BPM Forum 2017

Barcelona – 14th September 2017

POLITECNICO DI MILANO

Giovanni Meroni, Pierluigi Plebani

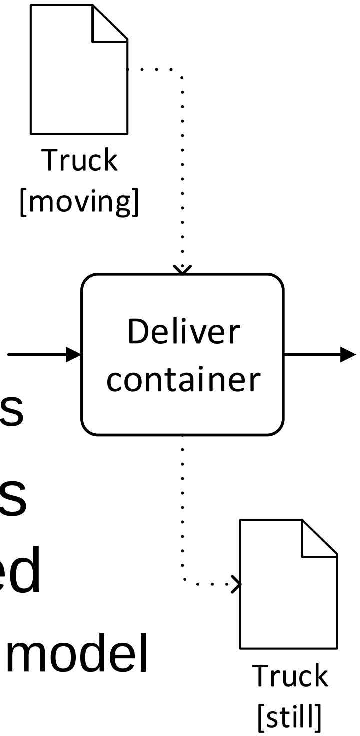
ARTIFACT-DRIVEN MONITORING FOR HUMAN-CENTRIC BUSINESS PROCESSES WITH SMART DEVICES: ASSESSMENT AND IMPROVEMENT

Context and problem statement

- Most business processes involve non-automated activities
  - I.e.: freight transportation, supply chain, etc…
- Human operators are required to execute these activities
- Traditional BPMSs rely on explicit notifications to detect when these activities are performed
  - They present to the operator a list of activities assigned to him
  - The operator has to indicate which activities are started and when they are finished
- Sending notifications is time consuming and prone to mistakes
  - The operator has to interrupt his work to interact with the BPMS
  - The operator may forget to send notifications
  - The operator may erroneously or intentionally send notifications on activities not being performed

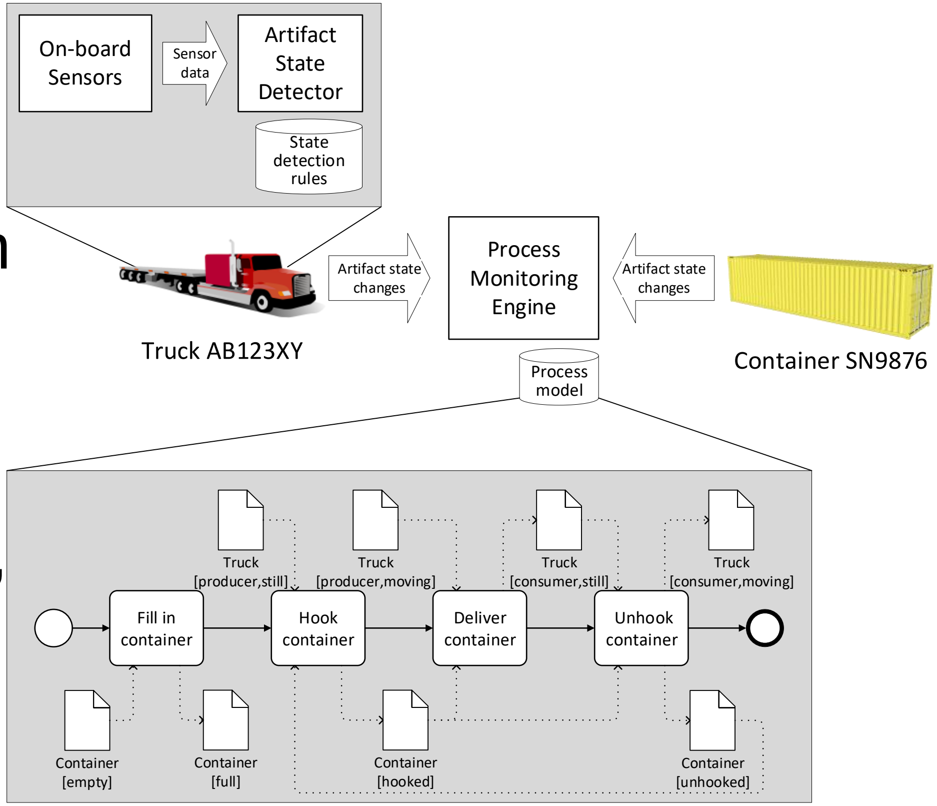
Artifact-driven process monitoring

- Instead of relying on notifications sent by operators, we rely on the physical artifacts participating to the process
- Artifacts represent the objects that interact with activities when the process is executed
  - Activities require some artifacts to have a specific state (i.e., certain characteristics) to be executed
  - Activities may alter the state of one or more artifacts
- Based on the state of the physical objects, it is possible to know when activities are performed
  - Artifacts and states are represented in the process model with data objects
  - An activity can only start when the artifacts assume the indicated state
  - An activity terminates when artifacts change their state



Exploiting the IoT to monitor processes

- Correctly determining the state of an artifact is paramount for the monitoring to be accurate
- The Internet of Things can equip physical objects with sensors, computing devices and communication interfaces, making them **smart**
- Smart objects can then autonomously infer their own state and forward it to the monitoring platform

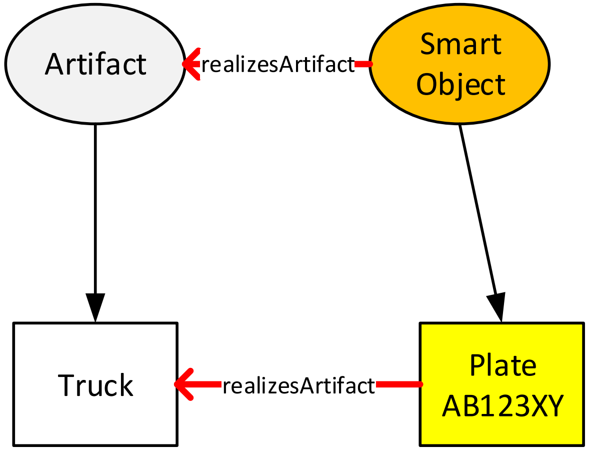


Process monitorability

- The **monitorability** of a process indicates how many activities in a process can be monitored by smart objects
- The monitorability depends on the capabilities of the smart objects embodying the artifacts:
  - Smart objects may lack sensors to determine one state
  - Smart objects may lack rules to derive one state from sensor data
  - If one state cannot be determined, activities that require or produce that state cannot be monitored
- We propose an ontology-based approach to:
  - Formalize the capabilities of smart objects
  - Estimate the monitorability
  - Provide suggestions to improve the monitorability

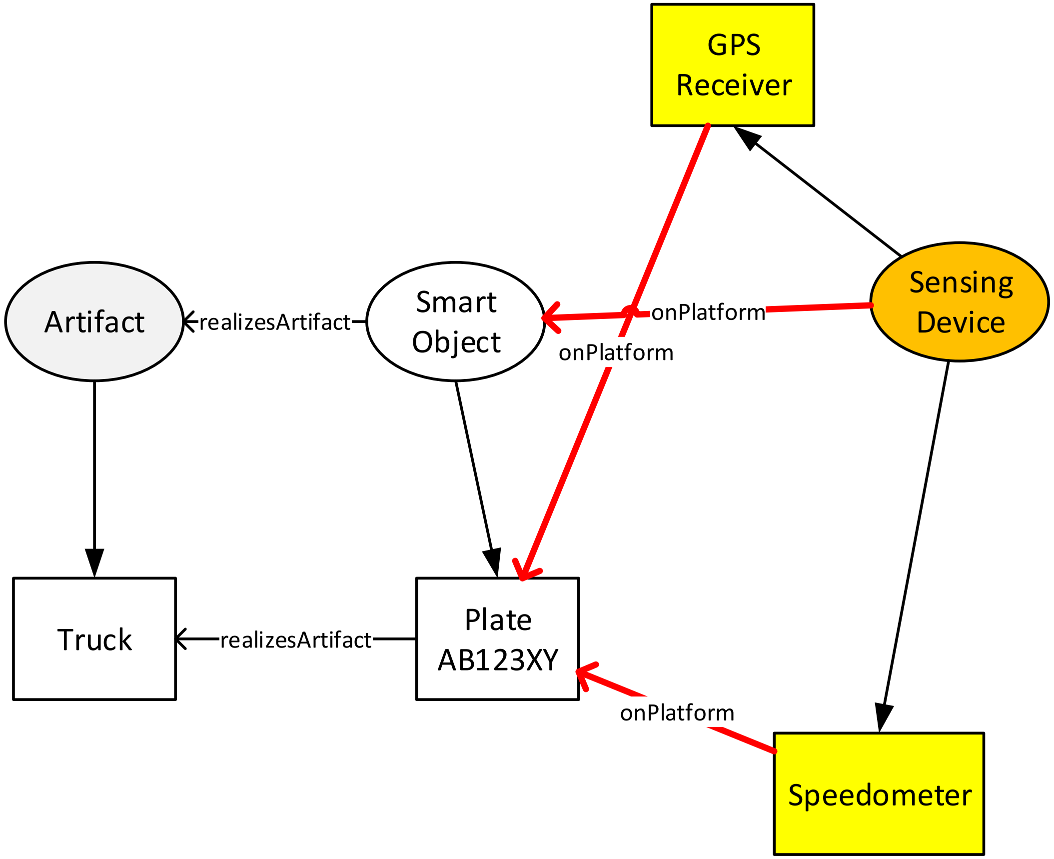
Smart object ontology

- Ontology derived from FIESTA-IoT that captures the capabilities of the smart object



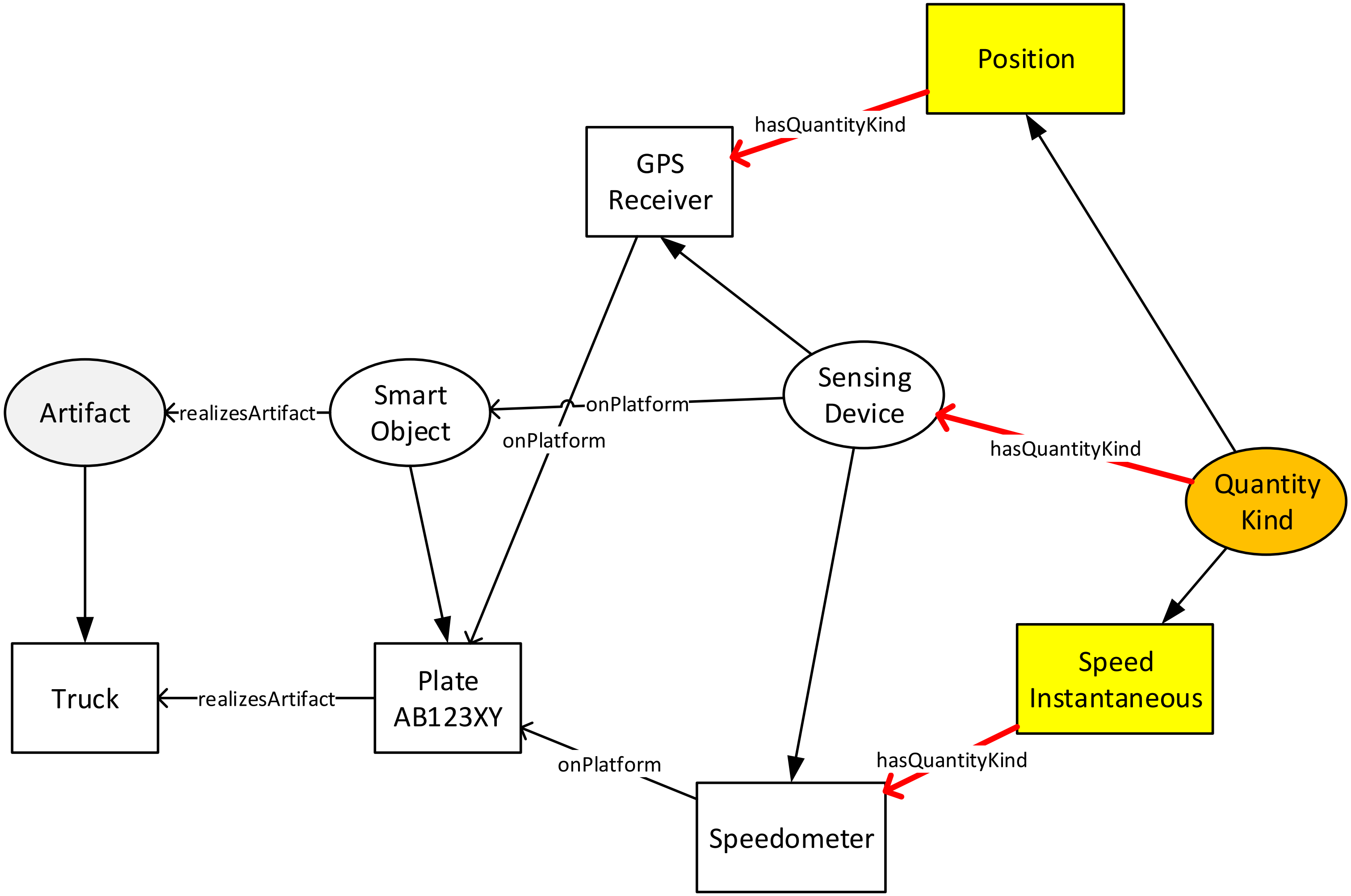
Smart object ontology

- Ontology derived from FIESTA-IoT that captures the capabilities of the smart object



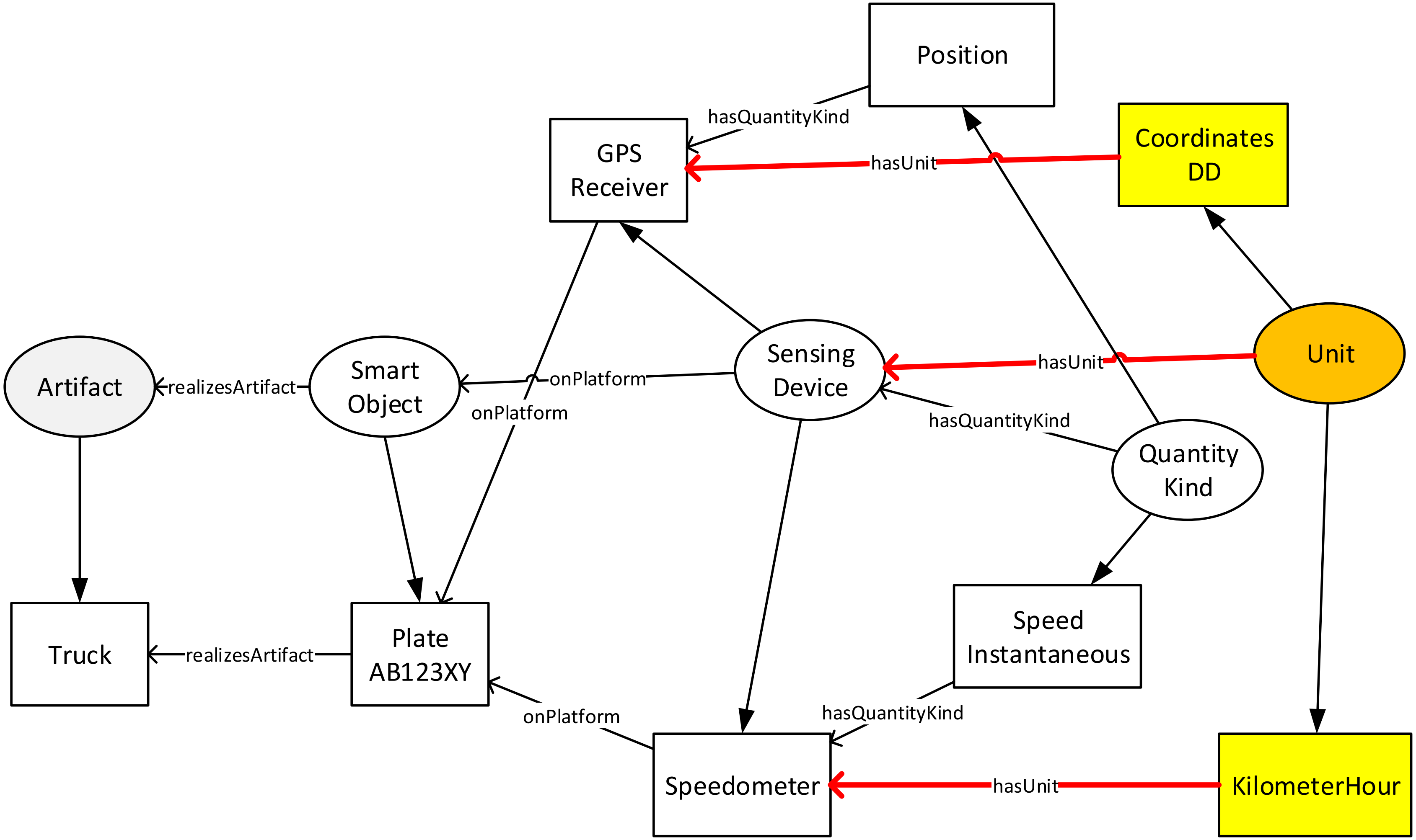
Smart object ontology

- Ontology derived from FIESTA-IoT that captures the capabilities of the smart object



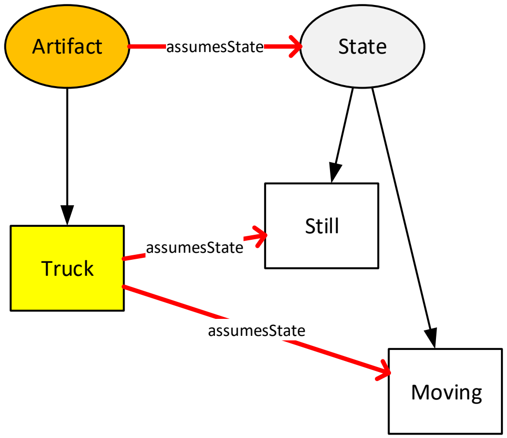
Smart object ontology

- Ontology derived from FIESTA-IoT that captures the capabilities of the smart object



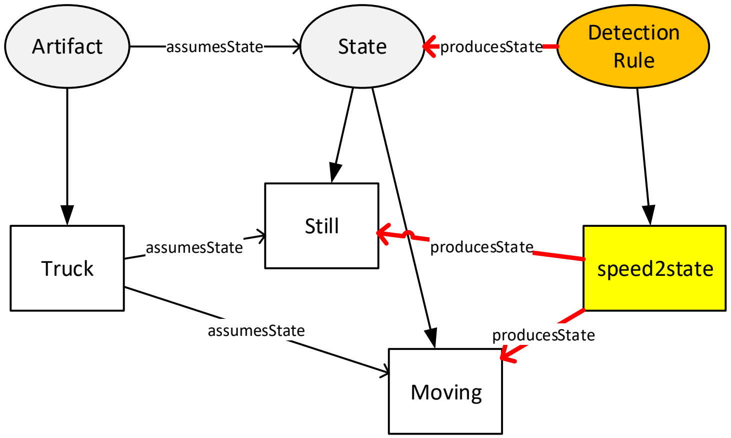
State detection rules ontology

- Ontology derived from Physics Domain ontology (Hachem et al. – MDS 2011) that formalizes how sensor data is used to infer a state



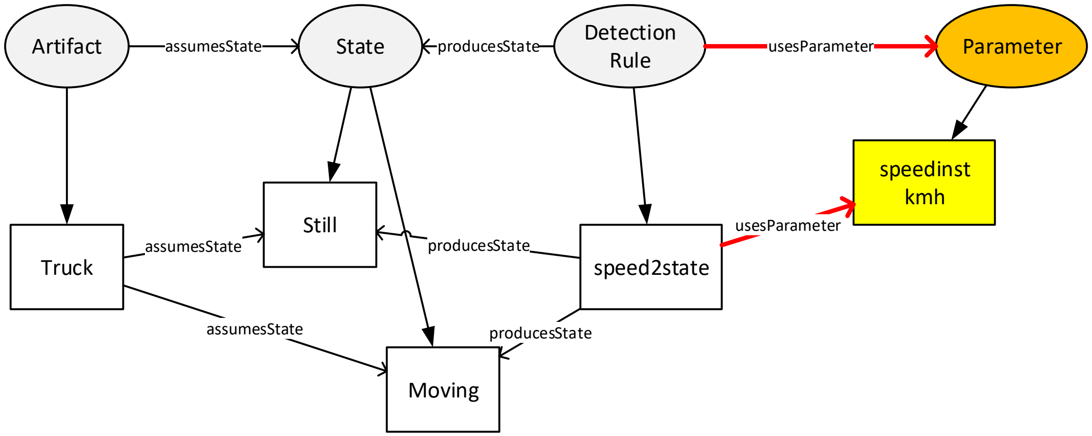
State detection rules ontology

- Ontology derived from Physics Domain ontology (Hachem et al. – MDS 2011) that formalizes how sensor data is used to infer a state



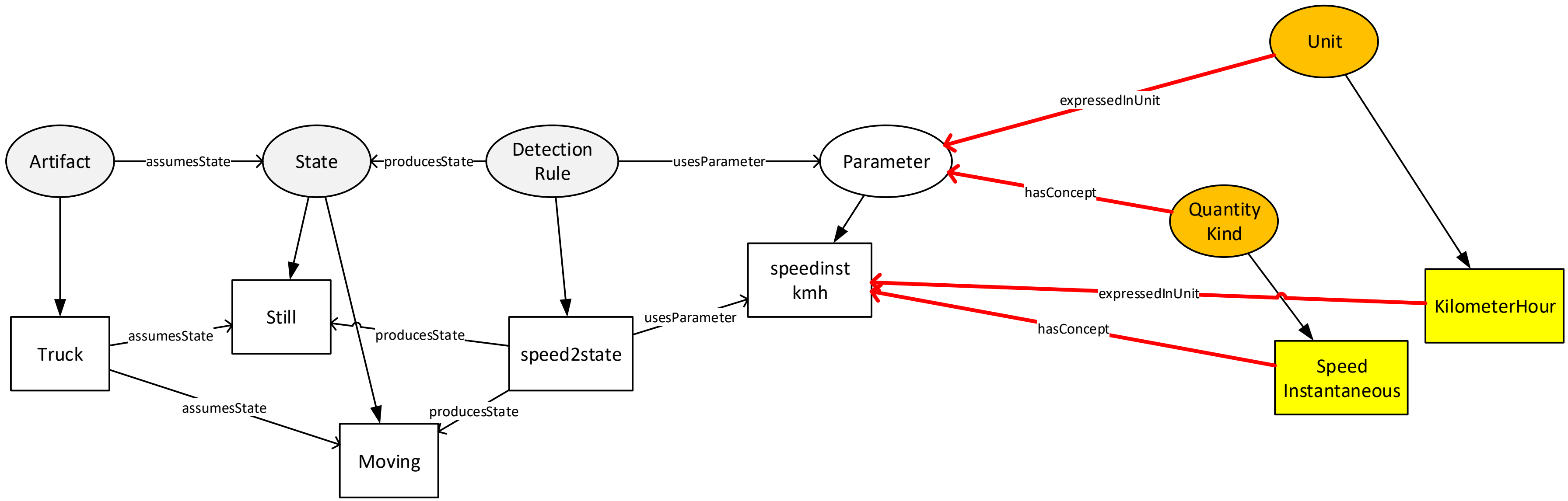
State detection rules ontology

- Ontology derived from Physics Domain ontology (Hachem et al. – MDS 2011) that formalizes how sensor data is used to infer a state



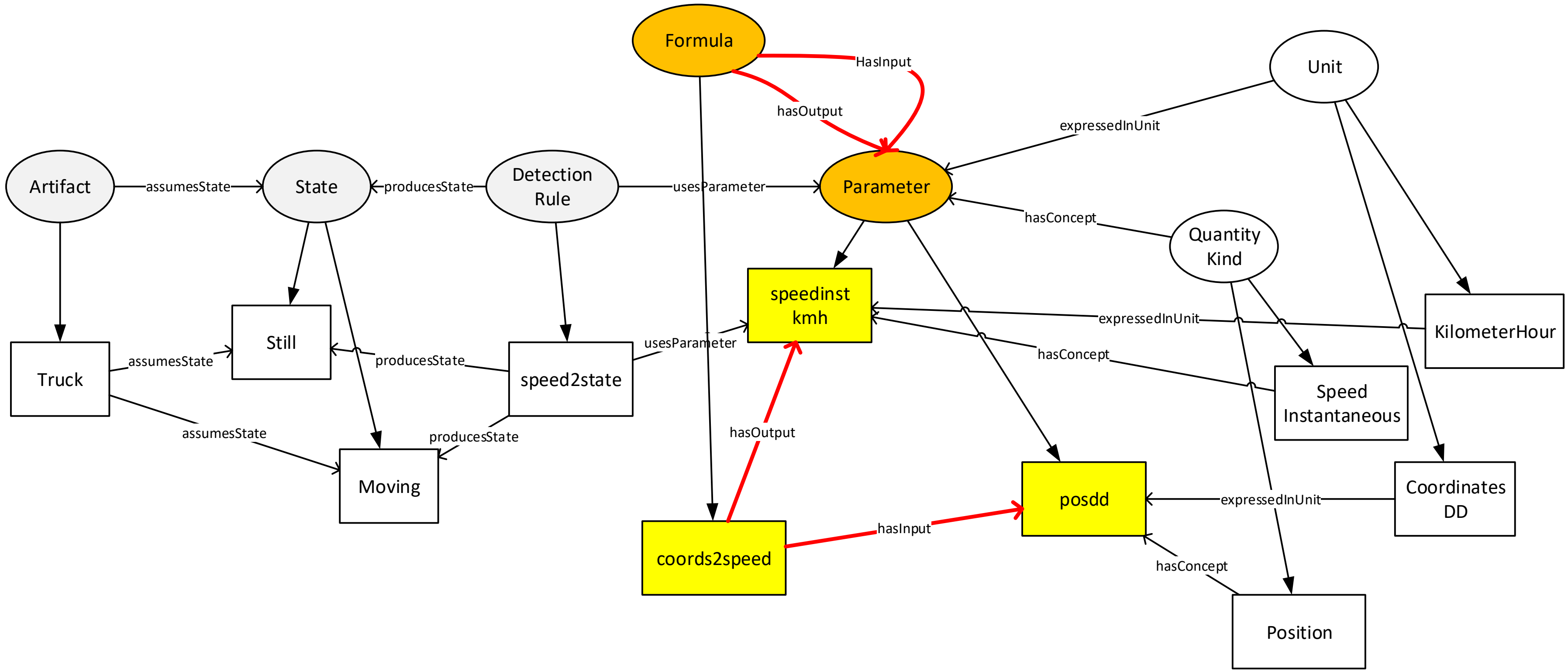
State detection rules ontology

- Ontology derived from Physics Domain ontology (Hachem et al. – MDS 2011) that formalizes how sensor data is used to infer a state



State detection rules ontology

- Ontology derived from Physics Domain ontology (Hachem et al. – MDS 2011) that formalizes how sensor data is used to infer a state



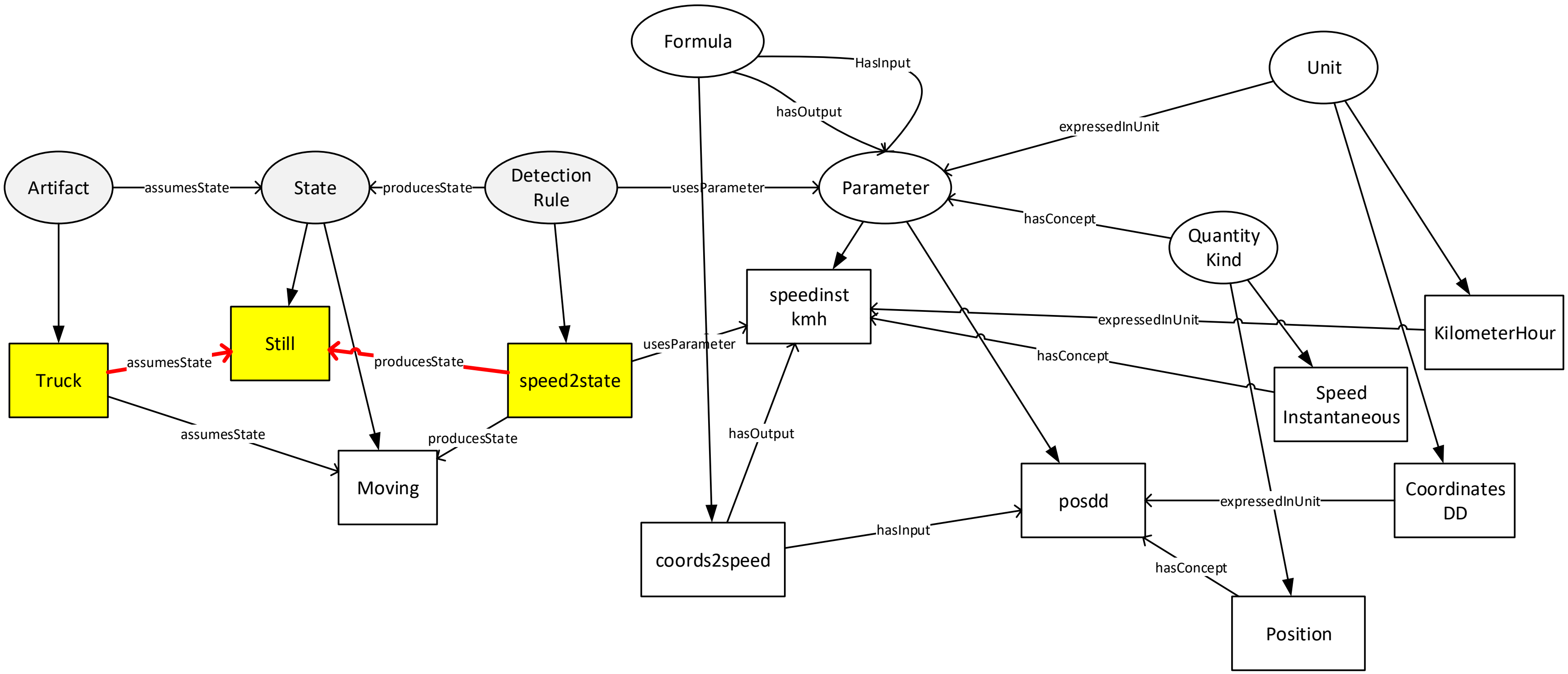
Process monitorability assessment

- For each couple **<artifact, state>** in the process model, we need to determine how many smart objects $\overline{SSO}$ can infer that state based on their capabilities
- To do so, for each smart object SSO that embodies **artifact**, the ontologies are queried to determine:
  - If a detection rule to infer **state** exists
  - Which parameters are required by that rule
  - If the sensors on the smart object provide the required parameters
- Then, the monitorability of **<artifact, state>** is computed as:

$$Mon^{ARS}(\langle artifact, state\rangle, I) \rightarrow [0,1] = \left|\overline{SSO}\right| \ / \ |SSO|$$

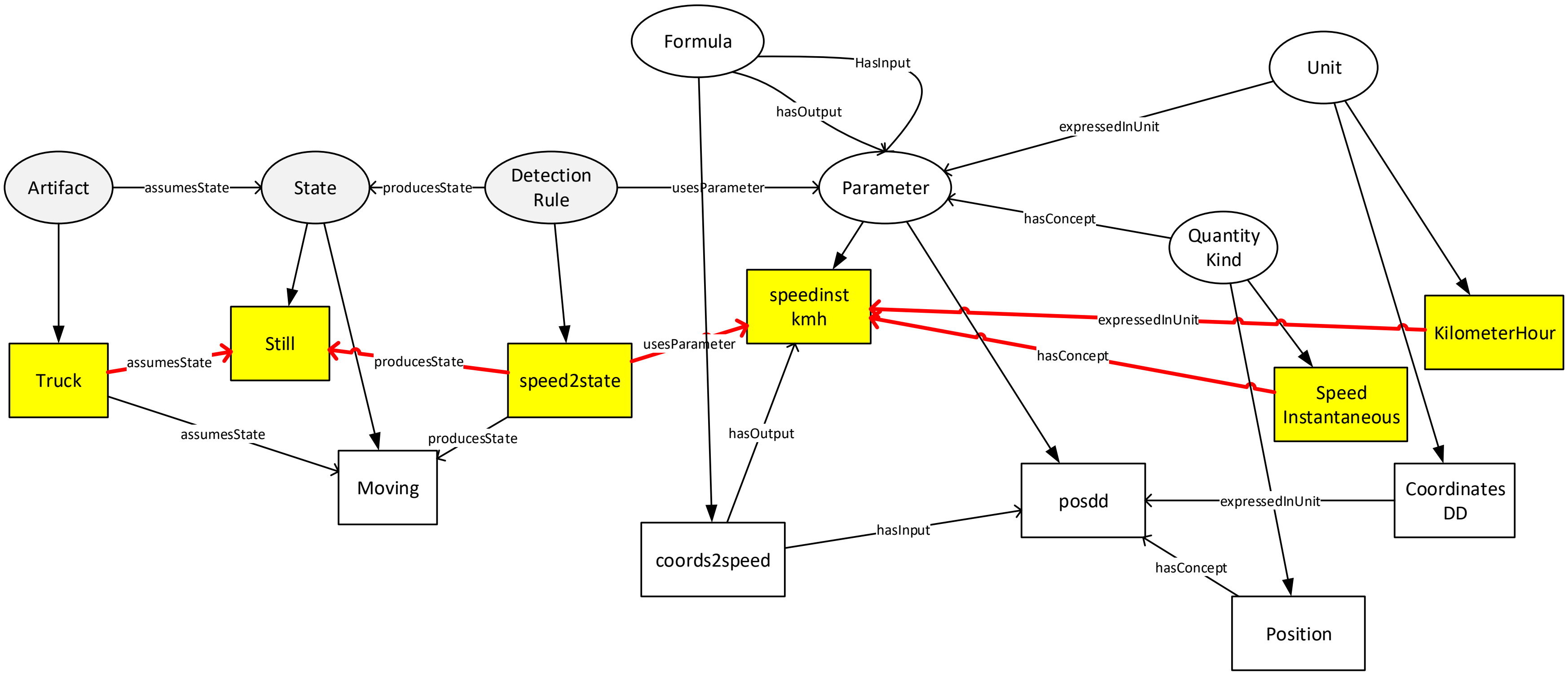
Process monitorability assessment

- Determine if truck **AB123XY** can infer **<truck, still>**:
  - ☑ If a detection rule to infer **still** exists
  - ☐ Which parameters are required by that rule
  - ☐ If the sensors on **AB123XY** provide the required parameters



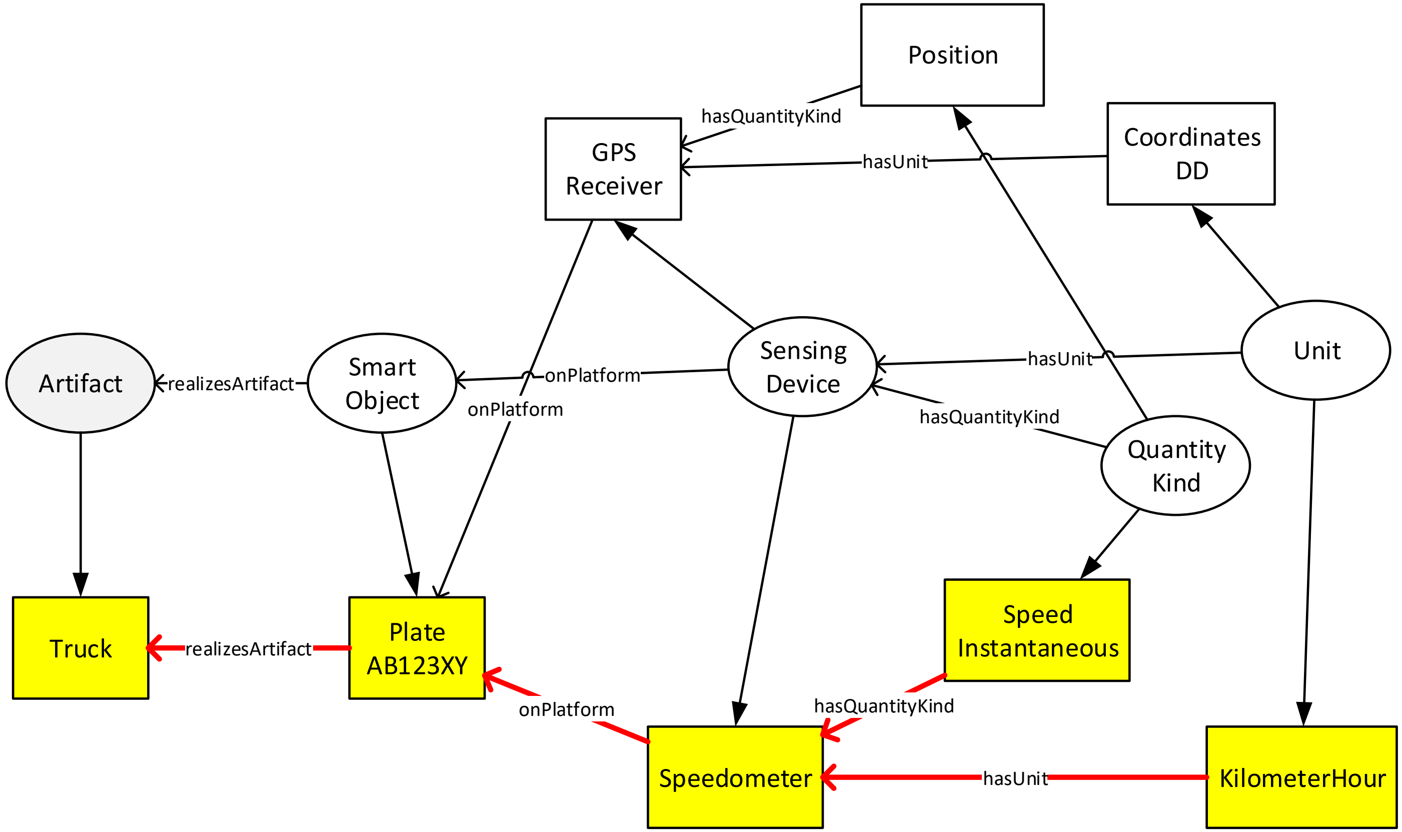
Process monitorability assessment

- Determine if truck **AB123XY** can infer **<truck, still>**:
  - ☑ If a detection rule to infer **still** exists
  - ☑ Which parameters are required by that rule
  - ☐ If the sensors on **AB123XY** provide the required parameters



Process monitorability assessment

- Determine if truck **AB123XY** can infer **<truck, still>**:
  - ☑ If a detection rule to infer **still** exists
  - ☑ Which parameters are required by that rule
  - ☑ If the sensors on **AB123XY** provide the required parameters



Process monitorability assessment

- Once Mon^{ARS} has been determined for every couple <artifact, state>, the monitorability of the activation and the termination of an activity is determined as:

$$Mon^{C}(A_i.C_i^{start}, I) \rightarrow [0,1] = \prod^{ARS_{i,j} \in A_i.C_i^{start}} Mon^{ARS}(ARS_{i,j}, I) \quad (1)$$

$$Mon^{C}(A_i.C_i^{stop}, I) \rightarrow [0,1] = \prod^{ARS_{i,k} \in A_i.C_i^{stop}} Mon^{ARS}(ARS_{i,k}, I) \quad (2)$$

- Then, the monitorability of an activity is:

$$Mon^{A}(A_i, I) \rightarrow [0,1] = \frac{1}{2} \cdot \Big(Mon^{C}(A_i.C_i^{start}, I) + Mon^{C}(A_i.C_i^{stop}, I) \Big)$$

- Finally, the monitorability of the process is:

$$Mon^{P}(P, I) \rightarrow [0,1] = \frac{\sum^{A_i \in P} Mon^{A}(A_i, I)}{|A_i \in P|}$$

Process monitorability improvement

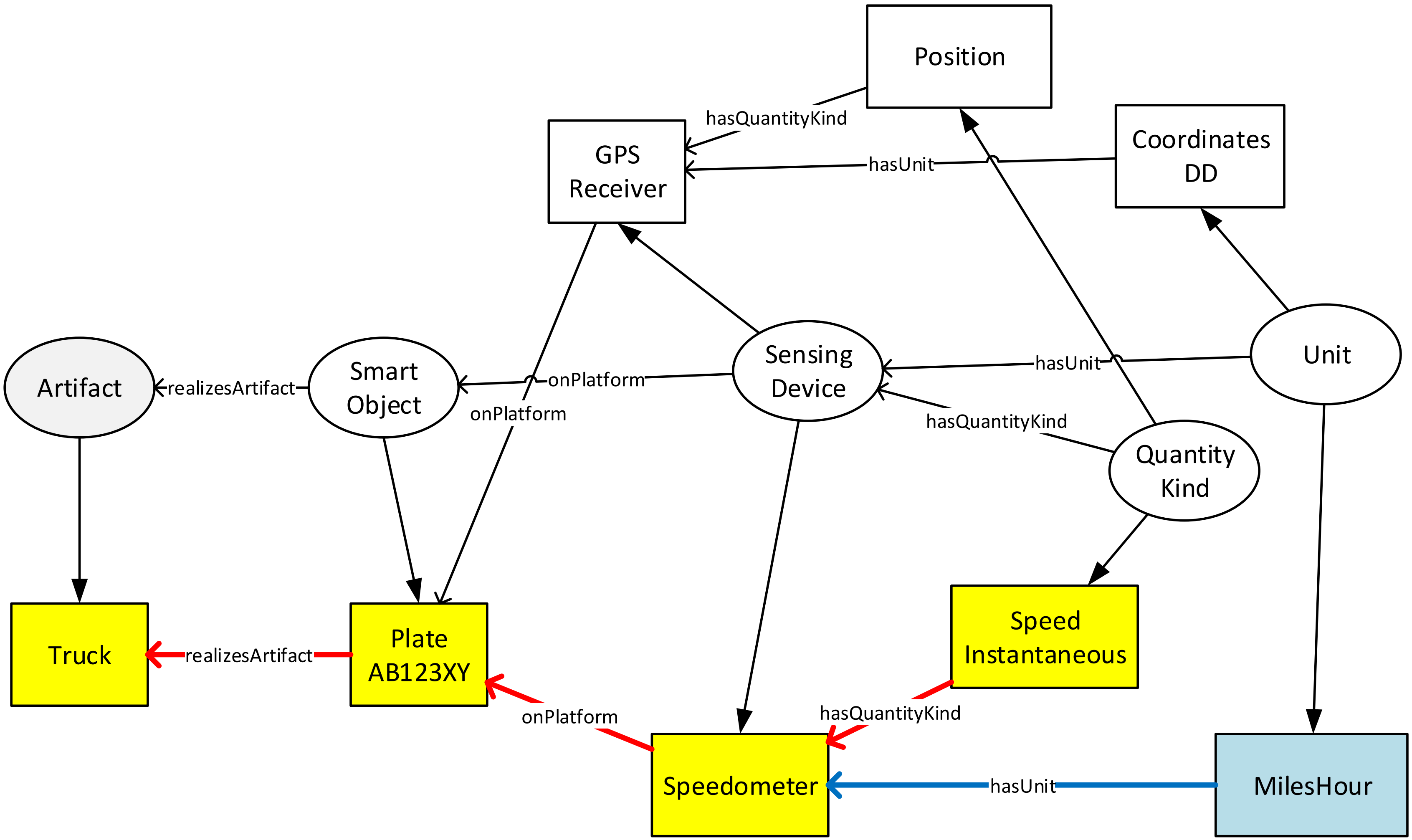
- To improve monitorability, three types of actions are possible:
  - Alter the process model to rely on different artifacts or states to determine when activities are executed
  - Improve the state detection rules
  - Modify the smart objects introducing new sensors
- When altering the process model, for each couple **<artifact, state>** that cannot be monitored, the ontologies can suggest:
  - Another state **state’** for **artifact** such that:
    $Mon^{ARS}(\langle artifact, state' \rangle, I) > 0$
  - Another artifact **artifact** such that
    $Mon^{ARS}(\langle artifact', state \rangle, I) > 0$

Process monitorability improvement

- To improve the state detection rules, the ontologies can detect smart objects that:
  - Cannot detect a state just because their sensors use a data format different from the one required by the detection rule
  - Cannot detect a state, but provide sensor data that can be used to indirectly derive that state
- By introducing a new detection rule similar to the existing one except for the input parameters, these smart objects can detect that state.
  - This positively affects the monitorability of the process
- For smart objects that cannot provide sensor data to detect that state, either directly or indirectly, the ontologies can suggest which sensors should be introduced

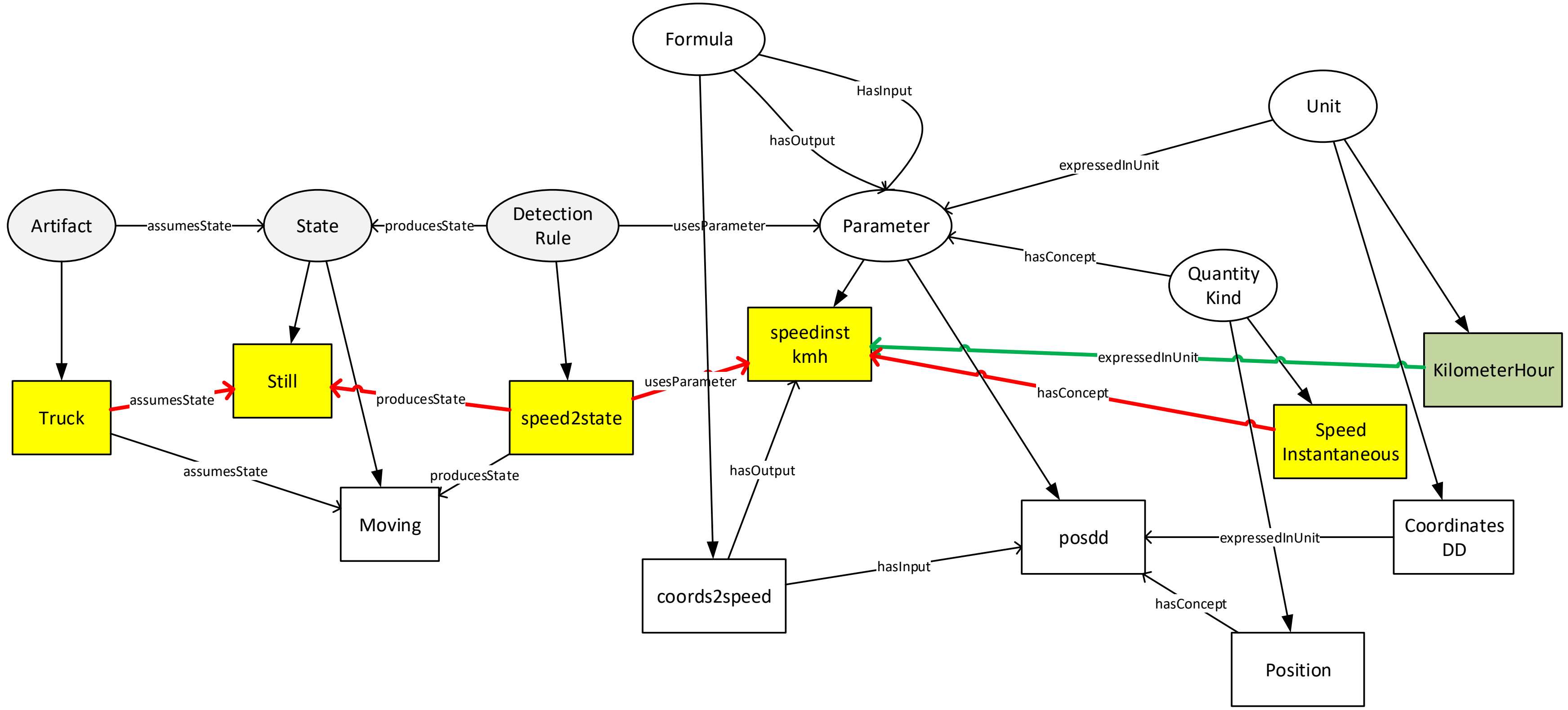
Process monitorability improvement

- Truck CD456WX provides the speed in miles per hour



Process monitorability improvement

- Truck CD456WX provides the speed in miles per hour
- To detect <truck, still>, rule speed2state requires the speed to be expressed in kilometers per hour
- Truck CD456WX cannot use speed2state, so it cannot detect <truck, still>

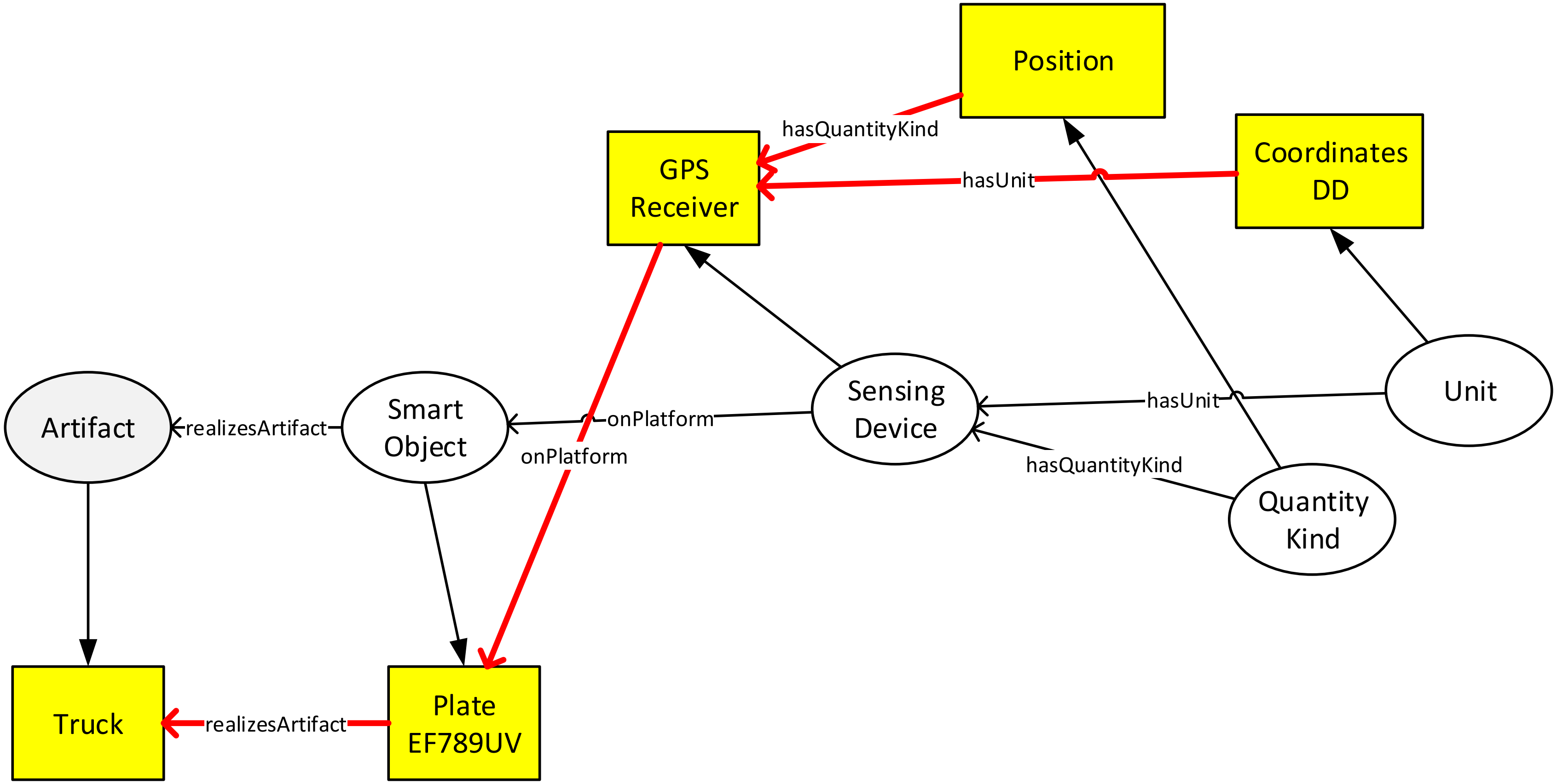


Process monitorability improvement

- Truck CD456WX provides the speed in miles per hour
- To detect <truck, still>, rule speed2state requires the speed to be expressed in kilometers per hour
- Truck CD456WX cannot use speed2state, so it cannot detect <truck, still>
- A new rule speed2state’ can be derived from speed2state by converting the speed from miles per hour to kilometers per hour
- With speed2state’, Truck CD456WX can now detect <truck, still>

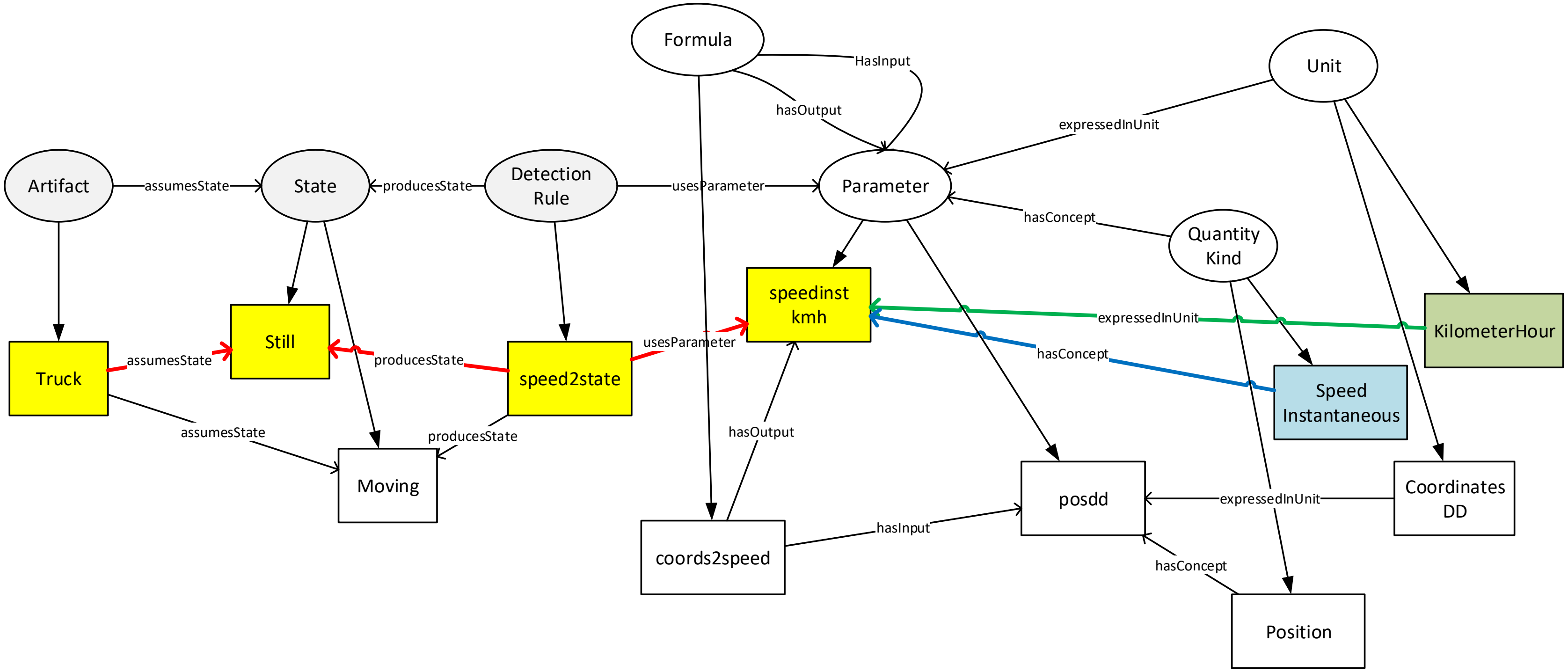
Process monitorability improvement

- Truck EF789UV provides its own position in decimal degrees coordinates



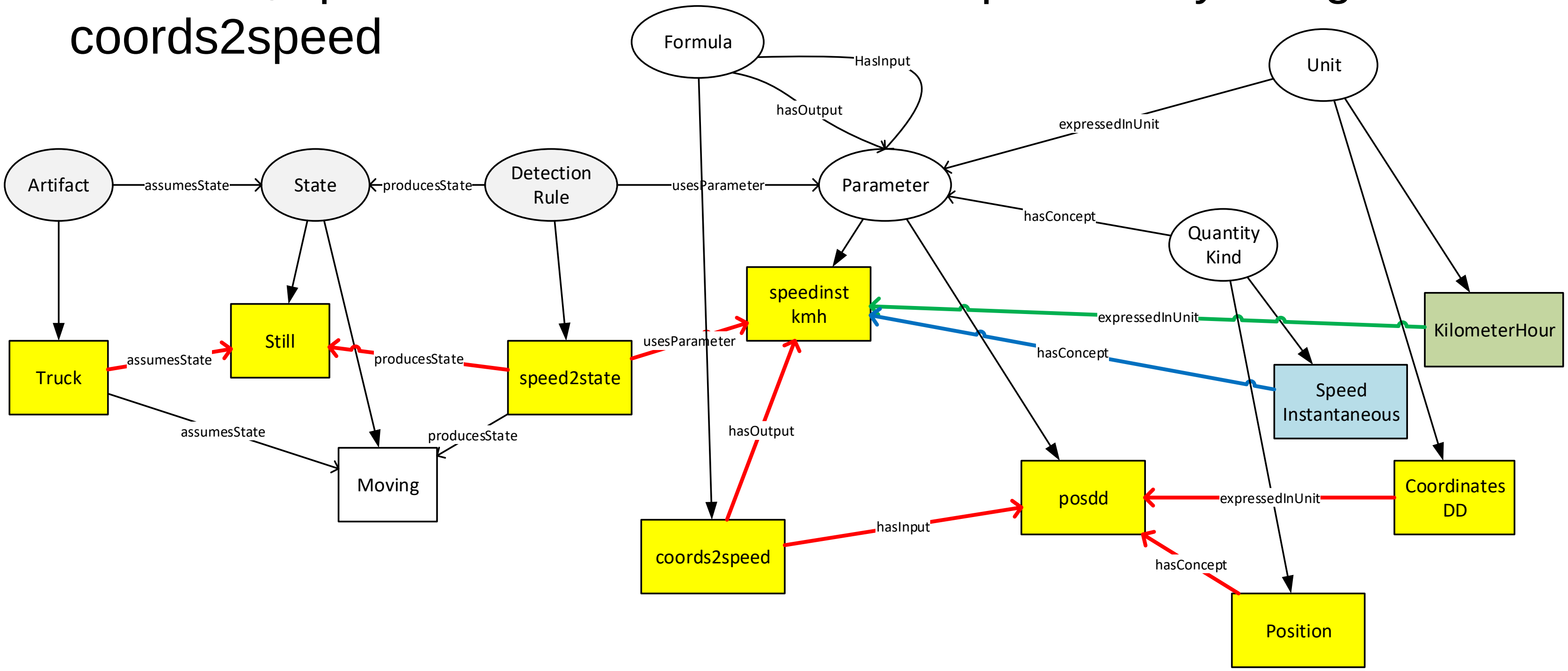
Process monitorability improvement

- Truck EF789UV provides its own position in decimal degrees coordinates
- To detect <truck, still>, rule speed2state requires the speed
- Truck EF789UV cannot use speed2state, so it cannot detect <truck, still>



Process monitorability improvement

- Truck EF789UV provides its own position in decimal degrees coordinates
- To detect <truck, still>, rule speed2state requires the speed
- Truck EF789UV cannot use speed2state, so it cannot detect <truck, still>
- However, speed can be derived from the position by using formula coords2speed



Process monitorability improvement

- Truck EF789UV provides its own position in decimal degrees coordinates
- To detect <truck, still>, rule speed2state requires the speed
- Truck EF789UV cannot use speed2state, so it cannot detect <truck, still>
- However, speed can be derived from the position by using formula coords2speed
- A new rule coords2state can be derived by combining speed2state with coords2speed
- With coords2state, Truck CD456WX can now detect <truck, still>

Conclusion

- Artifact-driven process monitoring relies on the state of the artifacts participating to a process to determine when activities are performed
- The monitorability of a process depends on the capabilities of the smart objects embodying the artifacts
- Capabilities can be formalized with ontologies
- Ontologies can be queried to:
  - Automatically determine the monitorability of a process based on a process model
  - Suggest modifications in the process model, smart objects and detection rules to improve monitorability
- Future work on automatically configuring smart objects
  - i.e. deploying on smart objects only detection rules that are needed to monitor a process

Thanks for your attention

Any question?

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