

# Integrating PVSio-web with Overture

**Paolo Masci**

( [p.m.masci@qmul.ac.uk](mailto:p.m.masci@qmul.ac.uk) )

Queen Mary University of London  
United Kingdom

Joint work with

**Luis Diogo Couto & Peter Gorm Larsen**

# PVSio-web

## Prototyping environment based on formal models



## Example prototypes generated with PVSio-web



# Main contributions

1. **Basic integration of PVSio-web modelling and prototyping environment with Overture**
  
2. **Demonstrative example based on a medical device**
  - Generation of VDM model from a graphical diagram
  - Prototyping of a realistic device based on the model
  - Example analysis using PVSio-web and Overture

# Outline of the talk

- **PVSio-web**
  - Motivations, design, applications & impact stories
- **Our VDM extension for PVSio-web**
  - Demonstrative example
- **Demonstrative examples**
  - Automatic generation of realistic prototypes from formal models
  - Lightweight formal analysis using PVSio-web

# PVSio-web

Toolkit for multi-disciplinary design and evaluation of interactive (human-computer) systems

## Graphical animation environment



+

## Formal Analysis Tools



Y  
2013



2014



2015



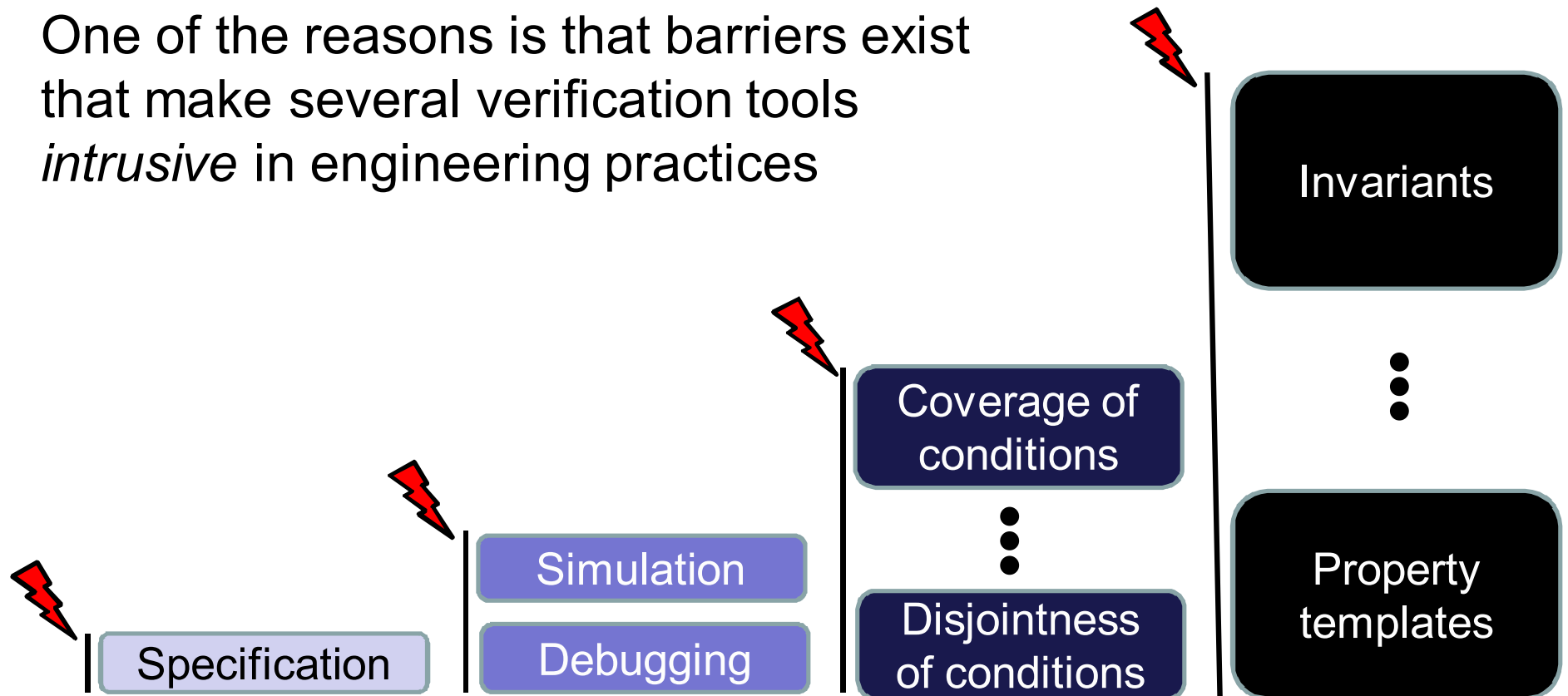
KeYmaera

2016

# Engineering practices

The use of verification tools is neglected by manufacturers when designing user interfaces

One of the reasons is that barriers exist that make several verification tools *intrusive* in engineering practices



## Example workflow that integrates verification tools in user interface design

**Step 1.** Specify a model of the user interface

**Step 2.** Engage with engineers and domain experts

**Step 3.** Iterate step 1 and 2 until agreement is reached

**Step 4.** Verify properties. Go to to step 1 to fix identified issues.

# Example workflow that integrates verification tools in user interface design

**Step 1.** Specify a model of the user interface

**Step 2.** Engage with engineers and domain experts

**Step 3.** Iterate step 1 and 2 until agreement is reached

**Step 4.** Use verification tools to fix identified issues.

- ❖ User interface behaviour
- ❖ Visual aspect



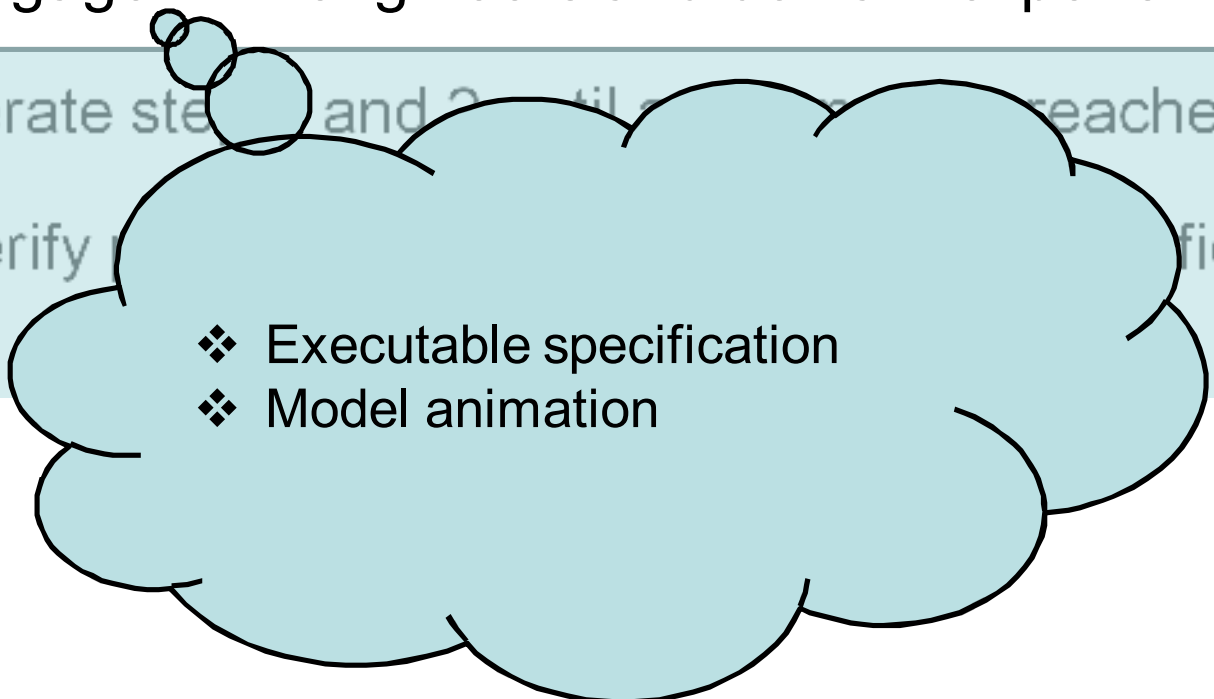
# Example workflow that integrates verification tools in user interface design

Step 1. Specify a model of the user interface

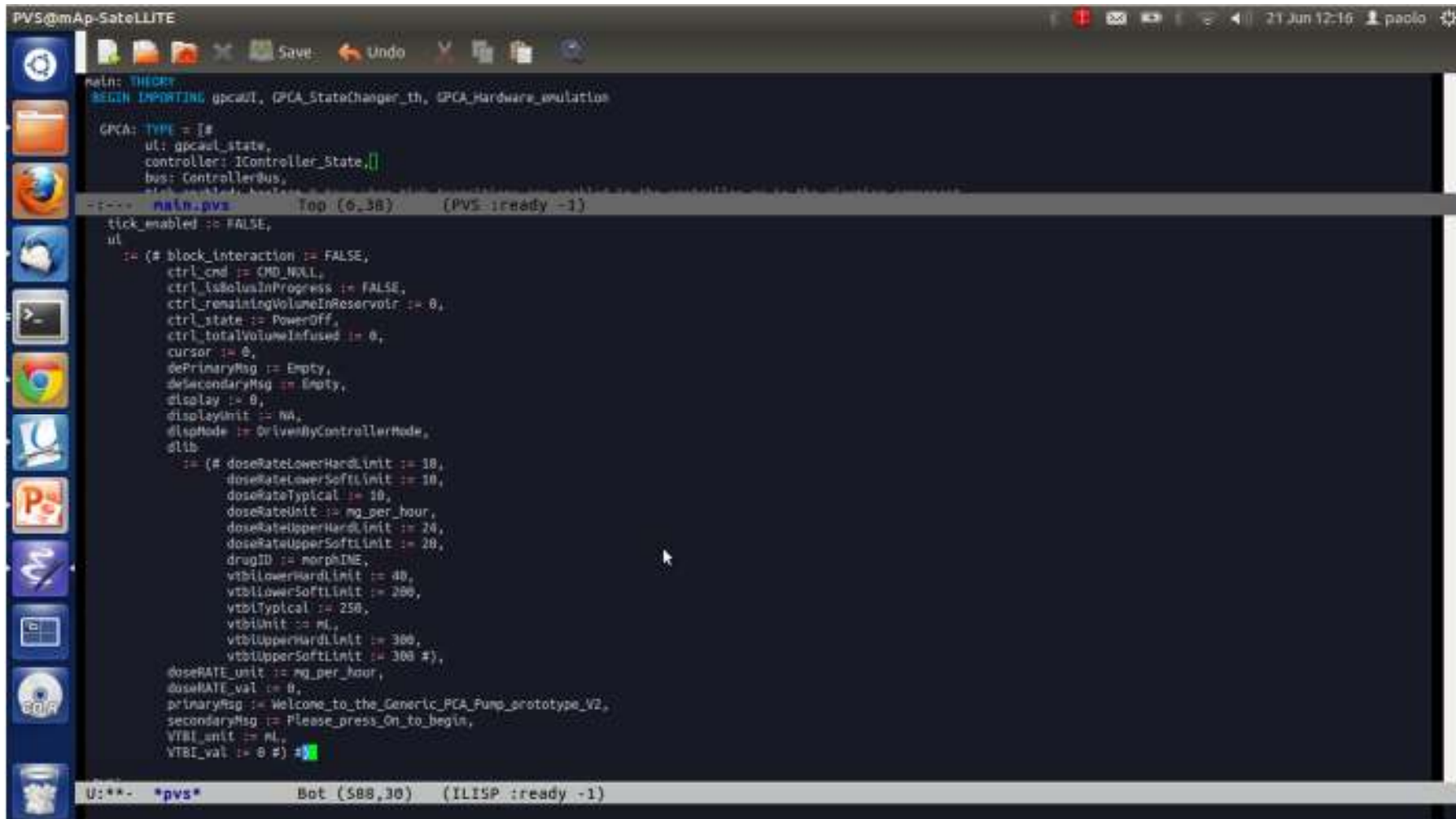
**Step 2. Engage with engineers and domain experts**

Step 3. Iterate step 1 and 2 until a specification is reached

Step 4. Verify the specification against identified issues.

- 
- ❖ Executable specification
  - ❖ Model animation

## Typical output of formal tools

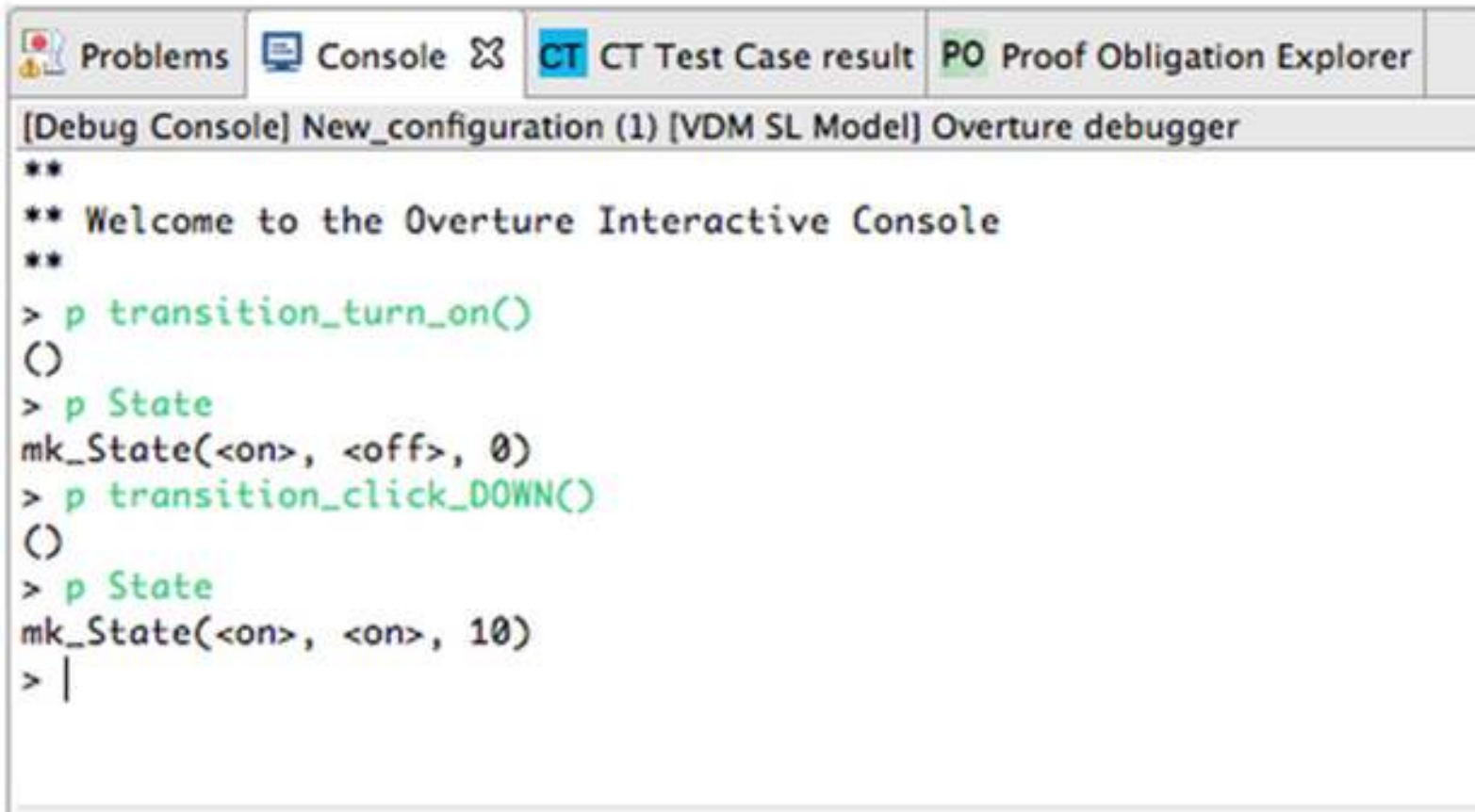


```
PVS@mAp-SateLLITE
main: THEORY
BEGIN IMPORTING gpcal, GPCA_StateChanger_th, GPCA_Hardware_emulation

GPCA: TYPE = [#
  ul: gpcal_state,
  controller: IController_State,[]
]

--*** main.pvs Top (6,38) (PVS :ready -1)
tick_enabled := FALSE,
ul
:= (# block_interaction := FALSE,
  ctrl_cmd := CMD_NULL,
  ctrl_isBolusInProgress := FALSE,
  ctrl_remainingVolumeInReservoir := 0,
  ctrl_state := PowerOff,
  ctrl_totalVolumeInfused := 0,
  cursor := 0,
  dePrimaryMsg := Empty,
  deSecondaryMsg := Empty,
  display := 0,
  displayUnit := NA,
  dispMode := DrivenByControllerMode,
  dlib
  := (# doseRateLowerHardLimit := 10,
    doseRateLowerSoftLimit := 10,
    doseRateTypical := 10,
    doseRateUnit := mg_per_hour,
    doseRateUpperHardLimit := 20,
    doseRateUpperSoftLimit := 20,
    drugID := morphine,
    vtblLowerHardLimit := 40,
    vtblLowerSoftLimit := 200,
    vtblTypical := 250,
    vtblUnit := mL,
    vtblUpperHardLimit := 300,
    vtblUpperSoftLimit := 300 #),
  doseRate_unit := mg_per_hour,
  doseRate_val := 0,
  primaryMsg := Welcome_to_the_Generic_PCA_Pump_prototype_v2,
  secondaryMsg := Please_press_On_to_begin,
  VTBI_unit := mL,
  VTBI_val := 0 #) #]
U:*** *pvs* Bot (588,30) (ILISP :ready -1)
```

## Typical output of formal tools



```
Problems Console CT Test Case result PO Proof Obligation Explorer
[Debug Console] New_configuration (1) [VDM SL Model] Overture debugger
**
** Welcome to the Overture Interactive Console
**
> p transition_turn_on()
()
> p State
mk_State(<on>, <off>, 0)
> p transition_click_DOWN()
()
> p State
mk_State(<on>, <on>, 10)
> |
```

# A more effective presentation of the model

The image displays a medical device interface with three main sections:

- Volume Display:** A large white box with a black border showing "17.04 mL".
- Control Panel:** A central area with a directional pad (a four-pointed star with a circle in the center) and six diamond-shaped buttons labeled "Cancel", "Bolus", "Stop", "Ok", "Edit", and "Start".
- Status/Action Panel:** A vertical panel on the right with a diamond-shaped button labeled "On" (top half) and "Off" (bottom half).

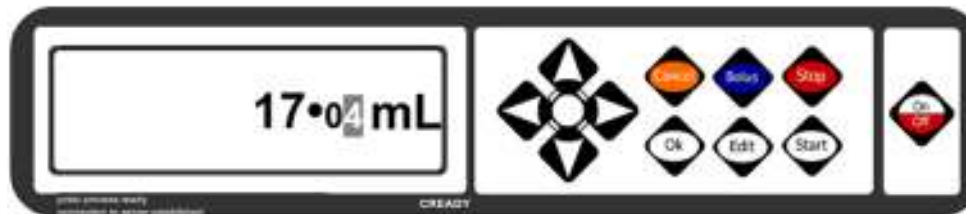
Below the main interface is a terminal window with the following text:

```
pvso process ready  
connection to server established  
CREADY
```

```
pvso (588,30) (PVSO ready -1)  
tick_enabled == FALSE,  
id  
- (* block_interaction == FALSE,  
  ctrl_cmd == CTRL_NULL,  
  ctrl_labelledInProgress == FALSE,  
  ctrl_remainingVolumeInReservoir == 0,  
  ctrl_state == Poweroff,  
  ctrl_totalVolumeInfused == 0,  
  cursor == 0,  
  defPrimaryMsg == Inits,  
  defSecondaryMsg == Error,  
  display == 0,  
  displayUnit == ML,  
  displayMode == DrivenByControllerFeds,  
  #118  
  - (* doseRateLowerHardLimit == 30,  
    doseRateLowerSoftLimit == 30,  
    doseRateTypical == 30,  
    doseRateUnit == mg_per_hour,  
    doseRateUpperHardLimit == 24,  
    doseRateUpperSoftLimit == 20,  
    drugID == morphine,  
    vtblowerHardLimit == 40,  
    vtblowerSoftLimit == 200,  
    vtblTypical == 250,  
    vtblUnit == ml,  
    vtblUpperHardLimit == 300,  
    vtblUpperSoftLimit == 300 *)  
  doseRateLimit == mg_per_hour,  
  doseRate_val == 0,  
  defPrimaryMsg == Welcome_to_the_General_PCA_flow_prototype_V2,  
  secondaryMsg == Please_press_OK_to_Continue,  
  VTR1_unit == ML,  
  VTR1_val == 0.0 *) #119
```

```
pvso (588,30) (TLLSP ready -1)
```

# PVSio-web: the concept

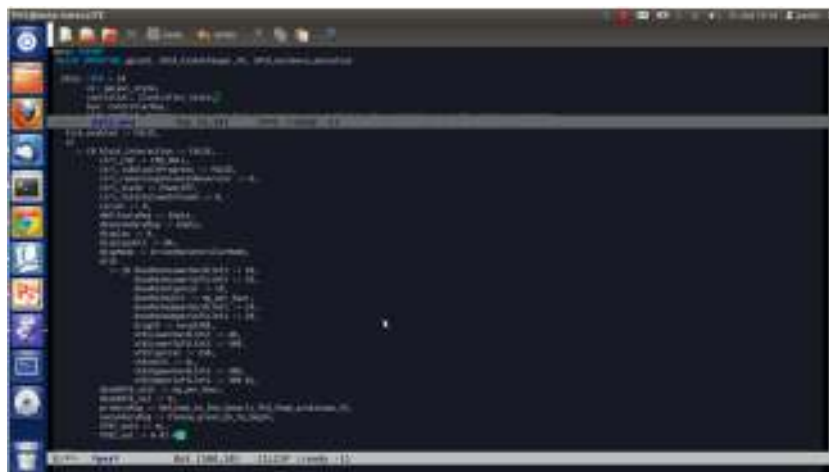


Specification of the graphical layout

Formal interpreter commands

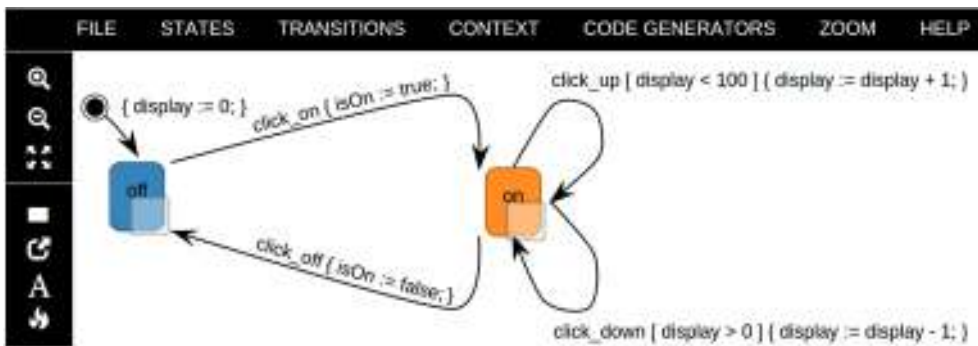


Formal interpreter response



Formal model

# Specialised user interfaces for different class of users



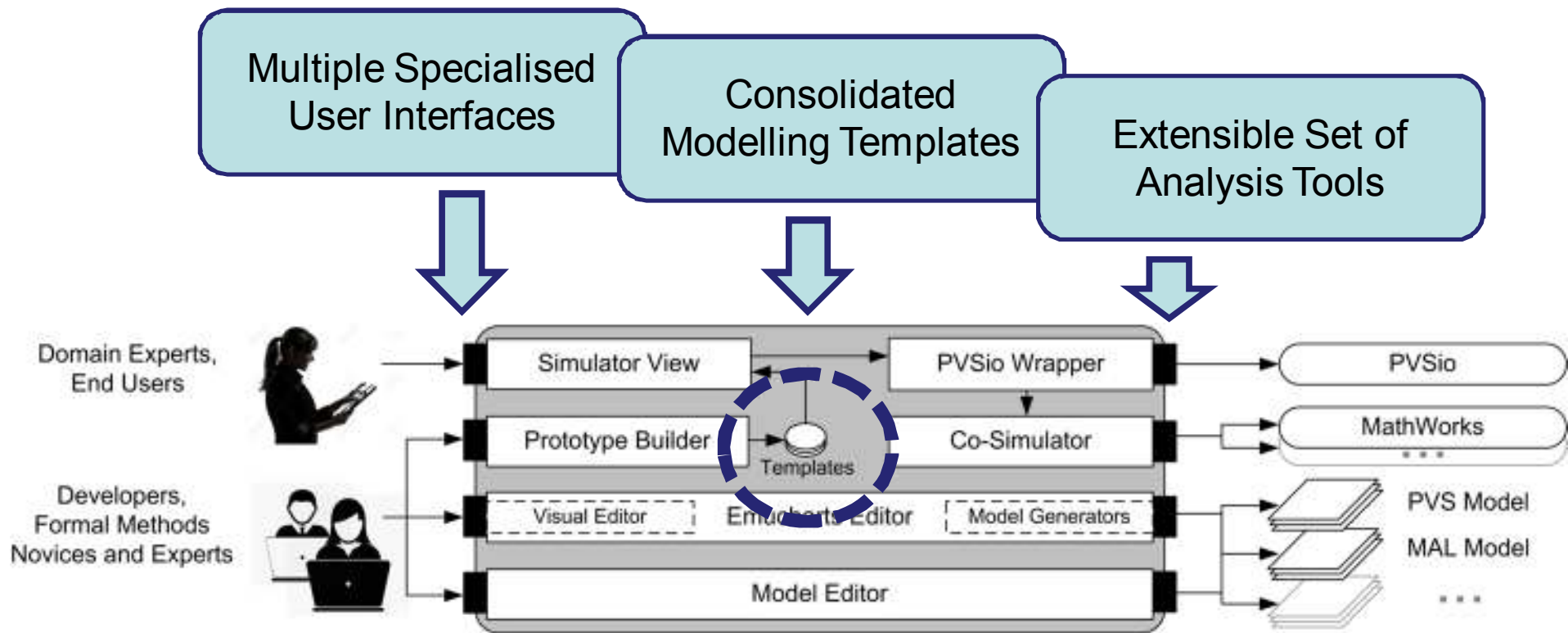
```
Compile Import Files Set As Main File Save Selected File
1+ emucharts th: THEORY BEGIN IMPORTING utils_th
2
3 MachineState: TYPE = { off, on }
4 State: TYPE = [# current_state : MachineState, previous_state: MachineState, displ
5
6 per click up(st: State): bool = ((current_state(st) = on) AND (display(st) < 100))
7+ click up(st: (per click up)): State =
8+   COND {current state(st) = on} AND {display(st) < 100}
9+   -> LET new st = leave state(on)(st),
10      new st = new st WITH [ display := display(st) + 1 ]
11      IN enter_into(on)(new_st) ENDCOND
```

- Real Device Users
- Device Experts / Regulators
- Human Factors Specialists

- Software Engineers
- Formal Methods Novices

- Formal Methods Experts

# The PVSio-web Architecture



Ref: **PVSio-web 2.0: Joining PVS to HCI**

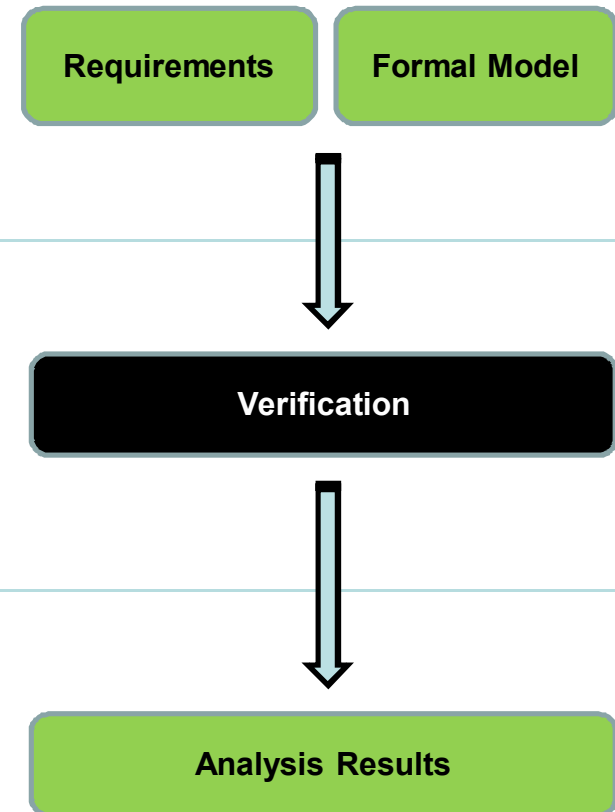
P. Masci, P. Oladimeji, Y. Zhang, P. Jones, P. Curzon, H. Thimbleby  
submitted to CAV2015, Jan, 2015

# Applications

- Validation of requirements
- Validation of formal models

- Lightweight formal analysis  
(e.g., expert walkthrough of prototypes)

- Demonstration of analysis results





# Key Achievements

## PVSio-web helped us

- to spot previously undetected software defects in commercial medical devices
- to develop training material for hospitals, to raise awareness about the identified design issues



[Design issues in medical user interfaces](#)  
(material available on YouTube)

# Impact stories

- **Healthcare**

- FDA is trialling our toolkit for assessing safety and usability aspects of medical devices
- Hospitals are using our results to improve procurement and to raise awareness about design issues in medical devices



- **Avionics / Aerospace**

- We are starting collaborations with NASA for the analysis of flight decks



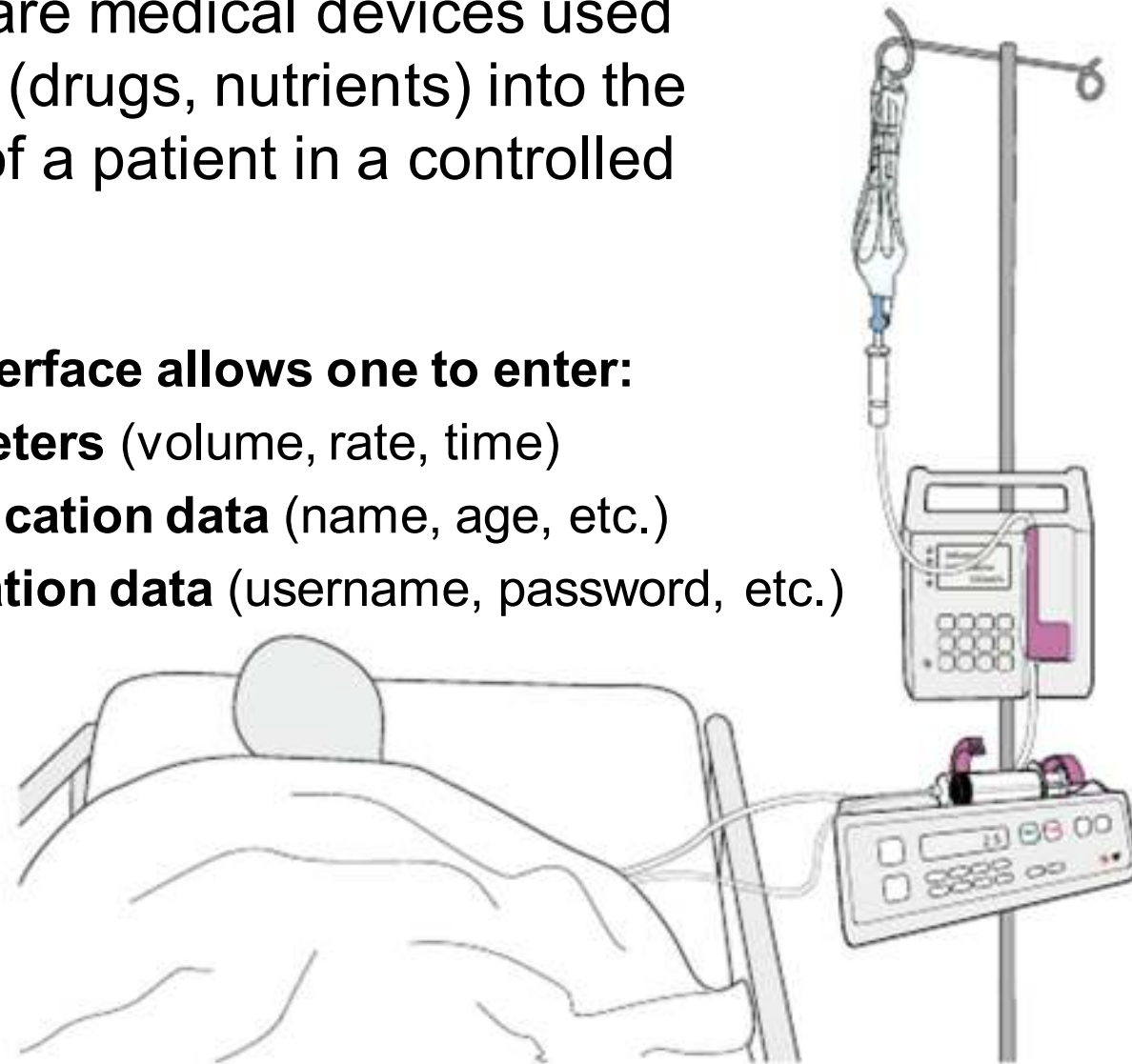
# Demonstrative example based on medical devices

# The example of infusion pumps

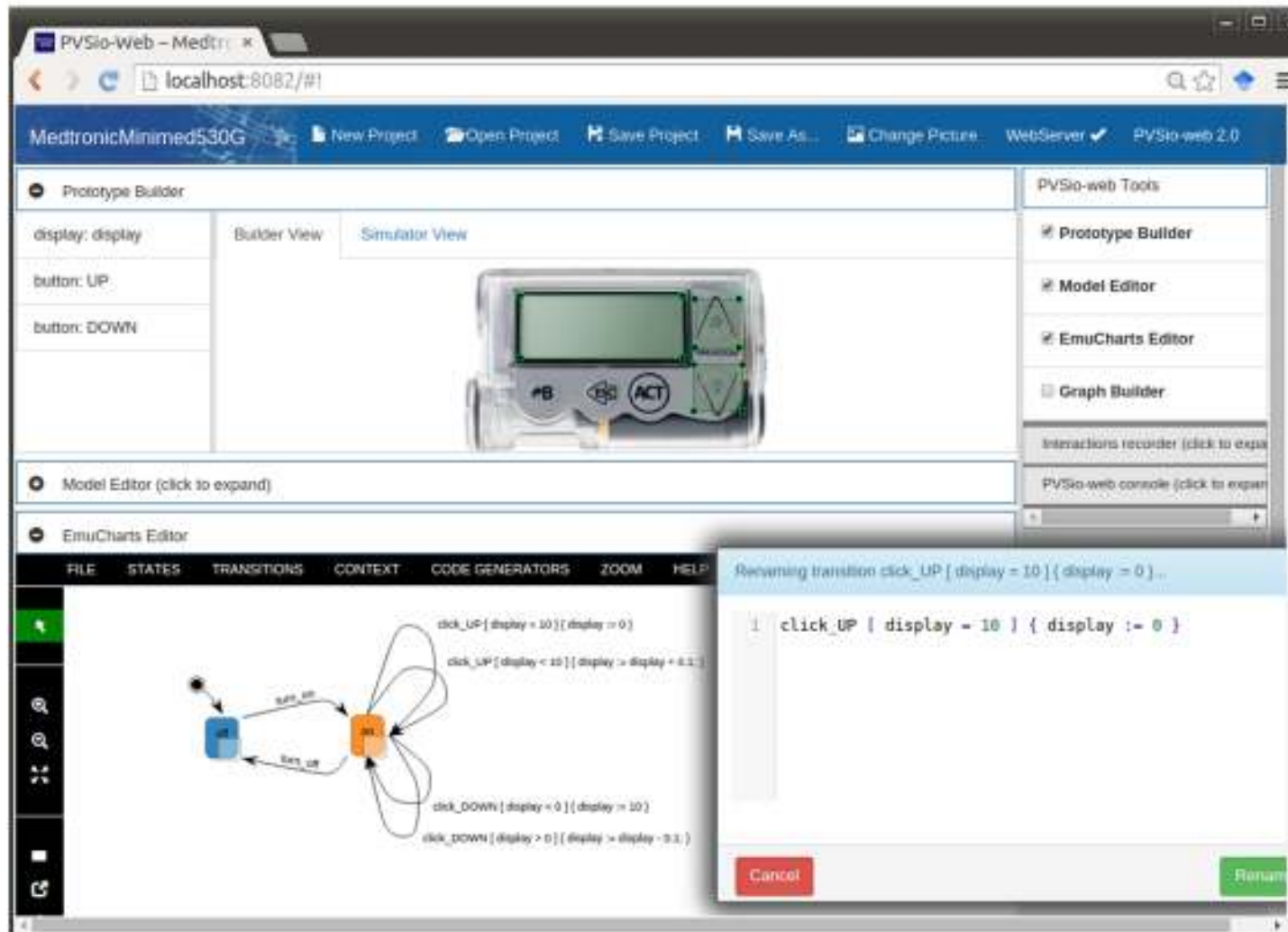
Infusion pumps are medical devices used to inject fluids (drugs, nutrients) into the bloodstream of a patient in a controlled manner.

The pump user interface allows one to enter:

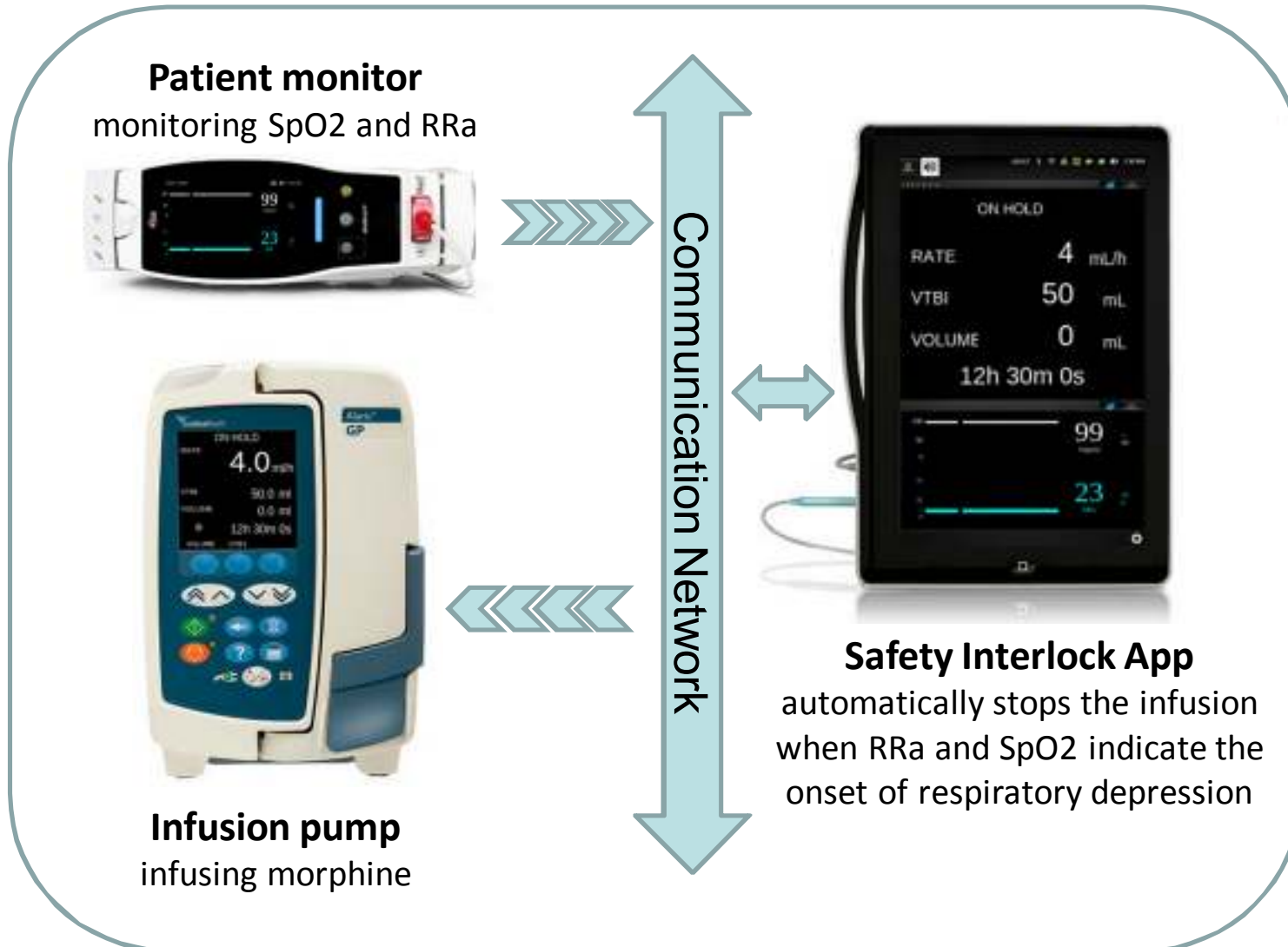
- **Infusion parameters** (volume, rate, time)
- **Patient authentication data** (name, age, etc.)
- **User authentication data** (username, password, etc.)



# Example 1: Prototyping the user interface software of an insulin pump



# Example 2: Interoperability Extension



# Ongoing work

- Templates for modelling and verification
- Template process wrappers for external tools
- Co-Simulator / Interoperability extension
- Code generators (C, Java, JavaScript, ...)
- Libraries of new user interface widget
- ...

# Overture evolution

- **1Y:** Web-based version of the toolset
  - Better integration with the PVSio-web toolkit
- **5Y:** Modelling templates
  - Possibly supported by verification strategies
- **10Y:** Trusted toolchain
  - From requirements to code