

# SUGOI Software Used to Gain Ontology Interchangeability Algorithm

Zubeida Casmod Khan<sup>1,2</sup> and C. Maria Keet<sup>1</sup>

<sup>1</sup> Department of Computer Science, University of Cape Town, South Africa  
mkeet@cs.uct.ac.za

<sup>2</sup> Council for Scientific and Industrial Research, Pretoria, South Africa  
zkhan@csir.co.za

---

**Algorithm 1:** SUGOI ontology interchangeability algorithm

---

```
input : sourceOntology, sourceFoundationalOntology,  
        targetFoundationalOntology mapOntology  
output: targetOntology  
  
/* Steps 1-2. Create an ontology targetOntology. Copy axioms from the  
   target foundational ontology to the targetOntology */  
1 targetOntology ← targetFoundationalOntology;  
/* Step 3. Copy domain axioms to the target ontology */  
2 foreach entity in sourceOntology do  
3   | if entity not in sourceFoundationalOntology then  
4   |   | currentAxiom ← get current axiom;  
5   |   | add currentAxiom to targetOntology;  
6   |   end  
7 end  
/* Step 4. Map domain entities to the target foundational ontology */  
8 foreach entity in targetOntology do  
   | /* if entity is a domain entity */  
9   | if entity not in sourceFoundationalOntology and entity not in  
   | targetFoundationalOntology then  
10  |   | currentAxiom ← get current axiom;  
11  |   | entitySet ← get entities in signature of currentAxiom;  
12  |   | foreach signatureEntity in entitySet do  
13  |   |   | if signatureEntity in mapOntology then  
14  |   |   |   | eSignatureEntity ← get equivalent entity of signatureEntity;  
15  |   |   |   | currentAxiom ← replace signatureEntity with  
16  |   |   |   | eSignatureEntity in currentAxiom;  
16  |   |   | end  
17  |   |   end  
18  |   end  
19 end
```

---

---

```

20   /* Step 5. Perform on-the-fly subsumption, if a domain entity from
21      previous step is not linked to target foundational ontology */
22   foreach entity in targetOntology do
23     if entity not in targetFoundationalOntology then
24       if entity has no superclasses in targetOntology then
25         ancestorSet ← get ancestor entities of entity from
26           sourceFoundationalOntology;
27         mappableSet ← empty set;
28         foreach ancestorEntity in ancestorSet do
29           if ancestorEntity exists in mapOntology then
30             add ancestorEntity to mappableSet;
31           end
32         end
33         /* get lowest level entity */
34         selectedEntity ← get lowest level entity from mappableSet;
35         mappedSelectedEntity ← get entity equivalent to selectedEntity
36           from mapOntology;
37         newAxiom ← create axiom stating that entity is a subclass of
38           mappedSelectedEntity;
39         add newAxiom to targetOntology;
40       end
41     end
42   end
43   /* Step 6. Delete source foundational ontology entities that are not
44      referenced by the domain entities */
45   foreach entity in targetOntology do
46     if entity in sourceFoundationalOntology then
47       entitySet ← get referencing entities of entity;
48       foreach referencedEntity in entitySet do
49         /* if referencedEntity is a domain entity */
50         if referencedEntity not in sourceFoundationalOntology and
51           referencedEntity not in targetFoundationalOntology then
52           checker ← true;
53         end
54       end
55       /* if entity is not referenced by any domain entities */
56       if checker == false then
57         if entity in targetFoundationalOntology then
58           remove entity from targetOntology;
59         end
60       end
61     end
62   end
63   save targetOntology;
64   generate log file with metrics;
65   if user agrees then
66     save sourceOntology, targetOntology to server;
67   end

```

---