Model Driven Engineering using COMPASS and Simulink

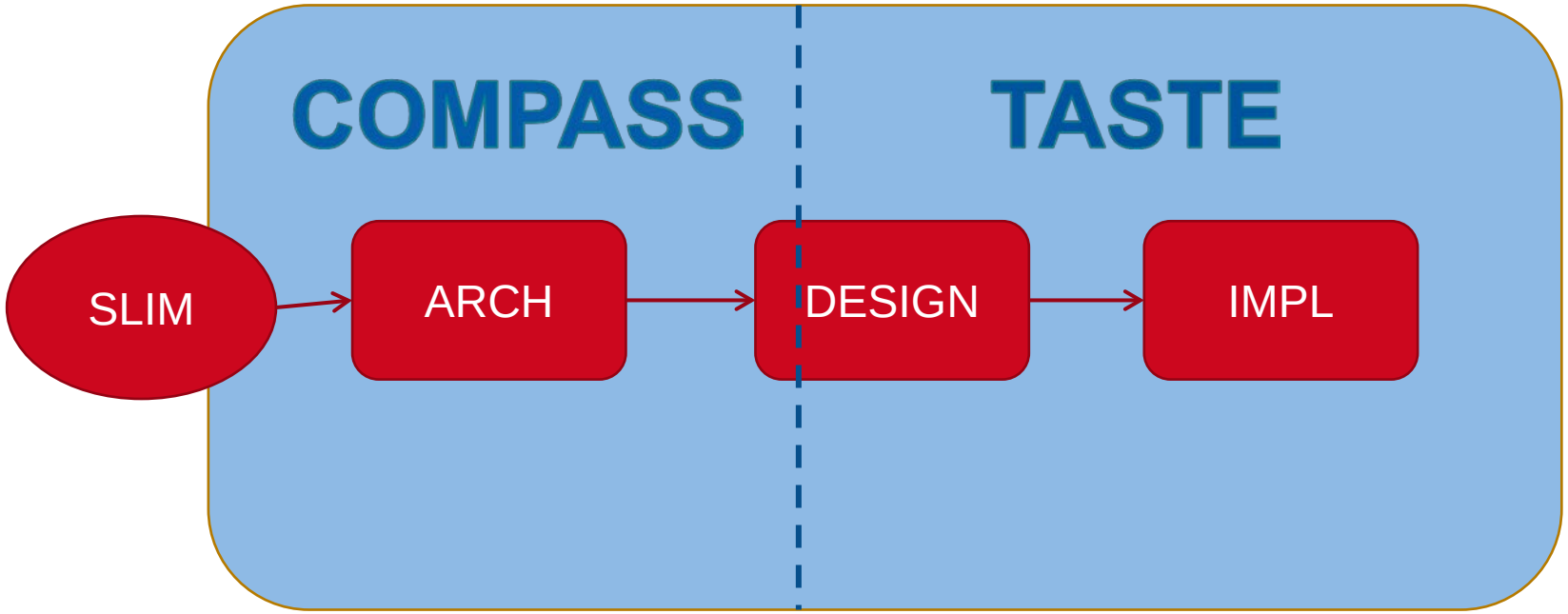
Lukas Armborst, **Harold Bruintjes**, Joost-Pieter Katoen

MBSSE ‘17
ESTEC, Noordwijk
09.12.2016

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Concept

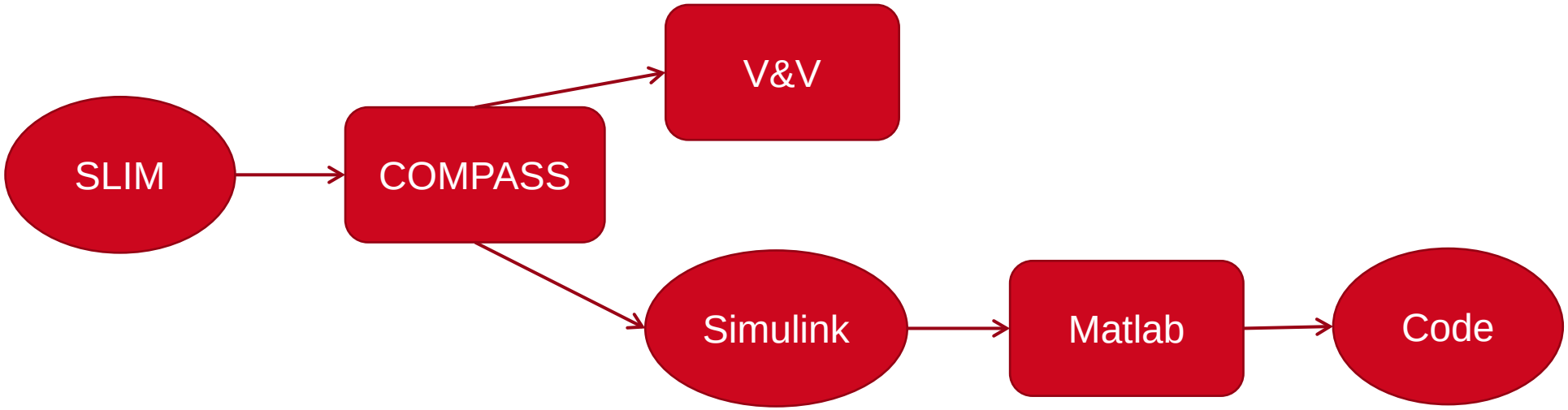
- Normally, COMPASS used for Architecural phase, and .e.g TASTE for Design/Implementation:



- Here: Implementation starting from the COMPASS model directly

Concept

- Translate SLIM to Simulink, and generate code from it



SLIM

- AADL based

- Components
- Ports
- Modes
- Data components

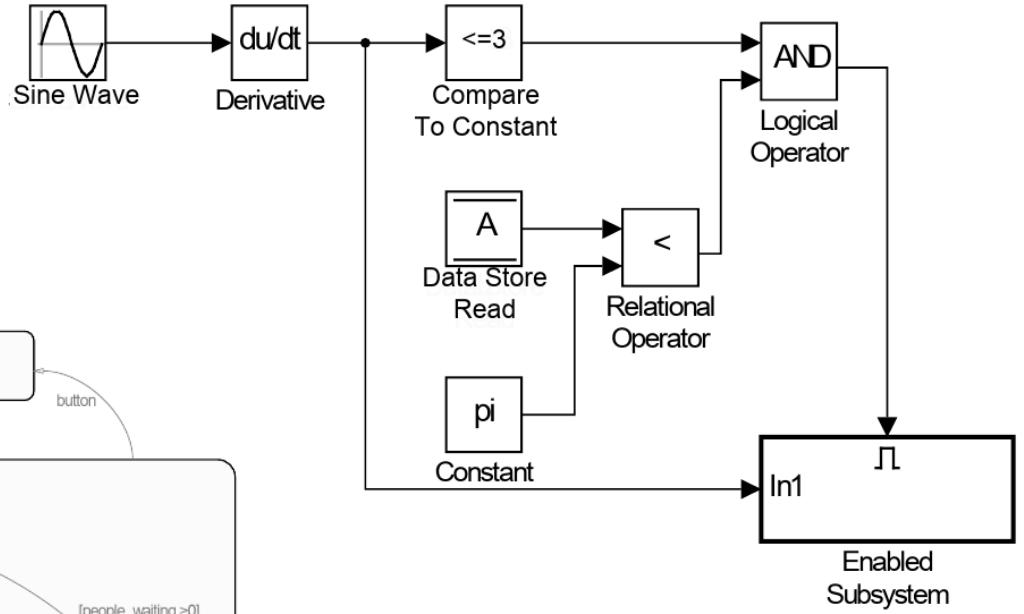
- Model extension (does not apply here)

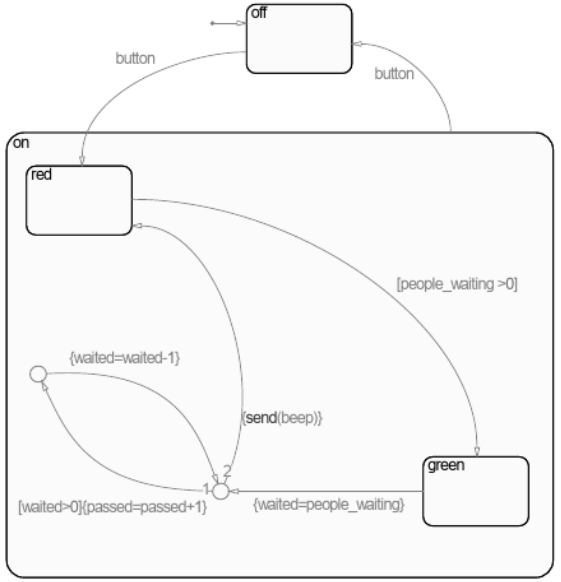
```
system Car
  features
    battery_status : in data port enum(OK, DEAD);
end Car;

system implementation Car.Impl
  subcomponents
    battery: device Battery.Impl;
  flows
    battery_status := case battery.output > 0 : OK
                          otherwise DEAD end;
end Car.Impl;
```

Simulink and Stateflow

- Graphical language
- Simulink:
  - Blocks
  - Datastores
  - Connections
  - Nested subsystems
- Stateflow: State based model with transitions





SLIM vs Simulink

Key differences

- Difference in handling non-determinism
  - SLIM permits all possibilities
  - Stateflow picks first enabled transition at earliest point in time
- Hybrid behaviour
  - SLIM: Linear equations (but exact)
  - Simulink: ODE's (numerical)
- Data types:
  - SLIM: Unbound integers
  - Simulink: fixed-width integers
- Simulink is not formally defined
  - Formal descriptions do exist, but are 3rd party

From SLIM to Simulink

Translation details

- Main structure fairly straightforward
  - Subcomponents are subsystems
  - SLIM modes and transitions make stateflow charts
  - Data subcomponents are data stores

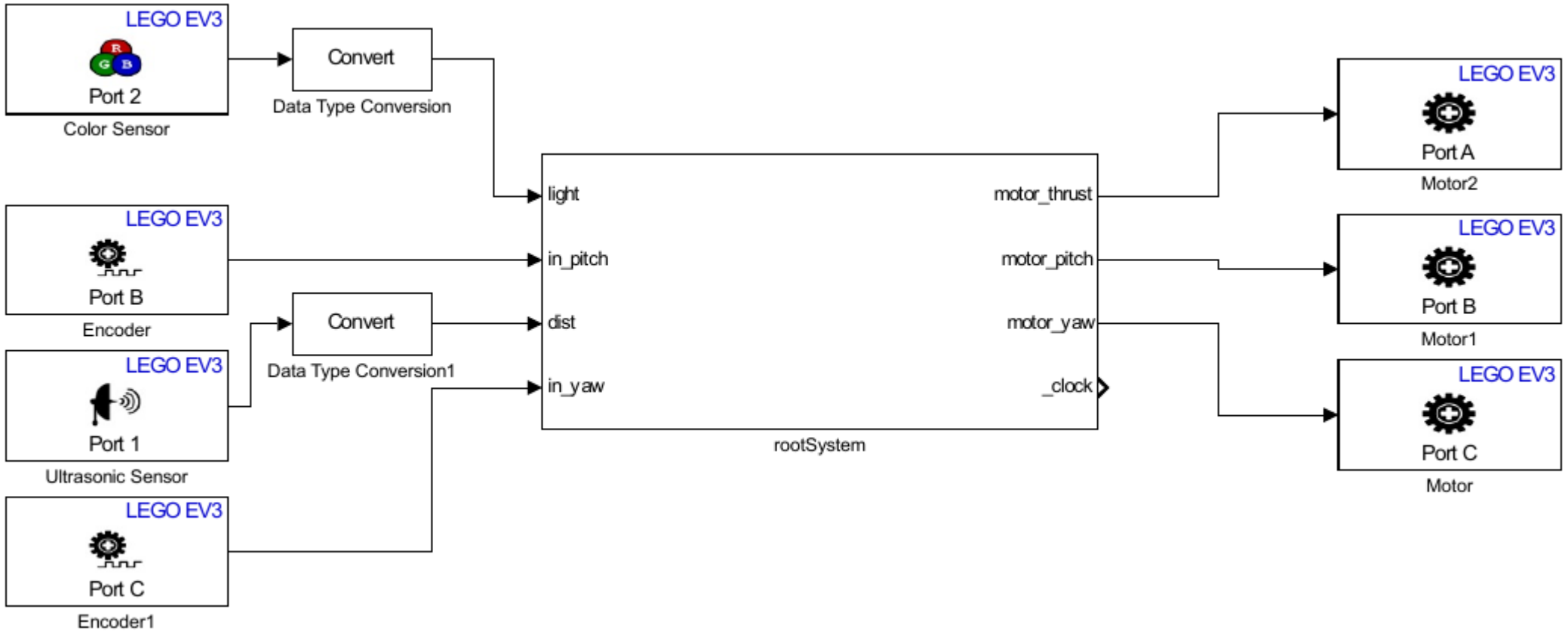
- Some key hurdles:
  - Global clock keeping track of time
    - local clock stores updated by a separate component
  - Simulink enforces execution order.
  - Cyclic data dependencies not allowed
  - Solved with execution cycles

Simulink to LEGO® Mindstorms®

- Mostly automated, Matlab can generate Mindstorms programs from Simulink models

- I/O has to be done by hand
  - The exporter wraps the entire model in a Subsystem to make this easier
  - Place blocks corresponding to Lego sensors/actuators, and set their port numbers
  - Add converters for data values

Simulink to LEGO® Mindstorms®



Simulink model with EV3 blocks

Case Study

Satellite model

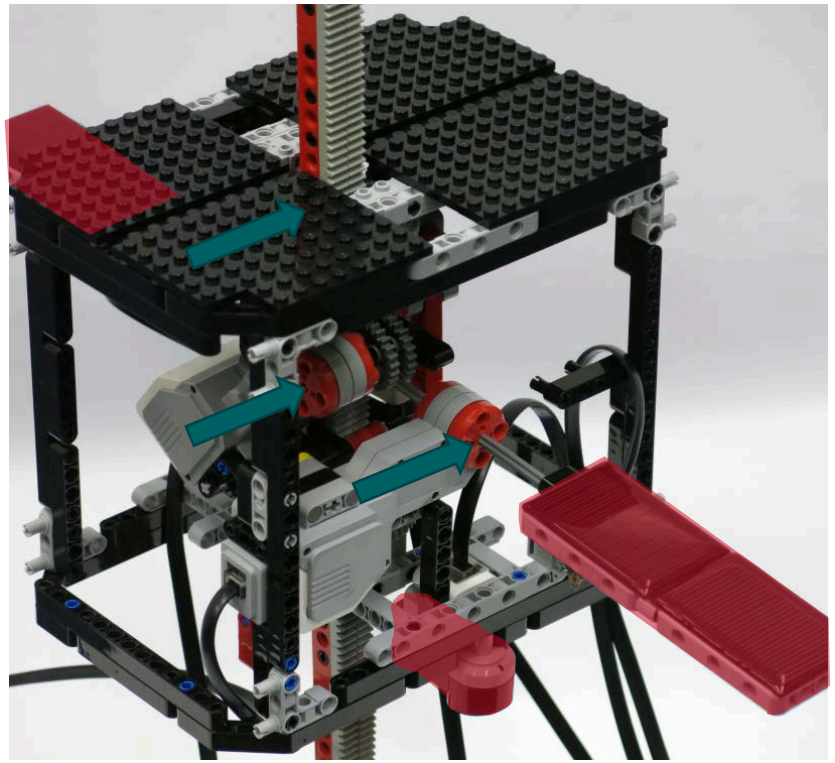
- Internally models battery status
- Continuous orbital decay
- Three modes:
  - Re-orbit: Consumes power
  - Attitude adjustment (yaw): Consumes power
  - Recharge: Reset attitude and adjust solar panels
- Failure when batteries are “empty”

- Discrete model (no continuous behavior)

Case Study

Lego Satellite

- 3 DOF
  - Orbital height
  - Yaw
  - Solar panel angle
- Brick controls the satellite
- Three motors
  - Altitude, Yaw, Solar panels
- Two sensors
  - Height sensor (ultrasonic)
  - Light sensor (solar panels)
- LED on EV3 brick indicates status (mode)



The End

http://www.compass-toolset.org

https://moves.rwth-aachen.de