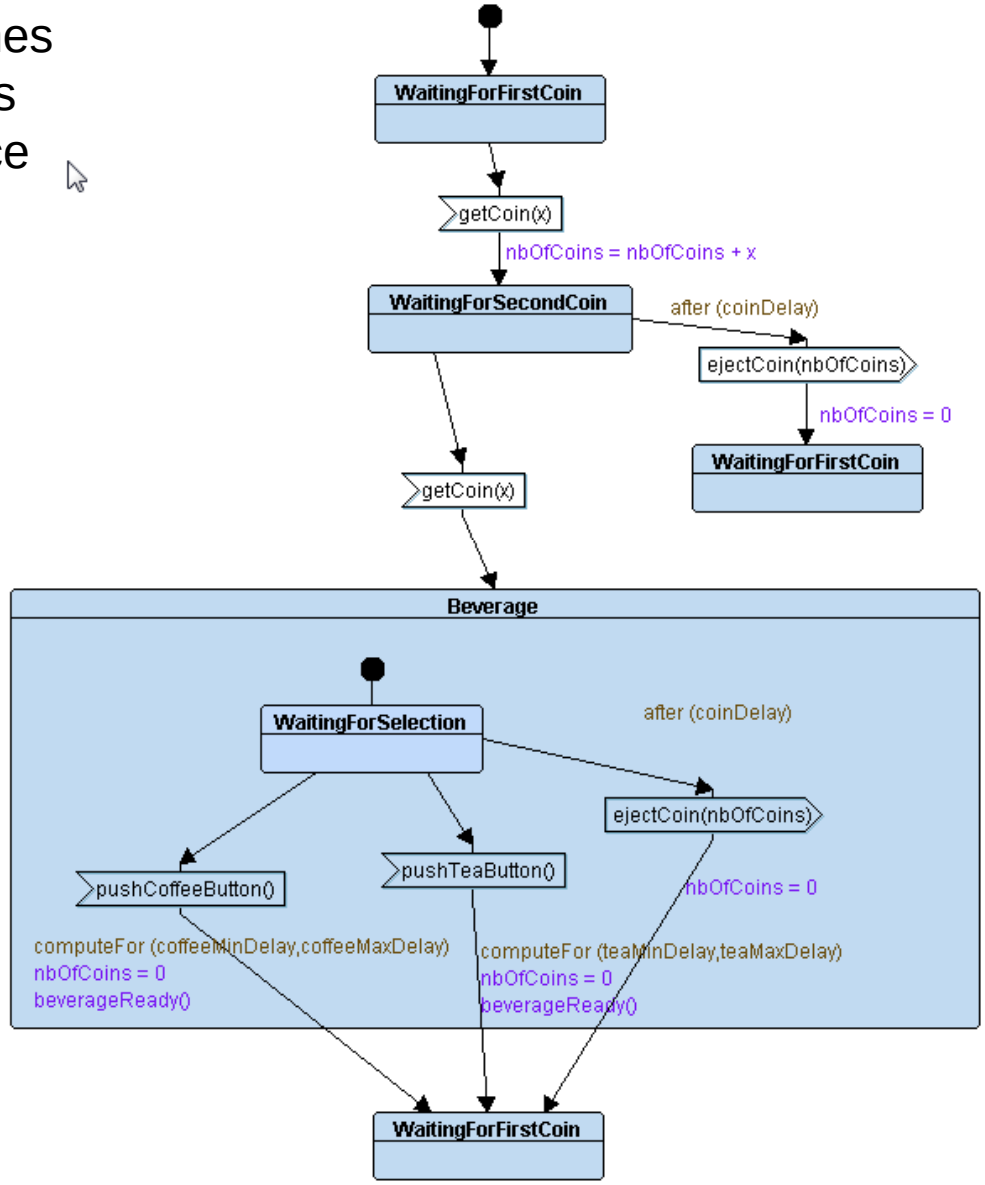


Machine à café : diagramme de blocks

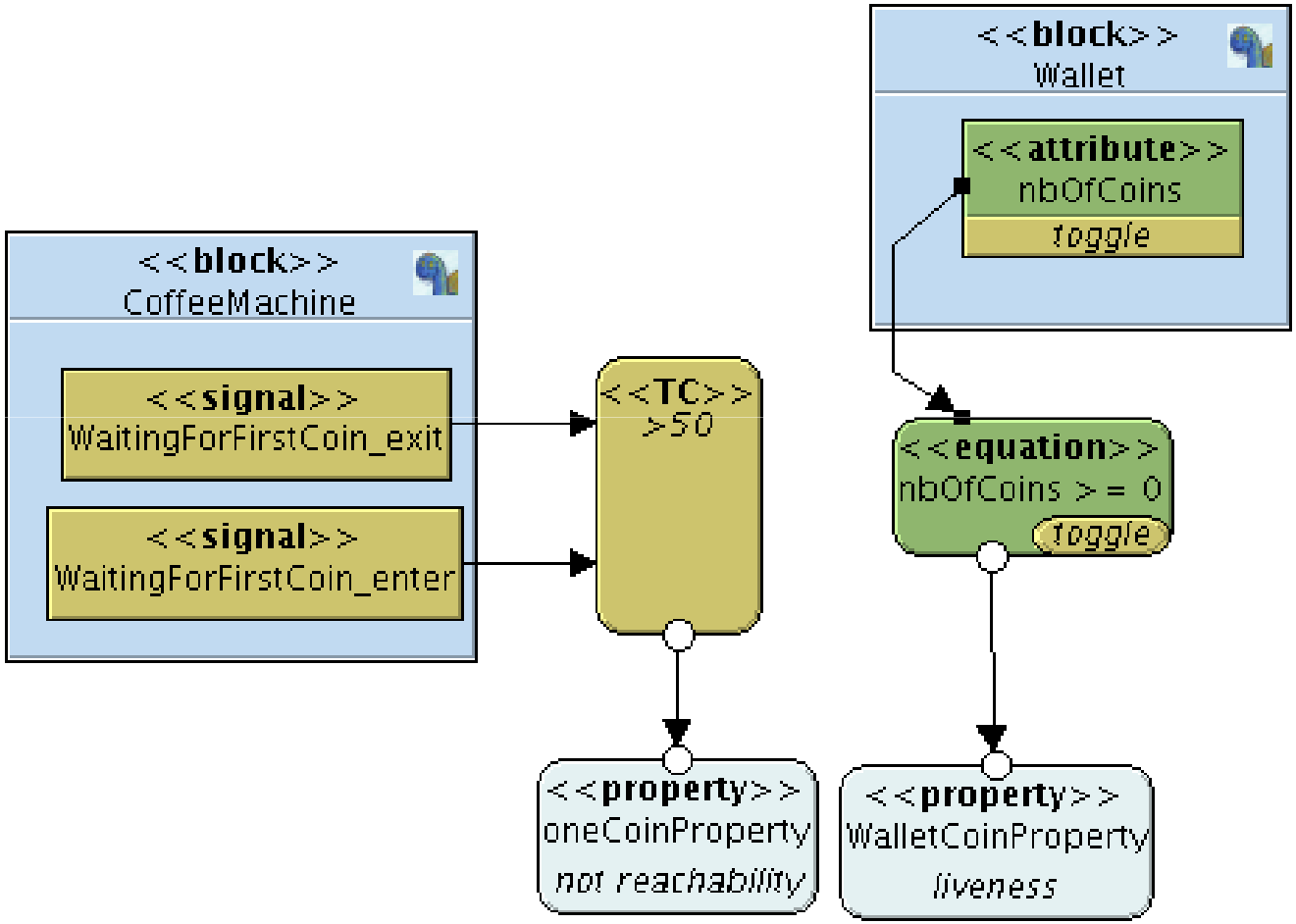


Machine à café:diagramme de machine à états

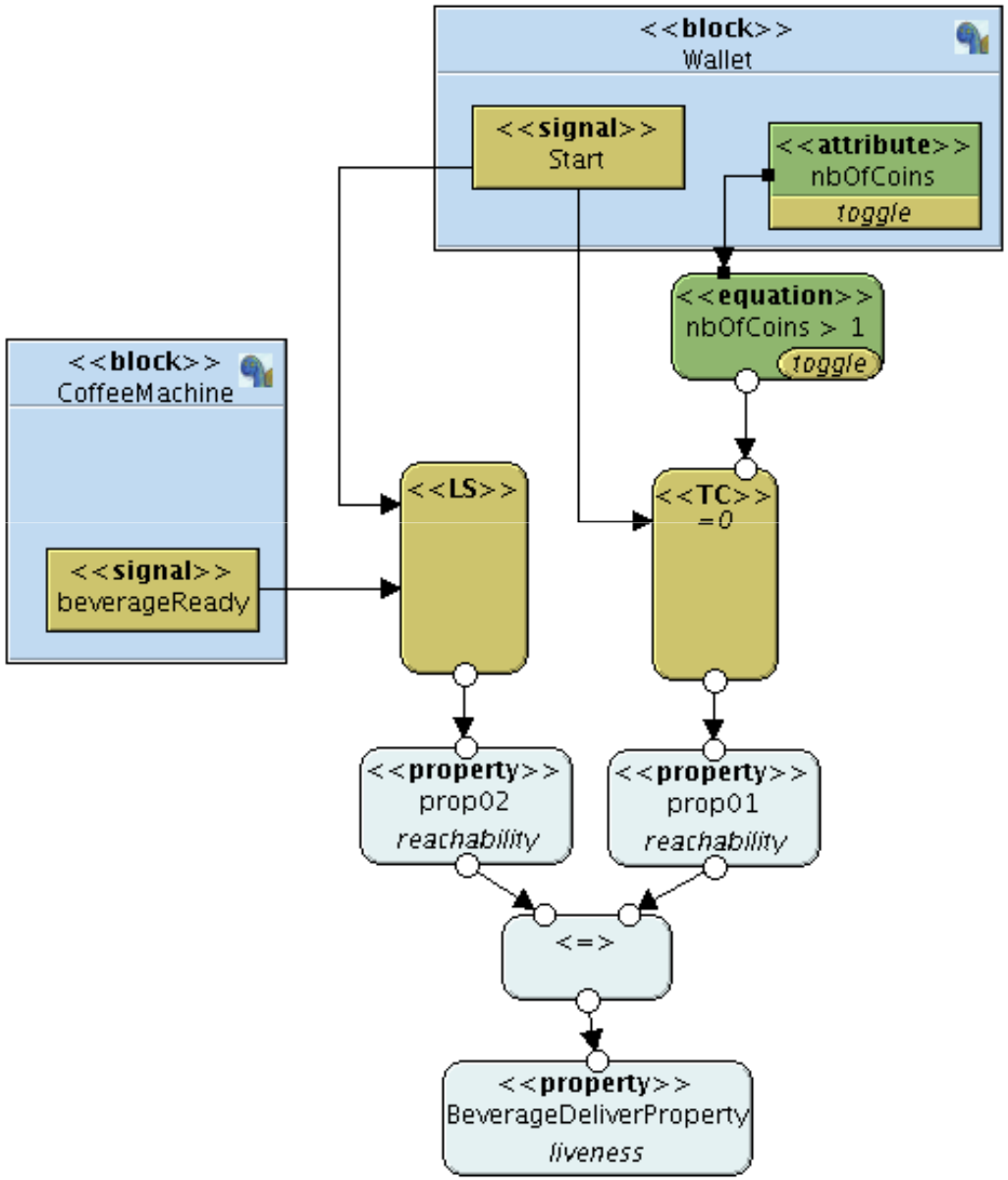
Les autres machines à états ne sont pas présentées dans ce diaporama



Machine à café : diagramme paramétrique



Machine à café : diagramme paramétrique



Vérification formelle de propriétés temporelles

Vérification formelle de propriétés de safety

- **Diagramme TEPE associé à la machine à café**
- **Génération de code UPPAAL**
- **Test d'accessibilité (entre autres) depuis l'interface de TTool sans écrire de formule de logique**
- **Aller plus loin en entrant des formules dans UPPAAL via TTool**
- **Couverture du modèle (parties explorées ou non)**
- **Traçabilité des exigences**

Vérification formelle de propriétés de sécurité

AVATAR: modélisation et propriétés orientées sécurité

- **Idée générale : Spécification dans des notes des diagrammes de block**
  - Référence à des blocks, des attributs, et des états de machine à états

- **Confidentialité**

  # Confidentiality block1.attribute1

- **Authenticité**

  #Authenticity block1.state1.attribute1 block2.state2.attribute2

- **Spécification de données pré-partagées (clés, etc.)**

  #InitialCommonData block1.attribute1 block2.attribute2

AVATAR: preuve de propriétés de sécurité

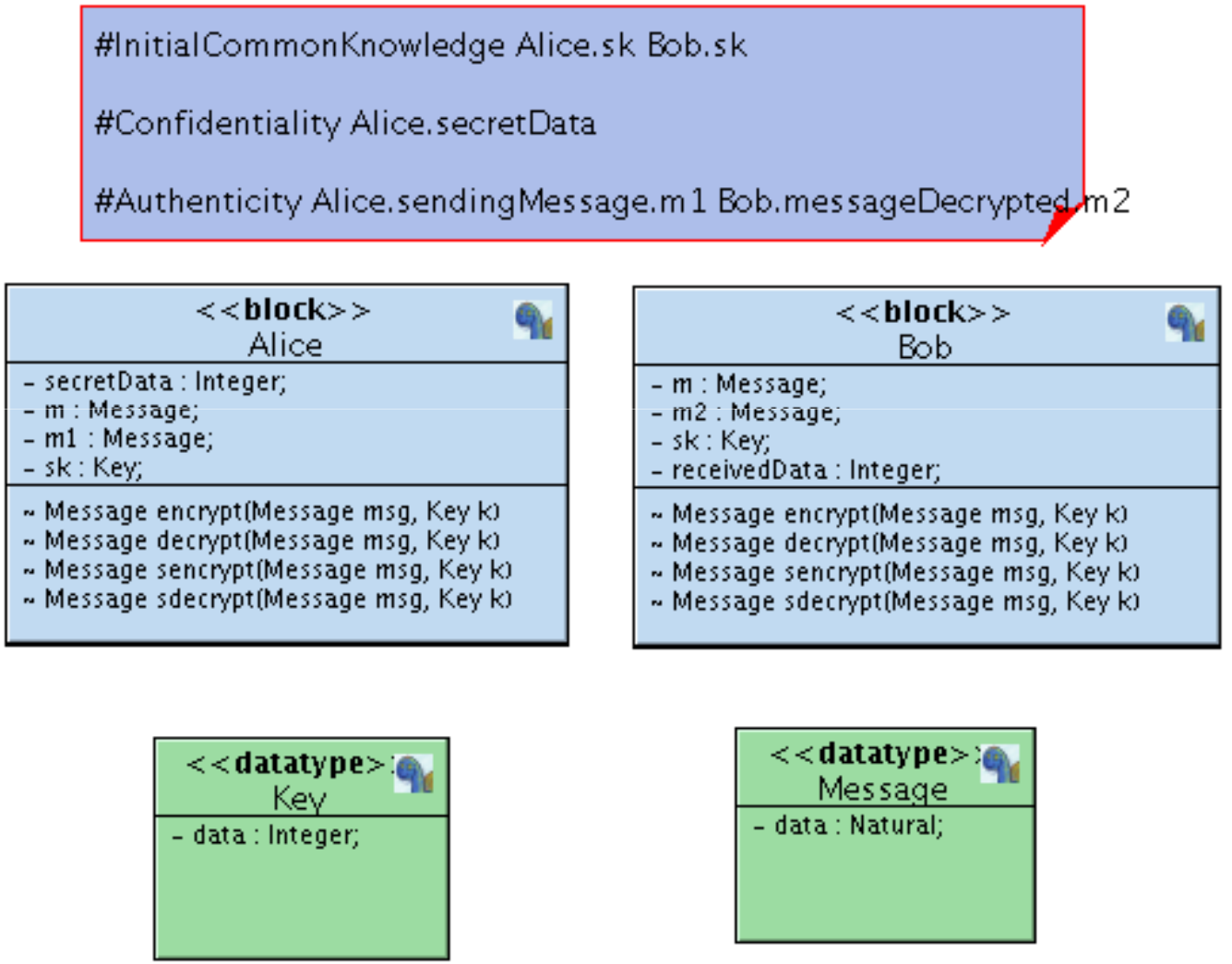
- **AVATAR s'appuie sur l'outil ProVerif de l'ENS**
  - Sémantique basée sur pi-calculus + langage de propriétés (*queries*)
    - Modèle transformé en clauses de Horn par ProVerif
    - Résolution des clauses par ProVerif
    - Preuves réalisées pour un nombre infini d'instances
  - Modèle d'attaquant = Dolev-Yao

- **AVATAR permet la preuve de propriété de**
  - Confidentialité
  - Authenticité

Vérification formelle de propriétés de sécurité

- **Protocole simple d'envoi d'une donnée confidentielle**
  - Alice -> Bob
  - Utilisation d'une clé pré-partagée
- **Génération de code pi-calculus (approche presse-bouton)**
- **Preuve de deux propriétés**
  - Confidentialité
  - Authenticité
- **Couverture du modèle (parties explorées ou non)**

Diagramme de blocks



Positionnement et Discussion

Positionnement par rapport aux travaux du domaine

- **Langage**
  - Pouvoir d'expression
  - Conformité aux standards
    - MARTE
    - SysML
  - Sémantique
    - UPPAAL, ProVerif
- **Outils**
  - Papyrus
  - TOPCASED
- **Méthodologie**

Perspectives

- **Langage + TTool + méthodologie**
  - Assistant méthodologique
  - Exploiter le graphe d'accessibilité d'UPPAAL pour le minimiser et caractériser par exemple l'automate du service rendu par une couche de protocole
  - Améliorer le processus de vérification formelle en tenant compte des dépendances entre éléments d'un modèle
- **Un outil pédagogique accessible**
  - Rédaction de tutoriels
  - Formation (Telecom Paristech, ISAE, …)
- **Applicabilité de TEPE en dehors du langage AVATAR**

Bibliographie annotée

- Article fondateur du profil UML temps réel TURTLE qui a scellé le lien entre UML/TURTLE et vérification outillée de propriétés temporelles

  L. Apvrille, J.-P. Courtiat, C. Lohr, P de Saqui-Sannes

  TURTLE: A Real-Time UML Profile Supported by a Formal Validation Toolkit"

  IEEE Transactions on Software Engineering, Vol. 30, No. 7, pp. 473-487, July 2004.

- Présentation du langage d'expression de propriété TEPE

  D. Knorreck, L. Apvrille, P. de Saqui-Sannes,

  TEPE: A SysML Language for Timed-Constrained Property Modeling and Formal Verification

  UML/FM 2010, Shangai, China, November 2010

- Site Web de l'outil open source TTool qui supporte le profil AVATAR
  - **http://labsoc.comelec.enst.fr/turtle/ttool.html**
  - Sous google : TTool UML (premier lien !)

- Sites Web des outils auxquels TTool s'interface (vérification formelle)
  - UPPAAL: http://www.uppaal.com/
  - ProVerif: http://www.proverif.ens.fr/