CPS Infrastructure Rules   
(refers to CPS Profile)

Every defined rule has the following attributes:

* Rule level
* Rule aspect
* Rule element (from profile)
* Rule description
* Rule condition (algorithm)
* Model evolution recommendation
* Model evolution recommendation domain
* Non-functional property (quality) affected
* Source or justification
* Rule-specific attribute (optional)
* Rule-specific attribute execution period (optional)

Rules are categorized in different levels (application, infrastructure, integration) as per division in U-Test project. Rule aspects are categorized as per division proposed by NIST CPS Framework[[1]](#footnote-1) - data, functional, business, human, trustworthiness, timing, boundaries, composition, lifecycle, plus security, as intrinsic to all aspects of CPS. Rules reference the profile which they are using as a basis for uncertainty detection and a specific profile element. Each rule has a verbal description and the accompanying algorithm, with the conditions which have to be met. Each rule provides a possible recommendation for model evolution, while stating the possible domain for further research. Each rule affects one or more non-functional properties (qualities) of a CPS. Each rule has to be backed up by at least one source (paper, best practice recommendation, existing standards etc.), or in experimental research domains, if there are no sources, it needs to have a justifiable clarification. If there is no attribute in referenced profile to apply the rule, we define them as rule-specific attributes. In such case, we define if the attribute value represents the design-time (expected) or run-time (measured) value.

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::Unit** |
| Rule description | Sensor data should be timestamped, to monitor the latency between when a sensor measurement event actually occurred and the time at which the data was made available to the CPS Unit.  Ensure that the temporal aspects of data that are common to more system components are based on a common understanding of reference timescales so that logical operations and computations on these data are meaningful. |
| Rule condition (algorithm) | IF **Unit.timestampMechanism==notImplemented** |
| Model evolution recommendation | Please implement the timestamp mechanism. |
| Model evolution domain | Data timestamping |
| Non-functional property | Latency |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 4 |
| Rule-specific attribute | Element Unit, attribute timestampMechanism values: implemented, notImplemented |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::**CommunicationProtocol |
| Rule description | In CPS machine-to-machine communication, new generations of lightweight communication protocols exist and should be considered to reduce the bandwith, latency and power usage |
| Rule condition (algorithm) | IF **CommunicationProtocol.protocolType==TCP** |
| Model evolution recommendation | Please evolve the system to use lightweight M2M protocols, i.e. MQTT to increase latency and reduce bandwidth |
| Model evolution domain | Lightweight protocols (CPS, IoT) |
| Non-functional property (quality) | Latency |
| Source or justification | As described |
| Domain-specific attribute | Element **CommunicationProtocol**, attributes **protocolType**, value MQTT |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::Sensor** |
| Rule description | Sensors occasionally may tend to produce out-of-range and inaccurate data. Please implement the sensor value boundaries and test the system. |
| Rule condition (algorithm) | IF **sensor.valueMin**=NULL OR IF **sensor.valueMin**=NULL |
| Model evolution recommendation | Please implement the sensor value boundaries or similar sensor defect detection strategy and test the system. |
| Model evolution domain | Sensor defect detection |
| Non-functional property | Believability |
| Source or justification | [1] Ehlenbröker, Jan-Friedrich, Uwe Mönks, and Volker Lohweg. "Sensor defect detection in multisensor information fusion." *Journal of Sensors and Sensor Systems* 5.2 (2016): 337. |
| Rule-specific attribute | Element Sensor, attributes valueMin:int and valueMax:int |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::Sensor** |
| Rule description | Sensor data should be timestamped, to monitor the latency between when a sensor measurement event actually occurred and the time at which the data was made available to the CPS Unit.  Ensure that the temporal aspects of data that are common to more system components are based on a common understanding of reference timescales so that logical operations and computations on these data are meaningful. |
| Rule condition (algorithm) | IF **Sensor.timestampMechanism==notImplemented** |
| Model evolution recommendation | Please implement the timestamp mechanism. |
| Model evolution domain | Data timestamping and monitoring |
| Non-functional property | Latency |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 4, 53 |
| Rule-specific attribute | Element Unit, attribute timestampMechanism values: implemented, notImplemented |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::Unit** |
| Rule description | CPS Units should have defined maximum latency for their mutual communication. CPS timing requirements are generally expressed as constraints on the time intervals between pairs of system significant events. These should be monitored |
| Rule condition (algorithm) | IF **Unit.timestampMaxLatency==NULL** |
| Model evolution recommendation | Please implement the timestamp monitoring mechanism. |
| Model evolution domain | Data timestamping and monitoring |
| Non-functional property | Latency |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 31 |
| Rule-specific attribute | Element Unit, attribute timestampMaxLatency::int |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::PhysicalUnit::Actuator** |
| Rule description | Components that contain actuators should have an appropriate level of awareness of physical location and time. For example, the accuracy requirement for location will change based upon the application. To support such applications, components may need the ability to access and/or report both location and the associated uncertainty of the location. |
| Rule condition (algorithm) | IF **Actuator.location==NULL** |
| Model evolution recommendation | Please implement the communication between actuators. |
| Model evolution domain | Location management, indoor object tracking |
| Non-functional property | Safety |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 13 |
| Rule-specific attribute | Element Actuator, attribute location::int |
| Rule-specific attribute execution period | Desing-time and run-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::PhysicalUnit::Sensor** |
| Rule description | Components that contain sensors should have an appropriate level of awareness of physical location and time. For example, the accuracy requirement for location will change based upon the application. To support such applications, components may need the ability to access and/or report both location and the associated uncertainty of the location. |
| Rule condition (algorithm) | IF **Sensor.location==NULL** |
| Model evolution recommendation | Please implement the communication between actuators. |
| Model evolution domain | Location management |
| Non-functional property | Safety |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 13 |
| Rule-specific attribute | Element Sensor, attribute location::int |
| Rule-specific attribute execution period | Desing-time and run-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::PhysicalUnit::Sensor** |
| Rule description | Components that contain sensors should have an appropriate level of awareness of physical location and time. For example, the accuracy requirement for location will change based upon the application. To support such applications, components may need the ability to access and/or report both location and the associated uncertainty of the location. |
| Rule condition (algorithm) | IF **Sensor.location==NULL** |
| Model evolution recommendation | Please implement the communication between actuators. |
| Model evolution domain | Location management |
| Non-functional property | Safety |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 13 |
| Rule-specific attribute | Element Sensor, attribute location::int |
| Rule-specific attribute execution period | Desing-time and run-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Security |
| Rule Element (from profile) | **CPSProfile::PhysicalUnit::Actuator** |
| Rule description | If an actuator is safety-critical (e.g. centrifuge in chemical plant, that may cause harm to a CPS), consider adding adding new physical controls over the CPS (e.g. manual valves) to reduce possible harm in case of misuse. |
| Rule condition (algorithm) | IF **Actuator.safetyCritical==TRUE** |
| Model evolution recommendation | Please consider adding adding new physical controls over the CPS (e.g. manual valves) to reduce possible harm in case of misuse. |
| Model evolution domain | Physical safety |
| Non-functional property | Safety |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 79 |
| Rule-specific attribute | Element Actuator, attribute safetyCritical values: true, false |
| Rule-specific attribute execution period | Desing-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Functional, composition |
| Rule Element (from profile) |  |
| Rule description | Concerns related to the ability of the CPS to achieve an intended purpose in the face of changing external conditions such as the need to upgrade or otherwise reconfigure a CPS to meet new conditions, needs, or objectives. |
| Rule condition (algorithm) | IF |
| Model evolution recommendation | Please ensure elasticity |
| Model evolution domain |  |
| Non-functional property | Adaptability |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 79 |
| Rule-specific attribute | Element, attribute |
| Rule-specific attribute execution period | Desing-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Security |
| Rule Element (from profile) | **CPSProfile::PhysicalUnit::Actuator** |
| Rule description | When actuators act together, i.e. there is more than one physically moving, a communication mechanism should be ensured between them to avoid possible undesired effect, e.g. collision. |
| Rule condition (algorithm) | IF **Actuator.composition==true** |
| Model evolution recommendation | Please implement the communication between actuators. |
| Model evolution domain | ? |
| Non-functional property | Safety |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 9 |
| Rule-specific attribute | Element Actuator, attribute composition  Values: true, false |
| Rule-specific attribute execution period | Design-time and run-time |

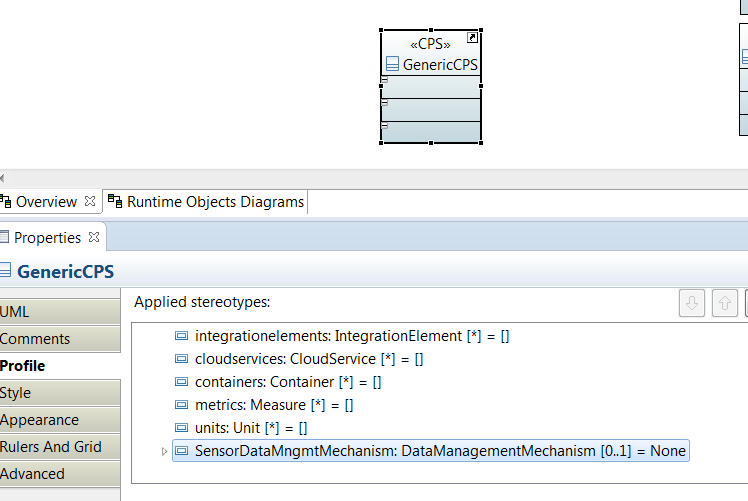
|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Trustworthiness |
| Rule Element (from profile) | **CPSProfile::Communication** |
| Rule description | Message exchange between CPS elements should not be tampered with. Please |
| Rule condition (algorithm) | IF **Communication.encryption==FALSE** |
| Model evolution recommendation | Please ensure encryption in communication whenever possible. |
| Model evolution domain | Encryption, End-to-end encryption |
| Non-functional property | Security |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 35 |
| Rule-specific attribute | Element Communication, attribute encryption values: true, false |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Trustworthiness |
| Rule Element (from profile) | **CPSProfile::Communication** |
| Rule description | Message exchange between CPS elements should not be tampered with. Please |
| Rule condition (algorithm) | IF **Communication.encryption==FALSE** |
| Model evolution recommendation | Please ensure encryption in communication whenever possible. |
| Model evolution domain | Encryption, End-to-end encryption |
| Non-functional property | Security |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 35 |
| Rule-specific attribute | Element Communication, attribute encryption values: true, false |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Security |
| Rule Element (from profile) | **CPSProfile::PhysicalUnit::Actuator** |
| Rule description | If an actuator is mission-critical , in case of cyber attacks, please ensure graceful degradation in accordance with mission- or business-driven priorities, and enable the system to fail-safe in those circumstances in which resilience cannot be provided in the face of threat. |
| Rule condition (algorithm) | IF **Actuator.missionCritical==TRUE** |
| Model evolution recommendation | please ensure graceful degradation in accordance with mission- or business-driven priorities, and enable the system to fail-safe in those circumstances in which resilience cannot be provided in the face of threat. |
| Model evolution domain | Mission-critical CPS |
| Non-functional property | Safety |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 70 |
| Rule-specific attribute | Element Actuator, attribute missionCritical values: true, false |
| Rule-specific attribute execution period | Desing-time |

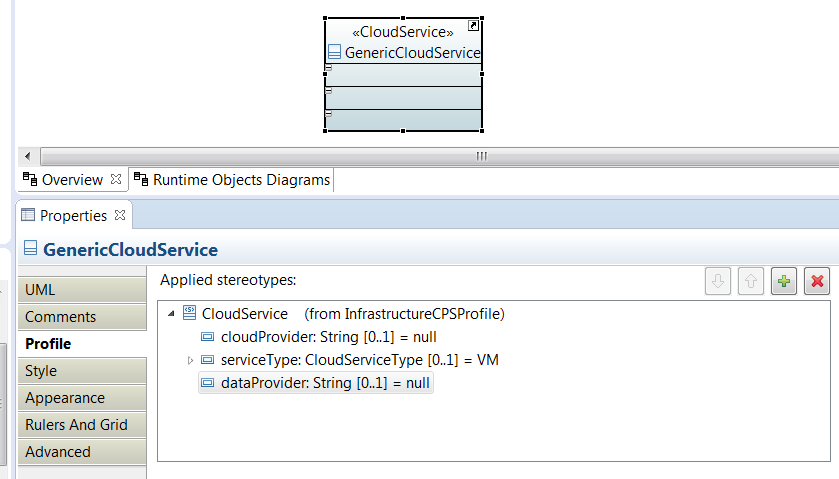
|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Security |
| Rule Element (from profile) | **CPSProfile::PhysicalUnit::Sensor** |
| Rule description | If Sensor’s location is external, and it can be easily physically accessed, please consider possibility of  of deception attacks (e.g., attacks on sensors that can lead them to input malicious data to the cyber component and, as a result, to provide wrong, or even dangerous, output from the cyber component) |
| Rule condition (algorithm) | IF **Sensor.location==external AND Sensor.physicallyAccessible==TRUE** |
| Model evolution recommendation | Please ensure sensor’s physical security. |
| Model evolution domain | Sensor physical security |
| Non-functional property | Safety |
| Source or justification | NIST Framework for Cyber-Physical Systems, page 88 |
| Rule-specific attribute | Element Sensor, attribute physicallyAcsessible values: true, false |
| Rule-specific attribute execution period | Desing-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::**Sensor |
| Rule description | Sensors may fail in sending the data to other components of the system. Please ensure the mechanisms for detection and recovery in such cases |
| Rule condition (algorithm) | IF **Sensor.value**=NULL |
| Model evolution recommendation | Please ensure the mechanisms for detection and recovery in such cases, e.g. disable the sensor data processing when NULL reading occurs, and attempt to re-establish the connection with the sensor |
| Model evolution domain | Networking |
| Non-functional property (quality) | Accuracy |
| Source or justification | As described |
| Domain-specific attribute | Element **Sensor**, attributes **value:int** |
| Rule-specific attribute execution period | Run-time |



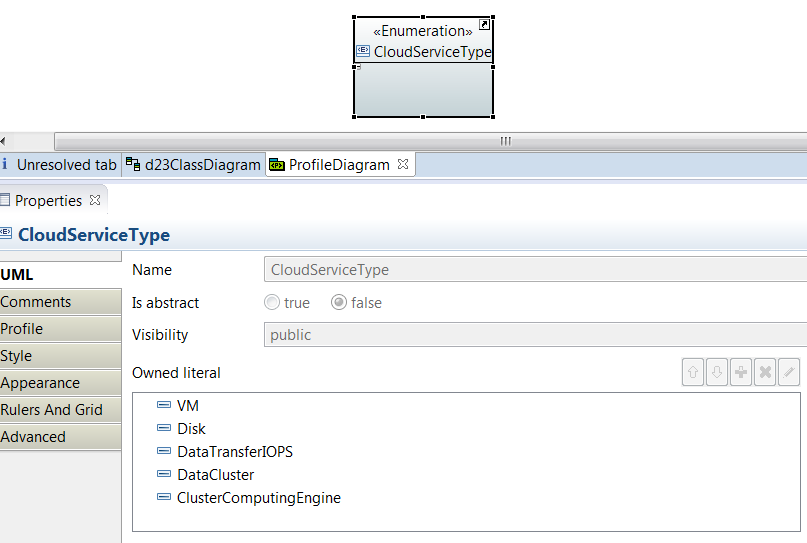
|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect | Data |
| Rule Element (from profile) | **CPSProfile::**Sensor, **CPSProfile::CPS** |
| Rule description | Sensors may produce too much data (e.g. if sensors are activated by certain events) which the CPS can not handle due to its limitations. A mechanism for handling such events is necessary. |
| Rule condition (algorithm) | **IF CPS::SensorDataManagementMechanism != None** |
| Model evolution recommendation | Please test the system with both maximum and minimum workload of sensors to find out its limitations. Additionally, please ensure the mechanisms for ensuring the elasticity of the CPS in such occasions. |
| Model evolution domain | CPS Scalability, Elasticity |
| Non-functional property (quality) | Scalability |
| Source or justification | As described |
| Domain-specific attribute | Element Sensor, **Element CPS** attributes |
| Rule-specific attribute execution period | Design-time |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect |  |
| Rule Element (from profile) | **CPSProfile::**CloudService |
| Rule description | Cloud data service, as part of CPS, may have uptime may lower than 99.9%. In such occasions, mechanisms for fail-safe work should be ensured. |
| Rule condition (algorithm) | IF **CloudService.localBackup**!=NULL |
| Model evolution recommendation | Please evolve the system that relies on Cloud data infrastructure to ensure that it is fail-safe, i.e. that the data is backed up both locally and in the Cloud.  Additionally, please ensure the mechanisms for checking the accuracy of data stored locally and in the Cloud. |
| Model evolution domain | Local and Cloud backup |
| Non-functional property (quality) | Availability, Accuracy, |
| Source or justification | As described |
| Domain-specific attribute | CloudService.localBackup, values::implemented/notImplemented |
| Rule-specific attribute execution period | Run-time |



ServiceType enumeration is provided by another Internal UTest Library that I do not modify to not break compatibility with other partners’ tools.

Here below the definition of the CloudServiceType Enumeration



|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect |  |
| Rule Element (from profile) | **CPSProfile::**CommunicationProtocol |
| Rule description | In CPS machine-to-machine communication protocols such as TCP or UDP, it may happen that transport layer sends packages too big for the network, which reduces QoS.  Please implement the mechanism of sending and receiving fragmented packages if they exceed the predetermined size. |
| Rule condition (algorithm) | IF CommunicationProtocol.protocolType==TCP OR UDP |
| Model evolution recommendation | Please evolve the system such that it implements the mechanism of sending and receiving fragmented packages if they exceed the predetermined size. |
| Model evolution domain |  |
| Non-functional property (quality) | Latency |
| Source or justification | [5] Network Challenges for Cyber Physical Systems with Tiny Wireless Devices: A Case Study on Reliable Pipeline  Condition Monitoring |
| Domain-specific attribute |  |
| Rule-specific attribute execution period |  |

|  |  |
| --- | --- |
| CPS Rule | |
| Rule level | CPS infrastructure |
| Rule aspect |  |
| Rule Element (from profile) | **CPSProfile::**VirtualGateway |
| Rule description | The challenge of thousands of sensor nodes need to be addressed on both sides of the gateway, i.e. the Cloud and the sensing gateway.  Please ensure self-management policies such that resources (CPU time, bandwidth memory, energy) can be be controlled and monitored. |
| Rule condition (algorithm) | IF VirtualGateway and nextState==CloudService |
| Model evolution recommendation | Please evolve the system to ensure self-management policies such that resources (CPU time, bandwidth memory, energy) can be controlled and monitored. |
| Model evolution domain |  |
| Non-functional property (quality) | QoS in general |
| Source or justification | [5] - Network Challenges for Cyber Physical Systems with Tiny Wireless Devices: A Case Study on Reliable Pipeline  Condition Monitoring |
| Domain-specific attribute |  |
| Rule-specific attribute execution period |  |

1. NIST – National Institute of Standards and Technology

   CPS PWG Cyber-Physical Systems (CPS) Framework Release 1.0, 2015, <https://pages.nist.gov/cpspwg/> [↑](#footnote-ref-1)