25th anniversary
annual INCOSE
international symposium
Seattle, WA
July 13 - 16, 2015

Integrating an Upgraded Constituent System in a System of Systems: A SysML case study

Claire Ingram, John Fitzgerald
Newcastle University, UK
Jon Holt, Scarecrow Consultants, UK
Nico Plat, West Consulting, Netherlands

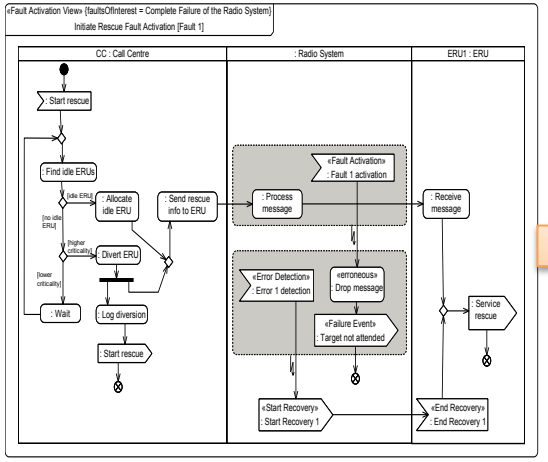
Newcastle University

Introduction

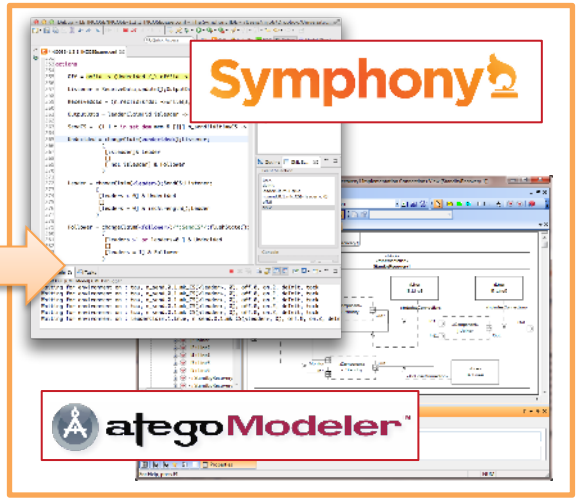
- SoSs are comprised of elements that are themselves independent systems
- Often exhibit:
  - Operational & managerial independence
  - Distribution
  - Reliance placed on emergence
  - Evolution
- Challenging aspects include:
  - Operational & Managerial Independence of Constituent Systems
  - Complexity of confirming/refuting SoS-level properties
  - Semantic heterogeneity

COMPASS

www.thecompassclub.org




```
process CallCentreProc = begin
actions
 MERGE1(r) =
  (dcl e: set of ERUId @ e := findIdleERUs();
   (do e = {} -> DECISION2(r)   |
        e <> {} ->  (dcl e1: ERUId @
                          e1 :=  allocateIdleERU(e, r);
                                 MERGE2(e1, r))
   end)) …

process InitiateRescue =
CallCentreProc [| SEND_CHANNELS |]
RadioSystemProc  [| RCV_CHANNELS |] ERUsProc
```



Architectural Modelling

- SoS Modelling Frameworks
- … instantiated to domains
- **SoS Modelling patterns** & profiles, e.g. Fault-Error-Failure
- Guidelines on negotiation, requirements, integration, test, etc.

Underpinning Formalisms

- Behavioural semantics of SoS
- Tight link to modelling frameworks
- Cope with multiple paradigms.
- Compositional Design
- Dynamic response to adaptation & evolution
- Covering cyber elements, physical, human, economic, social, …

Tool-supported V&V:

- Exploration of Design Space
- Efficient verification by model-checking and proof
- Test generation
- Simulation
- Tools Robustness
- Conformance during evolution, and emergence

Newcastle University

25th anniversary annual INCOSE international symposium
Seattle, WA
July 13 - 16, 2015

Outline

1. Integration in an SoS
2. Introduction to the case study
3. Application of the Integration Framework
   - Constituent system identification process
   - Integration process
   - Validation process
4. Conclusions

Outline

1. **Integration in an SoS**
2. Introduction to the case study
3. Application of the Integration Framework
   - Constituent system identification process
   - Integration process
   - Validation process
4. Conclusions

Integration in an SoS

- An SoS is a system comprising components which are each independent systems themselves
  - Constituent parts operated and managed separately
  - Large-scale, distributed
  - Emergent behaviour
  - Continually evolving

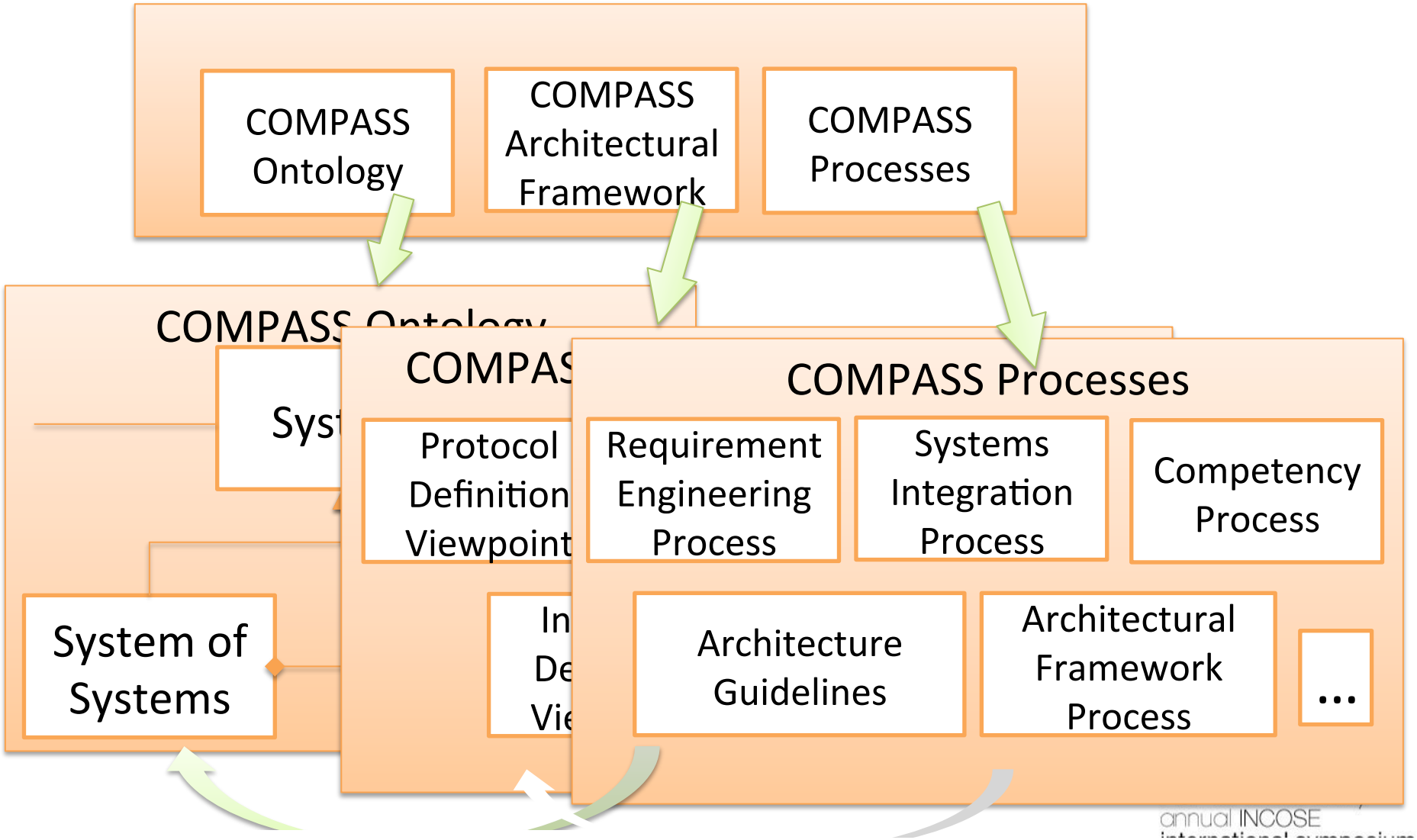
Integration in an SoS

- Integration a key challenge for an SoS
  - SoSs are continually changing
- Several activities covered by “integration”
  - e.g. ensuring no negative effects felt after a changed/new constituent is integrated
- Modelling is important for SoSs
  - Scale and SoS operating environment difficult to replicate in a lab
  - Testing accurately difficult/expensive

Architectural Modelling

- We follow COMPASS architectural approach
  - patterns and guidelines
- Use collections of *modelling patterns* to define SoS structure and behaviour
- COMPASS architectural modelling approach also includes *guidelines* for SoS integration and development lifecycles

COMPASS Approach



Outline

1. Integration in an SoS
2. **Introduction to the case study**
3. Application of the Integration Framework
   - Constituent system identification process
   - Integration process
   - Validation process
4. Conclusions

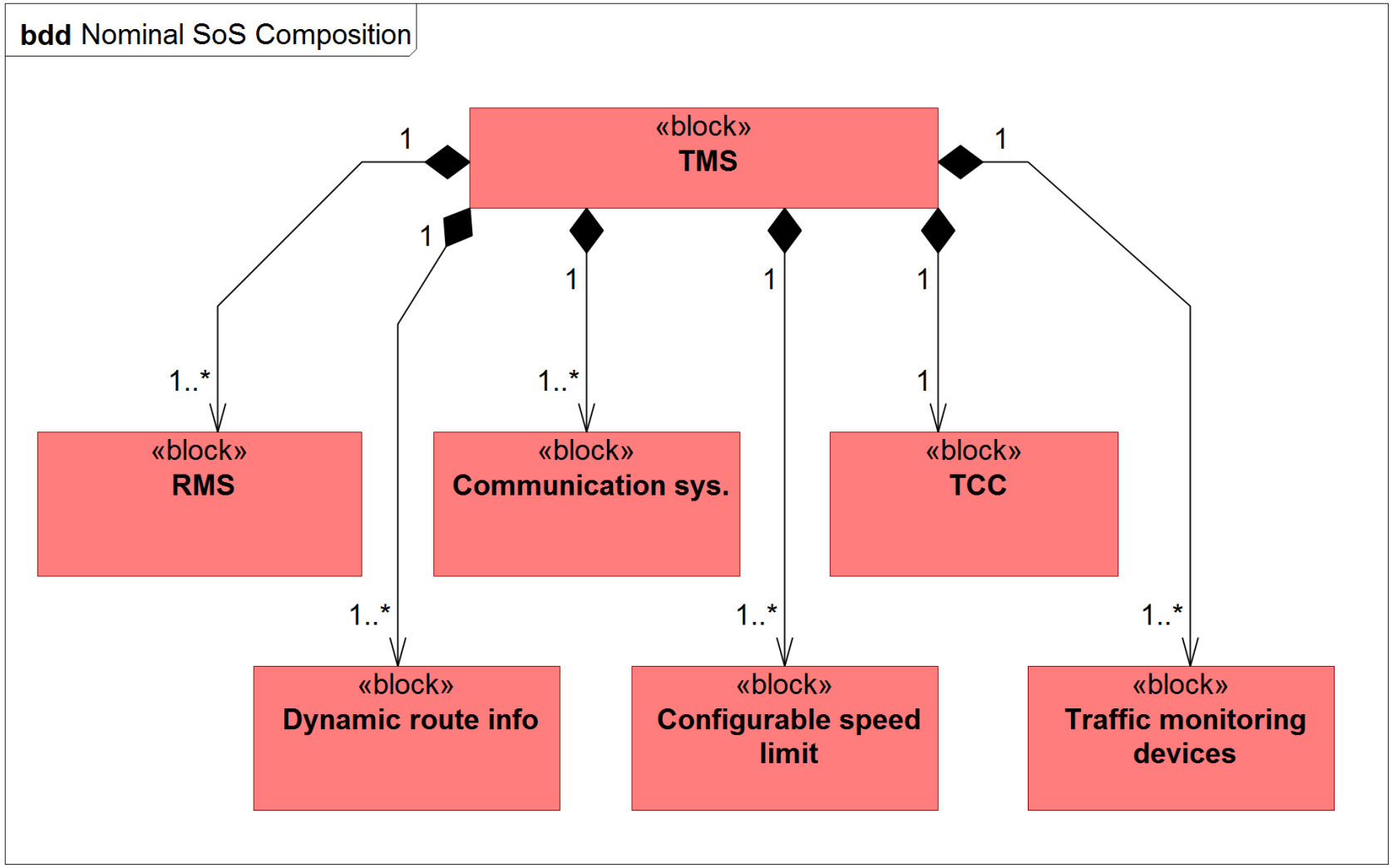
Case Study: Road Traffic Management

- Traffic Management System (TMS) collects information about conditions and traffic flow
- Takes actions to achieve traffic behaviour goals:
  - improving efficiency of the road network
  - ensuring road safety
  - reducing the impact accidents, blockages etc
  - reducing environmental impact
- Inter-urban road network in the Netherlands

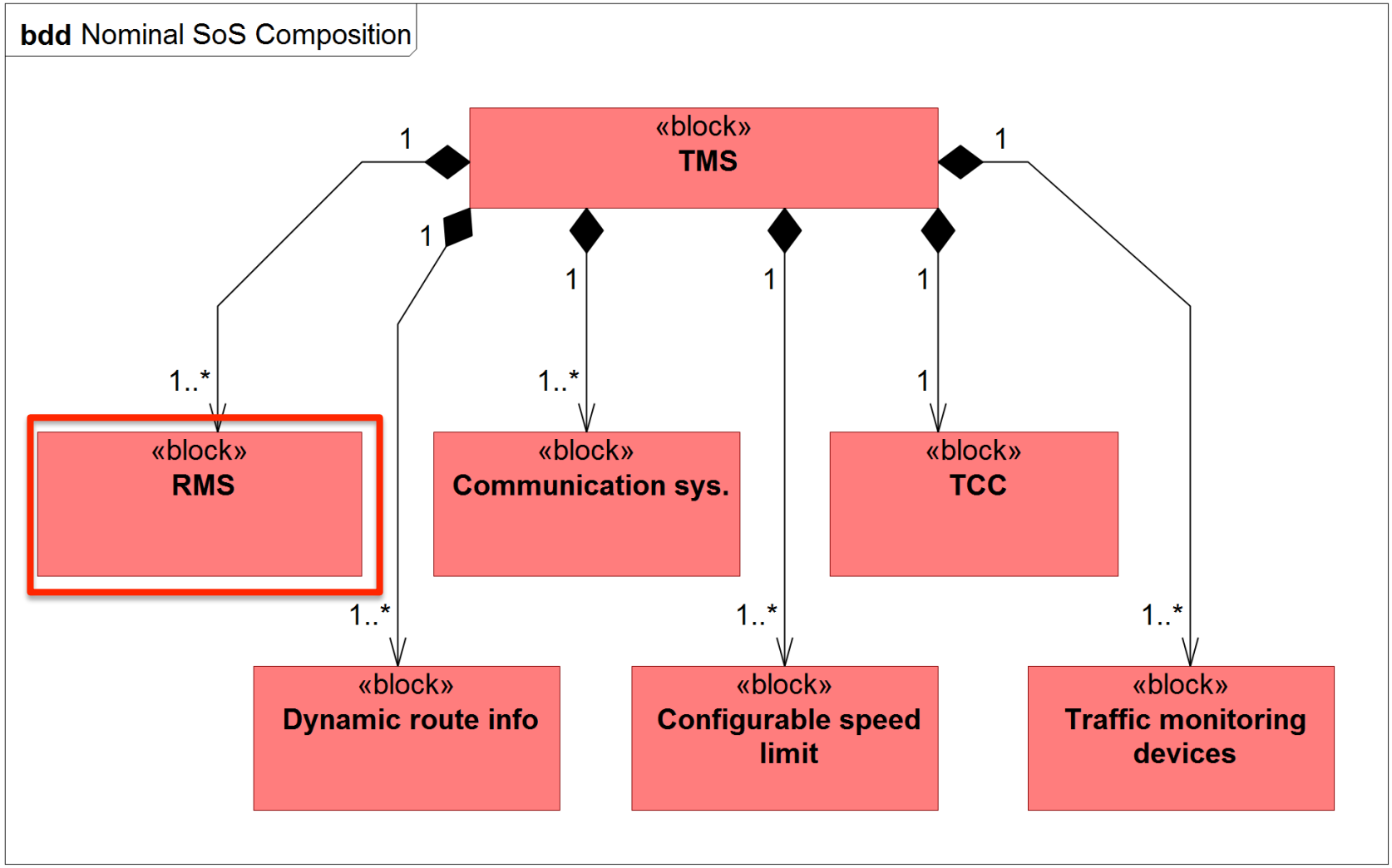
Case Study: Road Traffic Management

- Wide variety of CSs
  - Traffic monitoring systems
  - Systems for influencing traffic flow
  - Control rooms
- Interact with TMSs operated by third parties in adjacent regions

Road Traffic Management

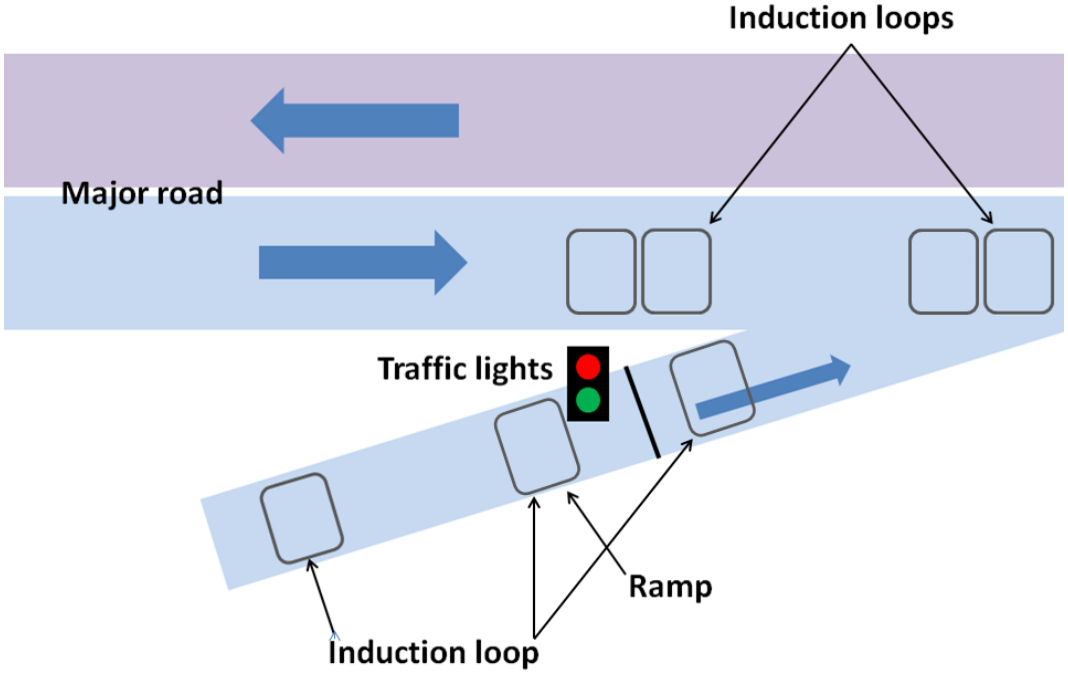


Road Traffic Management



Case Study: Road Traffic Management

- Ramp meter system (RMS)
  - Two-phase traffic lights control vehicle rate
  - Prevents bottlenecks and improves vehicle distribution
  - Can reduce accidents caused by high speed merges.



Case Study: Road Traffic Management

- An RMS typically has access to data about traffic in its immediate vicinity
- The RMS operates in one of several modes:
  - Fixed-time mode
  - Responsive mode – responds to current traffic conditions:
    - Responsive/Isolated mode
    - Collaborative mode

Outline

1. Integration in a SoS
2. Introduction to the case study
3. **Application of the Integration Framework**
   - Constituent system identification process
   - Integration process
   - Validation process
4. Conclusions

Scenario

- We assume that we are adding a newly evolved version of a single RMS
- There are already RMSs in the SoS
- We use the COMPASS Integration Framework to model this scenario
- We use a requirements process designed for SoSs, SoS-ACRE, to capture requirements

COMPASS Integration Framework

CIF provides a coherent set of views & concepts to:

- Identify a number of Viewpoints required for the integration of SoSs
- Describe sequences of activities that should be carried out
- Provide guidance on how the activities should be implemented

COMPASS Integration Framework

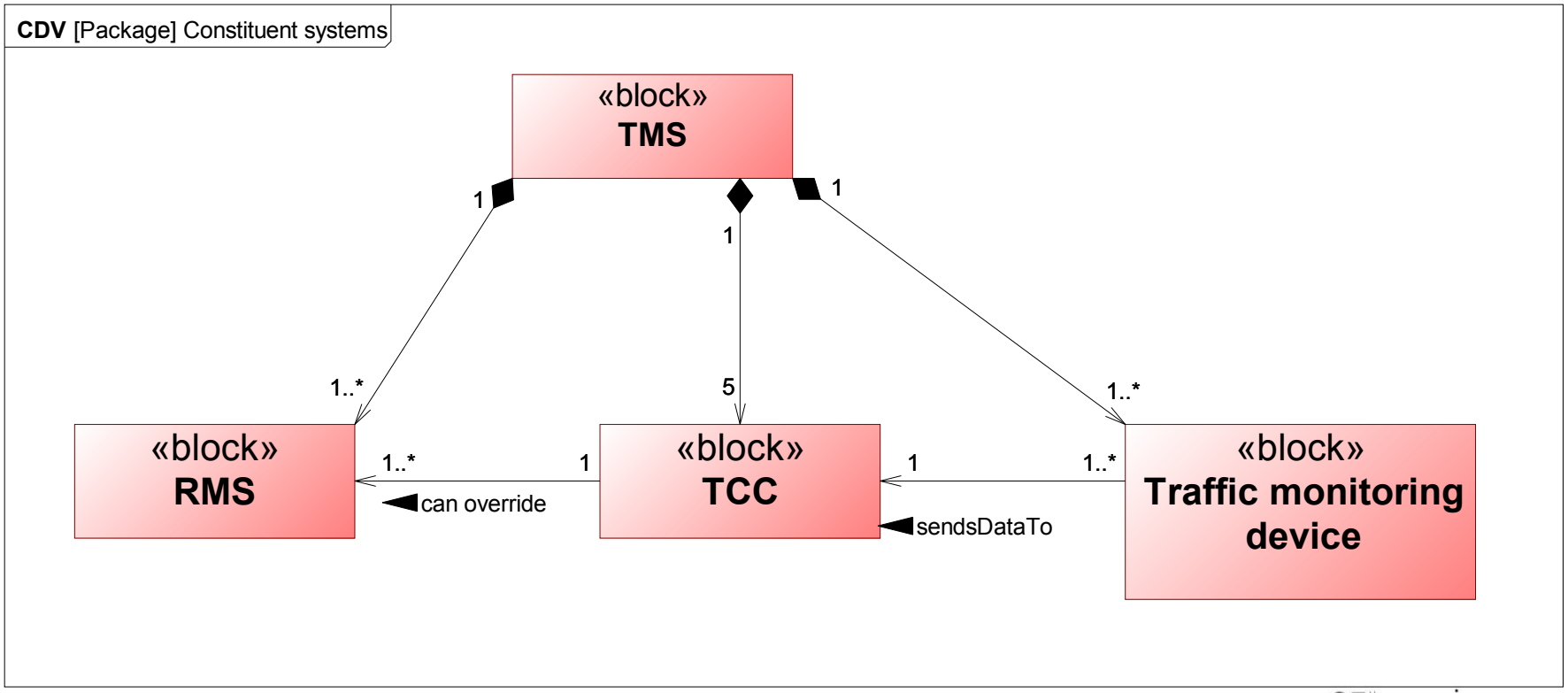
Modelling a change in a single RMS:

1. **Constituent system identification**. Captures and models requirements and CSs.
2. **Integration**. Captures and models interfaces and how they connect.
3. **Validation**. Identifies scenarios for validating the models.

Repeat the process twice: once for existing RMS (RMSv1), once for new version (RMSv2).

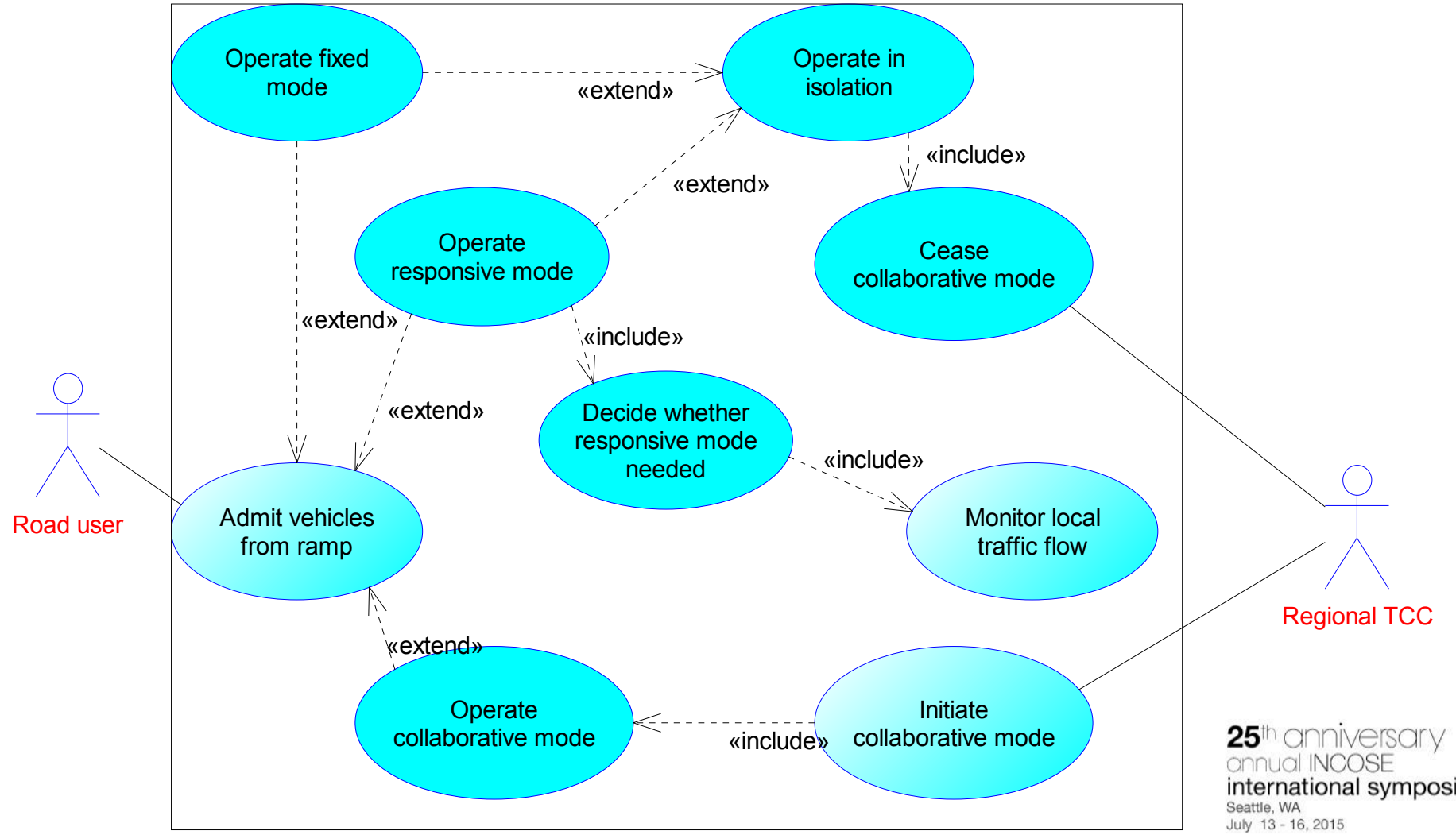
Constituent System Identification

Context Definition Viewpoint



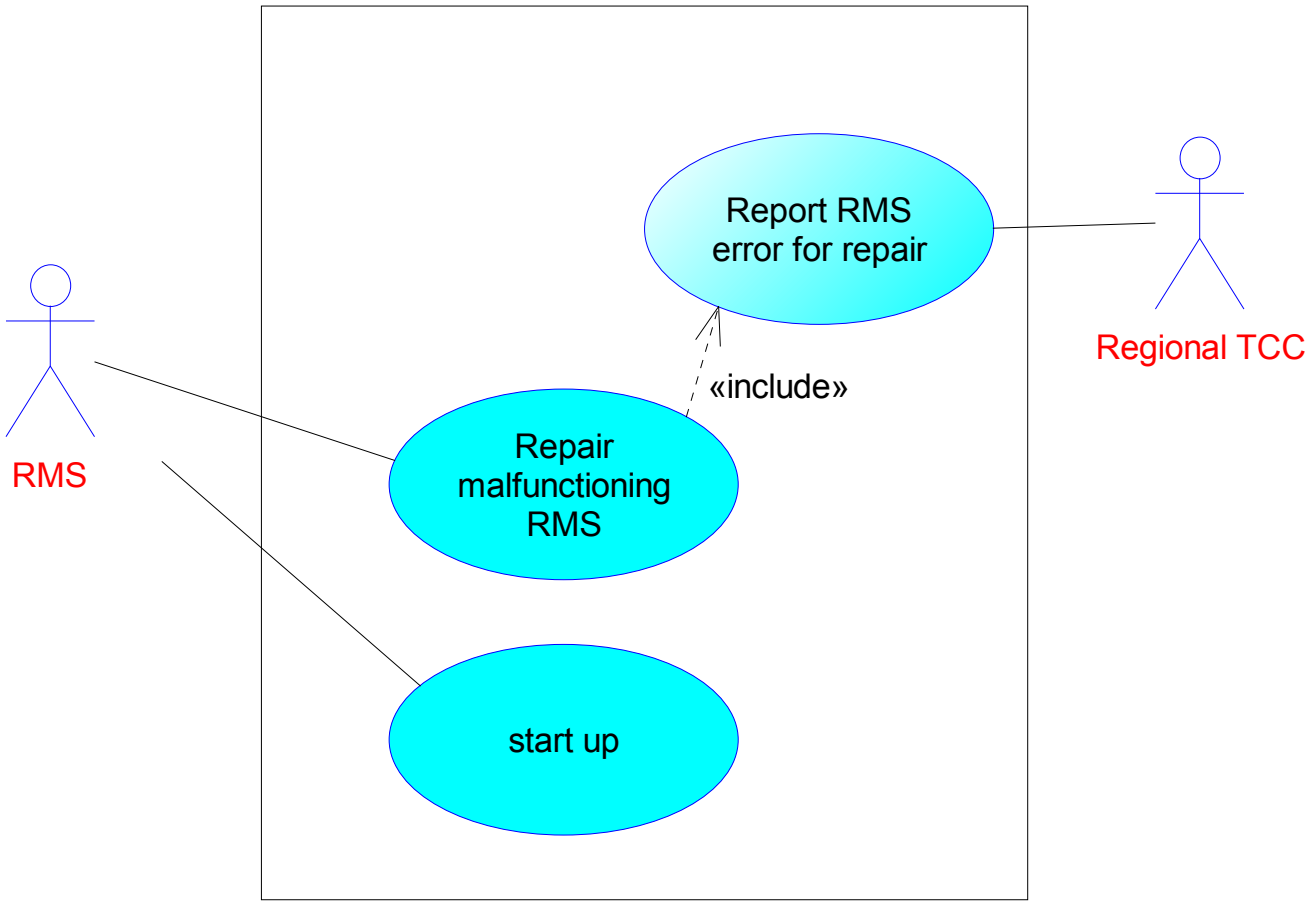
Constituent System Identification

Requirement Context Viewpoint: RMS POV



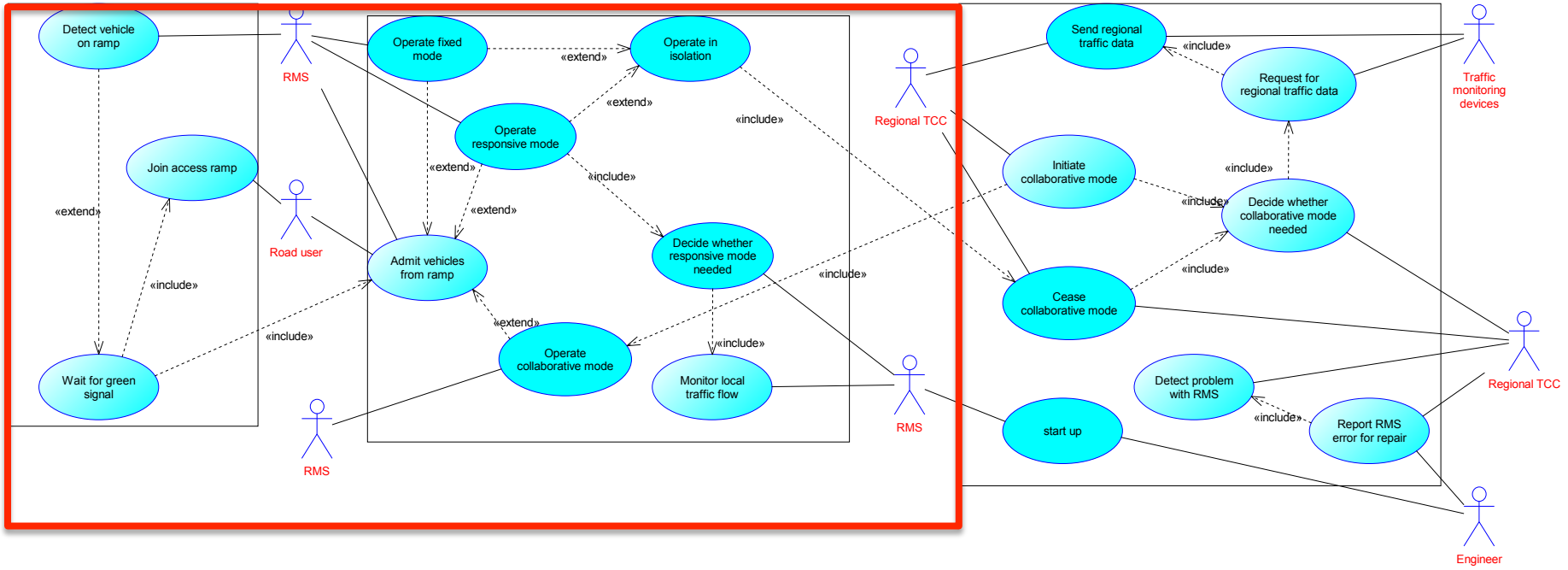
Constituent System Identification

Requirements Context Viewpoint: engineer POV



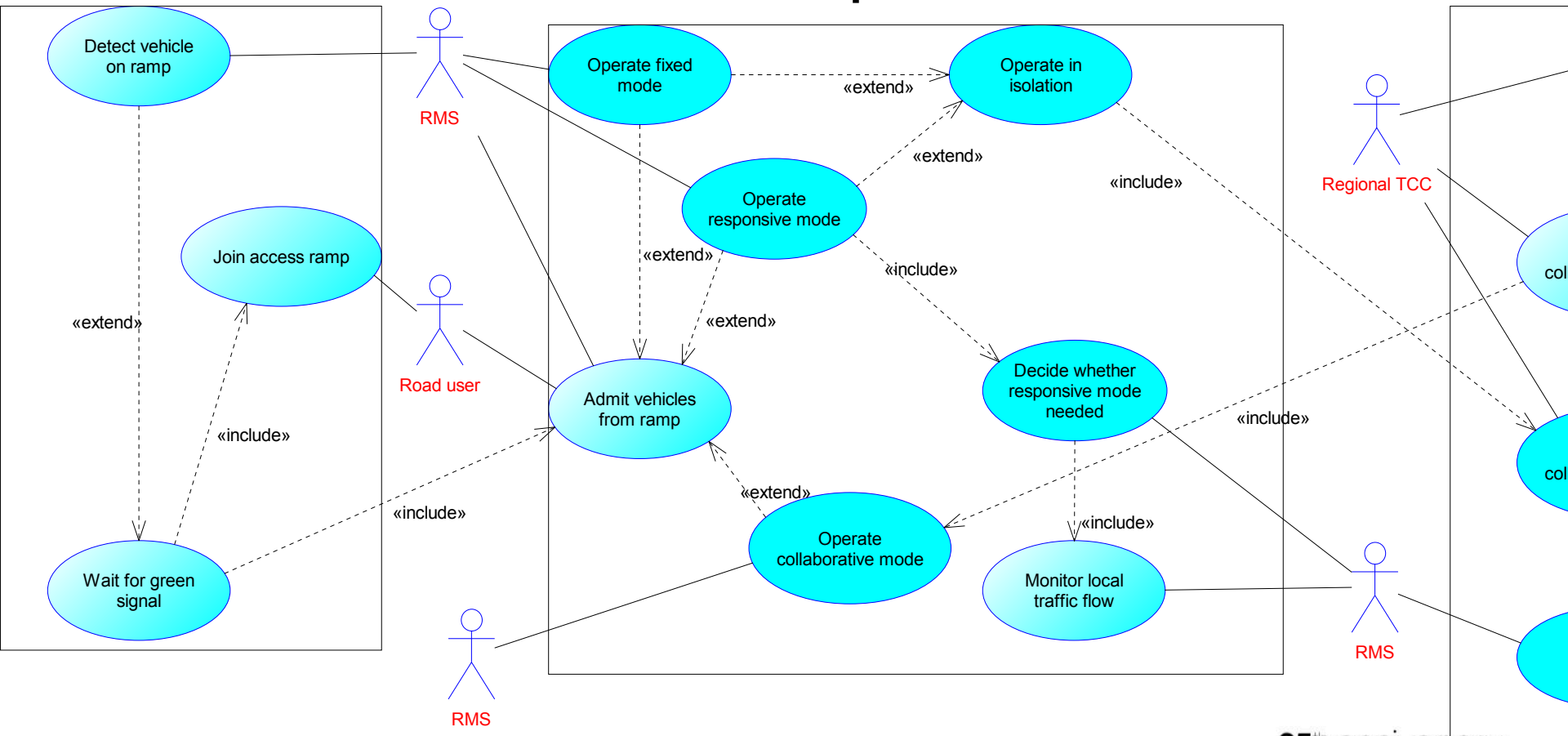
Constituent System Identification

Context Interaction Viewpoint



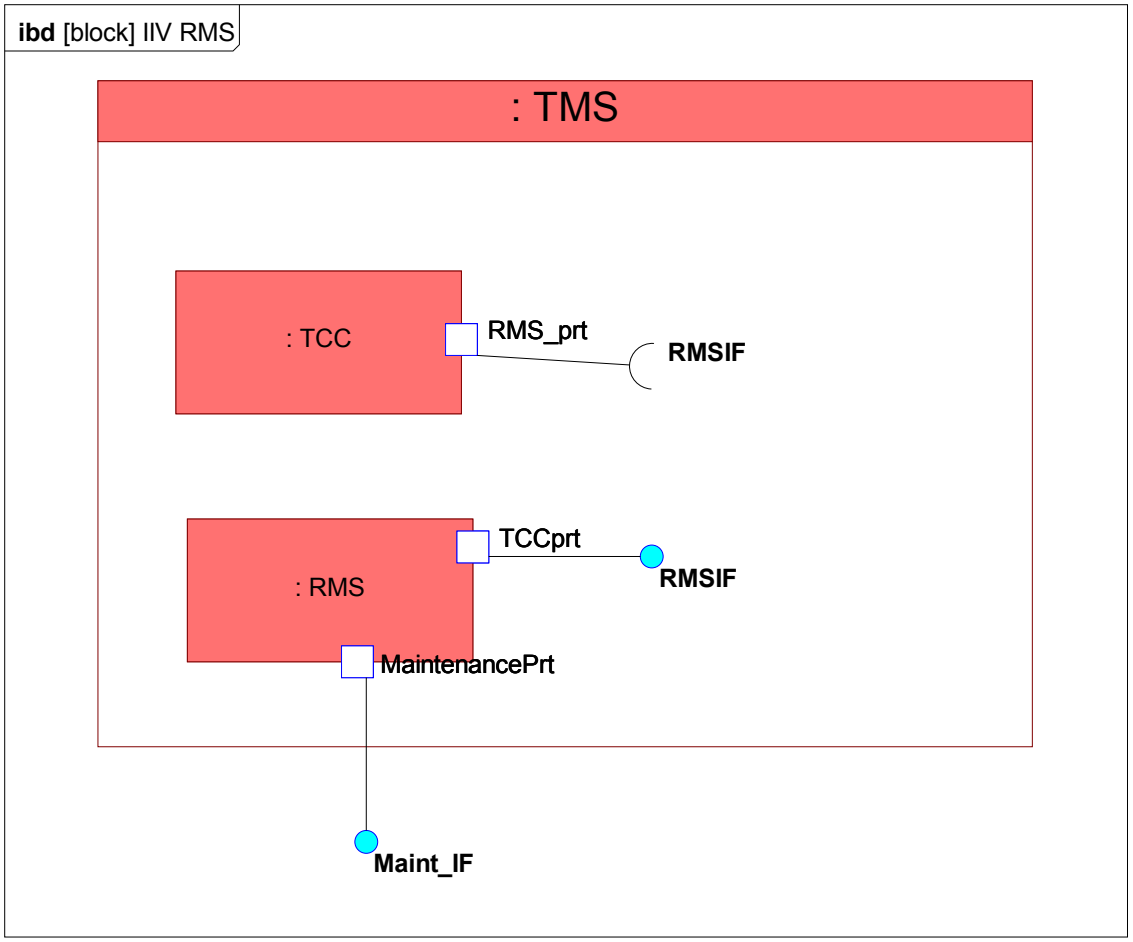
Constituent System Identification

Context Interaction Viewpoint



Constituent System Identification

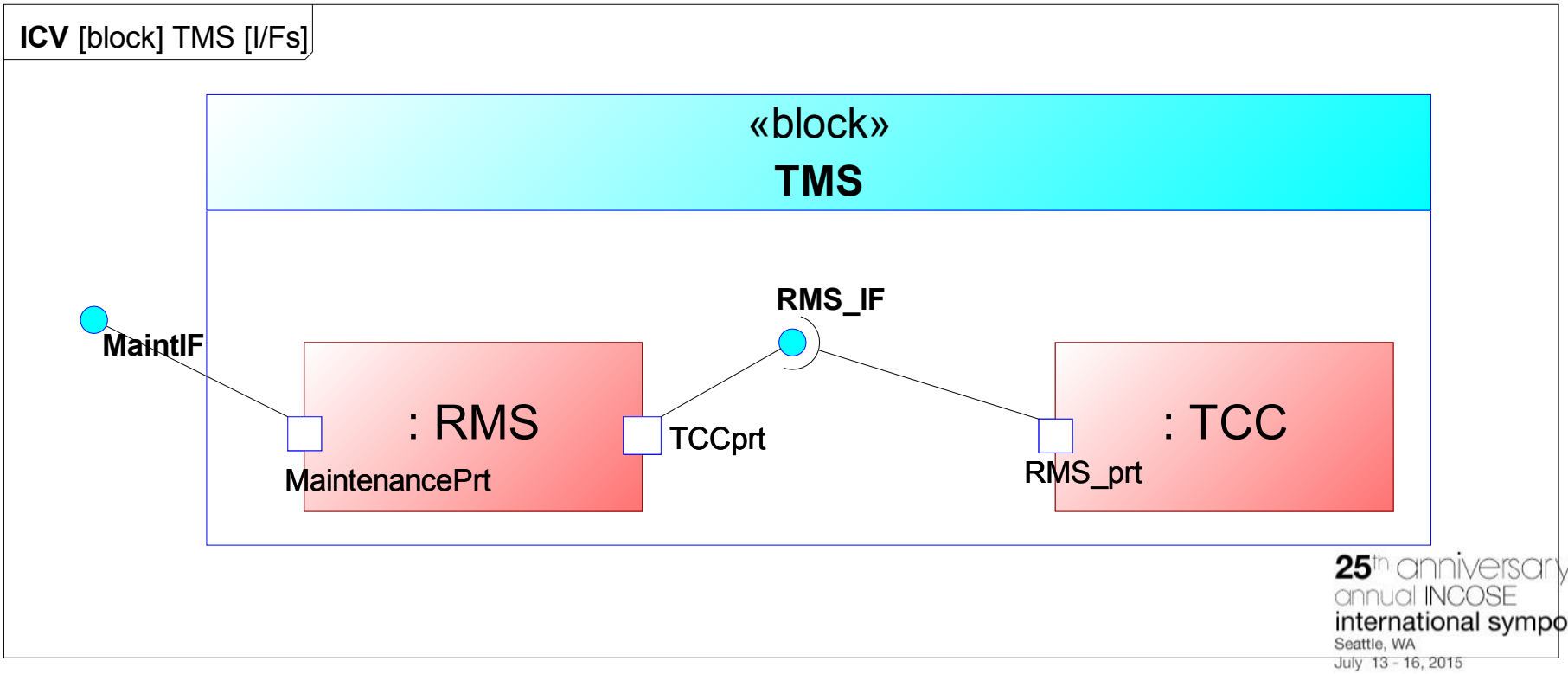
Interface Identification Viewpoint



Integration Process

Interface Connectivity Viewpoint

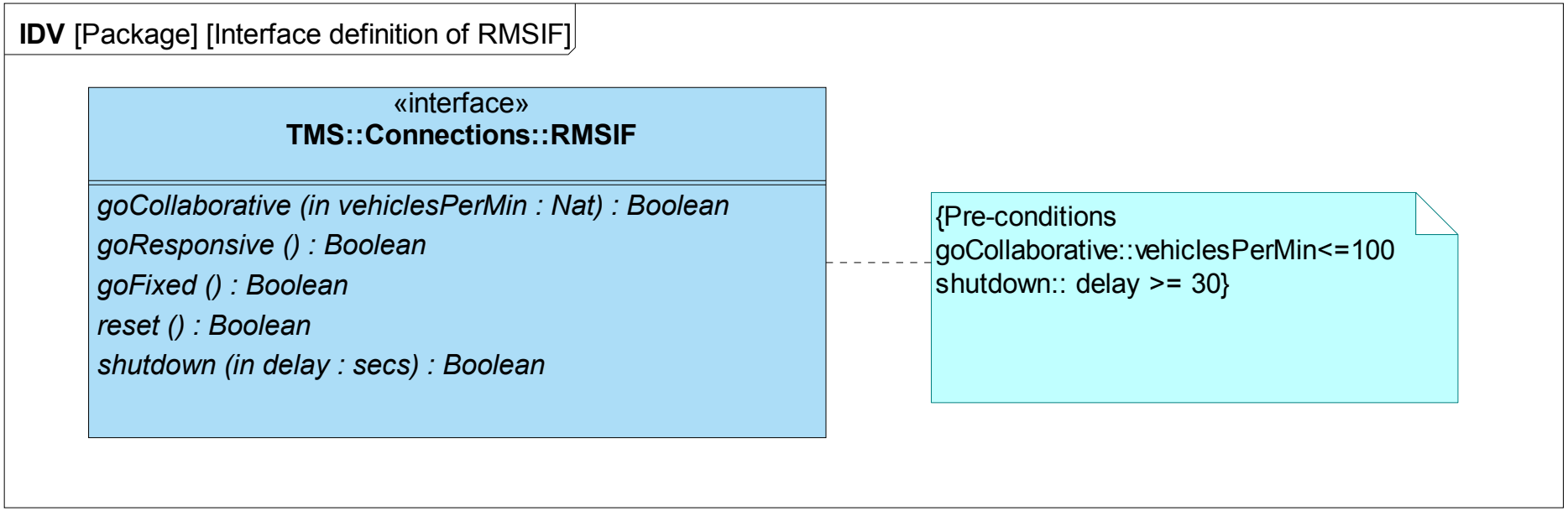
- defines how the interfaces in the IIV are connected



Integration Process

Interface Definition Viewpoint

- Defines structure of each interface in IIV



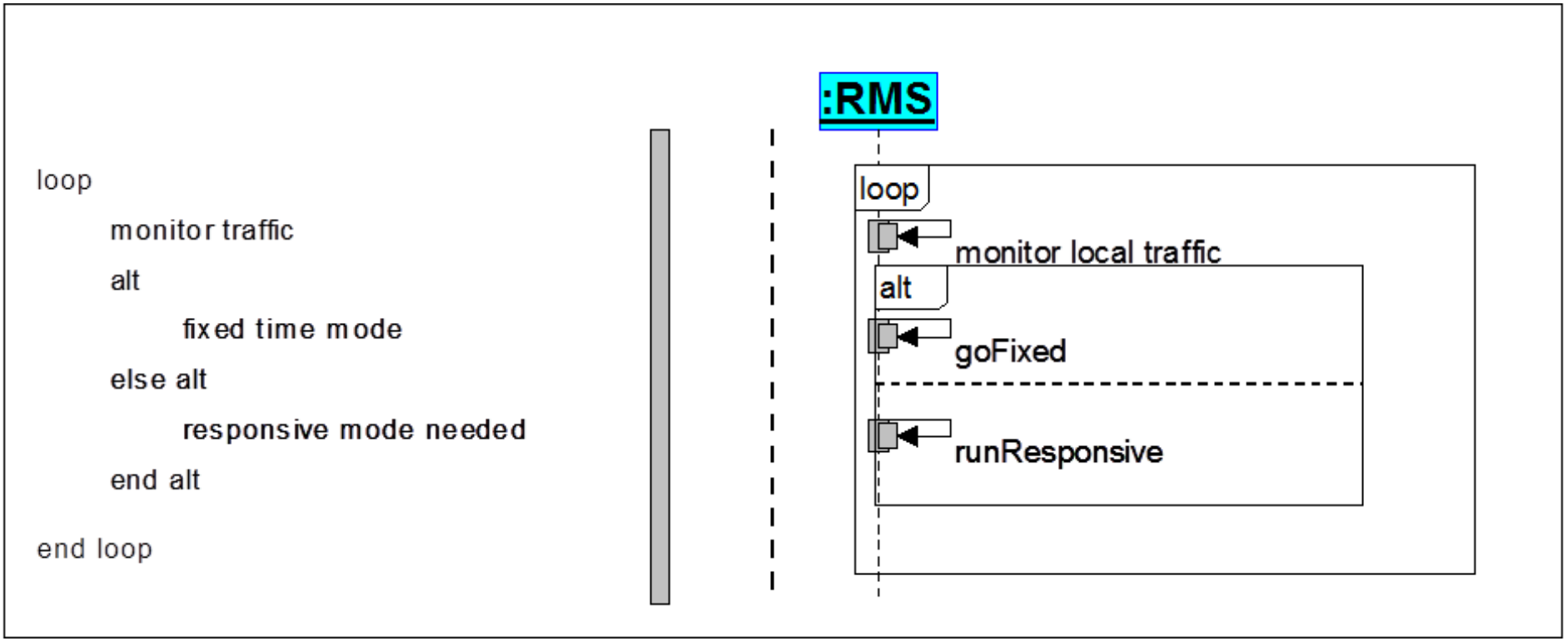
Integration Process

Interface Behaviour Viewpoint

- IBV defines interactions between interfaces we have already identified
- Implemented as a selection of sequence diagrams
- Model both acceptable and unacceptable sequences
- Five scenarios are developed and one IBV developed for each

Integration Process

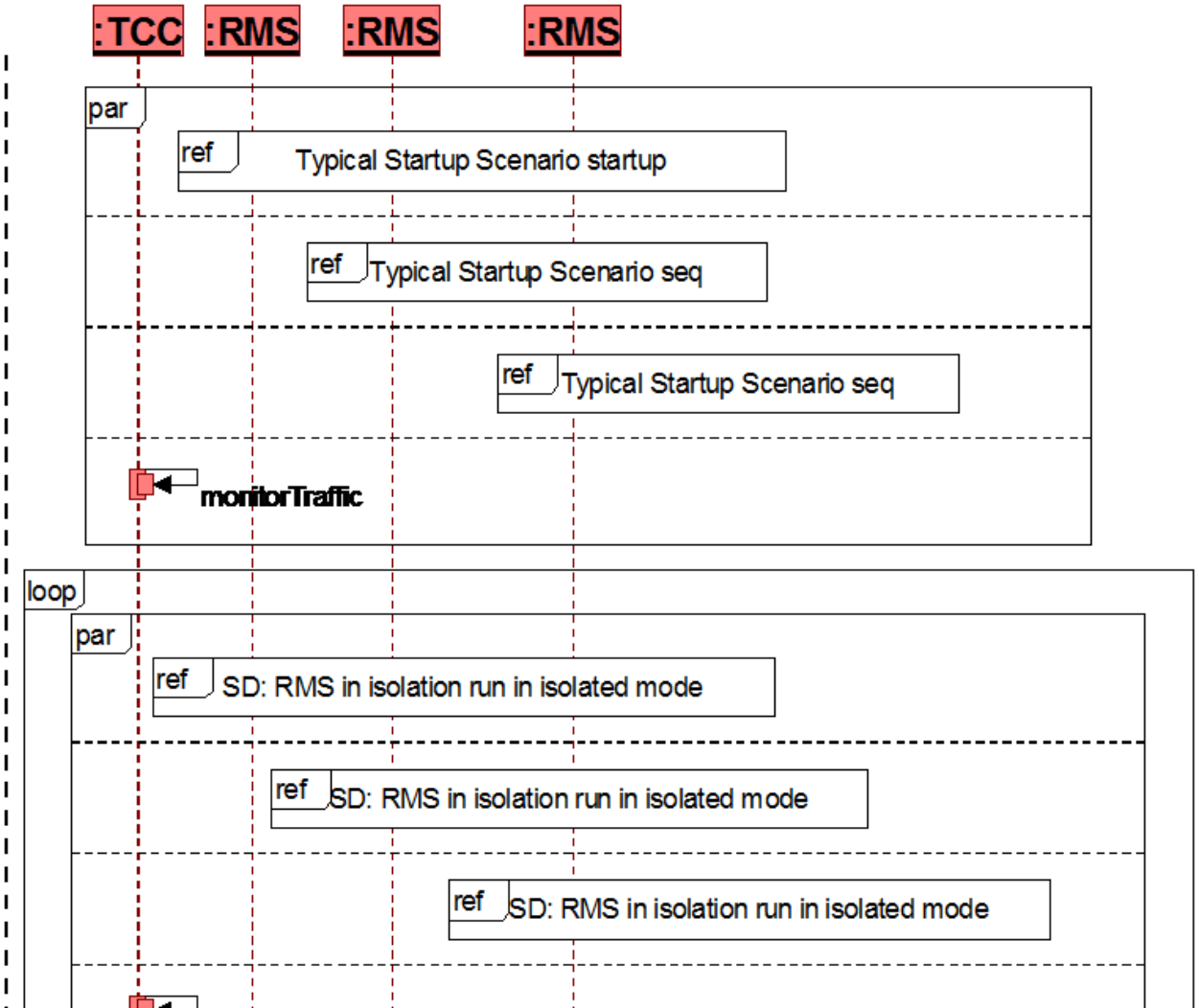
Interface Behaviour Viewpoint: an RMS operates in isolated mode



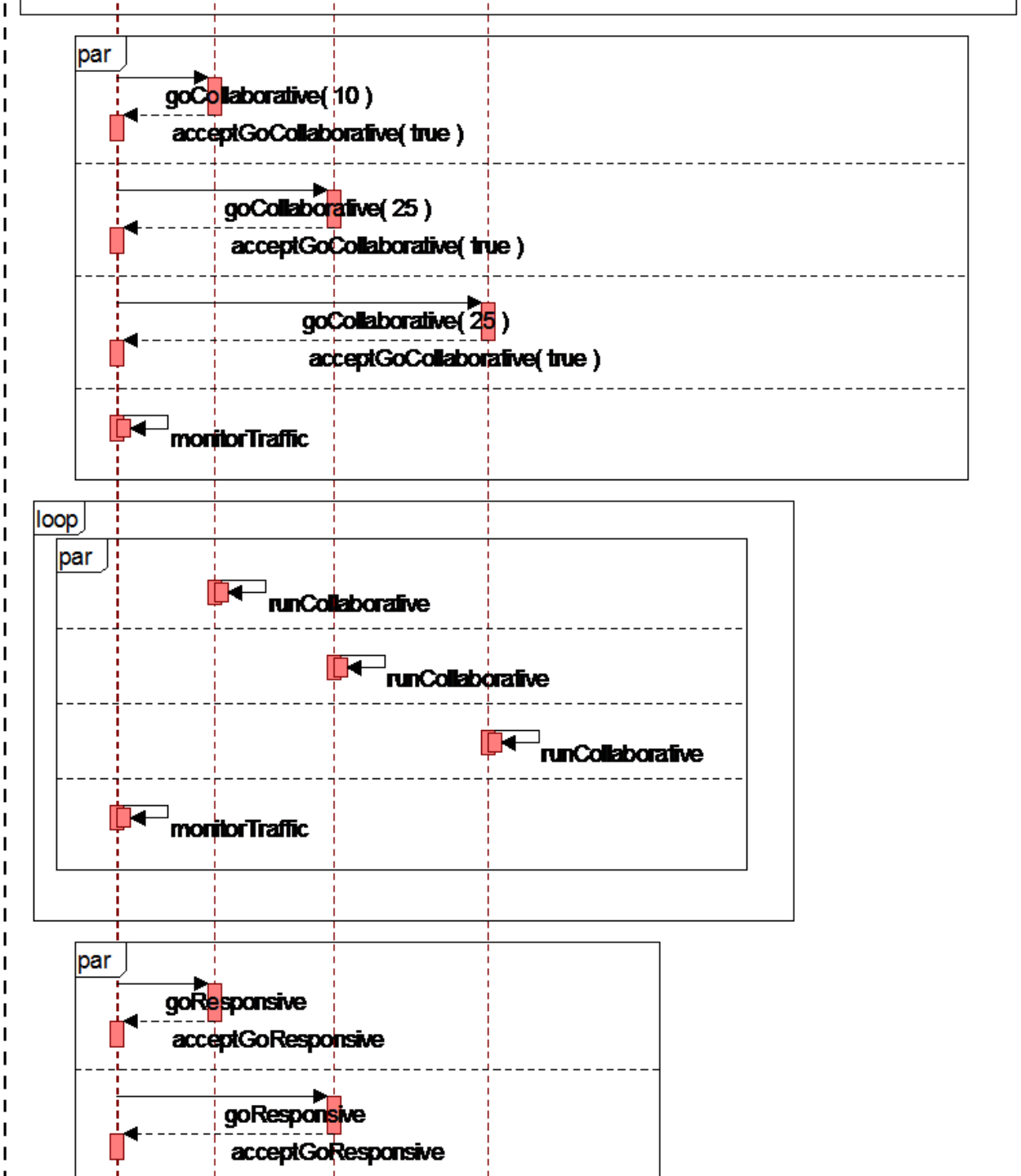
Integration Process

Interface Behaviour Viewpoint: acceptable behaviour

par
    startup
also par
    seq
also par
    seq
also par
    monitor regional traffic
end par
loop
    par
        run in isolated mode
    also par
        run in isolated mode
    also par
        run in isolated mode
    also par
        monitor regional traffic

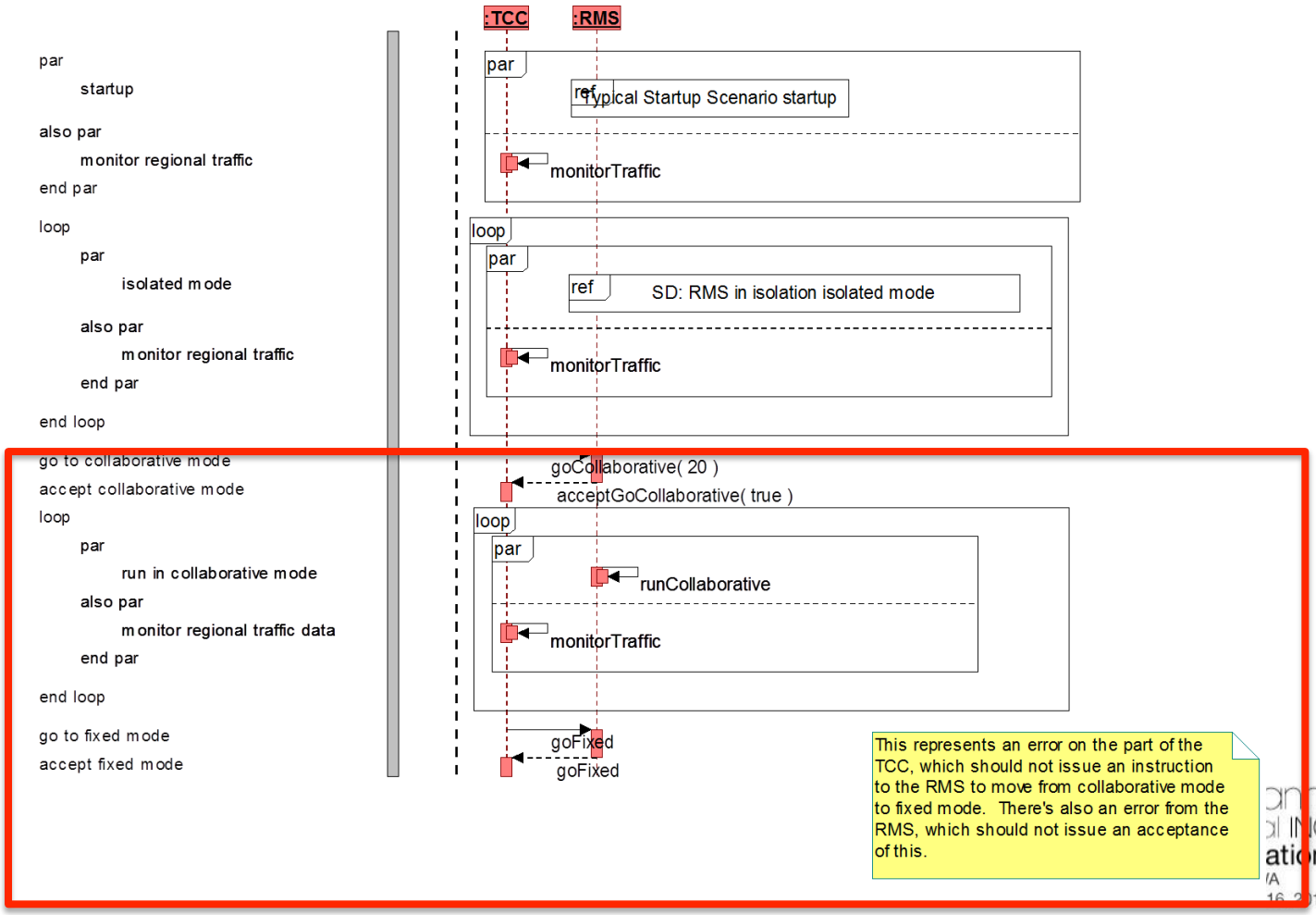


```
par
    go to collaborative mode
    accept collaborative mode
also par
    go to collaborative mode
    accept collaborative mode
also par
    go to collaborative mode
    accept collaborative mode
also par
    monitor regional traffic
end par

loop
    par
        run collaborative
    also par
        run collaborative mode
    also par
        run collaborative mode
    also par
        monitor regional traffic
    end par

end loop

par
    go to responsive mode
    accept responsive mode
also par
    go to responsive mode
    accept responsive mode
```



Integration Process

Interface Behaviour Viewpoint: unacceptable behaviour



Integration Process

```
        also par
                monitor regional traffic
        end par
end loop
go to collaborative mode
accept collaborative mode
loop
        par
                run in collaborative mode
        also par
                monitor regional traffic data
        end par
end loop
go to fixed mode
accept fixed mode
```

This represents an error c
TCC, which should not iss
to the RMS to move from
to fixed mode. There's al
RMS, which should not is
of this.

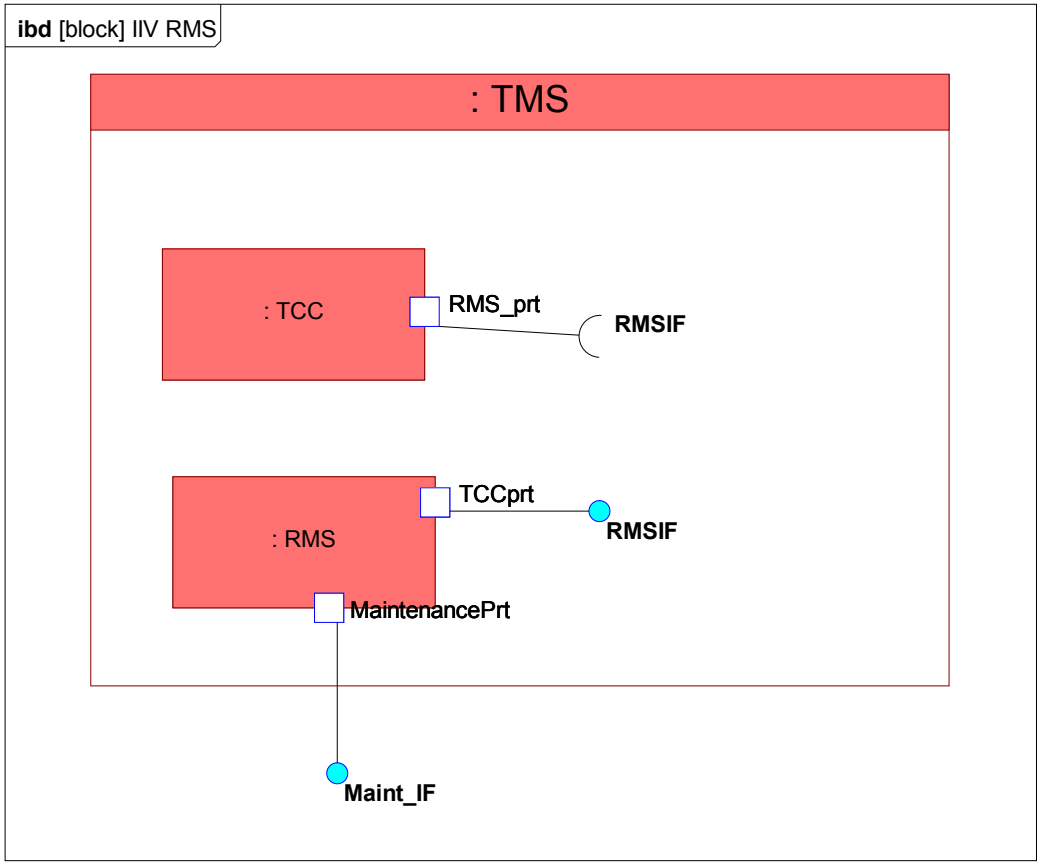
Integration Process

Interface Connectivity Viewpoint

- Defines how the interfaces in the IIV are connected

Integration Process

Interface Identification Viewpoint

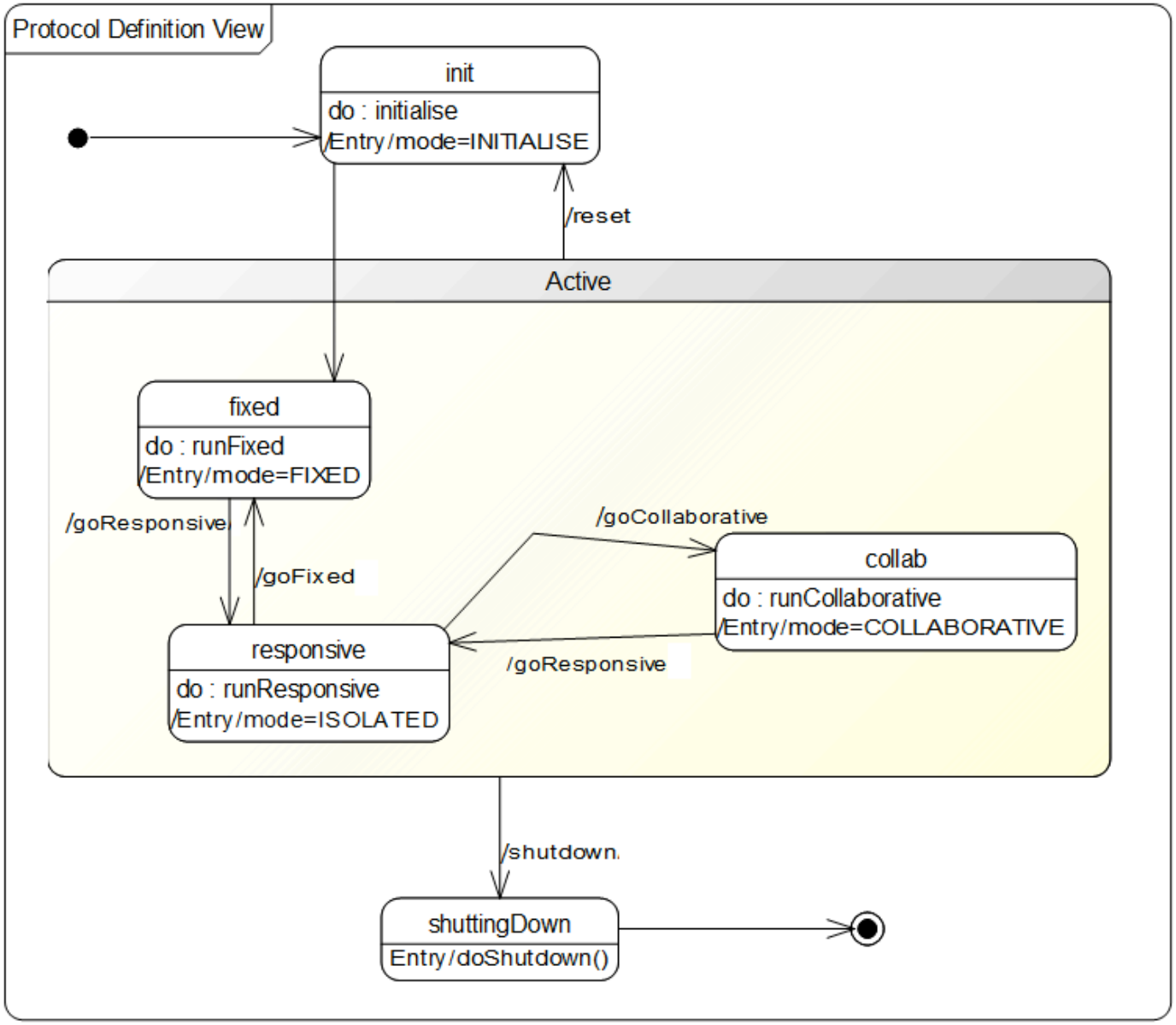


Integration Process

Protocols Definition Viewpoint

- Defines protocols for each interface identified during previous modelling efforts

Protocol Definition

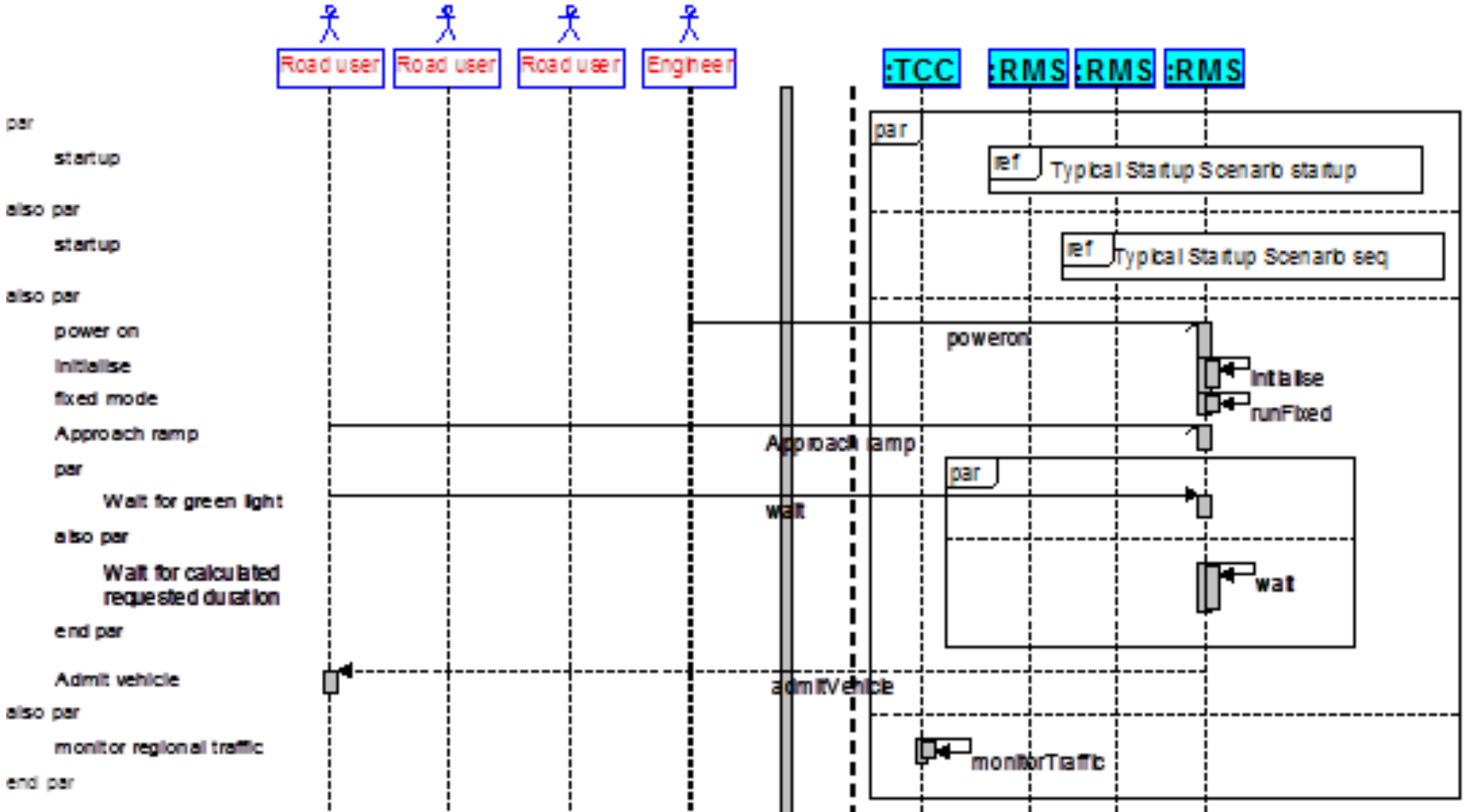


Validation

- Integration Framework also includes validation phase
- We validate our SoS-level use cases using Interface Behaviour Viewpoints (already presented)
- CIF recommends further validation, in the form of a Validation Interaction Viewpoint
  - integrates the previous sequence diagrams all onto one view
  - Identify inconsistencies or duplication

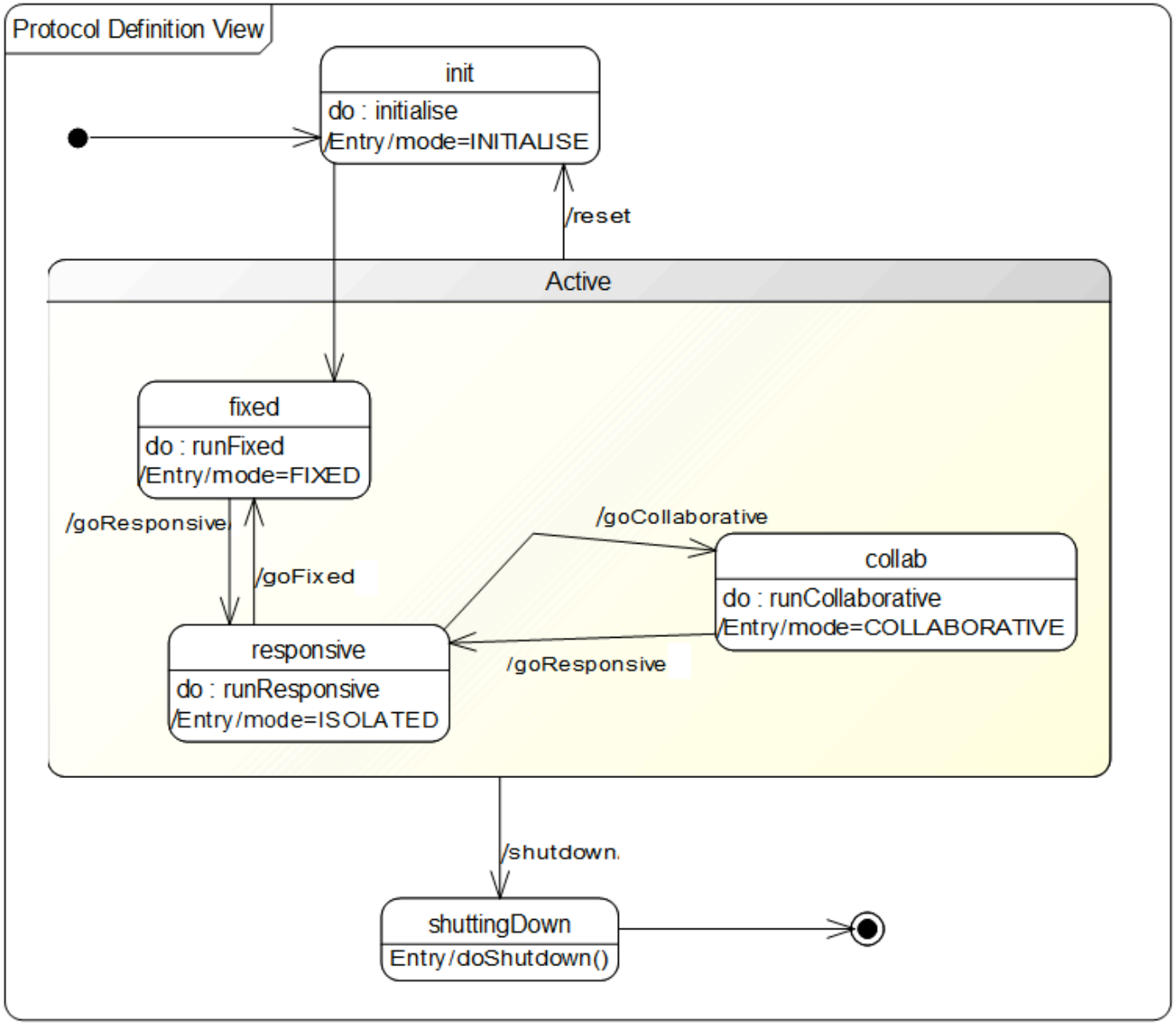
Validation

Validation Interaction Viewpoint



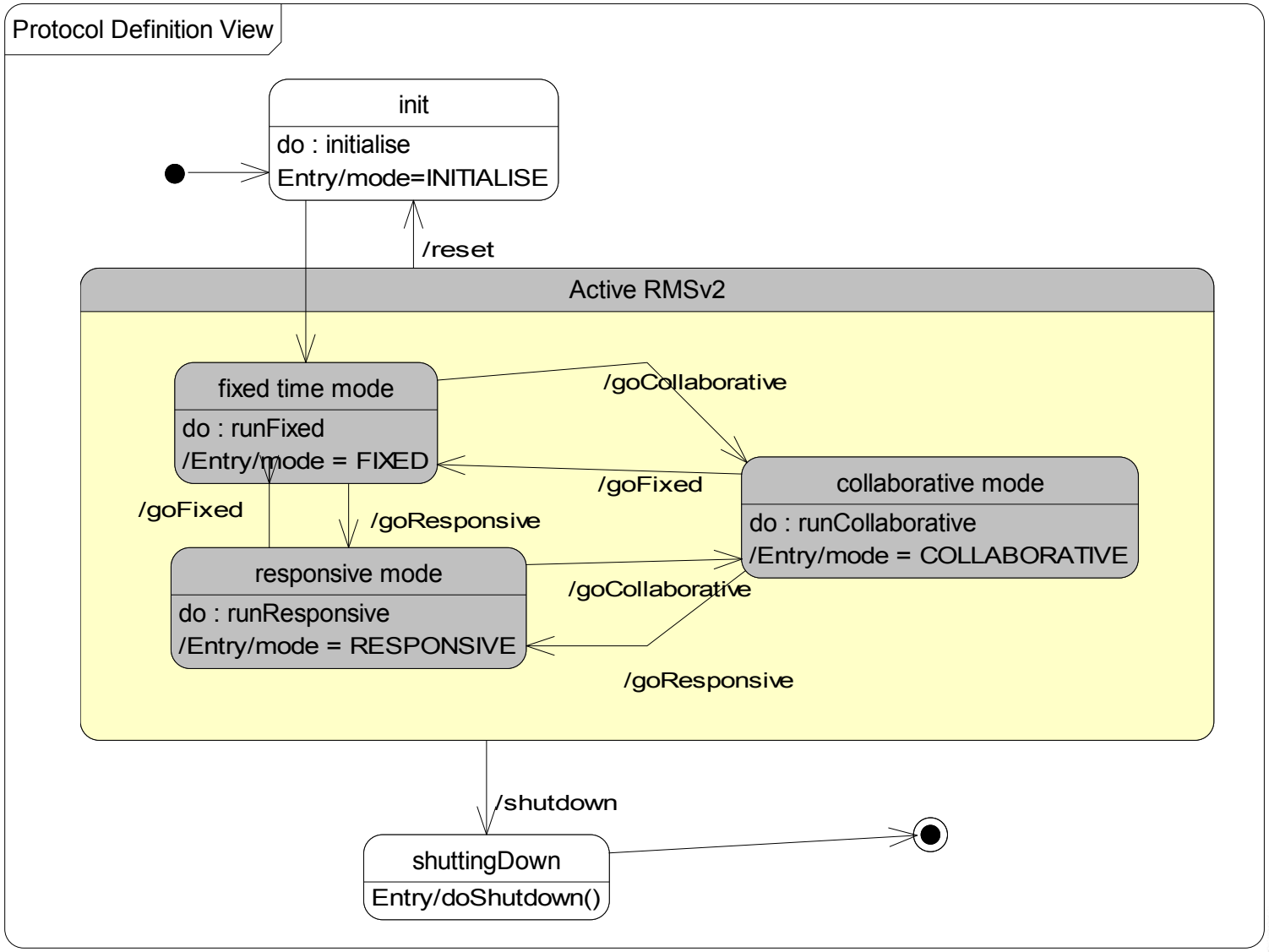
COMPASS Integration Framework

- The result is a set of models describing our existing RMS, which we call **RMSv1**
- Next: we repeat the same complete process, to generate a set of similar models for a new version of the RMS, which we call **RMSv2**
- Only one model differs for RMSv2 - Protocols Definition Viewpoint


Protocol Definition View
init
do : initialise
/Entry/mode=INITIALISE
/reset
Active
fixed
do : runFixed
/Entry/mode=FIXED
/goResponsive
/goFixed
/goCollaborative
collab
do : runCollaborative
/Entry/mode=COLLABORATIVE
responsive
do : runResponsive
/Entry/mode=ISOLATED
/goResponsive
/shutdown
shuttingDown
Entry/doShutdown()

RMSv1

RMSv2


Protocol Definition View
init
do : initialise
Entry/mode=INITIALISE
/reset
Active RMSv2
fixed time mode
do : runFixed
/Entry/mode = FIXED
/goCollaborative
/goFixed
/goFixed
/goResponsive
collaborative mode
do : runCollaborative
/Entry/mode = COLLABORATIVE
responsive mode
do : runResponsive
/Entry/mode = RESPONSIVE
/goCollaborative
/goResponsive
/shutdown
shuttingDown
Entry/doShutdown()

COMPASS Integration Framework

CIF process:

1. Architectural models, as prescribed
2. Translate to a formal modelling notation
3. Analysis and validation with automated or semi-automated support

Outline

1. Integration in a SoS
2. Introduction to the case study
3. Application of the Integration Framework
   - Constituent system identification process
   - Integration process
   - Validation process
4. **Conclusions**

Conclusions

- CIF offers a structured, guided process for modelling various integration scenarios
- SoSs have long life-spans; CIF can help to plan regression testing
- Our approach here is a good introductory step for formal modelling of the SoS
- More work to be done – including further work to integrate with automated testing and formal modelling techniques where possible

claire.ingram@ncl.ac.uk @_Claire_Ingram
john.fitzgerald@ncl.ac.uk @NclFitz

This work is part of the COMPASS project: research into model-based techniques for developing, maintaining and analysing SoSs

COMPASS
thecompassclub.org

25th anniversary annual INCOSE international symposium
Seattle, WA
July 13 - 16, 2015