



# MỘT SỐ PHƯƠNG PHÁP BỔ TRỢ CHO VIỆC TRÍCH XUẤT MÔ HÌNH ER TỪ OWL2

Võ Hoàng Liên Minh\*, Hoàng Quang

Trường Đại học Khoa học, Đại học Huế, 77 Nguyễn Huệ, Huế, Việt Nam

**Tóm tắt.** OWL là ngôn ngữ để mô hình hóa các hệ thống thông tin thông qua việc mô tả các lớp, các thuộc tính và các mối quan hệ giữa các thành phần này giúp cho máy tính có thể “hiểu” và xử lý thông tin. Tuy nhiên, định dạng của OWL là khó có thể hiểu về cú pháp của OWL cũng như khó thao tác và khó truy vấn. Các nghiên cứu đã đề xuất phương pháp ánh xạ từ mô hình ER (Entity-Relationship) và ER mở rộng sang OWL (Web Ontology Language) nhằm sử dụng lại các hệ thống cũ giúp giảm chi phí thay vì phải thiết kế các ontology từ đầu. Vấn đề trích xuất một mô hình ER và ER mở rộng từ OWL cho trước được xem là việc xác định một ánh xạ ngược của ánh xạ chuyển đổi từ mô hình ER và ER mở rộng sang OWL. Trên thực tế, kỹ thuật chuyển đổi ngược này sẽ cho phép chúng ta xây dựng một mô hình dữ liệu mức khái niệm có thể đã được sử dụng để thiết kế một ontology. Để giúp cho việc hiểu rõ cấu trúc của OWL cũng như dữ liệu OWL, bài báo này giới thiệu tập các quy tắc trích xuất mô hình ER từ OWL, trong đó đề xuất bổ sung một số trường hợp trích xuất mô hình ER từ OWL như thuộc tính khóa, thuộc tính phức hợp hay mối quan hệ phản xạ.

**Từ khóa:** OWL, ontology, ER, kỹ thuật chuyển đổi ngược

## 1 Giới thiệu

Sự phát triển của Web ngữ nghĩa sẽ giúp cho các máy tính dễ dàng nhận dạng và xử lý các thông tin, tích hợp thông tin và hỗ trợ con người. Ontology Web Language (OWL) là ngôn ngữ biểu diễn các thực thể trên một miền dữ liệu nhất định, hoặc để diễn đạt ý nghĩa khác nhau của các thuật ngữ vốn có thể không thực sự tương minh và xác định các mối quan hệ giữa chúng. OWL được thiết kế cho web ngữ nghĩa, thường được sử dụng bởi các ứng dụng cần xử lý nội dung thông tin và thực hiện suy luận thay vì chỉ trình bày thông tin.

Mô hình ER được đề xuất bởi P. Chen [12] là một mô hình khái niệm, dựa vào việc nhận thức thế giới thực thông qua tập các đối tượng được gọi là các thực thể và các mối quan hệ giữa những đối tượng này. Ngoài việc đóng vai trò là mô hình khái niệm, mô hình ER còn được xem là một trong những mô hình dữ liệu ngữ nghĩa. Thông qua mô hình ER, người ta có thể xác định được các tập thực thể của một hệ thống thông tin, đồng thời ngữ nghĩa của mô hình còn được thể hiện bởi các mối quan hệ giữa các tập thực thể.

\* Liên hệ: [minhvhl@gmail.com](mailto:minhvhl@gmail.com)

Các nghiên cứu của các tác giả M. Fahad [1], I. Myroshnichenko [2], P. Chujai [3], Toan Van Nguyen [4], S. Brockmans [5] đã đề xuất phương pháp chuyển đổi từ mô hình ER và ER mở rộng sang OWL ontology nhằm sử dụng lại các hệ thống cũ thay vì phải thiết kế các ontology từ đầu. Vấn đề trích xuất mô hình mức khái niệm từ OWL ontology cho trước được xem là việc xác định một ánh xạ ngược của ánh xạ chuyển đổi từ mô hình mức khái niệm sang OWL ontology. S. Bagui [6] đã tập trung vào việc phát triển tập các quy tắc chuyển đổi OWL thành mô hình ER và EER (Enhanced Entity-Relationship) bằng cách lần lượt xác định các thành phần cơ bản của mô hình ER và EER trong một OWL cho trước. Tác giả đề xuất 11 quy tắc chính để ánh xạ các lớp và thuộc tính trong OWL thành các thực thể, thuộc tính OWL và mối quan hệ trong mô hình ER. Tuy nhiên, nhiều thành phần tương ứng chưa đề cập như thuộc tính phức hợp, mối quan hệ phản xạ... Hay tác giả chỉ mới phân tích trên OWL1 chứ chưa xem xét trên OWL2 nên chưa xem xét đầy đủ cho các trường hợp như thuộc tính khóa trong quá trình chuyển đổi.

C. Wang và cộng sự [16] giới thiệu phương pháp ánh xạ từ OWL-s thành biểu đồ hành động. Weijun Li [7] đã đề xuất các khái niệm mới về Onto2 và mô hình UML (Unified Modeling Language), từ đó đề xuất cách ánh xạ OWL2 thành mô hình UML. Tuy nhiên, tác giả [7] chỉ mới đưa ra khái niệm về Onto2 và mô hình UML bằng lý thuyết toán học mà chưa giải quyết một cách tường minh; ví dụ, làm thế nào để xác định trường hợp nào là tương ứng với các loại quan hệ trong UML.

Bài báo này giới thiệu bổ sung các quy tắc trích xuất nhằm đưa ra một giải pháp giải quyết các trường hợp chưa đề xuất, từ đó cung cấp kỹ thuật chuyển đổi hoàn chỉnh nhất có thể. Điều này là hoàn toàn hợp lý vì cần phải chuyển đổi tất cả các thành phần liên quan bằng cách xem xét cấu trúc OWL trong quá trình trích xuất sang mô hình ER từ OWL.

Bài báo sẽ phân tích OWL và xác định cấu trúc tương đương khi trích xuất thành mô hình ER, như làm thế nào để nhận diện tập thực thể, thuộc tính, hay một mối quan hệ. Theo đó, cấu trúc của bài báo như sau: Mục tiếp theo trình bày bổ sung phương pháp chuyển đổi OWL ontology thành mô hình ER. Mục 3 là phần kết luận và thảo luận về hướng nghiên cứu tiếp theo.

## 2 Trích xuất mô hình ER từ OWL2

Để phân tích sự tương đồng giữa mô hình ER và ER mở rộng với OWL ontology đã có nhiều tác giả nghiên cứu như [1] [2] [3] [4] [5] [6]. Bài báo kế thừa những phân tích đó và thảo luận làm thế nào để chuyển đổi cấu trúc OWL sang mô hình ER. Chúng tôi phân tích trên cấu trúc OWL2 để tổng quát hóa các quy tắc trích xuất.

### A. Trích xuất tập thực thể từ OWL

Trong OWL, lớp định nghĩa một nhóm các cá thể vì chúng có cùng thuộc tính. Có hai cấu trúc khác nhau có thể được sử dụng để định nghĩa một lớp. Cấu trúc thứ nhất được sử dụng để định nghĩa một lớp là *owl:class*. Cấu trúc thứ hai được sử dụng để xác định một lớp (hoặc lớp con) là *rdfs:subClassOf* [6]. Ta thấy rằng khái niệm lớp trong OWL tương đương với tập thực thể trong mô hình ER; vì vậy, quy tắc trích xuất như sau:

**Quy tắc OWL1:** Lớp hoặc lớp con  $C(E)$  khai báo bằng cú pháp *owl:class* hoặc *rdfs:subClassOf* được chuyển đổi thành tập thực thể  $E$  trong mô hình ER [6].

#### B. Trích xuất thuộc tính từ OWL

Trong OWL, thuộc tính kiểu dữ liệu mô tả các đối tượng cá thể của một lớp và cung cấp các dữ kiện cụ thể về các đối tượng cá thể trong một lớp. Thuộc tính kiểu dữ liệu liên kết các đối tượng cá thể với các giá trị kiểu dữ liệu. Với mô hình ER, thực thể được mô tả bởi các thuộc tính sau.

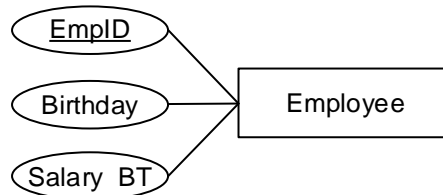
##### 1) Trích xuất thuộc tính đơn trị

Trong OWL, thuộc tính kiểu dữ liệu thường liên kết các cá thể với các giá trị dữ liệu. Tuy nhiên, theo mặc định, OWL cho phép các thuộc tính có nhiều giá trị. Để một thuộc tính kiểu dữ liệu nhận một giá trị duy nhất, thì thuộc tính đó được thiết lập tính chất hàm *FunctionalDataProperty*. Thuộc tính kiểu dữ liệu là kiểu dữ liệu đơn. Vì thế, chúng tương đương với thuộc tính đơn trong mô hình ER; điều này tương ứng ngữ nghĩa với thuộc tính đơn trị của mô hình ER.

Vì vậy, ta có quy tắc trích xuất thuộc tính đơn trị như sau:

**Quy tắc OWL2:** Thuộc tính kiểu dữ liệu *attE* có tính chất hàm *FunctionalDataProperty*, có miền là  $C(E)$  và kiểu dữ liệu nguyên thủy trong OWL được chuyển đổi thành thuộc tính đơn trị *attE* của lớp  $E$ , có kiểu dữ liệu tương ứng trong mô hình ER [6].

```
<owl:DatatypeProperty rdf:about="#Birthday">
  <rdf:type rdf:resource="#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Employee"/>
  <rdfs:range rdf:resource="#dateTime"/>
</owl:DatatypeProperty>
```



**Ví dụ 1.** Mã nguồn biểu diễn thuộc tính trong OWL

**Hình 1.** Trích xuất thuộc tính đơn trị

Như ở Ví dụ 1, thuộc tính kiểu dữ liệu *Birthday* được biểu diễn có tính chất hàm *FunctionalDataProperty*, có miền là lớp *Employee* và kiểu dữ liệu là kiểu ngày *dateTime*. Vì thế, nó được trích xuất thành thuộc tính của lớp *Employee* (Hình 1).

## 2) Trích xuất thuộc tính khoá

Trong OWL1, không có khái niệm khoá chính cũng như hỗ trợ khai báo khoá, nhưng trong OWL2, một tập hợp các thuộc tính (dữ liệu hoặc đối tượng) có thể được gán như là khoá bằng cách sử dụng cú pháp *owl:hasKey*. Điều này tương đương với ngữ nghĩa của khoá trong mô hình ER vì thuộc tính khoá phải là thuộc tính đơn, chỉ nhận một giá trị duy nhất, không nhận giá trị *null* nên ta có quy tắc chuyển đổi thuộc tính khoá như sau:

**Quy tắc OWL3:** Thuộc tính kiểu dữ liệu đơn *KE* có miền là  $C(E)$ , phạm vi là kiểu dữ liệu nguyên thủy trong OWL được khai báo bằng cú pháp *owl:hasKey* và ràng buộc *minQualifiedCardinality* và *maxQualifiedCardinality* thiết lập là 1 thì chuyển đổi thành thuộc tính khoá *KE* của thực thể *E*.

```
<owl:Class rdf:about="#Employee">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <owl:hasKey rdf:parseType="Collection">
    <rdf:Description rdf:about="#EmpID"/>
  </owl:hasKey>
</owl:Class>
<owl:DatatypeProperty rdf:about="#EmpID">
  <rdf:type rdf:resource="#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Employee"/>
  <rdfs:domain>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#EmpID"/>
      <owl:minQualifiedCardinality rdf:datatype="#nonNegativeInteger">
1</owl:minQualifiedCardinality>
      <owl:onDataRange rdf:resource="#int"/>
    </owl:Restriction>
  </rdfs:domain>
  <rdfs:domain>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#EmpID"/>
      <owl:maxQualifiedCardinality rdf:datatype="#nonNegativeInteger">
1</owl:maxQualifiedCardinality>
      <owl:onDataRange rdf:resource="#int"/>
    </owl:Restriction>
  </rdfs:domain>
  <rdfs:range rdf:resource="#string"/>
</owl:DatatypeProperty>
```

**Ví dụ 2.** Mã nguồn biểu diễn thuộc tính khoá trong OWL

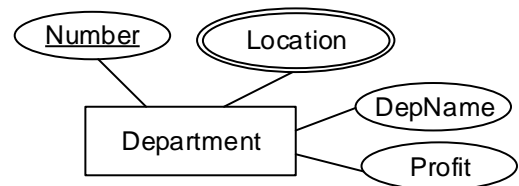
Xét ở Ví dụ 2, lớp *Employee* có khai báo *owl:hasKey* là thuộc tính *EmpID*, đồng thời có thuộc tính kiểu dữ liệu *EmpID* có miền là lớp *Employee* và phạm vi là kiểu dữ liệu *Interger*, ràng buộc *minQualifiedCardinality* và *maxQualifiedCardinality* thiết lập là 1. Vì thế, nó được trích xuất thành thuộc tính khóa *EmpID* của tập thực thể *Employee* (Hình 1).

### 3) Trích xuất thuộc tính đa trị

Trong mô hình ER, thuộc tính đa trị có thể có nhiều giá trị thuộc tính cho mỗi thực thể. Trong OWL, thuộc tính kiểu dữ liệu biểu diễn đa trị ã được xác định bằng hạn chế bản số. Nếu khai báo *cardinality* hoặc *minCardinality* lớn hơn 1 cho thuộc tính của một lớp thì bất kỳ thể hiện của lớp có thể liên quan đến nhiều cá thể của thuộc tính đó. Như đã đề cập ở trên, OWL cho phép các thuộc tính có nhiều giá trị, vì thế có ngữ nghĩa tương đương với thuộc tính đa trị trong mô hình ER.

**Quy tắc OWL4:** Thuộc tính kiểu dữ liệu *attE* có *cardinality* hoặc *minCardinality* lớn hơn 1, có miền là  $C(E)$  và kiểu dữ liệu nguyên thủy trong OWL được chuyển đổi thành thuộc tính đa trị *attE* của lớp *E*, có kiểu dữ liệu tương ứng trong mô hình ER.

```
<owl:DatatypeProperty rdf:about="#Location">
  <rdfs:domain rdf:resource="#Department"/>
  <rdfs:range rdf:resource="#string"/>
</owl:DatatypeProperty>
```



**Ví dụ 3.** Mã nguồn biểu diễn thuộc tính trong OWL

**Hình 2.** Trích xuất thuộc tính đa trị

### C. Trích xuất thuộc tính phức hợp từ OWL

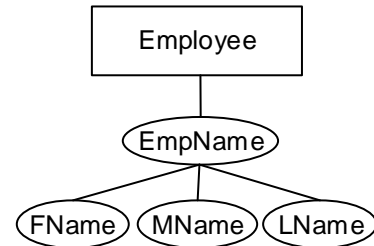
Thuộc tính phức hợp được tạo từ những thuộc tính đơn khác nhau. Nếu các thuộc tính có cấu trúc phức tạp hoặc có các thuộc tính liên hệ với nhau và có ngữ nghĩa cụ thể trong thế giới thực thì nên tách nó ra thành các thuộc tính độc lập với nhau. Tuy nhiên, trong thực tế, vẫn có một số trường hợp thuộc tính vẫn giữ cấu trúc phức tạp để biểu diễn đúng ngữ nghĩa của thế giới thực. Trong OWL, thuộc tính kiểu dữ liệu có nhiều thuộc tính con thành phần, biểu diễn bằng cú pháp *rdfs:subPropertyOf*. Ví dụ 4 cho thấy một thuộc tính dữ liệu có thuộc tính con được mô tả trong OWL.

Điều này tương đương với ngữ nghĩa của thuộc tính phức hợp của mô hình ER, vì thuộc tính phức hợp là thuộc tính được tạo từ những thuộc tính đơn khác nhau. Ví dụ, ngày sinh cấu thành từ ngày, tháng và năm sinh, hay như tên của một người gồm họ, tên đệm và tên. Như ở Ví dụ 4, thuộc tính kiểu dữ liệu con *Fname* được khai báo là thuộc tính con của thuộc tính

*EmpName* bằng cấu trúc *rdfs:subPropertyOf rdf:resource="#EmpName"*. Vì vậy, ta có thể phát biểu quy tắc trích xuất thuộc tính phức hợp như sau:

**Quy tắc OWL5:** Thuộc tính kiểu dữ liệu con *sub\_attE* được khai báo bằng cấu trúc *<rdfs:subPropertyOf rdf:resource="#attE" />*, có miền là thuộc tính *attE* và phạm vi là kiểu dữ liệu nguyên thủy thì trích xuất thành thuộc tính con *sub\_attE* của thuộc tính phức hợp *attE*.

```
<owl:DatatypeProperty rdf:about="#EmpName">
  <rdf:type rdf:resource="#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Employee"/>
  <rdfs:range rdf:resource="#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#Fname">
  <rdfs:subPropertyOf rdf:resource="#EmpName"/>
  <rdf:type rdf:resource="#FunctionalProperty"/>
  <rdfs:domain>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#EmpName"/>
      <owl:someValuesFrom rdf:resource="#string"/>
    </owl:Restriction>
  </rdfs:domain>
  <rdfs:range rdf:resource="#string"/>
</owl:DatatypeProperty>
```



Hình 3. Trích xuất thuộc tính phức hợp

**Ví dụ 4.** Thuộc tính kiểu dữ liệu con *Fname*

#### D. Trích xuất mối quan hệ

Trong mô hình ER, một mối quan hệ nhị nguyên  $R$  giữa hai tập thực thể  $E_1$  và  $E_2$ . Trong OWL, thuộc tính đối tượng được sử dụng để biểu diễn mối quan hệ nhị nguyên giữa các lớp  $C_i$  và  $C_j$  có liên quan với nhau, nó giúp liên kết các cá thể của  $C_i$  đến các cá thể của  $C_j$  [6]. Để diễn tả số lần xuất hiện tối thiểu và tối đa của các thể hiện của tập thực thể trong một mối quan hệ người ta dùng bản số. Bản số (cardinality) là một cặp số nguyên (min, max), chứa số tối thiểu và số tối đa có thể có của các thể hiện của tập thực thể tham gia vào mối quan hệ. Bản số giúp xác định loại của mối quan hệ là 1:1, 1:N, N:1 hay N:N, tức là để xác định mỗi thể hiện của tập thực thể này có liên hệ với bao nhiêu thể hiện của tập thực thể kia thông qua mối quan hệ.

Vậy làm thế nào để xác định mối quan hệ là 1:1, 1:N hay N:N giữa hai lớp trong OWL? Điều này sẽ được xác định bởi ràng buộc min/max của cặp thuộc tính đối tượng.

```
<owl:Class rdf:about="#Department">
  <rdfs:subClassOf>
    <owl:Restriction>
```

```

    <owl:onProperty rdf:resource="#Department_Manages"/>
    <owl:minQualifiedCardinality
rdf:datatype="#nonNegativeInteger">0</owl:minQualifiedCardinality>
    <owl:onClass rdf:resource="#Department"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#Department_Manages"/>
    <owl:maxQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:maxQualifiedCardinality>
    <owl:onClass rdf:resource="#Department"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Manages">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Manages_Department"/>
      <owl:minQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:minQualifiedCardinality>
      <owl:onClass rdf:resource="#Manages"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Manages_Employee"/>
      <owl:minQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:minQualifiedCardinality>
      <owl:onClass rdf:resource="#Manages"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Manages_Department"/>
      <owl:maxQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:maxQualifiedCardinality>
      <owl:onClass rdf:resource="#Manages"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

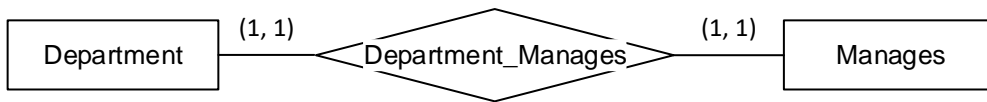
```

**Ví dụ 5.** Cặp thuộc tính đối tượng biểu diễn mối quan hệ 1:1

Ở Ví dụ 5, thuộc tính đối tượng *Department\_Manages* có *minCardinality* và *maxCardinality* là 1 tương ứng với lớp *Department*. *minCardinality* bằng 1 có nghĩa là thuộc tính đối tượng đó

phải có giá trị cho tất cả các thể hiện của lớp này. Tuy nhiên, trong một số trường hợp OWL không khai báo ràng buộc bản số, thì có nghĩa là bản số của quan hệ là 0:N. Vì thế, ta có quy tắc trích xuất như sau:

**Quy tắc OWL6:** Nếu mỗi thuộc tính đối tượng giữa hai lớp  $C(E_1)$  và  $C(E_2)$  với  $minQualifiedCardinality$  là 0 hoặc 1 và  $maxQualifiedCardinality$  là 1; hoặc  $qualifiedCardinality$  là 1, và thuộc tính đối tượng ngược  $inverseOf$  của nó với  $minQualifiedCardinality$  là 0 hoặc 1 và  $maxQualifiedCardinality$  là 1, hoặc  $qualifiedCardinality$  là 1 thì chuyển đổi thành quan hệ nhị nguyên giữa hai tập thực thể  $C(E_1)$  và  $C(E_2)$  với bản số là 1.1.



Hình 4. Trích xuất thành mối quan hệ nhị nguyên 1:1

```

<owl:Class rdf:about="#Department">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Department_Project"/>
      <owl:minQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:minQualifiedCardinality>
      <owl:onClass rdf:resource="#Department"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Department_Project"/>
      <owl:maxQualifiedCardinality
rdf:datatype="#nonNegativeInteger">100</owl:maxQualifiedCardinality>
      <owl:onClass rdf:resource="#Department"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Project">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Project_Department"/>
      <owl:minQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:minQualifiedCardinality>
      <owl:onClass rdf:resource="#Project"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</rdfs:subClassOf>
  
```



```

<owl:Restriction>
  <owl:onProperty rdf:resource="#Project_Department"/>
  <owl:maxQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:maxQualifiedCardinality>
  <owl:onClass rdf:resource="#Project"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

**Ví dụ 6.** Cặp thuộc tính đối tượng biểu diễn mối quan hệ 1:N

Ta thấy rằng ở Ví dụ 6, thuộc tính đối tượng *Department\_Project* có *minCardinality* và *maxCardinality* đều là 1, nhưng *Project\_Department* có *minQualifiedCardinality* là 1 và *maxQualifiedCardinality* lớn hơn 1. Vì thế, ta có thể phát biểu quy tắc trích xuất mối quan hệ nhị nguyên 1:N như sau:

**Quy tắc OWL7:** Nếu mỗi thuộc tính đối tượng giữa hai lớp  $C(E_1)$  và  $C(E_2)$  với *minQualifiedCardinality* là 0 hoặc 1 và *maxQualifiedCardinality* là 1, hoặc *qualifiedCardinality* là 1, và thuộc tính đối tượng ngược *inverseOf* của nó với *minQualifiedCardinality* là 0 hoặc 1 và *maxQualifiedCardinality* lớn hơn 1, hoặc *qualifiedCardinality* lớn hơn 1 thì chuyển đổi thành mối quan hệ nhị nguyên giữa hai lớp  $C(E_1)$  và  $C(E_2)$  với bản số là 1:N.

Và tương tự, ta có quy tắc chuyển đổi quan hệ N:N như sau:

**Quy tắc OWL8:** Nếu mỗi thuộc tính đối tượng giữa hai lớp  $C(E_1)$  và  $C(E_2)$  với *minQualifiedCardinality* là 0 hoặc 1 và *maxQualifiedCardinality* lớn hơn 1, hoặc *qualifiedCardinality* lớn hơn 1, và thuộc tính đối tượng ngược *inverseOf* của nó với *minQualifiedCardinality* là 0 hoặc 1 và *maxQualifiedCardinality* lớn hơn 1, hoặc *qualifiedCardinality* lớn hơn 1 thì trích xuất thành mối quan hệ nhị nguyên bản số là N:N.

4) *Trích xuất mối quan hệ phản xạ*

Trong mô hình ER, mối quan hệ phản xạ là mối quan hệ giữa các thực thể của cùng một tập thực thể. Trong mối quan hệ phản xạ, một tập hợp các thể hiện có thể đảm nhận một vai trò duy nhất hoặc nhiều vai trò trong cùng mối quan hệ [8]. Việc kiểm tra các vai trò cho phép chúng ta phân loại tất cả các mối quan hệ phản xạ thành kiểu mối quan hệ đối xứng hoặc bất đối xứng.

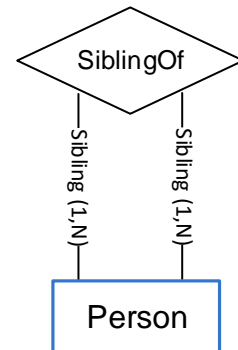
Một mối quan hệ phản xạ được gọi là có tính chất đối xứng khi tất cả các thể hiện tham gia vào mối quan hệ có một vai trò duy nhất và có cùng ngữ nghĩa [9]. Mối quan hệ phản xạ là bất đối xứng khi có một mối quan hệ giữa hai nhóm vai trò khác nhau trong cùng một tập thực thể và ngữ nghĩa của mối quan hệ này là khác nhau tùy thuộc vào hướng mà các nhóm vai trò này quan hệ với nhau [10].

Vậy vấn đề đặt ra là làm thế nào để nhận diện mối quan hệ phản xạ là đối xứng hay bất đối xứng trong OWL? Với mỗi quan hệ phản xạ đối xứng, ta thấy rằng chỉ có một vai trò và bản số tham gia trong mỗi quan hệ là 1:1. Như đã đề cập, thuộc tính đối tượng dùng để biểu diễn mỗi quan hệ trong OWL; vì thế, một thuộc tính đối tượng có miền và phạm vi đều là một lớp tương đương với mỗi quan hệ phản xạ đối xứng. Để đảm bảo biểu diễn đúng bản số 1:1 của mỗi quan hệ phản xạ đối xứng, ta cần xác định thuộc tính đối tượng đó có thiết lập *minQualifiedCardinality* và *maxQualifiedCardinality* là 1. Ta có quy tắc trích xuất như sau:

**Quy tắc OWL9:** Với thuộc tính đối tượng có miền và phạm vi đều là lớp  $C(E)$ , được thiết lập *minQualifiedCardinality* và *maxQualifiedCardinality* là 1 thì trích xuất thành mỗi quan hệ phản xạ đối xứng của tập thực thể  $E$ , vai trò của mỗi quan hệ phản xạ là tên của thuộc tính đối tượng đó.

Ví dụ 7 có thuộc tính đối tượng *siblingOf* có miền và phạm vi là lớp *Person*, và không có thuộc tính ngược và có ràng buộc bằng số *minQualifiedCardinality* là 1. Vì thế, nó được trích xuất thành quan hệ phản xạ đối xứng *siblingOf* của tập thực thể *Person* (Hình 5).

```
<owl:ObjectProperty rdf:about="#siblingOf">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Person"/>
  <rdfs:range>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#siblingOf"/>
      <owl:minQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:minQualifiedCardi
nality>
      <owl:onClass rdf:resource="#Person"/>
    </owl:Restriction>
  </rdfs:range>
</owl:ObjectProperty>
```



**Ví dụ 7.** Ví dụ thuộc tính đối tượng biểu diễn quan hệ phản xạ đối xứng

**Hình 5.** Trích xuất thành mỗi quan hệ phản xạ đối xứng

Với mỗi quan hệ phản xạ bất đối xứng, ta thấy rằng các vai trò là  $role_1$  và  $role_2$ . Vì thế tương ứng với mỗi vai trò chính là một thuộc tính đối tượng có miền và phạm vi đều cùng một lớp, và hai thuộc tính đối tượng này là ngược nhau.

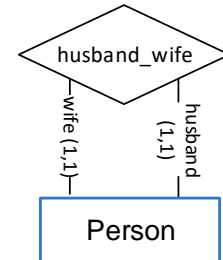
**Quy tắc OWL10:** Với cặp thuộc tính đối tượng ngược nhau  $role_1$  và  $role_2$  có miền và phạm vi đều là lớp  $C(E)$  thì trích xuất thành mỗi quan hệ phản xạ bất đối xứng của tập thực thể  $E$ . Hai

vai trò của mỗi quan hệ phản xạ là tên của hai thuộc tính đối tượng đó. Tùy thuộc vào *minQualifiedCardinality* và *maxQualifiedCardinality* trên từng thuộc tính đối tượng *role<sub>1</sub>* và *role<sub>2</sub>* mà ta thiết lập bản số tương ứng cho mỗi quan hệ phản xạ bất đối xứng đó.

```

<owl:ObjectProperty rdf:about="#husband">
  <owl:inverseOf rdf:resource="#wife"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Person"/>
  <rdfs:range>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#husband"/>
      <owl:minQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:minQualifiedCardinal
ity>
      <owl:onClass rdf:resource="#Person"/>
    </owl:Restriction>
  </rdfs:range>
</rdfs:range>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#husband"/>
    <owl:maxQualifiedCardinality
rdf:datatype="#nonNegativeInteger">1</owl:maxQualifiedCardinal
ity>
    <owl:onClass rdf:resource="#Person"/>
  </owl:Restriction>
</rdfs:range>
</owl:ObjectProperty>

```



**Ví dụ 8.** Ví dụ thuộc tính đối tượng biểu diễn quan hệ phản xạ bất đối xứng

**Hình 6.** Trích xuất thành mỗi quan hệ phản xạ bất đối xứng

Ví dụ 8 có thuộc tính đối tượng *husband* có miền và phạm vi là lớp *Person*, và là thuộc tính ngược của thuộc tính đối tượng *wife*, có ràng buộc bằng số *minQualifiedCardinality* và *maxQualifiedCardinality* là 1. Vì thế, nó được trích xuất thành mỗi quan hệ phản xạ bất đối xứng *husband\_wife* của tập thực thể *Person*.

##### 5) Trích xuất mỗi quan hệ tổng quát hóa/chuyên biệt hóa

Tổng quát hóa và chuyên biệt hóa là hai cách nhìn dưới lên và trên xuống về sự phân cấp các tập thực thể, mô tả khả năng quản lý cấp độ phức tạp của hệ thống bằng cách trừu tượng hóa. Khái niệm tổng quát hóa và chuyên biệt hóa trong mô hình ER mở rộng và OWL là tương

tự. Nếu  $C'$  là lớp con của lớp  $C$  và  $i$  là một thể hiện tương ứng của cá thể thì trong cả hai trường hợp ta có  $C'(i) \rightarrow C(i)$  [11]. Quy tắc trích xuất quan hệ chuyên biệt hóa từ OWL như sau:

**Quy tắc OWL11:** Chuyển đổi lớp con  $C(\text{sub}_E)$  của lớp  $C(E)$  trong OWL thành lớp chuyên biệt hóa.

Trong mô hình ER mở rộng, chuyên biệt hóa được phân thành hai loại: chồng lấp (overlap) và phân ly (disjoint) hoặc có thể kết hợp cả hai loại này. Chúng ta sẽ thảo luận làm thế nào để chuyển đổi chuyên biệt hóa phân ly và chồng lấp.

Mối quan hệ chuyên biệt hóa là phân ly khi một thực thể của lớp cha chỉ có thể thuộc về một trong các lớp con. Trong OWL, cú pháp để có thể biểu diễn cho một mối quan hệ phân ly là *owl:disjointClasses*. Cấu trúc *owl:disjointClasses* có nghĩa là không có thành viên nào chung giữa các lớp. Cấu trúc thứ hai để định nghĩa mối quan hệ phân ly là *owl:complementOf*. Cấu trúc *owl:complementOf* chọn tất cả các cá thể không thuộc một lớp nhất định [17]. Ta có thể phát biểu quy tắc trích xuất mối quan hệ Tổng quát hóa/Chuyên biệt hóa là phân ly như sau:

**Quy tắc OWL12:** Trong cấu trúc lớp – lớp con của OWL, cú pháp *owl:disjointWith* hoặc *owl:complementOf* sẽ được trích xuất thành mối quan hệ Chuyên biệt hóa là phân ly trong mô hình ER mở rộng.

Mối quan hệ Chuyên biệt hóa là chồng lấp giữa các lớp mà tất cả các lớp con đã được chỉ định, và không có lớp con nào thêm. Mỗi thể hiện của lớp cha phải là thể hiện của ít nhất một trong các lớp con. Trong OWL, cấu trúc để biểu diễn cho một mối quan hệ Tổng quát hóa/Chuyên biệt hóa là chồng lấp là *owl:unionOf*. Vì vậy ta có thể phát biểu quy tắc trích xuất như sau:

**Quy tắc OWL13:** Trong cấu trúc lớp cha – lớp con của OWL, cú pháp *owl:unionOf* sẽ được trích xuất thành mối quan hệ chuyên biệt hóa là chồng lấp trong mô hình ER.

#### 6) Trích xuất mối quan hệ hợp

Mối quan hệ hợp cho phép một tập thực thể được kế thừa từ các tập thực thể khác. Trong OWL2, để biểu diễn kế thừa sử dụng cấu trúc *owl:intersectionOf*. Vì thế, ta có thể phát biểu quy tắc trích xuất như sau:

**Quy tắc OWL14:** Trong OWL, lớp  $C(E)$  khai báo bằng cú pháp *owl:intersectionOf* với các lớp  $C(E_i)$  được trích xuất thành quan hệ kế thừa trong mô hình ER.

### 3 Kết luận

OWL được xây dựng bằng cấu trúc phân cấp, nhưng hầu như tất cả các thành phần của mô hình ER có thể xác định được từ OWL ontology. Trong bài báo này chúng tôi đã trình bày

tập các quy tắc nhằm trích xuất các thành phần có trong mô hình ER từ một OWL ontology cho trước. Bài báo đã xác định quy tắc trích xuất tập thực thể, các thuộc tính, các kiểu mối quan hệ giữa các tập thực thể. Bài báo cũng đã đề xuất quy tắc làm thế nào để xác định mối quan hệ phản xạ cũng như xác định rõ mối quan hệ phản xạ đối xứng hay bất đối xứng.

Nếu xem ánh xạ  $f$  là phương pháp chuyển đổi từ mô hình ER và ER mở rộng sang OWL ontology, thực nghiệm đã chứng tỏ được rằng phương pháp trích xuất mà chúng tôi giới thiệu trong bài báo này chính là ánh xạ  $f^1$  (ánh xạ ngược của ánh xạ  $f$ ). Mỗi mô hình ER kết quả  $u = f^{-1}(o)$  mà chúng tôi thu được từ OWL ontology  $o$  cho trước thỏa mãn  $f(u) = o$ .

Tuy nhiên, do một số đặc điểm của mô hình ER và ER mở rộng nên vấn đề trích xuất từ OWL2 đảm bảo tính chất “bảo toàn ngữ nghĩa” vẫn hàm chứa nhiều thách thức. Các quy tắc đề xuất là chưa đầy đủ vì có một số thành phần quan trọng của một ontology OWL như tập các tiên đề (axiom), entity axioms, key axioms và tập các biểu thức (ví dụ, class expressions, data property expressions). Trong thời gian tới, chúng tôi sẽ tiếp tục nghiên cứu để hoàn chỉnh bộ quy tắc chuyển đổi đối với việc trích xuất mô hình ER và ER mở rộng từ OWL.

### Tài liệu tham khảo

1. M. Fahad, (2008), *ER2OWL: Generating OWL Ontology from ER Diagram*, IFIP – The International Federation for Information Processing.
2. Igor Myroshnichenko, M.S., Marguerite C. Murphy, Ph.D., (2009), *Mapping ER Schemas to OWL Ontologies*, Semantic Computing. ICSC '09. IEEE International Conference, pp. 324-329.
3. Pasapitch Chujai, Nittaya Kerdprasop, Kittisak Kerdprasop, (2014), On Transforming the ER Model to Ontology Using Protégé OWL Tool, *International Journal of Computer Theory and Engineering*, vol. 6.
4. Toan Van Nguyen, Hoang Lien Minh Vo, Quang Hoang, Hanh Huu Hoang, (2016), *A new method for transforming TimeER model-based specification into OWL ontology*, in 8th Asian Conference on Intelligent Information and Database Systems ACIIDS, Da Nang, Viet Nam.
5. S Brockmans; R Colomb; P Haase; E Kendall; Evan K. Wallace; G Xie, (2006), *A Model-Driven Approach for Building OWL DL and OWL Full Ontologies*, in Proceedings of the International Semantic Web Conference (ISWC) 200.
6. S. Bagui, (2009), Mapping OWL to the Entity Relationship and Extended Entity Relationship models, *Int. J. Knowledge and Web Intelligence*, Vol. 1, Nos. 1/2.
7. Antoniou, G. and Harmelen, F.V. (2004), *A Semantic Web Primer*, Cambridge: The MIT Press, .
8. "OWL Web Ontology Language," W3C Recommendation, 10 2 2004. [Online]. Available: <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>. [Accessed 2 September 2018].
9. Elmasri R., Navathe S.B., (2015), *Fundamentals of Database Systems*, Addison-Wesley, 7th edn.
10. Võ Hoàng Liên Minh, Hoàng Quang, Hoàng Hữu Hạnh, (2017), Chuyển đổi thuộc tính đa trị phức hợp lồng nhau trên mô hình ER sang OWL Ontology, *Tạp chí Khoa học và Công nghệ, Đại học Khoa học Huế*, 2017, Tập 10, số 1, trang 27-38.

11. Võ Hoàng Liên Minh, Hoàng Quang, (2018), *Một phương pháp trích xuất biểu đồ lớp UML từ OWL ontology*, Hội thảo Quốc gia lần thứ XXI "Một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông", Thanh Hóa, trang 205-210.
12. P. Chen, (1976), *The entity-relationship model — toward a unified view of data*, *ACM Transactions on Database Systems (TODS)*, Vol. 1, Issue 1.

## SUPPORT METHODS FOR EXTRACTION OF ER MODELS FROM OWL2

Minh Hoang Lien Vo\*, Quang Hoang

University of Sciences, Hue University, 77 Nguyen Hue St., Hue, Vietnam

**Abstract.** For knowledge representation, OWL (Web Ontology Language) is a language for modelling information systems by describing the classes, properties, and relationships among these components that enable the computer to "understand" and process the information. However, the format of OWL makes it difficult to understand OWL syntax and query. Researchers have proposed mapping methods from the ER (Entity–Relationship) and EER (Enhanced Entity-Relationship) model to OWL to reuse older systems, thus reducing costs instead of having to design ontologies from scratch. An extraction of the ER and EER model from an ontology is seen as the reverse engineering of the mapping from the ER and EER model to OWL. Solving this problem will allow us to "investigate" a conceptual model used to design an ontology. To help understand the structure of OWL as well as the OWL data, this paper introduces a set of rules for extracting the ER model from OWL, which suggests adding some instances of extracting the ER model from OWL, such as key attributes, composite attributes, or recursive relationships.

**Keywords:** OWL, ontology, ER model, reverse engineering