PARSEC

Les apports de l'IDM à la sécurité et à la sûreté de fonctionnement

Une application à la radio logicielle

Paris, le 7 juin 2012

Les systèmes critiques sont fortement contraints

- **Exigences de Sûreté de Fonctionnement**
  - Fiabilité, Disponibilité, Maintenabilité, Sécurité-Innocuité, Intégrité, Confidentialité
- **Exigences de Certification**
  - Pour autoriser la mise sur le marché du système
  - Les objectifs de certification dépendent du domaine applicatif
- **Contraintes d'intégration système**
  - Partage des ressources matérielles entre plusieurs applications
  - Exemple des architectures IMA dans le domaine avionique
    - Une architecture matérielle, fourni par l'intégrateur, pour plusieurs applications
    - Isolation spatiale et temporelle entre niveaux de criticité différents (ARINC653)
- **Pression économique sur les coûts**
  - Besoin de minimiser les coûts sans nuire à la fiabilité du système
  - Coûts d'utilisation et de maintenance en condition opérationnelle
  - Contraintes de réutilisation de composants déjà certifiés

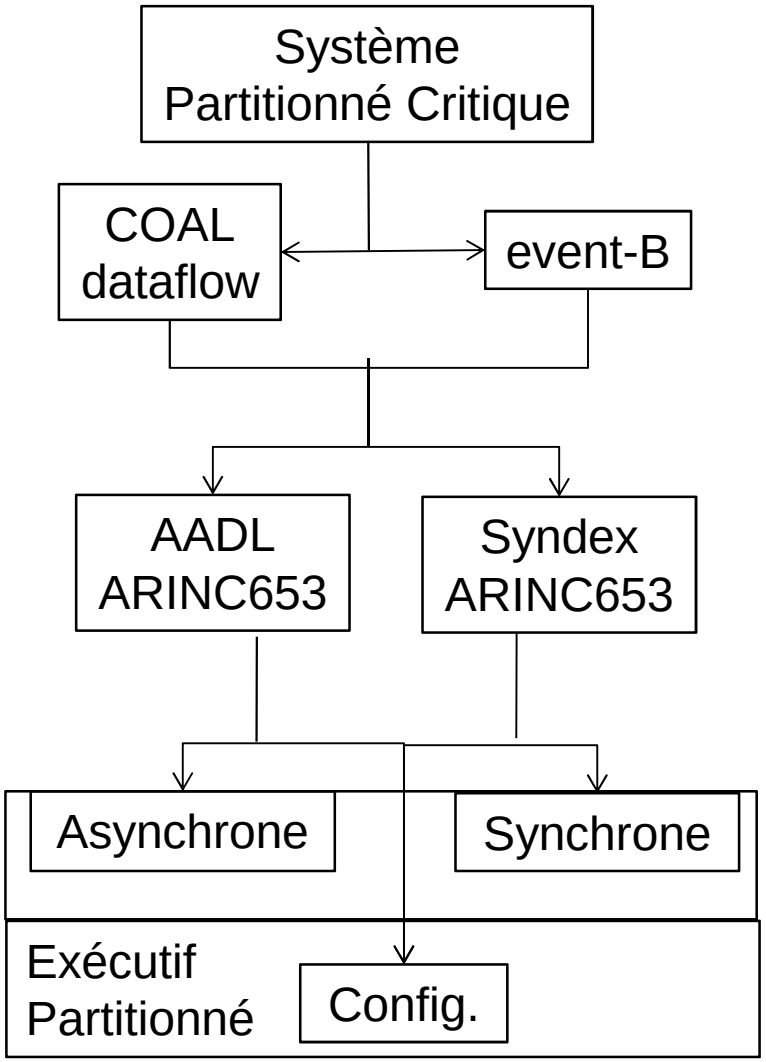
Objectifs

- Proposer un processus unifié et outillé pour la réalisation de bout en bout de systèmes critiques dans les domaines avionique et ferroviaire.
- Aider à la qualification des systèmes développés

- **Existant**
  - des chaînes outillées pour le développement de systèmes critiques, mais pas de méthodologie ou de processus de bout en bout clairement défini, du fait de la variété et de la complexité des besoins.
- **Objectif**
  - Proposer un processus unifié et outillé pour la réalisation de bout en bout de systèmes critiques dans les domaines avionique et ferroviaire.
  - Aider à la qualification des systèmes développés (DO-178C / EN-50128)
- **Intérêt**
  - Démontrer la faisabilité de la définition d'un tel processus
  - Consolider la définition et l'évaluation du processus en le confrontant aux spécificités de deux domaines applicatifs différents.
  - Faciliter la mise en commun ou le transfert de technologies entre les domaines avioniques et ferroviaires
    - exemple: utilisation des OS ARINC653 dans le domaine ferroviaire.
- **Verrou**
  - Impossibilité de proposer un processus universel qui réponde parfaitement à la diversité et à la complexité des besoins et des contraintes alloués
    - au système critique à développer
    - au processus de développement associé.
- **Solution**
  - Ajuster le processus proposé en fonction du contexte et du besoin concret.

- **Spécification:** modèle prouvé de l'attendu du système
  - Modélisation formelle du comportement en B événementiel
- **Conception:** modèle formel de l'architecture logiciel
  - Utilisation d'une sémantique déterministe (formalisable en B événementiel) du type flots de données et d'un environnement d'exécution associé
  - Répartition des fonctions du système (issues de la spécification système) entre différents composants logiciels, avec des contrats bien définis
  - Modèle d'activation, de configuration et de déploiement des composants sur la plateforme cible
- **Implantation:** génération de code correct par construction
  - Interprétation du modèle flots de données de l'architecture selon
    - une **approche asynchrone,** adaptable à des variations imprévues de l'environnement extérieur (rafale d'événements etc…)
    - une **approche synchrone,** dont on peut garantir plus facilement le déterminisme
- **Evaluation:** génération de tests unitaires et d'intégration
  - Intégration de l'outil PathCrawler *de* génération de scénarios de tests

Aperçu du Processus PARSEC



Focus de la présentation

Conception

- Modèle à composant COAL
- Mapping vers AADL

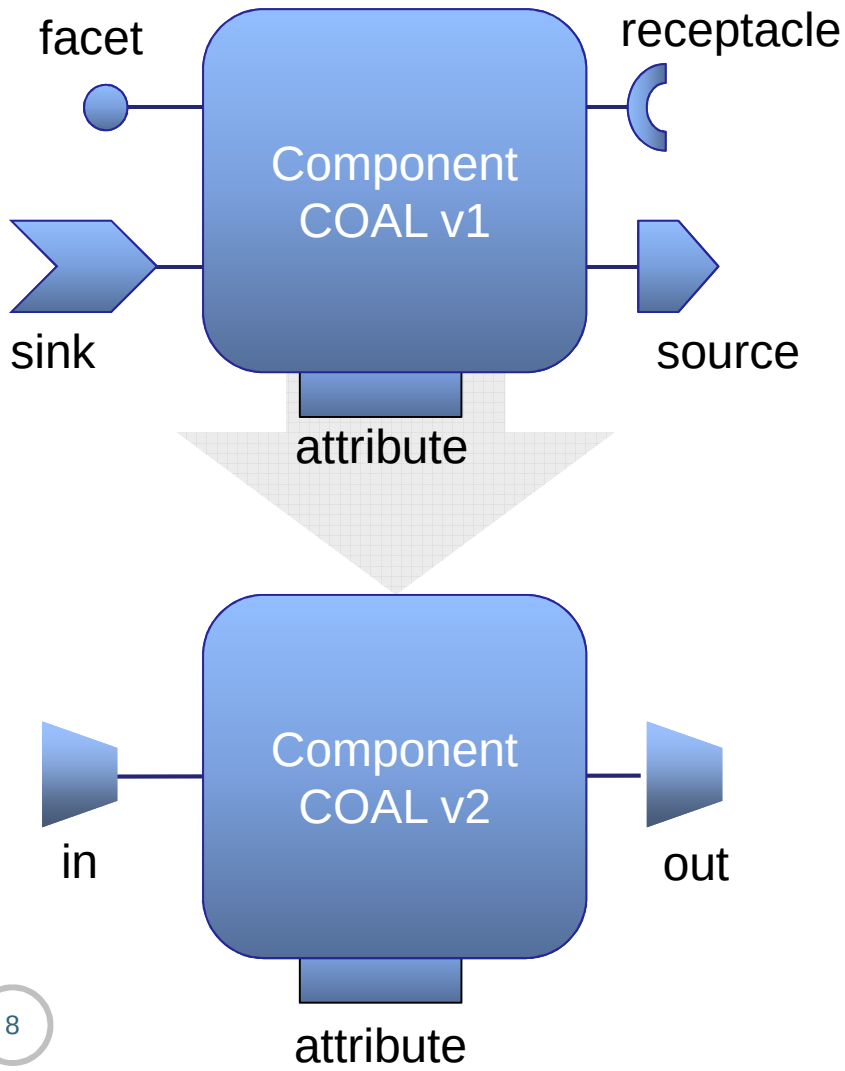
Implantation

- Asynchrone en utilisant AADL
- Sychrone en utilisant SynDEX

Cas d'étude

Problème: équivalence sémantique entre modèle et code généré

- **Problème**
  - Comment prouver que le code généré est sémantiquement équivalent à la description d'architecture ?
  - manque d'information dans le langage de description d'architecture COAL v1 sur le comportement interne des composants logiciels
  - non-déterminisme lié au modèle de calcul asynchrone mis en œuvre dans COAL v1
- **Solution**
  - MAJ du langage COAL pour modéliser le comportement interne des composants (flot de données, automates de mode)
  - Traduction des contrats fonctionnels (pre et post conditions) des composants COAL en spécifications ACSL
  - Vérification automatique des contrats des composants par:
    - des techniques d'analyse statique
    - la génération de tests unitaires et d'intégration



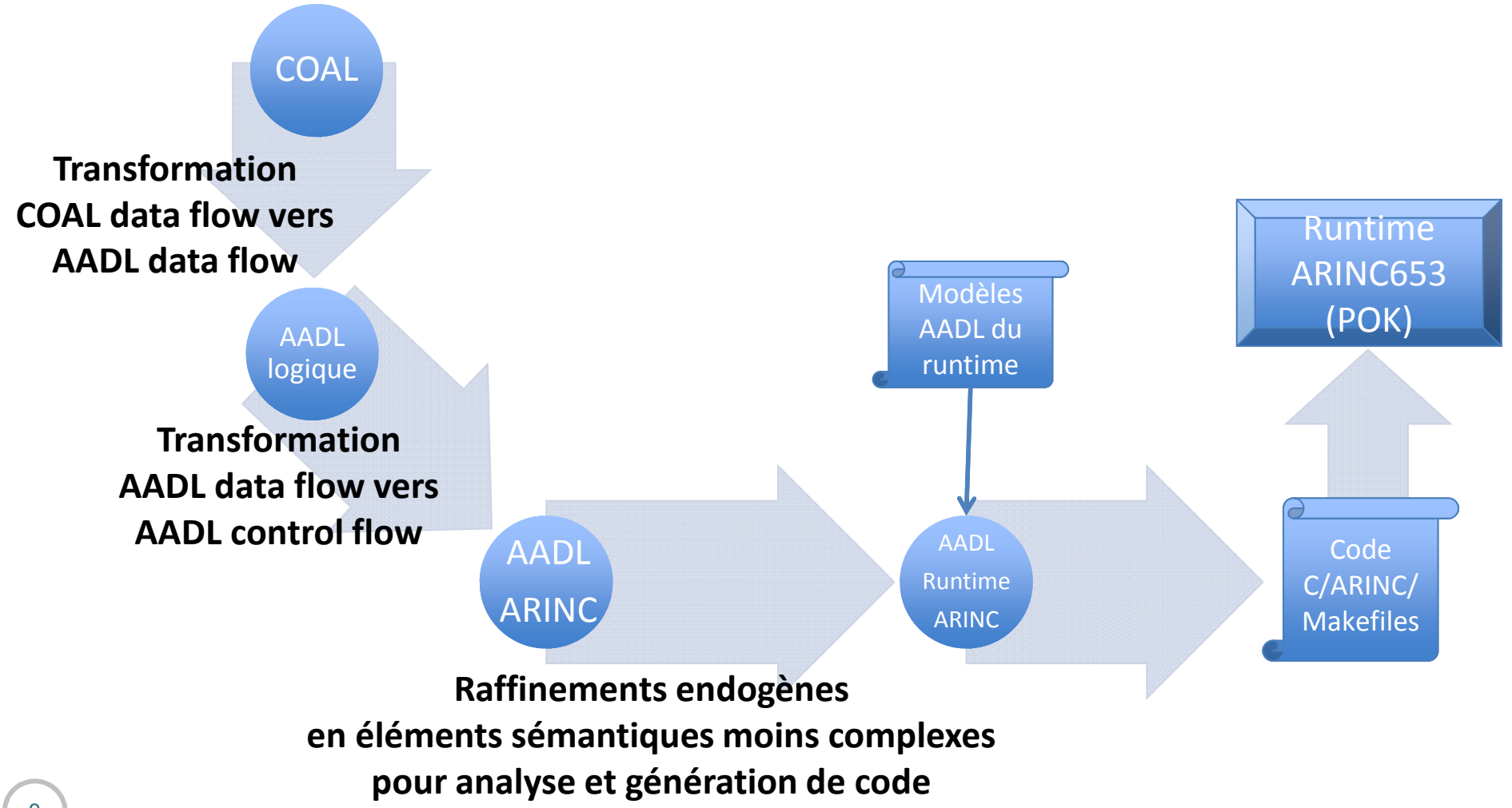
Point d'entrée

- **COAL v1 (NEPTUNE'09)** : sémantique d'interaction asynchrone (incompatible d'une interprétation synchrone)

Travaux réalisés

- **COAL v2** : passage à une sémantique flot de données générique
- **Sémantique d'exécution** : définition intrinsèque de la sémantique d'exécution flot de données
  - Opération: activable seulement si données d'entrées valides
  - Delay: peut produire ses sorties avant de consommer ses entrées (pour casser les cycles de dépendance de données)

Déploiement de COAL dans la branche asynchrone



Problème: équivalence sémantique entre modèle et code généré

Problème

Comment prouver que le code généré est sémantiquement équivalent à la description d'architecture ?

- Points de variations sémantique dans le langage AADL (pas de façon standard et précise d'implanter un type de connexion)
- Différentes implémentations et optimisations possibles du modèle d'entrée
- Surcoûts masqués liés à ces choix d'implémentation

Solution

- Identifie un sous-ensemble d'AADL avec une correspondance 1 pour 1 vis à vis des artefacts de déploiement.
- Traduction progressive du modèle COAL en AADL intermédiaires, résultants de différentes phases d'expansion

Raffinement architecturale pour les STRE critiques

Spécificités

Réduire le fossé entre les modèles du système initial analysé et ceux du système final déployé

- Optimiser les ressources induites par le code généré
- Préserver les propriétés initialement démontrées
- Assurer la portabilité vers différentes cibles

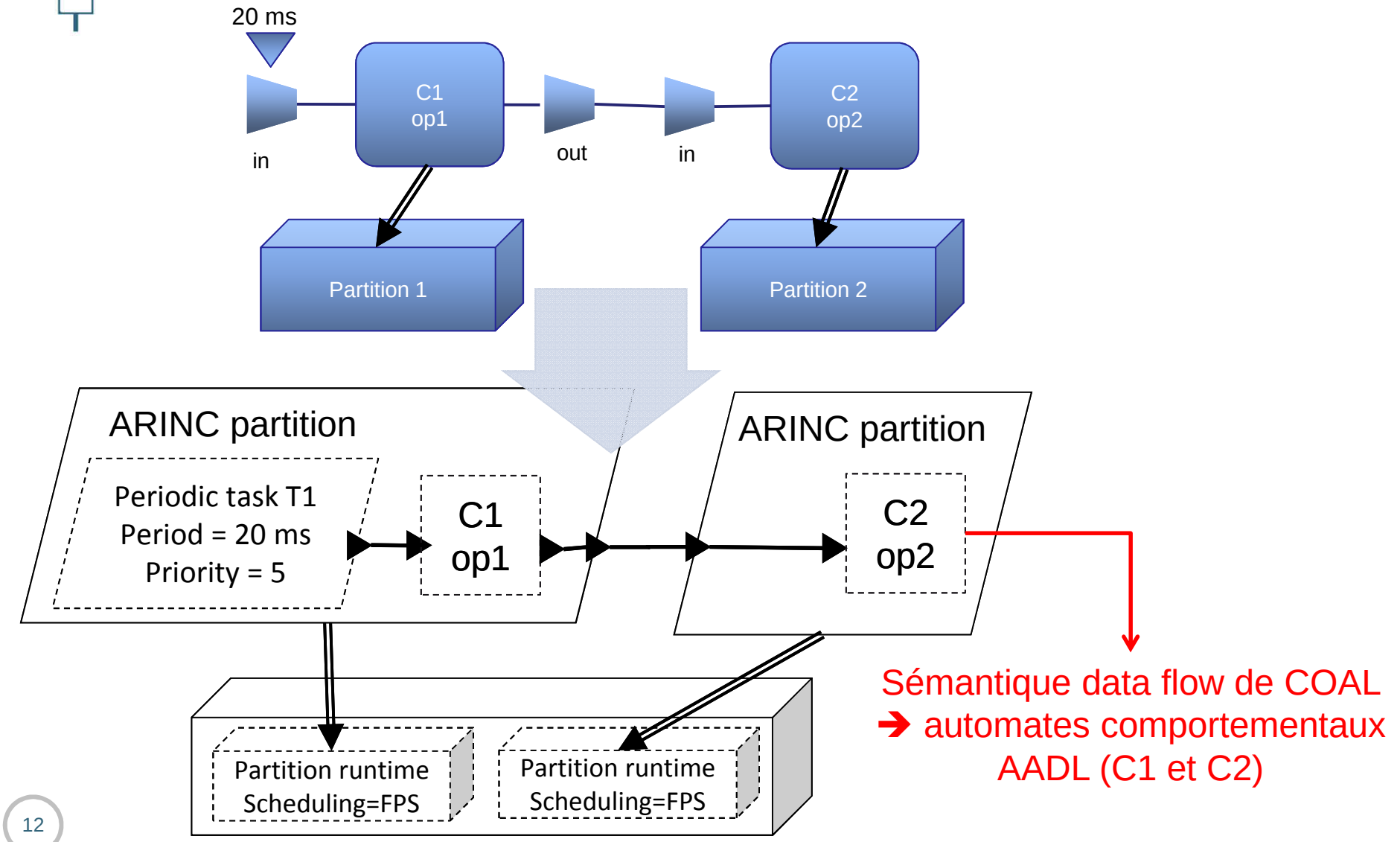
Solution

Raffiner les modèles AADL en décomposant les éléments en d'autres éléments moins complexes

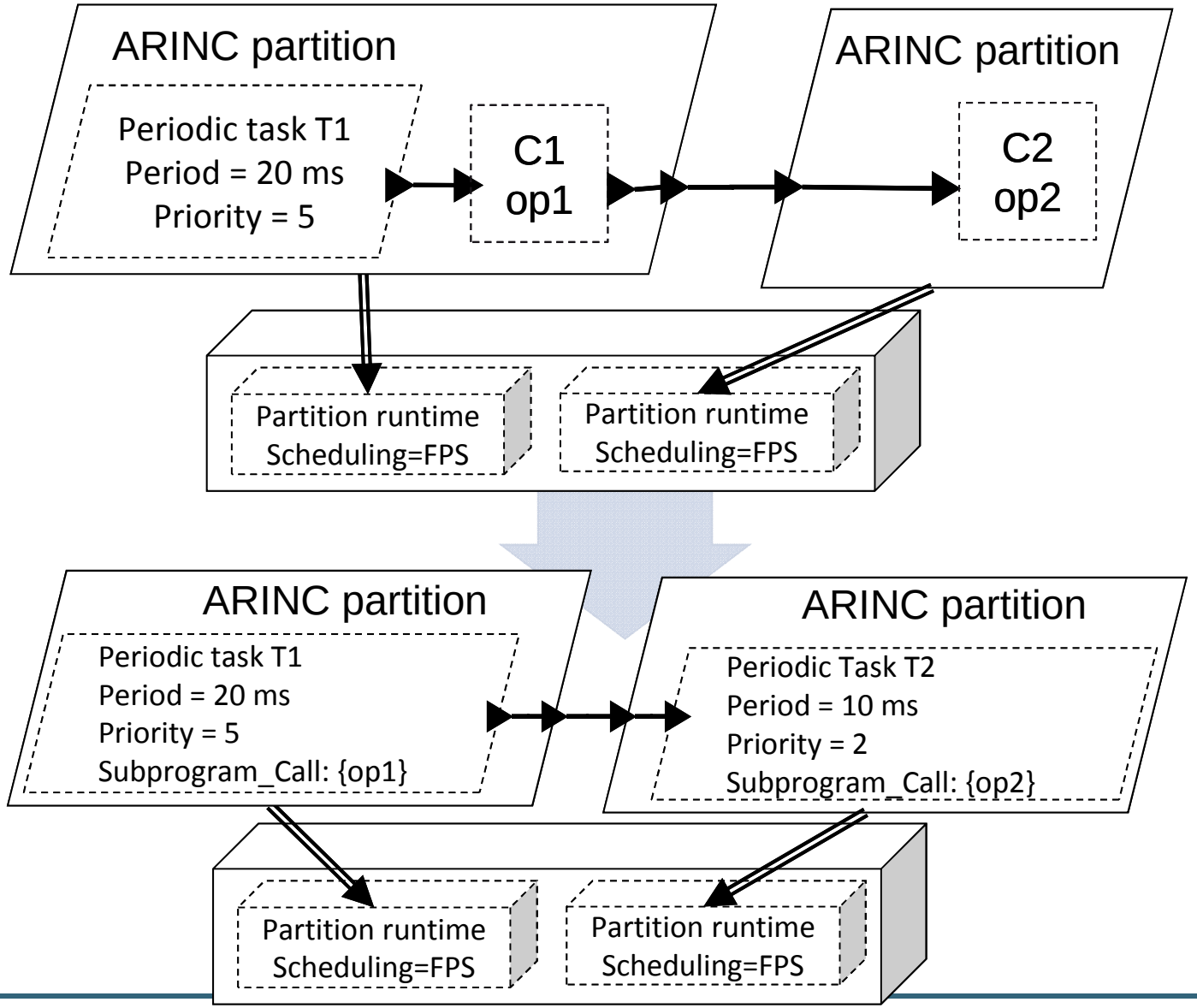
- Raffiner par transformations endogènes successives
- Enchainer les transformations selon les résultats d'analyse
- Sur-imposer les règles de transformation pour portabilité

ICECCS'2012: Design Patterns for Rule-based Refinement of Safety Critical Embedded Systems Models

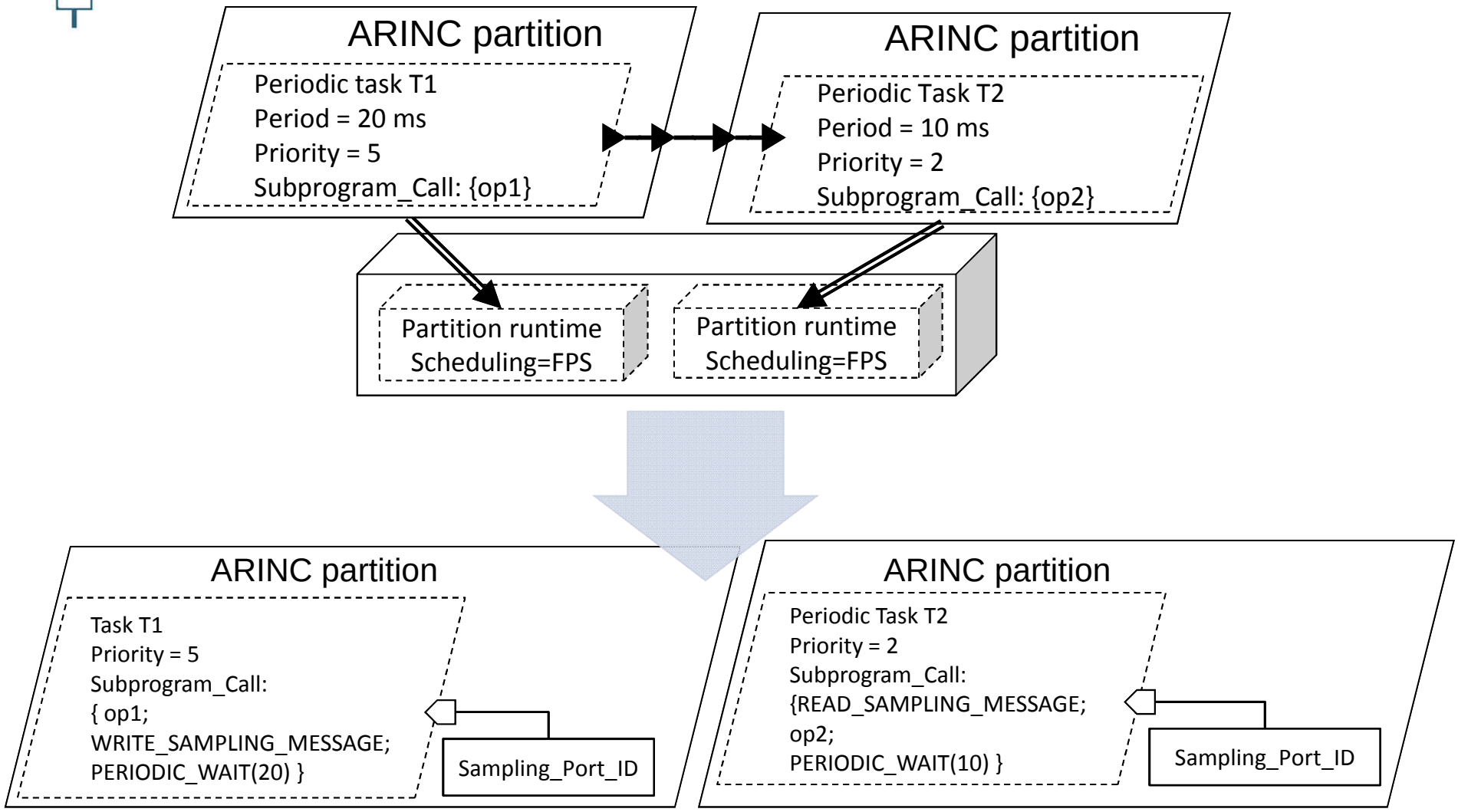
Synopsis du processus de transformation de modèle



Synopsis du processus de transformation de modèle



Synopsis du processus de transformation de modèle



Implémentation dans RAMSES

- **Concerne la 3° phase de génération**
  - production du modèle AADL de déploiement
  - génération de code

- **RAMSES = Refinement of AADL Models for Synthesis of Embedded Systems**
  - Intégration à OSATE2
  - Implémenté en Java et ATL pour le raffinement de modèles AADL
  - Sera distribué en licence EPL à horizon « début septembre 2012 »
  - Cible la plate-forme ARINC653 et OSEK pour illustrer la « portabilité du générateur de code »

- **Travaux en cours pour augmenter l'analysabilité et le déterminisme des modèles AADL considérés**

- **Problème**
  - Contraintes d'ordonnancement au niveau des partitions imposées par l'intégrateur système sous la forme d'un fenêtrage temporel (Major Time Frame)
  - L'ordonnancement intra-partition est par défaut de nature asynchrone et laissé à la charge du fournisseur d'application
  - Besoin de pouvoir générer automatiquement un ordonnancement statique intra-partition à partir d'une description du type flots de données de l'application respectant le fenêtrage temporel des partitions.
- **Solution**
  - Extension de la méthodologie AAA et du logiciel SynDEx pour permettre une prise en compte des architectures d'implantation partitionnées définies par le standard avionique ARINC 653.

Démonstrateur radio-logicielle en avionique

- **Objet: forme d'onde aéronautique**

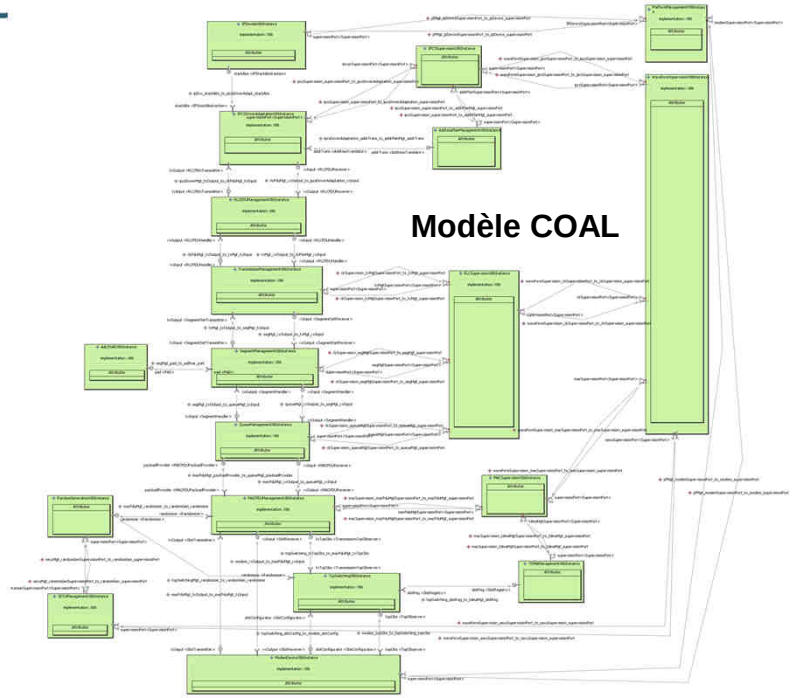
- **Intérêt: Certification D0-178 DAL C des systèmes de radiocommunication logicielle**

- **Spécification et modélisation en COAL v2 de la forme d'onde aéro PWF**

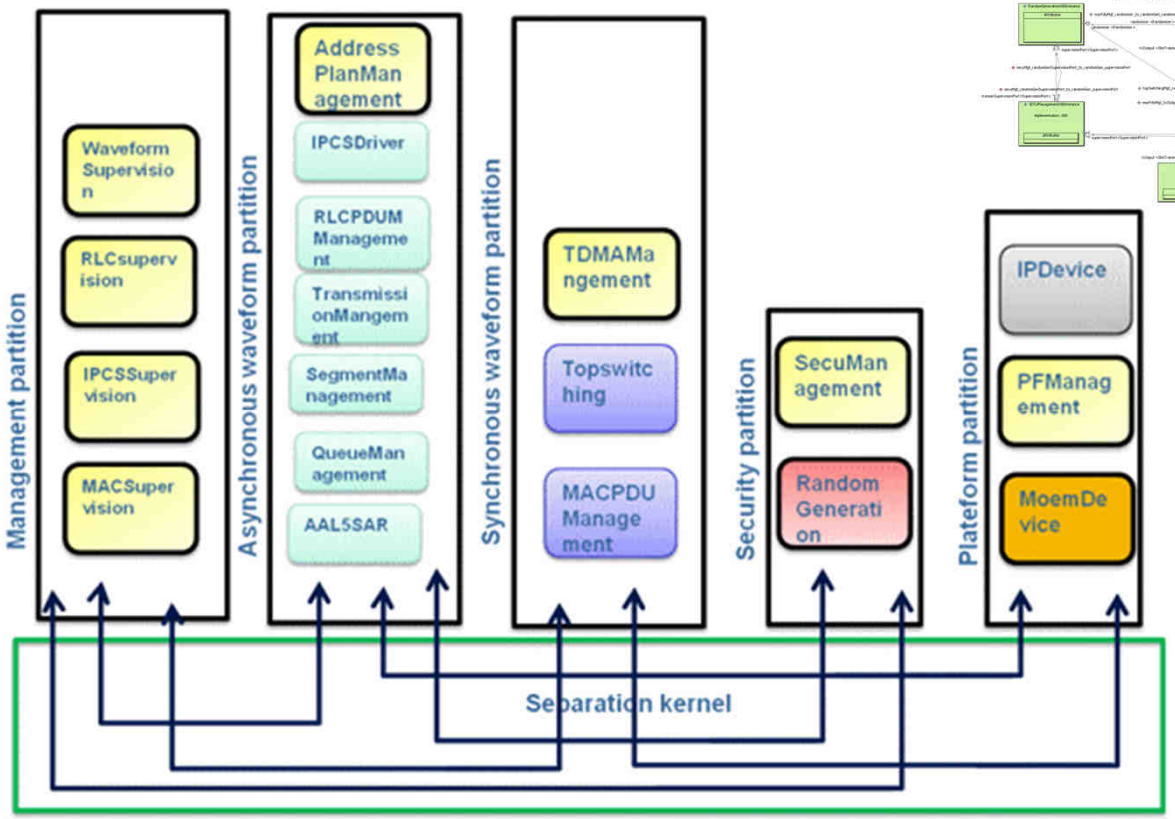
Le travail sur le cas d'étude avionique jusqu'à T0+24 a permis de:

- valider la sémantique d'exécution flots de données de COAL v2
- valider le mapping COAL vers AADL et SynDEx

Déploiement sur un OS ARINC653



Modèle COAL



Deployment Arinc653

THALES INRIA ROCQUENCOURT TELECOM ParisTech

Conclusion

- **Le projet PARSEC propose un processus de conception, basé sur l'IDM, pour les STRE critiques**
  - De la spécification formelle des exigences (event-B)
  - Jusqu'à la validation du code (génération des cas de test)
  - En passant par l'implantation asynchrone ou synchrone de ces modèles
- **Nous nous sommes ici concentré sur l'implémentation asynchrone pour montrer**
  - Les apports de l'IDM (modélisation et transformation de modèles) dans un processus de développement des systèmes critiques
- **Concernant la sécurité et la sûreté, ces travaux ont débouché sur un travail de thèse sur la composition de patron de sécurité et de sûreté de fonctionnement dans l'IDM**