Model-based Verification of Safety Contracts

Elena Gómez-Martínez†, **Ricardo J. Rodríguez**‡, Leire Etxeberria Elorza*, Miren Illarramendi Rezabal*, Clara Benac Earle†

{egomez,cbenac}@babel.ls.fi.upm.es, rj.rodriguez@unileon.es,

{letxeberria,millarramendi}@mondragon.edu

†Technical University of Madrid
Madrid, Spain

‡RIASC, University of León
León, Spain

*Mondragon Unibertsitatea
Arrasate-Mondragón, Spain

September 1, 2014

1st International Workshop on Safety & Formal Methods
Grenoble (France)

Agenda

Agenda

Introduction (I): Motivation

Safety assessment

- Needed by some systems (e.g. critical systems)
  - Industrial equipment, road vehicles, avionics. . .
  - Requirements specified by industrial standards (IEC-61508, ISO-26262, DO-178C)
- Later verification induces budget overruns
  - Example: Half of the overall costs in avionics software domain

RIASC

Introduction (II): Motivation

Safety assessment needs to be incorporated early into software design process

Introduction (II): Motivation

Safety assessment needs to be incorporated early into software design process

Contract-based design

- Popular approach for the design of complex systems
- Safety properties are difficult to guarantee $\rightarrow$ use of contracts

Introduction (II): Motivation

Safety assessment needs to be incorporated early into software design process

Contract-based design

- Popular approach for the design of complex systems
- Safety properties are difficult to guarantee → use of contracts

Contracts

- Commonly used to specify relationships between system components
- Pre- and post-conditions of a system component
- Refinement idea: safety contract
  - Assumptions; Guarantees
  - Aim: to assure a certain level of confidence of a component

Introduction (III)

UML

- Well-known modelling language in the industry
- Vehicle to integrate safety requirements into software lifecycle

RIASC

Introduction (III)

UML

- Well-known modelling language in the industry
- Vehicle to integrate safety requirements into software lifecycle
- Two (current) approaches:
  - Object Constraint Language
  - Specific UML profiles

Introduction (III)

UML

- Well-known modelling language in the industry
- Vehicle to integrate safety requirements into software lifecycle
- Two (current) approaches:
  - Object Constraint Language
  - Specific UML profiles

Merging two domains...

- UML: Standard engineering practice
  - UML SM and UML SD: Dynamic part of the system
  - UML Composite diagram: Static one → enriched with safety contracts
  - UML profile (MARTE): Performance system information
  - Representation of safety contracts as OCL constraints

Introduction (III)

UML

- Well-known modelling language in the industry
- Vehicle to integrate safety requirements into software lifecycle
- Two (current) approaches:
  - Object Constraint Language
  - Specific UML profiles

Merging two domains...

- UML: Standard engineering practice
  - UML SM and UML SD: Dynamic part of the system
  - UML Composite diagram: Static one → enriched with safety contracts
  - UML profile (MARTE): Performance system information
  - Representation of safety contracts as OCL constraints
- Petri nets: Formal safety analysis
  - Compute probabilities of reaching "safe conditions"

Agenda

Previous Concepts (I)

UML and UML profiles

- Semi-formal modelling language

Previous Concepts (I)

UML and UML profiles

- Semi-formal modelling language
- Tailored for specific domains by profiling
  - Stereotypes: Concepts in the target domain
  - Tagged values: Stereotype attributes
- Enriches UML semantics, commonly used for NFPs specification

Previous Concepts (I)

UML and UML profiles

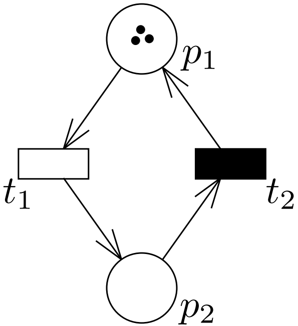
- Semi-formal modelling language
- Tailored for specific domains by profiling
  - Stereotypes: Concepts in the target domain
  - Tagged values: Stereotype attributes
- Enriches UML semantics, commonly used for NFPs specification
- Profile examples:
  - Modelling and Analysis of RT and Embedded systems (MARTE)
    - Generic Quantitative Analysis Model framework, `gaStep` stereotype (activity durations)
  - Dependability Analysis and Modelling (DAM)
  - Security Analysis and Modelling (SecAM)

Previous Concepts (II)

- UML + MARTE not suitable for performance evaluation or model-checking
- Formal models may help for this goal
  - UML + MARTE → Petri nets (namely, Generalised Stochastic PN)

Previous Concepts (II)

- UML + MARTE not suitable for performance evaluation or model-checking
- Formal models may help for this goal
  - UML + MARTE → Petri nets (namely, Generalised Stochastic PN)



GSPN

- Bipartite graph
- Places (circles, p_X)
- Transitions (bars, t_X)
  - Immediate ($t = 0$)
  - Timed (exponential, deterministic firing distributions)
- Arcs (with directions, and weight)
- Tokens

Agenda

Case Study (I): TCMS

Train Control and Management System

- Complex system distributed along the train
- Controls all train subsystems
- Composed of I/O modules plus PLCs and communication buses

Case Study (I): TCMS

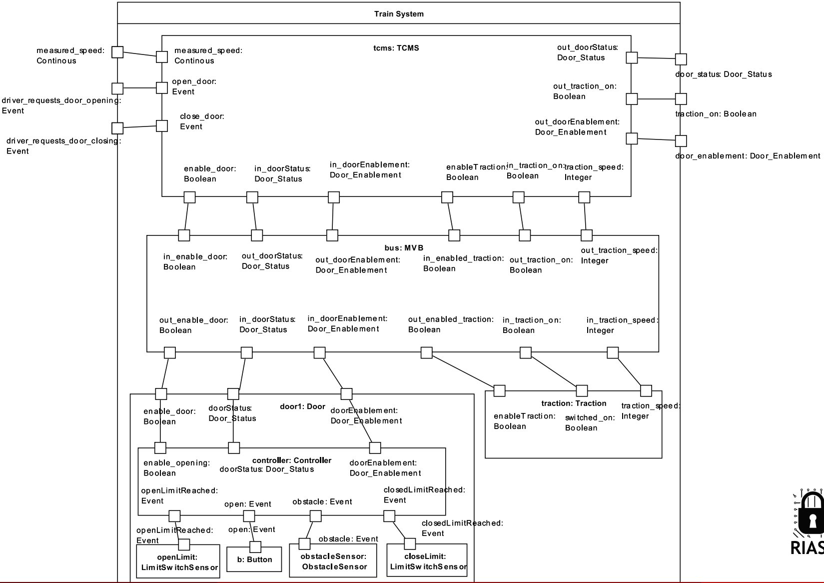
Train Control and Management System

- Complex system distributed along the train
- Controls all train subsystems
- Composed of I/O modules plus PLCs and communication buses

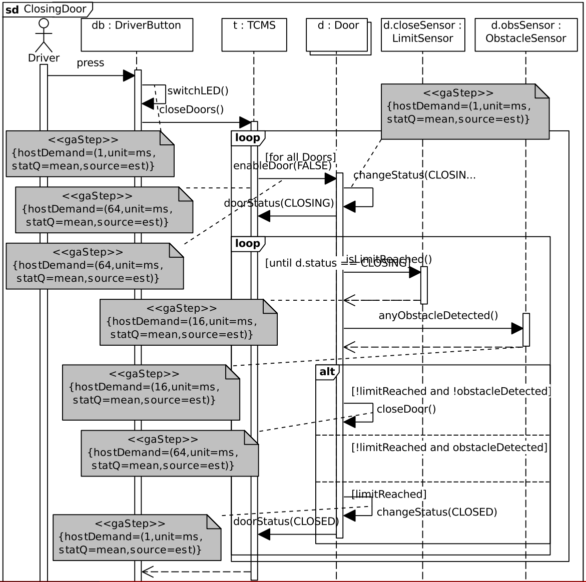
Door control management

- Several actors involved: TCMS, Door, Traction, MVB
  - TCMS: Decides whether enabling or disabling the doors
  - Door: Enabled → opened; disabled → closed
  - Traction: Deals with train movement
  - Multifunction Vehicle Bus: Communicates all components among them

Case Study (II): TCMS UML Composite Diagram



Case Study (III): Door closing UML Sequence Diagram



RIASC

Case Study (IV): Some remarks

- Model is too complex (but also the life. . .)
- The TCMS needs to be safety-certified, no matter its complexity. . .

Case Study (IV): Some remarks

- Model is too complex (but also the life. . .)
- The TCMS needs to be safety-certified, no matter its complexity. . .
- Contract-based design methodology
  - Separate components $\mathcal{C} = \langle \mathcal{I}, \mathcal{O} \rangle$: Safety and non-safety ones
  - They interact with the environment
  - Safety critical components are associated to safety contract fragments (SCF)

Case Study (IV): Some remarks

- Model is too complex (but also the life. . .)
- The TCMS needs to be safety-certified, no matter its complexity. . .
- Contract-based design methodology
  - Separate components $\mathcal{C} = \langle \mathcal{I}, \mathcal{O} \rangle$: Safety and non-safety ones
  - They interact with the environment
  - Safety critical components are associated to safety contract fragments (SCF)

Safety contract fragment $\mathcal{S}_\mathcal{C} = \langle \mathcal{A}, \mathcal{G} \rangle$

- $\mathcal{A}$: Assumptions on the component's environment
- $\mathcal{G}$: What the component guarantees under such an environment
- A component *implements* its contract if it satisfies the guarantees when the environment meets the assumptions

Safety contract transformed to OCL constraints

Agenda

Safety Contract Fragment to OCL (I)

SCF $\mathcal{S}_{\mathcal{C}} = \langle \mathcal{A}, \mathcal{G} \rangle$

- $\mathcal{A} = \mathcal{A}^{+} \bigcup \mathcal{A}^{*}$ (assumptions, input ports)
- $\mathcal{G} = \mathcal{G}^{+} \bigcup \mathcal{G}^{*}$ (guarantees, output ports)

Safety Contract Fragment to OCL (I)

SCF $\mathcal{S}_{\mathcal{C}} = \langle \mathcal{A}, \mathcal{G} \rangle$

- $\mathcal{A} = \mathcal{A}^{+} \bigcup \mathcal{A}^{*}$ (assumptions, input ports)
- $\mathcal{G} = \mathcal{G}^{+} \bigcup \mathcal{G}^{*}$ (guarantees, output ports)

OCL

- Express constraints within UML models
- Defined over a context that describes where constraint is acting
- OCL invariant: $\mathcal{R} = \langle \mathcal{X}, \mathcal{V} \rangle$
  - $\mathcal{X}$: Context
  - $\mathcal{V} = \langle ls, rs \rangle$ (joined by a boolean or `implies` operator)

Safety Contract Fragment to OCL (I)

SCF $\mathcal{S}_\mathcal{C} = \langle \mathcal{A}, \mathcal{G} \rangle$

- $\mathcal{A} = \mathcal{A}^+ \bigcup \mathcal{A}^*$ (assumptions, input ports)
- $\mathcal{G} = \mathcal{G}^+ \bigcup \mathcal{G}^*$ (guarantees, output ports)

OCL

- Express constraints within UML models
- Defined over a context that describes where constraint is acting
- OCL invariant: $\mathcal{R} = \langle \mathcal{X}, \mathcal{V} \rangle$
  - $\mathcal{X}$: Context
  - $\mathcal{V} = \langle ls, rs \rangle$ (joined by a boolean or `implies` operator)

Given a component $\mathcal{C}$, and $\mathcal{S}_\mathcal{C} = \langle \mathcal{A}, \mathcal{G} \rangle \rightarrow \mathcal{R} = \langle \mathcal{X}, \mathcal{V} \rangle$, where $\mathcal{X} = \mathcal{C}$ and $\mathcal{V} = \langle \mathcal{A}, \mathcal{G} \rangle$

RIASC

Safety Contract Fragment to OCL (II): Examples (1)

SR1. *The door opening is not enabled when the traction is on or the train speed is distinct than zero*

- $\mathcal{S}_1 = \langle (traction\ OR\ (tractionSpeed \neq 0)), (NOT\ enableOpening) \rangle$ (TCMS)

Safety Contract Fragment to OCL (II): Examples (1)

SR1. *The door opening is not enabled when the traction is on or the train speed is distinct than zero*

- $\mathcal{S}_1 = \langle (traction\ OR\ (tractionSpeed \neq 0)), (NOT\ enableOpening) \rangle$ (TCMS)

```
context TCMS_SR1
   inv: (traction or tractionSpeed <> 0)
             implies not enableOpening
```

Safety Contract Fragment to OCL (II): Examples (1)

SR1. *The door opening is not enabled when the traction is on or the train speed is distinct than zero*

- $\mathcal{S}_1 = \langle (traction\ OR\ (tractionSpeed \neq 0)), (NOT\ enableOpening) \rangle$ (TCMS)

```
context TCMS_SR1
   inv: (traction or tractionSpeed <> 0)
                implies not enableOpening
```

SR2. *The door must be closed but remains open when some obstacle has been detected*

- $\mathcal{S}_2 = \langle obstacle, doorStatus = opening \rangle$ (DoorController)

Safety Contract Fragment to OCL (II): Examples (1)

SR1. *The door opening is not enabled when the traction is on or the train speed is distinct than zero*

- $\mathcal{S}_1 = \langle (traction\ OR\ (tractionSpeed \neq 0)), (NOT\ enableOpening)\rangle$ (TCMS)

```
context TCMS_SR1
   inv: (traction or tractionSpeed <> 0)
            implies not enableOpening
```

SR2. *The door must be closed but remains open when some obstacle has been detected*

- $\mathcal{S}_2 = \langle obstacle, doorStatus = opening\rangle$ (DoorController)

```
context DoorController_SR2
   inv: obstacle
            implies (doorStatus = opening)
```

Safety Contract Fragment to OCL (II): Examples (2)

SR3. *The door is closed when the door opening is enabled and the close event is received*

- $\mathcal{S}_3 = \langle (enableOpening\ AND\ close), doorStatus = isClosed \rangle$ (`Door`)

Safety Contract Fragment to OCL (II): Examples (2)

SR3. *The door is closed when the door opening is enabled and the close event is received*

- $\mathcal{S}_3 = \langle (enableOpening\ AND\ close), doorStatus = isClosed \rangle$ (Door)

```
context Door_SR3
   inv: (enableOpening and close)
            implies doorStatus = isClosed
```

Safety Contract Fragment to OCL (II): Examples (2)

SR3. *The door is closed when the door opening is enabled and the close event is received*

- $\mathcal{S}_3 = \langle (enableOpening\ AND\ close), doorStatus = isClosed \rangle$ (Door)

```
context Door_SR3
   inv: (enableOpening and close)
             implies doorStatus = isClosed
```

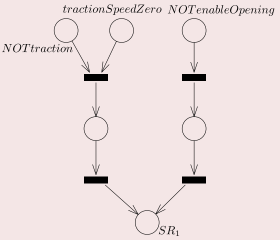
So, until here we have expressed safety contracts using OCL within UML. Now, **we express these constraints using Petri nets to verify them, check next slide!**

Agenda

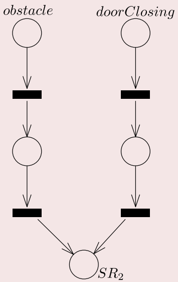
From OCL constraints to Petri nets (I)

- Places representing each condition in the OCL invariant
- $p \Rightarrow q \Leftrightarrow \neg p \vee q$
- Compute the (output) place marking probabilities (by simulating)

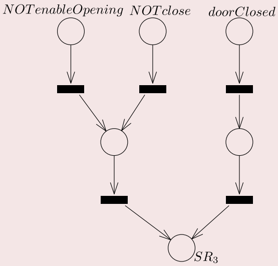
SR1



SR2



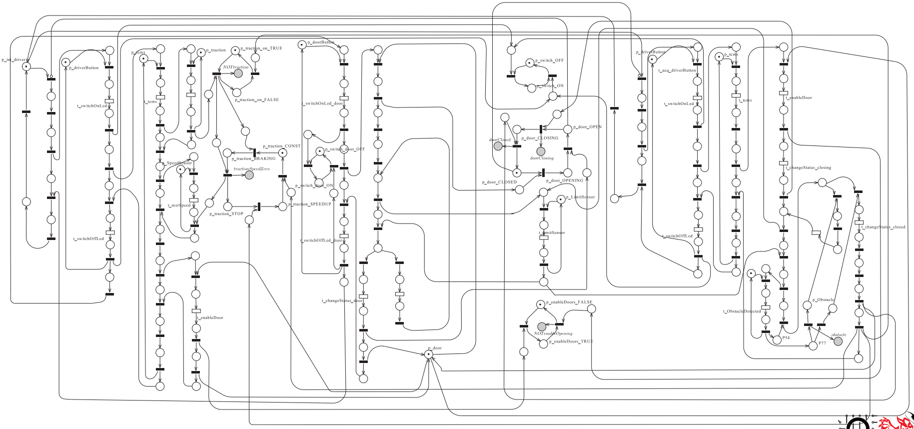
SR3



RIASC

From OCL constraints to Petri nets (II)

Petri net of the door controller



Agenda

Related Work (I)

Formal expression of contracts

- Requirements Specification Language, Othello (based on LTL), Modal Transmission Systems
- **Advantage**: expressiveness
- **Disadvantages**:
  - Needed to learn a new formalism each time used
  - Lack of verification (some of them)

Related Work (I)

Formal expression of contracts

- Requirements Specification Language, Othello (based on LTL), Modal Transmission Systems
- **Advantage**: expressiveness
- **Disadvantages**:
  - Needed to learn a new formalism each time used
  - Lack of verification (some of them)

Our proposal

- Enables to analyse also non-functional properties
- Safety contract fragments expressed as OCL
- Could complement OCRA analysis (non-functional properties)
- Strong, weak assumptions: Weak implicitly described with MARTE

Agenda

Conclusions and Future Work

- Contract-based design: Good approach for safety-critical systems
- Safety contracts expressed as OCL, and verified into the PN
- All this performed at design phase! → saves budget overruns

Future Work

- Increase complexity of contracts expressed by OCL
  - Event order? Temporal information?
- Safety assessment methodology + a tool to automatise the process

Conclusions and Future Work

- Contract-based design: Good approach for safety-critical systems
- Safety contracts expressed as OCL, and verified into the PN
- All this performed at design phase! → saves budget overruns

Future Work

- Increase complexity of contracts expressed by OCL
  - Event order? Temporal information?
- Safety assessment methodology + a tool to automatise the process

A last remark

- Final effort must be done in implementation
  - Assure it matches the system model, or otherwise it may lead the system to an unsafe system

- **Acknowledgements**: ARTEMIS JU nSafeCer, n^o 295373

Model-based Verification of Safety Contracts

Elena Gómez-Martínez[†], **Ricardo J. Rodríguez**[‡], Leire Etxeberria Elorza[*], Miren Illarramendi Rezabal[*], Clara Benac Earle[†]

{egomez,cbenac}@babel.ls.fi.upm.es, rj.rodriguez@unileon.es,

{letxeberria,millarramendi}@mondragon.edu

[†]Technical University of Madrid
Madrid, Spain

[‡]RIASC, University of León
León, Spain

[*]Mondragon Unibertsitatea
Arrasate-Mondragón, Spain

September 1, 2014

1st International Workshop on Safety & Formal Methods
Grenoble (France)