

Un modèle pour l'interaction entre composants logiciels



Philippe ANIORTE

LIUPPA - IUT Bayonne

Equipe ISIHM

aniorte@iutbayonne.univ-pau.fr

Plan de la présentation



⌘ 1. Introduction

⌘ 2. Problématique (générale) de nos travaux

⌘ 3. Solutions apportées en terme de modèle d'interaction

1. Introduction - IS-IHM



⌘ IS : Ingénierie de Systèmes (d'Information)

⌘ IHM : Interface Homme-Machine

⌘ IS-IHM : dénominateur commun =
interaction

1. IS-IHM - Actions



- ⌘ Projets européens ASIMIL & ELeGI
- ⌘ Colloque ALCAA (Agents Logiciels, Coopération, Apprentissage, Activité humaine)
- ⌘ GT MFI ...

1. GT MFI



- ⌘ Problématique de nos travaux
- ⌘ Solutions apportées
- ⌘ Mise en perspective de ces solutions et des travaux du groupe (au cours et/ou après la présentation)

2. Problématique - Objectifs des travaux



- ⌘ Apporter des solutions aux nouveaux besoins des entreprises en matière d'ingénierie de systèmes
- ⌘ Supportées par des réseaux
- ⌘ Intégration d'entités distribuées et hétérogènes

2. Domaines de recherche impliqués



⌘ Réutilisation d'entités « a posteriori »

⌘ Interopérabilité entre entités distribuées hétérogènes

⌘ (Méta)modélisation

2. Éléments de solution

- ⌘ (Méta)modèles : architecture de métamodélisation à 4 niveaux [OMG], modèles pour la réutilisation ...
- ⌘ Outils : middlewares [CORBA] [EJB], langages (ADL) ...
- ⌘ Processus : démarche MDA [OMG] ...

2. Discussion



- ⌘ Les différentes solutions ciblent des problèmes spécifiques
- ⌘ Traités à différents niveaux d'abstraction
- ⌘ Pas de solution globale adaptée au problème considéré

3. Solutions - Plan



⌘ 3.1. Introduction

⌘ 3.2. Présentation du métamodèle

⌘ 3.3. Utilisation du métamodèle

3.1. Introduction - Concepts de base



- ⌘ Composant : entité logicielle de haut niveau d'abstraction autonome à l'exécution, potentiellement distribuée et hétérogène
- ⌘ Interaction : dispositif permettant à plusieurs composants d'interagir (interopérer)

3.1. Éléments essentiels de nos propositions

- ⌘ Un métamodèle de composants basé sur l'architecture de métamodélisation à 4 niveaux = le MOF métamodèle « Ugatze »
- ⌘ Un processus guidé par les modèles
- ⌘ Un ensemble d'outils liés au processus et au métamodèle

3.1. Le processus

- ⌘ Etape 1 : (re)spécification des composants (gestion de la réutilisation)
- ⌘ Etape 2 : intégration des composants basée sur les interactions (architecture logicielle du système)
- ⌘ Etape 3 : implantation du système

3.2. Le métamodèle Ugatze - Principes



- ⌘ Haut niveau de découplage entre composants
- ⌘ Abstraction
- ⌘ Séparation donnée / contrôle

3.2. Ugatze - Organisation



⌘ Le point de vue « Component »

⌘ Le point de vue « Interaction »

⌘ Le point de vue « Run time environment »

3.2. Ugatze : : Component Package (1/3)

- ⌘ Etape de spécification des composants
- ⌘ Chaque composant est « emballé » dans une interface définie par un ensemble de points d'interaction permettant sa réutilisation
- ⌘ Points d'interaction : E/S données/contrôle

3.2. Ugatze : : Component Package (2/3)

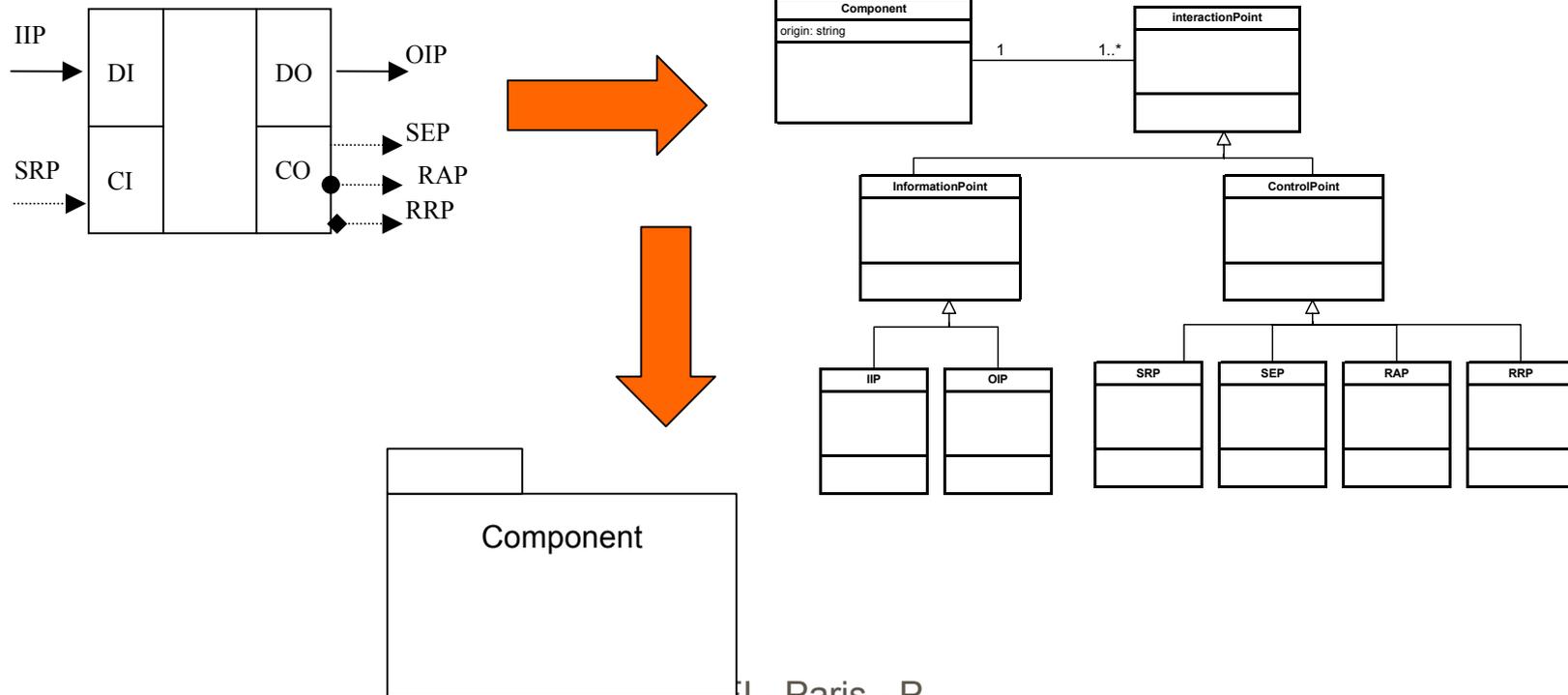


⌘ Représentation graphique

⌘ Règles de bonne formation (« well-formedness rules » MDA) exprimées en OCL (Object Constraint Language)

3.2. Ugatez : : Component Package (3/3)

Le MOF Component Package



3.2. Points d'interaction



⌘ Point d'opération (synchrone)

⌘ Point de flot continu (« streaming »)

⌘ Point asynchrone

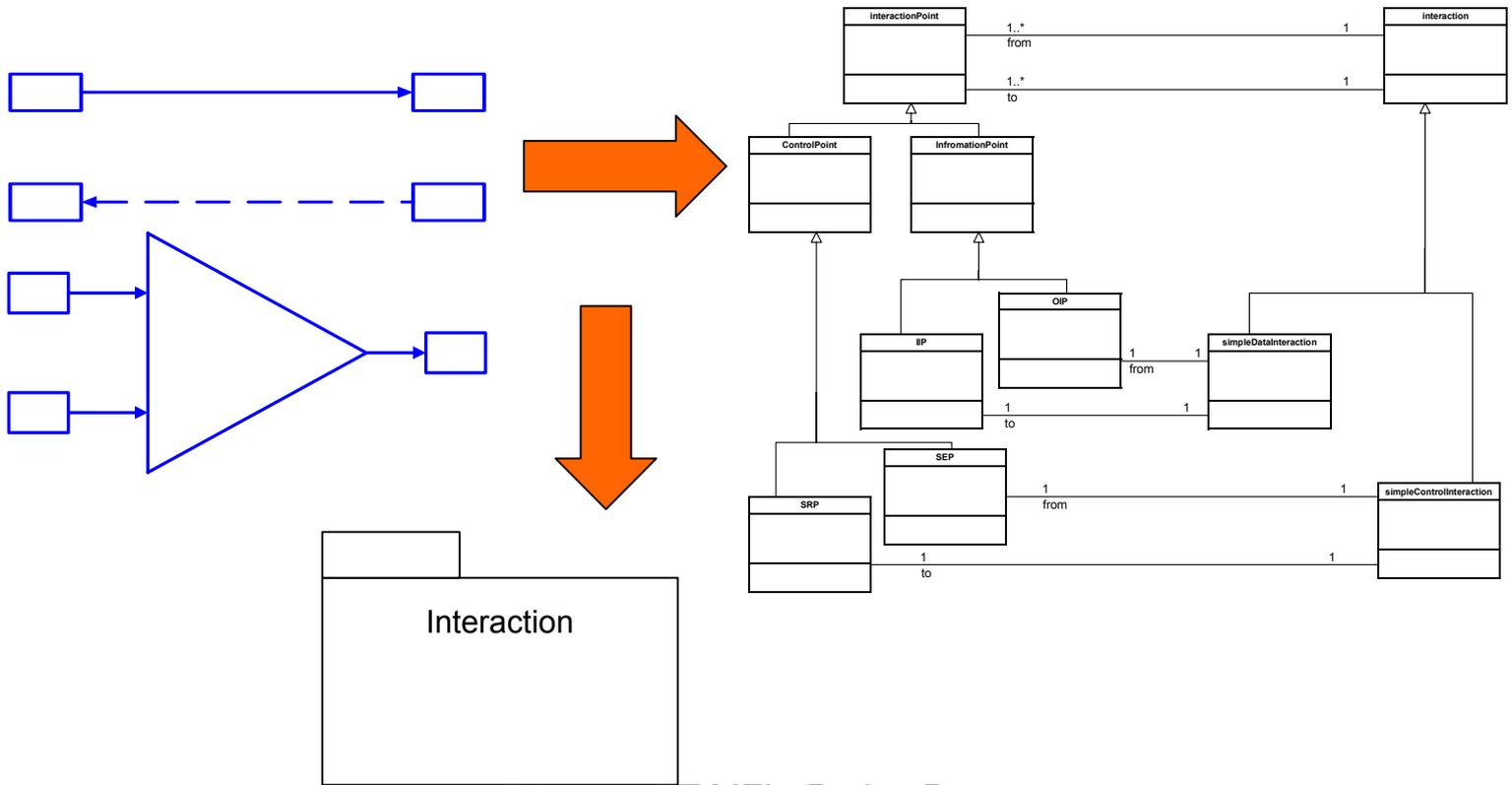
3.2. Ugatze : : Interaction Package (1/2)



- ⌘ Etape d'intégration des composants
- ⌘ Gestion de l'interopérabilité conceptuelle
- ⌘ Représentation graphique / règles OCL

3.2. Ugate : : Interaction Package (2/2)

⌘ Le MOF Interaction Package



3.2. Interactions

- ⌘ Une interaction est une relation entre 1 ou plusieurs points d'interaction d'entrée et 1 ou plusieurs points d'interaction de sortie
- ⌘ Interactions prédéfinies (dans le paquetage)
- ⌘ Interactions « à façon » conçues par l'architecte

3.3. Mise en œuvre d'Ugatze

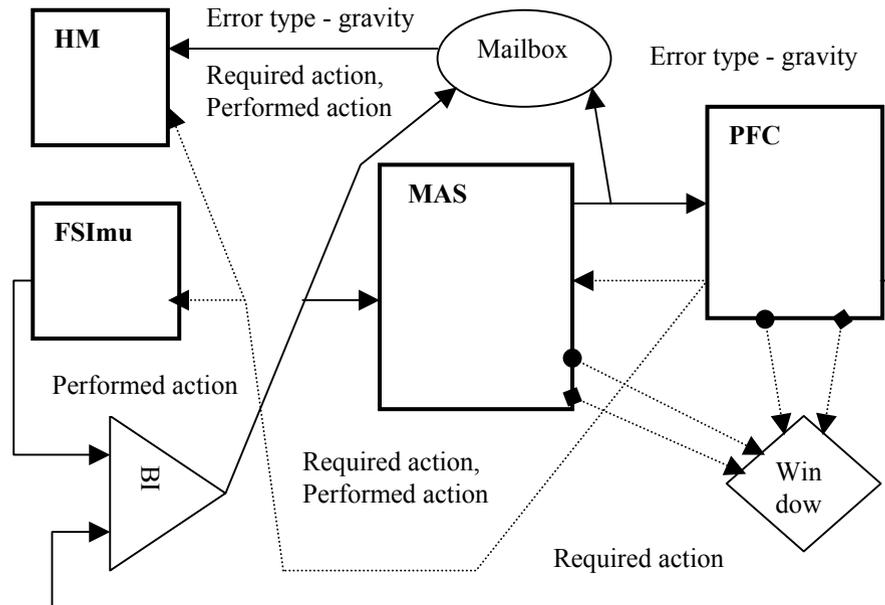
- ⌘ Spécification des composants (Component Package) & intégration (Interaction Package)
- ⌘ Instanciation du métamodèle pour obtenir le modèle d'application du système (PIM MDA)
- ⌘ Résultat = graphe d'interaction

3.3. Illustration - Le projet ASIMIL



- ⌘ Aero user-friendly SIMulation-based dIstance Learning
- ⌘ Différents composants distribués hétérogènes
- ⌘ Nécessité d'intégrer ces composants

3.3. Le graphe d'interaction d'ASIMIL



3.3. Exploitation du graphe

- ⌘ Après avoir été validé (règles OCL)
- ⌘ Il constitue le point de départ de l'étape d'implantation du système

3.3. Transformations du graphe



- ⌘ Utilisation du Run time environment Package
- ⌘ Gestion de l'interopérabilité technique
- ⌘ Sous-graphes d'interaction (PSM MDA) & interactions entre sous-graphes

Conclusion - Bilan



- ⌘ Un (méta)Modèle pour l'Interaction (MI)
- ⌘ Entre composants logiciels
- ⌘ Un processus de développement guidé par les modèles, inspiré de la démarche MDA

Conclusion - Perspectives



⌘ Points d'interaction & interactions ?

⌘ Adéquation / adaptation pour des Grid Services (projet européen ELeGI) ?

⌘ Formalisation (F de MFI) ?