



Application MBDA : calcul d'une translation entre deux images

Présentation de l'application MBDA

- Objectifs
- Problématique
- Besoins et solutions de modélisation retenues
- Architecture logicielle
- Architecture matérielle
- Travaux à finaliser
- Evolutions à prévoir

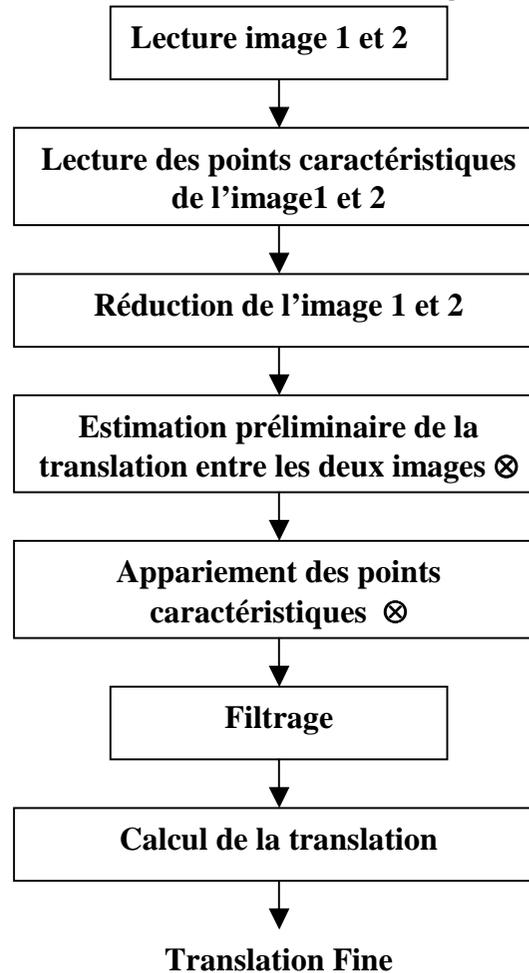
Application MBDA : Objectifs

Modélisation en langage UML d'une application de traitement d'images intégrant les notions de co-design : développement conjoint matériel/logiciel

- Cette phase de modélisation doit se faire en deux étapes
 - **Modélisation de l'architecture logicielle**
 - **Modélisation de l'architecture matérielle (multi-composant hétérogènes)**

- **Validation des modèles**

Application MBDA : Principe



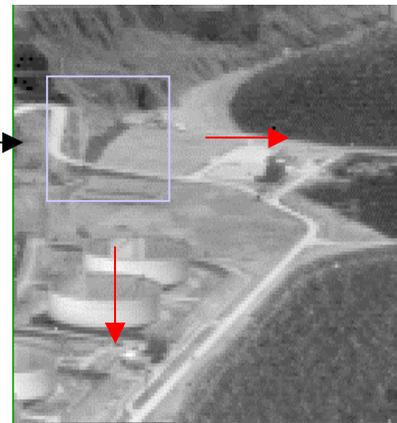
Application MBDA : Corrélation et appariement



image originale à t



Noyau de corrélation



$$\text{Cost}(i,j) = \sum_{(k,l) \in \text{noyau}} | \text{noyau}(k,l) - \text{image}(k+i,l+j) |$$



position optimale du noyau à t+dt



2 images successives à t et t+dt



appariement des points caractéristiques



nouvelle estimation du point

Application MBDA : Problématique

Spécification du parallélisme potentiel d'opérations et de données dans le modèle

- Niveau de détail (granularité)
- Traitement de la répétition d'opération (factorisation)
- Traitement des données (tableaux, record, ...)
- Efficacité et lisibilité du modèle (hiérarchie)
- Contraintes relatives aux outils utilisés : gestion des tableaux, des répétitions

Application MBDA : différents choix de modélisation

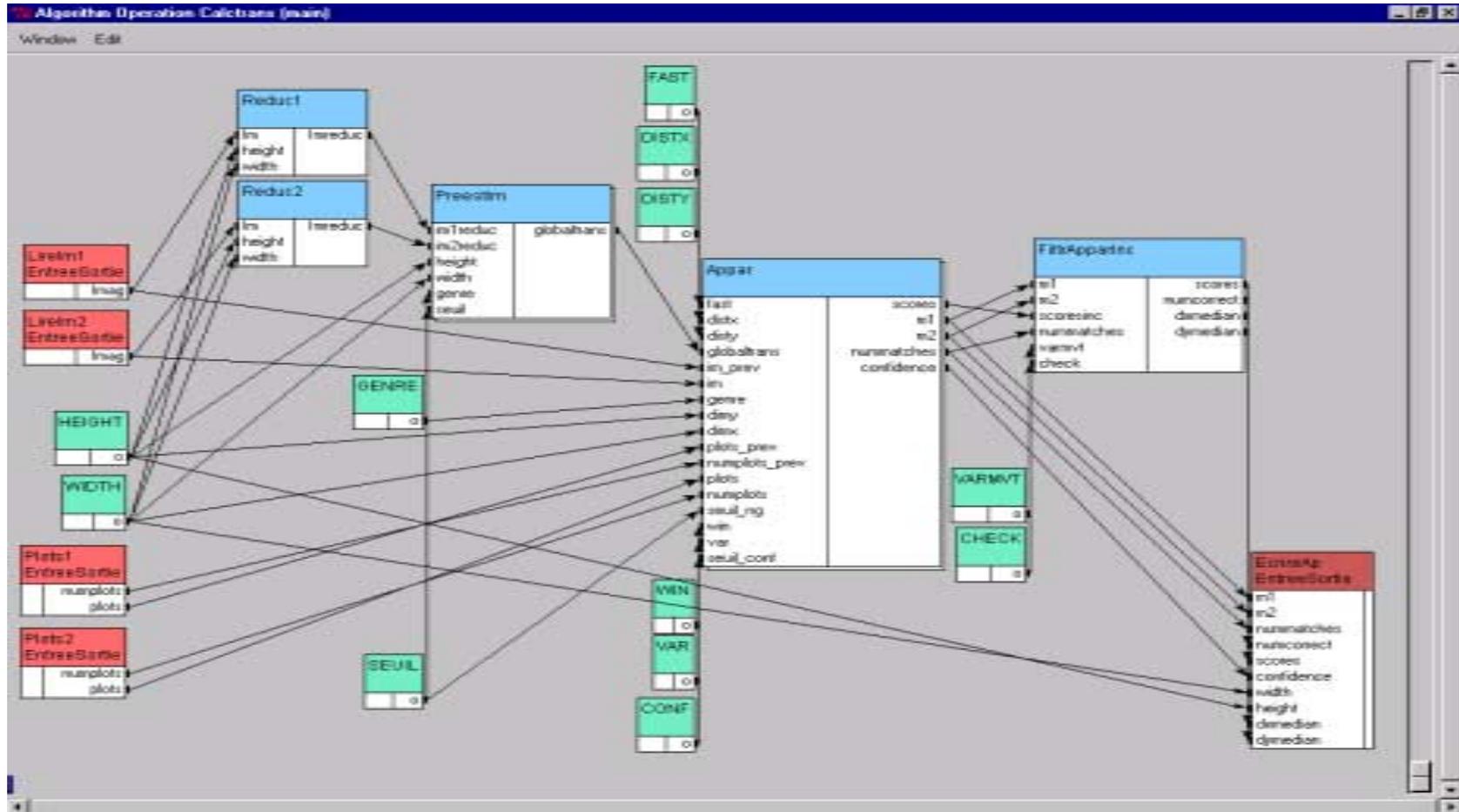
- Granularité
 - Classe générale : formalisme objet
 - Paquetages
 - Objets composites
- Contraintes temporelles (deadline, période) sur
 - Diagrammes de séquence
 - Diagrammes d'activités
- Répétitions
 - Autant d'objets que de répétitions nécessaires
 - Autant de services d'une classe que de répétitions nécessaires
- Types de données
 - Regroupement des tableaux dans un paquetage stéréotypé
- Efficacité & lisibilité
 - Utilisation de paquetages
 - Modèle à plat

Image
Height Width NbrePlots ...
lecture() réduction() corrélation() extracNoyaux()

Application MBDA : Besoins et solutions de modélisation

- Niveau de granularité : le niveau de détails augmente en fonction du parallélisme potentiel d'opération que l'on veut introduire
 - **Choix du modèle de référence SynDEX comme base de la modélisation: rétro-ingénierie**
 - **Réciprocité Classe \leftrightarrow Opération**
- Répétition des opérations :
 - **Utilisation de la multiplicité des associations entre classes : le nombre d'instances (objets) possibles est relatif au nombre de répétitions**
 - **Gestion des « split » (fork) et « merge » (joint)**
- Gestion des données :
 - **Utilisation du paquetage « Domain Types » permettant la manipulation des tableaux de façon discrète : transparence pour le code SIGNAL**
- Lisibilité
 - **Utilisation des « classes englobantes » et des paquetages : fonctionnement par appel de services répondant au principe de modularité du langage SIGNAL**
- Relation avec les interfaces extérieures au système:
 - **Communication par signaux (spécification de communication entre objets qui transporte des informations ou pas dans le but de déclencher une activité → mail box)**

Application MBDA : Modèle de référence SynDEX

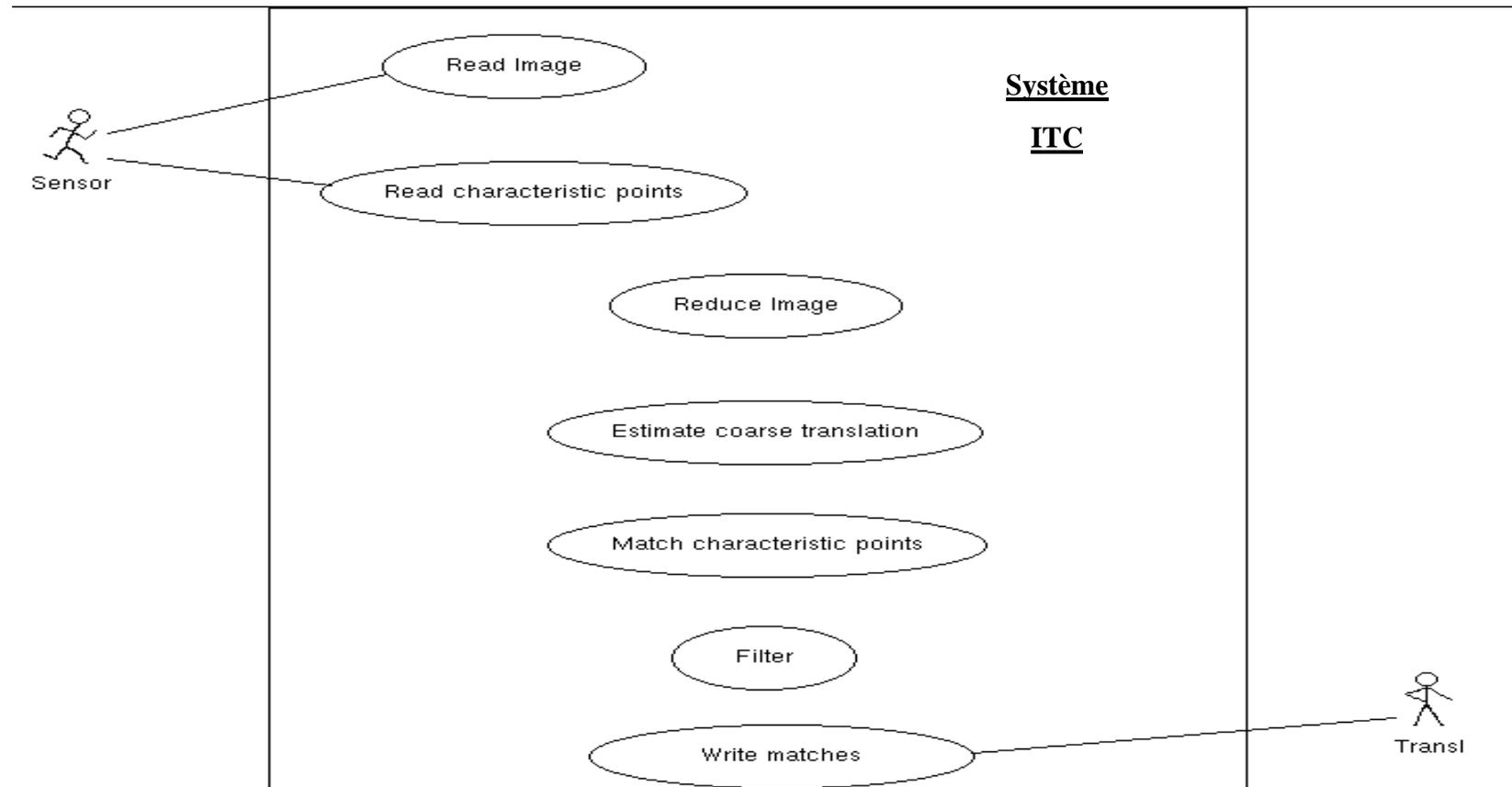


Application MBDA : Architecture logicielle

- Méthodologie ACCORD UML : conception du modèle d'analyse
 - Etape 1 : PAM (Preliminary Analysis Modelling) spécifier l'ensemble des fonctions de l'application ainsi que les interactions du système avec l'extérieur
 - ➔ ce que fait le système
 - Utilisation des diagrammes de cas d'utilisation et des diagrammes de séquence principalement.

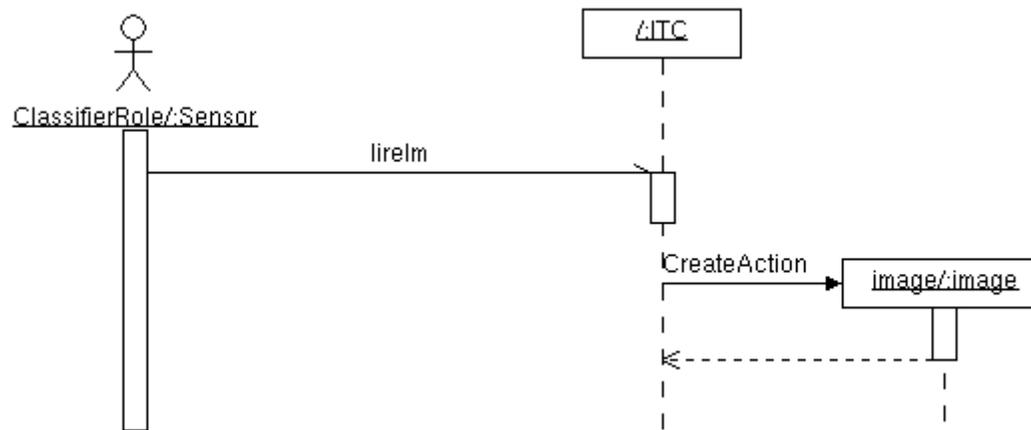
Application MBDA : Architecture logicielle PAM

■ USE CASE (diagramme de cas d'utilisation)



Application MBDA : Architecture logicielle PAM

- Diagramme de séquence : Read Image

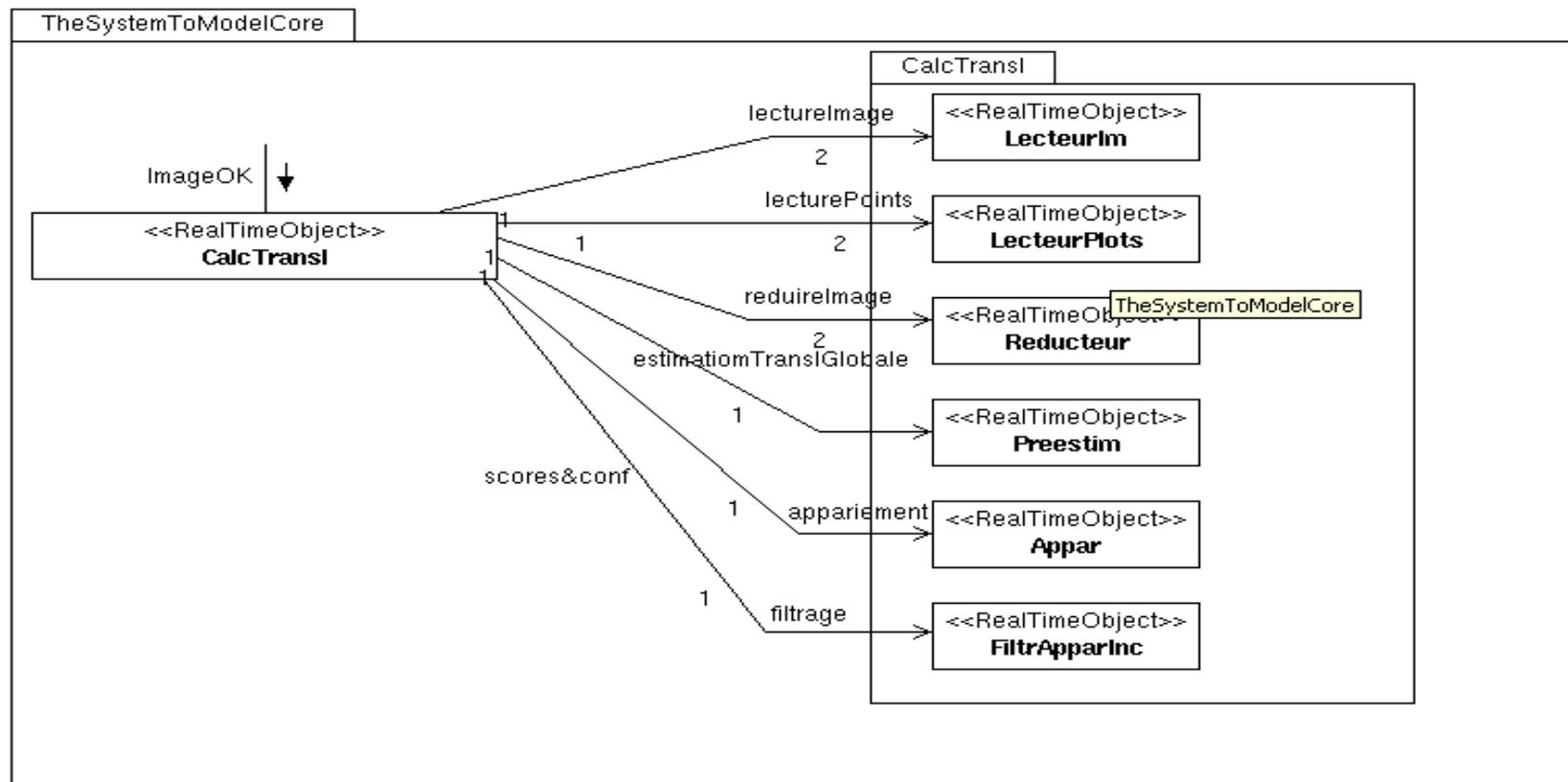


Application MBDA : Architecture logicielle

- Méthodologie ACCORD UML : conception du modèle d'analyse
 - Etape 2 : DAM (Detailed Analysis Modelling) décrire en détails les spécifications et les fonctions de l'application décrites préalablement dans le PAM → comment le système fonctionne.
 - Utilisation des diagrammes de classe, d'état-transition et d'activité.

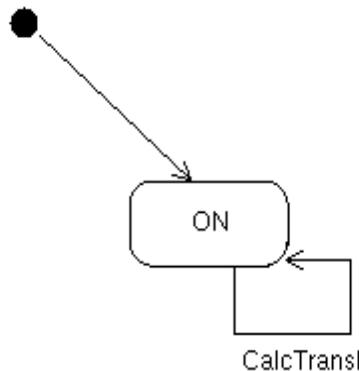
Application MBDA : Architecture logicielle DAM

- Diagramme de classe : classe englobante CalcTransl

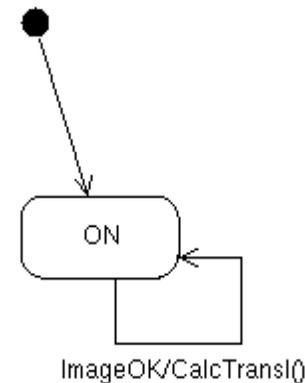


Application MBDA : Architecture logicielle DAM

- Diagramme d'état-transition
 - Deux vues complémentaires pour modéliser :
 - Réception d'un événement
 - Déclenchement du service associé
 - Tous les services de toutes les classes sont associées à ces diagrammes



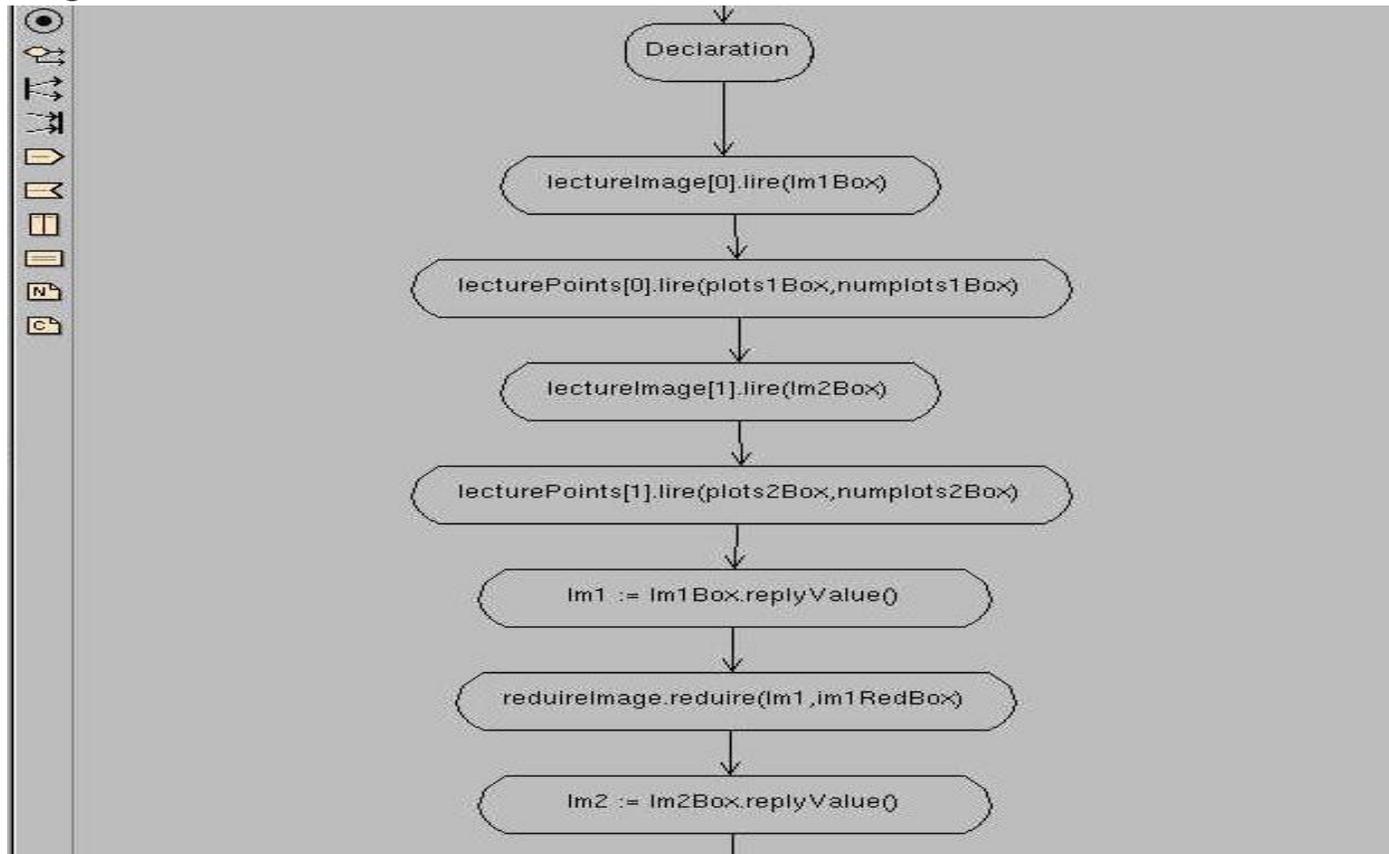
State diagram : CalcTransl protocol view



State diagram : CalcTransl triggering view

Application MBDA : Architecture logicielle DAM

■ Diagramme d'activité

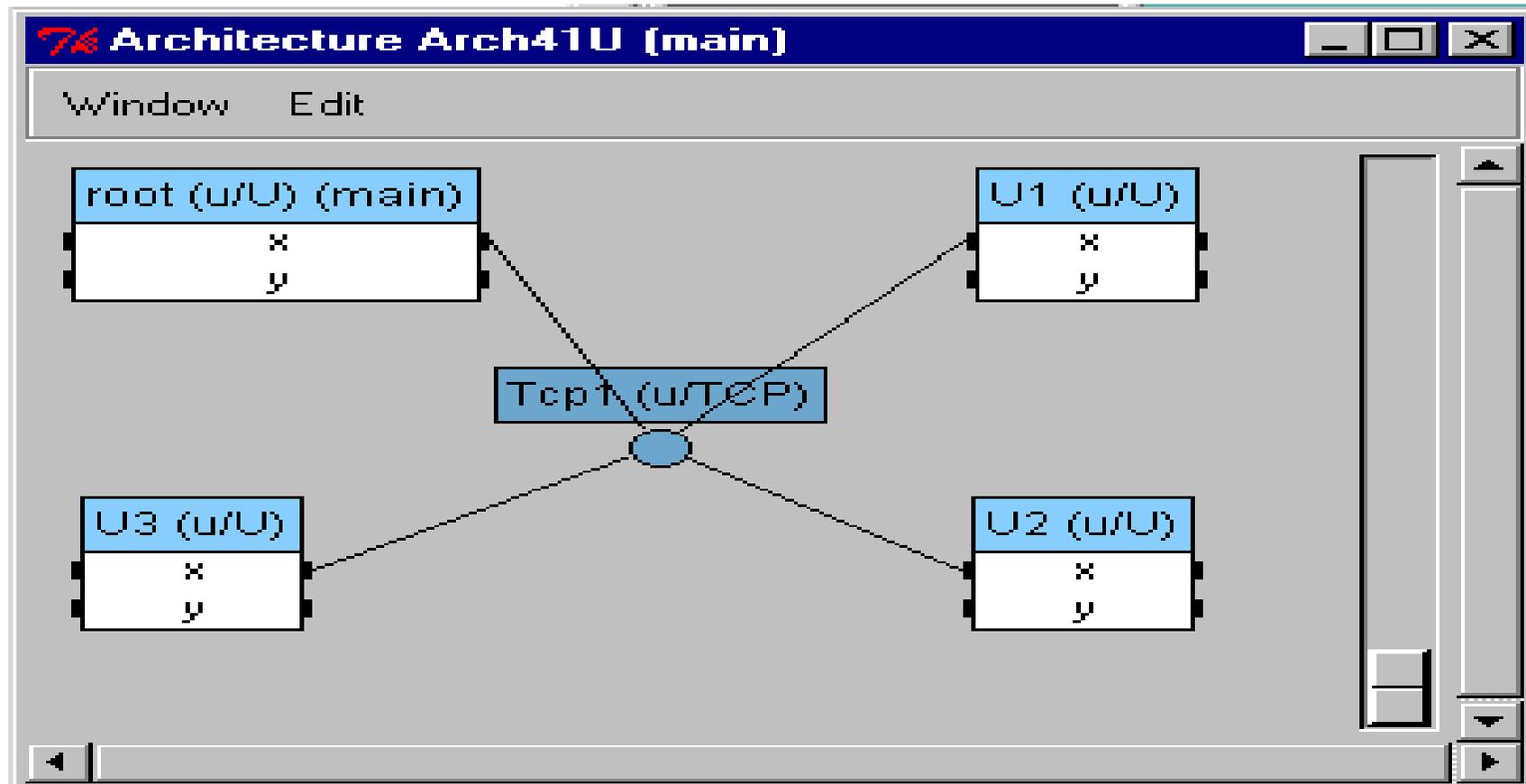


Application MBDA : Architecture matérielle

- Méthodologie ACCORD UML : conception du modèle d'architecture
 - Modélisation d'instances d'architecture prenant en compte les nœuds (ressources de calcul) et les médias (Bus ou liens de communication)
 - Différents types d'architecture modélisés :
 - Point à point en anneau
 - point à point entièrement connecté
 - réseau point à point en anneau avec bus de communication
- Utilisation des diagrammes de déploiement

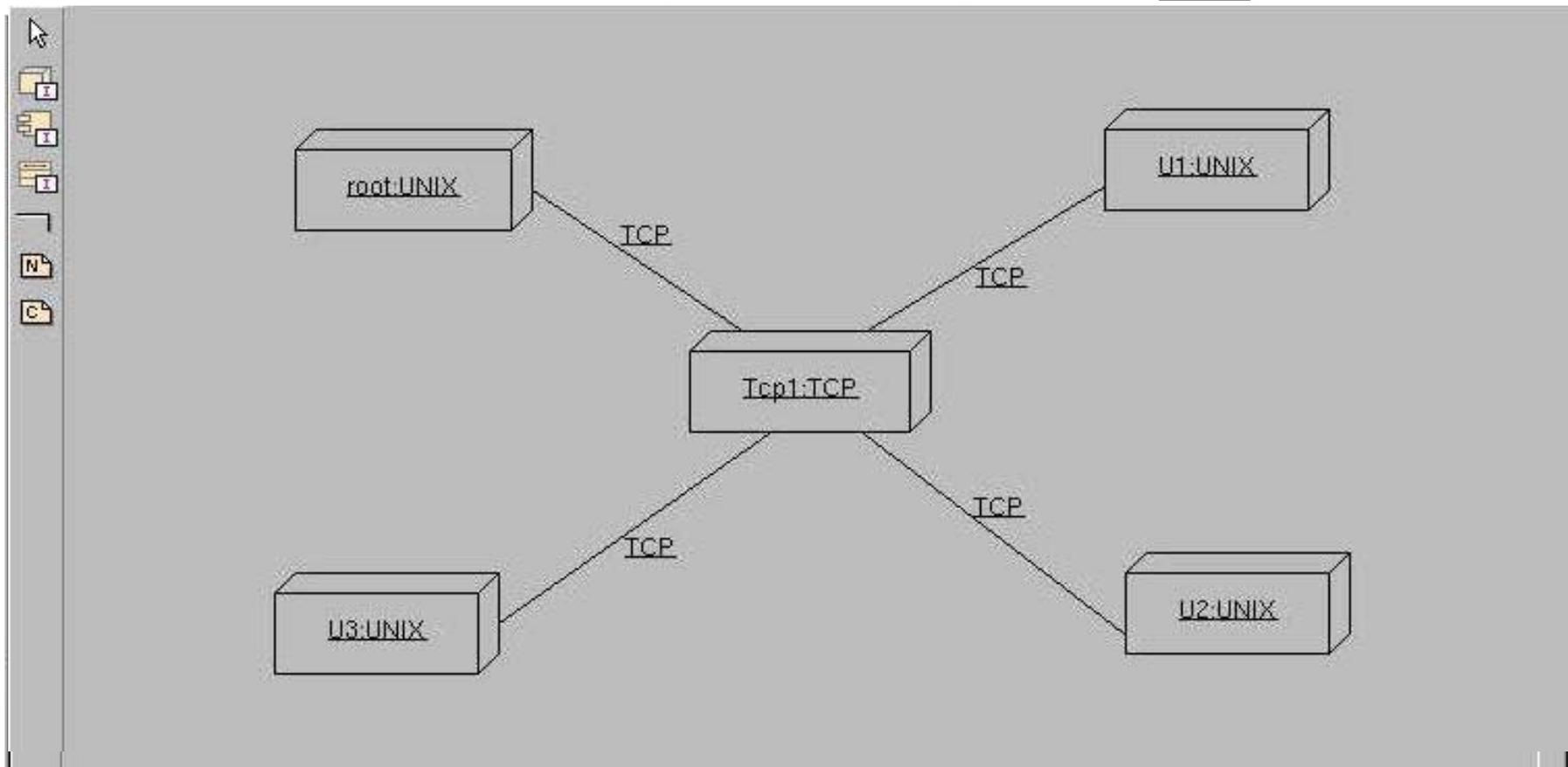
Application MBDA : Architecture matérielle

- Réseau entièrement connecté (4 stations UNIX) : modèle de référence **SynDEx**



Application MBDA : Architecture matérielle

- Réseau entièrement connecté (4 stations UNIX) : modèle UML



Travaux à finaliser

- Phase de validation des modèles
 - Chaîne algorithmique
 - Génération et validation du code SIGNAL
 - Validation du modèle sur l'outil SynDEX
 - Chaîne architecture
 - Validation de l'architecture d'implantation « réseau entièrement connecté »
 - Adéquation Algorithme Architecture
 - Génération de code exécutable pour stations Unix

Evolution à prévoir

- Évolution du modèle algorithmique vers un formalisme objet en reprenant les concepts de
 - **Tableaux**
 - **Composition**
 - **Prises en compte des contraintes temps réel et du conditionnement**

- **Modèle objet plus cohérent**