A virtual On Board Control Unit for system tests

Ove Kalkan (ove.kalkan@ese.de)
Agenda

**Introduction:**
- What is an OBCU
- System Test Approach

**Virtualization**
- Approach
- Realization

**Conclusion**
- Results
- Virtual OBCU in VER/VAL under EN 50126/50128
- Next steps
What is an OBCU?

OBCU = On Board Control Unit

Non-Vital

- Driving Assistance
- Control of Communication
- Control of Information Systems

Vital

- Speed Supervision
- Control Doors & Platforms
- Control of Coupling & Train Integrity

A virtual OBCU for system tests - Introduction
The Vehicle

A virtual OBCU for system tests - Introduction
System test for real OBCUs

A virtual OBCU for system tests - Introduction
Disadvantages

Costs:
- Investment costs
- Operational costs

Lack of Availability:
- Long ordering and manufacturing time
- Limited number of test systems

Inflexibility:
- Long term planning required
- Only suitable for limited number of versions and projects
Virtualization

„In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices and computer network resources.“

The virtual OBCU

A virtual OBCU for system tests - Virtualization
Objective

Functional Equivalence:
- The virtual OBCU shall behave like the real one on functional level.
- To be proven by test cases.
- OBCU software shall not be modified.
System test for virtual OBCUs

Test Environment

- Interface Layer
  - Test Control

Virtual OBCU

- Virtual Machine
  - Interface Computer
    - Virtual Machine
      - ATO
    - Virtual Machine
      - ATP

A virtual OBCU for system tests - Virtualization
Basic Principle: Interface Adapter

Example: Digital IO

e.g. Door Control, Emergency Brake
Basic Principle: Interface Adapter

Example: Digital IO

Test Environment

- Interface Layer
- Test Control

MODBUS (IEC 61158)

e.g. Door Control, Emergency Brake

ATP

A virtual OBCU for system tests - Virtualization
Basic Principle: Interface Adapter

Example: Digital IO

Test Environment

- Interface Layer
  - Test Control

A virtual OBCU for system tests - Virtualization
**Technical Challenge**

**Software:**
- e.g. timing conditions that require real-time kernel support.

**Hardware:**
- hardware with incomplete specification (e.g. closed source)
- tight timing conditions (e.g. FPGAs).

A virtual OBCU for system tests - Virtualization
Technical Challenge

Software:
- e.g. timing conditions that require real-time kernel support.

Hardware:
- hardware with incomplete specification (e.g. closed source)
- tight timing conditions (e.g. FPGAs).
System setup

Virtualization Host:
- Intel Xeon E31240, 3.3GHz (8 cores), 32GB RAM
- PROXMOX VE 4.4 (Debian 8 based Linux)

Virtualization:
- QEMU/KVM to run Linux based VMs
- QEMU: Open Source and extending by custom emulated hardware is possible

Emulation:
- Custom emulator for embedded fail-safe system used to operate the ATP, running in a Linux VM
Virtualization Server

Virtual OBCU

Virtual Machine

ATO

Virtual Machine

ATP

Virtual Machine

Interface Computer

Virtual Machine

ATP

Virtual Machine

ATO

Virtual Machine

Interface Computer

Virtual Machine

ATO

Virtual Machine

Interface Computer

Virtual Machine

ATP

Virtual Machine

ATO

Virtual Machine

Interface Computer

Virtual Machine

ATP

Virtual Machine

ATO

Virtual Machine

Interface Computer

Virtual Machine

ATP

Virtual Machine

ATO

Virtual Machine

Interface Computer

Virtual Machine

ATP

Test Environment

Virtual Machine

Interface Layer

Virtual Machine

Interface Layer
Results – does it work?

**Manual Testing:**
- The virtual vehicle can be driven manually in the test environment like the real OBCU based vehicles.

**Automated Testing:**
- Final proof is in progress.
Can we use it for VER/VAL?

Full VER/VAL as required in EN 50126/50128 will not be possible.

**Limitations:**

- Hardware is replaced by an emulated hardware, which may not be the same in all aspects of behavior.
- Non-functional aspects may not be equivalent – such as timings, especially under real-time conditions.
- Missing “Tool Qualification” of Hardware, Operating System and Virtualization Environment according to EN 50126/50128.
How can we use it for VER/VAL?

VER/VAL can benefit from virtualization in ...

**Benefits:**

- Support during development, e.g. component or test cases.
- Functional testing can be done with virtual OBCUs in pre-tests.
- Test of configuration data, e.g. track, missions.
- Complex and cost extensive scenarios, e.g. coupling of many vehicles, can be examined with virtual vehicles.
- Support Test Life Cycle
A virtual OBCU for system tests - Conclusion
Next steps ...

Scale up the vehicle cluster:

- Increase the number of virtual vehicles.
- New functions, e.g. support coupling.
- Improve cluster management, e.g. Setup, Load Balancing and Life Migration
- Tool Qualification accordingly to EN 50126/50128
- Support more versions and projects.
Questions & Answers

ESE Engineering und Software – Entwicklung GmbH

Unternehmenssitz Braunschweig

Am Alten Bahnhof 16
D-38122 Braunschweig

Telefon: +49 531 23880-30
Telefax: +49 531 23880-33
E-Mail: info@ese.de
Ove Kalkan (year 1971)

I’m for about 10 years employed at the ESE Engineering and Softwareentwicklung GmbH in the position as software architect and project manager. My work focusses on software development in rail automation, aviation and others. Special interests are software development processes, model driven software development and testing/quality. Formerly I have been working in the multi media consumer industry and for the Technical University of Braunschweig in research and development of image processing and photogrammetry applications.