

Spécification de comportements dynamiques de composants avec STS

Jean-Claude Royer

OBASCO EMN - INRIA

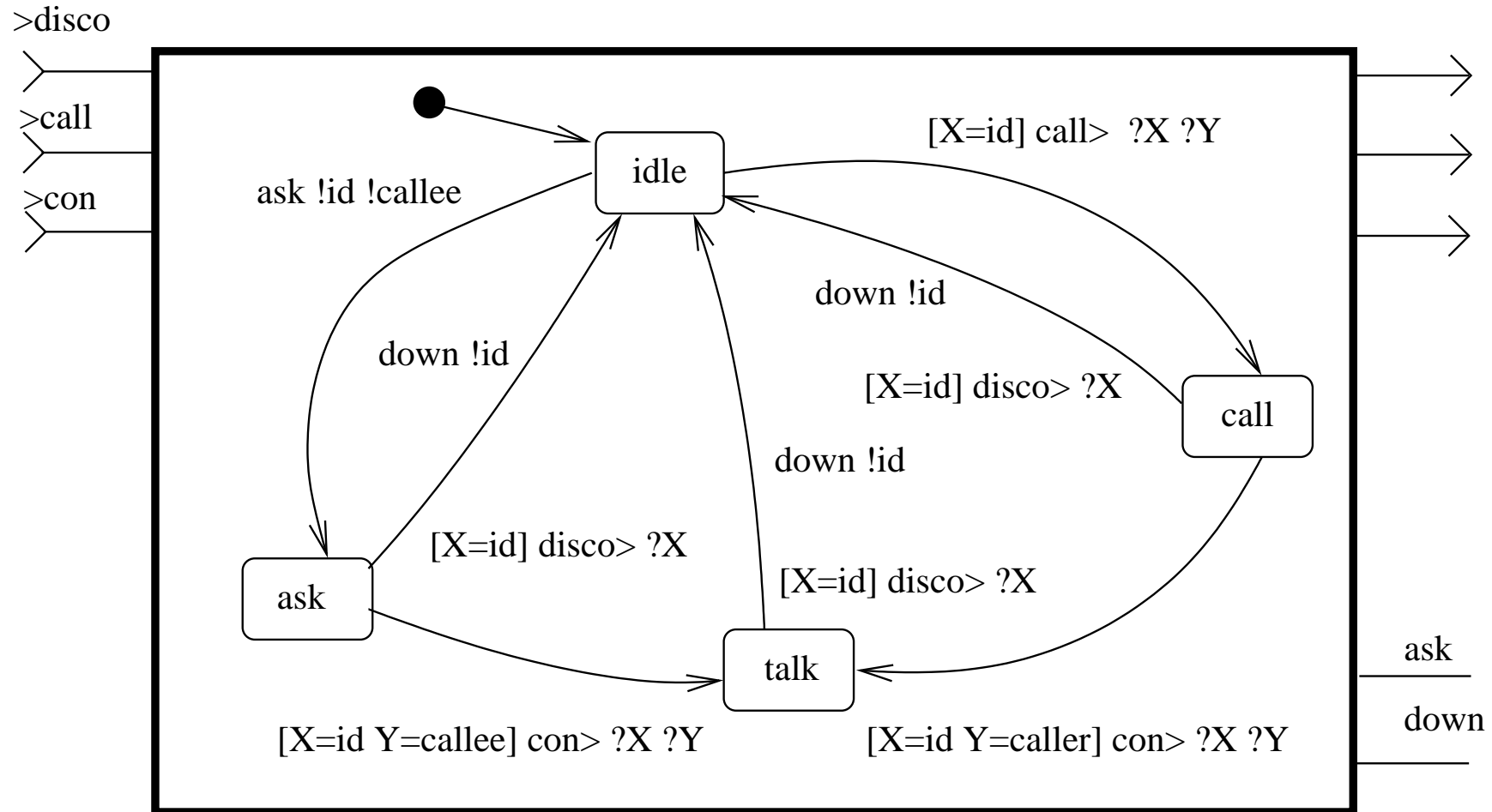
Ecole des Mines de Nantes – France

Les besoins

- Introduction de protocoles explicites dans les composants
- Un formalisme visuel basé sur les machines à nombre d'état fini
- Un moyen de contrôle du problème d'explosion du nombre état/transition
- Communications synchrones et asynchrones
- Expressions de propriétés sur les données sans restriction

- Système de Transition Symbolique
- STS = LTS + variables + gardes
- $\langle S \rangle - [\text{garde}(*)] \text{ label } * \rightarrow \langle T \rangle$
- Plus abstrait et plus puissant que LTS
- Forme compacte d'un LTS (éventuellement infini)
- Par contre vérification spécifique à définir

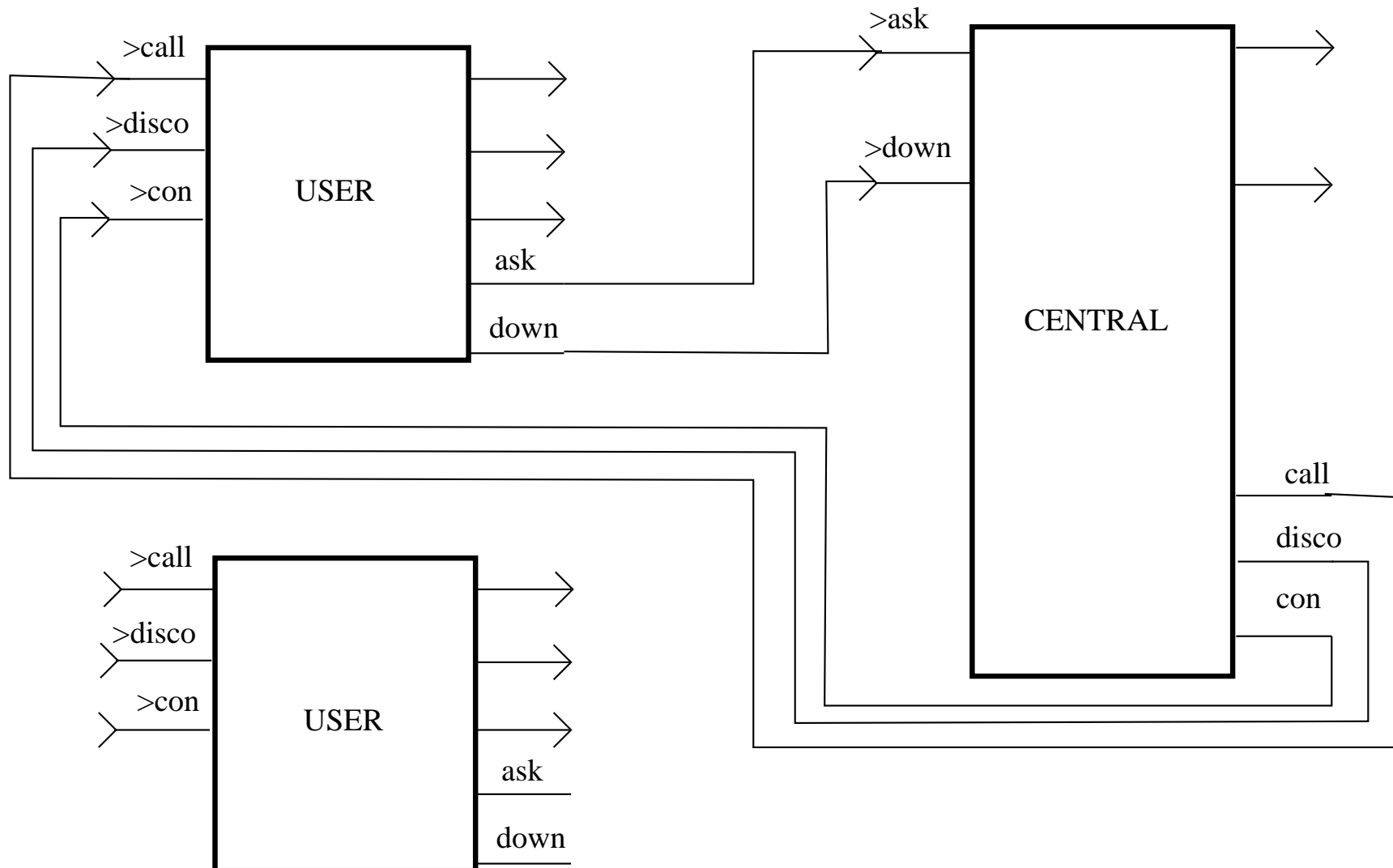
Exemple



id : identité du client

callee : identité de l'appelé • caller : identité de l'appelant

Exemple d'architecture



Travaux récents

- Un langage ADL papier (DOA 2003) : protocole explicite, communication asynchrone, présentation graphique
- Implémentation Java par S. Pavel (en cours)
- Analyses des communication asynchrones
- Spécifications pour les composants et les architectures
- Lien avec la logique temporelle
- Vérifications

Communication asynchrones 1/3

- Introduction : boîte à lettres, réception, exécution
- Modèle dissymétrique car envoi synchrone avec la réception
- Tests d'implémentation avec ProActive
- Autres essais en cours :
 - DEA de Sedkaoui (pur Java)
 - Thèse de Pavel (réflexion, ...)

Communication asynchrones 2/3

- Importance du problème de borne des boîtes à lettres : bonne formation du protocole
- Travail sur des dictionnaires
- Le problème est indécidable avec FIFO
- Décidable pour les dictionnaires
- Une condition suffisante pour les FIFO : analyse avec dictionnaire bornée
- Preuve et applications (DOA 2004)

Communication asynchrones 3/3

- Bibliothèque Java pour les STS (DEA Maréchal, Cissé)
- Algorithmes adaptés pour nos STS
- Optimiser les algorithmes
- Proposer ou adapter d'autres critères utiles (borné ou non borné)
- La littérature est pauvre (Jard/Jeron, Leue/Wei wei)
- Utiliser une approche compositionnelle ...

Spécifications pour composants 1/3

- Un modèle général avec
 - Système de transition symbolique
 - Hiérarchique
 - Communications synchrones et asynchrones
 - Type de données sans restriction
- Travaux précédents sur les spécifications mixtes (Informatica'04)
- Approche avec le prouveur PVS (FMPPTA'02)

Spécifications pour composants 2/3

- Les communications asynchrones complexifient fortement les spécifications
- Une solution (WADT'04) est d'utiliser
 - Une analyse de borne
 - Une extraction et spécialisation des STS des sous composants
 - De régénérer automatiquement une spécification plus simple pour les sous-composants
 - La spécification du composite étant généralement automatique

Lien avec la logique temporelle

- Spécification avec logique des prédicats
- Logique temporelle en introduisant des opérateurs algébriques (next, prefixe)
- Formules de sûreté, de vivacité et d'équité
- On montre extension naturelle de CTL* (IASSE'04)
- Permet d'écrire des formules CTL* avec des données et de les prouver *via* PVS

Vérifications 1/2

- Trois approches possibles : vérification de modèles, prouveur général, algorithmes spécialisés
- Les STS sont compatibles, en partie, avec la vérification de modèles
- Problèmes voisins de la vérification de modèles des systèmes infinis
- Approche générale de preuve en PVS, essai de stratégie
- Algorithmes spécifiques (exemple de la “bornitude”)

Vérifications pour composants 2/2

- CTL* : $AG EX true$
- Utilisation de la vérification de modèles
- Logique du premier ordre :

$$\forall self : TI, \exists *, D_{TI}(self) \Rightarrow \bigvee_{op_R} (precond_{op_R}(self, *))$$

- Stratégies de preuves
- Formule des arcs :

$$P_s(self) \wedge G(self, X) \supset \bigcup_t G_t(op_t(self, X))$$

Perspectives

- Poursuivre le niveau langage extension de Java
- Préciser la notion de disponibilité par l'étude de l'allocateur de ressource de Gligor/Yu
- Expressions de propriétés de disponibilité
- Etudier des moyens pour les vérifier
- +++ coherence avec autres, cadre general