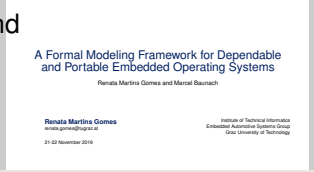
A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

Renata Martins Gomes and Marcel Baunach

Renata Martins Gomes
renata.gomes@tugraz.at

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

21-22 November 2019



A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

Long title for a 10-minute talk!
Let's go by parts.

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

I guess you all know what OS means...

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

Not talking about desktop Linux, Windows...
Focus on IoT:
devices embedded into our environment,
ranging from small and simple sensor devices to complex ECUs inside cars.

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

The OS must be dependable:
safe, secure, available, reliable, etc...

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

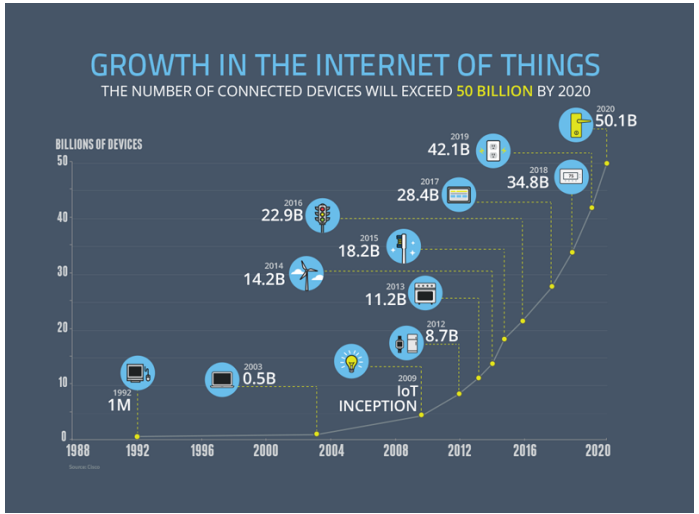
A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

Isn't it solved yet?
you can separate code: compiled for all devices (**high-level**)
from code that must be rewritten or adapted (**low-level**)

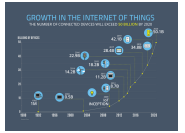
To port low-level, you must be an expert on both sides: HW and SW

- How to test it? You know there are bugs hiding!
- Do you even know what you must implement? Is there any specification?
- And how can you assess the dependability of each port?
- Including functional and non-functional behavior? Do all ports behave as they should?

Ah but we only do this for a couple of devices! Is it really? Or the OSs only support a couple of devices because they are not that portable?
Much better if it can run anywhere (like Android?)



A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems



- Billions of devices
- Variety of devices
- dependability requirements
- specialized and reconfigurable HW

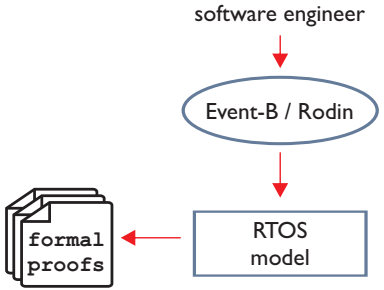
Finally, what is our plan to solve this?
Don't get scared, but... formal methods!

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

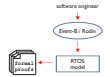
A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

Model the OS's intended low-level functionality,
automatically generate low-level code for target devices.

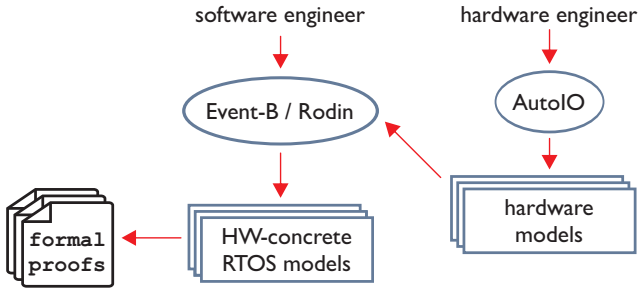


A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems

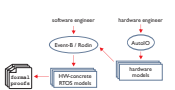


Formally model the RTOS considering generic HW
bonus: FM improve SW quality. We can verify the model for safety, correctness, non-functional properties...

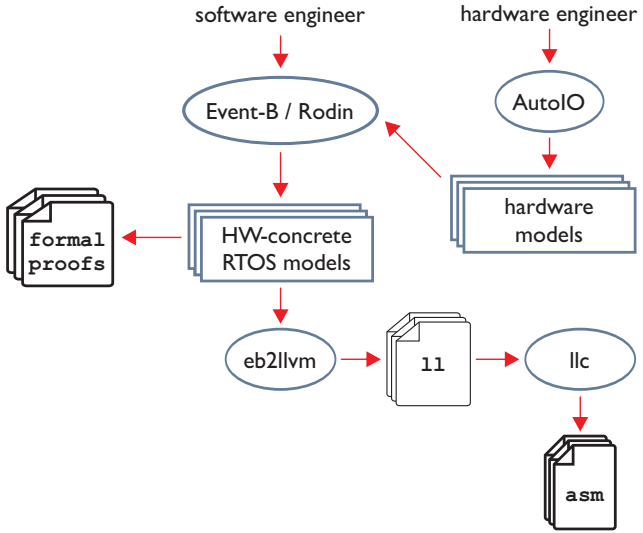
Event-B: formal method based on set theory and state transitions.

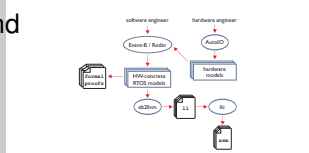


A Formal Modeling Framework for Dependable and Portable Embedded Operating Systems



With generic model ready and verified, we instantiate it with HW models and verify again
(modeling part under review)





Code generation (work in progress)
inputs welcome!

convert math language (Event-B) into LLVM Intermediate Representation
can be compiled to target code.

can already generate some code, but some HW-specifics from the model must still be translated
(e.g. registers cannot always be selected by the compiler. Intrinsics?

Thank you!

Ideas, feedback, questions? Get in touch!
renata.gomes@tugraz.at

Thank you!
Ideas, feedback, questions? Get in touch!
renata.gomes@tugraz.at