



**HAL**  
open science

## A runtime adaptation tool for cyber-physical systems

Guillaume Roumage, Selma Azaiez, Stephane Louise

► **To cite this version:**

Guillaume Roumage, Selma Azaiez, Stephane Louise. A runtime adaptation tool for cyber-physical systems. ACACES 2022 - 18th International Summer School on Advanced Computer Architecture and Compilation for High-performance Embedded Systems, Jul 2022, Fiuggi, Italy. , 2022. cea-04520724

**HAL Id: cea-04520724**

**<https://cea.hal.science/cea-04520724>**

Submitted on 25 Mar 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



In this PhD work, we want to provide a fault mitigation capability at run-time to a Cyber-Physical System (CPS) whose design and implementation are based on a Dataflow Model of Computation and Communication (MoCC).

## Context, Motivations & Objectives

### Deterministic Dataflow MoCC-based CPS design

- Deterministic Dataflow Models of Computation and Communication (MoCC) are well fitted to model and form the base of design and implementations of distributed Cyber-Physical Systems (CPS).
- The most popular ones are the Synchronous DataFlow (SDF, [1]) and the Cyclo-Static DataFlow (CSDF, [2]).
- The Dataflow MoCC PolyGraph ([3]) is one of the most expressive of the deterministic dataflow:
  - Additional frequency and phase constraints on **actors** (i.e. distributed tasks).
  - Rational production and consumption rate instead of integers (for clock domain synchronizations).

### Runtime Verification

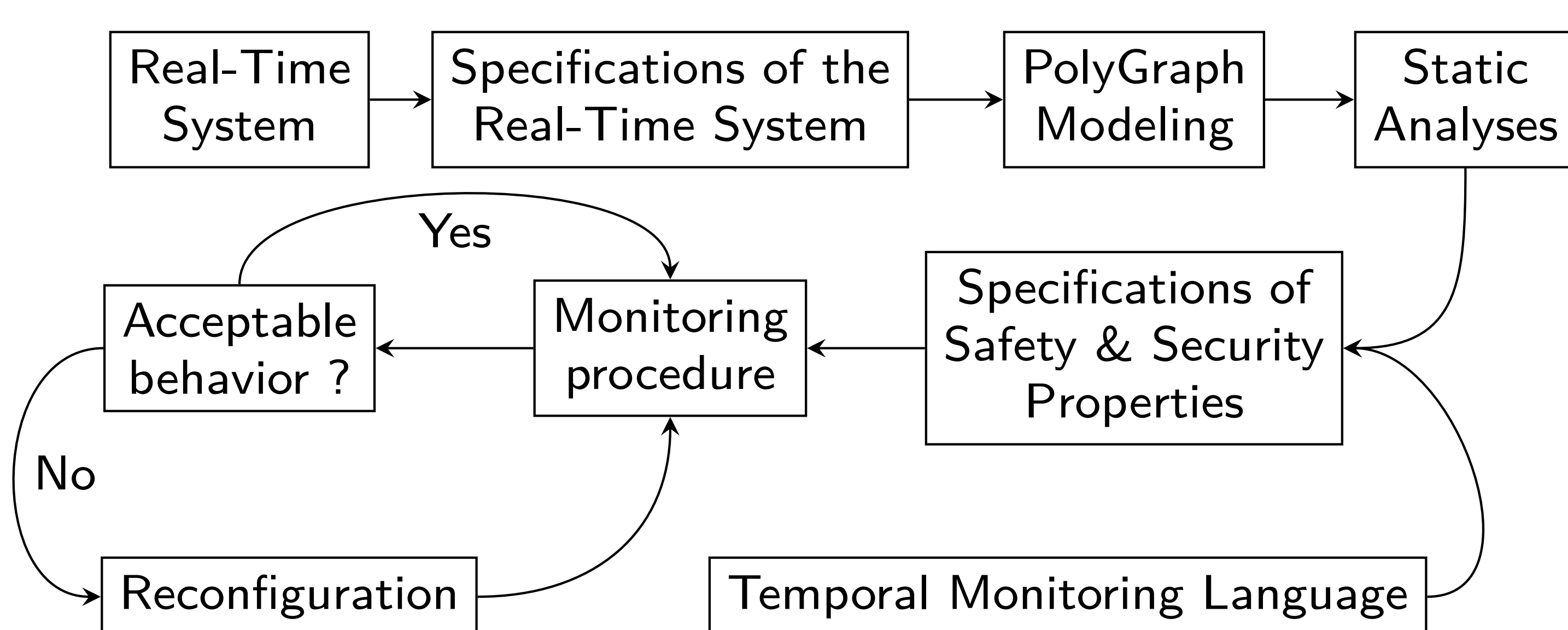
- Runtime verification tools evaluate the system's behavior at run-time.
- The monitoring of global properties has serious drawbacks.
- The system in [4] re-constructs the global trace from local ones.
- Decentralized monitoring of Linear Temporal Logic (LTL) properties is explored [6].

### Targeted Contributions

- A Temporal Monitoring Language tailored for a dataflow CPS design.
- A runtime verification tool whose monitoring specifications are derived from PolyGraph's static analyses.
- The validation of global properties from local monitoring.
- The rise of an early alert and a reconfiguration process that mitigates run-time faults.

## Runtime Adaptation Tool

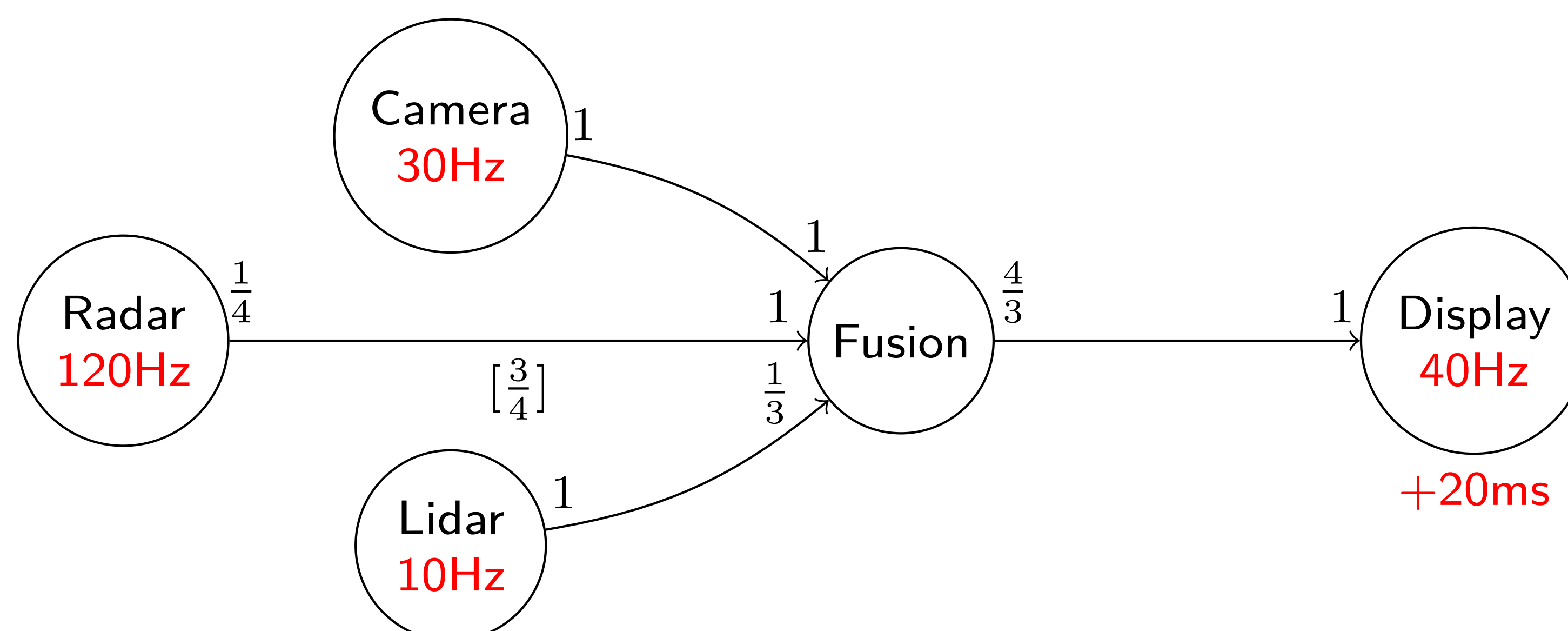
### Targeted Tool Overview



- The system provides a fault mitigation capability at run-time.

### PolyGraph

- Below is the PolyGraph modeling of a data fusion system for ADAS.



The actors with a frequency must fire at that frequency. The first fire can be delayed with a phase, like the Display actor. The rates on the channels represent token production/consumption between the actors at both endpoints. The values between brackets are the initials tokens.

- Dynamic PolyGraph ([5]) enhances PolyGraph ([3]) with a runtime reconfiguration mechanism.

### Use Cases

**The Ingenuity Mars Helicopter:** Position, velocity, and altitude are estimated using an Inertial Measurement Unit, and navigation adjustments are made using a camera.



**An Advanced Driver Assistance System (ADAS):** An odometer measures the current speed, a lidar perceives nearby obstacles, and a stereo camera films the road ahead.



## Perspectives

### Static analyses

- Formalize the Temporal Monitoring Language to specify the actors' (task) execution window, latency, and throughput.
- Define a PolyGraph's mode and the triggering mechanisms.

### Implementation & Evaluation

- Implement run-time verification & reconfiguration procedures, and quantify the monitor's overhead.
- Evaluation with faults injection and ensure the system detects them.

[1] Lee, E., Messerschmitt, D. (1987). Synchronous data flow. Proceedings of the IEEE, 75(9), 1235–1245.

[2] Bilsen, G., Engels, M., Lauwereins, R., Peperstraete, J. (1996). Cyclo-static dataflow. IEEE Transactions on Signal Processing, 44(2), 397–408.

[3] Dubrulle, P., Gaston, C., Kosmatov, N., Lapitre, A., Louise, S. (2019). A Data Flow Model with Frequency Arithmetic. Fundamental Approaches to Software Engineering (Vol. 11424, pp. 369–385).

[4] Falcone, Y., Nazarpour, H., Bensalem, S., Bozga, M. (2021). Monitoring Distributed Component-Based Systems. International Conference on Formal Aspects of Component Software, 153–173.

[5] Dubrulle, P., Gaston, C., Kosmatov, N., Lapitre, A. (2019). Dynamic Reconfigurations in Frequency Constrained Data Flow. Integrated Formal Methods.

[6] Bauer, A., Falcone, Y. (2012). Decentralised LTL Monitoring. International Symposium on Formal Methods.