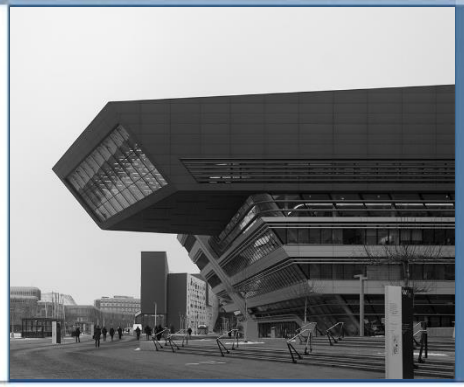
ICSOC 2017

Malaga – 14th November 2017

POLITECNICO DI MILANO

WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS



Giovanni Meroni, Claudio Di Ciccio and Jan Mendling

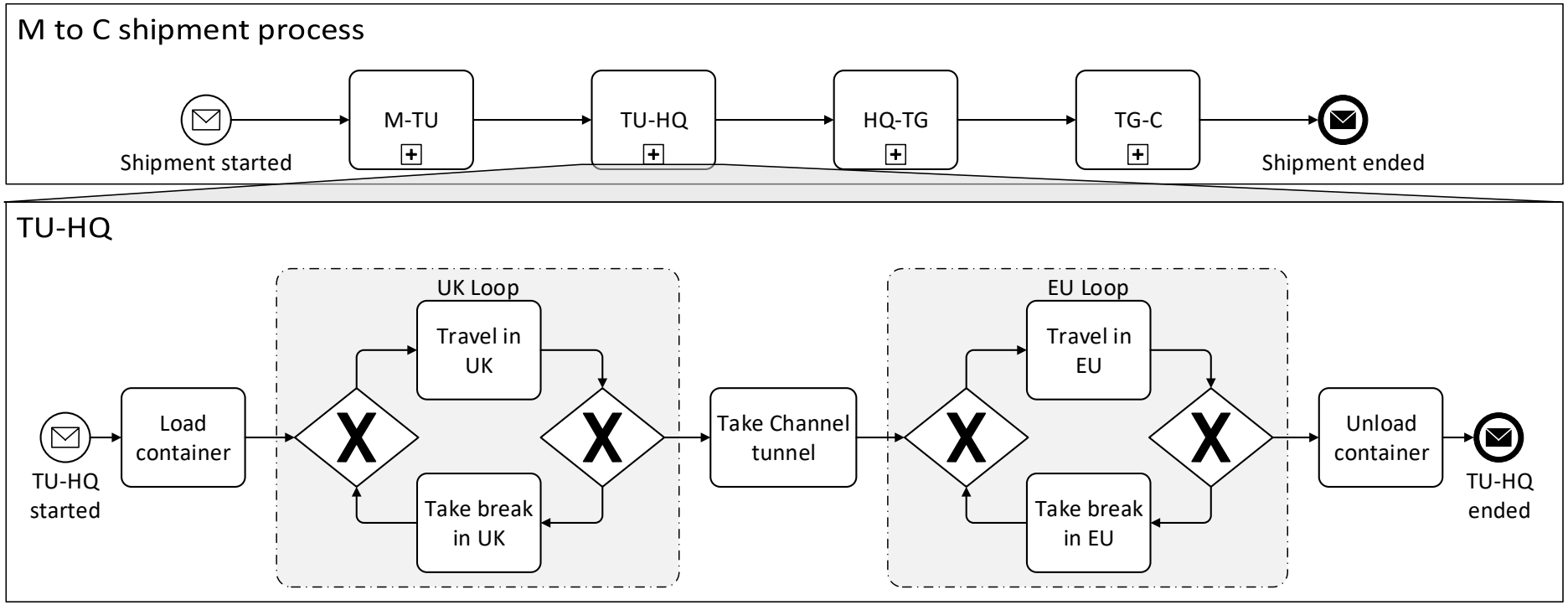
AN ARTIFACT-DRIVEN APPROACH TO MONITOR BUSINESS PROCESSES THROUGH REAL-WORLD OBJECTS

Context

- Organizations outsource internal processes to service providers
  - I.e.: freight transportation, supply chain, etc…
- Once internal processes become inter-organizational
- Stakeholders control only a portion of these processes
  - Cannot enforce the portion carried out by service providers
- Process monitoring becomes critical
  - Organizations can promptly react to violations by taking countermeasures

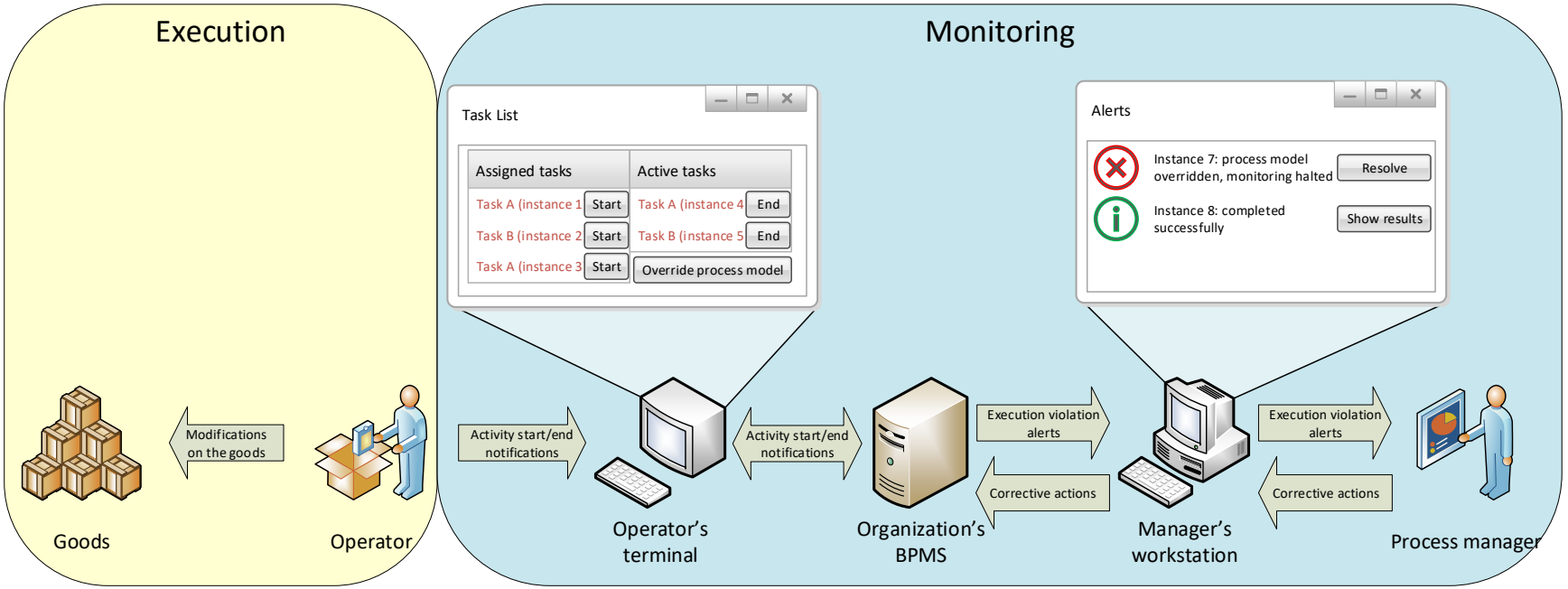
Motivating example

- Manufacturer ***M*** outsources logistics processes to logistics company ***L***
- To ship goods to customer ***C***, ***L*** organizes a four-legged shipment
  - M to TU, TU to HQ, HQ to TG, TG to C
  - Each leg carried out by a different truck shipper
  - No organization fully controls the process



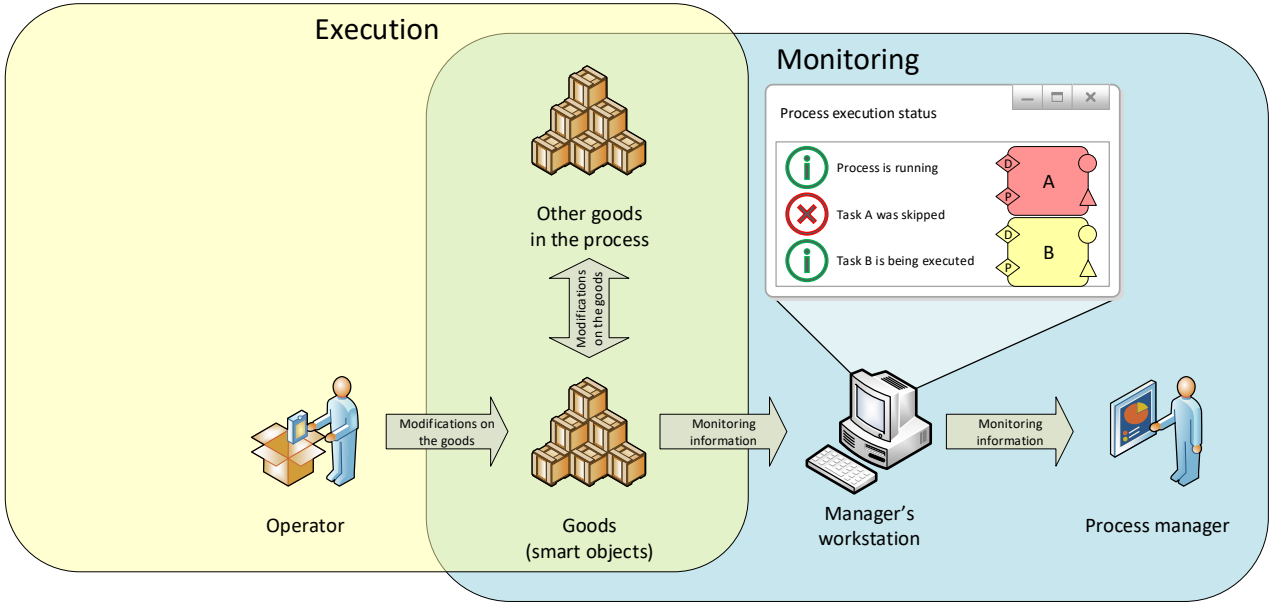
Limitations of BPMSs

- Traditional BPMSs require human intervention when monitoring multi-party processes
  - The BPMS expects explicit notification when activities start or end
  - When not automated, notifications must be sent manually
  - When a violation in the execution occurs, the BPMS stops until the issue is solved



Introducing Artifact-driven monitoring

- Idea: rely on physical objects participating in the process
  - Physical objects have visibility on the whole process
  - The state of the physical objects determines the activation and termination of activities
- Thank to the IoT, physical objects become “smart”:
  - They can autonomously infer their state
  - A monitoring platform can be run on top of them

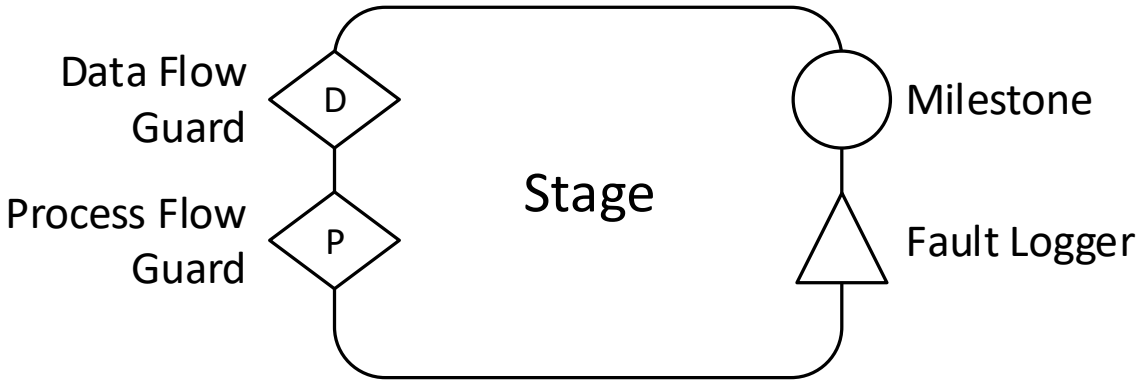


E-GSM to model monitored processes

- Imperative process modeling languages not suited for autonomous monitoring
  - Dependencies among activities are prescriptive
  - When dependencies are violated, human intervention is required
- E-GSM (Extended-GSM) language is more flexible
  - Extension of Guard-Stage-Milestone declarative language
  - Dependencies are descriptive
  - Can deal with violations, detecting which activities are affected

Extended-GSM overview

- **Stages** represent activities and process portions
  - Stages can be nested
  - Atomic stages represent atomic tasks
- **Data Flow Guards** determine stage activation
- **Process Flow Guards** define stage dependencies
  - Evaluated when data flow guards are triggered, before the stage is active
  - If not fulfilled, the stage is flagged as not respecting the model
- **Milestones** determine stage termination



Binding smart objects to processes

- Identity of Smart Objects often known after the process started
- Smart Objects may participate to a portion of the process.
  - Information exchanged before/after that portion would be useless or misleading for the monitoring
- Rules to dynamically bind smart objects to the process are needed

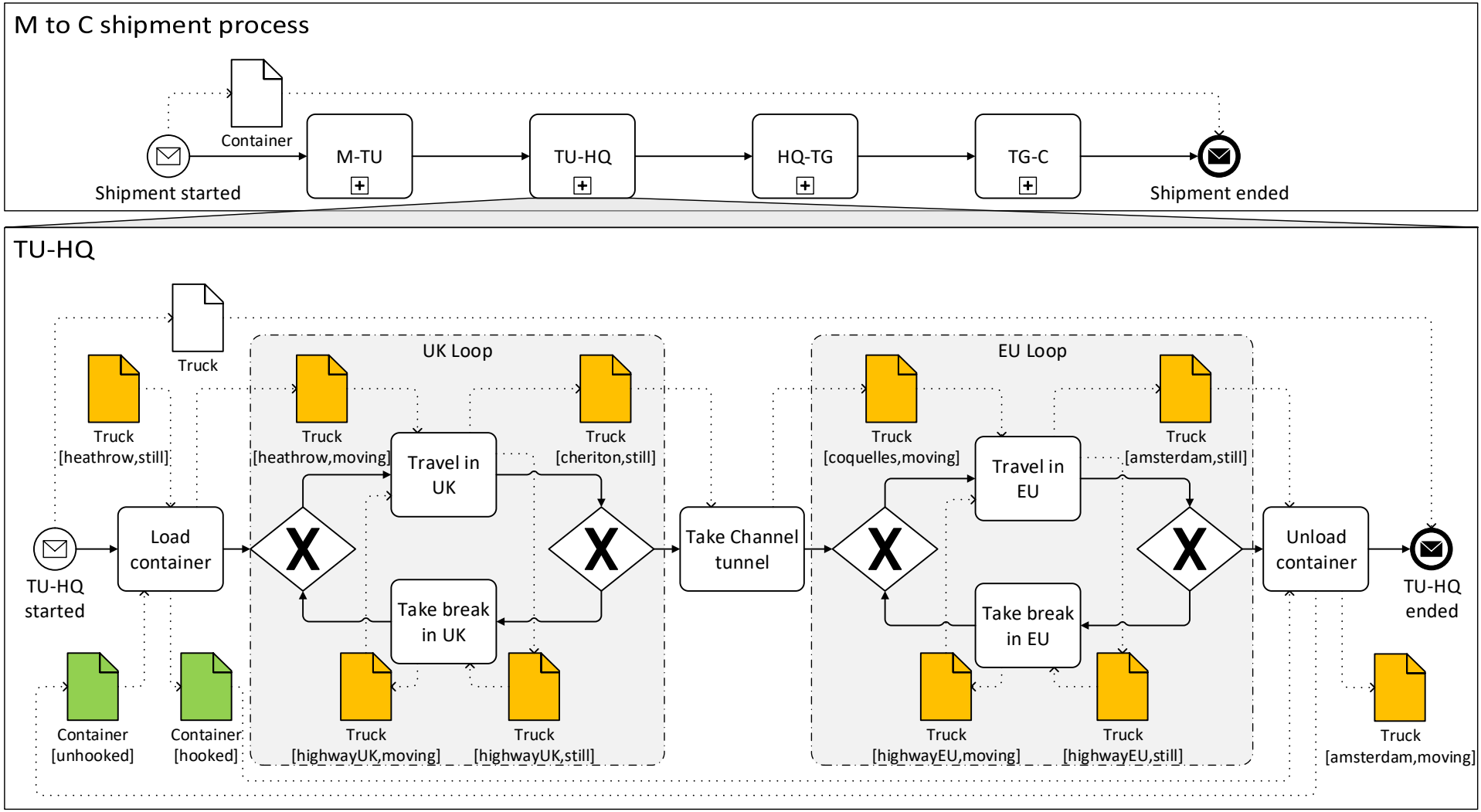
Adopting artifact-driven monitoring

- Adopting artifact-driven monitoring can be difficult
  - Process must be redesigned in E-GSM
  - E-GSM less intuitive than BPMN
  - Identity of smart objects needed
  - Binding rules (smart objects ←→ process instance) needed
- Solution: guided approach starting from BPMN
  - Step 1: BPMN model enriched with objects
  - Step 2: enriched BPMN translated to E-GSM
  - Step 3: binding rules derived from enriched BPMN

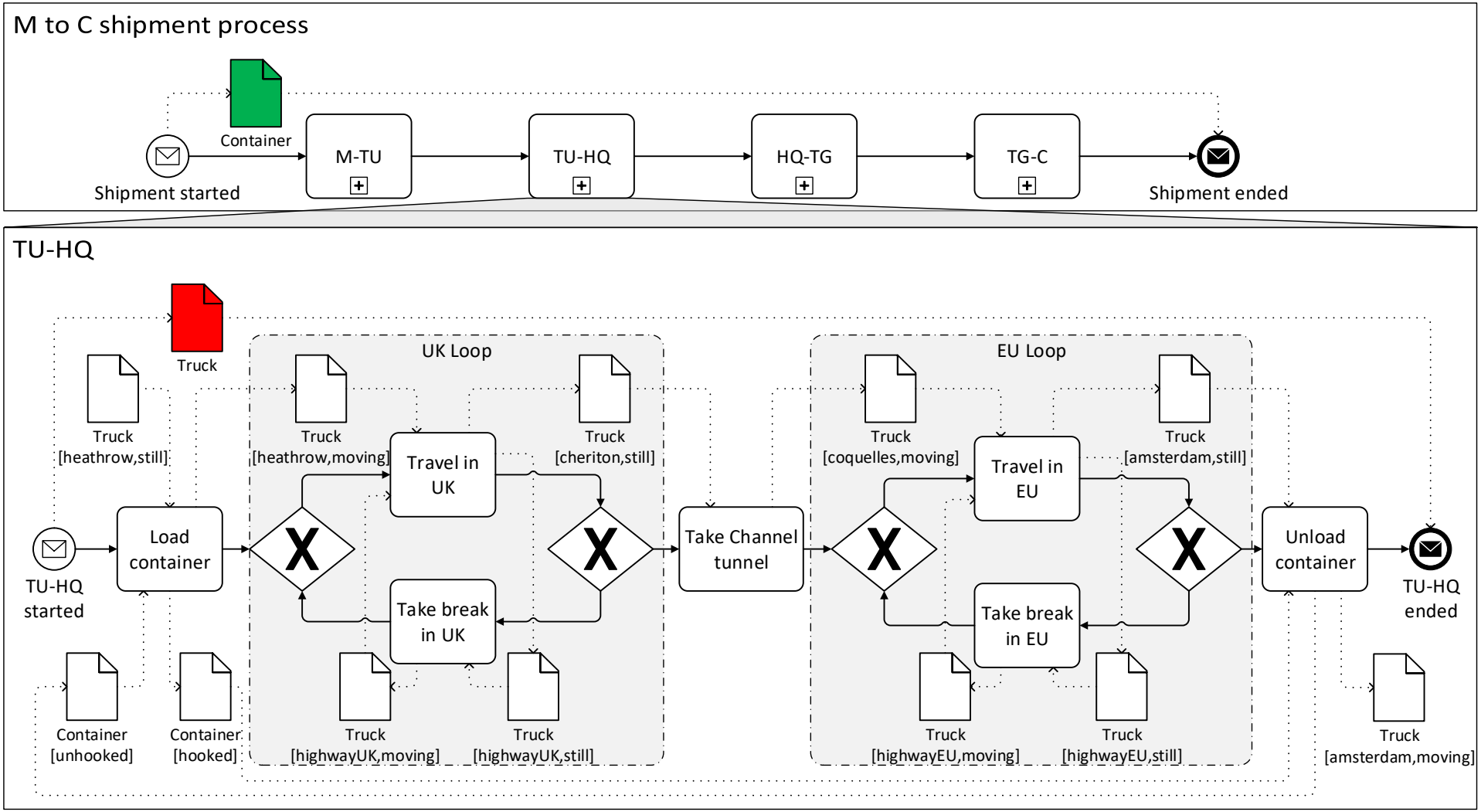
Step 1 – Enriching BPMN with objects

- Data Objects adopted to represent smart objects
  - Data state indicates smart objects' conditions
- Data Objects connected to activities indicate when the activities are executed
  - Activity starts when all input data objects have the indicated state
  - Activity ends when all output data objects have the indicated state
- Data Objects connected to events indicate when the smart objects interact with the process
  - Start event: smart object starts interaction
  - End event: smart object stops interaction

Step 1 – Enriching BPMN with objects



Step 1 – Enriching BPMN with objects

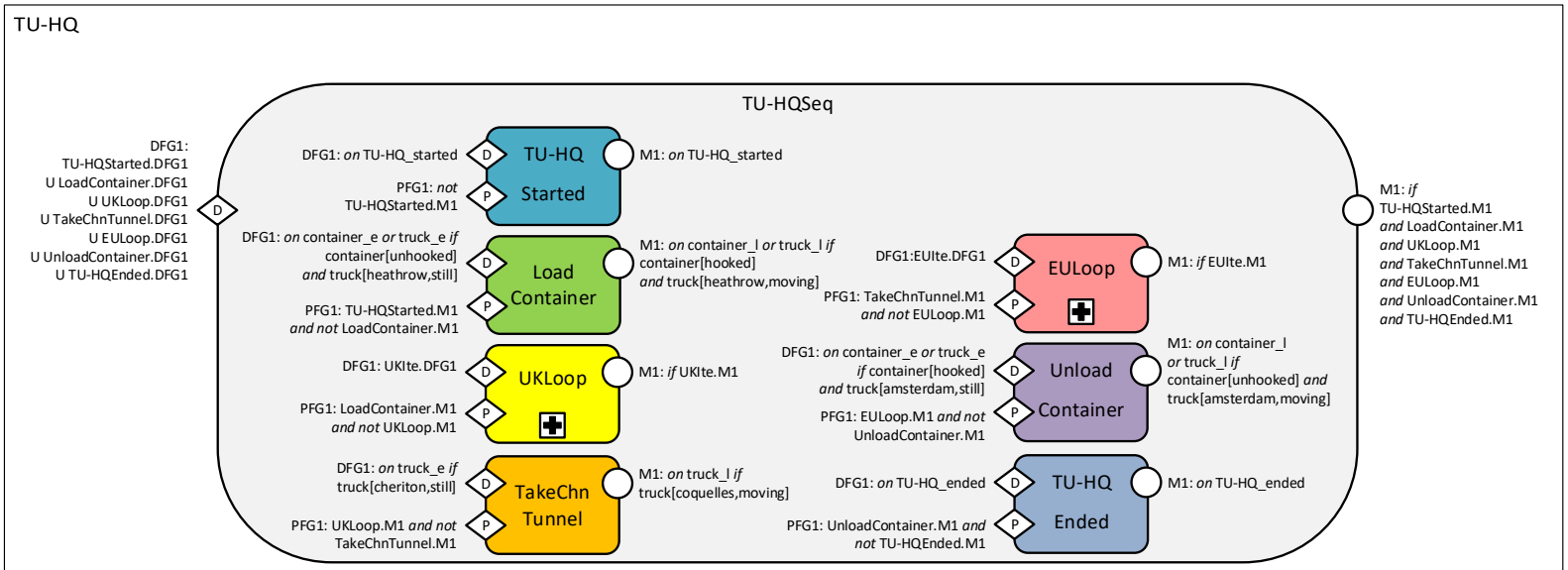


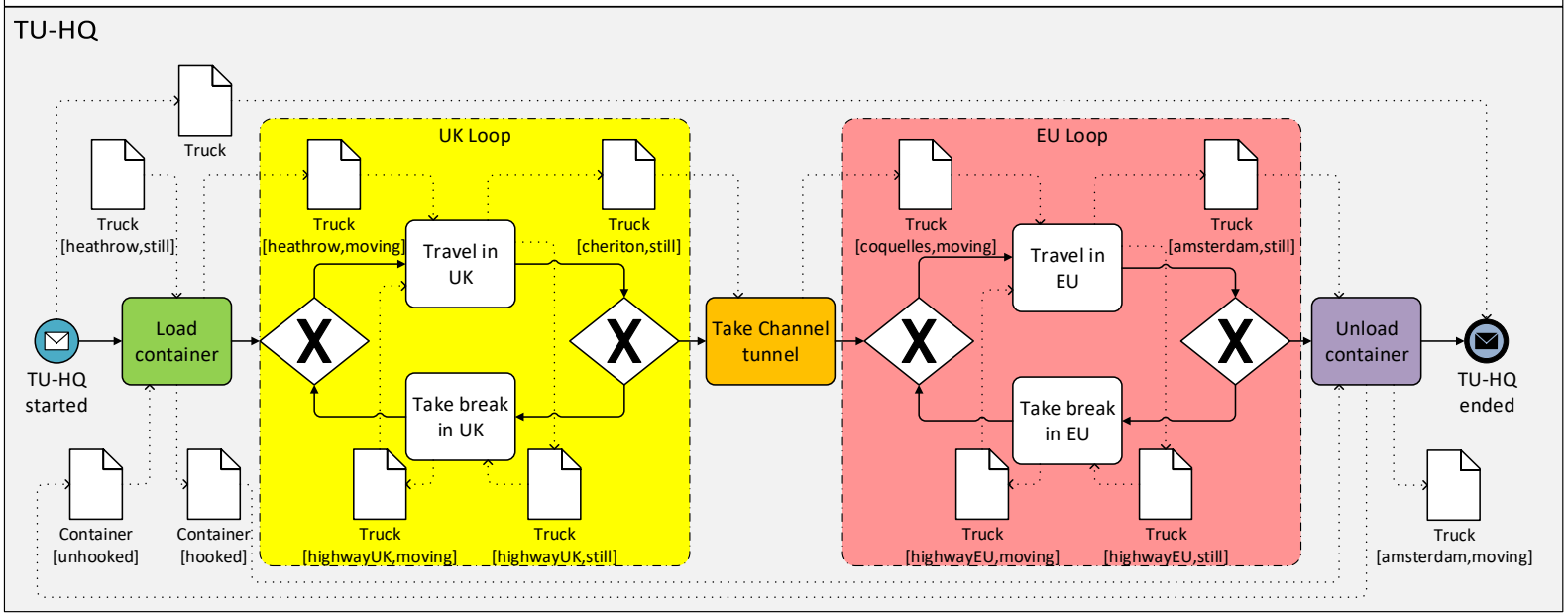
Step 2 – Generating the E-GSM model

- Following [1], BPMN process translated into E-GSM:
  - Process decomposed into nested blocks
  - Atomic activities and events translated to atomic stages
  - Blocks translated to nested stages embedding inner blocks
  - Process flow guards reflect the dependencies outlined in the BPMN model

[1] L. Baresi, G. Meroni, P. Plebani: Using the Guard-Stage-Milestone Notation for Monitoring BPMN-based Processes, BPMDS 2016 Proceedings

Step 2 – Generating the E-GSM model

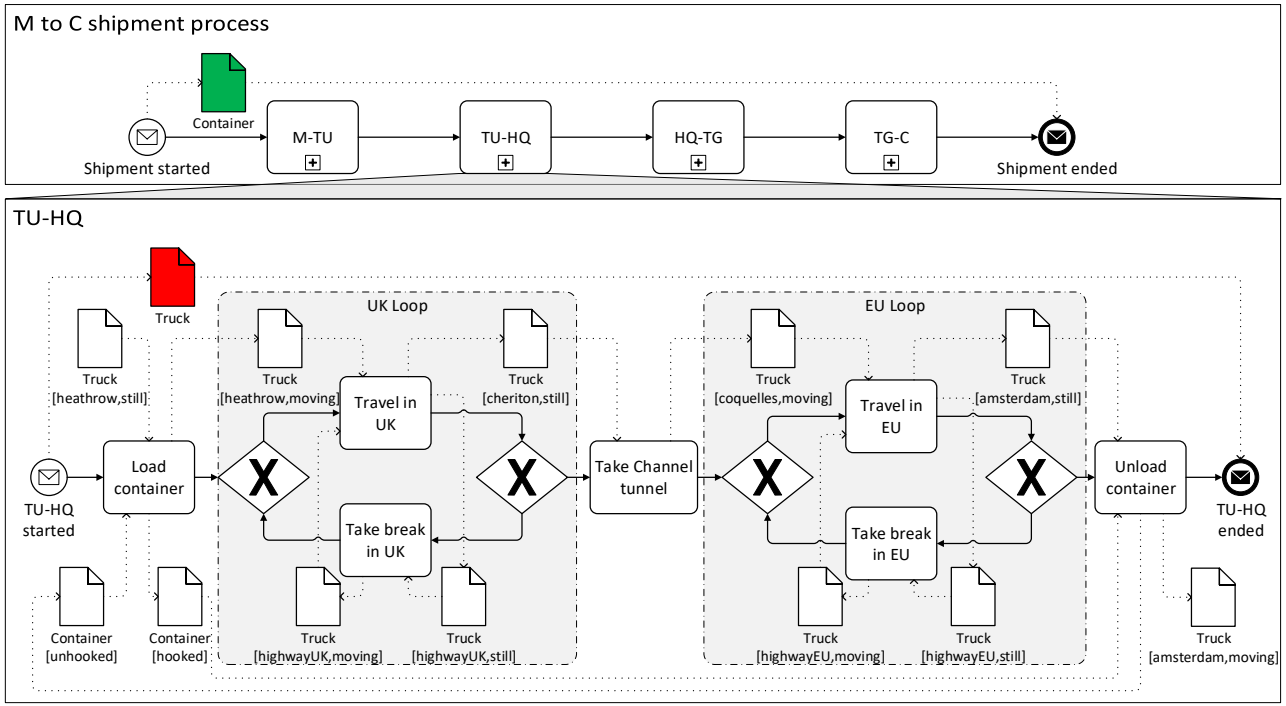




Step 3 – Deriving mapping criteria

- The E-GSM model predicates on the state of ***generic*** smart objects (e.g., a truck)
- When the monitoring starts, they are replaced with ***specific*** smart objects (e.g., the truck with plate AB123XY)
- Criteria to map specific smart objects to generic ones derived from enriched BPMN model
  - Data objects connected to start events produce binding criteria
  - Data objects connected to end events produce unbinding criteria

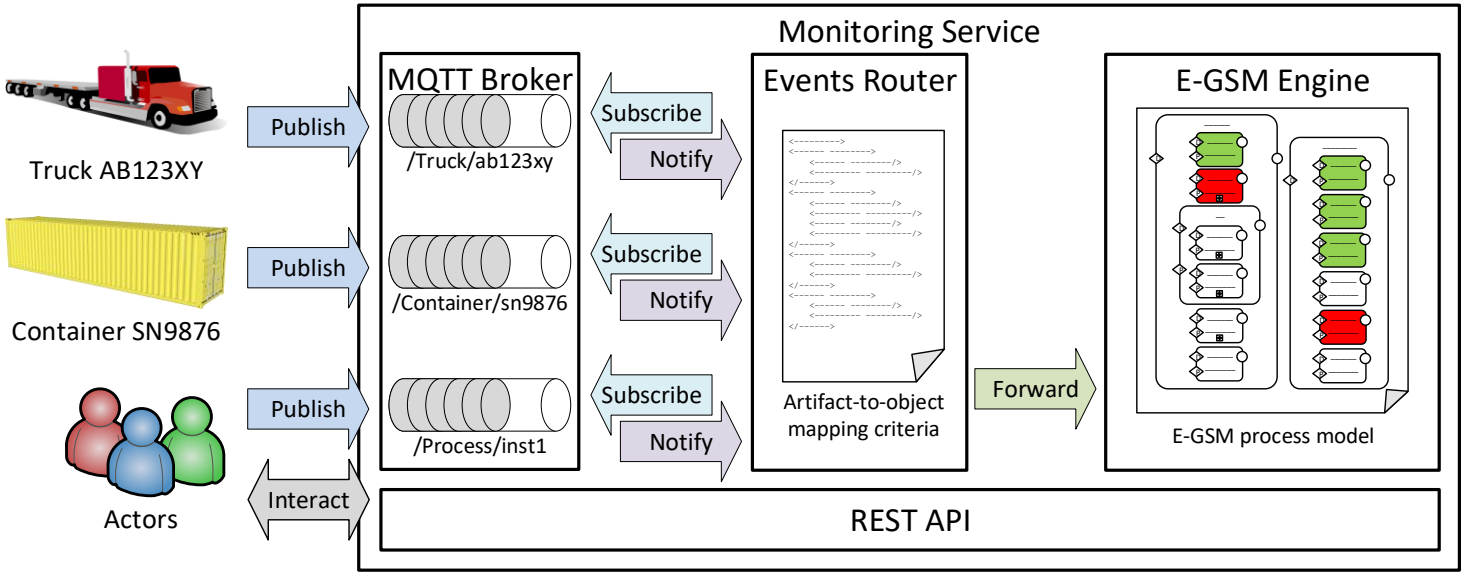
Step 3 – Deriving mapping criteria




```
<Mapping>
  <Artifact name="Container">
    <BindingEvent id="shipment_started"/><UnbindingEvent id="shipment_ended"/>
  </Artifact>
  <Artifact name="Truck">
    <BindingEvent id="tu_hq_started"/><UnbindingEvent id="tu_hq_ended"/>
  </Artifact>
</Mapping>
```

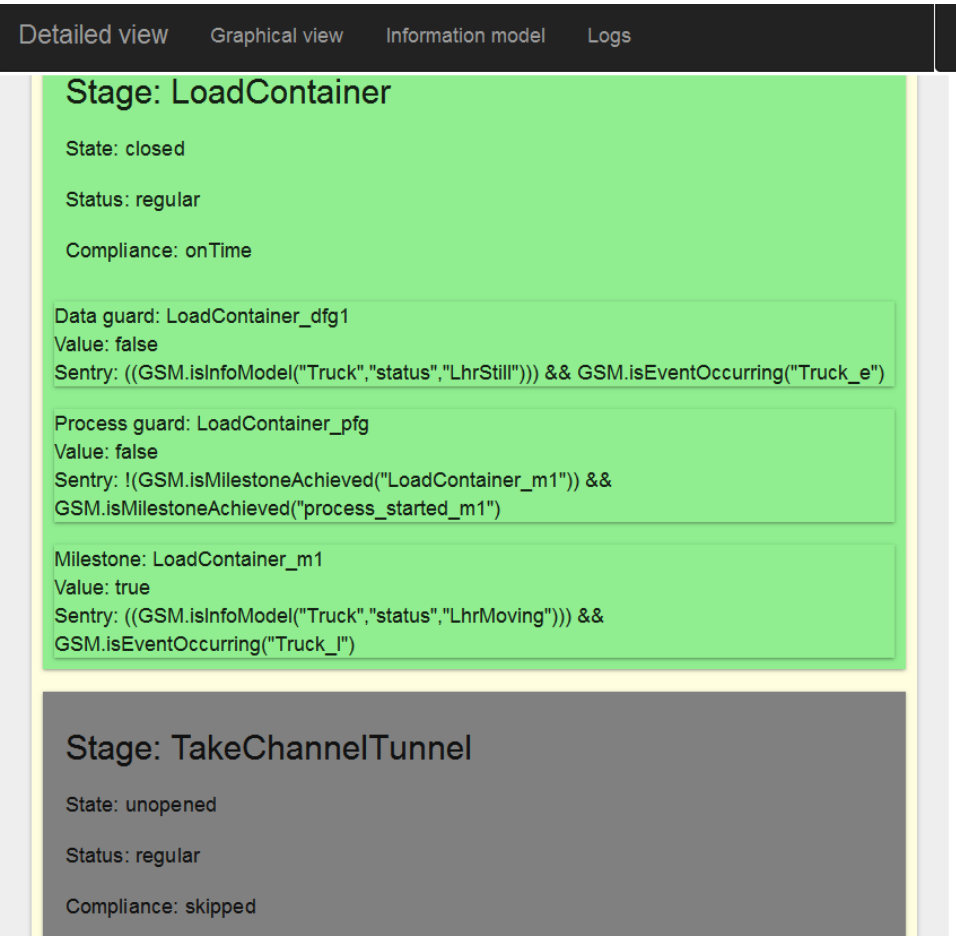
Validating artifact-driven monitoring

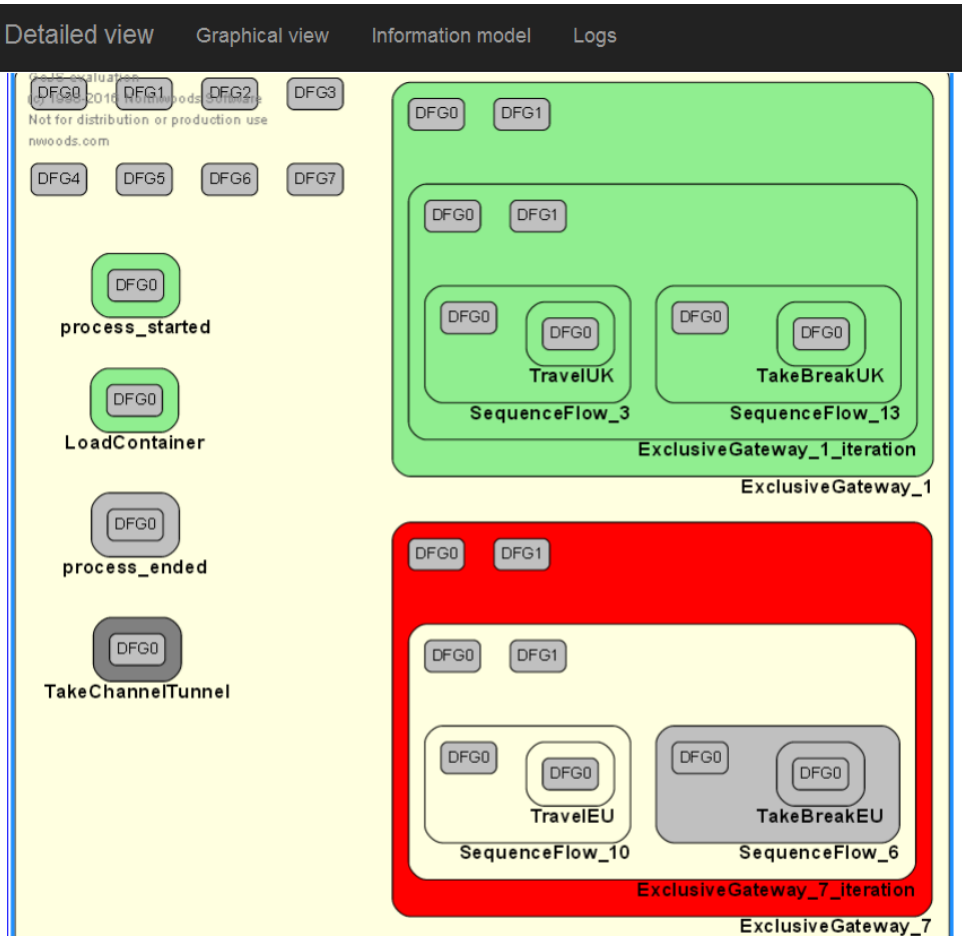
- To prove the feasibility of artifact-driven monitoring, the SMARTifact platform was developed [2]
  - Configurable with REST API
  - Smart objects communicate to the platform with MQTT
  - E-GSM engine monitors the process
  - Event router forwards to E-GSM engine only relevant information



[2] L. Baresi, C. Di Ciccio, J. Mendling, G. Meroni, P. Plebani: mArtifact – an Artifact-driven Process Monitoring Platform, BPM Demo Track 2017 Proceedings

Validating artifact-driven monitoring





Validating artifact-driven monitoring

- Eight shipment processes provided by a large European logistics company
  - Amsterdam to London, Bruxelles, Paris, Frankfurt and vice-versa
- Two datasets related to 77 shipments
  - Dataset 1: position and speed of trucks (19966 entries)
  - Dataset 2: activation and termination of activities in shipment processes, manually notified by truck drivers (815 entries)
- Dataset 1 was replayed on mArtifact
- The results of the monitoring were compared with Dataset 2
  - Over 93% of the shipments were correctly monitored
  - mArtifact detected more activities than manual notifications
  - The median detection delay was less than 5 minutes, while the processes lasted on average 533 minutes

Conclusions

- Artifact-driven process monitoring can effectively monitor inter-organizational processes
  - IoT smart objects to detect execution of activities
  - Manual notifications no longer required
  - Violations detected at runtime
  - Continuous monitoring
- Guided approach to configure artifact-driven platform
  - BPMN process as starting point
  - E-GSM and binding criteria derived

Thanks for your attention

Full article here:

THIS WORK HAS BEEN PARTIALLY FUNDED BY THE ITALIAN PROJECT ITS2020 UNDER THE TECHNOLOGICAL NATIONAL CLUSTERS PROGRAM

TRASPORTI ITALIA 2020

UNIONE EUROPEA
Fondo europeo di sviluppo regionale

PON Ricerca e Competitività 2007-2013

Ministero dell'Istruzione, dell'Università e della Ricerca