Formalising Device Driver Interfaces

Leonid Ryzhyk  Ihor Kuz  Gernot Heiser
NICTA and the University of New South Wales
Why a new language?

- To specify service contracts between the OS and drivers
- To detect contract violations
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus

Process:
- txPacket
Example: a USB-to-Ethernet controller driver

IP stack

USB-to-Ethernet driver

USB bus

- txPacket
- requestStart
Example: a USB-to-Ethernet controller driver

IP stack

USB-to-Ethernet driver

USB bus
Example: a USB-to-Ethernet controller driver

IP stack

USB-to-Ethernet driver

txPacket

requestStart

requestComplete

USB bus

requestStart

txPacket
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus

Events:
- txPacket
- requestStart
- txPacket
- requestComplete
- txPacketComplete
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus

**Transactions:**
- `txPacket` from IP stack to USB-to-Ethernet driver
- `requestStart` from USB-to-Ethernet driver to USB bus
- `txPacketComplete` from USB-to-Ethernet driver to IP stack
- `requestComplete` from USB bus to USB-to-Ethernet driver
- `txPacketComplete` from USB bus to IP stack
Example: a USB-to-Ethernet controller driver
Example: a USB-to-Ethernet controller driver

IP stack

USB-to-Ethernet driver

USB bus

- txPacket
- requestStart
- txPacket
- requestStart
- sleep
- endpointAbort
Example: a USB-to-Ethernet controller driver
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus

- txPacket
- requestStart
- sleep
- endpointAbort
- requestAbort
- txPacketAbort
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus

- txPacket
- txPacket
- sleep
- txPacketAbort
- txPacketAbort
- sleeping
- requestStart
- requestStart
- endpointAbort
- requestAbort
- requestAbort
- sleeping
Example: a USB-to-Ethernet controller driver

- IP stack
  - txPacket
  - sleep
- USB-to-Ethernet driver
  - requestStart
  - unplugged
- USB bus
  - txPacket Abort
  - request Abort
  - sleeping
Example: a USB-to-Ethernet controller driver

- IP stack
- USB-to-Ethernet driver
- USB bus

Events:
- txPacket
- txPacket
- sleep
- txPacketAbort
- requestStart
- requestStart
- unplugged
- requestAbort
Example: a USB-to-Ethernet controller driver
Example: a USB-to-Ethernet controller driver

IP stack
- txPacket
- sleep
- txPacketAbort
- stopComplete

USB-to-Ethernet driver
- requestStart
- unplugged
- requestAbort

USB bus
- requestStart
- txPacket
- txPacketAbort
- stopComplete
... the kernel APIs for plug-and-play and power management are quite complex. Often, kernel experts in these areas would disagree with one another about subtle points in the rules.

Ball et.al, Thorough static analysis of device drivers
The DPSM language

**Driver Protocol State Machines** (DPSM) – a formal language for specifying driver protocols

Requirements:
- Expressiveness
- Understandability

Current implementation – on top of the L4/Iguana embedded OS
The basics:

- Drivers communicate with the OS via messages
- The interface of a driver is decomposed into protocols
  - Examples: Lifecycle, PowerManagement, USBHub
- A protocol is specified using FSM-like syntaxes
Example: the Lifecycle protocol

Lifecycle protocol

OS

Driver

start

startComplete

startFailed

stop

stopComplete
Example: the Lifecycle protocol

Lifecycle protocol

OS

Driver

- start
- startComplete
- unplugged
- stopComplete
Example: the Lifecycle protocol

Lifecycle protocol

OS

Driver

| start
| startComplete
| unplugged
| stopComplete

Diagram:

- init
- plugged
- starting
- running
- stopping
- unplugged

Events:
- ?start
- !startFailed
- ?unplugged
- !stopComplete
- !stopComplete
Example: the USBHub protocol

```c
!portReportFeatures/
new USBHubPort(params.portNum)
```

```
ports_init
```
Example: the USBHub protocol

```java
!portReportFeatures/
new USBHubPort(params.portNum)
```

ports_init
Example: the USBHub protocol

```java
ports_init

!portReportFeatures/
new USHUbPort(params.portNum)
```
Example: the USBHub protocol

```java
!portReportFeatures/
new USBHubPort(params.portNum)
```

ports_init
Example: the USBHub protocol

```c
portReportFeatures/
new USBHubPort(params.portNum)
```

ports_init
Other DPSM features:

- Protocol variables
- Timeout states
Runtime failure detection

IP stack

USB-to-Ethernet driver

USB framework
Runtime failure detection

IP stack

USB-to-Ethernet driver

USB framework
Runtime failure detection

IP stack

USB-to-Ethernet driver

USB framework

EthernetController protocol SM

USB interface protocol SM
Static verification of device drivers

Can we statically check drivers against protocol specifications?

- Correctness may depend on the hardware interface
- Probably beyond reach of current model checkers for C/C++
Project status

Where we are today:

- Defined DPSM syntax and semantics
- Specified and implemented protocols for Ethernet and USB frameworks
- Current work: design and verification of the USB framework
- Future work: provably correct driver recovery algorithms
Thank you!
Building a formally verified USB framework:

- The framework is written in a FSM-based language (Esterel)
- The design-for-verification approach
- Module interfaces are specified in DPSM
Introduction to the I/O system architecture

- Ethernet controller
- USB hub
- Ethernet framework
- USB framework
- USB HC driver
- PCI bridge
- PCI framework
- PCI bridge driver
- OS Kernel
- Ethernet family protocol
  - Ethernet driver
  - USB HC driver

- CPU
- Memory controller
- PCI bus
- USB bus
- USB HC
- PCI bridge
- CPU
Introduction to the I/O system architecture

- CPU
- Memory controller
- PCI bridge
- USB HC
- Ethernet controller

- USB bus
- PCI bus

- PCI framework
- Ethernet framework
- USB framework
- Ethernet driver
- USB HC driver
- Ethernet family protocol
- Device-specific protocol
- USB bus protocol
- PCI bridge driver
- OS Kernel
- Ethernet controller
- USB HC
Introduction to the I/O system architecture