**SQWRL Expressions, descriptions, and results**

The inferences were run in a Lenovo Ideapad 700 with 16GB RAM, Core i5 6300HQ.

**Table 1. Case Study: Metal Separation Process.**

|  |  |  |
| --- | --- | --- |
| **SQWRL Expression** | **Description** | **Result** |
| State(?s) ^ isActive(?s, true) -> sqwrl:select(?s) | Verifies which states are active dynamically. | Active State: :State-0  Elapsed time (ms): 5035 |
| CurrentState(?cs) -> sqwrl:select(?cs) | Verifies which states are currently active dynamically. | Current State: :State-0  Elapsed time (ms): 4951 |
| NextState(?ns) -> sqwrl:select(?ns) | Verifies inference which states are next states of the current state(s) dynamically. | Next State: :State-4  Elapsed time (ms): 4898 |
| PreviousState(?ps) -> sqwrl:select(?ps) | Verifies which states are previous states of the current state(s) dynamically. | Previous State: :State-4  Elapsed time (ms): 5185 |
| Thing(?x) -> sqwrl:selectDistinct(?x) | Verifies which agents are Things in the system. | I am a Thing: :Agent-I-RA-Reader  Elapsed time (ms): 5066 |
| Agent(?a) ^ Service(?s) ^ offers(?a, ?s) -> sqwrl:select(?a, ?s) | Verifies which services offer each agent in the system. | I am: :Agent-II-RA-Separator and I offer: :Perform-Separation-Service  I am: :Agent-III-RA-Saver and I offer: :Save-Data-Service  I am: :Agent-I-RA-Reader and I offer: :Send-Piece-Data-Service  Elapsed time (ms): 5610 |
| Standard(?s) ^ hasEmbeddingCapability(?s, ?c) -> sqwrl:select(?s, ?c) | Verifies which standards are tagged with None, Low, Medium, or High embedding capability in the system. | Standard: :Standard-IEC-62264-ISA-95 | Embedding capability: "High"^^xsd:string  Standard: :Standard-W3C | Embedding capability: "High"^^xsd:string  Standard: :Standard-OPC-UA | Embedding capability: "High"^^xsd:string  Standard: :Standard-IEC-61131 | Embedding capability: "High"^^xsd:string  Standard: :Standard-IEC-61499 | Embedding capability: "High"^^xsd:string  Standard: :Standard-IEC-62890 | Embedding capability: "Low"^^xsd:string  Standard: :Standard-IEC-61512-ISA-88 | Embedding capability: "Medium"^^xsd:string  Elapsed time (ms): 5339 |
| Sequence(?s) ^ isParentSetElement(?s, true) -> sqwrl:select(?s) | Verifies which sequences are parent (main) sequences in the system. | Parent Sequence: :Metal-Separation-Sequence  Elapsed time (ms): 5268 |
| Agent(?a) ^ belongsToArchitectureLayer(?a, ?lvl) -> sqwrl:selectDistinct(?a, ?lvl) | Verifies the level of the ISA 95 architecture that each agent in the system belongs to. | Agent :Agent-I-RA-Reader located in : "1"^^xsd:integer architecture layer (ISA95)  Agent :Agent-I-RA-Reader located in : "3"^^xsd:integer architecture layer (ISA95)  Agent :Agent-II-RA-Separator located in : "2"^^xsd:integer architecture layer (ISA95)  Agent :Agent-III-RA-Saver located in : "4"^^xsd:integer architecture layer (ISA95)  Elapsed time (ms): 4955 |
| Agent(?a) ^ hasDescription(?a, ?d) ^ swrlb:contains(?d, "controller") -> sqwrl:select(?a) | Consists of verifying by means of inference which agents contains the word ‘controller’ in the description. | Agent :Agent-II-RA-Separator contains 'controller' in description  Elapsed time (ms): 5094 |
| ConcurrentState(?cs) -> sqwrl:select(?cs) | Consists of verifying by means of inference which states run concurrently or in parallel execution dynamically. | (Lacks of updating assertions via SWRL Engine online)  Elapsed time (ms): 5199 |
| Device(?x) -> sqwrl:selectDistinct(?x) | Consists of verifying by means of inference which agents are Devices in the system. | I am a Device : :Agent-II-RA-Separator  Elapsed time (ms): 5066 |
| FinalState(?fs) -> sqwrl:select(?fs) | Consists of verifying by means of inference which states are final states in the system. | Final State : :State-6  Final State : :State-7  Elapsed time (ms): 4952 |
| hasInteroperabilityDegree(?x, "High") ^ Agent(?x) -> sqwrl:selectDistinct(?x) | Consists of verifying by means of inference which agents are tagged with a High interoperability degree in the system. | (Lacks of updating assertions via SWRL Engine online)  Elapsed time (ms): 5104 |
| InitialState(?is) -> sqwrl:select(?is) | Consists of verifying by means of inference which states are initial states in the system. | Initial state : :State-0  Elapsed time (ms): 5189 |
| Service(?s) ^ hasDescription(?s, ?d) ^ swrlb:contains(?d, "database") -> sqwrl:select(?s) | Consists of verifying by means of inference which services contains the word ‘database’ in the description. | Service : :Save-Data-Service contains 'database' in service description  Elapsed time (ms): 5943 |
| SoftwareResource(?x) -> sqwrl:selectDistinct(?x) | Consists of verifying by means of inference which actors are Software Resources in the system. | I am a Software Resource : :Agent-I-RA-Reader  I am a Software Resource : :Agent-III-RA-Saver  Elapsed time (ms): 5823 |
| SynchronousState(?ss) -> sqwrl:select(?ss) | Consists of verifying by means of inference which states are synchronous states in the system dynamically. | (Lacks of updating assertions via SWRL Engine online)  Elapsed time (ms): 5682 |
| Standard(?s) ^ Device(?d) ^ standardizes(?s, ?d) -> sqwrl:selectDistinct(?s) | Consists of verifying by means of inference which standards concern to devices in the system. | Standard : :Standard-IEC-61131 concerns to devices  Standard : :Standard-IEC-61499 concerns to devices  Standard : :Standard-IEC-62264-ISA-95 concerns to devices  Standard : :Standard-OPC-UA concerns to devices  Elapsed time (ms): 5446 |
| interacts(?x, ?y) ^ Actor(?y) ^ Actor(?x) -> sqwrl:selectDistinct(?x, ?y) | Consists of verifying by means of inference which actors communicate or interact one another. | (Lacks of updating assertions via SWRL Engine online)  Elapsed time (ms): 6060 |
| interacts(?x, ?y) ^ Agent(?y) ^ Agent(?x) -> sqwrl:selectDistinct(?x, ?y) | Consists of verifying by means of inference which agents communicate or interact one another. | (Lacks of updating assertions via SWRL Engine online)  Elapsed time (ms): 6060 |
| Agent(?a) ^ hasFeature(?a, ?f) ^ swrlb:equal(?f, "Proactive") -> sqwrl:selectDistinct(?a) | Queries distinct agents which are classified as ‘proactive’ in the system. | Agent :Agent-III-RA-Saver is proactive  Elapsed time (ms): 7402 |
| Agent(?a) ^ hasFeature(?a, ?f) ^ swrlb:equal(?f, "Reactive") -> sqwrl:selectDistinct(?a) | Queries distinct agents which are classified as ‘reactive’ in the system. | Agent :Agent-I-RA-Reader is reactive  Agent :Agent-II-RA-Separator is reactive  Elapsed time (ms): 5792 |
| Asset(?a) ^ isVirtualizedIn(?a, ?dt) ^ AdministrationShell(?dt) ^ hasFile(?dt, ?f) -> sqwrl:selectDistinct(?a, ?dt, ?f) | Identifies triples of Assets/DigitalTwins (Admin Shell)/Files for assets in the process. | (No individuals were provided)  Elapsed time (ms): 8282 |

**Table 2. Case Study: xPPU – Scenario 11 (taken from: https://mediatum.ub.tum.de/node?id=1468863).**

|  |  |  |
| --- | --- | --- |
| **SQWRL Expression** | **Description** | **Result** |
| Output(?o) ^ Object(?ob) actsOver(?o,?ob) -> sqwrl:selectDistinct(?o, ?ob) | Retrieves outputs – objects pairs | Output :Conveyor\_Belt\_Forward\_Output acts over the object :Conveyor\_Belt  Output :Pushing\_Cylinder\_Ramp\_1\_Valve\_Extension\_Output acts over the object :Conveyor\_Belt  Output :Pushing\_Cylinder\_Ramp\_2\_Valve\_Extension\_Output acts over the object :Conveyor\_Belt  Elapsed time (ms): 4805 |
| DataInput(?o) ^ Object(?ob) ^ monitors(?o,?ob) -> sqwrl:selectDistinct(?o, ?ob) | Retrieves data inputs – objects pairs | Input :Current\_Filling\_Ramp\_1 is a data input which monitors :Conveyor\_Belt  Input :Current\_Filling\_Ramp\_2 is a data input which monitors :Conveyor\_Belt  Input :Current\_Filling\_Ramp\_End is a data input which monitors :Conveyor\_Belt |
| Agent(?a) ^ hasDescription(?a, ?d) ^ swrlb:contains(?d, \"xPPU\") -> sqwrl:select(?a) | Retrieves agents with 'xPPU' in the description | Agent :Agent\_Crane contains 'xPPU' in description  Agent :Agent\_Stamp contains 'xPPU' in description  Agent :Agent\_Conveyor\_Belt contains 'xPPU' in description  Elapsed time (ms): 9422 |
| State(?s) ^ DynamicsModel(?m) ^ hasModelElement(?m,?s) -> sqwrl:selectDistinct(?m,?s) | Retrieves the states per dynamics model | Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the stateautogen1:\_5\_WP\_reaches\_light\_and\_inductive\_sensors\_ramp\_2 to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the stateautogen4:\_Presence\_Sensor\_activated to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:Initial\_State\_WP\_Process to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:Stand\_by\_state\_-\_before\_working to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State9\_1\_ConvBelt\_is\_not\_full\_of\_capacity to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_2\_ConvBelt\_moving\_forward to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_3\_WP\_reaches\_light\_and\_inductive\_sensors\_ramp\_1 to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_4\_1\_WP\_is\_white\_and\_pushed\_into\_ramp\_1\_for\_storage to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_4\_WP\_reaches\_pushing\_cylinder\_ramp\_1 to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_6\_1\_WP\_is\_metallic\_and\_pushed\_into\_ramp\_2\_for\_storage to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_6\_WP\_reaches\_pushing\_cylinder\_ramp\_2 to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_7\_1\_WP\_is\_black\_and\_reaches\_end\_ramp\_for\_storage to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_8\_ConvBelt\_stopping to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:State\_9\_ConvBelt\_is\_full\_of\_capacity to perform the automatic operation  Model :xPPU\_separation\_model\_-Conveyor\_Belt\_sequence- contains the state:Stop\_state\_WP\_process to perform the automatic operation  Elapsed time (ms): 4747 |
| Service(?s) ^ hasDescription(?s, ?d) ^ offers(?a,?s) ^ swrlb:contains(?d, \"buffering\") -> sqwrl:select(?s,?a) | Retrieves services with ‘buffering’ in description | Service : :Buffering\_Service contains 'buffering' in service description, provided by agent :Agent\_Stamp  Elapsed time (ms): 5103 |
| Agent(?a) ^ Service(?s) ^ offers(?a, ?s) -> sqwrl:select(?a, ?s) | Verifies which services offer each agent in the system. | I am: :Agent\_Crane and I offer: :Moving\_WPs\_Service  I am: :Agent\_Conveyor\_Belt and I offer: :Storaging\_Service  I am: :Agent\_Stamp and I offer: :Stamping\_WPs\_Service  I am: :Agent\_Stamp and I offer: :Buffering\_Service  Elapsed time (ms): 4153 |
| Agent(?a) ^ isVirtualizedIn(?a, ?dt) ^ AdministrationShell(?dt) ^ hasFile(?dt, ?f) -> sqwrl:selectDistinct(?a, ?dt, ?f) | Retrieves the agent-administration shell-file triples | Agent :Agent\_Conveyor\_Belt has digital twin :Digital\_Twin\_Conveyor\_Belt\_xPPU with file "models/Papyrus-Scenario\_11/model\_Sc11.uml#LargeSortingConveyor\*"^^xsd:string  Agent :Agent\_Crane has digital twin :Digital\_Twin\_Crane\_xPPU with file "models/Papyrus-Scenario\_11/model\_Sc11.uml#Crane\*"^^xsd:string  Agent :Agent\_Stamp has digital twin :Digital\_Twin\_Stamp\_xPPU with file "models/Papyrus-Scenario\_11/model\_Sc11.uml#Stamp\*"^^xsd:string  Elapsed time (ms): 5458 |
| Agent(?a) -> sqwrl:count(?a) | Retrieves the count of agents | Count of agents: "3"^^xsd:int  Elapsed time (ms): 5691 |
| Output(?o) -> sqwrl:count(?o) | Retrieves the count of outputs | Count of outputs: "3"^^xsd:int  Elapsed time (ms): 5177 |
| Input(?i) -> sqwrl:count(?i) | Retrieves the count of inputs | Count of inputs: "15"^^xsd:int  Elapsed time (ms): 5307 |