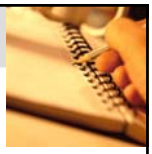
Composition fiable d'architectures orientées services et composants : le projet FAROS

Philippe Collet

Equipe MODALIS

27 janvier 2009

Plan



- Thématiques du groupe COSMAL
- FAROS : présentation générale
- FAROS : procédé
- FAROS : métamodèles « *inside* »
- FAROS : bilan et perspectives
- Retour sur la problématique et sur les trois sessions à suivre

Thématique générale du groupe COSMAL

- **Objets, Composants Services**
  - Abstraire par des modèles
  - Utiliser des langages
  - Proposer des architectures
- **Permettre ou faciliter la construction**
  - économiquement compétitive
  - de logiciels fiables
  - moins coûteux à maintenir et à modifier
  - et de plus en plus adaptables
  - même dynamiquement, à leur contexte d'utilisation

Thèmes affichés

- Modélisation et de la conception d'architectures logicielles
- **Composition d'entités logicielles (composant, service et modèle)**
  - **Journée Composition : Mettre en exergue les différents défis de la composition dans l'élaboration du logiciel**
- Représentation des connaissances
- Techniques de compilation et typage adaptées
- Mise en œuvre de la réutilisation des entités logicielles (atelier de développement, ingénierie)
- Utilisation d'une approche contractuelle pour la garantie de propriétés
- Séparation de préoccupations

FAROS : identité

❑ **composition de contrats pour la Fiabilité d'ARchitectures Orientées Services**

❑ Appel RNTL 2005

❑ Exploratoire : 3 ans
- 2006-2008
- Prolongé de 9 mois jusqu'en septembre 2009

❑Industriels	❑Académiques	❑PME
EDF France Telecom	IRISA ❑ Equipe Triskell i3S ❑ Equipe Rainbow LIFL ❑ Equipe Goal	Alicante (sous traitant)

❑ http://www.lifl.fr/faros/

Constat

❑ **Essor de SOA pour**
- offrir rapidement de nouveaux services,
- intégrer divers SI…

❑ **Supports d'exécution variés**
- Architecture/infrastructure (ESB, SCA…)
- Env. contraints/embarqués

❑ **Garanties (QdS…) pour**
- différencier les fournisseurs
- négocier (exigences / offres de services), établir les responsabilités…

Les solutions actuelles d'intégration de services

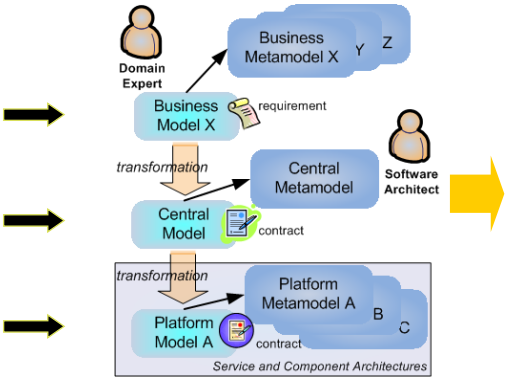
❑ manquent de moyens pour exprimer/ vérifier des garanties dans les compositions de services

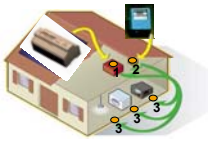
❑ manquent de guides méthodologiques pour automatiser la prise en compte de contrats depuis le besoin métier jusqu'aux plateformes d'exécution

FAROS : Objectifs

- **Construire un outil pour concevoir des applications**
  - par composition de services
  - avec des notations faisant abstraction des technologies de service
  - rendues fiables par l'emploi de contrats
- **Démontrer l'approche sur des applications et des plateformes cibles variées (SOA/CBSE avec ou sans support contractuel)**
- **Eviter les solutions ad-hoc : factoriser les transformations**

FAROS : Procédé



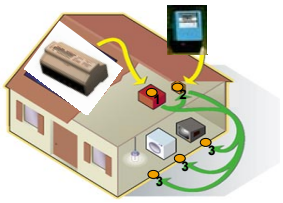


- **Métiers**
  - Diffusion d'informations (SEDUITE, DMP)
  - Compteur intelligent (CIRE/EDF)
- **Plateformes (SOA/CBSE + contrats/aspects…)**
  - **Adore**, **WComp**, **ConFract**, AoKell, ORQOS
- **Transformation de modèles (Kermeta) avec utilisation d'un niveau pivot**

Etudes de cas : 3 applications

SEDUITE : Système de diffusion d'informations en milieu scolaire (une architecture orientée services en exploitation)

DMP : Accès au dossier médical personnel d'un patient, dans un contexte d'urgence et/ou dans un contexte mobile

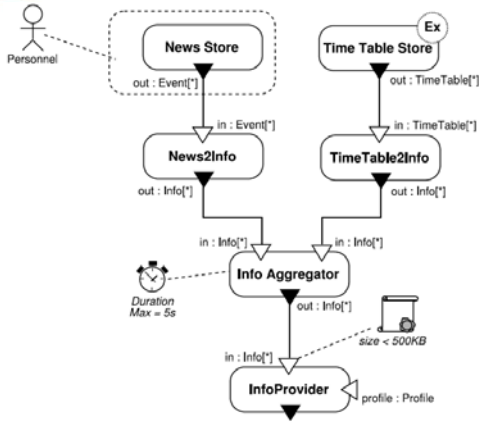


CIRÉ : Boîtier intelligent en aval du compteur électrique offrant des services de maîtrise de la demande énergétique

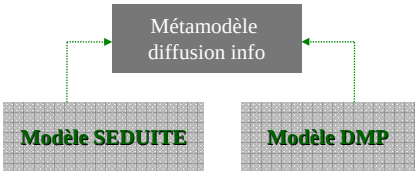
Métamodèles métiers

- **Les différents métamodèles métiers**
  - servent à présenter des concepts spécialisés du domaine des concepteurs métier,
  - pour plus de facilité de conception,
  - à la façon des langages spécifiques de domaine
- **Ils permettent de garantir le respect de contraintes métier quel que soit le modèle produit**
- **Ils permettent de réutiliser des briques par réassemblage et donc de capitaliser le savoir-faire**
- **Démarche**
  - Une approche par métamodèle (profils UML)
  - Une approche par transformation (comportement)
  - Fusion des applications Séduite/DMP : un seul métier
  - Une mise en œuvre avec Eclipse/Ecore

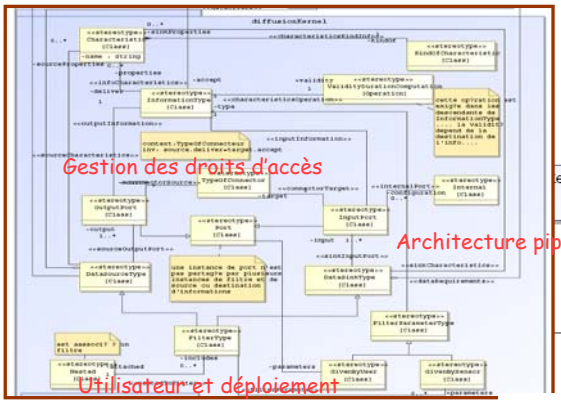
Analyse du métier : SEDUITE et DMP

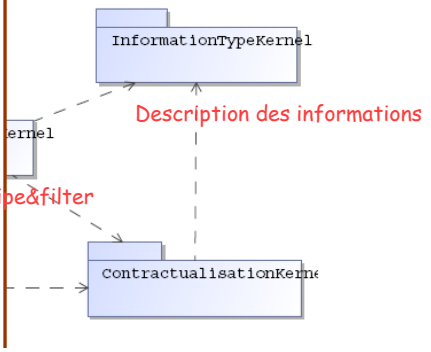


- **Diffusion d'information :**
  - ≈ architecture Pipe & Filter, DataSource, Connector, Port



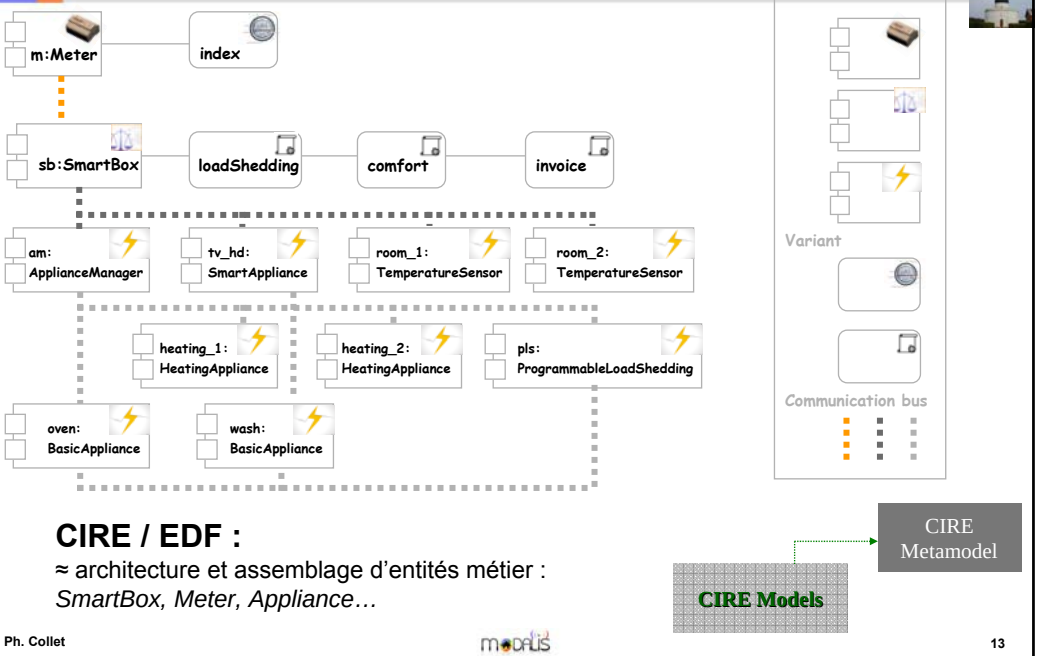
Métamodèle résultant





- Modélisation des sources, cibles et filtres
- Une communication à travers des ports
- Des informations typées

Métier et métamodèle CIRE



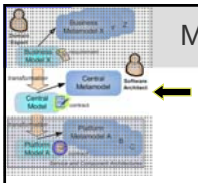
CIRE / EDF :

≈ architecture et assemblage d'entités métier :
SmartBox, Meter, Appliance…

CIRE Metamodel

CIRE Models

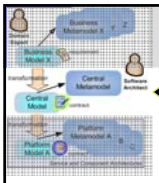
Métamodèle pivot



- **Objectifs :**
  - Découpler méta-modèles métier et plate-forme
    - Approche similaire au code intermédiaire dans les compilateurs
  - Permettre la réutilisation de transformations

- **Fonctionnalités :**
  - Capacité à conserver l'information structurée de bout en bout
  - Transformation de bout en bout de structures particulières
    - Coordination (orchestration), dimensions de qualité de service

- **Démarche :**
  - Limiter le nombre de concepts (cibler l'utilisation du métamodèle)
  - Maîtriser la complexité des transformations



Métamodèle pivot (suite)

❑ **Contrat de service :**

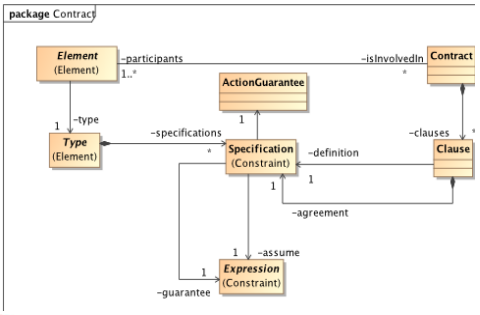
- Modèle ouvert (Participant, Hypothèse, Garantie, etc.)
- Types (structurel, comportemental, extra-fonctionnel)
- Expression des propriétés contractuelles

Métamodèle pivot

Modèle pivot

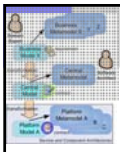
❑ **Structure et comportement :**

- Réification service / composant possible
- Focus sur l'aspect compositionnel
  - services primitifs vs composites
  - Composite ≈ orchestration de services



❑ **Modèle d'exécution :**

- Modèle événementiel extensible



Métamodèles des plateformes d'exécution

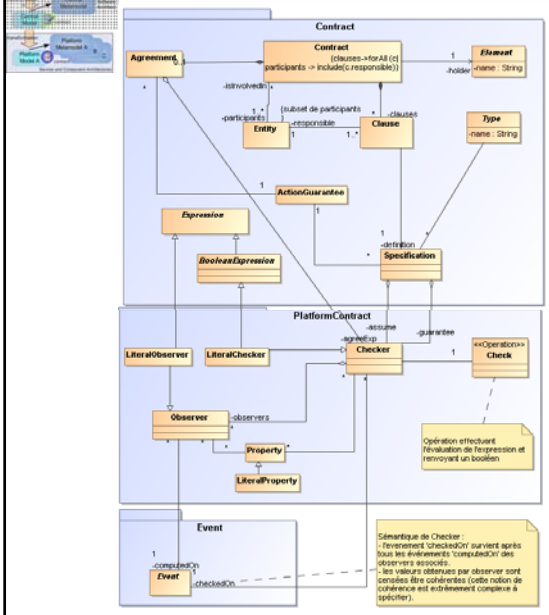
Plates-formes cibles :

- ❑ ADORE (I3S) : tissage sur BPEL
- ❑ WComp (I3S) : aspects d'assemblage sur composants événementiels
- ❑ ConFract (I3S, FT R&D) : langage d'assertions sur composants hiérarchiques
- ❑ AOKell-FAC (LIFL) : aspects sur composants hiérarchiques
- ❑ ORQOS (FT R&D) : transformations de BPEL

❑ **Démarche :**

- 8 patrons de contrats types (en relation avec les 3 applications de validation)
- un modèle événementiel commun pour la définition de ces patrons
- Des schémas de mise en œuvre de ces patrons sur les 5 plates-formes

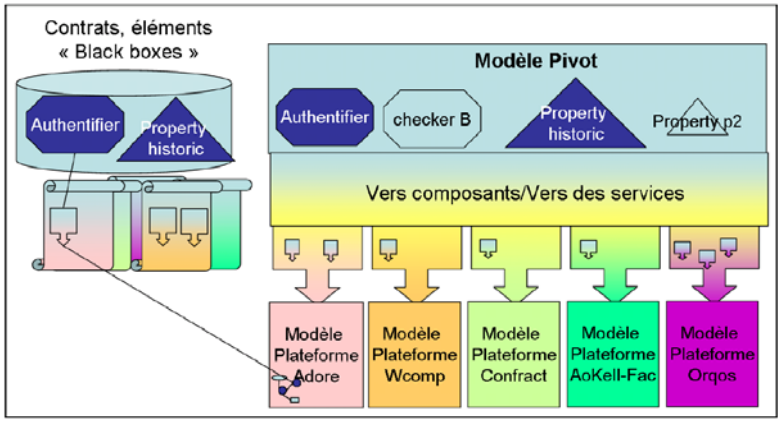
Métamodèles des plateformes d'exécution (2)



- **Vision globale : extension du MM pivot**
  - Définition des événements sur lesquels les contrats sont observés et validés dans les PF
  - Définition des instants d'observation et de validation - par des exemples de cycle de vie
  - Définition d'un langage de surface et d'un modèle de contrat pour la mise en œuvre dans les PF

Plateforme : transformation et intégration

- Nécessité d'intégrer des mécanismes pour supporter les transformations
  - tout n'est pas fait dans les transformations
- Avantage : la plateforme n'est pas une simple VM
  - elle contient ses propres mécanismes qui permettent la manipulation et l'exploitation de ce qui a été généré par transformation



Transformations : rôles et mise en œuvre

- **Le mécanisme de transformation retenu joue plusieurs rôles**
  - **Séparation des préoccupations**
  - **Mécanismes de contrôle**
    - Modèle flot de données (Séduite/DMP) : injection/configuration d'un ordonnanceur parmi n possibilités
    - Modèle type Web services : injection/configuration d'une orchestration
  - **Mécanismes de validation**
    - Validation statique dans l'outil IDM
    - Injection/synthèse d'un sous-ensemble d'observation (observateur boîte blanche)
    - Paramétrage d'un observateur plate-forme (observateur boîte noire)
- **Les transformations s'appuient sur le concept de métamodèle exécutable**
  - Organisation sous forme de phases structurées
  - Code de transformation dans des mixins
  - Traçabilité des actions de transformation faites

Kerméta

Bilan

- **Résultats et technologies transférables**
  - **Processus de développement d'applications (métiers/pivot/plateformes)**
    - Rapprochement des mondes composant et service (pivot : vision unifiée)
    - Intégration de contrats sur tout le cycle de développement
  - **Mise en œuvre du procédé intégrée dans Eclipse/EMF + langage Kermeta**
    - Projections vers un ensemble de plates-formes à composants et à services
    - Extension : support des contrats pour plusieurs d'entre elles
- **Difficultés**
  - Définition de MM Métier, syntaxe de surface, outillage au niveau modeleur
  - Définition et rôle de MM Pivot
  - Nombre de plates-formes visées

Perspectives

❑ Coté métier

- Promouvoir la réutilisation des (méta-)modèles, *off-the-shelf*
- Gestion de la *variabilité* des modèles (contexte, propriétés extra-fonctionnelles, capacités de composition) et opérationnalisation dans des *lignes de produits*

❑ Coté plateforme

- Architecture autonomique pour services et composants à grande échelle
- Réutilisation et adaptation des parties contractuelles / monitoring
- Développement d'une boucle de rétro-action réutilisant notamment ces informations contractualisées

Journée *Composition*

❑ Mettre en exergue les différents défis de la composition dans l'élaboration du logiciel

❑ Quid du projet FAROS ?

- **Apports**
  - Applications composites dans le sens général (composants, services)
  - IDM pour maîtriser « statiquement » et « dynamiquement »
- **Limites, perspectives**
  - on ne fait pas de SCA, ni de SxxZVq (qui va surement sortir en fin d'année…)
  - on n'intègre pas de formalismes/langages pour la partie contractuelle

❑ Et finalement, des problèmes…

- **De composition « statique »**
- **De composition « dynamique »**
- **D'architectures, langages et API**

Problématiques du groupe : zoom...

❑ Composition « statique »

- Younes Lakhrissi, Iulian Ober, Bernard Coulette. **Spécification et composition du comportement dans le profil VUML.**
- O.Caron, B.Carré, A.Muller, G.Vanwormhoudt. **Adaptation fonctionnelle de composants gros-grain avec JBOSS/AOP.**
- Pascal André, Gilles Ardourel, Christian Attiogbé, Arnaud Lanoix, Mohamed Messabihi. **Prise en compte d'assertions pour la correction d'assemblages de composants Kmelia.**

❑ Composition « dynamique »

- Ismael Bpuassida Rodriguez. **Gestion des architectures dynamiques pour l'adaptabilité des applications distribuées coopératives. Application au provisionnement de la qualité de service.**
- Riadh BEN HALIMA. **Spécification, conception et réalisation d'applications coopératives distribuées autoréparables à base de Services Web.**
- Anthony HOCK-KOON, Mourad OUSSALAH. **Composition Dynamique de Services dans les Environnements d'Intelligence Ambiante.**

❑ Architectures : langages et/ou API ?

- Zawar Qayyum, Flavio Oquendo. **Supporting Architectural Composition at Runtime with π-ADL.NET**.
- Francoise Baude, Virginie Legrand-Contes. **Exécution distribuée et agile de compositions de services.**
- Luc Fabresse. **Des modèles aux programmes à base de composants : Besoin de langages à composants.**