ModelMap: A Model-based Multi-domain Application Framework for Centralized Automotive Systems

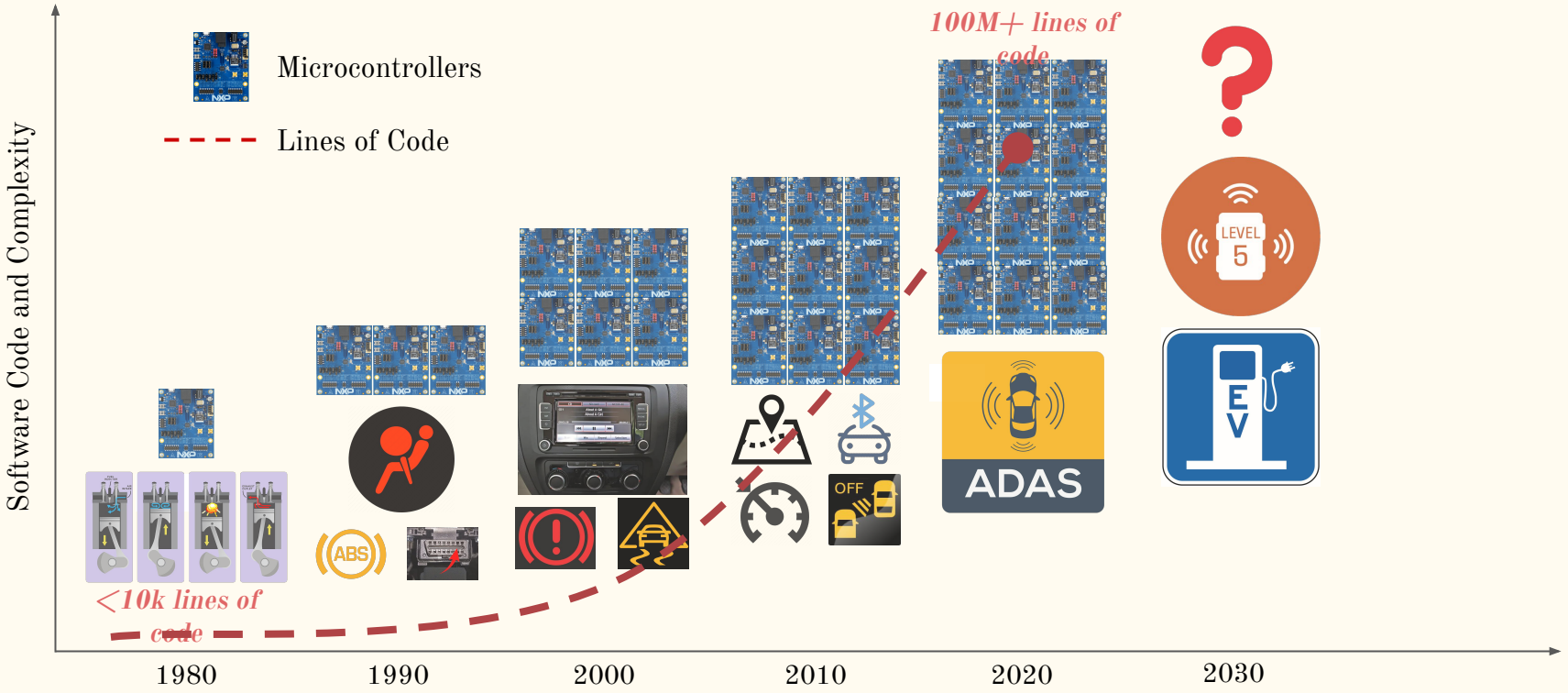
—

Soham Sinha*, Anam Farrukh, and Richard West
Department of Computer Science, Boston University
(*presently at Nvidia)

Vehicle Functions in Modern Cars

- Instrument Cluster
- In-vehicle Infotainment
- Advanced Driver Assistance Systems (ADAS)
- Heating, Ventilation and Air-conditioning (HVAC)
- Battery Management System (BMS)
- Torque Vectoring
- ...

Proliferation of Software and Electronics in Vehicles



Source: Mckinsey 2018 Report

Reconceptualized from BEVA 2020 Slides by Prof. West

Integrated Vehicle Management System (VMS)

- Heating, Ventilation and Air-conditioning (HVAC)
- Powertrain
- Battery Management

...

...

- Instrument Cluster (IC)
- In-vehicle Infotainment (IVI)
- Advanced Driver Assistance Systems (ADAS)

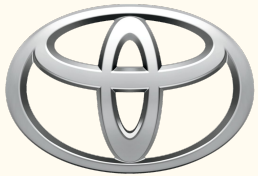
...

...

Integrated VMS: Examples

- DriveOS™ by Drako Motors (Not the same as DRIVE by Nvidia)
- MB.OS by Mercedes-Benz
- AreneOS by Toyota
- Ultifi by General Motors
- ...

Mercedes-Benz

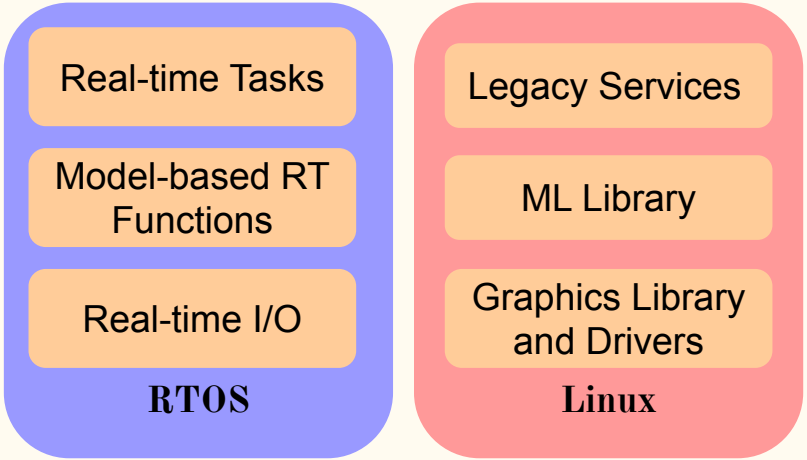


Integrated VMS: Examples

- **DriveOS** by Drako Motors
- MB.OS by Mercedes-Benz
- AreneOS by Toyota
- Ultifi by General Motors
- ...

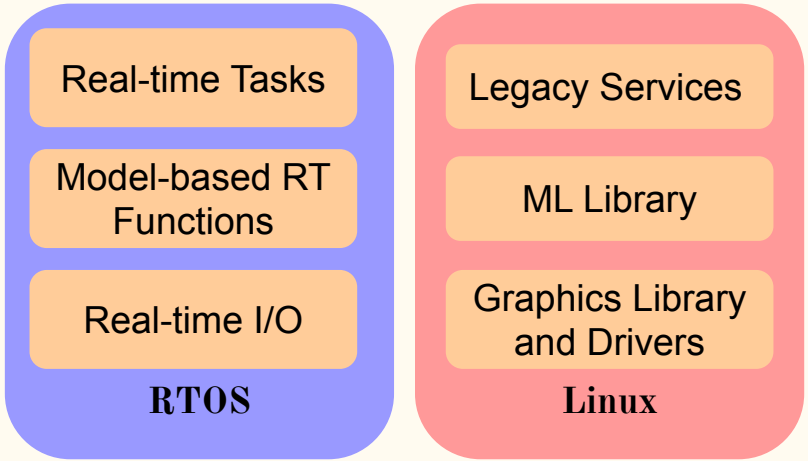
Design Overview of DriveOS

- Co-existence of a Real-time Operating System (RTOS) and a General-purpose Operating System (GPOS)
- Enabled by machine virtualization features
- RTOS is used for critical applications
  - Powertrain
  - HVAC
- GPOS is used for non-critical applications
  - Instrument Cluster
  - In-vehicle Infotainment
- Multiple Criticality Domains



Mixed-criticality Applications in DriveOS

How to design an application that spans multiple domains or operating systems?



Objective

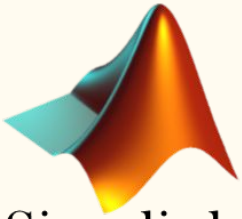
To develop a mixed-criticality multi-domain application development framework in a model-based language for automotive functions

Why model-based language?

Vehicle Applications using Model-based Language

Advantages:

- Familiar to the automotive engineers
- Correctness by construction
- Testing in simulation before large-scale deployment
- Model-in-the-loop, processor-in-the-loop, hardware-in-the-loop testing

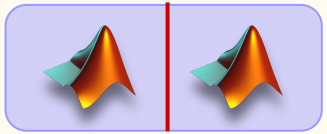


Challenges in Model-based Language

- Model-based languages usually target simple RTOSs running on an ECU
- No support for *mixed-criticality* applications
- No support for multiple operating systems or domains like in a DriveOS VMS

ModelMap

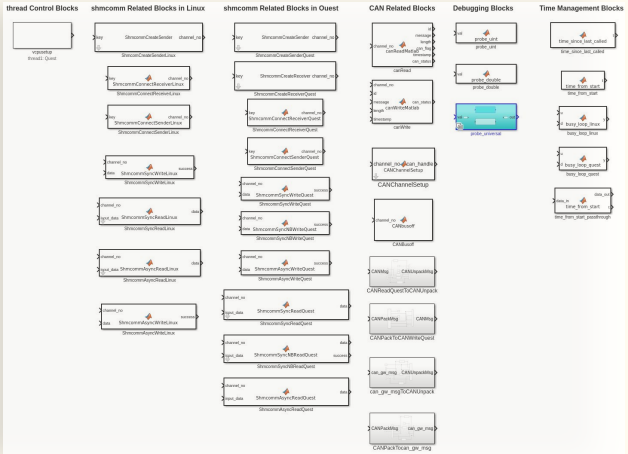
ModelMap



- ModelMap is a **model**-based **m**ulti-domain **ap**plication framework for DriveOS
- Set of design tools in Simulink for DriveOS applications
- Shows how to map a Simulink model to DriveOS OS domains
- Encapsulates multi-domain models in a nested binary
  - Multiple ABI in a single binary executable

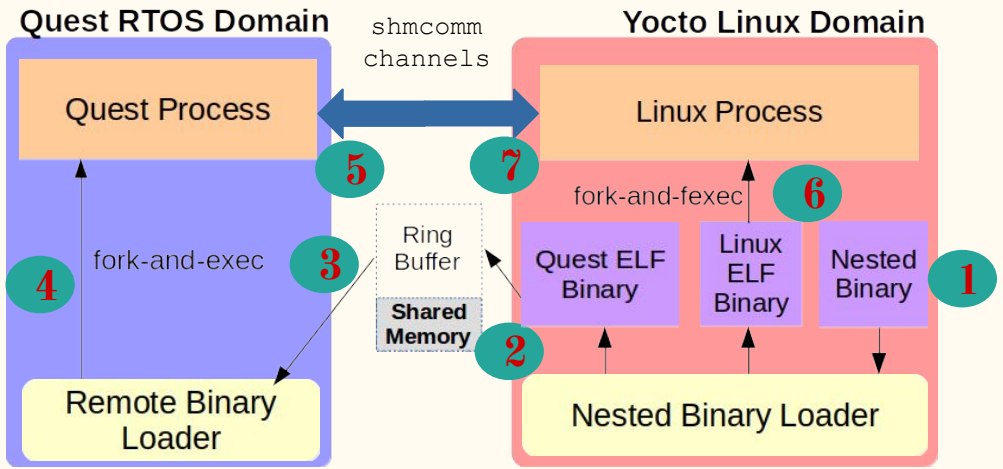
ModelMap Design Tools

- **Thread Blocks**
  - Real-time periodic threads in Quest RTOS and Linux domains of DriveOS
- **Inter-task Communication Blocks**
  - Synchronous and Asynchronous message passing
  - Cross-domain message passing
- **Real-time I/O Blocks**
  - To the vehicle CAN bus



ModelMap: Nested Binaries

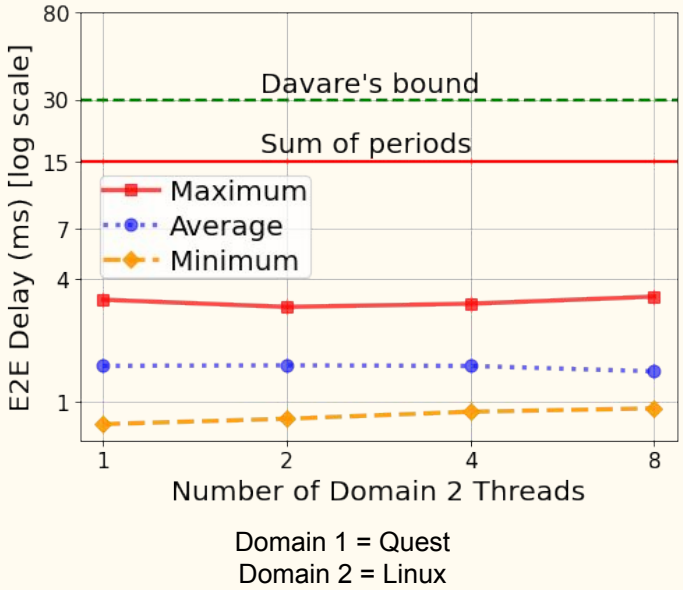
- Multiple ELF binaries in a single executable ELF binary
- Metadata of mapping between a binary to runtime domain
- Nested Binary Compiler and Nested Binary Loader



Evaluation

Predictable Latency: A Scalable CAN Gateway

- CAN Gateway in Quest RTOS connected to upto 8 application threads in Linux
- End-to-end delay of CAN messages
  - Compare with theoretical upper bounds:
    - Sum of periods
    - Davare's bound (double of sum of periods) [DZN07]



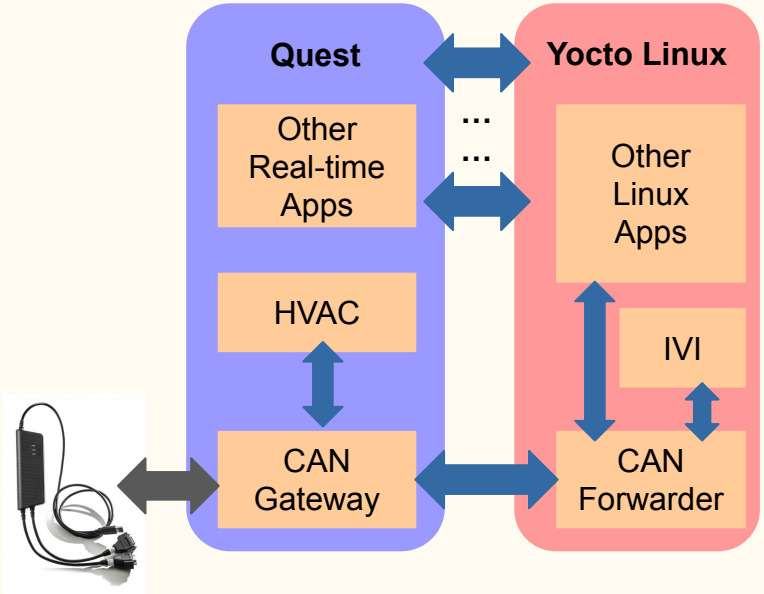
E2E delay remain under target bounds

Temporal isolation between threads

[DZN07] Davare et al. Period Optimization for Hard Real-time Distributed Automotive Systems. **DAC 2007**

Functional Correctness: Drako Motors' EV HVAC

- Designed as a Simulink model
  - Model-in-the-loop (MIL)
- Imported to DriveOS using ModelMap
  - Hardware-in-the-loop (HIL)
- Verified functional correctness by matching the outputs of MIL and HIL executions



Conclusions

- ModelMap presents the first open model-based multi-domain application framework for an integrated VMS
- Multi-domain code generation in Simulink
- Encapsulation of multiple types of binary executables in a nested binary
- Experimental evaluation empirically demonstrates predictable end-to-end latency and functional correctness of vehicle applications

Thanks!
Questions?

Reference

- Some slides are taken from Soham Sinha's PhD Thesis Presentation
- Some images are taken from Google Images