# A Rule-based Method for Extracting RDF(S) and OWL Sub-ontologies

Sajjad Hussain and Syed Sibte Raza Abidi

NICHE Research Group, Faculty of Computer Science, Dahousie University, Canada

**Abstract.** Sub-Ontology extraction, from a large ontology, leads to the generation of a specialized knowledge model that is pertinent to specific problems. Existing sub-ontology extraction methods tend to either render a too generalized or a too restricted sub-ontology that at times does not capture the entire semantics of the parent ontology. We present a sub-ontology extraction method that using N3 rules to extract a sub-ontology from RDF(S) and OWL ontologies whilst extending the semantics of the extracted concepts and their relationships in the sub-ontology. Our approach features the following tenets (i) identifying the user-selected concepts; (ii) extracting the user-selected concepts, their roles and their individuals; and (iii) extracting other concepts, roles and individuals that are *immediate structurally-connected* with the user-selected concepts. As a test-case, we present the sub-ontology extraction outcome for a healthcare ontology representing Prostate Cancer care management.

## 1 Introduction

Ontologies [1] serve as the backbone for the Semantic Web as they provide formal constructs to model a domain along shared concepts and their relationships. A key aspect of Semantic Web research is knowledge modeling that is pursued by developing a domain-specific ontology. The ontology, then, serves as the foundational representation formalism for a variety of activities, such as web-based information sharing, retrieval, mediation, collaboration and decision support. Knowledge modelers, or ontology engineers, develop domain ontologies in a variety of ways–i.e. developing a new ontology by abstracting domain concepts and relations, adding new concepts to an existing ontology or even aligning/merging multiple existing ontologies to realize a specialized ontology. For mature domains–such as medicine, genetics and business–the common practice of researchers is to leverage large-scale domain ontologies in order to systematically select a sub-set of concepts or a specialized 'sub-ontology' that is representative of the task at hand. A sub-ontology presents a focused representation, at a desired level of abstraction, of selected aspects of the principal ontology whilst offering the usual operational processing and knowledge constraints associated with ontologies. The rationale for sub-ontology extraction is that working with the larger ontology leads to the introduction of irrelevant concepts to the required knowledge model for a specialized problem, which

in turn not only extends the deductive closure of the model to undesired interpretations but also compromises the efficiency, complexity and focus of the knowledge model. Therefore, for solving specialized problem in large domains there is a need to extract a sub-ontology that offers a consistent fragment of the source ontology. In this paper, we explore the problem of sub-ontology extraction and present a rule-based method for sub-ontology extraction for RDF(S) and OWL ontologies.

In this paper, we present our approach for sub-ontology extraction, from both RDF(S) and OWL-DL ontologies, that use N3 rules [2] to infer a controlled closure of RDF(S) and OWL ontologies under their complete inference rules [3]. Based on the defined semantics in [3], we have defined N3 rules to extract the axioms and assertions for a selected concept (and its associated concepts, roles and individuals) in RDF(S) and OWL ontologies. These rules can be used by various Semantic Web reasoning engines, we use Euler inference engine [4] to infer interpretations beyond basic ontological structures to extract a more semantically-rich sub-ontology. In this way, we argue that our approach extends the semantics of the sub-ontology compared to the semantics offered by existing sub-ontology extraction approaches [5–9]. We have applied our sub-ontology extraction method for the problem of knowledge morphing [10] in the healthcare domain, where we extract sub-ontology from a Prostate Cancer ontology [11]. Finally, we present a comparison of our approach with existing sub-ontology extraction approaches.

## 2 Sub-Ontology Extraction: Overview and Approaches

Sub-ontology extraction is guided by a particular 'context' that leads to the selection, adaptation, re-use or reconciliation of concepts from a source ontology to yield a sub-ontology that reflects the identified and inferred correspondences between the user-defined concepts and the concepts modeled in the ontology [12]. However, the challenge is to ensure that the generated sub-ontology does not lead to (i) differences or errors in domain conceptualization, (ii) logical contradictions, and (iii) inaccurate realizations of the concepts and their roles as per context, as these issues render the sub-ontology to become inconsistent [13]. In this case, it is important that the extracted sub-ontology is semantically consistent. To extract a sub-ontology, the source ontology is considered as an un-directed graph [5–9], and the extracted sub-ontology is a sub-graph that describes the knowledge about the selected concepts as per the context.

One of the initial attempts at sub-ontology extraction is the Materialized Ontology View Extraction (MOVE) process [5, 6], where a sub-ontology is extracted based on the *Requirements Consistency Optimization Scheme* (RCOS) and *Semantic Completeness Optimization Scheme* (SCOS) [14]. Given a source ontology, MOVE allows the user to label both concepts and roles as *selected* or *un-selected*. RCOS is applied to extract triples for a selected/un-selected role $R$ as follows: (i) extract the concept axioms for $R$; (ii) extract the role assertions for $R$ and the concept axioms for concepts that are associated with $R$; (iii) for an

un-selected role assertion, do not extract its role axioms; and (iv) $R$ must be connect with a concept in the extracted sub-ontology. SCOS is applied to preserve the defined semantic completeness for a selected concept $C$ as follows: (a) the super-concepts of $C$ and their associated roles must be selected; (b) the sub-concepts of $C$ and their associated roles must be selected; (c) the roles in $C$ that require a minimum cardinality greater than zero, and their role assertions must be selected. It may be noted that when the MOVE approach is applied to a connected graph (of an ontology), where there exists a path between every two nodes, due to the SCOS criterion the extracted sub-graph can potentially result in the extraction of the entire graph itself (i.e. the whole ontology). This leads to the potential extension of the deductive closure of the extracted sub-ontology to include more generalized concepts relative to the desired specialized focus of the sub-ontology. Therefore, there is a need to improve the MOVE process by introducing fine-grained filtering techniques that will limit the knowledge scope or deductive closure of the extracted sub-ontology.

Another sub-ontology extraction approach is described by Miao et. al. [7] that extracts sub-ontologies from multiple RDF(S) ontologies. Similar to MOVE, this approach allows users to either select or un-select a concept as per their context. This approach views the RDF(S) ontology as a graph and proceeds to extract a *sub-graph* from the closure of the RDF(S) ontology under given RDF(S) inference rules. Here, the approach is guided by the closure of an RDF(S) ontology and only finds a restricted sub-graph that is induced by the set of selected nodes (concepts)–it extracts only the selected nodes (concepts, roles, and individuals), and does not extract any further concepts, roles and individuals associated to the selected nodes. Therefore extracted sub-ontologies form this method are restricted compared to the sub-ontologies extracted by the MOVE. Hence, in this method the user needs to have an in-depth understanding of the ontology in order to list not only the concepts that are of user's interest, but also their roles, individuals and other related concepts.

We argue that the above-mentioned sub-ontology-extraction approaches have certain weaknesses: In MOVE, due to its SCOS criterion, the extracted sub-graph may have irrelevant knowledge about the selected concepts; whereas the extracted sub-graph by the later approach [7] is too restricted. Therefore, in our sub-ontology extraction approach we present a middle ground– i.e. extract a sub-graph that is rich enough (as compared to the Miao et. al. [7] approach) but not over-inclusive (as compared to the MOVE approach).

In our approach, we deal with the SCOS criterion differently than the existing MOVE approach. The key feature of our approach is that for a given target concept $C$ we restrict the recursive selection of the super-concepts of $C$ (as practiced by MOVE) in order to avoid the unnecessary selection of the sub-concepts of the super-concept that results in the expansion of the sub-graph to include concepts that are not relevant. However, for semantic completeness of $C$ and its sub-concepts we inherit the roles of the super-concepts and these roles are now described with $C$. In this way, we avoid a situation where the sub-ontology is too-generalized by the undesired inclusion of higher levels of concepts that

may even extend all the way to `owl:Thing`. Furthermore, we improve on the rather restricted sub-ontology extraction approach by Miao et. al. [7], by extending our approach to both RDF(S) and OWL ontologies, where our sub-ontology extraction method is applied on the closure of RDF(S) and OWL ontologies under their complete inference rules [3]. The complete set of inference rules for RDF(S) and OWL ontologies provides an extended semantics that generates an inferred ontology model that offers additional axioms or assertions about the extracted concepts, roles and individuals that are valid under the RDF(S) and OWL semantics. For example, for the asserted concept axiom `:C3 owl:intersectionOf (:C1 :C2)` in an OWL ontology that defines a complex concept `:C3` as an intersection of two other concepts `:C1` and `:C2`, based on the OWL inference rules, additional axioms `:C3 rdfs:subClassOf :C1` and `:C3 rdfs:subClassOf :C2` are inferred–that provide additional RDF(S) semantics for OWL concepts `:C1`, `:C2` and `:C3` (see Section 4.4).

## 3 Preliminaries: N3 Logic and Reasoning in Euler

For our purpose, we consider ontology as a Description Logic (DL) [1] knowledge base $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, which is comprised of terminology axioms $\mathcal{T}$ and assertional axioms $\mathcal{A}$. Terminology and assertional axioms are defined based on a structured vocabulary $\mathcal{V} = \langle \mathcal{C}, \mathcal{R} \rangle$, comprised of concepts $\mathcal{C}$ and roles $\mathcal{R}$. Terminology axioms (concept axioms and roles axioms) $\mathcal{T}$ are of the form $C \sqsubseteq D$ ($R \sqsubseteq S$) or $C \equiv D$ ($R \equiv S$) such that $C, D \in \mathcal{C}$ and $R, S \in \mathcal{R}$. For a set of individuals $\mathcal{I}$, assertional axioms (concept assertions and role assertions) $\mathcal{A}$ are defined of the form $C(a)$ or $R(b, c)$ such that $C \in \mathcal{C}$, $R \in \mathcal{R}$, $a, b, c \in \mathcal{I}$. The semantics of an ontology is defined by an *interpretation* that provides semantic mapping for (i) ontology individuals, (ii) ontology concepts and (iii) ontology roles as (a) elements of the domain, (b) subsets of the domain and (c) binary relations on the domain, respectively. A *model* of an ontology is such an interpretation, under which all axioms of the ontology are satisfied. An ontology is called *consistent*, iff there exists a model for it. An ontology that has no model is called an *inconsistent ontology* [13].

Our sub-ontology extraction method is defined via N3 rules. N3 is a compact and readable alternative to RDF/XML syntax. N3Logic uses N3 syntax and extends RDF with a vocabulary of predicates [2]. N3Logic is defined to be a minimal extension to the RDF data model, which is used for both logic and data, and is compatible with the architectural principles of the Web. Based on its extended vocabulary, N3Logic allows rules to be integrated with RDF data model, and provides essential built-in functions that support reasoning over the Web. N3Logic maintains *monotonicity*, and performs reasoning under the assumption of *Scoped Negation As Failure* (SNAF) [15], which allows users to ask negative queries with explicit scope–without affecting the monotonicity of reasoning. For example, `log:notIncludes` (negated form of `log:includes`) is one of the available built-in functions in N3Logic that assumes SNAF.

Euler is an inference engine supporting logic-based proofs based on coherent logic [4]. It is a *backward-forward-backward* chaining reasoner enhanced with Euler path detection. The backward-forward-backward chaining is realized via (i) an underlying Prolog backward chaining, (ii) a forward meta-level reasoning, and (iii) a backward proof construction. Euler uses N3 (N3Logic) [2] for its input, perform reasoning using Prolog Coherent Logic (PCL) [16], and outputs data/proof back in N3. Euler is interoperable with W3C Cwm [2]. Euler preserves monotonicity, and through its built-in functions, it allows (negative) queries under the assumption of SNAF [15]. One of such Euler built-ins we have used in our work is `e:findall`, which determines whether a N3 formula holds under a given scope or not, and is defined (in N3 syntax) as follows:
`e:findall a rdf:Property; rdfs:domain rdf:List; rdfs:range rdf:List; :comment` "A built-in `(?SCOPE ?SPAN) e:findall (?SELECT ?WHERE ?ANSWER)`, which unifies `?ANSWER` with a list that contains all the instantiations of `?SELECT` (represented as a N3 formula) satisfying the `?WHERE` clause (represented as a N3 formula) in the `?SCOPE ?SPAN` of all asserted N3 formulae and their `log:conclusion`".

## 4 Extracting RDF(S) and OWL Sub-ontologies via N3 Rules

As per our approach we allow the user to label concepts (or roles) as selected. For a selected concept $C$, we extract a RDF-Sub-Graph [17] (i.e. sub-ontology) from RDF(S) and OWL ontologies by extracting triples that corresponds to the axioms and assertions for (i) the user-selected concept $C$ (ii) the individuals for $C$; (iii) the roles in $C$; (iv) the range-concepts for the roles in $C$; (v) the sub-concepts of $C$; (vi) the equivalent-concepts for $C$; (vii) the restrictions on $C$; (viii) complex concepts that are composed of $C$; (ix) only the roles of the super-concepts that are also associated with $C$, and (x) the super-concepts of $C$ as an RDF Blank-Nodes. Our sub-ontology extraction method for extracting sub-ontologies form RDF(S) and OWL Sub-ontologies is described in the following sub-sections.

### 4.1 Concept Selection

Let $\mathcal{SC} \subseteq \mathcal{C}$ be the set of selected concepts. In our method, selected concepts are labelled as $\mathcal{O}$ `usr:selected` $C$, where $C \in \mathcal{SC}$ is an user-selected concept for extracting a sub-ontology $\mathcal{O}$. All selected concepts and their roles are extracted ($\mathcal{O}$ `sbont:extract` $C$) to the sub-ontology $\mathcal{O}$, by the following N3 rule: `{?O usr:selected ?C} => {?O sbont:extract ?C}.` We also check whether a concept (or role) is `sbont:extract` (i.e. included in the extracted sub-ontology) or not, using the Euler built-in `e:findall` (see Section 3). If a concept (or role) is not found to be `sbont:extract`, then the concept (or role) is inferred as `sbont:noValue` under the following N3 rules:

```
{?C a rdfs:Class. (?scp 3) e:findall (? {?O sbont:extract ?C} ())} =>
{?O sbont:noValue ?C}.
{?P a owl:ObjectProperty. (?scp 3) e:findall (? {?O sbont:extract ?P} ())} =>
{?O sbont:noValue ?P}.
{?P a owl:DatatypeProperty. (?scp 3) e:findall (? {?O sbont:extract ?P} ())} =>
{?O sbont:noValue ?P}.
```

### 4.2 Identification of Selected Concepts, Sub-concepts, and their Roles

Identification of sub-concepts, roles, and role-ranges of a selected concept is done by the following N3 rules:
*Sub-concept*: All sub-concepts of a `usr:selected` concept, are also `usr:selected`: `{?O usr:selected ?C. ?C a rdfs:Class. ?S rdfs:subClassOf ?C} => {?O usr:selected ?S}`, where `rdfs:subClassOf a owl:TransitiveProperty`.
*Roles*: All roles whose domains are labelled as `usr:selected`, are inferred as `sbont:extract`: `{?O usr:selected ?C. ?C a rdfs:Class. ?P rdfs:domain ?C} => {?O sbont:extract ?P}`. All inherited roles (from the super-concepts) of a selected concept, are inferred as `sbont:extract` (see Section 4.3): `{?O usr:selected ?C. ?C a rdfs:Class. ?C rdfs:subClassOf ?C1. ?P rdfs:domain ?C1} => {?O sbont:extract ?P}`.
*Role-ranges*: All concepts that are defined as ranges for `sbont:extract` roles, are also inferred as `sbont:extract`: (i) `{?O usr:selected ?C. ?C a rdfs:Class. ?P rdfs:domain ?C. ?P rdfs:range ?C1} => {?O sbont:extract ?C1}`; (ii) `{?O usr:selected ?C. ?C a rdfs:Class. ?C rdfs:subClassOf ?D. ?P rdfs:domain ?D. ?P rdfs:range ?C1} => {?O sbont:extract ?C1}`.

### 4.3 Inheriting Roles from Super-concepts

For a selected concept $C \in \mathcal{SC}$, our method does not extract the super-concepts of $C$, rather we extract the roles that are applied to it and are inherited in $C$. In order to discuss the rationale behind this step, consider our extracted sub-ontology from the example ontology $\mathcal{O}_2$ in Example 4.1. The user is interested in a specialized concept `:C2`, but not its super-concept `:C1`–that is more general to `:C2`. Since `:C2 rdfs:subClassOf :C1`, therefore `:P1` is an inherited role in `:C2`. Although `:P1` is structurally-connected with `:C2` via its individuals (e.g. `:i2 :P1 :i4.`). However for `:P1`, including its domain concept `:C1` would result into an over-generalized extracted sub-ontology. For this, we replace the super-concepts (e.g. `:C1`) of selected concepts (e.g. `:C1`) by unique RDF blank-nodes (e.g. `_:t8 e:tuple (:C1)`), using the Euler built-in `e:tuple`. Section 4.8 and 4.7 describe the construction for such RDF blank-nodes. In Example 4.1, `_:t8` is the unique RDF blank-node for `:C1`, and is structurally-connected with `:C2` by `:C2 rdfs:subClassOf _:t8` and `:P1 rdfs:domain _:t8`.

### 4.4 Extraction of Complex Concepts

OWL constructs, such as `owl:intersectionOf` and `owl:unionOf`, allow to define complex concepts using the defined (atomic and complex) concepts. Based on the semantics of `owl:intersectionOf` and `owl:unionOf` constructs described in OWL [3], we define the following rules to describe such complex concepts in RDF(S) semantics:
(a) `{?C owl:intersectionOf ?L. ?L a rdf:List. ?X list:in ?L} => {?C rdfs:subClassOf ?X}.`
(b) `{?C owl:unionOf ?L. ?L a rdf:List. ?X list:in ?L} => {?X rdfs:subClassOf ?C}.`

Using the above rules, complex OWL concepts (that are either nested union or intersection of concepts) can be defined using the `rdfs:subClassOf` construct. In the example ontology $\mathcal{O}_2$ (see left-side of Example 4.1), `:C4` is defined as an intersection of the concepts `:C2` and `:C3`. Therefore, by rule (a), `:C4 rdfs:subClassOf :C2` and `:C4 rdfs:subClassOf :C3`. Similarly, by rule (b), `:C2 rdfs:subClassOf :C5` and `:C4 rdfs:subClassOf :C5`. Given `:C2` as a selected concept, the extracted sub-ontology from $\mathcal{O}_2$ is shown (on the right-side) in Example 4.1. Given `:C4 rdfs:subClassOf :C2` in $\mathcal{O}_2$, by *sub-concept* rule (in Section 4.2), `:C4` is also considered as a selected concept. For the selected concept `:C2`, the super-concepts are `:C1` and `:C5`. For `:C4`, the super-concepts are `:C1`, `:C2`, `:C3` and `:C5`. Based on the earlier described rationale (see Section 4.3), all the non-selected super-concepts of the selected concepts `:C2` and `:C4` are described as RDF blank-nodes (see bottom-part of Example 4.1).

***Example 4.1: Sub-Ontology extraction with complex concepts***

**OWL Ontology ($\mathcal{O}_2$):**

```
:C1 a owl:Class; rdfs:subClassOf owl:Thing.
:C2 a owl:Class; rdfs:subClassOf :C1.
:C3 a owl:Class; rdfs:subClassOf :C1.
:C4 a owl:Class; owl:intersectionOf (:C2 :C3).

:C5 a owl:Class; owl:unionOf (:C4 :C2).

:P1 a owl:ObjectProperty;
rdfs:domain :C1; rdfs:range :C2.
:P2 a owl:ObjectProperty;
rdfs:domain :C2; rdfs:range :C2.
:i1 a :C1. :i11 a :C1.
:i2 a :C2. :i22 a :C2.
:i3 a :C3. :i33 a :C3.
:i4 a :C4. :i5 a :C5.
:i1 :P1 :i2. :i1 :P1 :i4.
:i2 :P1 :i4. :i2 :P2 :i2.
:i3 :P1 :i4. :i4 :P1 :i2.
```

**Extracted Sub-ontology:**

```
_:t8 a owl:Class; rdfs:subClassOf owl:Thing.
:C2 a owl:Class; rdfs:subClassOf _:t8.
_:t10 a owl:Class; rdfs:subClassOf _:t8.
:C4 a owl:Class.
:C4 rdfs:subClassOf :C2, _:t10, _:t8.
_:t9 a owl:Class. :C2 rdfs:subClassOf _:t9.
:C4 rdfs:subClassOf _:t9.
:P1 a owl:ObjectProperty;
rdfs:domain _:t8; rdfs:range :C2.
:P2 a owl:ObjectProperty;
rdfs:domain :C2; rdfs:range :C2.
~~:i1 a :C1.~~ ~~:i11 a :C1.~~
:i2 a :C2. :i22 a :C2.
~~:i3 a :C3.~~ ~~:i33 a :C3.~~
:i4 a :C4. ~~:i5 a :C5.~~
~~:i1 :P1 :i2.~~ ~~:i1 :P1 :i4.~~
:i2 :P1 :i4. :i2 :P2 :i2.
~~:i3 :P1 :i4.~~ :i4 :P1 :i2.
```

**RDF Blank-Nodes:** `_:t8 e:tuple (:C1). _:t10 e:tuple (:C3). _:t9 e:tuple (:C5).`

### 4.5 Dealing with Concept Axioms for `sbont:extract` Concepts

Identification of concepts that are structurally-connected with `sbont:extract` concepts (through the concept axioms) is done by the following N3 rules:

*Sub-concept*: The sub-concepts of a `sbont:extract` concept, are also `sbont:extract`: `{?O sbont:extract ?C. ?C a rdfs:Class. ?S rdfs:subClassOf ?C.} => {?O sbont:extract ?S}`, where `rdfs:subClassOf a owl:TransitiveProperty`.

*Equivalent-class*: The equivalent concepts of a `sbont:extract` concept, are also `sbont:extract`: `{?O sbont:extract ?C. ?C owl:equivalentClass ?S.} => {?O sbont:extract ?S}`, where `owl:equivalentClass a owl:SymmetricProperty, owl:TransitiveProperty`.

*Complement-of*: The concepts that are complement with a `sbont:extract` concept, are also `sbont:extract`: `{?O sbont:extract ?C. ?C owl:complementOf ?S.} => {?O sbont:extract ?S}`.

*Restriction*: The concepts that impose restrictions on a `sbont:extract` concept, are also `sbont:extract`: `{?O sbont:extract ?C. ?C a rdfs:Class. ?C rdfs:subClassOf ?R. ?R a owl:Restriction} => {?O sbont:extract ?R}`.

### 4.6 Dealing with Role Axioms for `sbont:extract` Roles

Identification of roles that are structurally-connected with `sbont:extract` roles (through the role axioms) is done by the following N3 rules:
*Sub-property*: Implicitly done by the rule *Sub-concept* (see Section 4.2 and 4.5).
*Equivalent-property*: The equivalent roles of a `sbont:extract` role, are also `sbont:extract`: `{?O sbont:extract ?P. ?P owl:equivalentProperty ?P1} => {?O usr:selected ?P1}.`
*Inverse-property*: The inverse roles of a `sbont:extract` role, are also `sbont:extract`: `{?O sbont:extract ?P. ?P owl:inverseOf ?P1} => {?O sbont:extract ?P1}.`

### 4.7 Extracting RDF-Graphs for `sbont:extract` Concepts

Extraction for a `sbont:extract` concept $C$ is performed by the following N3 rule that extracts three types of axioms and assertions for $C$ in individual RDF-Sub-Graphs: (i) concept axioms and concept assertions that need to be extracted, (ii) concept axioms that need to be replaced, and (iii) concept axioms and concept assertions that need to be removed. Extraction for a `sbont:extract` concept $C$ is described in detail as follows:

```
{?O sbont:extract ?C. ?C a rdfs:Class. (?scp 2) e:findall ({?C ?P ?Q} {?C ?P ?Q}
?list1). ?list1 log:conjunction ?G1. (?scp 2) e:findall ({?i a ?C} {?C log:notEqualTo
owl:Thing. ?O usr:selected ?C. ?i a ?C.} ?list2). ?list2 log:conjunction ?G2. (?scp
2) e:findall ({?i1 a ?C} {?C log:notEqualTo owl:Thing. ?i1 a ?C. ?O sbont:extract
?P1. ?P1 a rdf:Property. ?O usr:selected ?C2. ?C2 a rdfs:Class. ?i2 a ?C2.
?i1 ?P1 ?i2} ?list3). ?list3 log:conjunction ?G3. (?scp 2) e:findall ({?i2 a ?C} {?C
log:notEqualTo owl:Thing. ?i2 a ?C. ?O sbont:extract ?P1. ?P1 a rdf:Property. ?O
usr:selected ?C1. ?C1 a rdfs:Class. ?i1 a ?C1. ?i1 ?P1 ?i2} ?list4). ?list4 log:con
junction ?G4. (?scp 3) e:findall ({?C rdfs:subClassOf ?B. ?B a Q.} {?C rdfs:subClassOf
?C1. ?O sbont:noValue ?C1. ?B e:tuple (?C1). ?C1 a ?Q} ?list5). ?list5 log:conjunction
?G5. (?G1 ?G2 ?G3 ?G4 ?G5) log:conjunction ?G. (?scp 3) e:findall ({?C rdfs:subClassOf
?C1} {?C rdfs:subClassOf ?C1. ?O sbont:noValue ?C1.} ?list6). ?list6 log:conjunction
?G6. (?G6 {?X1 usr:selected ?Y1. ?X2 sbont:extract ?Y2. ?X3 sbont:noValue ?Y3. ?X4
owl:unionOf ?Y4. ?X5 owl:intersectionOf ?Y5}) log:conjunction ?R.
(?G ?R) e:graphDifference ?GRAPH.} => {?O sbont:triple_set ?GRAPH}.
```

The first four RDF-Sub-Graphs (`?G1`, `?G2`, `?G3`, `?G4`) consist of the concept axioms and concept assertions that need to be extracted for a `sbont:extract` concept $C$, where (i) `?G1` consists of the concept axioms for $C$; (ii) `?G2` consists of all the concept assertions (of the form $C(a)$ such that $C \in \mathcal{SC}, a \in \mathcal{I}$) for $C$, if $C$ is also `usr:selected`; (iii) `?G3` and `?G4` consist of only those concept axioms for $C$ that are defined via role assertions using the `sbont:extract` roles and `usr:selected` concepts. In the Example 4.1, since `:C2` is a `usr:selected` concept, therefore all the concept assertions (e.g. `:i2 a :C2. :i22 a :C2`) for `:C2` are also extracted in the sub-ontology (i.e. concept assertions that are collected in `?G2`). The RDF-Sub-Graph `?G5` consists of the concept axioms that need to be replaced for a `sbont:extract` concept $C$. Such concept axioms are of the

form $C \sqsubseteq D \in \mathcal{T}$ such that $C \in \mathcal{SC}, D \notin \mathcal{SC}$. Based on the described motivation in Section 4.3, these concepts axioms are replaced by the axioms of the following form $C \sqsubseteq B$, where $B$ `e:tuple` $(D)$. The RDF-Sub-Graphs `?G1`, `?G2`, `?G3`, `?G4` and `?G5` are combined in the RDF-Graph `?G`. Axioms of the form $C \sqsubseteq D \in \mathcal{T}$ such that $C \in \mathcal{SC}, D \notin \mathcal{SC}$, need to be removed, and are collected in `?G6`. The RDF-Sub-Graph `?G6` and other non-related assertions are combined in `?R`. The RDF-Graph `?GRAPH` is the *graph-difference* (performed by the Euler built-in `e:graphDifference`) of `?R` from `?G` (i.e. $GRAPH = G - R$) that represents the final extracted RDF-Graph for a `sbont:extract` concept $C$, and gets included in the sub-ontology $\mathcal{O}$ as `?O sbont:triple_set ?GRAPH`.

### 4.8 Extracting RDF-Graphs Graph for `sbont:extract` Roles

Similar to the extraction for a `sbont:extract` concept $C$ (see Section 4.7), extraction for a `sbont:extract` role $P$ is performed by the following N3 rule that extracts three types of axioms and assertions for $P$ in individual RDF-Sub-Graphs: (i) role axioms and role assertions that need to be extracted, (ii) role axioms that need to be replaced, and (iii) role axioms and role assertions that need to be removed.

```
{?O sbont:extract ?P. ?P a rdf:Property. (?scp 2) e:findall ({?P ?S ?Q} {?P ?S ?Q}
?list1). ?list1 log:conjunction ?G1. (?scp 2) e:findall ({?S ?Q ?P} {?S ?Q ?P} ?list2).
?list2 log:conjunction ?G2. (?scp 2) e:findall ({?S ?P ?Q} {?S ?P ?Q. ?S a ?E1. ?Q a
?E2. ?O usr:selected ?E1. ?O sbont:extract ?E2. ?E1 log:notEqualTo owl:Thing. ?E2
log:notEqualTo owl:Thing.} ?list3). ?list3 log:conjunction ?G3. (?scp 2) e:findall
({?S ?P ?Q} {?S ?P ?Q. ?S a ?E1. ?Q a ?E2. ?O sbont:extract ?E1. ?O usr:selected ?E2.
?E1 log:notEqualTo owl:Thing. ?E2 log:notEqualTo owl:Thing.} ?list4). ?list4
log:conjunction ?G4. (?scp 3) e:findall ({?P rdfs:domain ?B. ?E1 rdfs:subClassOf ?B.
?B a owl:Class} {?P rdfs:domain ?E. ?O sbont:noValue ?E. ?E1 rdfs:subClassOf ?E. ?O
sbont:extract ?E1. ?B e:tuple (?E)} ?list5). ?list5 log:conjunction ?G5. (?G1 ?G2 ?G3
?G4 ?G5) log:conjunction ?G. (?scp 3) e:findall ({?P rdfs:domain ?E.} {?P rdfs:domain
?E. ?O sbont:noValue ?E} ?list6). ?list6 log:conjunction ?G6. (?scp 3) e:findall ({?P
rdfs:range ?E.} {?P a owl:ObjectProperty. ?P rdfs:range ?E. ?O sbont:noValue ?E.}
?list7). ?list7 log:conjunction ?G7. (?G6 ?G7 {?O usr:selected ?P. ?O sbont:extract
?P. ?X sbont:noValue ?Y.}) log:conjunction ?R. (?G ?R) e:graphDifference ?GRAPH.} =>
{?O sbont:triple_set ?GRAPH}.
```

The first four RDF-Sub-Graphs (`?G1`, `?G2`, `?G3`, `?G4`) consist of the role axioms and role assertions that need to be extracted for a `sbont:extract` role $P$, where (i) `?G1` and `?G2` consists of the role axioms for $P$; and (ii) `?G3` and `?G4` consist of only those role assertions for $P$ that are defined between the `sbont:extract` and `usr:selected` concepts. In the Example 4.1, since `:i2 :P1 :i4` is a role assertion defined between `usr:selected` concepts `:C2` and `:C4`, therefore `:i2 :P1 :i4` is extracted in the sub-ontology (i.e. concept assertions that are collected in `?G1`). Whereas, `:i1 :P1 :i2` is not extracted; because `:i1 a :C1`, `:C1` $\notin \mathcal{SC}$. Based on the described motivation in Section 4.3, given $C \sqsubseteq D \in \mathcal{T}$ such that $C \in \mathcal{SC}, D \notin \mathcal{SC}$ and $P$ is inherited from $D$, the role axiom for the domain of $P$ is replaced by `?P rdfs:domain ?B`, where $B$ `e:tuple` $(D)$

(see Section 4.3 for an example). The RDF-Sub-Graphs `?G1`, `?G2`, `?G3`, `?G4` and `?G5` are combined in the RDF-Graph `?G`. Axioms of the form $P$ `rdfs:domain` $D$ such that $C \in \mathcal{SC}, D \notin \mathcal{SC}$, need to be removed, and are collected in `?G6`. Similarly, axioms of the form $P$ `rdfs:range` $D$ such that $C \in \mathcal{SC}, D \notin \mathcal{SC}$, need to be removed, and are collected in `?G7`. The RDF-Sub-Graphs `?G6`, `?G7` and other non-related assertions are combined in `?R`. The RDF-Graph `?GRAPH` is the *graph-difference* of `?R` from `?G` (i.e. $GRAPH = G - R$) that represents the final extracted RDF-Graph for a `sbont:extract` role $P$, and gets included in the sub-ontology $\mathcal{O}$ as `?O sbont:triple_set ?GRAPH`.

## 5 Experiment and Results: PC Test-case

As a test-case, we use a Prostate Cancer (PC) ontology $\mathcal{O}_{PC}$ that describes PC clinical pathways [11]. The PC ontology is defined using RDF(S) and OWL constructs, and deals with four major types of clinical tasks, namely (a) *Consultation Task*; (b) *Non-consulation Task*; (c) *Referal Task*; and (d) *Followup Task*, represented as concepts in $\mathcal{O}_{PC}$. Such tasks are supported by other concepts such as *Clinician*, *Decision Criteria*, *Frequency*, *Interval Duration*, *Investigation*, *Patient Condition Severity*, *Test Result*, and *Treatment*. In total $\mathcal{O}_{PC}$ consists of 30 concepts, 23 roles and 96 individuals. The user is interested in such an extracted sub-ontology $\mathcal{O}'_{PC} \prec \mathcal{O}_{PC}$ that describes only (i) the treatments, (ii) their durations, (iii) their follow-ups, (iv) their care-settings, and (v) the practitioners involved for them. Based on user interest, two concepts *Treatment* and *Clinician* were labelled as `usr:selected` concepts. Given the user-selected concepts, their immediate structurally-connected concepts, roles and individuals were extracted in $\mathcal{O}'_{PC}$. The extracted sub-ontology $\mathcal{O}'_{PC}$ was validated for conceptual consistency and completeness. Concepts (including their axioms and assertions) that were extracted in $\mathcal{O}'_{PC}$ are *Treatment*, *Followup*, *Frequency*, *Interval Duration*, and *Clinician*. Extraction of the PC sub-ontology $\mathcal{O}'_{PC}$ took 92.80 sec. However, in future, we plan to optimize the above described N3 rules–in order to improve the reasoning and reduce the time-complexity of our extraction method.

## 6 Evaluation

We compare our sub-ontology method with two other mentioned approaches MOVE [5, 14] and Miao et. al. [7]. For a given user-selected concept `:C2`, Table 5.1 shows a comparison between the sub-ontologies extracted by the MOVE method (on the left) and by our method (on the right). Both methods are applied on the inferred ontology of the example OWL ontology $\mathcal{O}_2$ (under RDF(S) and OWL inference rules; see Example 4.1). Due to the 1st SCOS criteria in MOVE (see Section 2), concepts `:C1` and `:C5` (and their roles) also got selected; because `:C1` and `:C5` are super-concepts of `:C2`. Since `:C1` and `:C5` are selected, therefore by the 2nd SCOS criteria, all its sub-concepts `:C2`, `:C3`, and `:C4` (and their roles) are also selected. Hence all the concepts and their roles are in $\mathcal{O}_2$ are selected, and therefore are extracted in the sub-ontology shown (left-side) in Table 5.1.

The extracted sub-ontology by Miao et. al. [7] is too restricted, and consists of the single axioms `:C2 a owl:Class`.

***Table 5.1: Comparison Between Extracted Sub-ontologies***

| **By MOVE Method [7]:** | **By Our Method:** |
|---|---|
| `:C1 a owl:Class; rdfs:subClassOf owl:Thing.` | `_:t8 a owl:Class; rdfs:subClassOf owl:Thing.` |
| `:C2 a owl:Class; rdfs:subClassOf :C1.` | `:C2 a owl:Class; rdfs:subClassOf _:t8.` |
| `:C3 a owl:Class; rdfs:subClassOf :C1.` | `_:t10 a owl:Class; rdfs:subClassOf _:t8.` |
| `:C4 a owl:Class.` | `:C4 a owl:Class.` |
| `:C4 rdfs:subClassOf :C2, C3, :C1.` | `:C4 rdfs:subClassOf :C2, _:t10, _:t8.` |
| `C5 a owl:Class. :C2 rdfs:subClassOf C5.` | `_:t9 a owl:Class. :C2 rdfs:subClassOf _:t9.` |
| `:C4 rdfs:subClassOf C5.` | `:C4 rdfs:subClassOf _:t9.` |
| `:P1 a owl:ObjectProperty;` | `:P1 a owl:ObjectProperty;` |
| `rdfs:domain :C1; rdfs:range :C2.` | `rdfs:domain _:t8; rdfs:range :C2.` |
| `:P2 a owl:ObjectProperty;` | `:P2 a owl:ObjectProperty;` |
| `rdfs:domain :C2; rdfs:range :C2.` | `rdfs:domain :C2; rdfs:range :C2.` |
| `:i1 a :C1. :i11 a :C1.` | ~~`:i1 a :C1.`~~ ~~`:i11 a :C1.`~~ |
| `:i2 a :C2. :i22 a :C2.` | `:i2 a :C2. :i22 a :C2.` |
| `:i3 a :C3. :i33 a :C3.` | ~~`:i3 a :C3.`~~ ~~`:i33 a :C3.`~~ |
| `:i4 a :C4. :i5 a :C5.` | `:i4 a :C4.` ~~`:i5 a :C5.`~~ |
| `:i1 :P1 :i2. :i1 :P1 :i4.` | ~~`:i1 :P1 :i2.`~~ ~~`:i1 :P1 :i4.`~~ |
| `:i2 :P1 :i4. :i2 :P2 :i2.` | `:i2 :P1 :i4. :i2 :P2 :i2.` |
| `:i3 :P1 :i4. :i4 :P1 :i2.` | ~~`:i3 :P1 :i4.`~~ `:i4 :P1 :i2.` |

**RDF Blank-Nodes:** `_:t8 e:tuple (:C1). _:t10 e:tuple (:C3). _:t9 e:tuple (:C5).`

Whereas in our approach, the concept `:C4` and its roles are selected; because `:C4` is (the only) sub-concept of `:C2`. Since all the other concepts `:C1`, `:C3` and `:C5` are the super-concepts of either `:C2` or `:C4`, therefore `:C1`, `:C3` and `:C5` are extracted as RDF blank-nodes `_:t8`, `_:t10` and `_:t9`, respectively. However, the role `:P1` (and its role assertions) in `:C1` and the role `:P2` (and its role assertions) in `:C2`, are also extracted (see Section 4.3). Since `:C1`, `:C3` and `:C5` are not extracted in their original form, therefore their individuals (`:i1, :i11, :i3, :i33, :i5`) are also not extracted. In addition, any role assertions (`:i1 :P1 :i2.`,`:i1 :P1 :i4. and :i3 :P1 :i4.`) that are described through non-extracted individuals are not extracted in the sub-ontology. Hence compared to the MOVE, the extracted sub-ontology by our method provides all the pertinent knowledge about the selected concept `:C2`; without describing extra and irrelevant knowledge about `:C2`.

## 7 Conclusion

Extracting sub-ontologies from source ontologies is viable for supporting specialized utilization and scoped re-conciliation of ontologies. We presented our approach for sub-ontology extraction, where the concepts (and roles) that are pertinent for a target ontology are extracted, and then can either be reused or reconciled with other extracted concepts and relationships. Similar to the related work [9], we proposed our framework for Semantic Web Based Knowledge Representation and Context-driven Morphing ($\mathcal{K}$-$\mathcal{MORPH}$) [18], where (i) sub-ontologies are extracted based on the given problem-context–known as *contextualized sub-ontologies*; (ii) a morphed ontology is generated by integrating contextualized sub-ontologies under domain-specific and context-specific axioms; and (iii) inconsistencies occur in the morphed ontology are detected and resolved. Future work includes extraction and consistent integration of con-

textualized sub-ontologies based on given domain-specific and context-specific axioms, whereas some of our initial efforts are reported in [18].

This research is funded by a grant from Agfa Healthcare (Canada).

## References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Description Logic Handbook. Cambridge University Press (2003)
2. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3Logic: A logical framework for the world wide web. Theory and Practice of Logic Programming **8**(03) (May 2008) 249–269
3. W3C: OWL 2 Web Ontology Language: http://www.w3.org/TR/owl2-profiles/.
4. Roo, J.D.: Euler proof mechanism http://eulersharp.sourceforge.net/.
5. Wouters, C., Dillon, T., Rahayu, W., Chang, E., Meersman, R.: A practical approach to the derivation of materialized ontology view. In: Web Information Systems. Idea Group Publishing (2004) 191–226
6. Rajugan, R., Chang, E., Dillon, T.S.: Sub-ontologies and ontology views; a theoretical perspective. Int. J. Metadata Semant. Ontologies **2**(2) (2007) 94–111
7. Miao, Z., Wang, J., Zhou, B., Zhang, Y., Lu, J.: Method for extracting RDF(S) sub-ontology. (2008) 348–353
8. Mao, Y., Cheung, W.K., Wu, Z., Liu, J.: Dynamic sub-ontology evolution for collaborative problem-solving. In: AAAI fall symposium. (2005) 1–8
9. Kang, D., Xu, B., Lu, J., Wang, P., Li, Y.: Extracting sub-ontology from multiple ontologies. In: OTM 2004 Workshop on Ontologies, Semantics, and E-learning (WOSE). Volume 3292 of LNCS. (2004) 731740
10. Abidi, S.S.R.: Medical knowledge morphing: Towards the integration of medical knowledge resources. Computer-Based Medical Systems (June 23-24 2005)
11. Abidi, S., Abidi, R., Hussain, S., Butler, L.: Ontology-based modeling and merging of institution-specific prostate cancer clinical pathways. In: Knowledge Management for Healthcare Processes Workshop at ECAI 2008, Patras, Greece (July 21-25 2008)
12. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
13. Haase, P., Qi, G.: An analysis of approaches to resolving inconsistencies in DL-based ontologies. In: International Workshop on Ontology Dynamics. (June 2007)
14. Bhatt, M., Wouters, C., Flahive, A., Rahayu, W., Taniar, D.: Semantic completeness in sub-ontology extraction using distributed methods. In: ICCSA. Volume 3045 of LNCS., Springer (2004) 508–517
15. Polleres, A., Feier, C., Harth, A.: Rules with contextually scoped negation. In: ESWC 2006. Volume 4011 of LNCS., Springer (2006) 332347
16. Bezem, M., Coquand, T.: Automating coherent logic. In: 12th International conference on Logic for programming, artificial intelligence, and reasoning, LPAR 2005. Volume 3835 of LNCS., Jamaica, Springer (December 2-6 2005) 246–260
17. Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics **3**(4) (2005) 247–267
18. Hussain, S., Abidi, S.S.R.: K-MORPH: A semantic web based knowledge representation and context-driven morphing framework. In: Workshop on Context and Ontologies, at ECAI'2008, Patras, Greece (July 21-25 2008)