

## **CHUYỂN ĐỔI MÔ HÌNH QUAN HỆ SANG MÔ HÌNH NGỮ NGHĨA DỰA TRÊN ONTOLOGY**

*Nguyễn Lý Hữu Huân<sup>1</sup>, Hoàng Hữu Hạnh<sup>2</sup>, Lê Mạnh Thạnh<sup>2</sup>*

*<sup>1</sup>Trường Đại học Khoa học, Đại học Huế*

*<sup>2</sup>Đại học Huế*

**Tóm tắt.** Bài báo này nghiên cứu và đề xuất một hướng tiếp cận trong việc chuyển đổi dữ liệu từ dạng cơ sở dữ liệu quan hệ (RDB) sang Ontology được biểu diễn bằng các ngôn ngữ RDF và OWL. Quá trình chuyển đổi bao gồm 2 bước chính: (i) Ánh xạ lược đồ (tạo ra các lớp từ các bảng, tạo ra các thuộc tính đối tượng (object property) từ các khóa ngoại và tạo ra thuộc tính kiểu dữ liệu (datatype property) từ các thuộc tính không tham gia vào khóa ngoại), (ii) Ánh xạ dữ liệu (tạo ra các thể hiện).

### **1. Giới thiệu**

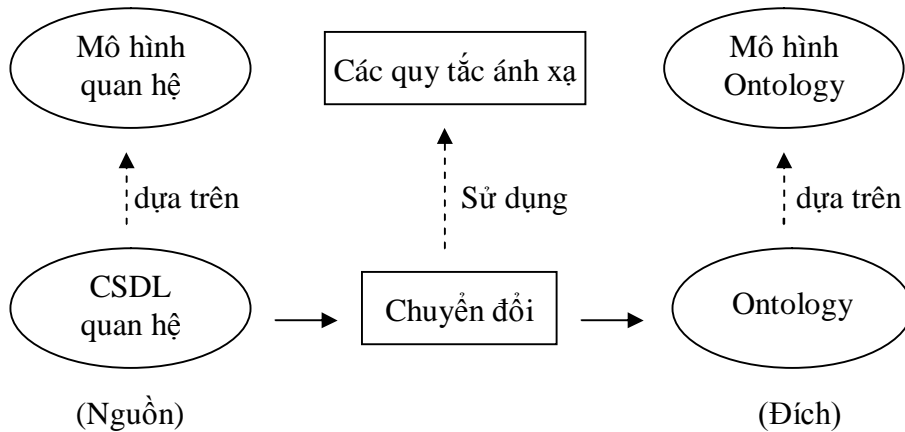
Chúng ta đã biết, hiện nay phần lớn dữ liệu trên web đang được lưu trữ trong các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) với các ưu điểm đã được chứng minh về các mặt: khả năng mở rộng, lưu trữ hiệu quả, tối ưu hóa việc thực thi các câu truy vấn, độ an toàn. Tuy nhiên, các cơ sở dữ liệu quan hệ (RDB) thường là tách biệt nhau, không đồng nhất về lược đồ, thuật ngữ, định danh và mức độ chi tiết của sự biểu diễn dữ liệu. Do đó, người ta đã sử dụng kỹ thuật RDF và Ontology nhằm cung cấp một nền tảng cho việc tích hợp tất cả các nguồn dữ liệu đó.

Hiện nay, một vấn đề đang được các nhà khoa học quan tâm và cũng là một phần quan trọng trong việc phát triển của Web ngữ nghĩa là làm thế nào để sử dụng nguồn dữ liệu khổng lồ của web hiện tại một cách có hiệu quả, nghĩa là cần có cơ chế để thực hiện việc chuyển đổi dữ liệu từ mô hình quan hệ sang mô hình dữ liệu của Web ngữ nghĩa (là Ontology được biểu diễn bằng các ngôn ngữ RDF và OWL). Hơn nữa việc chuyển đổi này phải được thực hiện một cách tự động, trong suốt.

Trong bài báo này, chúng tôi sẽ đề xuất các quy tắc để ánh xạ từ một cơ sở dữ liệu quan hệ sang RDF/OWL một cách tự động và không cần sử dụng đến Ontology đã có sẵn. Chúng tôi không sử dụng đến việc mở rộng bất cứ một ngôn ngữ Web ngữ nghĩa nào để hỗ trợ các ràng buộc, do đó sẽ không làm cho Web ngữ nghĩa trở nên rắc rối hơn và quá trình ánh xạ cũng dễ dàng được triển khai.

Hình 1 nhằm minh họa cho ý tưởng cơ bản của phương pháp này. Một cơ sở dữ

liệu quan hệ được chuyển đổi sang Ontology bằng cách sử dụng một tập các quy tắc ánh xạ. Các quy tắc này ánh xạ các cấu trúc của một mô hình quan hệ (các bảng, cột, ràng buộc, ...) với các cấu trúc của tương ứng của mô hình Ontology (các lớp, thuộc tính, giới hạn, ...). Chúng tôi sẽ áp dụng các quy tắc này vào cơ sở dữ liệu quan hệ để sinh ra Ontology. Bởi vì các quy tắc ánh xạ là xác định trên mức độ mô hình, nên nó có thể được áp dụng cho bất kỳ một cơ sở dữ liệu nào miễn là cùng dựa trên một mô hình quan hệ cho trước.



**Hình 1.** Phương pháp chuyển đổi từ CSDL quan hệ sang Ontology.

Trong phần tiếp theo, chúng tôi sẽ trình bày phương pháp chuyển đổi của mình, bao gồm bảng chuyển đổi kiểu dữ liệu, ánh xạ một số ràng buộc trên thuộc tính, và các quy tắc chuyển đổi bảng và thuộc tính trong từng trường hợp cụ thể.

## 2. Phương pháp chuyển đổi từ cơ sở dữ liệu quan hệ sang Ontology

Quá trình ánh xạ được trình bày ở đây bao gồm việc tạo ra các lớp từ các bảng, tạo ra các thuộc tính đối tượng (object property) từ các khóa ngoại và tạo ra thuộc tính kiểu dữ liệu (datatype property) từ các thuộc tính không tham gia vào khóa ngoại.

### 2.1. Bảng chuyển đổi kiểu dữ liệu

Việc chuyển đổi từ các lược đồ quan hệ sang Ontology đòi hỏi phải giữ nguyên các thông tin về kiểu dữ liệu. Không giống như trong SQL, OWL không có các kiểu dữ liệu được định nghĩa sẵn, thay vào đó, nó sử dụng các kiểu dữ liệu của XML Schema (XSD). Bảng 1 là danh sách một số kiểu dữ liệu thường dùng trong SQL cùng với kiểu dữ liệu tương ứng trong XSD.

**Bảng 1:** Ánh xạ kiểu dữ liệu từ SQL sang XSD.

Kiểu dữ liệu trong SQL	Kiểu dữ liệu trong XSD
<b>Kiểu số</b>	
Decimal, Numeric	xsd:decimal

---

Real	xsd:float
Float	xsd:double
Integer, Int	xsd:Integer, xsd:positiveInteger, xsd:negativeInteger xsd:nonPositiveInteger, xsd:nonNegativeInteger
BigInt	xsd:long
SmallInt	xsd:short
TinyInt	xsd:unsignedByte
<b>Kiểu chuỗi, ký tự</b>	
Char, VarChar, Nchar, NVarChar, Text, NText	xsd:string
<b>Kiểu ngày tháng, thời gian</b>	
DateTime	xsd:dateTime
Date	xsd:date
Time	xsd:time
<b>Các kiểu khác</b>	
Binary, VarBinary	xsd:base64Binary
Boolean	xsd:boolean
Variant	xsd:anyType

---

## ***2.2. Quy tắc ánh xạ một số ràng buộc trên thuộc tính***

Ngôn ngữ SQL hỗ trợ các ràng buộc UNIQUE, NOT NULL, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT. Tuy nhiên, không phải tất cả các ràng buộc này đều có thể được ánh xạ sang OWL. Chẳng hạn như ràng buộc DEFAULT (quy định giá trị mặc định cho một thuộc tính) là không có thành phần tương ứng trong OWL, vì vậy nó sẽ không được xét đến.

Sau đây chúng tôi sẽ đưa ra các quy tắc ánh xạ đối với một số ràng buộc trên mỗi thuộc tính của một lược đồ quan hệ. Từ đây trở đi, chúng tôi quy ước là luôn sử dụng kết hợp các quy tắc này với các quy tắc ánh xạ thuộc tính trong từng trường hợp cụ thể sẽ được bàn đến ở phần sau.

### 2.2.1. Ràng buộc *UNIQUE*

Quy tắc R1: Ràng buộc *UNIQUE* trên thuộc tính trong mô hình RDB sẽ được ánh xạ thành tính chất functional của thuộc tính tương ứng trong Ontology.

### 2.2.2. Ràng buộc *NOT NULL*

Quy tắc R2: Ràng buộc *NOT NULL* trên thuộc tính trong mô hình RDB sẽ được ánh xạ thành ràng buộc bản số cực tiểu bằng 1 trên thuộc tính tương ứng trong Ontology.

### 2.2.3. Ràng buộc *PRIMARY KEY*

Có hai hình thức của ràng buộc *PRIMARY KEY*:

- Ràng buộc *PRIMARY KEY* trên khóa đơn (ràng buộc trên một thuộc tính): trường hợp này cũng giống như thuộc tính vừa có ràng buộc *UNIQUE* vừa có ràng buộc *NOT NULL*.

- Ràng buộc *PRIMARY KEY* trên khóa hợp (trên nhiều thuộc tính): nếu ràng buộc này tham gia vào việc tạo ra các mối quan hệ đặc biệt (sự phân mảnh, kế thừa, ...) thì nó sẽ được ánh xạ bằng các quy tắc trong từng trường hợp tương ứng (sẽ được khảo sát cụ thể ở phần sau). Nếu không, trong OWL không có thành phần tương ứng nào có thể biểu diễn được ràng buộc này, do đó các thuộc tính nằm trong khóa hợp này chỉ được chuyển thành các thuộc tính kiểu dữ liệu thông thường, hoặc ta sẽ ghép các thuộc tính này thành một thuộc tính đóng vai trò khóa đơn, rồi thực hiện quy tắc sau.

Quy tắc R3: Ràng buộc *PRIMARY KEY* trên khóa đơn trong mô hình RDB sẽ được ánh xạ thành tính chất functional và ràng buộc bản số cực tiểu bằng 1 trên thuộc tính tương ứng trong Ontology.

### 2.2.4. Ràng buộc *CHECK*

- Dạng *CHECK (attr IN (val1, val2, ...))*

Quy tắc R4: Ràng buộc *CHECK* với một danh sách giá trị trên thuộc tính trong mô hình RDB sẽ được ánh xạ thành ràng buộc *owl:oneOf* và *owl:DataRange* (kiểu dữ liệu liệt kê) trên range của thuộc tính tương ứng trong Ontology. Mỗi phần tử trong kiểu dữ liệu liệt kê là một phần tử trong danh sách giá trị.

- Dạng *CHECK (attr = val)*

Quy tắc R5: Ràng buộc *CHECK* với một giá trị cụ thể trên thuộc tính trong mô hình RDB sẽ được ánh xạ thành ràng buộc *owl:hasValue* cùng với giá trị đó trên thuộc tính tương ứng trong Ontology.

- Dạng *CHECK (attr > 0)*, *CHECK (attr >= 0)*, *CHECK (attr < 0)*, *CHECK (attr <= 0)*

Quy tắc R6: Các ràng buộc *CHECK* dạng này sẽ được sử dụng để ánh xạ kiểu

dữ liệu (range) của thuộc tính tương ứng, cụ thể:

- CHECK ( $attr > 0$ ) tương ứng *xsd;positiveInteger*
- CHECK ( $attr \geq 0$ ) tương ứng *xsd;nonNegativeInteger*
- CHECK ( $attr < 0$ ) tương ứng *xsd;negativeInteger*
- CHECK ( $attr \leq 0$ ) tương ứng *xsd;nonPositiveInteger*
- Dạng CHECK ( $attr > val1$ ), CHECK ( $attr < val2$ ), CHECK ( $val1 < attr < val2$ ), ..., với  $val1 \neq 0$  và  $val2 \neq 0$

Hiện tại OWL chưa hỗ trợ biểu diễn các ràng buộc dạng này, do đó các ràng buộc dạng này sẽ được bỏ qua khi ánh xạ thuộc tính.

### 2.2.5. Ràng buộc FOREIGN KEY và REFERENCES

Đối với các thuộc tính khóa ngoại tham gia vào việc tạo thành các mối quan hệ đặc biệt (phân mảnh, kế thừa, ...), chúng sẽ được ánh xạ bằng các quy tắc trong từng trường hợp tương ứng. Còn đối với các thuộc tính khóa ngoại thông thường, vì yêu cầu các giá trị trên thuộc tính khóa ngoại phải tồn tại trong bảng được tham chiếu, do đó thuộc tính tương ứng phải có ràng buộc bản số cực tiểu bằng 1.

Quy tắc R7: Ràng buộc FOREIGN KEY trên các thuộc tính khóa ngoại thông thường trong mô hình RDB sẽ được ánh xạ thành ràng buộc bản số cực tiểu bằng 1 trên thuộc tính tương ứng trong Ontology.

## 2.3. Các quy tắc ánh xạ lớp và thuộc tính

### 2.3.1. Các quy tắc đối với sự phân mảnh dữ liệu

Các bảng được xem là tạo thành sự phân mảnh dọc nếu có ít nhất là hai bảng có cùng khóa chính, trong đó có đúng một bảng được gọi là bảng chính. Khóa chính của bảng chính không phải là khóa ngoại, trong khi đó khóa chính của các bảng còn lại cũng là khóa ngoại tham chiếu đến bảng chính. Ngoài ra, tất cả các giá trị của khóa chính trong bảng chính phải tồn tại ở trong tất cả các bảng còn lại.

#### \* Ánh xạ lớp

Quy tắc C1: Các bảng tham gia vào sự phân mảnh sẽ được gộp chung thành một lớp trong Ontology.

#### \* Ánh xạ các thuộc tính đối tượng

Quy tắc OP1: Tất cả các thuộc tính khóa ngoại mà đồng thời cũng là khóa chính trong các bảng phân mảnh thì sẽ không được ánh xạ thành các thuộc tính đối tượng.

### 2.3.2. Các quy tắc đối với sự kế thừa

Các bảng được xem là tạo thành sự kế thừa nếu có hai bảng có cùng khóa chính,

trong đó có một bảng gọi là bảng chính. Khóa chính của bảng chính không phải là một khóa ngoại, trong khi đó khóa chính của bảng còn lại cũng là khóa ngoại tham chiếu đến bảng chính. Ngoài ra không bắt buộc tất cả các giá trị khóa chính trong bảng chính phải tồn tại trong bảng kia.

### \* **Ánh xạ lớp**

Quy tắc C2: Nếu bảng  $r_2$  là kế thừa từ bảng  $r_1$  thì khi đó ta sẽ xây dựng lớp  $c_1$  tương ứng với  $r_1$ , lớp  $c_2$  tương ứng với  $r_2$  và  $c_2$  sẽ là một lớp con của  $c_1$ .

#### 2.3.3. Các quy tắc đối với bảng biểu diễn thuộc tính đa trị

Điều kiện để xác định bảng biểu diễn thuộc tính đa trị là bảng chỉ có đúng 2 thuộc tính, trong đó có một thuộc tính là khóa ngoại, còn thuộc tính còn lại không phải là khóa ngoại. Đồng thời khóa chính của bảng là khóa hợp của 2 thuộc tính này.

### \* **Ánh xạ lớp**

Quy tắc C3: Bảng dùng để biểu diễn một thuộc tính đa trị sẽ không được ánh xạ thành lớp trong Ontology, ta sẽ gộp chung bảng chính và bảng đa trị thành một lớp bằng cách biểu diễn thuộc tính đa trị trong lớp tương ứng với bảng chính.

### \* **Ánh xạ các thuộc tính đối tượng**

Quy tắc OP2: Thuộc tính khóa ngoại trong bảng đa trị sẽ không được ánh xạ thành thuộc tính đối tượng trong Ontology.

### \* **Ánh xạ các thuộc tính kiểu dữ liệu**

Quy tắc DP1: Thuộc tính đa trị ở trong bảng đa trị sẽ được chuyển thành một thuộc tính kiểu dữ liệu có domain là lớp tương ứng với bảng chính và range là kiểu dữ liệu liệt kê với ràng buộc *owl:someValuesFrom*, tập các phần tử của danh sách liệt kê là miền trị của thuộc tính đa trị.

#### 2.3.3. Các quy tắc đối với bảng biểu diễn mối quan hệ nhị nguyên nhiều-nhiều

*Bảng biểu diễn mối quan hệ nhị nguyên* là bảng chỉ có đúng 2 thuộc tính, và cả hai thuộc tính đó đều là khóa ngoại tham chiếu đến hai bảng khác.

### \* **Ánh xạ lớp**

Quy tắc C4: Bảng biểu diễn mối quan hệ nhị nguyên (không có thuộc tính định kèm) sẽ không được chuyển thành lớp trong Ontology.

### \* **Ánh xạ các thuộc tính đối tượng**

Quy tắc OP3: Đối với bảng  $r$  biểu diễn mối quan hệ nhị nguyên giữa hai bảng  $s$  và  $t$ , ta có thể tạo ra hai thuộc tính đối tượng đảo nhau  $P$  và  $P'$ .  $P$  có domain là  $c_s$  và range là  $c_t$ ,  $P'$  có domain là  $c_t$  và range là  $c_s$ , trong đó  $c_s$  và  $c_t$  là các lớp tương ứng với các bảng  $s$  và  $t$ .

#### 2.3.4. Các quy tắc đối với các trường hợp còn lại

Trường hợp này bao gồm các bảng không có khóa ngoại; các bảng có khóa ngoại không đồng thời là khóa chính (các khóa ngoại thông thường); các bảng chứa đúng hai khóa ngoại, hai khóa ngoại này đồng thời là khóa chính, nhưng ngoài ra còn có thêm các thuộc tính không nằm trong khóa ngoại (trường hợp này biểu diễn cho mỗi quan hệ nhị nguyên có thuộc tính đính kèm); các bảng có nhiều hơn hai khóa ngoại (mỗi quan hệ đa nguyên, bảng biểu diễn cho một tập thực thể hoặc mỗi quan hệ đa nguyên có thuộc tính đính kèm).

##### \* **Ánh xạ lớp**

Quy tắc C5: Các bảng chưa được ánh xạ bằng các quy tắc C1, C2, C3 và C4 sẽ được xây dựng thành một lớp trong Ontology.

##### \* **Ánh xạ các thuộc tính đối tượng**

Ta sẽ xét tất cả các khóa ngoại chưa được ánh xạ bằng các quy tắc OP1, OP2 và OP3.

Quy tắc OP4.1 (đối với mỗi quan hệ 1-1): Các thuộc tính khóa ngoại trong trường hợp này sẽ được chuyển thành các thuộc tính đối tượng, với domain là lớp tương ứng của bảng chứa thuộc tính, range là lớp tương ứng của bảng được tham chiếu. Đồng thời ta cũng định nghĩa một thuộc tính đối tượng đảo của nó, và cả hai đều có ràng buộc bản số cực đại bằng 1.

Quy tắc OP4.2 (đối với mỗi quan hệ 1-nhiều): Các thuộc tính khóa ngoại trong trường hợp này sẽ được chuyển thành thuộc tính đối tượng, với domain là lớp tương ứng của bảng chứa thuộc tính, range là lớp tương ứng của bảng được tham chiếu, có ràng buộc bản số cực đại bằng 1. Ta cũng tạo ra một thuộc tính đối tượng đảo của nó không có ràng buộc về bản số cực đại.

Đối với mỗi quan hệ nhiều-nhiều: trường hợp không có thuộc tính đính kèm đã được ánh xạ bằng quy tắc OP3, còn trong trường hợp có thuộc tính đính kèm, bảng sẽ được xem như là biểu diễn một tập thực thể, và lúc đó khóa ngoại sẽ được chuyển đổi bằng quy tắc OP4.2.

##### \* **Ánh xạ các thuộc tính kiểu dữ liệu**

Quy tắc DP2: Các thuộc tính không tham gia vào khóa ngoại và không áp dụng được quy tắc DP1 thì sẽ được ánh xạ thành các thuộc tính kiểu dữ liệu có domain là lớp tương ứng của bảng chứa thuộc tính và range là kiểu dữ liệu tương ứng của chúng trong OWL.

#### 2.4. Quy tắc ánh xạ các thể hiện

Bước đầu tiên là gán một định danh duy nhất cho mỗi thể hiện. Giá trị phù hợp nhất cho định danh này là giá trị của thuộc tính khóa chính. Đặc biệt, định danh của các

thuộc tính đối tượng biểu diễn mối quan hệ nhiều-nhiều được tạo thành bằng cách ghép nối các khóa chính của cả hai phía trong mỗi quan hệ.

Bước thứ hai là tạo ra các thể hiện cho các lớp từ dữ liệu ở các bảng tương ứng. Với một lớp Ontology  $c$ , nếu  $c$  là tương ứng với các bảng  $r_1, r_2, \dots, r_n$  trong cơ sở dữ liệu, thì mọi bộ dữ liệu  $t_i, t_i \in r_1 \bowtie r_2 \bowtie \dots \bowtie r_n$  có thể được ánh xạ thành một thể hiện của  $c$ .

### 2.5. Nhận xét

Chúng tôi thấy rằng phương pháp được đề xuất trên đây đã đáp ứng được các nội dung cơ bản sau:

Việc suy luận trên các Ontology là một mục tiêu chủ yếu của Web ngữ nghĩa, do đó kết quả chuyển đổi phải đảm bảo khả năng quyết định cho một hệ thống suy luận OWL. OWL-DL là một ngôn ngữ con của OWL bảo đảm khả năng quyết định, do đó các quy tắc chuyển đổi phải tạo ra được OWL-DL.

Các quy tắc chuyển đổi được đặc tả hình thức nhằm tránh việc nhập nhằng trong cú pháp hay ngữ nghĩa. Đồng thời, các quy tắc được định nghĩa trong các hệ thống hình thức như logic bậc nhất có thể dễ dàng triển khai trong các ngôn ngữ như Prolog hay Datalog.

Khi xây dựng các quy tắc, chúng tôi đã chú ý đến việc tránh sự ảnh hưởng của các ví dụ trên một miền cụ thể. Hệ thống chuyển đổi chỉ được tiến hành trích xuất ngữ nghĩa trong các ngôn ngữ định nghĩa lược đồ. Đôi khi sự ảnh hưởng của các ví dụ trong một miền đặc biệt có thể dẫn đến sai lầm trong các quy tắc.

## 3. Kết luận và hướng phát triển

Trên đây chúng tôi đã đề xuất một phương pháp nhằm chuyển đổi tự động từ các lược đồ quan hệ được biểu diễn bằng SQL-DDL sang OWL Ontology. Khi định nghĩa các quy tắc, chúng tôi cũng đã bảo đảm khả năng tương thích với các ngôn ngữ OWL dựa trên logic mô tả (như OWL DL).

Phương pháp của chúng tôi đã phân biệt giữa sự phân mảnh và mối quan hệ kế thừa. Đồng thời chúng tôi cũng đã giải quyết được các mối quan hệ nhị nguyên, mối quan hệ đa nguyên, các thuộc tính đa trị. Các vấn đề này đã không được giải quyết hoàn toàn trong một số phương pháp tiếp cận trước đây.

Tuy nhiên quá trình chuyển đổi này cũng còn gặp phải một số nhược điểm khi phải xác định sự kế thừa và một số ngữ nghĩa khác. Các nhược điểm này xuất phát từ sự hạn chế về khả năng biểu cảm của SQL-DDL, và nó sẽ thể hiện khi ta so sánh Ontology được tạo ra bởi hệ thống với Ontology được tạo ra bởi các chuyên gia.

Trong tương lai, chúng tôi sẽ tiếp tục nghiên cứu để có thể xử lý các nguồn dữ liệu đa tạp. Chúng tôi cũng sẽ giảm bớt các điểm hạn chế của hệ thống bằng cách nhận



ra những trường hợp cần phải có sự can thiệp của con người và từ đó phát triển thành những giải pháp chuyển đổi bán tự động.

Một vấn đề nữa mà chúng tôi sẽ tiếp tục nghiên cứu là nhằm hỗ trợ các ràng buộc động của SQL như các TRIGGER dạng ON DELETE UPDATE, ON DELETE CASCADE.

## TÀI LIỆU THAM KHẢO

1. Astrova I., Kalja A., *Towards the Semantic Web: Extracting OWL Ontologies from SQL Relational Schemata*, Proceedings of IADIS International Conference WWW/Internet, (2006), 62–66.
2. Astrova I., Korda N. and Kalja A., *Rule-Based Transformation of SQL Relational Databases to OWL Ontologies*, Proceedings of the 2<sup>nd</sup> International Conference on Metadata & Semantics Research, 2007.
3. Benslimane S., Benslimane D. and Malki M., *Acquiring OWL Ontologies from Data-Intensive Web Sites*, USA. ACM, Palo Alto, California, 2006.
4. Blakeley C., *RDF Views of SQL Data (Declarative SQL Schema to RDF Mapping)*, OpenLink Software, 2007.
5. Buccella A., Penabad M., Rodriguez F., Farina A., Cechich A., *From Relational Databases to OWL Ontologies*, Proceedings of the 6<sup>th</sup> National Russian Research Conference, 2004.
6. Green J., Dolbear C., Hart G., Engelbrecht P., Goodwin J., *Creating a semantic integration system using spatial data*, International Semantic Web Conference, Karlsruhe, Germany, 2008.
7. Hu W., Qu Y., *Discovering Simple Mappings Between Relational Database Schemas and Ontologies*, Proceedings of 6<sup>th</sup> International Semantic Web Conference (ISWC 2007), 2<sup>nd</sup> Asian Semantic Web Conference (ASWC 2007), LNCS 4825, Busan, Korea, (2007), 225-238.
8. Li M., Du X., Wang S., *A Semi-automatic Ontology Acquisition Method for the Semantic Web*, Proceedings of Advances in Web-Age Information Management 6<sup>th</sup> International Conference, WAIM 2005, Hangzhou, China, (2005), 209-220.
9. Li M., Du X., Wang S., *Learning ontology from Relational Database*, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, China, (2005), 18-21.
10. Sahoo S. S., Bodenreider O., Rutter J., Skinner K., Sheth A., *An ontology-driven semantic mash-up of gene and biological pathway information: Application to the domain of nicotine dependence*, Journal of Biomedical Informatics (Special Issue:

Semantic Biomedical Mashups), 2008.

11. Stojanovic L., Stojanovic N., Volz R., *Migrating data-intensive Web Sites into the Semantic Web*, Proceedings of the 17<sup>th</sup> ACM Symposium on Applied Computing (SAC 2002), Madrid, Spain, (2002), 1100-1107.
12. Tirmizi S., Sequeda J., Miranker D., *Translating SQL Applications to the Semantic Web*, Lecture Notes in Computer Science, Volume 5181/2008 (Database and Expert Systems Applications), 2008.

## **TRANSFORMATION OF RELATIONAL MODELS TO ONTOLOGICAL MODELS**

*Nguyen Ly Huu Huan<sup>1</sup>, Hoang Huu Hanh<sup>2</sup>, Le Manh Thanh<sup>2</sup>*

<sup>1</sup>*College of Sciences, Hue University*

<sup>2</sup>*Hue University*

**Abstract.** This paper presents an approach in the transformation of relational database models (RDB) to ontological models represented in RDF and OWL languages. The transformation consists of two main phases: (i) Schema mapping (creating classes from tables, object properties from foreign keys, datatype properties from normal properties) and (ii) Data mapping (creating instances).