Systerel

UNIVERSITY OF Southampton
School of Electronics
and Computer Science

Rodin User and Developer Workshop
University of Düsseldorf
20-22 September 2010

Decomposition Tool: Development and Usage

Renato Silva (University of Southampton)
Carine Pascal (Systerel)
T. S. Hoang (ETH Zurich)
Michael Butler (University of Southampton)

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

www.deploy-project.eu
www.event-b.org

deploy

Outline

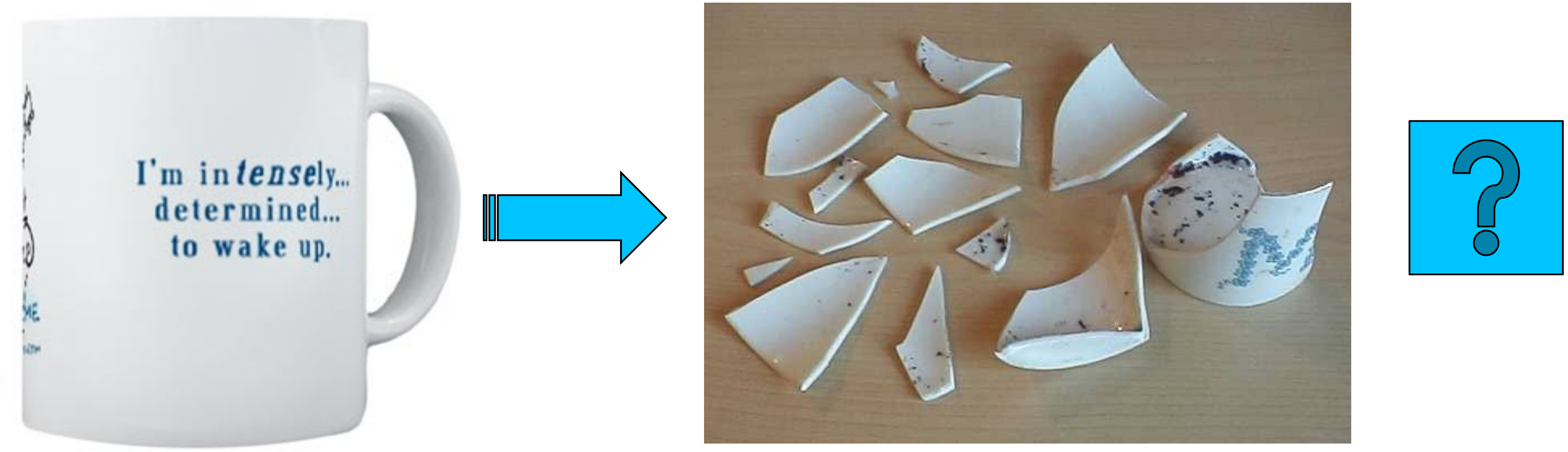
- Motivation
- Decomposition in Event-B: Shared Variable and Shared Event
- Tool
- Demo
- Conclusions/Future work

Motivation

- "Top-down" development style: new events and data-refinement of variables during refinements
- Problem: increasing complexity of the refinement process when having to deal with many events and many state variables (and likely many POs)
- Possible solution: Decomposition
  - Splitting a large model into smaller sub-components
  - Design/architectural decision
  - Team development
  - Alleviate the complexity of discharging POs

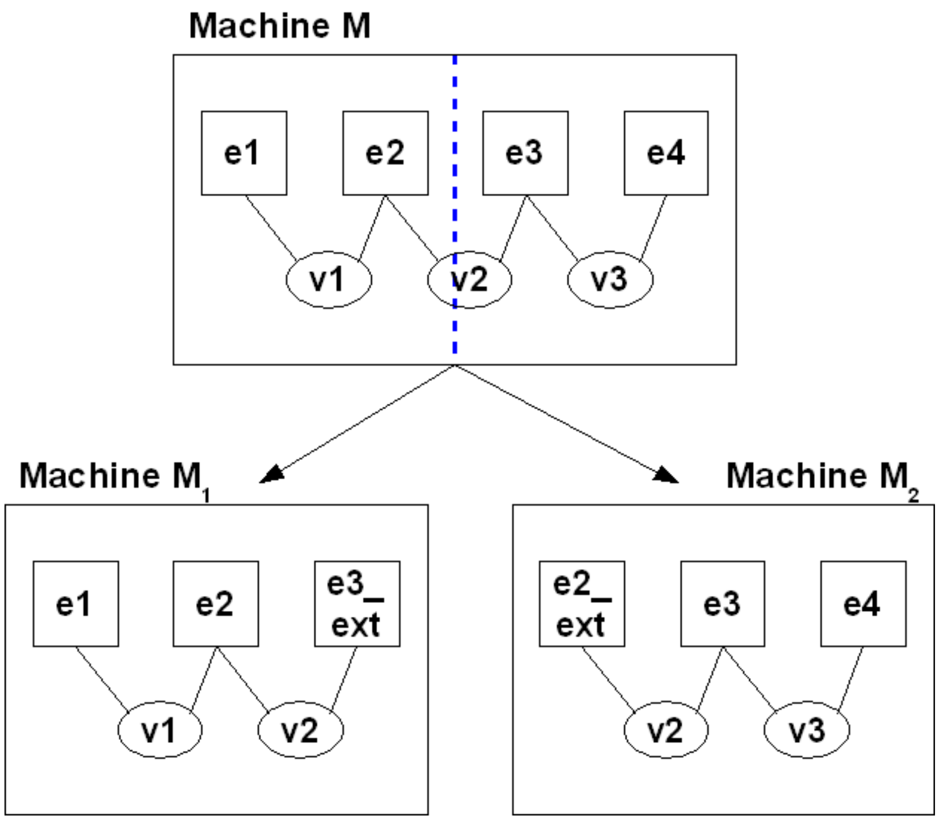
Motivation (cont.)

- Decomposition = “break up” something that is composed.



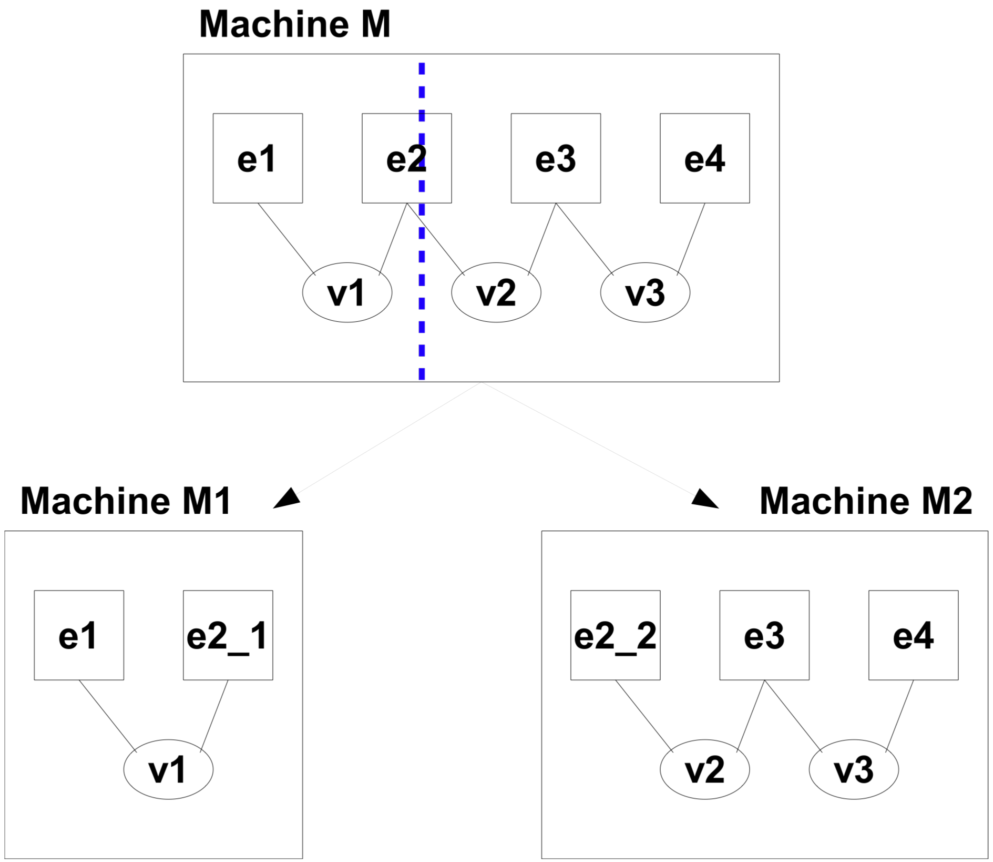
- How to do it?

Shared Variable decomposition



- Events of M are first partitioned into M1, ..., Mn.
- Variable partition is a consequence of the event splitting.
- Shared variable: variable accessed by events of distinct sub-machines (in opposition to private variable).
- External event: event of a sub-machine which is built from an event of the non-decomposed machine and simulates the way the shared variables are handled in the non-decomposed machine (in opposition to internal event).

Shared Event decomposition



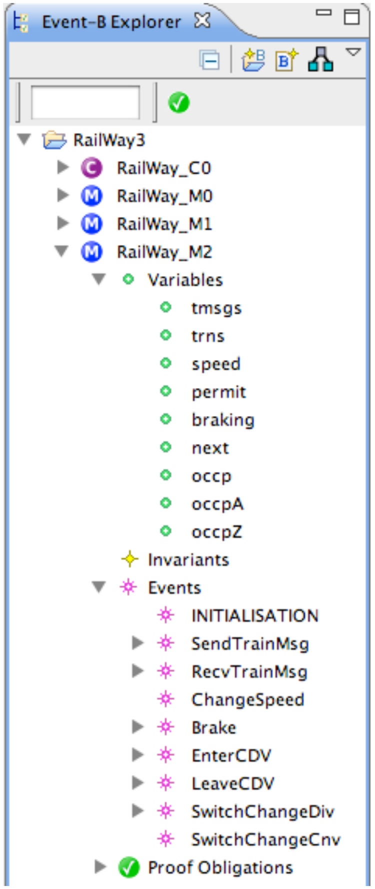
- Variables of M are first partitioned into M1, ..., Mn. Variables are not shared.
- Event partition is a consequence of the variable splitting.
- No notion of external events.
- If an event shares variables belonging to different sub-models, that event (parameters, guards, actions) is split over the sub-models (*validation is required*).
- Interaction between sub-components: synchronized events communicate via shared parameters.

Which decomposition is better?

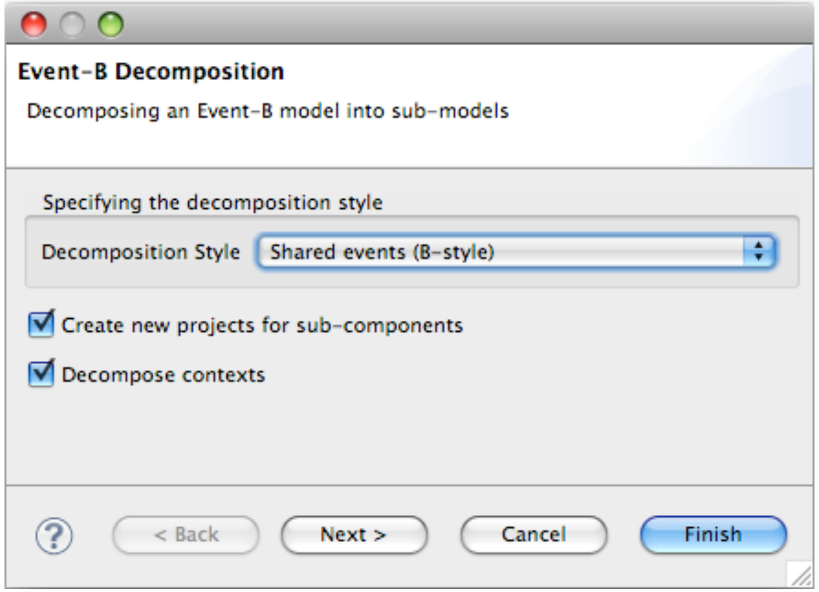
- It depends on :
  - the system being developed
  - user habits
- Shared Variable:
  - Variables being shared suggest use in parallel applications
- Shared Event:
  - Suggests the use in message passing distributed systems or middleware systems.

Tool

- A single plug-in for both styles in the Rodin platform.
- Usage of extension mechanisms: wizard, menu, editor, static checker.
- Decomposition wizard:



```
▼ RailWay3
  ▶ RailWay_C0
  ▶ RailWay_M0
  ▶ RailWay_M1
  ▼ RailWay_M2
    ▼ Variables
        tmsgs
        trns
        speed
        permit
        braking
        next
        occp
        occpA
        occpZ
      Invariants
    ▼ Events
        INITIALISATION
      ▶ SendTrainMsg
      ▶ RecvTrainMsg
        ChangeSpeed
      ▶ Brake
      ▶ EnterCDV
      ▶ LeaveCDV
      ▶ SwitchChangeDiv
        SwitchChangeCnv
    ▶ Proof Obligations
```



Event-B Decomposition

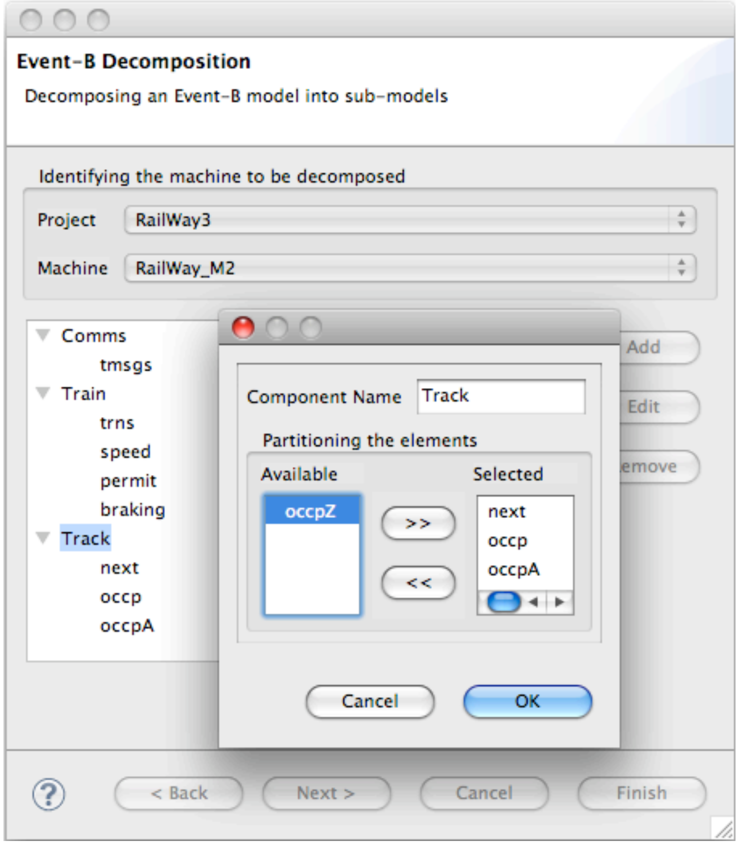
Decomposing an Event-B model into sub-models

Specifying the decomposition style

Decomposition Style: Shared events (B-style)

☑ Create new projects for sub-components

☑ Decompose contexts



Event-B Decomposition

Decomposing an Event-B model into sub-models

Identifying the machine to be decomposed

Project: RailWay3

Machine: RailWay_M2

```
▼ Comms
    tmsgs
▼ Train
    trns
    speed
    permit
    braking
▼ Track
    next
    occp
    occpA
```

Component Name: Track

Partitioning the elements

Available	Selected
occpZ	next occp occpA

Demo

- Select machine to be decomposed
- Select decomposition style
- Select where to store the sub-components. Also option to to decompose contexts
- Define sub-components and respective elements
- Save decomposition file
- Run decomposition

Conclusion

- Decomposition: used to decrease the complexity and increase the modularity of large systems
- Main benefits:
  - Further refinement of independent sub-models in parallel (monotonicity).
  - Allow team development for each sub-model (attractive option for the industry)
  - Distribution of POs

Conclusion (cont.)

- Decomposition in Event-B:
  - Shared Variable approach: seems more suitable for modelling parallel systems*
  - Shared Event approach: seems more suitable for modelling messaging-passing distributed systems**
- Decomposition tool is available since the release 1.2 of the Rodin platform.
  - Already used in several case studies with positive feedback ☺

* Hoang, T., Abrial, J.R.: Event-B Decomposition for Parallel Programs. Abstract State Machines, Alloy, B and Z (2010) 319–333

** Butler, M.: An Approach to the Design of Distributed Systems with B AMN. In: Proc. 10th Int. Conf. of Z Users: The Z Formal Specification Notation (ZUM), LNCS 1212. (1997) 221–241

Future work

- Visual perspective of decomposition
    - It seems easier to partition a system by visualising how to allocate the elements in the sub-components
    - Option: using Graphical Modelling Framework (GMF)
- Application of more complex case studies and analyse the results (Ongoing work)
- Integration of the decomposition plug-in with other plug-ins (compatibility problems): Records, Modularisation plug-in, etc

Questions?

Thank you