Towards Dynamic Composition of Dependable Embedded Systems

Leandro Batista Ribeiro
lbatistaribeiro@tugraz.at

Institute of Technical Informatics
Embedded Automotive Systems Group
Graz University of Technology
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems

Software

Modules
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems
Towards Dynamic Composition of Dependable Embedded Systems

Integration at Runtime

Software
Towards Dynamic Composition of Dependable Embedded Systems

Integration at Runtime

Module

Transmission

Software
Towards Dynamic Composition of Dependable Embedded Systems

Integration at Runtime

Module

Transmission

Software
Inherent Problems
Inherent Problems

Memory
Inherent Problems

Memory
Inherent Problems

Memory
Inherent Problems

Memory
Inherent Problems

Memory

Dependencies
Inherent Problems

Memory

Dependencies
Inherent Problems

- Memory
- Dependencies
- Behavior
Inherent Problems

- Memory
- Dependencies
- Behavior

Functional Requirements
Inherent Problems

Memory

Dependencies

Behavior

(Non-)Functional Requirements
Research Question

How to partially update embedded systems and guarantee the correctness of the resulting software composition?
Research Question

How to partially update embedded systems and guarantee the correctness of the resulting software composition?
Partial Updates
SmartOS and Update Protocol

Memory

Dependencies
Partial Updates
SmartOS and Update Protocol

Memory

Dependencies

EWSN'20

SmartOS
Sustainable modular adaptive real-time Operating System
Partial Updates
SmartOS and Update Protocol

Memory

Dependencies

SmartOS
Sustainable modular adaptive real-time Operating System

Server
Device

EWSN’20
Partial Updates

Maintainability

Memory
Partial Updates

Maintainability

Memory

Defrag.
Partial Updates

Maintainability

- Memory
- Defrag.
- Update

or
Partial Updates

Maintainability

Memory

Defrag. or Update
Partial Updates
Maintainability

Memory

Defrag.
or

Update
Partial Updates

Maintainability

Memory

Defrag. or Update

CODES’21

HW Support
Partial Updates

Maintainability

Memory

Defrag. or Update

CODES’21

HW Support

HW Relocation
Correctness

Our Focus

Behavior

(Non-)Functional Requirements
Correctness

Our Focus

- Real-Time

Behavior

(Non-)Functional Requirements
Correctness

Our Focus

- Real-Time
- Liveness

Behavior

(Non-)Functional Requirements
Correctness

Our Focus

- Real-Time
- Liveness
Correctness

Our Focus

- Real-Time

- Liveness
  - Freedom of Starvation

Dining Philosophers

[DP]
Correctness

Our Focus

- Real-Time
- Liveness
  - Freedom of Starvation
Correctness
Our Focus

- Real-Time
- Liveness
  - Freedom of Starvation

Dining Philosophers
Correctness
Our Focus

- Real-Time
- Liveness
  - Freedom of Starvation

Dining Philosophers

[DP]
Correctness

Our Focus

- Real-Time

- Liveness
  - Freedom of Starvation
  - Freedom of Deadlock

Dining Philosophers
Correctness
Our Focus

- Real-Time

- Liveness
  - Freedom of Starvation
  - Freedom of Deadlock
Correctness

Formal Methods

- Model and Verify Software Behavior
Correctness

Formal Methods

- Model and Verify Software Behavior
Correctness

Formal Methods

- Model and Verify Software Behavior
Correctness
Formal Methods

- Model and Verify Software Behavior
Correctness

Formal Methods

- Model and Verify Software Behavior

- Examples of Requirements
  - A deadlock must never occur
  - A given error location must never be reached
  - Internal lists must be sorted at all times
Correctness
Layered Compositional Model
Application

SmartOS
Sustainable modular adaptive real-time Operating System
Correctness
Layered Compositional Model

Application

OS Interface

SmartOS
Correctness

Layered Compositional Model

Application

OS Interface

SmartOS
Sustainable modular adaptive real-time Operating System
Correctness
Layered Compositional Model

Application

OS Interface

SmartOS
Sustainable modular adaptive real-time Operating System
Correctness
Layered Compositional Model

Application

OS Interface

SmartOS
Sustainable modular adaptive real-time Operating System
Correctness

Layered Compositional Model

Application

OS Interface

SmartOS
Sustainable modular adaptive real-time Operating System
Correctness

Verification Caveats

Application

OS Interface

SmartOS
Sustainable modular adaptive real-time Operating System
Correctness

Verification Caveats

- Successful Verification
Correctness
Verification Caveats

- Successful Verification
- OS layer not fully verified
Correctness
Verification Caveats

- Successful Verification
  - OS layer not fully verified

- Unsuccessful Verification
Correctness

Verification Caveats

- Successful Verification
  - OS layer not fully verified

- Unsuccessful Verification
  - Unknown source of error
Correctness

Verification Caveats

- Successful Verification
  - OS layer not fully verified

- Unsuccessful Verification
  - Unknown source of error

- Our Approach
  - Abstract application
Correctness

Verification Caveats

- Successful Verification
  - OS layer not fully verified

- Unsuccessful Verification
  - Unknown source of error

- Our Approach
  - Abstract application
    - Cheap full OS layer verification
Update Overview

Server

Device

Update Request
Update Overview
Update Overview

Server

Update Request

... Composition Correct: Install

Device
Update Overview

Server

Update Request

... Composition Incorrect: Abort

Device
Potential Improvements & Future Work

- Partial Updates: Technical Features

- Formal Verification: Research
Potential Improvements & Future Work

- Partial Updates: Technical Features
  - Security Aspects

- Formal Verification: Research
Potential Improvements & Future Work

- Partial Updates: Technical Features
  - Security Aspects
  - Execution State Transfer

- Formal Verification: Research
Potential Improvements & Future Work

- Partial Updates: Technical Features
  - Security Aspects
  - Execution State Transfer

- Formal Verification: Research
  - Other Requirements
Potential Improvements & Future Work

- Partial Updates: Technical Features
  - Security Aspects
  - Execution State Transfer

- Formal Verification: Research
  - Other Requirements
  - Code-Model Mapping Correctness
Potential Improvements & Future Work

- Partial Updates: Technical Features
  - Security Aspects
  - Execution State Transfer

- Formal Verification: Research
  - Other Requirements
  - Code-Model Mapping Correctness
  - Automatic Code Generation
Potential Improvements & Future Work

- Partial Updates: Technical Features
  - Security Aspects
  - Execution State Transfer

- Formal Verification: Research
  - Other Requirements
  - Code-Model Mapping Correctness
  - Automatic Code Generation
  - State Space Explosion
Summary

- Work divided in two major parts
  1. How to support partial updates
Summary

- Work divided in two major parts
  1. How to support partial updates
Summary

- Work divided in two major parts
  1. How to support partial updates
Summary

- Work divided in two major parts
  1. How to support partial updates
Summary

- Work divided in two major parts
  1. How to support partial updates
  2. How to guarantee correctness of software compositions
Summary

- Work divided in two major parts
  1. How to support partial updates
  2. How to guarantee correctness of software compositions
Summary

- Work divided in two major parts
  1. How to support partial updates
  2. How to guarantee correctness of software compositions
Summary

- Work divided in two major parts
  1. How to support partial updates
  2. How to guarantee correctness of software compositions
Summary

- Work divided in two major parts
  1. How to support partial updates
  2. How to guarantee correctness of software compositions
     - Current Focus
       - Liveness
       - Real-Time
Summary

- Work divided in two major parts
  1. How to support partial updates
  2. How to guarantee correctness of software compositions
    - Current Focus
      - Liveness
      - Real-Time

- Future Work
  - Partial Updates: Technical Improvements
Summary

- Work divided in two major parts
  1. How to support partial updates
  2. How to guarantee correctness of software compositions
    - Current Focus
      - Liveness
      - Real-Time

- Future Work
  - Partial Updates: Technical Improvements
  - Correctness: Research Challenges
Thank you!

Leandro Batista Ribeiro
lbatistaribeiro@tugraz.at

Institute of Technical Informatics
Embedded Automotive Systems Group
Graz University of Technology
References

