Fault tolerance view in Event-B development

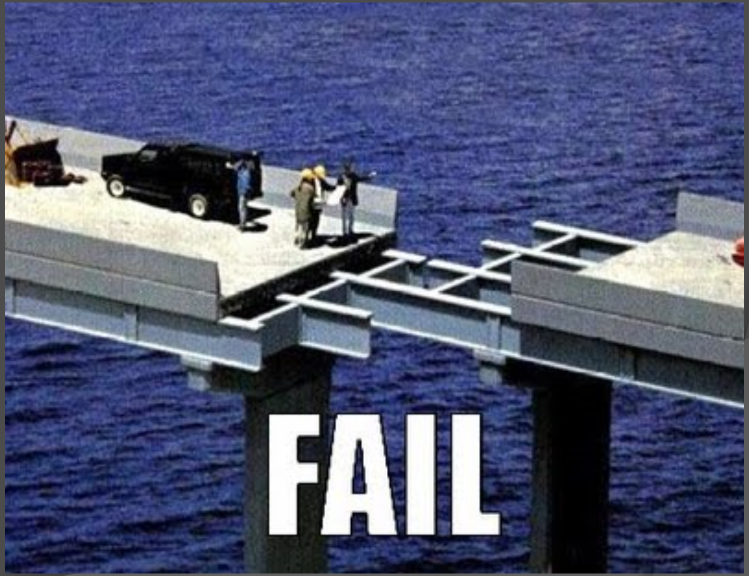
(Mode/FT Views plugin)

Ilya Lopatkin, Alexei Iliasov, Alexander Romanovsky
Newcastle University

Dusseldorf 22/09/2010

Motivations

- Amount of FT-related requirements to critical systems
- Early modelling of FT

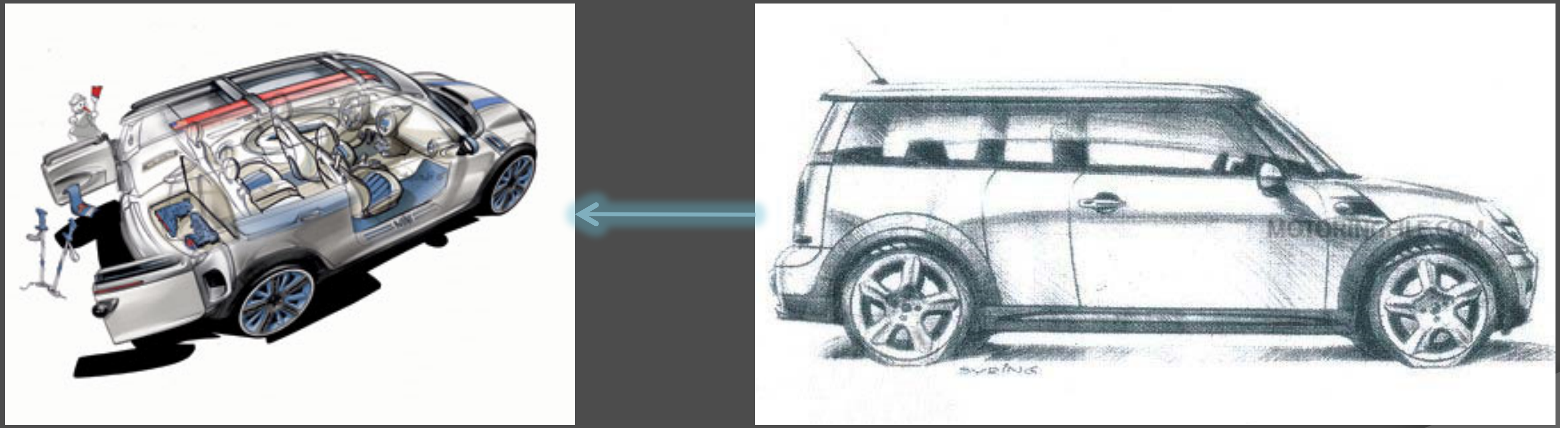


Motivations

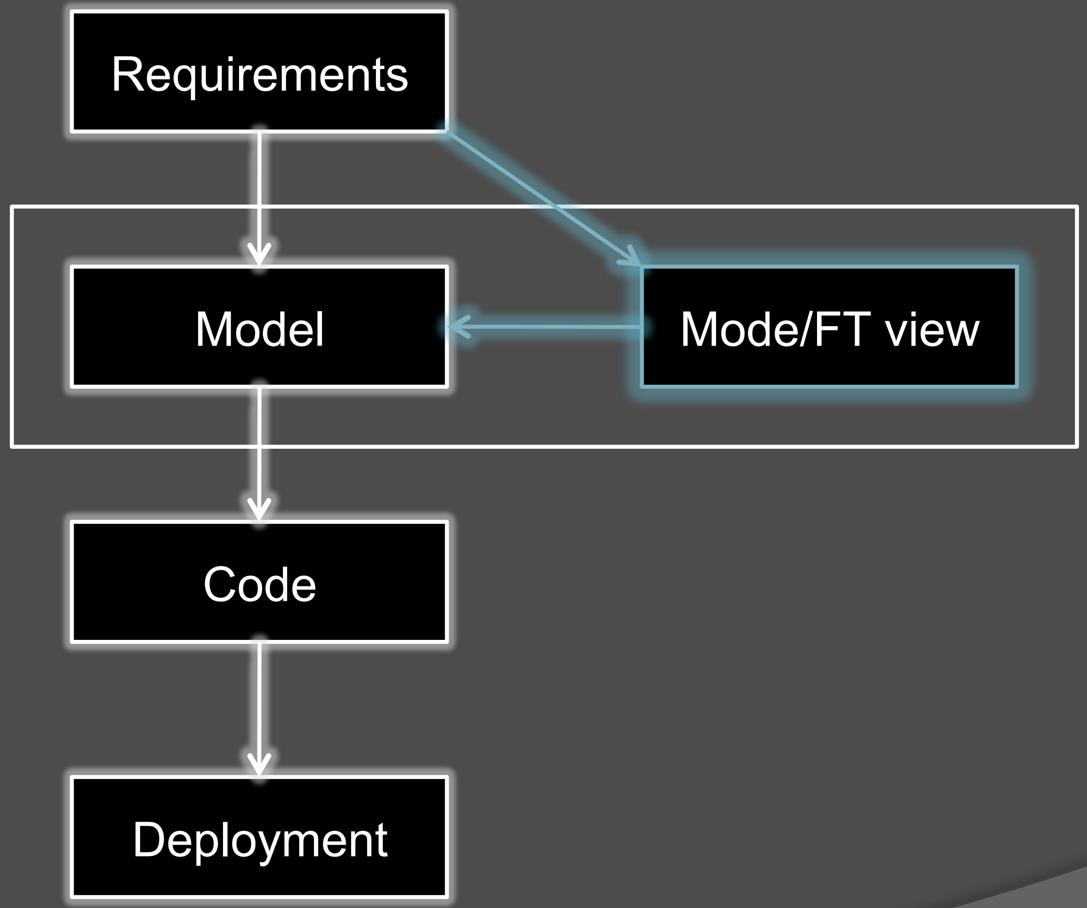
- Why model?
  - There are requirements
    - Define context, what can go wrong
  - Trace
  - Certify
- Recurring artefacts
- Separation of concerns
- Explicitness

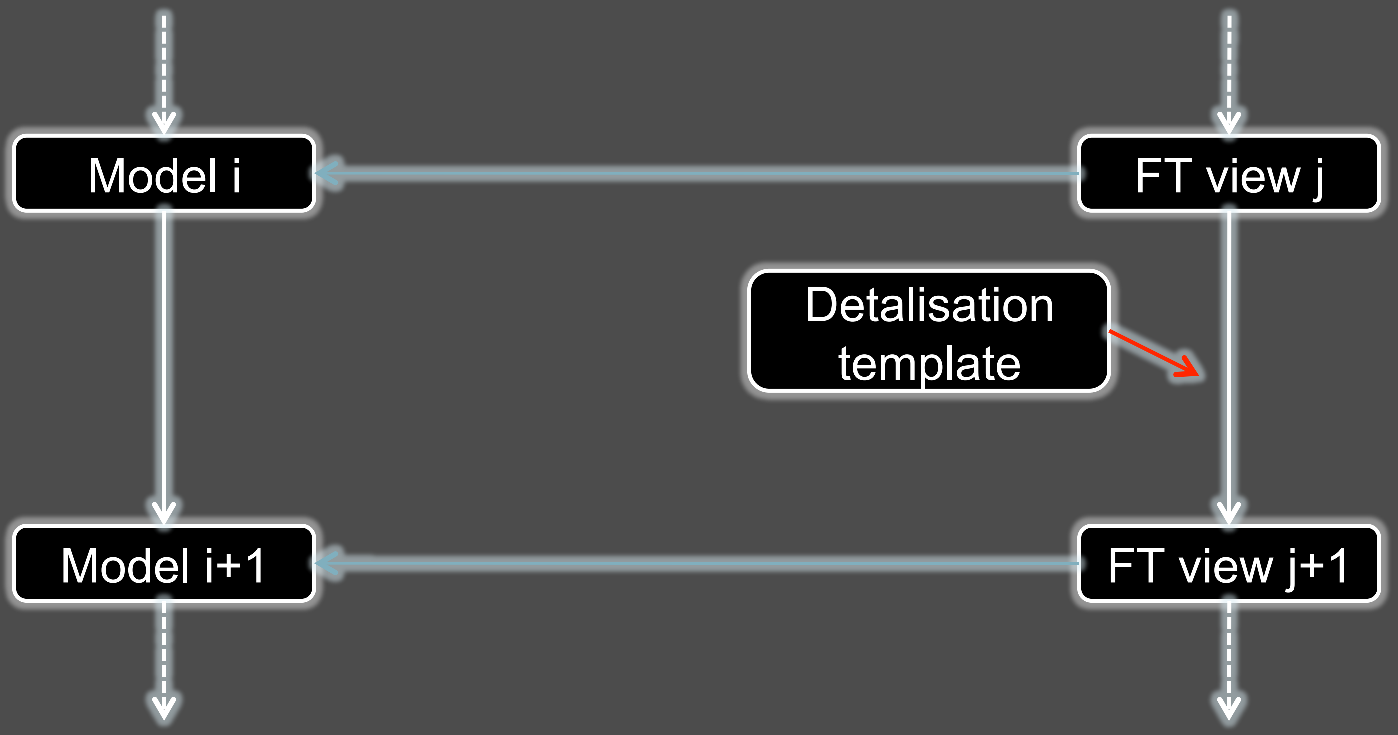
View

- Build complex document by linking simpler ones
- ANSI/IEEE Std 1471 :: ISO/IEC 42010



Where our view stands




Model i
FT view j
Detalisation template
Model i+1
FT view j+1

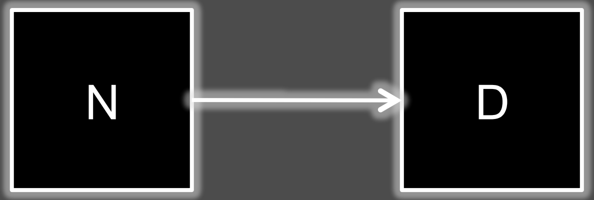
Where idea comes from

- Deploy documents
- Fault tolerance modelling
- Modal views by Ncl and Brazil
  - Modes ←→ Event-B

Abstract classes of FT systems

- Normal
  - All errors are recoverable

- Normal + Degraded
  - There are errors that cannot be masked

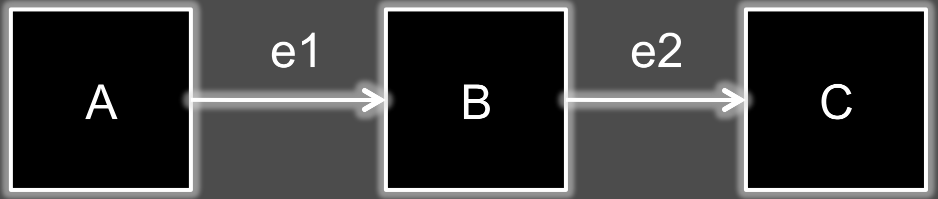


Modes

- Operation mode: the expected system functionality under distinguished working conditions of the system
- Mode transition: the possible changes in the working conditions of a system
- A modal system is a set of modes related by mode transitions

Mode/FT view concepts

- Modes
- Transitions



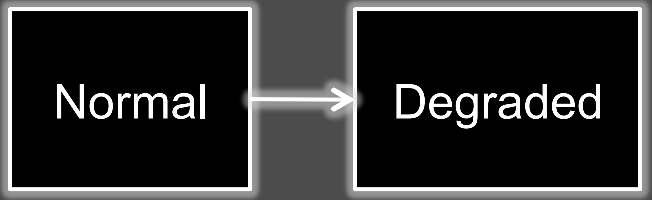
A

e1

B

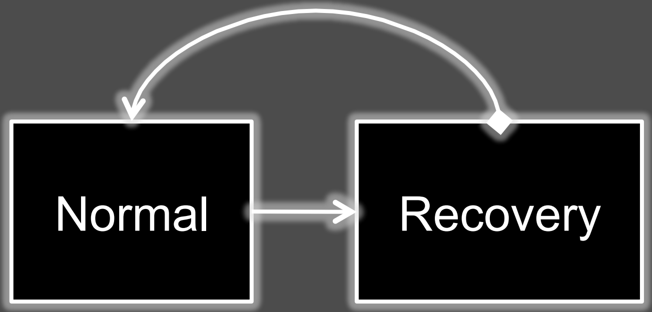
e2

C



Normal

Degraded

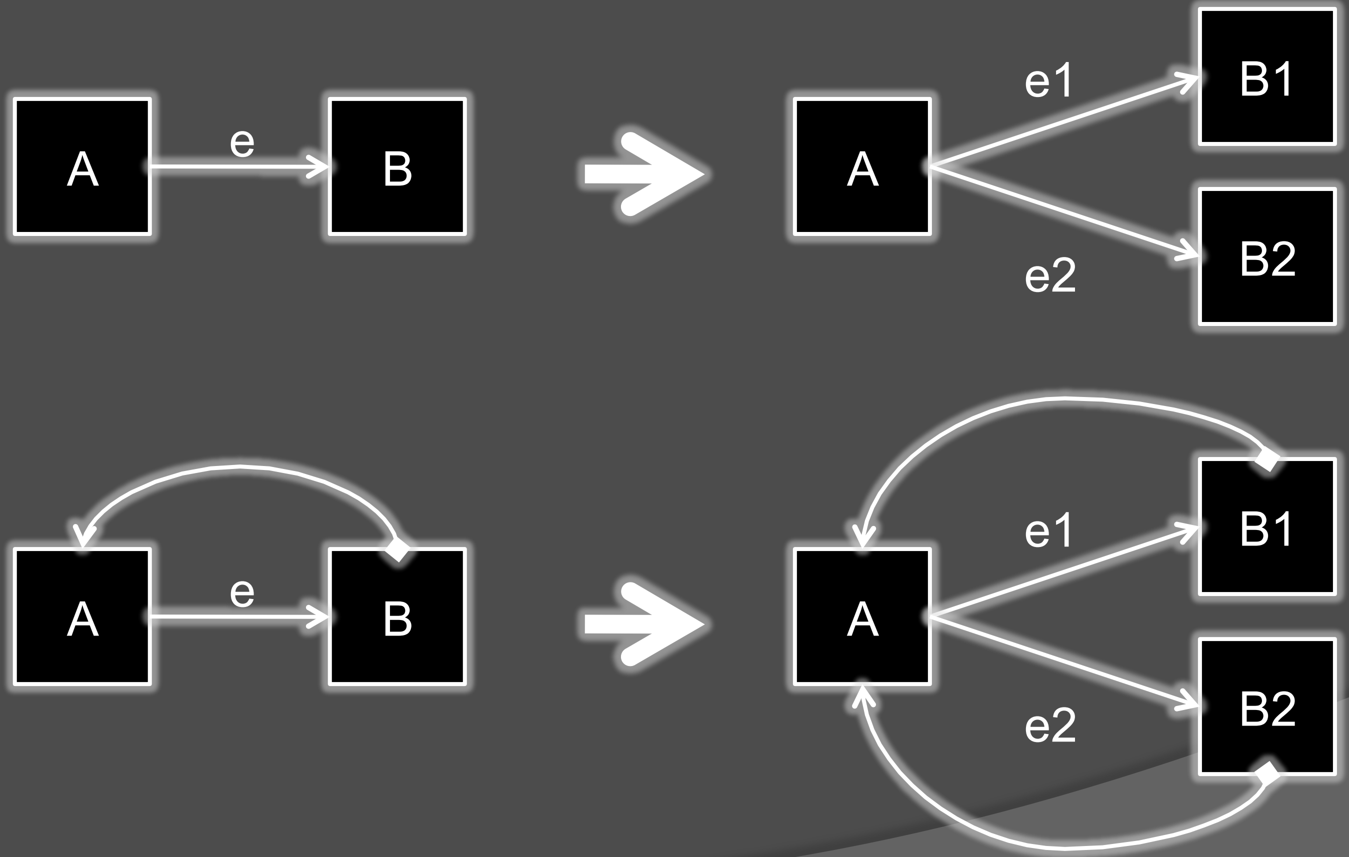


Normal

Recovery

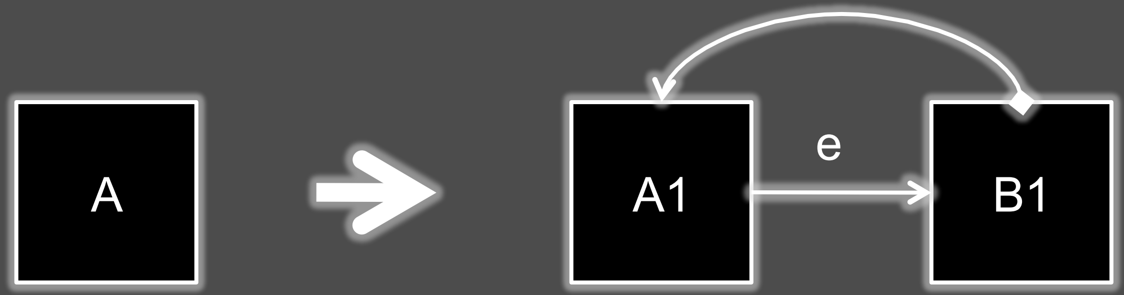
Detalisation templates

- Template 1: Detalisation of an error



Detalisation templates

- Template 2: New error



Detalisation

- Our “refinement”
- Proper projection
  - Modes into modes
  - Transitions into external and internal transitions

Mode/FT view formalisation

- Modes provide different functionalities under differing operating conditions
- Each mode is characterized by A/G
- $A(v)$ – assumption
- $G(v, v')$ – guarantee
- v – model variables

Mode/FT view formalisation

- Assumptions exhaust the invariant

$$I(v) \Rightarrow A_1 \vee A_2 \vee \cdots \vee A_n$$

- There exists a transition within mode

$$\exists v, v' \cdot I(v) \wedge A(v) \Rightarrow G(v, v')$$

- Modes do not overlap

$$I(v) \Leftarrow A_1(v) \oplus \cdots \oplus A_n(v)$$

Mode/FT view formalisation

- Detalisation conditions

$$A(v)/G(v,v') \sqsubseteq A'(u)/G'(u,u')$$
$$\text{iff } \begin{cases} J(v,u) \wedge A(v) \Rightarrow A'(u) \\ J(v,u) \wedge G'(u,u') \Rightarrow G(v,v') \end{cases}$$

$$A(v)/G(v,v') \sqsubseteq \begin{matrix} A_1(u)/G_1(u,u') \\ A_2(u)/G_2(u,u') \end{matrix} ,$$
$$\text{iff } \begin{cases} J(v,u) \wedge A(v) \Rightarrow A_1(u) \vee A_2(u) \\ J(v,u) \wedge G_1(u,u') \vee G_2(u,u') \Rightarrow G(v,v') \end{cases}$$

Mode/FT view formalisation

$$\begin{array}{l} A_1/G_1 \mapsto E_1 \\ A_2/G_2 \mapsto E_2 \\ \quad \ldots \\ A_n/G_n \mapsto E_n \end{array}$$

- Relate modes to events
- Events must satisfy the modes guarantee

$$I(v) \wedge A(v) \wedge H(v) \wedge R(v, v') \Rightarrow G(v, v')$$

- Partitioning of events into modes must agree with guards

$$\begin{array}{l} H(v) \Rightarrow A_1(v) \vee \cdots \vee A_k(v) \\ A_{k+1}(v) \vee \cdots \vee A_n(v) \Rightarrow \neg H(v) \end{array}$$

Ongoing, future, possible work

- Tool for Mode/FT views
- Link with requirements platform
- Patterns/templates on the model level

Thank you