Towards Correct Transformation: From High-Level Models to Time-Triggered Implementations

H. GUESMI\(^{(3)}\), B. BEN HEDIA\(^{(1)}\), S. BLIUDZE\(^{(2)}\), M. JAN\(^{(1)}\) and S. BENSALEM\(^{(3)}\)

\(^{(1)}\)CEA, LIST, Embedded Real Time Systems Laboratory, 91191 Gif-sur-Yvette, France
\(^{(2)}\)EPFL IC IINFCOM RiSD, Station 14, 1015 Lausanne, Switzerland
\(^{(3)}\)Verimag, Université Joseph Fourrier, 38610 Gieres, France
Introduction

Real-Time (RT) Systems

- Event-Triggered (ET)
- Time-Triggered (TT)

How to build Embedded RT systems of guaranteed quality, in a cost-effective manner??
How to build Embedded RT systems of guaranteed quality, in a cost-effective manner??

Component-based design framework

- Abstracts away implementation details
- Validates the model through different techniques such as formal verification, simulation, and testing
- Targets generic execution model
Introduction

Real-Time (RT) Systems

Event-Triggered (ET)

Time-Triggered (TT)

How to build Embedded RT systems of guaranteed quality, in a cost-effective manner??

Component-based design framework

- Abstracts away implementation details
- Validates the model through different techniques such as formal verification, simulation, and testing
- Targets generic execution model

RTOS-based implementation + TT approach

- No high-level programming models to tackle complexity
- Implement TT execution model
Introduction

Real-Time (RT) Systems

Event-Triggered (ET)  Time-Triggered (TT)

How to build Embedded RT systems of guaranteed quality, in a cost-effective manner??

Component-based design framework

- Abstracts away implementation details
- Validates the model through different techniques such as formal verification, simulation, and testing
- Targets generic execution model

RTOS-based implementation + TT approach

- No high-level programming models to tackle complexity
- Implement TT execution model
RT-BIP Framework [1]:

- Structure of a real-time BIP model:
Background Concepts

- RT-BIP Framework [1]:
  - Structure of a real-time BIP model:
    - Priorities
    - Interactions
    - Behavior
    - Mechanism for Conflict resolution between interactions
    - Connectors representing interactions
    - Timed automata

- TCA: Computation model of TT tasks in PharOS [2][5]:
  - The temporal behavior of a task is specified using a directed graph

Diagram:

- Synchronization node
- No constraint node
- "After" node
- "Before" node
1/ Step1 [3]

- TT-BIP = tasks components + communication components + unidirectional interactions.

**Approach**

- RT-BIP Model
- TT-BIP Model
- Executable code
- TT platform

1 - Model-to-model Transformation

TT paradigm
1/ Step1 [3]

- TT-BIP = tasks components + communication components + unidirectional interactions.

- TT-BIP is not easily translated to an executable
1/ Step 1 [3]
- TT-BIP = tasks components + communication components + unidirectional interactions.

2/ Step 2:
- Challenges:

Constraints only on the **start** instant of an action

Can be constrained both the **release** and **deadline** instants of an action

1. $1 \leq x \leq 4$
2. $x \leq 5$
3. Reset $x$
1/ Step1 [3]
- TT-BIP = tasks components + communication components + unidirectional interactions.

2/ Step2:
- Challenges:

Constraints only on the **start** instant of an action

Can be constrained both the **release** and **deadline** instants of an action

Nic 1: **Absolute** labeling of constraints

Nic 2: **Relative** labeling of constraints

---

![Diagram](image.png)
1/ Step1 [3]
- TT-BIP = tasks components + communication components + unidirectional interactions.

2/ Step2:
- Challenges:

1. Constraints only on the start instant of an action
2. Absolute labeling of constraints
3. Interactions = data transfer + synchronization between sending and receiving actions

Can be constrained both the release and deadline instants of an action
Relative labeling of constraints
Desynchronized interaction: sender provides new values at each synchronization point + Receivers can consult these values when their current time is equal or higher to the defined visibility dates.
1/ Step1 [3]

- TT-BIP = tasks components + communication components + unidirectional interactions.

2/ Step2:

- Challenges:
  - Constraints only on the start instant of an action
  - Absolute labeling of constraints
  - Interactions = data-
synchronization between sending and receiving actions

Algorithm details are given in the poster

Constraints can be constrained both the release and deadline instants of an action.

Relative labeling of constraints

Desynchronized interaction = sender provides new values at each synchronization point + Receivers can consult these values when their current time is equal or higher to the defined visibility dates.
1/ Step1 [3]
- TT-BIP = tasks components + communication components + unidirectional interactions.

2/ Step2:
- Challenges:
- Correctness proof:
  - Expressing semantics of each model in terms of Labelled Transition system (LTS).
  - Proving equivalence between two LTSs by using bisimulation technique.
    → trace equivalence
Thank you!