The logo for IRISATECH is a purple rectangle containing a stylized white and gold graphic of two overlapping triangles pointing right. The text "IRISATECH" is written in white, uppercase letters to the right of the graphic.

IRISATECH

le club des partenaires de l'Irisa

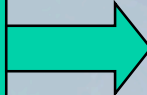
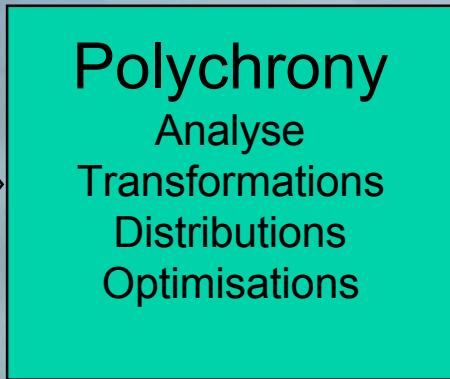
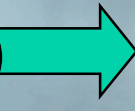
34e rencontre/22 juin 2005

Modélisation formelle de systèmes embarqués et applications

POLYCHRONY
Plate-forme de conception
synchrone (Signal)

Loïc Besnard (CNRS) / Thierry Gautier (INRIA)

**Notations
haut niveau**



Vérification

Signal

*SystemC, C, C++, ... (gcc)
GME*

**Génération
de code**

C, C++, Java,
SynDEx, Signal
VHDL

Polychrony

- Compilateur Signal « batch »
- Éditeur graphique couplé au compilateur
- Outils de preuve :
 - Sigali (Vertecs)
 - *Coq (Logical)*
- Bibliothèque de modèles :
 - RTOS APEX-ARINC (avionique)

« *Design space exploration* »

Fonctionnel, architecture, performance

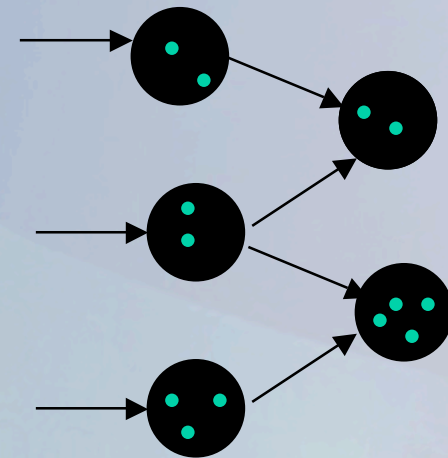
- Calculs formels
 - flot de contrôle (hiérarchies de BDD)
 - flot de données
- Ensemble de services : un service peut être appliqué selon l'objectif
 - Simulation
 - Vérification formelle
 - Code embarqué
 - Simulation d'architecture
 - Spécification hardware
 - ...

Simulation

- Calcul formel sur booléens + \perp (calcul d'horloges)
 - Structure le contrôle de l'application
 - Résoud les contraintes de synchronisation
- Complément formel des \perp (booléanisation)
- Séquentialisation
 - détection de cycle
- Génération de code
 - C, C++, Java

Simulation (2)

- *Calcul d'horloges*
- *Booléanisation*
- **Calcul des lignées**
- *Séquentialisation*
- *Génération de code*



Compilation séparée

Synthèse d'interface : abstractions « boîtes noires »

- Dépendances e/s
 - Fermeture transitive réduite aux e/s
 $X \rightarrow Y \text{ at } Hi$

- Projection d'horloges
 - Horloges des signaux e/s
 - Horloges des dépendances e/s

Compilation séparée (2)

Synthèse d'interface : abstractions « boîtes grises »

- *Dépendances e/s*
- *Projection d'horloges*
- Abstraction des lignées et leurs dépendances

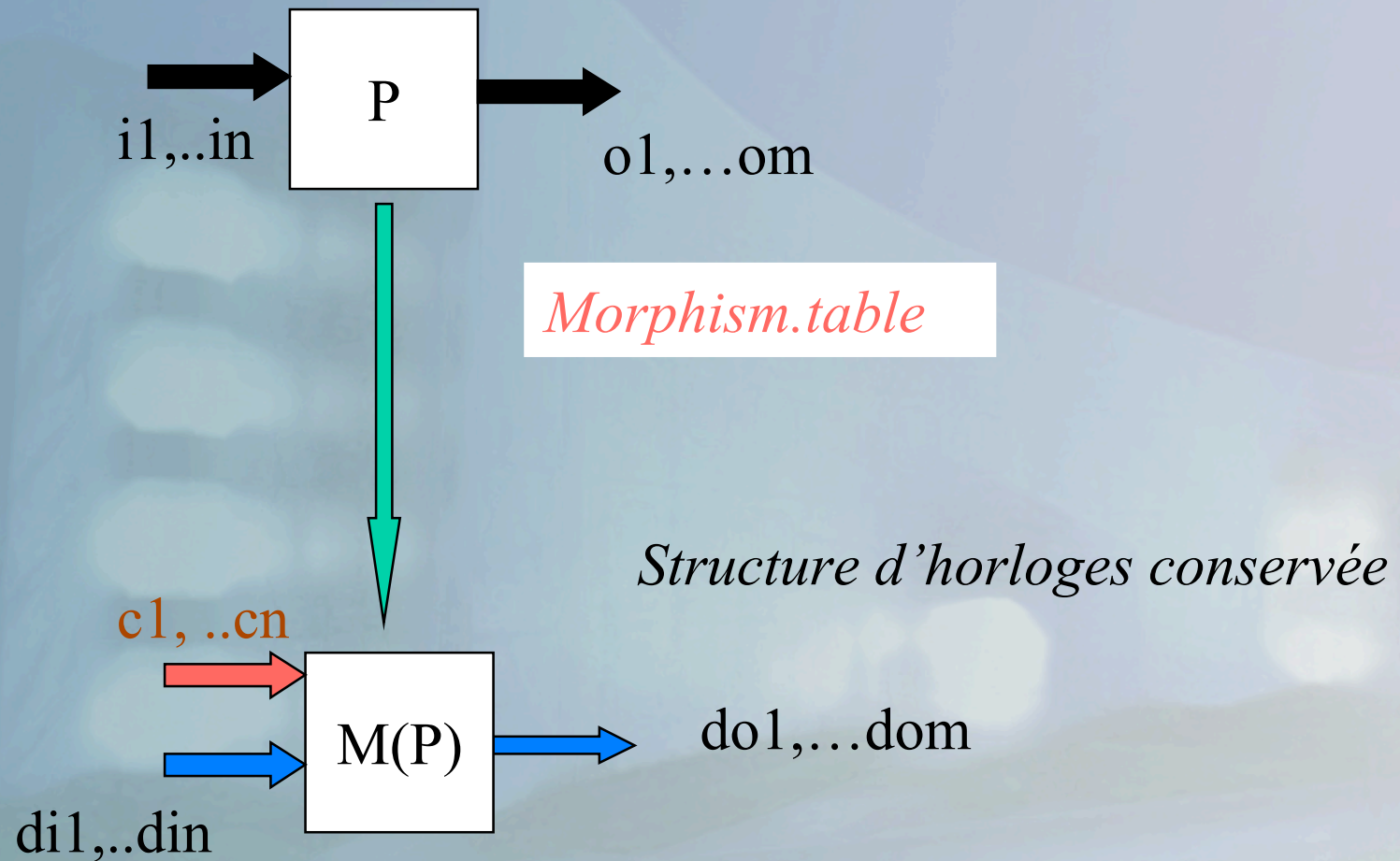
Exemple de partitionnement utilisateur vers la distribution

- Principes
 - Description de l'application
 - Description de l'architecture cible
 - « Mapping » de l'application sur l'architecture cible
 - « Compilation »

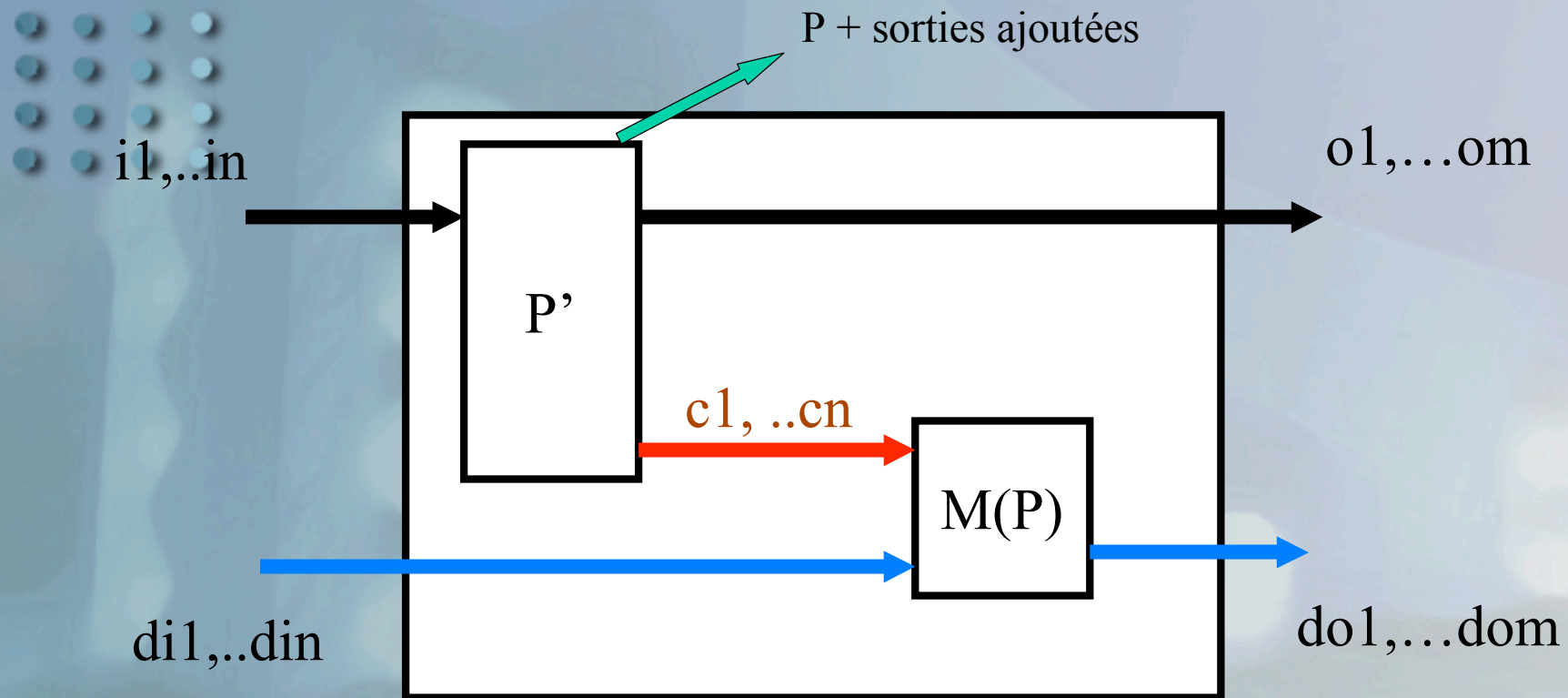
Étapes de la « compilation »

- *Calcul d'horloges*
- *Booléanisation*
- *Partitionnement selon la description d'architecture*
- *Extractions des sous-graphes*
- *Synthèse d'interface*
- *Retour vers l'environnement utilisateur*

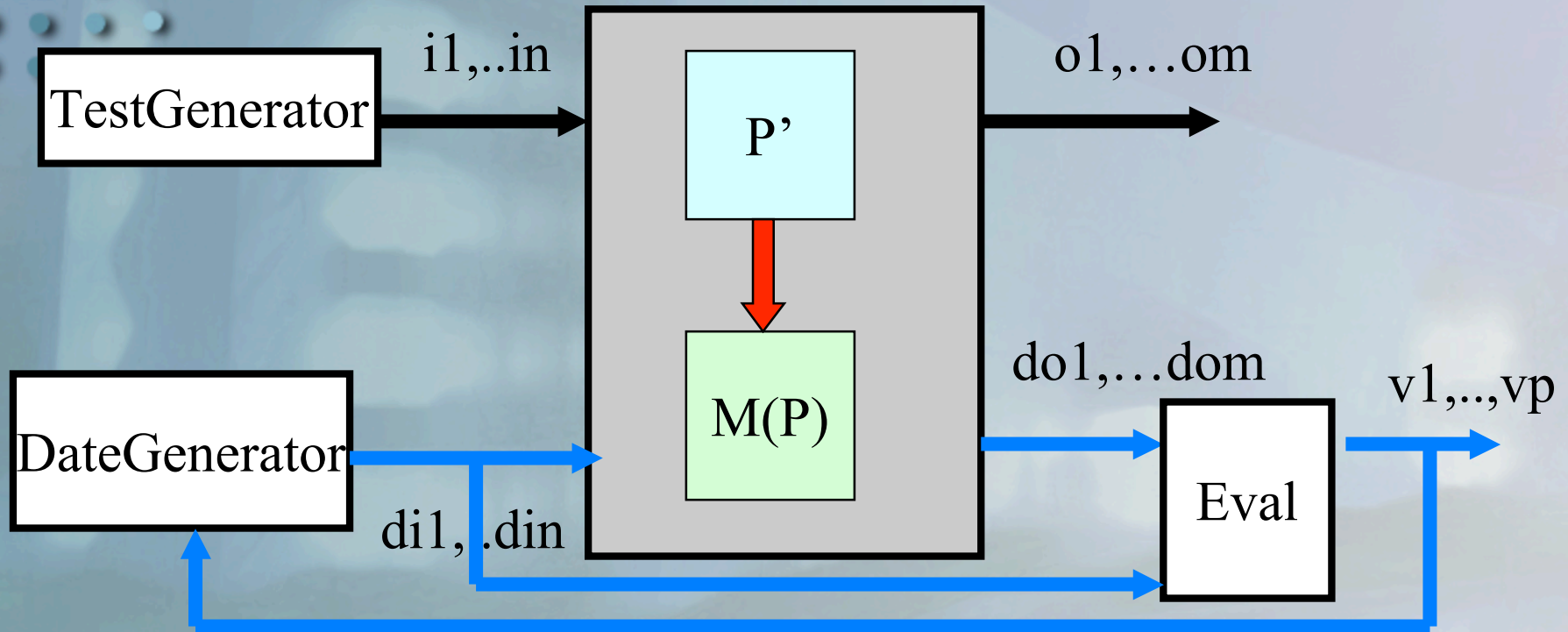
Évaluation de performance : morphisme



Expérimentations : co-simulation



Expérimentations



Expérimentations

- $P \dashrightarrow$ Séquentialisation \rightarrow Morphisme
Co-simulation
- $P \dashrightarrow$ Lignées(P) \rightarrow Séquentialisation \rightarrow Morphisme
Co-simulation

Expérimentations (suite)

$P = (| Q | R |)$ (« mapping »)

* Compilation globale

* Partitionnement : Q' et R'

* $Q1COSTQ1 = Clusters(Q'' | M(Q'))$ code séparé

* $Q2COSTQ2 = Clusters(R'' | M(R'))$ code séparé

Et

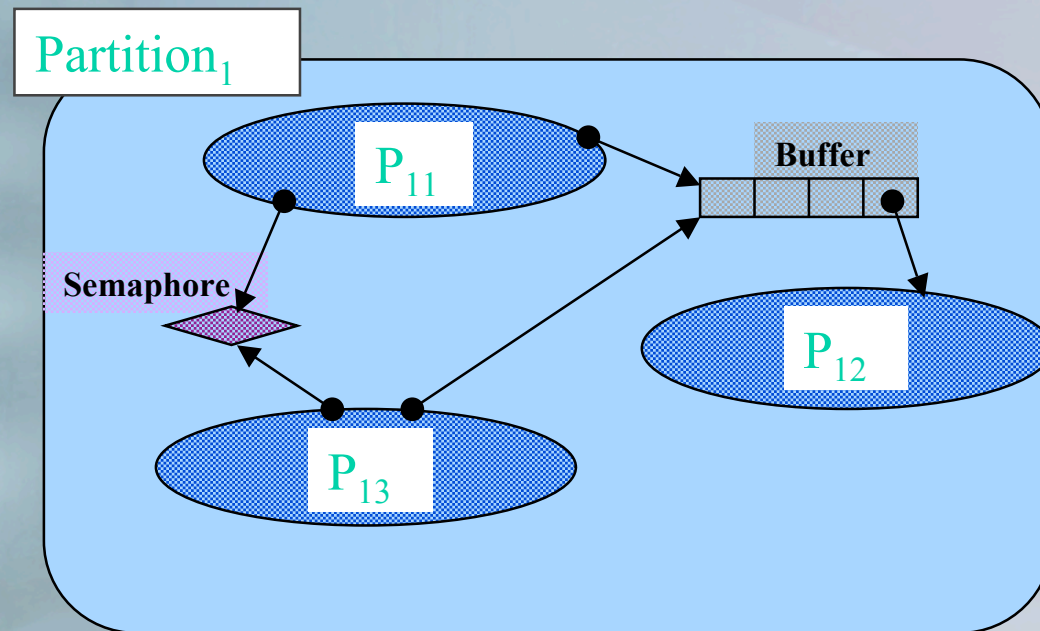
$(| Abs(Q1COSTQ1) | Abs(Q2COSTQ2) |)$

Modélisation d'Architectures Modulaires Avioniques

(standard ARINC 653)

Modèle SIGNAL de partition :

- modèle de services
- + modèle de process
- + modèle de l'OS (niveau partition)



Diffusion

- Site Web
 - <http://www.irisa.fr/espresso/Polychrony>
 - Documentation, manuel de référence, tutoriel
 - Base d'exemples
 - Solaris/ Linux/MacOS/Windows
 - ***Sources non diffusés***
- Version Commerciale
 - [RT-Builder \(Sildex\)](#) de TNI-Software

Perspectives

- Connexion IDE (GME, Eclipse,....)
- Importation de modèles (SystemC, Java, etc.)
 - **Format d'échanges avec outils externes**
 - **Transformation de (méta) modèles**
 - **Morphismes entre modèles hétérogènes**
 - **API d'interaction avec outils externes (et utilisateurs)**
 - Gestion et rapport d'erreurs
 - Appel des fonctionnalités (transformations)