From Sensors to Visualization Dashboards: Need for Language Composition

Sébastien Mosser, Ivan Logre, Nicolas Ferry, Philippe Collet
This is not a solution paper
«Real-Life»

Paper
Disillusion (aka «real-life» hurts)

Involved Models

Conclusions
Disillusion

aka «real-life» hurts
Problem: Sensors $\rightarrow$ Visualization
Idea: **Components**
«Real-Life»
Real Problem: IoT $\rightarrow$ IoS
Systematic Modelling
Eleven kinds of models!
Coherency?

Reusability?
we Failed
Interesting

Use Case!
Involved Models
This model represents the following scenario:

To check a patient's glycaemia, a physical wants, thanks to the data display [0], to detect a risky situation [1] and then record a value [2].

The [2]nd task needs information from the [1]st one and has to be executed after it, which is the meaning of the [ ] >> operator.

To achieve the [1]st task, the user can detect the risky situation by [1.1] analyzing the variation rates, and, at any time, detect it thanks to a threshold [1.2] and then continue to look at the rate. The | > operator has this "Suspend-Resume" semantic.
We expect these models to be coupled by a small and well-defined interfaces.

«The component model used to reify the cloud application is loosely coupled with the task models defined for each user role.»
Case study: EBike, equipped with 11 sensors
thing EBike {

  provided port Sensors {

    receives getAirTemp, getCtrlTemp, getCurrent, getDist, getPower, getSpeed, getVoltage

    sends AirTempValue, CtrlTempValue, CurrentValue, DistValue, PowerValue, SpeedValue, VoltageValue

  }

}
Needs for Language Composition

Sensor Network Model

uses

Thing Model

receives ..., getDist, getSpeed, ...

ground_speed

altitude

Altimeter

GPS

rotation

Altitude

cellphone

aggregation

sensor

external system

Graph model (ad hoc)

Case study: EBike, equipped with 11 sensors
context Root inv: measurements->forAll(m |
  (m.name.oclIsUndefined() implies ! baseName.oclIsUndefined()) and
  (m.unit.oclIsUndefined() implies ! baseUnit.oclIsUndefined()) and
  (m.time.oclIsUndefined() implies ! baseTime.oclIsUndefined())
)
Uses

Strong coupling.

An evolution of the «used» model invasively impacts the «using» one.

«Modifying the data model impacts the component model that works on this data representation.»
User Interface model (ad'hoc, inspired by [Brel’13])

Implementation of the dashboard using SensorVisu

Task Model

constrains

Layout Model

[Task Model](front-camera (4))

[Layout Model](can contain [maps (8)], [altitude (5)], [heading (6)], [speed (7)])

[Brel’13]
Constrains

Choices in model $M$ restricts expressiveness in model $M'$.

In our case study we observe a bi-directional nature in each occurrence of this relationship.
Case study: Diabetes monitoring based on glycaemia sensor

Requirements Model

Task Model

implements

[UML Use Case]
Several alternative task models can be used to implement the same use case.

«A task diagram actually implements a given use case.»
Composition Graph

- co-exist with
- constrains
- uses
- implements

Things Platform Dashboards

- communication
- data
- components
- task
- resource
- variability
- layout
- requirements
Conclusions
Eleven kinds of models!
Composition Graph

---

coop-exist with
constrains

uses
implements

communication
data

components

task

resource

layout

requirements

state machine

graph

Dashboards

Things

Platform
From Sensors to Visualization Dashboards: Need for Language Composition

Sébastien Mosser (UNS)
Ivan Logre (UNS)
Nicolas Ferry (SINTEF IKT)
Philippe Collet (UNS)

mosser@i3s.unice.fr