

CHUYỂN ĐỔI BIỂU ĐỒ LỚP UML SANG OWL ONTOLOGY

Võ Hoàng Liên Minh¹, Hoàng Quang¹, Hoàng Hữu Hạnh²

¹ Trường Đại học Khoa học, Đại học Huế, ² Đại học Huế

minhvh1@gmail.com, hquang10@gmail.com, hhhanh@hueuni.edu.vn

TÓM TẮT: Để biểu diễn tri thức, OWL ontology nổi lên như một ngôn ngữ để mô hình hóa khái niệm các hệ thống thông tin phức tạp. Nhiều hệ thống cũ trước đây được xây dựng dựa trên biểu đồ lớp UML, vì thế việc nâng cấp và chuyển đổi các biểu đồ lớp UML sang ontology nhằm sử dụng lại các hệ thống cũ, giúp giảm chi phí là thực sự cần thiết. Đã có nhiều nghiên cứu nhằm thực hiện việc chuyển đổi từ các mô hình cơ sở dữ liệu mức khái niệm sang ontology. Trên cơ sở kế thừa các phương pháp chuyển đổi mô hình ER sang OWL ontology, bài báo này tập trung vào việc đề xuất xây dựng một phương pháp chuyển đổi biểu đồ lớp UML sang OWL ontology.

Từ khóa: Semantic Web, OWL ontology, mô hình dữ liệu mức khái niệm, biểu đồ lớp UML.

I. GIỚI THIỆU

Biểu đồ lớp UML là mô hình dữ liệu mức khái niệm thường được sử dụng phổ biến cho việc thiết kế các hệ thống thông tin quản lý hiện nay. Ưu điểm của thiết kế hệ thống bằng UML là khả năng mô tả và phản ánh tốt thế giới thực của các hệ thống thông tin. Ngoài ra nó còn được sự hỗ trợ mạnh mẽ từ các nhà phát triển (Rational Rose, Enterprise Architect...) nên việc thiết kế UML đã trở nên phổ biến.

Trong những năm gần đây, ontology đã trở thành một thuật ngữ được biết đến trong nhiều lĩnh vực của khoa học máy tính để giúp con người và máy tính có thể cùng nhau làm việc, giúp máy tính có thể hiểu và có khả năng xử lý thông tin hiệu quả hơn. Về cơ bản, ontology cung cấp bộ từ vựng để mô tả dữ liệu với ngữ nghĩa mà máy tính có thể hiểu được. Có nhiều ngôn ngữ được thiết kế để biểu diễn ontology cho web ngữ nghĩa. Trong các ngôn ngữ này, RDFS và OWL là hai ngôn ngữ được sử dụng phổ biến nhất. OWL được xem là một mở rộng của RDFS nhằm khắc phục các nhược điểm của RDFS. Hiện tại, OWL đang được xem là ngôn ngữ chuẩn để biểu diễn ontology cho web ngữ nghĩa. OWL là ngôn ngữ mô tả các lớp, các thuộc tính và các quan hệ giữa các đối tượng này theo cách mà máy có thể hiểu được. OWL1 được phân thành ba loại khác nhau với khả năng biểu diễn ngữ nghĩa tăng dần: OWL Lite cung cấp các lớp, các thuộc tính có phân cấp, các ràng buộc đơn giản với đủ khả năng biểu đạt một ontology đơn giản. OWL DL tăng khả năng biểu đạt và tính quyết định của vấn đề phân lớp, cho phép hỗ trợ suy diễn hiệu quả. OWL DL cung cấp tất cả các cú pháp OWL dưới các giới hạn xác định. OWL Full là ngôn ngữ đầy đủ, không có các giới hạn, nhưng nó không đưa ra khả năng quyết định.

OWL2 được xem là một mở rộng của OWL DL bằng cách thêm vào các cú pháp biểu diễn mà vẫn không làm mất khả năng quyết định. Đặc biệt, OWL2 có khả năng biểu diễn khóa của một lớp trong ontology tương tự như thuộc tính khóa trong biểu đồ lớp UML. Để biên tập ontology, trường Đại học Stanford đã giới thiệu công cụ mã nguồn mở Protégé, cho phép định nghĩa các khái niệm ontology như lớp, thuộc tính, phân loại ngữ nghĩa và các ràng buộc hạn chế khác nhau.

Đã có nhiều nghiên cứu về việc chuyển đổi các mô hình mức khái niệm sang OWL ontology. Trong đó, các nghiên cứu về việc chuyển đổi từ mô hình ER sang ontology có: Upadhyaya và P. S. Kumar (2005) [1] giới thiệu công cụ ERONTO sử dụng Java và Jena 2.1 để trích xuất một ontology từ lược đồ ER. M. Fahad (2008) [2] đã đề xuất một phương pháp thiết kế các OWL Ontology từ mô hình ER dựa vào tập các quy tắc chuyển đổi các thành phần của một mô hình ER (các tập thực thể, các thuộc tính, và các mối quan hệ giữa các tập thực thể) thành các thành phần tương ứng trên OWL. I. Myroshnichenko (2009) [3] đã trình bày một giải pháp chuyển đổi tự động từ mô hình ER thành ngữ nghĩa tương đương trên OWL Lite Ontology. Toan Van Nguyen và nhóm tác giả [4] đã giới thiệu các quy tắc chuyển đổi mô hình EER sang OWL ontology.

Liên quan đến việc chuyển đổi các biểu đồ lớp UML sang OWL ontology có các tác giả như Dragan Gašević và nhóm tác giả [5] [6], Sara Brockmans và nhóm tác giả [7], David de Almeida Ferreira và Alberto Manuel Rodrigues da Silva [11] phân tích tính khả thi và để mô hình hoá một ontology bằng cách sử dụng các công cụ UML CASE phổ biến.

Noredine Gherabi và Mohamed Bahaj [8] mô tả làm thế nào UML có thể được chuyển đổi thành OWL Ontology, từ đó cho phép lập luận trên chúng bằng các ứng dụng web có ngữ nghĩa. Imants Zarembo và Sergejs Kodors [9] đề xuất chuyển đổi tự động của OWL 2 ontology từ mô hình dữ liệu của sơ đồ lớp UML ISO 19.103. Jesper Zedlitz và nhóm tác giả [10] [11] [12] đã phân tích sự khác nhau giữa UML và OWL, đồng thời trình bày sự chuyển đổi giữa biểu đồ lớp UML và OWL 2 ontology.

Oussama và nhóm của mình [13] đã so sánh chi tiết và toàn diện về sự khác biệt giữa lược đồ lớp UML và OWL, phân tích các phương pháp chuyển đổi mà các tác giả đã nghiên cứu và đề xuất trước đây, từ đó đề xuất quy tắc

cho việc ánh xạ UML thành OWL2 trực tiếp cho các trường hợp của biểu đồ lớp UML như thừa kế, các kiểu dữ liệu, các loại quan hệ (lớp kết hợp, kết hợp đa nguyên N-ary, phân xạ, phụ thuộc và kết hợp đơn giản ...).

Tuy nhiên các nghiên cứu này chưa hình thức hóa một cách đầy đủ các quy tắc chuyển đổi trên tất cả các thành phần của biểu đồ lớp UML. Vì vậy, bài báo này tập trung vào việc xây dựng một hệ các quy tắc nhằm chuyển đổi biểu đồ lớp UML bất kỳ sang OWL ontology.

Theo đó, cấu trúc của bài báo là như sau. Mục tiếp theo phân tích sự tương đồng giữa mô hình EER và biểu đồ lớp UML với OWL ontology và xác định các tính năng ngôn ngữ không phù hợp, theo đó mục III trình bày phương pháp chuyển đổi biểu đồ lớp UML sang OWL ontology. Mục IV là phần kết luận và thảo luận về hướng nghiên cứu tiếp theo.

II. SỰ TƯƠNG ĐỒNG GIỮA MÔ HÌNH EER, BIỂU ĐỒ LỚP UML VÀ OWL2

Đã có nhiều nghiên cứu nhằm thực hiện việc chuyển đổi từ các mô hình cơ sở dữ liệu mức khái niệm sang ontology. Mô hình EER, biểu đồ lớp UML và OWL2 được thiết kế với các mục đích khác nhau, tuy nhiên chúng có nhiều điểm tương đồng. Trên cơ sở kế thừa các phương pháp chuyển đổi mô hình ER sang OWL ontology [3] [2] [1] [4], để có thể đưa ra được quy tắc chuyển đổi, trước hết cần có cái nhìn tổng quát về sự khác nhau cơ bản giữa mô hình EER, biểu đồ lớp UML và OWL2.

Theo tác giả [14], mô hình EER hầu như là tương đồng với biểu đồ lớp UML với các thực thể/lớp, thuộc tính và các mối quan hệ. Tuy nhiên vẫn còn một số khác biệt giữa mô hình EER và biểu đồ lớp UML. Tương tự như thế, giữa biểu đồ lớp UML và OWL ontology có sự tương đồng, tuy nhiên vẫn có những khác biệt giữa biểu đồ lớp UML và OWL mà cần làm rõ để trong quá trình chuyển đổi được chính xác hơn.

Trong UML, mỗi phần tử có một tên duy nhất trong gói, tức là tên của các thuộc tính phải là duy nhất chỉ trong lớp mà nó thuộc về. Ngược lại, trong OWL tất cả các tên là toàn cục [11].

UML được định hướng theo mô hình hóa dữ liệu và xây dựng hệ thống để ngay cả khi được sử dụng để tạo ra một mô hình khái niệm của một miền. Vì vậy UML sử dụng giả thiết thế giới đóng (Closed-World Assumption - CWA), tức là tất cả các phát biểu chưa được đề cập một cách rõ ràng thì là sai. Ngược lại, OWL2 lại sử dụng giả thiết thế giới mở Open-World Assumption (OWA), thì có nghĩa thông tin thiếu được coi là chưa quyết định. Vì vậy trong một số trường hợp cần thiết, các ngữ nghĩa khác nhau được thêm bằng các ràng buộc khác nhau vào ontology trong quá trình chuyển đổi từ biểu đồ lớp UML sang OWL 2 ontology để bảo toàn ngữ nghĩa ban đầu của mô hình [11] [15].

UML sử dụng giả thiết tên duy nhất (Unique Name Assumption - UNA) tức là hai phần tử với tên gọi khác nhau đang được xem là khác nhau. OWL không sử dụng UNA, vì vậy cần phải đánh dấu một cách rõ ràng các phần tử khi chúng là khác nhau [11] [15].

OWL2 không thể biểu diễn được cấu trúc tương tự như khái niệm “*profiles*” cũng như lớp trừu tượng trong UML. Ngược lại OWL2 không thể xác định rằng một lớp không trực tiếp chứa bất kỳ cá thể nào. Để có thể biểu diễn ngữ nghĩa của một lớp trừu tượng phải sử dụng khai báo *DisjointUnion*. Điều này sẽ đảm bảo rằng bất kỳ cá thể thuộc về một lớp con cũng sẽ thuộc về lớp cha trừu tượng.

Trong UML, thuộc tính của lớp có đặc tính quản lý khả năng truy cập đối với các đối tượng khác thông qua các thuộc tính quản lý sự truy cập. Phạm vi của các thuộc tính có thể được thiết lập bằng cách đánh dấu chúng là “*public*”, “*private*”... OWL2 không có quản lý truy cập trên các thuộc tính như biểu đồ lớp UML. Sự khác biệt giữa UML và OWL2 là trong OWL2, thuộc tính được xác định là có phạm vi và miền là “*Thing*”, vì vậy một thuộc tính có thể áp dụng cho bất kỳ lớp nào, nên phạm vi là toàn cục. Trong UML, một thuộc tính là được giới hạn trong lớp con mà nó được định nghĩa [9].

Trong OWL2 có thể xác định thuộc tính đối tượng ở mức ontology. Việc khai báo kết nối đến các lớp bằng cách định nghĩa miền và phạm vi có thể không cần thiết. Một thuộc tính đối tượng kết nối giữa các cá thể với nhau. Tuy nhiên có một số trường hợp thuộc tính đối tượng không thiết lập miền và phạm vi, do đó miền và phạm vi có giá trị mặc định là *owl:Thing*. Lúc đó thuộc tính đối tượng đó có thể được sử dụng để kết nối tới bất kỳ hai cá thể nào, bởi vì mỗi cá thể đó đều thuộc lớp *owl:Thing* [10].

Và trong OWL2 cũng không thể biểu diễn thao tác trên các đối tượng. Vì thế trong quá trình chuyển đổi không xét đến phần điều khiển truy cập và các phép toán. Cả hai ngôn ngữ hỗ trợ cấu trúc mô đun, được gọi là gói trong UML và ontology trong OWL2. Chuyển đổi một gói trong UML sang ontology được thực hiện bằng cách ánh xạ gói UML thành ontology [9], [16].

III. CHUYỂN ĐỔI BIỂU ĐỒ LỚP UML SANG OWL ONTOLOGY

Khi ngôn ngữ mô hình hóa thống nhất UML được tổ chức OMG công nhận là chuẩn công nghiệp trong việc xây dựng và thiết kế các hệ thống phần mềm thì phương pháp phân tích và thiết kế hướng đối tượng đã trở nên phổ biến và hữu dụng. Để chuyển đổi mô hình dữ liệu được mô tả bởi biểu đồ lớp UML thành OWL, tập các quy tắc chuyển đổi

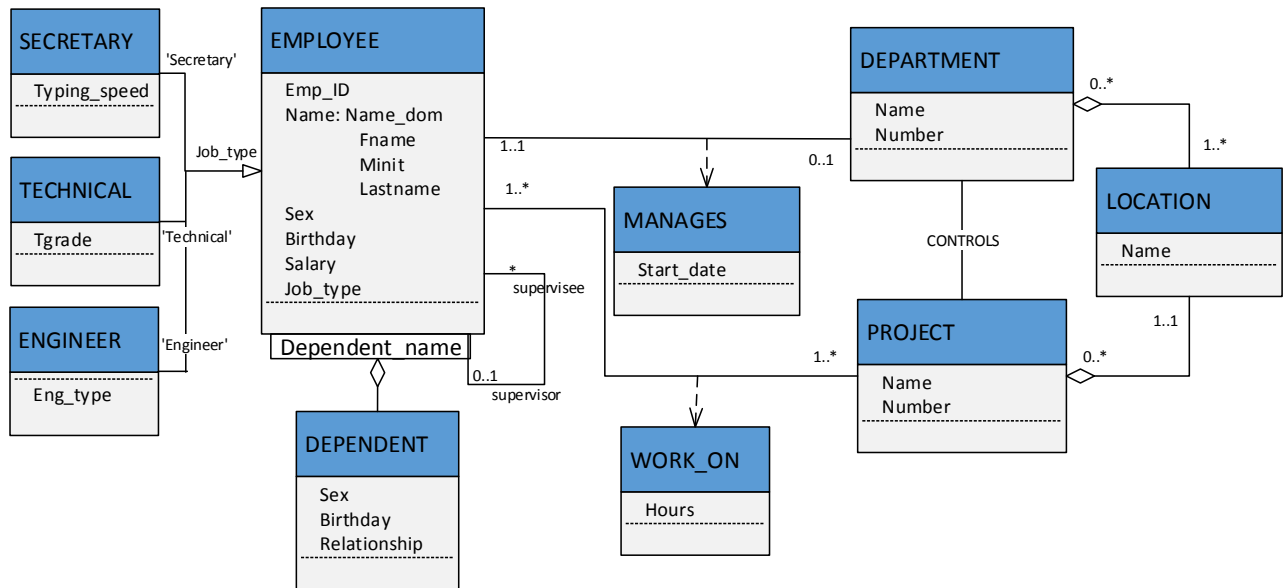
cần phải được xác định. Trong phần này, chúng tôi trình bày các quy tắc chuyển đổi biểu đồ lớp UML sang OWL. Trong các quy tắc chuyển đổi này, các thành phần trên biểu đồ lớp UML sẽ được chuyển đổi thành các thành phần tương ứng trong OWL sao cho các thành phần này phải biểu diễn được dữ liệu và các ràng buộc trên UML.

A. Lớp

Lớp mô tả cho một tập các đối tượng có cùng cấu trúc, hành vi và các mối quan hệ. Lớp có thể có các thuộc tính xác định cấu trúc và thao tác xác định hành vi của các thể hiện của lớp. UML và OWL2 đều có khái niệm lớp. Lớp trong OWL2 là một tập rỗng hoặc có nhiều thể hiện và không xác định thao tác trên các thể hiện đó. Lớp trong UML là cấu trúc tổng quát hơn, chứa một tập hợp các thể hiện [9]. Vì thế việc chuyển đổi quy tắc chuyển đổi như sau:

Quy tắc UML1. Với mỗi lớp UML E được chuyển đổi thành lớp $C(E)$ tương ứng trong OWL2, thiết lập tính chất *DisjointClasses* cho mỗi cặp lớp nếu chúng không phải là quan hệ tổng quát hóa [11]. Nếu lớp đối tượng E không có thuộc tính khóa thì bổ sung thuộc tính khóa E_ID cho lớp $C(E)$, thiết lập tính chất hàm và bản số là 1.

Trong Hình 1, lớp đối tượng *Employee* được chuyển thành lớp *Employee* trong OWL như ở Hình 2.



Hình 1. Ví dụ lược đồ lớp UML

B. Thuộc tính

Thuộc tính trong UML phải là duy nhất trong ngữ cảnh của lớp. Thuộc tính có thể có nhiều mức độ phạm vi sử dụng khác nhau, mô tả thuộc tính đó có thể được truy xuất từ các lớp khác. Với OWL, thuộc tính dữ liệu không hỗ trợ phạm vi truy xuất, vì thế chúng tôi chỉ xem xét trên mức độ tổng quát. Vì vậy, chuyển đổi thuộc tính trong UML sang OWL ontology tùy thuộc vào kiểu của thuộc tính. Nếu kiểu của thuộc tính là kiểu dữ liệu nguyên thủy, các thuộc tính sẽ được chuyển đổi thành thuộc tính dữ liệu trong OWL, nếu kiểu của thuộc tính là lớp, thuộc tính đó được chuyển đổi thành thuộc tính đối tượng. Ta có quy tắc như sau:

Quy tắc UML2. Chuyển đổi thuộc tính $attE$ của lớp đối tượng E có kiểu dữ liệu nguyên thủy trong UML thành thuộc tính kiểu dữ liệu $attE$ tương ứng trong lớp $C(E)$, có phạm vi là kiểu dữ liệu tương ứng trong OWL và miền là lớp $C(E)$, đồng thời thiết lập tính chất hàm [11] [13]. Với thuộc tính khóa KE , chuyển đổi thành thuộc tính kiểu dữ liệu KE trong tập thuộc tính khóa của lớp $C(E)$, thiết lập tính chất hàm và bản số là 1.

Như ở Hình 1, thuộc tính *Sex*, *Birthday* và *Salary* được chuyển đổi thành thuộc tính kiểu dữ liệu tương ứng như trong Hình 2. Thuộc tính khóa *Emp_ID* được chuyển thành thuộc tính kiểu dữ liệu với ràng buộc bản số là 1.

Quy tắc UML3. Chuyển đổi thuộc tính $attE$ của lớp đối tượng E có kiểu dữ liệu là lớp F trong UML thành thuộc tính đối tượng $attE$ tương ứng trong lớp $C(E)$, có miền là lớp $C(E)$ và phạm vi là lớp $C(F)$ tương ứng.

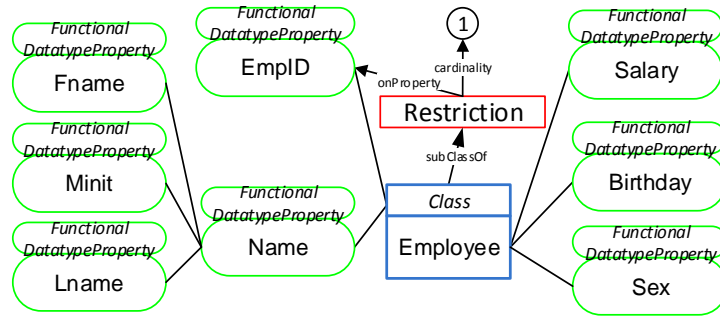
Ở ví dụ Hình 3, thuộc tính $att2$ có kiểu dữ liệu là lớp *Class2* được chuyển đổi thành thuộc tính đối tượng $att2$ với miền là lớp *Class1* và phạm vi là lớp *Class2*.

C. Thuộc tính có cấu trúc

Quy tắc UML4. Thuộc tính có cấu trúc $attE$ của lớp đối tượng E với tập các thuộc tính thành phần là sub_attE được chuyển đổi thành thuộc tính kiểu dữ liệu $attE$ của $C(E)$. Các thuộc tính thành phần sub_attE được chuyển đổi

thành các thuộc tính kiểu dữ liệu sub_attE và là thuộc tính con của thuộc tính kiểu dữ liệu $attE$, có tính chất hàm, miền là thuộc tính dữ liệu $attE$ và phạm vi là kiểu dữ liệu tương ứng trong OWL.

Trong ví dụ Hình 1, các thuộc tính thành phần $Fname$, $Minit$, $Lname$ của thuộc tính có cấu trúc $Name$ được chuyển đổi thành các thuộc tính kiểu dữ liệu tương ứng như Hình 2.



Hình 2. Ví dụ chuyển đổi thuộc tính của lớp đối tượng

D. Chuyển đổi quan hệ giữa các lớp

Quan hệ là kết nối ngữ nghĩa giữa các lớp đối tượng, trong đó thể hiện mối liên quan về các thuộc tính, các thao tác của chúng với nhau trong hệ thống. Trong UML có các loại quan hệ khác nhau là quan hệ kết hợp, quan hệ kết tập, quan hệ kế thừa và quan hệ phụ thuộc.

1. Quan hệ kết hợp

Trong UML, quan hệ kết hợp có thể là một chiều hoặc là vô hướng. Một quan hệ kết hợp có thể được chuyển đổi thành thuộc tính đối tượng trong OWL. Đối với quan hệ kết hợp hai chiều sẽ được chuyển đổi thành hai thuộc tính đối tượng cho mỗi hướng. Để đảm bảo rằng cả hai thuộc tính đối tượng này là một phần của quan hệ kết hợp, vì thế thiết lập tính chất ngược nhau [11] [13].

Quy tắc UML5. Xét mỗi quan hệ kết hợp R giữa hai lớp đối tượng E_1 và E_2 , ta có quy tắc chuyển đổi như sau:

- Bổ sung hai thuộc tính đối tượng ngược nhau biểu diễn mối quan hệ giữa hai lớp $C(E_1)$ và $C(E_2)$: E_1RE_2 và E_2RE_1 .

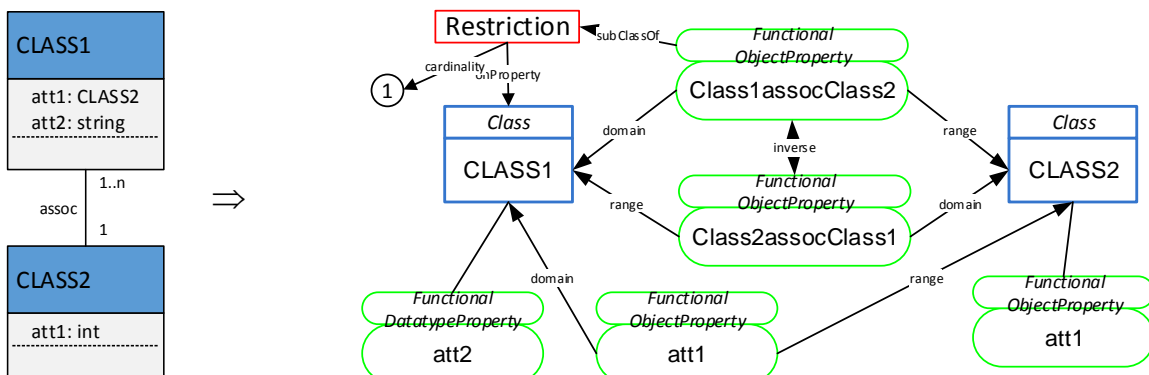
- Ràng buộc bản số trên lớp đối tượng UML sẽ được hoán đổi vị trí giữa hai lớp khi chuyển đổi sang OWL. Với mỗi giá trị bản số (min , max), nếu min khác 0 hoặc max khác N trên mỗi quan hệ R thì bổ sung ràng buộc số lượng cực tiểu/cực đại vào thuộc tính đối tượng tương ứng.

Chứng minh:

Theo [14], liên quan đến việc chuyển đổi mỗi quan hệ kết hợp của biểu đồ lớp UML sang mô hình ER, ta chuyển quan hệ kết hợp R thành mối quan hệ nhị nguyên R giữa các thực thể E_1 và E_2 . Bản số của mỗi quan hệ kết hợp sẽ được hoán đổi vị trí khi chuyển sang mô hình ER. (1)

Theo [4], một mối quan hệ nhị nguyên R giữa hai tập thực thể E_1 và E_2 được chuyển đổi thành cặp đối tượng ngược nhau E_1RE_2 và E_2RE_1 . Tùy thuộc vào giá trị của bản số (min , max) mà bổ sung ràng buộc bản số thích hợp vào thuộc tính đối tượng tương ứng. (2)

Từ (1) và (2), suy ra điều phải chứng minh.



Hình 3. Ví dụ chuyển đổi quan hệ kết hợp và thuộc tính có kiểu dữ liệu là lớp

2. Quan hệ kết hợp có lớp kết hợp

Quy tắc UML6. Xét quan hệ kết hợp giữa hai lớp E_1 và E_2 có lớp kết hợp A , ta có quy tắc chuyển đổi như sau:

- Bổ sung lớp $C(A)$, các thuộc tính $attA$ của lớp kết hợp A được chuyển thành các thuộc tính dữ liệu của lớp $C(A)$;
- Bổ sung hai thuộc tính đối tượng ngược nhau E_1HasA và $AOFE_1$ biểu diễn mối quan hệ giữa lớp $C(A)$ và lớp $C(E_1)$; hai thuộc tính đối tượng ngược nhau E_2HasA và $AOFE_2$ biểu diễn mối quan hệ giữa lớp $C(A)$ và lớp $C(E_2)$. Bổ sung tính chất hàm và ràng buộc cực tiểu là 1 vào hai thuộc tính: $AOFE_1, AOFE_2$;
- Ràng buộc bản số trên lớp đối tượng UML sẽ được hoán đổi vị trí giữa hai lớp khi chuyển đổi sang OWL. Với mỗi giá trị bản số (min, max), nếu min khác 0 và max khác N trên quan hệ kết hợp, bổ sung ràng buộc số lượng cực tiểu/cực đại tương ứng vào thuộc tính đối tượng E_1HasA và E_2HasA tương ứng. Nếu quan hệ kết hợp là mối quan hệ n-n thì bổ sung hai thuộc tính đối tượng $AOFE_1, AOFE_2$ vào tập thuộc tính khóa của lớp $C(A)$.

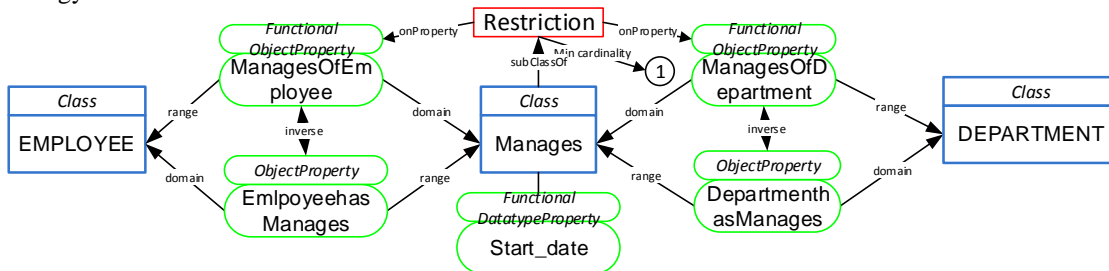
Chứng minh:

Theo [14], liên quan đến việc chuyển đổi mỗi quan hệ kết hợp có lớp kết hợp của biểu đồ lớp UML sang mô hình ER, ta chuyển quan hệ kết hợp có lớp kết hợp A thành mỗi quan hệ A giữa hai tập thực thể E_1 và E_2 , có các thuộc tính $attA$. Bản số của mỗi quan hệ kết hợp sẽ được hoán đổi vị trí khi chuyển sang mô hình ER. (3)

Theo [4], do mỗi quan hệ có thể xem như là tập thực thể, vì vậy ta xem mỗi quan hệ có thuộc tính là tập thực thể với bản số của mỗi quan hệ là (1, 1) ở phía tập thực thể mỗi quan hệ. Vì thế một mối quan hệ nhị nguyên A có thuộc tính giữa hai tập thực thể E_1 và E_2 được chuyển đổi thành lớp $C(A)$ và các thuộc tính thành phần $attA$. Đồng thời bổ sung hai cặp đối tượng ngược nhau E_1HasA và $AOFE_1$ biểu diễn mối quan hệ giữa lớp $C(A)$ và lớp $C(E_1)$; E_2HasA và $AOFE_2$ biểu diễn mối quan hệ giữa lớp $C(A)$ và lớp $C(E_2)$. Tùy thuộc vào giá trị của bản số (min, max) mà bổ sung ràng buộc bản số thích hợp vào thuộc tính đối tượng tương ứng. Đồng thời bổ sung tính chất hàm và ràng buộc cực tiểu là 1 vào hai thuộc tính: $AOFE_1, AOFE_2$; (4)

Từ (3) và (4), suy ra điều phải chứng minh.

Như ở Hình 1, lớp *Employee* quan hệ kết hợp với lớp *Department* có lớp kết hợp *Manages* được chuyển đổi sang OWL ontology như Hình 4.



Hình 4. Ví dụ chuyển đổi quan hệ kết hợp có lớp kết hợp

3. Quan hệ kết hợp đệ quy

Có thể liên kết một lớp với bản thân nó trong một quan hệ kết hợp. Quan hệ kết hợp ở đây vẫn thể hiện một sự liên quan ngữ nghĩa, nhưng các đối tượng được nối kết đều thuộc chung một lớp. Lúc này, một tập hợp các thể hiện có thể đảm nhận một vai trò duy nhất hoặc hai vai trò khác nhau trong cùng mối quan hệ. Việc kiểm tra các vai trò cho phép chúng ta phân loại tất cả các quan hệ kết hợp đệ quy đó thành là quan hệ kết hợp đệ quy đối xứng hoặc bất đối xứng.

Một mối quan hệ đệ quy (hay quan hệ phản xạ) được gọi là có tính chất đối xứng khi tất cả các thể hiện tham gia vào mỗi quan hệ có một vai trò duy nhất và có cùng ngữ nghĩa. Theo đó, nếu R là mối quan hệ đệ quy có tính chất đối xứng thì $role_1 \equiv role_2$, với $role_1$ và $role_2$ là tên của hai vai trò của mối quan hệ phản xạ R . Ngược lại, nếu R không có tính chất đối xứng, thì ta gọi mối quan hệ đệ quy là bất đối xứng.

Quy tắc UML7. Xét mỗi quan hệ kết hợp đệ quy đối xứng của lớp đối tượng E với vai trò là $role$, ta có quy tắc chuyển đổi như sau:

- Bổ sung thuộc tính đối tượng có định danh là tên vai trò $role$ trong mỗi quan hệ, có miền và phạm vi là lớp $C(E)$, thiết lập tính chất đối xứng cho thuộc tính đối tượng trên;
- Nếu bản số quan hệ đệ quy là 1:1, thêm ràng buộc số lượng cực đại bằng 1;
- Với ràng buộc bản số nhỏ nhất lần lượt là '1-1' thì thêm ràng buộc số lượng cực tiểu bằng 1 lên thuộc tính đối tượng được thêm vào.

Chứng minh:

Theo [14], liên quan đến việc chuyển đổi mối quan hệ kết hợp của biểu đồ lớp UML sang mô hình ER, ta chuyển quan hệ kết hợp R thành mối quan hệ nhị nguyên R giữa thực thể E và chính nó. Bản số của mối quan hệ kết hợp sẽ được hoán đổi vị trí khi chuyển sang mô hình ER. (5)

Theo [4], một mối quan hệ nhị nguyên R giữa hai tập thực thể E_1 và E_2 được chuyển đổi thành cặp đối tượng ngược nhau E_1RE_2 và E_2RE_1 . Vì trong trường hợp này E_1 và E_2 trùng nhau, vì thế cặp đối tượng chỉ là một thuộc tính đối tượng ngược nhau. Tùy thuộc vào giá trị của bản số (min, max) mà bổ sung ràng buộc bản số thích hợp vào thuộc tính đối tượng tương ứng. (6)

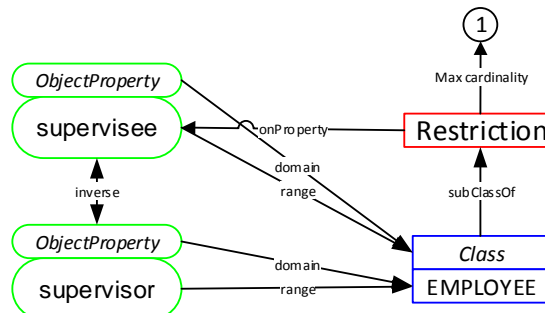
Từ (5) và (6) suy ra điều phải chứng minh.

Quy tắc UML8. Xét mối quan hệ đệ quy bất đối xứng của lớp đối tượng E với tên hai vai trò là $role_1$ và $role_2$, ta có quy tắc chuyển đổi như sau:

- Bổ sung cặp thuộc tính đối tượng ngược nhau có định danh lần lượt là tên của hai vai trò $role_1$ và $role_2$ trong mỗi quan hệ, có miền và phạm vi là lớp $C(E)$;
- Ràng buộc bản số trên lớp đối tượng UML sẽ được hoán đổi vị trí giữa hai vai trò khi chuyển đổi sang OWL. Với mỗi vai trò có ràng buộc bản số min/max khác 0 và khác N, bổ sung ràng buộc số lượng cực tiểu/cực đại tương ứng vào thuộc tính đối tượng ứng với vai trò đó.

Chứng minh quy tắc này tương tự như cách chứng minh quy tắc UML7.

Áp dụng quy tắc chuyển đổi mối quan hệ đệ quy bất đối xứng của lớp *Employee* như Hình 5:



Hình 5. Ví dụ chuyển đổi mối quan hệ kết hợp đệ quy bất đối xứng

4. Quan hệ kết tập

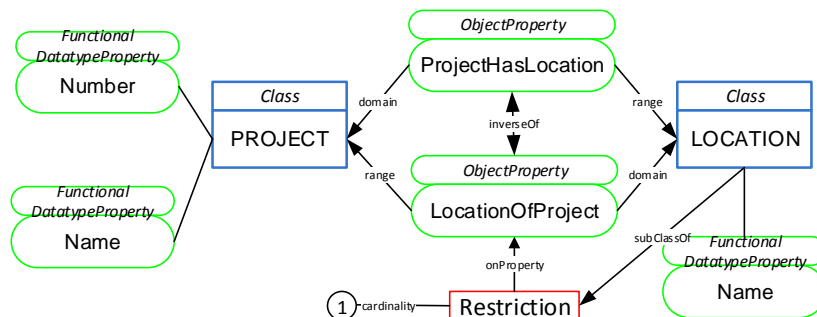
Quan hệ kết tập là một trường hợp đặc biệt của quan hệ kết hợp, tập trung thể hiện quan hệ giữa tổng thể và bộ phận. Trong UML, có ba loại quan hệ kết tập tùy thuộc vào ràng buộc của bộ phận trong tổng thể: một là kết tập thông thường, hai là bằng tham chiếu (kết tập chia sẻ), ba là theo giá trị (kết tập hợp thành). Kết tập hợp thành là tương đương với thành phần thuộc tính, vì vậy cách thức chuyển đổi tương tự như chuyển đổi thuộc tính.

Quan hệ kết tập chỉ ra rằng sự tồn tại của các bộ phận phụ thuộc vào tổng thể. Sự phụ thuộc này dẫn đến việc chuyển đổi quan hệ kết tập thành cặp thuộc tính đối tượng ngược nhau trong OWL với ràng buộc tương ứng.

Quy tắc UML9. Xét lớp tổng thể E quan hệ kết tập với lớp bộ phận F , ta có quy tắc chuyển đổi như sau:

- Tạo lớp $C(E)$ và $C(F)$;
- Các thuộc tính $attF$ của lớp bộ phận F được chuyển đổi thành các thuộc tính kiểu dữ liệu $attF$ và là thuộc tính con của lớp $C(F)$ có tính chất hàm, miền là lớp $C(F)$ và phạm vi là kiểu dữ liệu tương ứng trong OWL [11].

Như ở Hình 1, lớp *Project* quan hệ kết tập với lớp *Location* được chuyển đổi như Hình 6.



Hình 6. Ví dụ chuyển đổi quan hệ kết tập

Quy tắc UML10. Xét quan hệ kết tập có yếu tố hạn định Q giữa lớp tổng thể E với lớp bộ phận F , ta có quy tắc chuyển đổi như sau:

- Tạo hai lớp $C(E)$ và $C(F)$;
- Bổ sung thêm hai thuộc tính đối tượng ngược nhau thể hiện quan hệ giữa lớp $C(E)$ và lớp $C(F)$: $EhasF$ có miền là lớp $C(E)$, phạm vi là lớp $C(F)$; FOF có miền là lớp $C(F)$, phạm vi là lớp $C(E)$;
- Yếu tố hạn định Q được chuyển đổi thành thuộc tính kiểu dữ liệu Q trong lớp $C(F)$, có miền là lớp $C(F)$ và phạm vi là kiểu dữ liệu tương ứng trong OWL, thiết lập là thuộc tính khóa của lớp $C(F)$;
- Ràng buộc bản số trên lớp đối tượng UML sẽ được hoán đổi vị trí giữa hai vai trò khi chuyển đổi sang OWL;
- Khóa của lớp $C(F)$ được tạo bằng cách kết hợp khóa Q của lớp $C(F)$ với khóa KE của lớp $C(E)$.

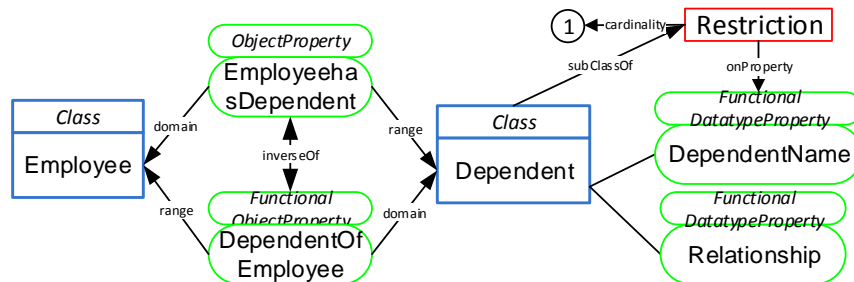
Chứng minh:

Theo [14], liên quan đến việc chuyển đổi mối quan hệ kết tập của biểu đồ lớp UML sang mô hình ER, ta chuyển quan hệ kết tập Q thành mối quan hệ nhị nguyên Q giữa thực thể E và tập thực thể F . Yếu tố hạn định Q được chuyển thành thuộc tính Q thuộc thực thể F . Bản số của mối quan hệ kết hợp sẽ được hoán đổi vị trí khi chuyển sang mô hình ER. (7)

Theo [4], một mối quan hệ nhị nguyên R giữa hai tập thực thể E và F được chuyển đổi thành lớp cặp đối tượng ngược nhau EQF và FQE . Tùy thuộc vào giá trị của bản số (min, max) mà bổ sung ràng buộc bản số thích hợp vào thuộc tính đối tượng tương ứng. (8)

Từ (7) và (8) suy ra điều phải chứng minh.

Như ở Hình 1, lớp $Employee$ quan hệ kết tập với lớp $Employee$ có yếu tố hạn định $Dependent_name$ được chuyển đổi thành hai lớp $Employee$ và $Employee$, yếu tố hạn định $Dependent_name$ được chuyển thành thuộc tính kiểu dữ liệu của lớp $Dependent$ như Hình 7.



Hình 7. Ví dụ chuyển đổi quan hệ kết tập có yếu tố hạn định

5. 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 lớp, 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 UML 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].

Trong UML có hai loại ràng buộc cho tổng quát hóa: hoàn toàn và phân ly (completeness và disjointness). Trong biểu đồ lớp, có thể xác định những ràng buộc này bằng cách thêm vào từ khóa *disjoint* hoặc *complete* gần đầu mũi tên tổng quát hóa. Tổng quát hóa mà không có từ khóa trên là không thuộc hai loại ràng buộc trên.

Quy tắc UML11. Xét quan hệ tổng quát hóa giữa lớp đối tượng E và các lớp đối tượng con E_i

- Tạo lớp $C(E)$ và các lớp $C(E_i)$ là lớp con của lớp $C(E)$;
- Nếu quan hệ tổng quát hóa là phân ly, thiết lập phân tách giữa các lớp $C(E_i)$ sử dụng cấu trúc *owl:disjointWith*; Nếu quan hệ tổng quát hóa là phân ly và hoàn toàn, thiết lập phân tách giữa các lớp $C(E)$ và $C(E_i)$ sử dụng cấu trúc *owl:disjointUnion* [11] [13].

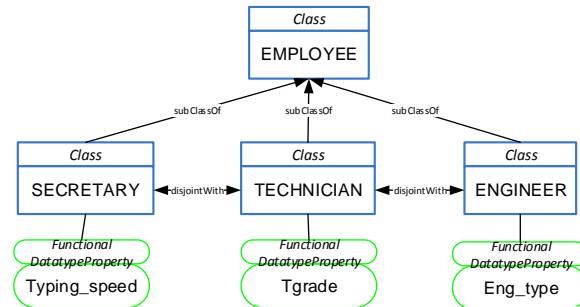
Chứng minh:

Theo [14], liên quan đến việc chuyển đổi mối quan hệ tổng quát hóa/chuyên biệt hóa của biểu đồ lớp UML sang mô hình ER, ta chuyển quan hệ kế thừa thành mối quan hệ Is-a giữa thực thể chủ E và tập thực thể con F . Bản số của mối quan hệ kết hợp sẽ được hoán đổi vị trí khi chuyển sang mô hình ER. (9)

Theo [4], tùy thuộc tập thực thể con là có ràng buộc hoàn toàn hay phân ly mà được chuyển đổi bằng cấu trúc *owl:disjointWith* hoặc *owl:disjointUnion*. (10)

Từ (9) và (10) suy ra điều phải chứng minh.

Với quan hệ tổng quát hóa của lớp *Employee* và các lớp đối tượng con *Secretary*, *Technical* và *Engineer* được chuyển đổi như Hình 8.



Hình 8. Ví dụ chuyển đổi quan hệ tổng quát hóa/chuyên biệt hóa

IV. KẾT LUẬN

Trên cơ sở kế thừa việc chuyển đổi từ các mô hình cơ sở dữ liệu mức khái niệm sang ontology, bài báo này đã bổ sung một số quy tắc chuyển đổi từ biểu đồ lớp UML sang OWL2, cụ thể đã đề xuất quy tắc chuyển đổi thuộc tính có kiểu dữ liệu là lớp, thuộc tính có cấu trúc, quan hệ đệ quy, quan hệ kết tập có yếu tố hạn định, là các vấn đề mà các nghiên cứu trước đây chưa đề cập đến. OWL ontology kết quả được kiểm chứng trên ứng dụng Protégé và bộ suy luận HermiT 1.3.8.3.

Tuy nhiên, do một số đặc điểm của biểu đồ lớp UML, nên vấn đề chuyển đổi sang OWL2 đảm bảo tính chất “bảo toàn ngữ nghĩa” vẫn hàm chứa nhiều thách thức. Trong thời gian tới, chúng tôi sẽ tiếp tục nghiên cứu thêm để hoàn chỉnh bộ quy tắc chuyển đổi. Đồng thời chúng tôi dự kiến nghiên cứu chuyển đổi ngược từ OWL 2 thành biểu đồ lớp UML để xác định một cách rõ ràng hơn sự đối sánh giữa biểu đồ lớp UML và OWL ontology.

TÀI LIỆU THAM KHẢO

- [1] Sujatha R Upadhyaya, P Sreenivasa Kumar, "ERONTO: A Tool for Extracting Ontologies from Extended E/R Diagrams," *ACM Symposium on Applied Computing*, pp. 666-670, 2005.
- [2] M. Fahad, "ER2OWL: Generating OWL Ontology from ER Diagram," in *IFIP – The International Federation for Information Processing*, 2008.
- [3] Igor Myroshnichenko, M.S., Marguerite C. Murphy, Ph.D., "Mapping ER Schemas to OWL Ontologies," *Semantic Computing. ICSC '09. IEEE International Conference*, pp. 324-329, 2009.
- [4] Toan Van Nguyen, Hoang Lien Minh Vo, Quang Hoang, Hanh Huu Hoang, "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, 2016.
- [5] Dragan Djurić, Dragan Gašević, Vladan Devedžić, "Ontology Modeling and MDA," *JOURNAL OF OBJECT TECHNOLOGY*, vol. 4, pp. 109-128, 2005.
- [6] Dragan Gašević, Dragan Djurić, Vladan Devedžić, Violeta Damjanović, "Converting UML to OWL Ontologies," in *ACM: Proceedings of the 13th international World Wide Web conference*, 2004.
- [7] S Brockmans; R Colomb; P Haase; E Kendall; Evan K. Wallace; G Xie, "A Model-Driven Approach for Building OWL DL and OWL Full Ontologies," in *Proceedings of the International Semantic Web Conference (ISWC) 2006*, 2006.
- [8] Noredine Gherabi, Mohamed Bahaj, "A New Method for Mapping UML Class into OWL Ontology," *Special Issue of International Journal of Computer Applications on Software Engineering, Databases and Expert Systems*, 2012.
- [9] Imants Zarembo, Sergejs Kodors, "Automatic Transformation of UML Geospatial Profile to OWL Ontologies," in *Virtual Multidisciplinary Conference QUAESTI*, 2013.
- [10] J Zedlitz, N Luttenberger, "Transforming Between UML Conceptual Models," in *Terra Cognita 2012 Workshop*, 2012.
- [11] Jesper Zedlitz, Jan Jörke, Norbert Luttenberger, "From UML to OWL 2," *Communications in Computer and Information Science*, vol. 295, 2012.
- [12] Jesper Zedlitz, Norbert Luttenberger, "Conceptual Modelling in UML and OWL-2," *International Journal on Advances in Software*, vol. 7, 2014.

- [13] Oussama El Hajjamy, Khadija Alaoui, Larbi Alaoui, Mohamed Bahaj, "MAPPING UML TO OWL2 ONTOLOGY," *Journal of Theoretical and Applied Information Technology*, vol. 90, no. 1, pp. 126-143, 2016.
- [14] Y. Ou, "On Mapping Between UML and Entity-Relationship Model," in *The Unified Modeling Language*, Mannheim, Germany, 1998.
- [15] Kilian Kiko, Colin Atkinson, "A Detailed Comparison of UML and OWL," Technical report, 2008.
- [16] M. E. K. Wallace, "Ontology Definition Metamodel," OMG, November 2007. [Online]. Available: <http://www.omg.org/cgi-bin/doc?ptc/07-09-09>. [Accessed 10 May 2017].
- [17] Elmasri R., Navathe S.B., *Fundamentals of Database Systems*, Addison-Wesley, 7th edn, 2015.

TRANSFORMATION OF UML TO OWL ONTOLOGY

Vo Hoang Lien Minh, Hoang Quang, Hoang Huu Hanh

ABSTRACT: *To knowledge representation, OWL is preferred to use, it is also seen as a language for concept modeling the complex information systems. Many legacy systems were built based on UML class diagrams, so the upgrading and transforming the UML class diagrams to ontology for reusing cost is really necessary. There are several studies aim at transforming from conceptual database models to ontology. On the basis of inheriting methods of transforming the ER model into an OWL ontology, this paper propose a method to transform the UML class diagram to OWL ontology.*