De la modélisation à l'analyse des systèmes temps-réel : Contributions au projet Waruna

Anh-Toan Bui Long
Directeurs de thèse : Emmanuel Grolleau, Yassine Ouhammou

LIAS Lab / ISAE - ENSMA
builonga@ensma.fr

Jeudi 21 décembre 2017

Contents

1. Introduction
2. Problématiques
3. Contribution 1 : CONSERT
4. Contribution 2 : Réseaux
5. Conclusion & Perspectives

Contents

1 Introduction

2 Problématiques

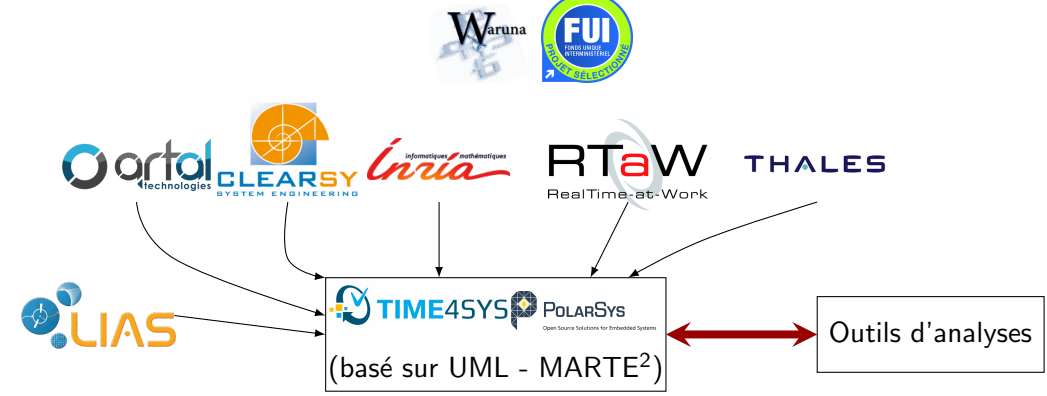
3 Contribution 1 : CONSERT

4 Contribution 2 : Réseaux

5 Conclusion & Perspectives

Waruna Project - Time4Sys Framework

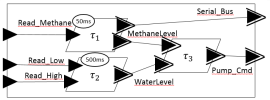
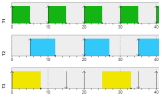
Financé via le projet Waruna[1] par Fonds Unique Interministériel



[1] www.waruna-projet.fr

[2] OMG. 2015. UML profile for MARTE: modeling and analysis of real-time embed-ded systems. (2015)

Conception et analyse

Design Model	Analysis Model
	Temporal view : 
Close to reality	Scheduling Analysis oriented
Full description of a system	A real-time abstraction of a system
Adapted for MDE	Rarely Adapted for MDE
Known by industry	**Known by real-time research community**

Problems

- Gap between design and analysis
- Transformations is time consuming

Travaux Existant

Recently some work have been proposed to fill the gap between design and analysis:

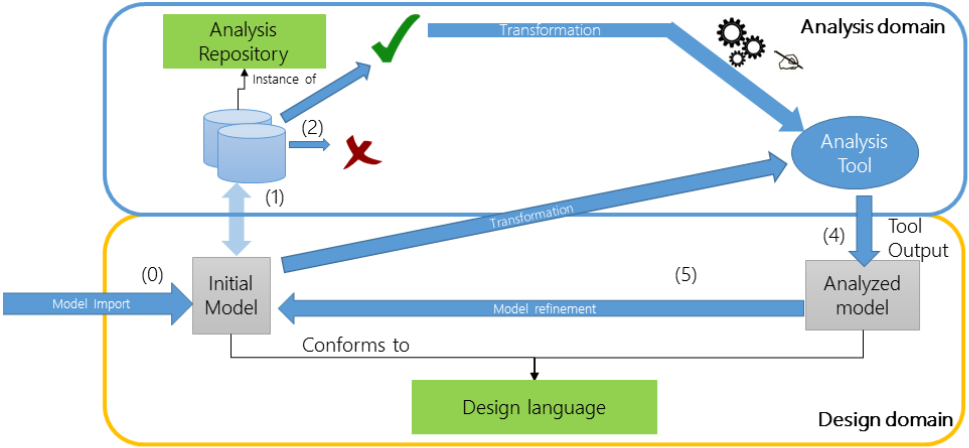
- Analysis repository: to determine the adequate analysis tool for the model[1] and transform to the analysis tool
- Hard-coded bridge between modeling (AADL) and analysis (Cheddar) [2]
- Transformation from ready-to-analyze model to the analysis [3]

[1] Yassine OUHAMMOU, *Model-based Framework for Using Advanced Scheduling Theory in Real-Time Systems Design*, Thesis, ISAE-ENSMA, december, 2013

[2] Vincent GAUDEL, et al. *An ada design pattern recognition tool for aadl performance analysis.* ACM SIGAda Ada Letters. ACM, 2011.

[3] Guillaume BRAU, *Integration of the analysis of non-functionnal properties in Model-Driven Engineering for embedded systems.* Thesis, University of Luxembourg, 2017

Processus d'Analyse avec MoSaRT/Time4Sys



Contents

1 Introduction

2 Problématiques

3 Contribution 1 : CONSERT

4 Contribution 2 : Réseaux

5 Conclusion & Perspectives

Besoins

En Réseaux

- Manque de détails pour l'analyse sur le métamodèle MoSaRT
- Quasi inexistant sur UML - MARTE
- Analyses réseaux temps-réel moins pessimistes

Analyse temps-réel

- Adaptation de la conception à l'analyse
- Réutilisation de la connaissance des analystes
- Concepteur autonome
- Transformation Conservatrice par rapport à la structure et au pessimisme

Contents

1 Introduction

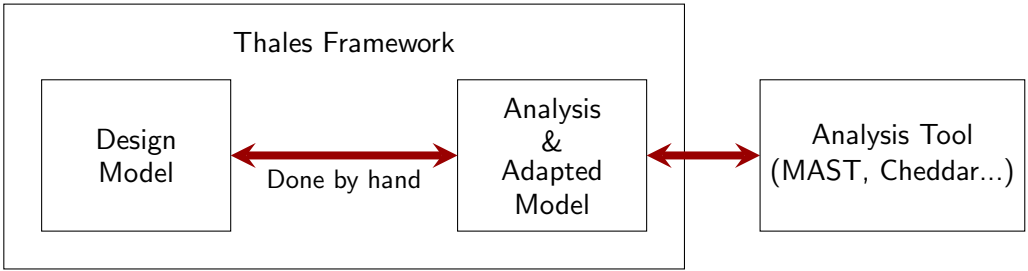
2 Problématiques

3 Contribution 1 : CONSERT

4 Contribution 2 : Réseaux

5 Conclusion & Perspectives

Une problématique industrielle

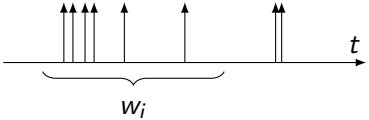


Problèmes industriels

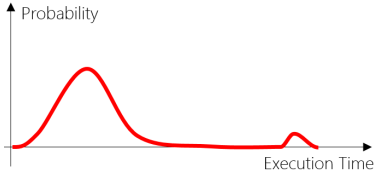
- Pour adapter le modèle, l'expérience de l'analyste est nécessaire
- Adaptation du modèle avec la connaissance de l'outil d'analyse
- Augmentation du temps
- Coût élevé de l'analyse

Motivationnal Examples - Industrial cases

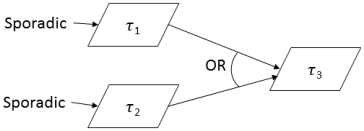
- Sliding window context
  For any time window of size w_i, there are at most N_{max} activations



- Probabilistic timing context



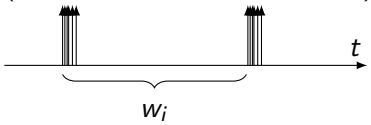
- Alternative triggering context



- Watchdogs context

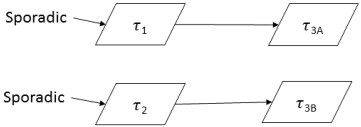
Possible transformations

- Sliding window context
  If the test is offset sustainable
  (critical instant ⇒ worst case)



- Probabilistic timing context
  (If it is C-sustainable)

- Alternative triggering context



Tasks τ_{3A} and τ_{3B} are co-allocated.

- Watchdogs context

if context is C, T and offset sustainable

CONSERT Transformation Repository

CONSERT : CONServative Endogenous Repository-based Transformations

- Constructed by the analyst to share his experience
- Used by the designer as a decision support

Principle

- Determine the transformations to apply to the model.
- Can be used with well known modeling languages.
- Keep a trace of the transformation

Detecting transformations

Objective

To propose an environement detecting non-analyzable cases then the adaptations to apply.

How

Set of detection rules expressed on a formal language (e.g. OCL[a]) compliant with design language to verify to execute a transformation.

[a] Object Constraint Language : Object Constraint Language, OMG Available Specification, Version 2.0, 2006

Examples of detection rules

- Monoprocessor case: `resource.ProcessingResource->size()=1`
- Alternative triggering: `task.predecessors->size()>=2`

Storing the knowledge - Part 1

Repository (instanciated by the analyst):

- Detection Rules
- Contexts
- Transformation File

Detection Rule :

$$R_i =< Expr, name > \quad (1)$$

Context :

$$C_n =<< R_i, Boolean >^{\mathbb{N}}, TransformationFile > \quad (2)$$

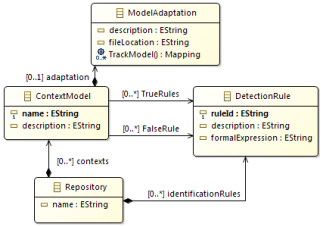
Repository :

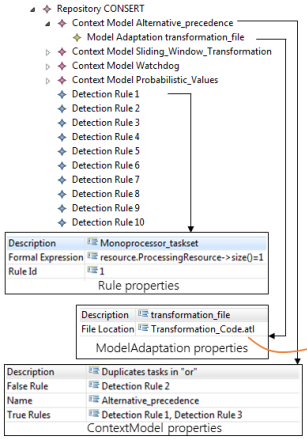
$$Repository =< C_n^{\mathbb{N}}, R_i^{\mathbb{N}} > \quad (3)$$

Storing the knowledge - Part 2

Repository (instanciated by the analyst):

- Detection Rules
- Contexts
- Transformation File

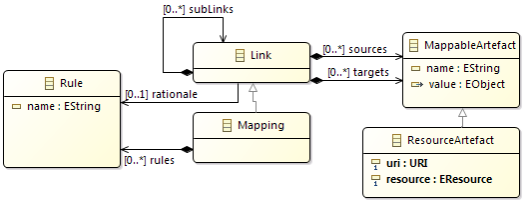




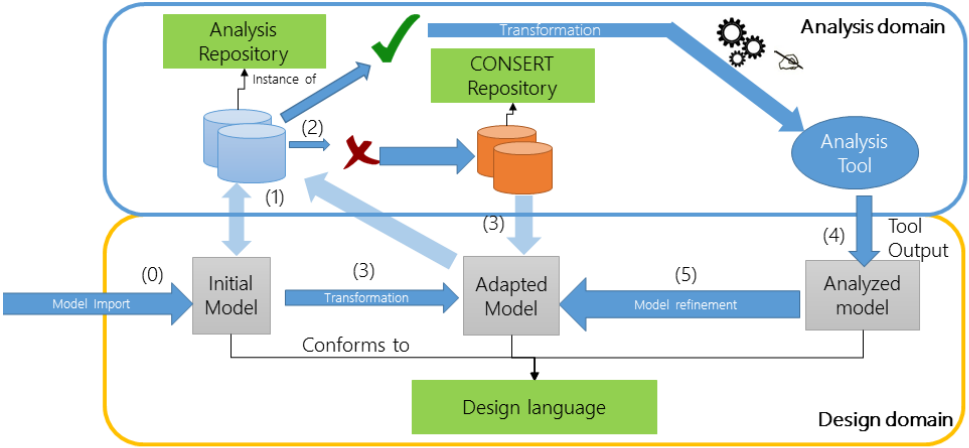

```
lazy rule ArrivalPattern {
    from
        s : Time4SysModelDesign!ArrivalPattern
    to
        t : Time4SysModelDesign!ArrivalPattern (
            parent <- s.parent,
            jitter <- thisModule.Duration(s.jitter),
            phase <- thisModule.Duration(s.phase)
        )
}
lazy rule SlidingWindow {
    from s : Time4SysModelNFP!SlidingWindowPattern (true)
    to t : Time4SysModelNFP!BurstPattern(
        reference <- s.reference, --
        minInterarrival <- thisModule.NFP_Duration(s.windowSize),
        burstSize <- s.nbEvents -- Integer
    )
}
lazy rule NFP_Duration {
    from s : Time4SysModelNFP!Duration (s.oclIsTypeOf(NFPModel!Duration))
    to t : Time4SysModelNFP!Duration(
        value <- s.value,
        unit <- s.unit,
        best <- s.best,
        worst <- s.worst,
        clock <- s.clock,
        precision <- s.precision
    )
}
```

Track transformations

- Necessity of tracking the transformations
- Generated by through the transformation
- Each execution creates a trace to cancel an adaptation



Processus d'Analyse avec MoSaRT/Time4Sys et Consert



Démonstration

Démonstration

Paper published

Bridging the gap between practical cases and temporal performance analysis: a models repository-based approach, RTNS 2017

Contents

1 Introduction

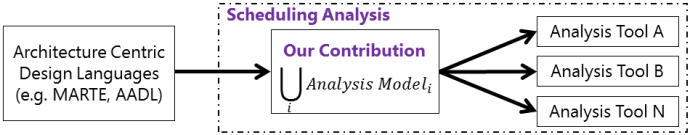
2 Problématiques

3 Contribution 1 : CONSERT

4 Contribution 2 : Réseaux

5 Conclusion & Perspectives

Real-Time Networks



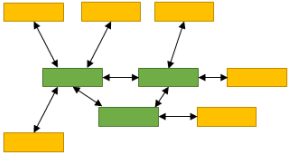
Our Objective

- Propose a network model adapted for architecture and analysis
- Explicitation of the network behavior with clear semantics
- Enhancing the reusability of networks architectures

Différent types de réseaux...

Switched architectures

⇒ A cable links 2 nodes

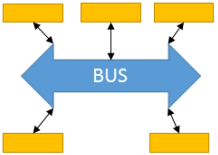


Examples :

- Ethernet
- AFDX (Avionics Full DupleX Switched Ethernet)

Shared bus architecture

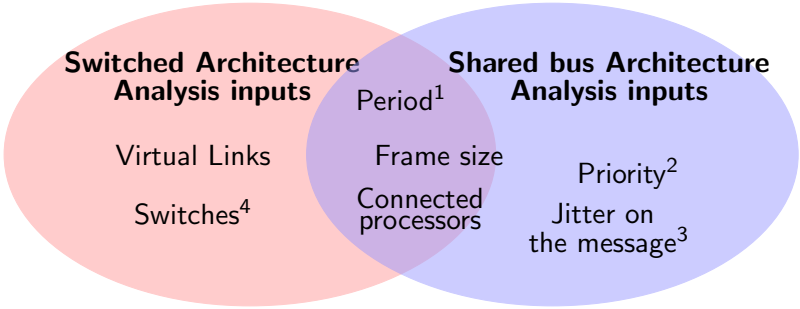
⇒ A cable links several nodes



Examples :

- Controller Area Network (CAN)
- Local Interconnect Network (LIN)

...Pour différentes analyses



[1] Liu, Chung Laung, and James W. Layland. "Scheduling algorithms for multiprogramming in a hard-real-time environment." Journal of the ACM (JACM) 20.1 (1973): 46-61.

[2] Davis, Robert I., et al. "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised." Real-Time Systems 35.3 (2007): 239-272

[3] Bril, Reinder J., Elisabeth FM Steffens, and Wim FJ Verhaegh. "Best-case response times and jitter analysis of real-time tasks." Journal of Scheduling 7.2 (2004): 133-147.

[4] Kemayo, Georges, et al. "A forward end-to-end delays analysis for packet switched networks." Proceedings of the 22nd International Conference on Real-Time Networks and Systems. ACM, 2014

Problems statement

- How to find an appropriate analysis referring to a given architecture ?
- How to integrate Real-Time networks design in an incremental model based process?
  - Since the choice of the network protocol comes later in the development life-cycle
- How to facilitate the reuse of systems architectures when changing network protocol?
  - With a low cost of design reengineering

Our Approach

- STEP 1: Capitalization of existing researches related to Real-Time Networks
- STEP 2: Integration into a model based process

Capitalization of existing researches related to Real-Time Networks

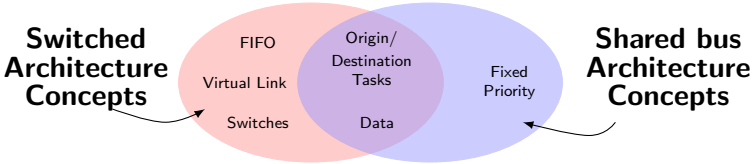
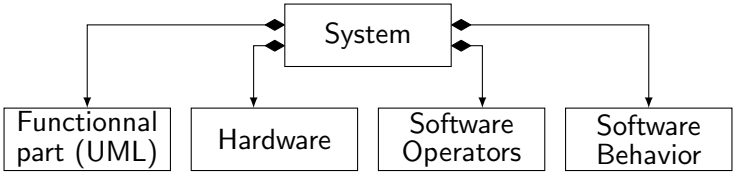


Table: Examples of protocol-independent and protocol-specific concepts

	Concept	AFDX	CAN
Protocol specific concept	Communication Medium	Switches & Ethernet cables	CAN cable
	Latency	On switches	Frame transmission time
Protocol independent concept	Origin & Destination tasks	One sender and possible multicast	
	Data	Data Frame (and frame size)	
	Router	Changing network i.e. changing protocol	

MoSaRT General Structure

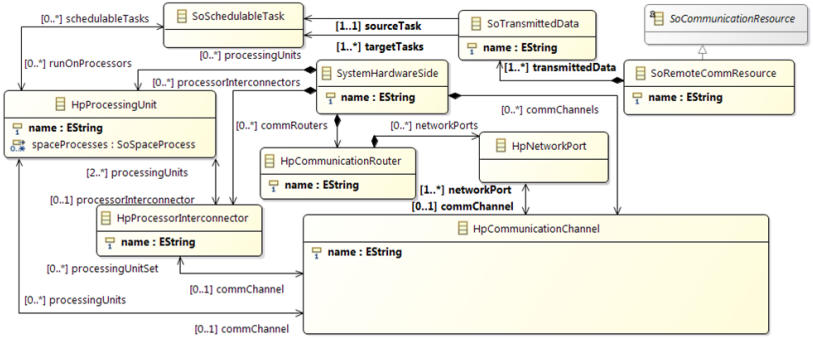


System is composed of:

- **Software operations**: representation of logic objects of the system(e.g. tasks, resources)
- Software behavior: representation of the temporal behavior of the system)
- **Hardware**
- Functionnal Part (UML)

Extension of the abstract (MetaModel) and the concrete syntaxes

Original Metamodel



Defining a network

- Un noeud "terminal", origine ou destination de flux :

$$TerminalNode =< \{processor, multiProcessor, router\}, CommunicationPort > \quad (4)$$

- Un noeud "réseau" communiquant avec les autres noeuds réseaux :

$$Node = \{CommunicationPort, CommunicationSwitch\} \quad (5)$$

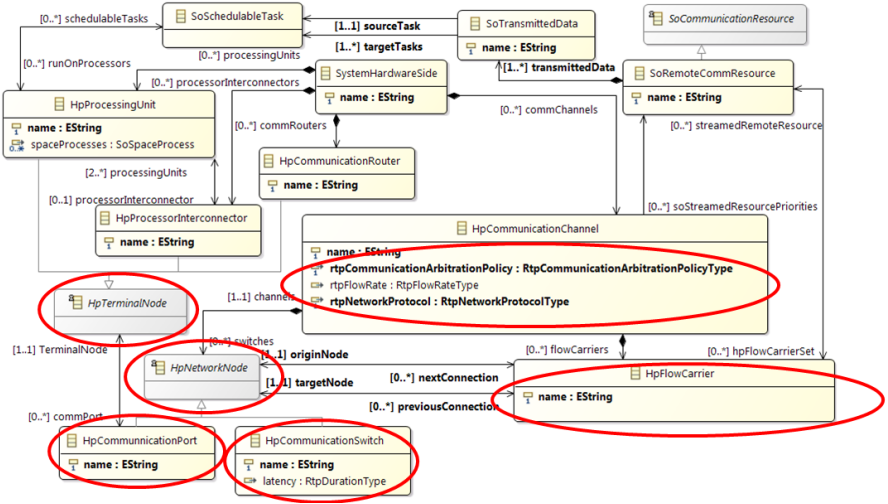
- Un lien réseau spécifiant l'origine et la destination :

$$Link =< Node_{origin}, Node_{Destination}, messages^{\mathbb{N}} > \quad (6)$$

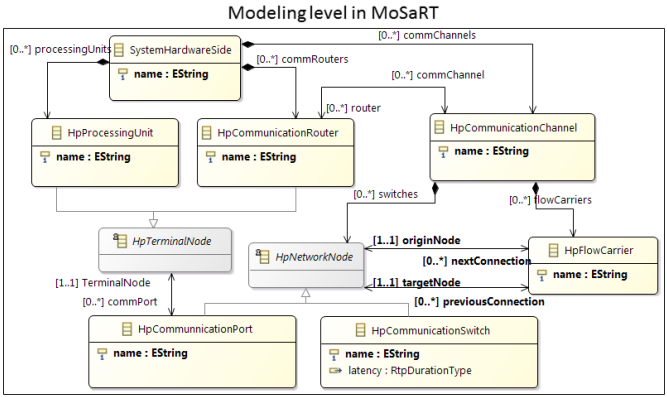
- La description d'un réseau :

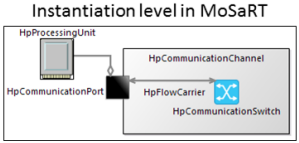
$$CommunicationChannel =< Node^{\mathbb{N}}, Link^{\mathbb{N}}, messages^{\mathbb{N}} > \quad (7)$$

Metamodel Extension

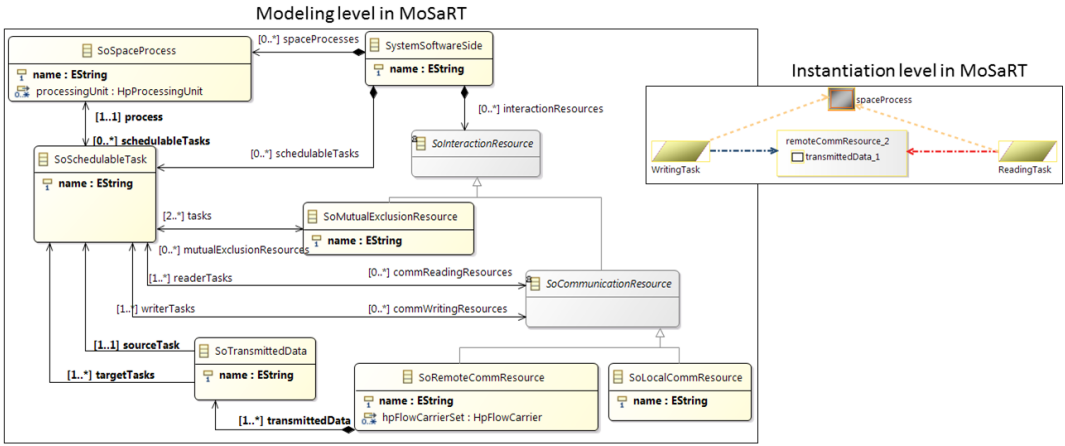


Modélisation en syntaxe concrète - Partie Hardware

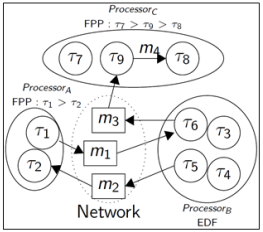




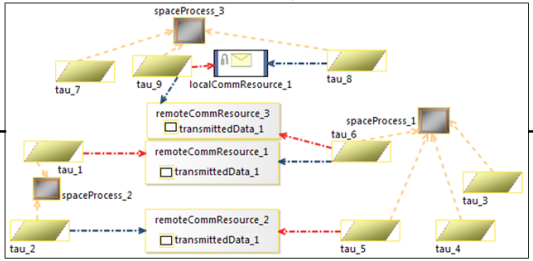
Modélisation en syntaxe concrète - Partie Software



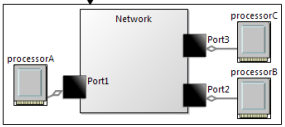
Possibilités



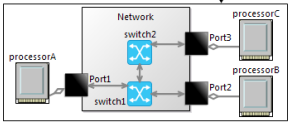
Software part



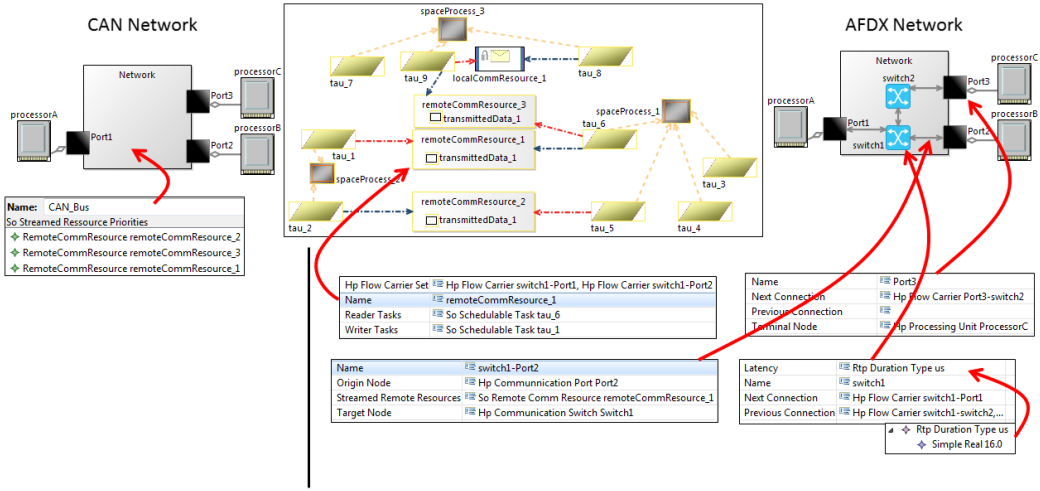
Generic Modeling:



Refined Modeling:



Exemple


CAN Network
Network
processorA
processorC
processorB
Port1
Port2
Port3
Name: CAN_Bus
So Streamed Ressource Priorities
RemoteCommResource remoteCommResource_2
RemoteCommResource remoteCommResource_3
RemoteCommResource remoteCommResource_1
spaceProcess_3
tau_7
tau_9
localCommResource_1
tau_8
remoteCommResource_3
transmittedData_1
spaceProcess_1
tau_6
remoteCommResource_1
transmittedData_1
tau_1
spaceProcess_2
tau_3
remoteCommResource_2
transmittedData_1
tau_2
tau_5
tau_4
AFDX Network
Network
switch2
switch1
processorA
processorC
processorB
Port1
Port2
Port3
Hp Flow Carrier Set Hp Flow Carrier switch1-Port1, Hp Flow Carrier switch1-Port2
Name remoteCommResource_1
Reader Tasks So Schedulable Task tau_6
Writer Tasks So Schedulable Task tau_1
Name Port3
Next Connection Hp Flow Carrier Port3-switch2
Previous Connection
Terminal Node Hp Processing Unit ProcessorC
Name switch1-Port2
Origin Node Hp Communication Port Port2
Streamed Remote Resources So Remote Comm Resource remoteCommResource_1
Target Node Hp Communication Switch Switch1
Latency Rtp Duration Type us
Name switch1
Next Connection Hp Flow Carrier switch1-Port1
Previous Connection Hp Flow Carrier switch1-switch2,...
Rtp Duration Type us
Simple Real 16.0

Papers published

Leveraging Real-Time Network Analyses by Extending a Model-based Framework, AICCSA 2017

Contents

1 Introduction

2 Problématiques

3 Contribution 1 : CONSERT

4 Contribution 2 : Réseaux

5 Conclusion & Perspectives

Perspectives

Conclusion

- Extension du language MoSaRT
- Amélioration du processus de Time4Sys

Perspectives

- Mise en place de la transformation vers les outils d'analyses
- Analyse et modélisation compositionnelle
- Export vers Time4Sys

Conclusion

Questions ?