**Import and edit ontology with Neo4j**

**Xiaochen Zheng**

ICT4SM, EPFL, 1015 Lausanne, Switzerland

[xiaochen.zheng@epfl.ch](mailto:xiaochen.zheng@epfl.ch)

(10-02-2022)

**Abstract:** This document introduces how to import an existing ontology to Neo4j graph database and edit it by adding new classes and relationships. It is executed with Cypher language and enabled by the neosemantics package.

**Keyword:** Neo4j, Neosemantics, Ontology, Python

## Neo4j graph database installation and configuration

A previous document[[1]](#footnote-1) has introduced how to install and configure a Neo4j graph database on an Azure cloud Ubuntu Virtual Machine (VM). Follow that introduction to install and configure the Neo4j database, and install the *neosemantics* package if it’s not done yet.

Before importing ontology, it is necessary to initialize the database with the following code[[2]](#footnote-2):

CALL n10s.graphconfig.init();

CREATE CONSTRAINT n10s\_unique\_uri ON (r:Resource) ASSERT r.uri IS UNIQUE;

The ontology to be imported in this case is developed following BFO and IOF-Core upper-level ontology. The detailed development method is introduced in a previous paper[[3]](#footnote-3). A draft version of the ontology has been uploaded to GitHub for testing. It can be retrieved with this link:

<https://raw.githubusercontent.com/zhengxiaochen/ontology_aircraft_system/main/Aircraft_assembly_process_ontology.ttl>

## Import ontology

1. Change the handleVocabUris property to ‘MAP’ to avoid errors when exporting graph[[4]](#footnote-4).

CALL n10s.graphconfig.init({ handleVocabUris: 'handleVocabUris' });

1. Import ontology:

CALL n10s.onto.import.fetch( "https://raw.githubusercontent.com/zhengxiaochen/ontology\_aircraft\_system/main/Aircraft\_assembly\_process\_ontology.ttl", "Turtle",{ handleVocabUris: "MAP" })

1. Add new prefixes to map the relationships with proper names:

CALL n10s.nsprefixes.add("prefix", "neo4j://orbitaljoint#");

call n10s.mapping.add("neo4j://orbitaljoint#/process", "Process");

call n10s.mapping.add("neo4j://orbitaljoint#/operation", "Operation");

call n10s.mapping.add("neo4j://orbitaljoint#/haspredecessor", "hasPredecessor");

call n10s.mapping.add("neo4j://orbitaljoint#/requiresresource", "requiresResource");

call n10s.mapping.add("neo4j://orbitaljoint#/isindividualof", "isIndividualOf");

call n10s.mapping.add("neo4j://orbitaljoint#/issubclassof", "isSubclassOf");

call n10s.mapping.add("neo4j://orbitaljoint#/hasEssentialOperation", "hasEssentialOperation");

call n10s.mapping.add("neo4j://orbitaljoint#/hasSubprocess", "hasSubprocess");

call n10s.mapping.add("neo4j://orbitaljoint#/hasOptionalOperation", "hasOptionalOperation");

call n10s.mapping.add("neo4j://orbitaljoint#/hasOperation", "hasOperation");

call n10s.mapping.add("neo4j://orbitaljoint#/hasOptionalAutoOperation", "hasOptionalAutoOperation");

call n10s.mapping.add("neo4j://orbitaljoint#/hasOptionalManualOperation", "hasOptionalManualOperation");

1. Set label for visualizing classes and rename Resource name to n4sch\_\_Class:

MATCH (n:Resource) SET n.n4sch\_\_label=n.label RETURN n;

MATCH (n:Resource) WITH collect(n) AS n1

CALL apoc.refactor.rename.label("Resource", "n4sch\_\_Class", n1)

YIELD committedOperations

RETURN \*;

1. Object properties (relationships) of the ontology are imported as Nodes. We need to update the relationships to properly show the relationship names

MATCH (p:Class)-[r]-(q:Class) WITH DISTINCT r MATCH (cr:Relationship) WHERE cr.uri CONTAINS r.onPropertyURI SET r.onPropertyName=cr.label;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasEssentialOperation"

CREATE (p)-[:hasEssentialOperation]->(q);

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasEssentialOperation"

DETACH DELETE r;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasPredecessor"

CREATE (p)-[:hasPredecessor]->(q);

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasPredecessor"

DETACH DELETE r;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "requiresResource"

CREATE (p)-[:requiresResource]->(q) ;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "requiresResource"

DETACH DELETE r;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasSubprocess"

CREATE (p)-[:hasSubprocess]->(q) ;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasSubprocess"

DETACH DELETE r;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasOptionalOperation"

CREATE (p)-[:hasOptionalOperation]->(q);

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasOptionalOperation"

DETACH DELETE r;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasOptionalAutoOperation"

CREATE (p)-[:hasOptionalAutoOperation]->(q);

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasOptionalAutoOperation"

DETACH DELETE r;

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasOptionalManualOperation"

CREATE (p)-[:hasOptionalManualOperation]->(q);

MATCH (p:Class)-[r:SCO\_RESTRICTION]->(q:Class) WHERE r.onPropertyName CONTAINS "hasOptionalManualOperation"

DETACH DELETE r;

MATCH (op:n4sch\_\_Class) SET op.name=op.n4sch\_\_label; //#SET all operations labels = name

The ontology should be imported correctly with all the names properly visualized. Step 3 and 4 will need to be updated with corresponding relationship names or class names if a different ontology is imported.

## Edit ontology to update class properties corresponding to a dummy dataset

A dummy dataset is provided representing some typical operations of an aircraft Orbital Junction Process as shown below. These operations have been defined in the imported ontology. Neo4j allows to edit the ontology to update properties and relationships of classes using Cypher querying. All the information indicated in the dataset can be added to the ontology with the following codes:

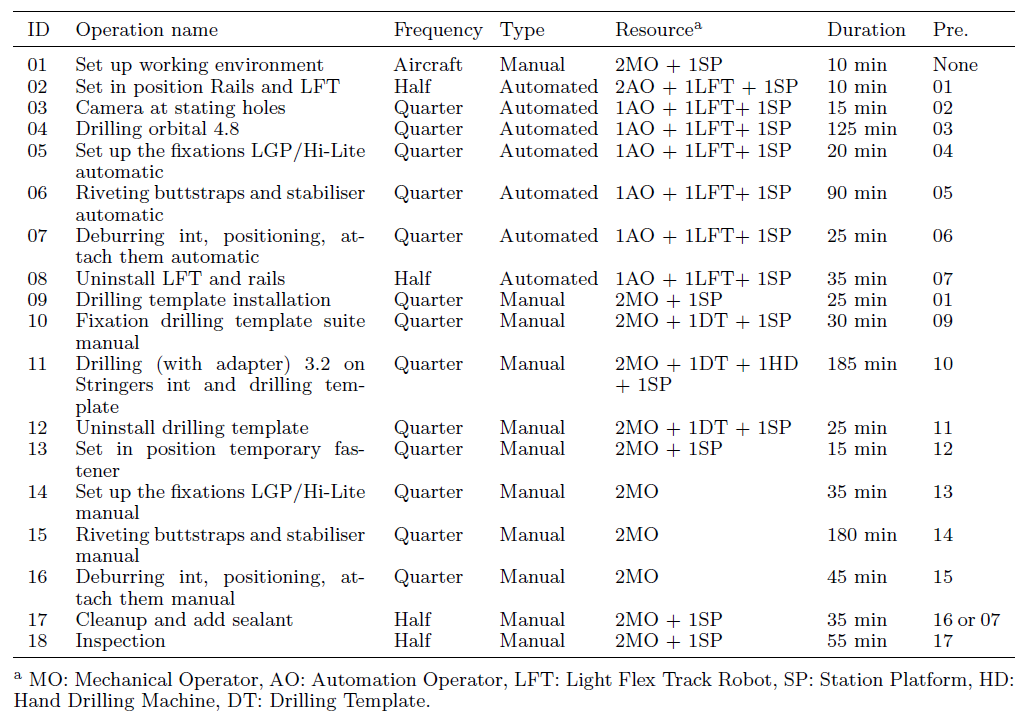


Figure Some typical operations of an aircraft Orbital Junction Process

1. Update Operation data properties: *duration, op\_type*

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_00001\_Jig in'}) SET p.duration = 60, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_00002\_Jig out'}) SET p.duration = 60, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_01001\_Set up working environment'}) SET p.duration = 10, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_02001\_Set in position Rails and LFT'}) SET p.duration = 10, p.op\_type = "Auto";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04001\_Camera at stating holes'}) SET p.duration = 15, p.op\_type = "Auto";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04002\_Drilling orbital 4,8'}) SET p.duration = 125, p.op\_type = "Auto";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04008\_Set up the fixations LGP/Hi-Lite automatic'}) SET p.duration = 20, p.op\_type = "Auto";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04010\_Riveting buttstraps and stabiliser automatic'}) SET p.duration = 90, p.op\_type = "Auto";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04012\_Deburring int, positioning, attach them automatic'}) SET p.duration = 25, p.op\_type = "Auto";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04014\_Deinstall LFT and rails'}) SET p.duration = 35, p.op\_type = "Auto";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04003\_Drilling template install'}) SET p.duration = 25, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04004\_Fixation drilling template suite manual'}) SET p.duration = 30, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04005\_Drilling (with adapter) 3,2 on Stringers int and drilling template'}) SET p.duration = 185, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04006\_Deinstall drilling template'}) SET p.duration = 25, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04007\_Set in position temporary fastener'}) SET p.duration = 15, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04009\_Set up the fixations LGP/Hi-Lite manual'}) SET p.duration = 35, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04011\_Riveting buttstraps and stabiliser manual'}) SET p.duration = 180, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_04013\_Deburring int, positioning, attach them manual'}) SET p.duration = 45, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_02002\_Cleanup and add sealant'}) SET p.duration = 35, p.op\_type = "Manual";

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'S40\_02003\_Inspection'}) SET p.duration = 55, p.op\_type = "Manual";

1. Update Resource data properties: *cost\_hour, calendar, number*

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Light Flex Track Robot'}) SET p.cost\_hour= 75, p.calendar= "24x7", p.number = 2;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Light Flex Track Rail'}) SET p.cost\_hour= 5, p.calendar= "24x7", p.number = 2;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Station'}) SET p.cost\_hour= 80, p.calendar= "24x7", p.number = 2;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Station platform'}) SET p.cost\_hour= 75, p.calendar= "24x7", p.number = 2;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Mechanical Operator'}) SET p.cost\_hour= 100, p.calendar= "shift\_40h\_week", p.number = 8;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Automation Operator'}) SET p.cost\_hour= 100, p.calendar= "shift\_40h\_week", p.number = 8;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Hand drilling machine'}) SET p.cost\_hour= 10, p.calendar= "24x7", p.number = 6;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Drilling template'}) SET p.cost\_hour= 10, p.calendar= "24x7", p.number = 5;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Crane'}) SET p.cost\_hour= 30, p.calendar= "24x7", p.number = 1;

MATCH (p:n4sch\_\_Class {n4sch\_\_label: 'Transportation tooling'}) SET p.cost\_hour= 80, p.calendar= "24x7", p.number = 1;

1. Update required number of Resource for each Operation

MATCH (p:n4sch\_\_Class)-[r:requiresResource]->(o:n4sch\_\_Class) WHERE p.name STARTS WITH 'S40\_0'

DETACH DELETE r;

MATCH (o19:n4sch\_\_Class),(o20:n4sch\_\_Class),(o1:n4sch\_\_Class), (o2:n4sch\_\_Class), (o3:n4sch\_\_Class), (o4:n4sch\_\_Class), (o5:n4sch\_\_Class), (o6:n4sch\_\_Class), (o7:n4sch\_\_Class), (o8:n4sch\_\_Class), (o9:n4sch\_\_Class), (o10:n4sch\_\_Class), (o11:n4sch\_\_Class), (o12:n4sch\_\_Class), (o13:n4sch\_\_Class), (o14:n4sch\_\_Class), (o15:n4sch\_\_Class), (o16:n4sch\_\_Class), (o17:n4sch\_\_Class), (o18:n4sch\_\_Class), (r11:n4sch\_\_Class { name: "Mechanical Operator" }),(r12:n4sch\_\_Class { name: "Automation Operator" }),(r2:n4sch\_\_Class { name: "Light Flex Track Robot" }),(r3:n4sch\_\_Class { name: "Light Flex Track Rail" }),(r4:n4sch\_\_Class { name: "Hand drilling machine" }),(r5:n4sch\_\_Class { name: "Drilling Template" }),(r6:n4sch\_\_Class { name: "Station platform" }),(r7:n4sch\_\_Class { name: "Station" }),(r8:n4sch\_\_Class { name: "Crane" }),(r9:n4sch\_\_Class { name: "Transportation tooling" })

WHERE o1.name STARTS WITH 'S40\_01001' AND o2.name STARTS WITH 'S40\_02001' AND o3.name STARTS WITH 'S40\_04001' AND o4.name STARTS WITH 'S40\_04002' AND o5.name STARTS WITH 'S40\_04003' AND o6.name STARTS WITH 'S40\_04004' AND o7.name STARTS WITH 'S40\_04005' AND o8.name STARTS WITH 'S40\_04006' AND o9.name STARTS WITH 'S40\_04007' AND o10.name STARTS WITH 'S40\_04008' AND o11.name STARTS WITH 'S40\_04009' AND o12.name STARTS WITH 'S40\_04010' AND o13.name STARTS WITH 'S40\_04011' AND o14.name STARTS WITH 'S40\_04012' AND o15.name STARTS WITH 'S40\_04013' AND o16.name STARTS WITH 'S40\_04014' AND o17.name STARTS WITH 'S40\_02002' AND o18.name STARTS WITH 'S40\_02003' AND o19.name STARTS WITH 'S40\_00001' AND o20.name STARTS WITH 'S40\_00002'

CREATE (o1)-[:requiresResource {number: 1}]->(r11), (r6)<-[:requiresResource {number: 1}]-(o1)-[:requiresResource {number:1}]->(r7),

(o2)-[:requiresResource {number:2}]->(r12), (r2)<-[:requiresResource {number: 1}]-(o2)-[:requiresResource {number:1}]->(r3), (r6)<-[:requiresResource {number: 1}]-(o2)-[:requiresResource {number:1}]->(r7),

(o3)-[:requiresResource {number:1}]->(r12), (r2)<-[:requiresResource {number: 1}]-(o3)-[:requiresResource {number:1}]->(r3), (r6)<-[:requiresResource {number: 1}]-(o3)-[:requiresResource {number:1}]->(r7),

(o4)-[:requiresResource {number:1}]->(r12), (r2)<-[:requiresResource {number: 1}]-(o4)-[:requiresResource {number:1}]->(r3), (r6)<-[:requiresResource {number: 1}]-(o4)-[:requiresResource {number:1}]->(r7),

(o5)-[:requiresResource {number:2}]->(r11), (r6)<-[:requiresResource {number: 1}]-(o5)-[:requiresResource {number:1}]->(r7),

(o6)-[:requiresResource {number:2}]->(r11), (o6)-[:requiresResource {number:1}]->(r5),(r6)<-[:requiresResource {number: 1}]-(o6)-[:requiresResource {number:1}]->(r7),

(o7)-[:requiresResource {number:2}]->(r11), (r4)<-[:requiresResource {number:2}]-(o7)-[:requiresResource {number:1}]->(r5),(r6)<-[:requiresResource {number: 1}]-(o7)-[:requiresResource {number:1}]->(r7),

(o8)-[:requiresResource {number:2}]->(r11), (o8)-[:requiresResource {number:1}]->(r5),(r6)<-[:requiresResource {number: 1}]-(o8)-[:requiresResource {number:1}]->(r7),

(o9)-[:requiresResource {number:2}]->(r11), (r6)<-[:requiresResource {number: 1}]-(o9)-[:requiresResource {number:1}]->(r7),

(o10)-[:requiresResource {number:1}]->(r12), (r2)<-[:requiresResource {number: 1}]-(o10)-[:requiresResource {number:1}]->(r3), (r6)<-[:requiresResource {number: 1}]-(o10)-[:requiresResource {number:1}]->(r7),

(o12)-[:requiresResource {number:1}]->(r12), (r2)<-[:requiresResource {number: 1}]-(o12)-[:requiresResource {number:1}]->(r3), (r6)<-[:requiresResource {number: 1}]-(o12)-[:requiresResource {number:1}]->(r7),

(o14)-[:requiresResource {number:1}]->(r12), (r2)<-[:requiresResource {number: 1}]-(o14)-[:requiresResource {number:1}]->(r3), (r6)<-[:requiresResource {number: 1}]-(o14)-[:requiresResource {number:1}]->(r7),

(o16)-[:requiresResource {number:1}]->(r12), (r2)<-[:requiresResource {number: 1}]-(o16)-[:requiresResource {number:1}]->(r3), (r6)<-[:requiresResource {number: 1}]-(o16)-[:requiresResource {number:1}]->(r7),

(o11)-[:requiresResource {number:2}]->(r11),

(o13)-[:requiresResource {number:2}]->(r11),

(o15)-[:requiresResource {number:2}]->(r11),

(o17)-[:requiresResource {number:2}]->(r11), (r6)<-[:requiresResource {number: 1}]-(o17)-[:requiresResource {number:1}]->(r7),

(o18)-[:requiresResource {number:2}]->(r11), (r6)<-[:requiresResource {number: 1}]-(o18)-[:requiresResource {number:1}]->(r7), (r8)<-[:requiresResource {number: 1}]-(o19)-[:requiresResource {number:1}]->(r9), (r8)<-[:requiresResource {number: 1}]-(o20)-[:requiresResource {number:1}]->(r9);

Now all the information of the dummy dataset is stored in the ontology. We can use this ontology to generate some Orbital Junction Process alternatives based on certain rules.

## Knowledge instantiating to create new Orbital Junction Process individuals

The example below shows how to generate an Orbital Junction Process: name starts with ‘N01’, 1/2 AUTO concurrent 1/2 AUTO.

1. Create new process and add essential operations

//## structure of the process

//MATCH (op) WHERE op.name STARTS WITH 'N1' DETACH DELETE op;

//MATCH (op),(ob) WHERE op.name STARTS WITH 'N01' AND ob.name STARTS WITH "R01" DETACH DELETE op, ob, res;

MATCH (p:n4sch\_\_Class{n4sch\_\_label: 'S40\_Orbital Joint Process'})

CREATE (np:Process{name: "R01" + p.n4sch\_\_label}), (np)-[:isIndividualOf]->(p)

WITH \* MATCH (p)-[:hasEssentialOperation]->(Eop:n4sch\_\_Class)

WITH DISTINCT Eop,np,p CREATE (nEop :Operation{name: "N01" + Eop.n4sch\_\_label}), (nEop)-[:isIndividualOf]->(Eop), (np)-[:hasOperation]->(nEop)

WITH \* MATCH (Eop)-[:hasPredecessor]->(c1)<-[:isIndividualOf]-(pr1)

MERGE (nEop)-[:hasPredecessor]->(pr1)

RETURN \* //3 EssentialOperation:Jig in, Jig out, set environment

//Subprocess EssentialOperation

MATCH (p:n4sch\_\_Class{n4sch\_\_label: 'S40\_Orbital Joint Process'})-[:hasSubprocess]->(sc:n4sch\_\_Class)-[:hasEssentialOperation]->(e:n4sch\_\_Class)//sc:upper and lower half

WITH DISTINCT e

CREATE (oe1:Operation{name: "N01" + e.n4sch\_\_label+"\_1"})-[:isIndividualOf]->(e), (oe2:Operation{name: "N01" + e.n4sch\_\_label+"\_2"})-[:isIndividualOf]->(e)

WITH \* MATCH (e)-[:hasPredecessor]->(c1)<-[:isIndividualOf]-(pr1), (c1)<-[:isIndividualOf]-(pr2)

WHERE pr1.name ENDS WITH '\_1' AND pr2.name ENDS WITH '\_2' AND pr1.name STARTS WITH 'N01' AND pr2.name STARTS WITH 'N01'

MERGE (oe1)-[:hasPredecessor]->(pr1) MERGE (oe2)-[:hasPredecessor]->(pr2)

RETURN \*

1. Add Subprocess Automated

MATCH (p:n4sch\_\_Class{n4sch\_\_label: 'S40\_Orbital Joint Process'})-[:hasSubprocess]->(sc:n4sch\_\_Class)-[:hasOptionalAutoOperation]->(Oop:n4sch\_\_Class)

WITH DISTINCT Oop CREATE (nOop:Operation{name: 'N01' + Oop.n4sch\_\_label})-[:isIndividualOf]->(Oop)

WITH \* MATCH (p:n4sch\_\_Class{n4sch\_\_label: 'S40\_Orbital Joint Process'})-[:hasSubprocess]->(sc)-[:hasSubprocess]->(ssc)-[:hasOptionalAutoOperation]->(Oop1)

WITH DISTINCT Oop1 CREATE (nOop1:Operation{name: 'N01' + Oop1.n4sch\_\_label + '\_1'})-[:isIndividualOf]->(Oop1), (nOop2:Operation{name: 'N01' + Oop1.n4sch\_\_label + '\_2'})-[:isIndividualOf]->(Oop1)

WITH \* MATCH (Oop1)-[:hasPredecessor]->(c1)<-[:isIndividualOf]-(pr1), (c1)<-[:isIndividualOf]-(pr2) WHERE pr1.name ENDS WITH '\_1' AND pr2.name ENDS WITH '\_2' AND pr1.name STARTS WITH 'N01' AND pr2.name STARTS WITH 'N01'

MERGE (nOop1)-[:hasPredecessor]->(pr1) MERGE (nOop2)-[:hasPredecessor]->(pr2)

WITH \* MATCH (p1:Operation), (p2:Operation), (p3:Operation), (p4:Operation), (p5:Operation), (p6:Operation) WHERE p1.name CONTAINS 'N01Camera at stating holes\_1' AND p2.name CONTAINS 'N01Deburring int, positioning, attach them automatic\_1' AND p3.name CONTAINS 'N01Camera at stating holes\_2' AND p4.name CONTAINS 'N01Deburring int, positioning, attach them automatic\_2' AND p5.name CONTAINS 'N01Set in position Rails and LFT' AND p6.name CONTAINS 'N01Deinstall LFT and rails'

MERGE (p1)-[:hasPredecessor]->(p5) MERGE (p3)-[:hasPredecessor]->(p2) MERGE (p6)-[:hasPredecessor]->(p4)

RETURN \*

1. Subprocess Manual Sequential

MATCH (p:n4sch\_\_Class{n4sch\_\_label: 'S40\_Orbital Joint Process'})-[:hasSubprocess]->(sc)-[:hasSubprocess]->(ssc)-[:hasOptionalManualOperation]->(Oop1)

WITH DISTINCT Oop1 CREATE (nOop1:Operation{name: 'N01' + Oop1.n4sch\_\_label + '\_1'})-[:isIndividualOf]->(Oop1), (nOop2:Operation{name: 'N01' + Oop1.n4sch\_\_label + '\_2'})-[:isIndividualOf]->(Oop1)

WITH \* MATCH (Oop1)-[:hasPredecessor]->(c1)<-[:isIndividualOf]-(pr1), (c1)<-[:isIndividualOf]-(pr2) WHERE pr1.name ENDS WITH '\_1' AND pr2.name ENDS WITH '\_2' AND pr1.name STARTS WITH 'N01' AND pr2.name STARTS WITH 'N01'

MERGE (nOop1)-[:hasPredecessor]->(pr1) MERGE (nOop2)-[:hasPredecessor]->(pr2)

WITH \* MATCH (p1:Operation), (p2:Operation),(p0:Operation), (p3:Operation) WHERE p1.name CONTAINS 'N01Deburring int, positioning, attach them manual\_1' AND p2.name CONTAINS 'N01Drilling template install\_2' AND p0.name CONTAINS 'N01Drilling template install\_1' AND p3.name CONTAINS 'N01Deburring int, positioning, attach them manual\_2'

MERGE (p2)-[:hasPredecessor]->(p1)

RETURN \*

1. Connect nodes

//Connect nodes sub1

MATCH (p1:Operation), (p2:Operation) WHERE p1.name CONTAINS 'N01S40\_04012\_Deburring int, positioning, attach them automatic\_1' AND p2.name CONTAINS 'N01S40\_04001\_Camera at stating holes\_2' MERGE (p2)-[:hasPredecessor]->(p1);

MATCH (p1:Operation), (p2:Operation) WHERE p1.name CONTAINS 'N01S40\_02001\_Set in position Rails and LFT' AND p2.name CONTAINS 'N01S40\_04001\_Camera at stating holes\_1' MERGE (p2)-[:hasPredecessor]->(p1);

MATCH (p1:Operation), (p2:Operation) WHERE p1.name CONTAINS 'N01S40\_04012\_Deburring int, positioning, attach them automatic\_2' AND p2.name CONTAINS 'N01S40\_04014\_Deinstall LFT and rails' MERGE (p2)-[:hasPredecessor]->(p1);

MATCH (p1:Operation), (p2:Operation) WHERE p1.name CONTAINS 'N01S40\_04014\_Deinstall LFT and rails' AND p2.name CONTAINS 'N01S40\_02002\_Cleanup and add sealant\_1' MERGE (p2)-[:hasPredecessor]->(p1);

//Connect nodes sub2

MATCH (p1:Operation), (p2:Operation) WHERE p1.name CONTAINS 'N01S40\_04013\_Deburring int, positioning, attach them manual\_1' AND p2.name CONTAINS 'N01S40\_04003\_Drilling template install\_2' MERGE (p2)-[:hasPredecessor]->(p1);

MATCH (p1:Operation), (p2:Operation) WHERE p1.name CONTAINS 'N01S40\_04013\_Deburring int, positioning, attach them manual\_2' AND p2.name CONTAINS 'N01S40\_02002\_Cleanup and add sealant\_2' MERGE (p2)-[:hasPredecessor]->(p1);

//Connect main and 2 sub1

MATCH (p1:Operation), (p2:Operation), (p3:Operation) WHERE p1.name CONTAINS 'N01S40\_02003\_Inspection\_1' AND p2.name CONTAINS ' N01S40\_02003\_Inspection\_2' AND p3.name CONTAINS 'N01S40\_00002\_Jig out' MERGE (p1)<-[:hasPredecessor]-(p3)-[:hasPredecessor]->(p2);

MATCH (p1:Operation), (p2:Operation), (p3:Operation) WHERE p1.name CONTAINS 'N01S40\_02001\_Set in position Rails and LFT' AND p2.name CONTAINS 'N01S40\_04003\_Drilling template install\_1' AND p3.name CONTAINS 'N01S40\_01001\_Set up working environment' MERGE (p1)-[:hasPredecessor]->(p3)<-[:hasPredecessor]-(p2);

1. Add resources and duration properties

MATCH (op:Operation)-[:isIndividualOf]->(cl:n4sch\_\_Class)-[:requiresResource]->(res:n4sch\_\_Class) WHERE op.name STARTS WITH 'N01'

WITH DISTINCT res CREATE (rob:Resource{name: 'N01' + res.n4sch\_\_label, n4sch\_\_label: 'N01' + res.n4sch\_\_label})-[:isIndividualOf]->(res)

WITH \* MATCH (op:Operation)-[:isIndividualOf]->(cl:n4sch\_\_Class)-[r]->(res:n4sch\_\_Class)<-[:isIndividualOf]-(rob) WHERE op.name STARTS WITH 'N01'

CREATE (op)-[:requiresResource{number: r.number}]->(rob);

MATCH (op:Operation)-[:isIndividualOf]->(cl:n4sch\_\_Class) WHERE op.name STARTS WITH 'N01'

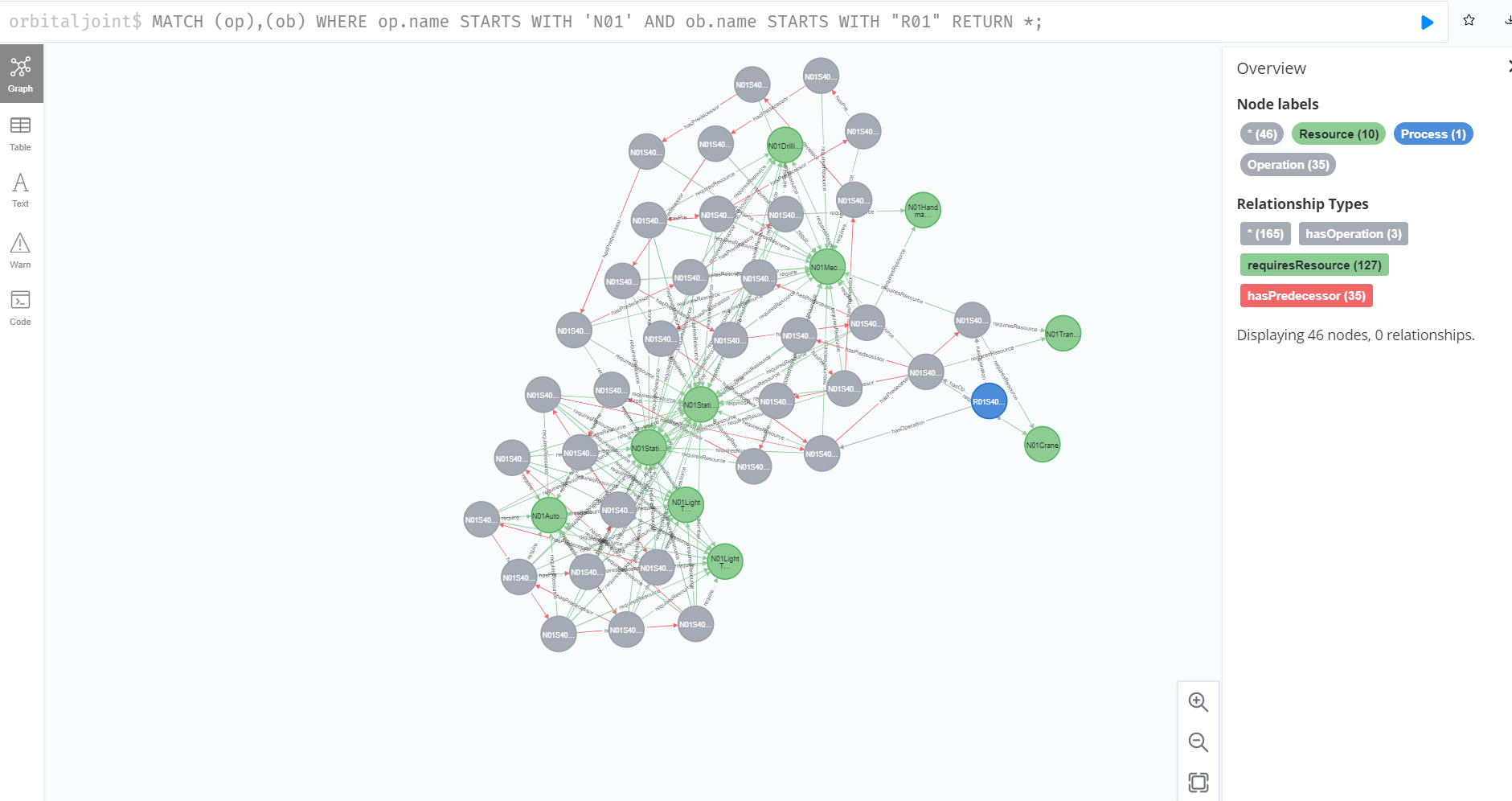
SET op.duration = cl.duration, op.op\_type=cl.op\_type

A new Orbital Junction Process has been created. We can query it to see how it looks like:

//Displaying 68 nodes, 328 relationships.

MATCH (operation)-[Relationship]-(entity) WHERE operation.name STARTS WITH 'N01'

RETURN \*;



It is also possible to export the query results in RDF format[[5]](#footnote-5):

:POST http://localhost:7474/rdf/orbitaljoint/cypher

{ "cypher" : "MATCH (operatoin)-[relationship]-(entity) WHERE operatoin.name STARTS WITH 'N1' RETURN \*", "format": "Turtle" }

It is also possible to make the above process generating automatic by embedding Cypher codes into a Python script, which is available in GitHub[[6]](#footnote-6).

All the Cypher code mentioned above is also available in a clean .cpy file without explanations[[7]](#footnote-7).

1. <https://www.researchgate.net/publication/355889982_Neo4j_graph_database_installation_on_Azure_cloud_Ubuntu_Virtual_Machine_and_configuration_of_Neosemantics_package_for_Ontology> [↑](#footnote-ref-1)
2. <https://neo4j.com/labs/neosemantics/4.2/config/> [↑](#footnote-ref-2)
3. Zheng, Xiaochen, et al. "Development of an Application Ontology for Knowledge Management to Support Aircraft Assembly System Design." Proceedings http://ceur-ws. org ISSN 1613 (2020): 0073. [↑](#footnote-ref-3)
4. <https://community.neo4j.com/t/problem-exporting-rdf-using-neosemantics-n10s/20880/9> [↑](#footnote-ref-4)
5. <https://neo4j.com/labs/neosemantics/4.0/export/> [↑](#footnote-ref-5)
6. <https://github.com/zhengxiaochen/ontology_aircraft_system/blob/main/auto_generate_process.py> [↑](#footnote-ref-6)
7. <https://github.com/zhengxiaochen/ontology_aircraft_system/blob/main/CYPHER_import_edit_ontology.cyp> [↑](#footnote-ref-7)