

Une méthodologie de conception des systèmes distribués basée sur UML

Nouvelles Technologies de la Répartition
(NOTERE'2005)

1^{er} septembre 2005

Ludovic Apvrille

Pierre de Saqui-Sannes

Axelle Apvrille



Plan

- 📄 **TURTLE : un profil UML outillé**
- 📄 **Limites de l'approche actuelle**
- 📄 **Méthodologie proposée**
- 📄 **Formalisation des diagrammes de déploiement**
- 📄 **Génération automatique de code**
- 📄 **Démonstration**
- 📄 **Conclusion**

Contexte : le profil UML TURTLE

Bref historique

- *Introduit en 1999*

- *Partenaires : LAAS/CNRS, ENSICA, Université Concordia, ENST*

- *TURTLE-P (2003), analyse TURTLE (2005)*

Profil UML dont la sémantique formelle est donnée par traduction en RT-LOTOS

- *Applications temps-réel, protocoles*

Outils supports : TTool (ENST) + RTL (LAAS-CNRS)

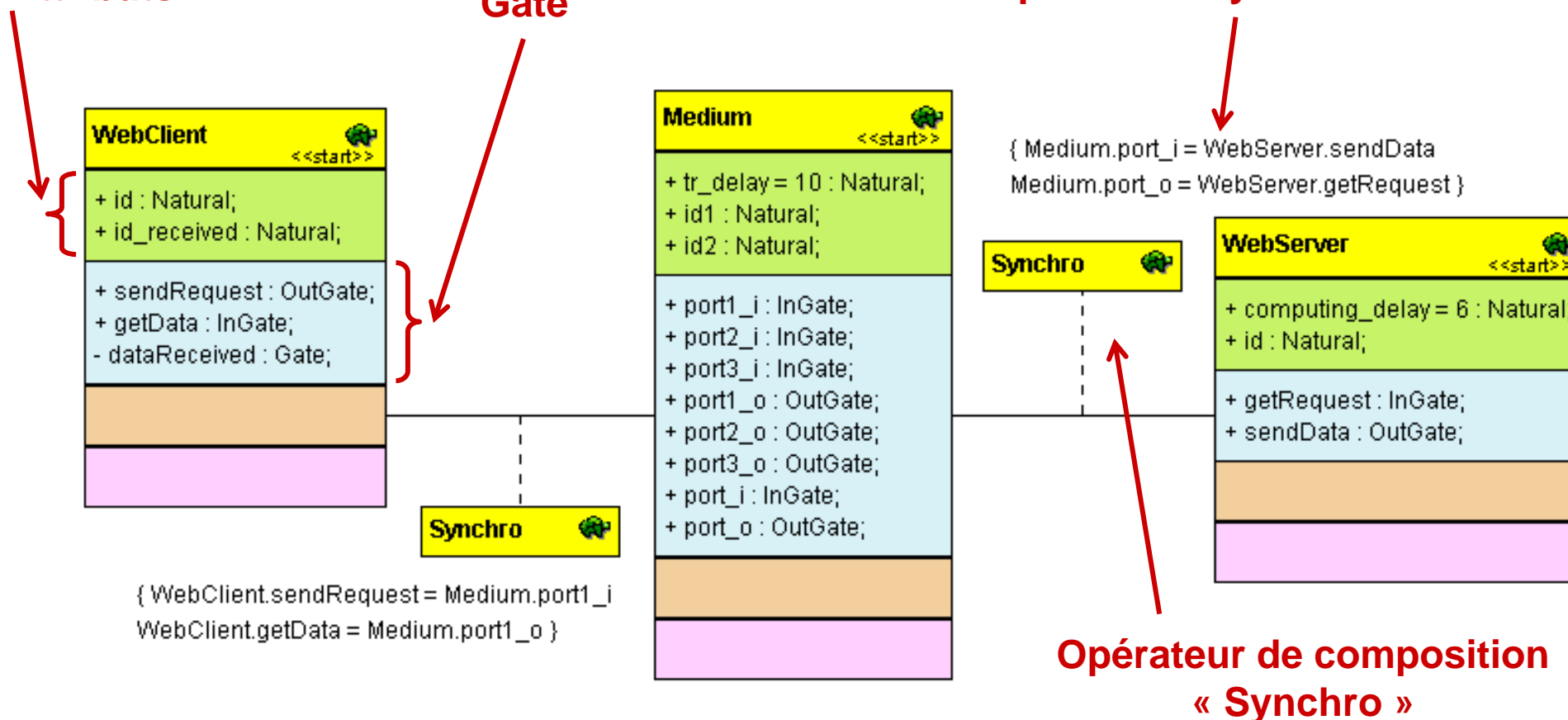
- *Analyse, conception et validation formelle de protocoles / applications*

Diagramme de classes TURTLE

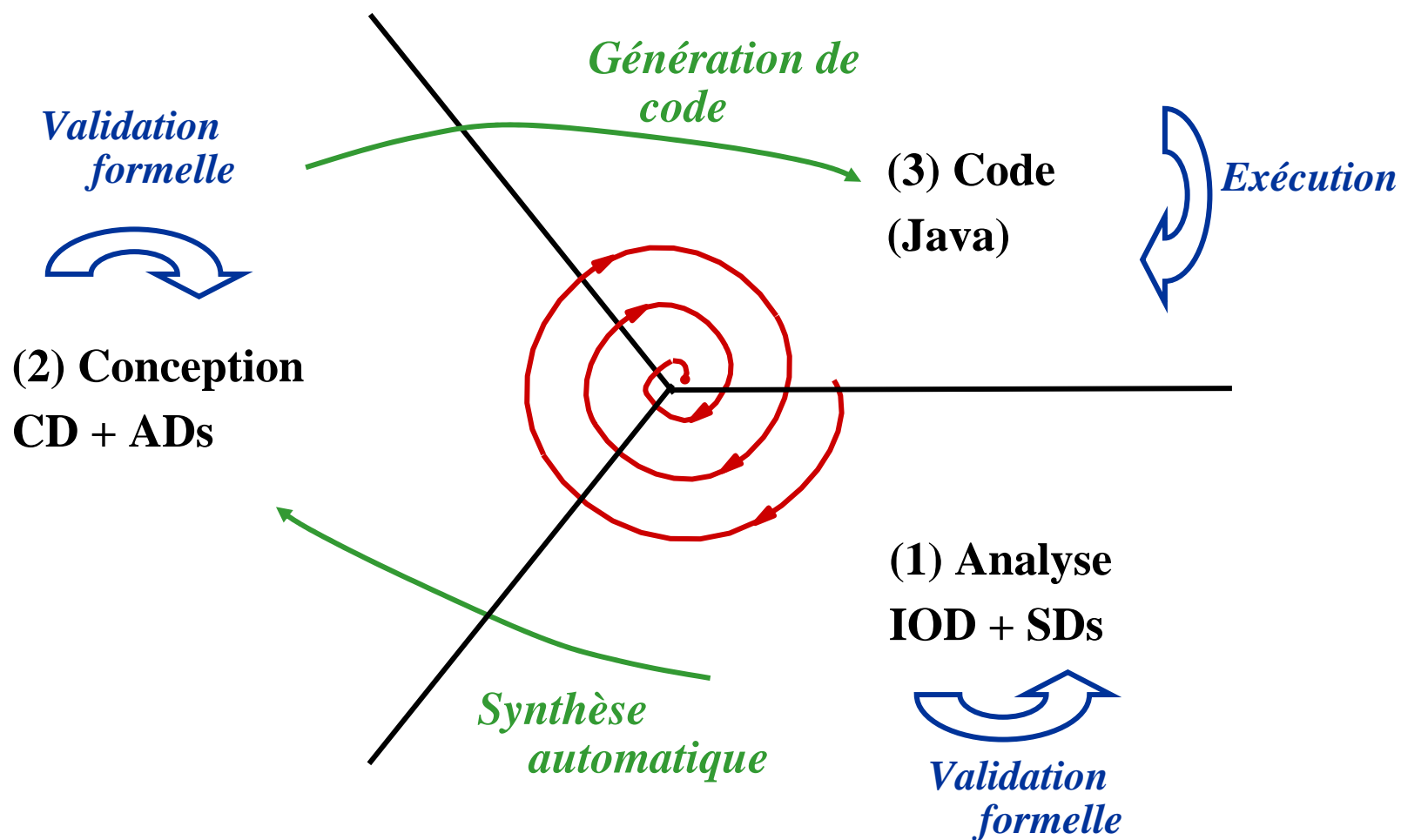
Attributs

Gate

Formule OCL associant des
portes de synchronisation



Approche actuelle



Vers une nouvelle extension

☞ Limites de l'approche actuelle

- ❑ *Applications conçues de façon monolithique*

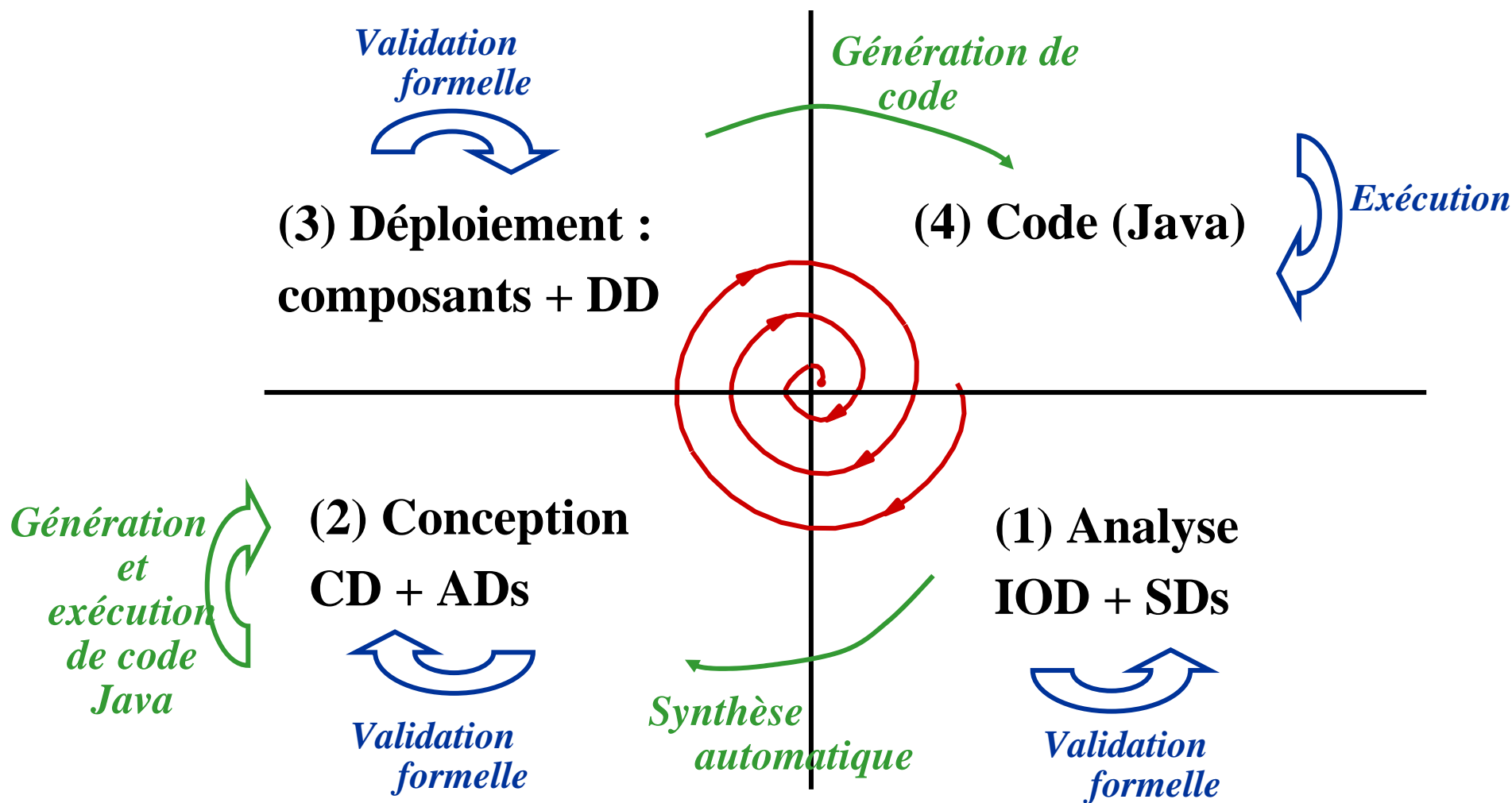
☞ Conséquences

- ❑ *Distribution de l'application non validée formellement*
- ❑ *Code Java monolithique*
 - Non exécutable dans un contexte distribué

☞ Contributions

- ❑ *Nouvelle méthodologie*
- ❑ *Sémantique formelle aux diagrammes de déploiement UML 2*
 - Notion de composants, notion de noeuds d'exécution
- ❑ *Génération de code qui tient compte de la distribution*
- ❑ *Mise à jour des outils supports*

Propositions : Méthodologie



Formalisation des diagrammes de déploiement UML 2

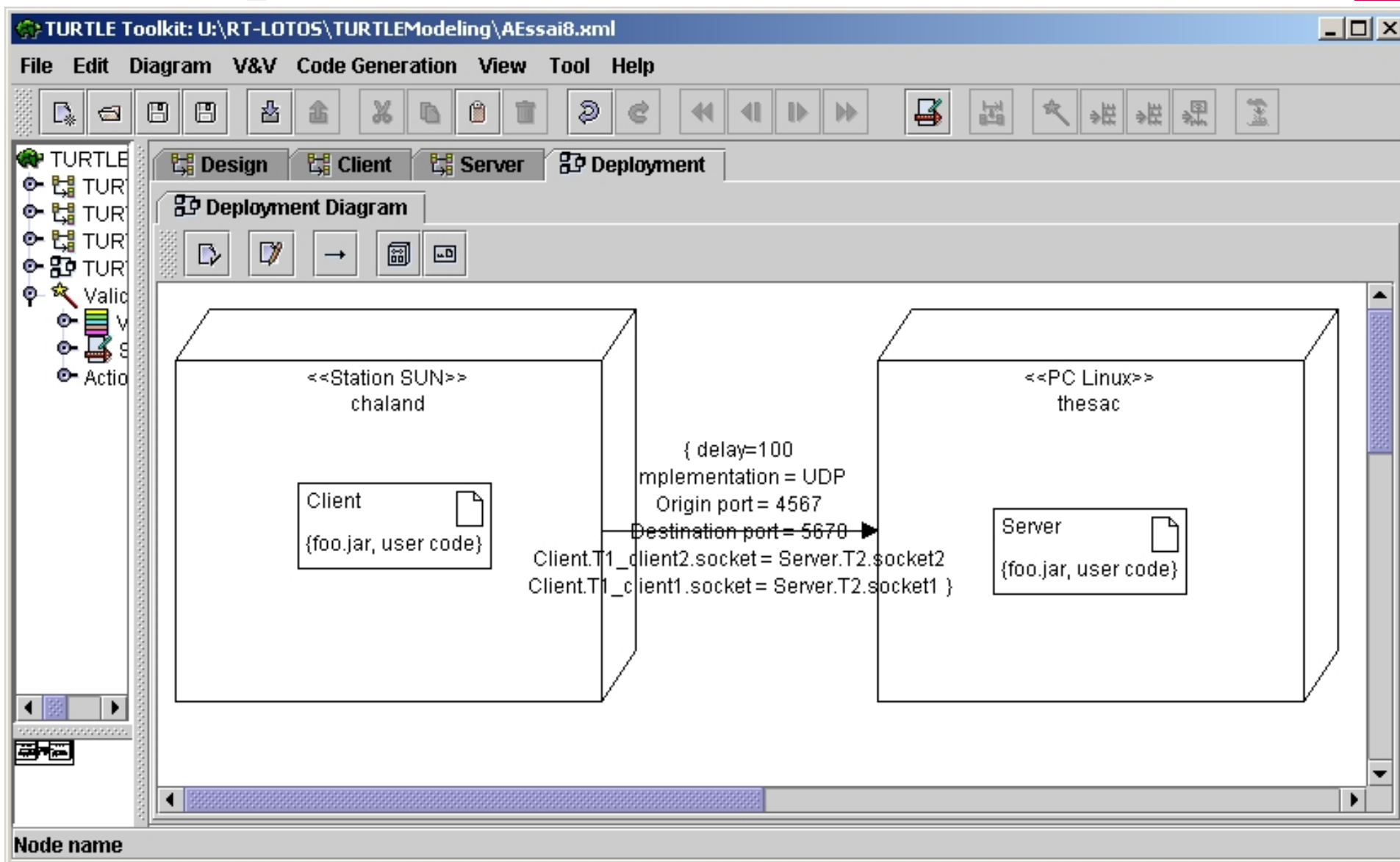
☞ Notion de composant TURTLE

- ☐ *Regroupement de classes TURTLE et des relations relatives à ces classes TURTLE*
- ☐ *Éléments internes d'un composant indissociables*

☞ Déploiement de composants TURTLE

- ☐ *Notion de nœud d'exécution = « node » en UML 2*
- ☐ *Instance d'un composant TURTLE sur un nœud d'exécution TURTLE = artéfact TURTLE*
- ☐ *Liens de communication*
 - Information d'interconnexion de portes
 - Délai, taux de perte
 - Information pour la génération de code « réseau »

Exemple



Sémantique formelle

☰ Diagramme de déploiement + description des composants -> conception TURTLE

☰ Classes des composants

☐ *Ajoutées à la conception TURTLE (renommage)*

☰ Liens asynchrones

☐ *Chaque lien est modélisé par une classe TURTLE ajoutée à la conception*

- Délai, taux de perte
- Acheminement des messages

☐ *Relation de synchronisation entre les classes des composants et les liens*

Génération de code Java (1/2)

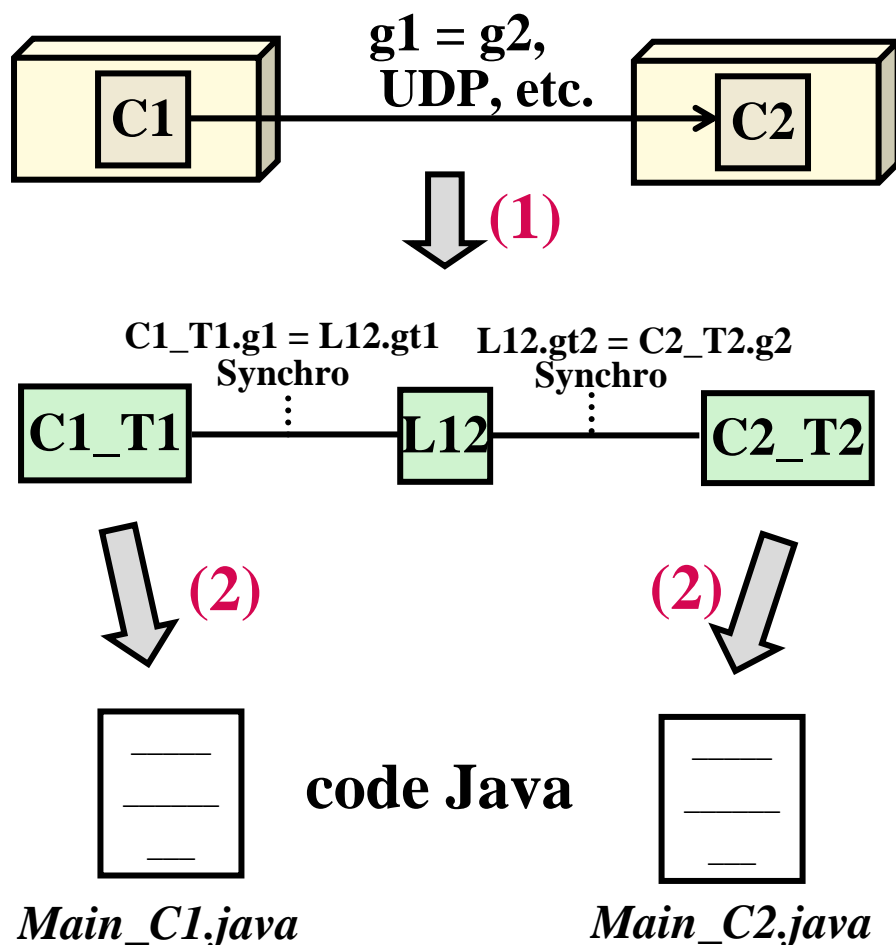
📄 Générateur de code Java pour les conceptions TURTLE

- ❑ *Une classe Java par classe TURTLE*
- ❑ *Une classe principale*
- ❑ *Synchronisations, opérateurs temporels, etc. sont implémentés sous forme de bibliothèques Java*

📄 Générateur de code pour un déploiement

- ❑ *Réutilisation du générateur pour la conception*
- ❑ *Problème des liens*
 - **Modélisés à des fins de validation formelle et de génération de code**
 - **Certains paramètres ne peuvent être pris en considération**
 - *Délai, taux de perte*
 - **Certains paramètres doivent être traités : protocoles utilisés, numéro de ports, adresses réseaux**

Génération de code Java (2/2)



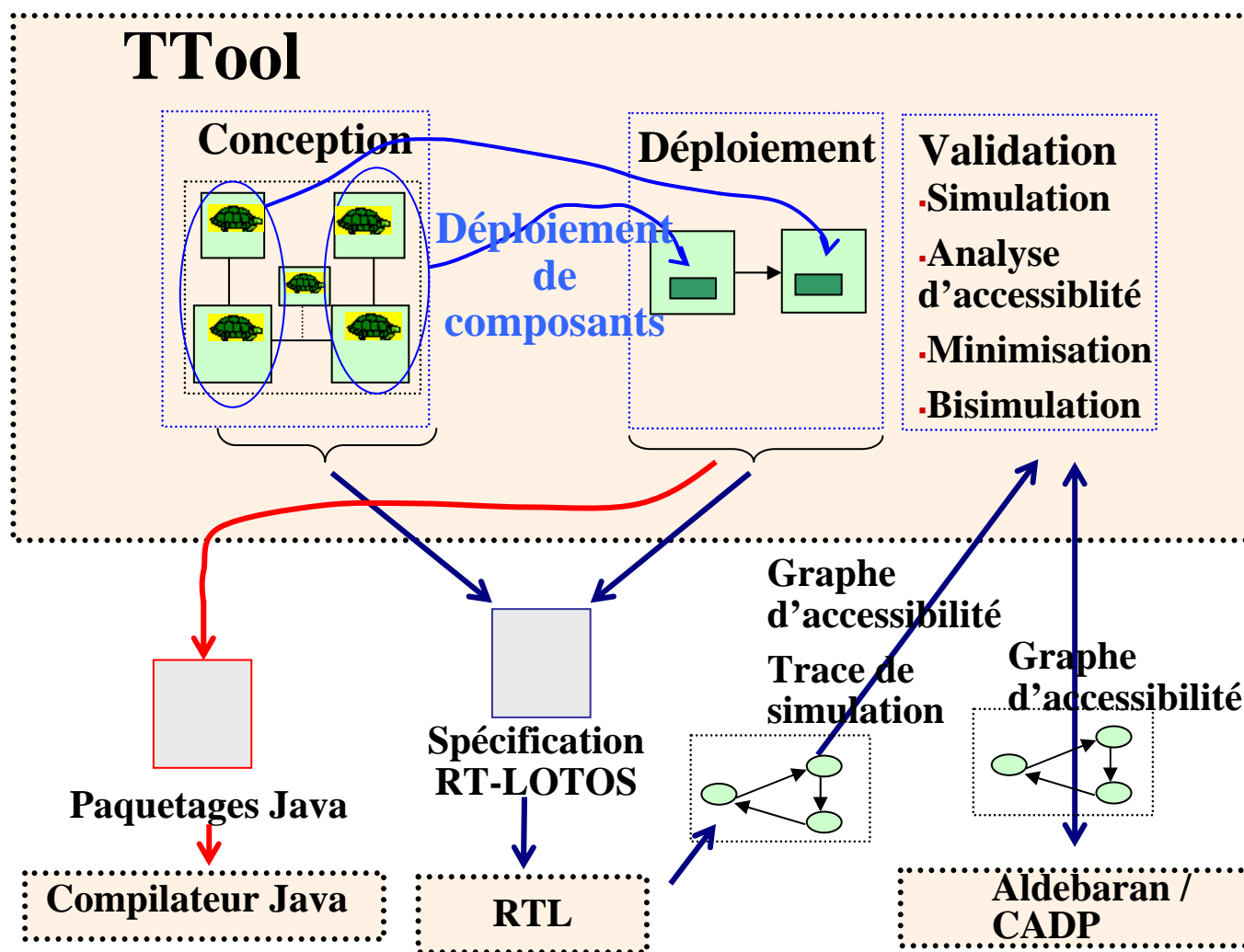
(1) Génération d'une conception TURTLE annotée

- *Portes UDP / TCP / UDP + paramètres associés*

(2) Générateur de code Java pour les conceptions TURTLE

- *Générateur « étendu »*
 - Les portes annotées « réseau » sont traitées différemment
 - Plusieurs modules logiciels exécutables sont générés
- *Librairies qui gèrent les actions sur les portes*
 - Émission, réception de données via UDP, TCP, RMI

Outillage



Conclusion

📄 Profil TURTLE

- ❑ *Arrivé à maturité*
- ❑ *Expériences académiques et industrielles*

📄 Enrichissement du profil

- ❑ *Phase de déploiement*
- ❑ *Sémantique formelle*
- ❑ *Génération de code « réseau »*

📄 Outillage

- ❑ *TTool + RTL*

📄 Travaux futurs

- ❑ *Conception TURTLE à la UML 2.0*
- ❑ *Génération de séquences de test*
- ❑ *TTool téléchargeable bientôt gratuitement !*
 - <http://www.eurecom.fr/~apvrille/TURTLE>

Questions ?

📄 Article NOTERE'2005

- ❑ *L. Apvrille, P. de Saqui-Sannes, “Une méthodologie de conception des systèmes distribués basée sur UML”, Nouvelles Technologies de la répartition (NOTERE'05), Gatineau, Québec, Canada, Aout-Septembre 2005.*

📄 Autres références

- ❑ *L. Apvrille, P. de Saqui-Sannes, F. Khendek, “Synthèse d'une conception UML temps-réel à partir de diagrammes de séquences”, Colloque Francophone sur l'ingénierie des protocoles (CFIP'05), Bordeaux, France, mars 2005.*
- ❑ *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.*
- ❑ *L. Apvrille, P de Saqui-Sannes, P. Senac , C. Lohr, “Verifying Service Continuity in a Satellite Reconfiguration Procedure”, Journal of Automated Software Engineering , Kluwer, issue 11:2, 2004.*
- ❑ *L. Apvrille, P. de Saqui-Sannes, C. Lohr, P. Sénac, J.-P. Courtiat, “A New UML Profile for Real-time System Formal Design and Validation”, Proceedings of the Fourth International Conference on the Unified Modeling Language (UML'2001), Toronto, Canada, October 2001.*