**SQWRL Expressions, descriptions, and results**

|  |  |  |
| --- | --- | --- |
| **SQWRL Expression** | **Description** | **Result** |
| State(?s) ^ isActive(?s, true) -> sqwrl:select(?s) | Verifies which states are active dynamically. | (Correct)  Active State: :State-6  Active State: :State-7 |
| CurrentState(?cs) -> sqwrl:select(?cs) | Verifies which states are currently active dynamically. | (Correct)  Current State: :State-6  Current State: :State-7 |
| NextState(?ns) -> sqwrl:select(?ns) | Verifies inference which states are next states of the current state(s) dynamically. | (Correct)  Next State: :State-0 |
| PreviousState(?ps) -> sqwrl:select(?ps) | Verifies which states are previous states of the current state(s) dynamically. | (Correct)  Previous State: :State-5 |
| Thing(?x) -> sqwrl:selectDistinct(?x) | Verifies which agents are Things in the system. | (Correct)  I am a Thing: :Device-I |
| Agent(?a) ^ Service(?s) ^ offers(?a, ?s) -> sqwrl:select(?a, ?s) | Verifies which services offer each agent in the system. | (Correct)  I am: :Server-I and I offer: :Save-Data-Service  I am: :Device-I and I offer: :Send-Piece-Data-Service  I am: :Device-II and I offer: :Perform-Separation-Service |
| 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. | (Correct)  Standard: :Standard-IEC-61499 | Embedding capability: "High"^^xsd:string  Standard: :Standard-OPC-UA | Embedding capability: "High"^^xsd:string  Standard: :Standard-IEC-62890 | Embedding capability: "Low"^^xsd:string  Standard: :Standard-IEC-62264-ISA-95 | Embedding capability: "High"^^xsd:string  Standard: :Standard-IEC-61512-ISA-88 | Embedding capability: "Medium"^^xsd:string  Standard: :Standard-IEC-61131 | Embedding capability: "High"^^xsd:string  Standard: :Standard-W3C | Embedding capability: "High"^^xsd:string |
| Sequence(?s) ^ isParentSetElement(?s, true) -> sqwrl:select(?s) | Verifies which sequences are parent (main) sequences in the system. | (Correct)  Parent Sequence: :Metal-Separation-Sequence |
| 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. | (Correct)  Actor :Device-I located in : "1"^^xsd:integer architecture level  Actor :Device-I located in : "3"^^xsd:integer architecture level  Actor :Device-II located in : "2"^^xsd:integer architecture level  Actor :Server-I located in : "4"^^xsd:integer architecture level |
| 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. | (Correct)  Actor :Device-II contains 'controller' in description. |
| ConcurrentState(?cs) -> sqwrl:select(?cs) | Consists of verifying by means of inference which states run concurrently or in parallel execution dynamically. | (Incorrect)  No result. |
| Device(?x) -> sqwrl:selectDistinct(?x) | Consists of verifying by means of inference which agents are Devices in the system. | (Correct)  I am a Device : :Device-II |
| FinalState(?fs) -> sqwrl:select(?fs) | Consists of verifying by means of inference which states are final states in the system. | (Correct)  Final State : :State-7  Final State : :State-6 |
| 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. | (Correct)  Actor with high interoperability degree : :Device-I  Actor with high interoperability degree : :Device-II  Actor with high interoperability degree : :Server-I |
| InitialState(?is) -> sqwrl:select(?is) | Consists of verifying by means of inference which states are initial states in the system. | (Correct)  Initial state : :State-0 |
| 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. | (Correct)  Service : :Save-Data-Service contains 'database' in service description. |
| SoftwareResource(?x) -> sqwrl:selectDistinct(?x) | Consists of verifying by means of inference which actors are Software Resources in the system. | (Correct)  I am a Software Resource : :Device-I  I am a Software Resource : :Server-I |
| SynchronousState(?ss) -> sqwrl:select(?ss) | Consists of verifying by means of inference which states are synchronous states in the system dynamically. | (Incorrect)  No result. |
| 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. | (Correct)  Standard : :Standard-IEC-61131 standardizes devices  Standard : :Standard-IEC-61499 standardizes devices  Standard : :Standard-IEC-62264-ISA-95 standardizes devices  Standard : :Standard-OPC-UA standardizes devices |
| 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. | (Correct)  Actor :Device-I interacts with Actor :Device-II  Actor :Device-I interacts with Actor :Server-I  Actor :Server-I interacts with Actor :Device-I |
| 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. |  |
| Agent(?a) ^ hasFeature(?a, ?f) ^ swrlb:equal(?f, "Proactive") -> sqwrl:selectDistinct(?a) | Queries distinct agents which are classified as ‘proactive’ in the system. |  |
| Agent(?a) ^ hasFeature(?a, ?f) ^ swrlb:equal(?f, "Reactive") -> sqwrl:selectDistinct(?a) | Queries distinct agents which are classified as ‘reactive’ in the system. |  |
| 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. |  |