Journal of Computer Science and Cybernetics, V.30, N.3 (2014), 239–252
DOI:10.15625/1813-9663/30/3/2953

TÍCH HỢP ONTOLOGY VỚI TIẾP CẬN LÝ THUYẾT ĐỒNG THUẬN

NGUYỄN VĂN TRUNG[1], PHAN BÁ TRÍ[2], HOÀNG HỮU HẠNH[3]

[1] *Trường Đại học Khoa học, Đại học Huế*
nvtrung@hueuni.edu.vn
[2] *Trường Đại học Phú Xuân, Huế*
trip182@gmail.com
[3] *Đại học Huế;*
hhhanh@hueuni.edu.vn

Tóm tắt. Việc sử dụng lại các ontology tham chiếu khi xây dựng các cơ sở tri thức mới không làm giảm hoàn toàn khả năng có xung đột giữa các cơ sở tri thức. Trong quá trình tích hợp ontology ở mức khái niệm, bên cạnh việc xác định tập thuộc tính cho khái niệm, chúng ta cần phải xác định miền cho thuộc tính từ các đặc tả thuộc tính ở các ontology thành phần. Bài báo này trình bày một thuật toán tích hợp các ontology có xung đột ở cấp độ khái niệm dựa trên lý thuyết đồng thuận và hàm đánh giá khoảng cách ngữ nghĩa của các khái niệm trên cây phân cấp. Bài báo chứng tỏ, lý thuyết đồng thuận là một công cụ hữu ích trong việc xây dựng tri thức tổng hợp từ nhiều nguồn khác nhau.

Từ khóa. Ontology, tích hợp, lý thuyết đồng thuận, khoảng cách ngữ nghĩa.

Abstract. Ontology reuse has been an important factor in developing shared knowledge in Semantic Web. However, this cannot completely reduce conflict potentials in knowledge bases. In the ontology integration process on the concept level, we need to determine domain and range from properties of integrating ontologies. This paper presents an algorithm for ontology integration on concept level based on the consensus theory and an evaluation function of similarity measure between concepts in its hierarchical structure. This paper also proves that the consensus theory is a useful tool for building collective knowledge from different sources.

Keywords. Ontology, integration, consensus theory, semantic distance.

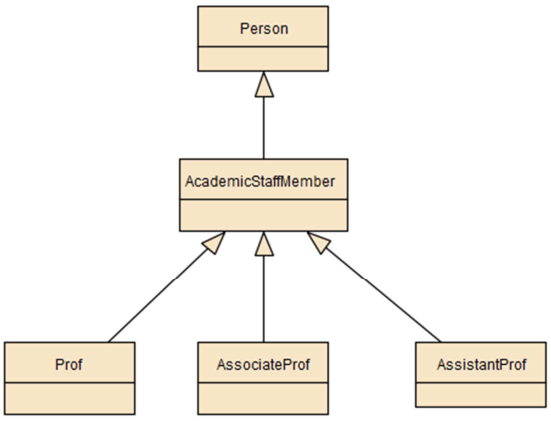
1. GIỚI THIỆU

Sự phát triển không ngừng của công nghệ thông tin và truyền thông dẫn đến một mặt trái: có quá nhiều dữ liệu, thông tin được sinh ra. Như một tất yếu, vấn đề quản lý sự không đồng nhất, không nhất quán giữa các nguồn thông tin trở nên cực kỳ quan trọng. Ontology cung cấp các bộ từ vựng để mô tả một cách hình thức tri thức về lĩnh vực nào đó [9]. Việc sử dụng ontology để biểu diễn các cơ sở tri thức làm giảm thiểu đáng kể sự không đồng nhất và xung đột giữa các cơ sở tri thức, đồng thời cho phép các cơ sở tri thức có thể tham chiếu lẫn nhau. Người ta có thể xây dựng các ontology của mình bằng cách tham chiếu đến các bộ từ vựng sẵn có như FOAF (www.foaf-project.org), Dublin Core (dublincore.org), ...

Tuy nhiên, việc tái sử dụng các ontology sẵn có trong quá trình xây dựng ontology mới không làm giảm hoàn toàn nguy cơ tạo ra các cơ sở tri thức xung đột, bởi các nhà xây dựng

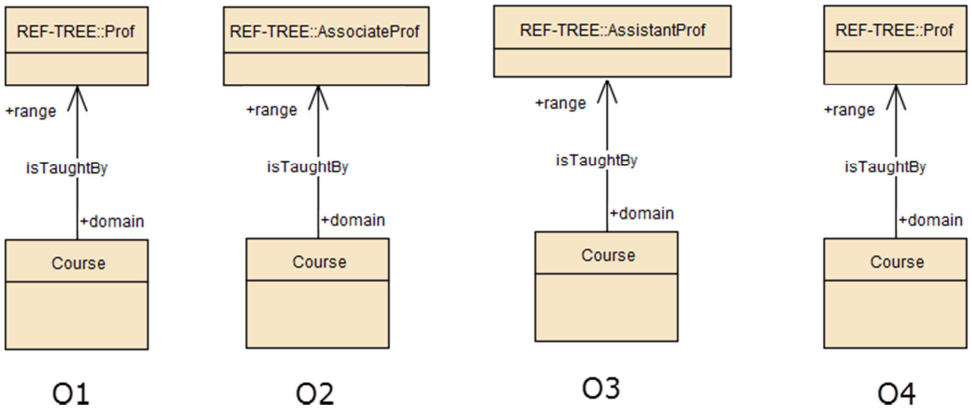
© 2014 Vietnam Academy of Science & Technology

ontology khác nhau có những cách nghĩ khác nhau để sử dụng ontology tham chiếu. Chẳng hạn, một ví dụ đơn giản, 4 người khác nhau cùng tham chiếu đến cây phân cấp khái niệm O_{REF_TREE} (Hình 1) để đặc tả thuộc tính *isTaughtBy* của khái niệm course theo những cách có thể là khác nhau (Hình 2). Câu hỏi đặt ra là: từ các đặc tả thuộc tính $isTaughtBy$ như



Hình 1: Cây phân cấp khái niệm O_{REF_TRE}

thế, chúng ta phải kết luận đặc tả thuộc tính tổng hợp phải là như thế nào để phù hợp với các đặc tả thành phần đã cho?



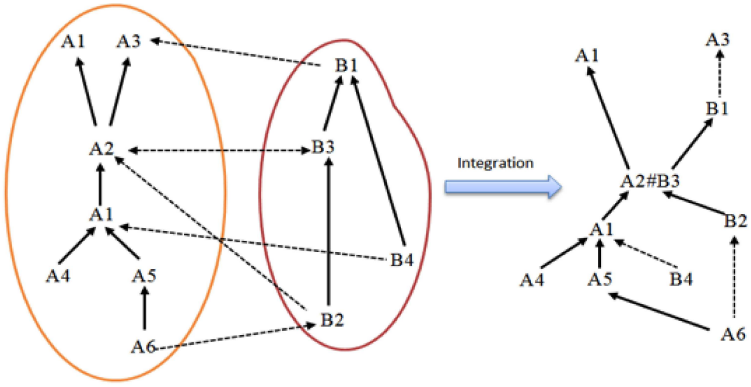
Hình 2: Trích dẫn cấu trúc của khái niệm Course trong các ontology

Bài báo này sẽ trình bày một phương pháp tích hợp ontology thuộc trường hợp như vậy dựa trên cách tiếp cận của lý thuyết đồng thuận [2]. Các phần tiếp theo của bài báo được trình bày theo trình tự như sau: phần 2 mô tả bài toán tích hợp ontology, các cấp độ xung đột ontology cùng với một số cách tiếp cận để giải quyết bài toán này; phần 3 trình bày một số khái niệm cơ sở của lý thuyết đồng thuận; phần 4, sau khi phát biểu bài toán tích hợp ontology ở cấp độ khái niệm dưới dạng phù hợp với mô hình có thể áp dụng được lý thuyết đồng thuận, chúng tôi sẽ trình bày cách thức xây dựng không gian khoảng cách dựa trên cây

phân cấp khái niệm và hàm đánh giá tương đồng ngữ nghĩa, và – đóng góp chính của bài báo – thuật toán tích hợp các ontology; phần 5 trình bày kết luận và một số hướng mở rộng cho bài báo.

2. TÍCH HỢP ONTOLOGY

Tích hợp là tiến trình xây dựng một ontology từ việc kết hợp hai hay nhiều ontology khác nhau, các ontology được kết hợp không nhất thiết cùng miền tri thức. Trong quá trình tích hợp, các ontology ban đầu được tổng hợp, liên kết, lắp ghép với nhau để tạo thành ontology kết quả, có khả năng tái sử dụng sau khi chịu một số thay đổi chẳng hạn như mở rộng ontology kết quả, hoặc gia tăng miền tri thức, hoặc ontology kết quả có khả năng tương thích tốt hơn.



Hình 3: Tích hợp hai ontology

Vấn đề tích hợp ontology được giải quyết với nhiều kỹ thuật khác nhau [5]:

- So khớp ontology (ontology matching): tìm kiếm các mối quan hệ hoặc các mối tương ứng giữa các thực thể của các ontology khác nhau. Các thực thể trong một ontology bao gồm lớp (class), cá thể (individual), quan hệ (relation), kiểu dữ liệu (data type), giá trị dữ liệu (data value). Kết quả của quá trình so khớp là các ánh xạ ontology (ontology alignment).

- Trộn ontology (ontology merging): tạo ra một ontology mới từ hai hoặc nhiều ontology nguồn. Các ontology này có thể chồng nhau.

Một định nghĩa cho quá trình tích hợp ontology được mô tả trong [13] là: Cho trước tập các ontology $\{O_1, O_2, \ldots, O_n\}$, cần xác định ontology O^* tốt nhất, có khả năng đại diện các ontology đã cho.

Điểm mấu chốt của bài toán tích hợp ontology đó là phải giải quyết sự xung đột giữa các thực thể trong các ontology nguồn. Người ta phân làm 3 cấp độ xung đột giữa các thực thể ontology như sau [5, trang 247]:

- Xung đột ở cấp độ thể hiện: một thể hiện được mô tả theo những cách khác nhau trong các ontology khác nhau.

- Xung đột ở cấp độ khái niệm: một lớp, hay khái niệm, có cùng tên nhưng lại có cấu trúc khác nhau trong các ontology khác nhau.

- Xung đột ở cấp độ quan hệ: các ontology khác nhau chứa các mối quan hệ khác nhau giữa cùng hai khái niệm.

Trong hơn 10 năm trở lại đây, bài toán giải quyết xung đột giữa các thực thể của ontology đã được cộng đồng khoa học quan tâm nghiên cứu, trong đó, việc xử lý xung đột ở cấp độ khái niệm thường được nghĩ đến trước tiên bởi khi xây dựng một ontology, người ta thường xây dựng cây phân cấp khái niệm trước. Bài báo này chỉ đề cập đến vấn đề giải quyết xung đột giữa các ontology ở cấp độ khái niệm. Phần dưới đây sẽ điểm qua các nhóm giải pháp xử lý xung đột ontology cho bài toán tích hợp tri thức.

Nhóm giải pháp thứ nhất, chẳng hạn như MOMIS [3] (Fergnani, 2001), MLMA+ [1] (Alasoud, 2010) đánh giá độ tương tự của các thực thể dựa vào độ tương tự của các cặp tên thực thể cũng như các thành phần bổ trợ (như các mô tả, ghi chú của thực tể bằng ngôn ngữ tự nhiên). Nhóm phương pháp này thường sử dụng các tài nguyên từ vựng tham chiếu như WordNet với các quan hệ từ đồng nghĩa, trái nghĩa để hỗ trợ trong quá trình xử lý.

Nhóm giải pháp thứ hai gồm ONION [11] (Mitra và cộng sự, 2002), S-MATCH [8] (Giunchiglia và Shvaiko, 2003), OLA [6] (Euzenat và Valtchev, 2004), H-Match [4] (Castano và cộng sự, 2003) dựa vào việc so sánh cấu trúc các đồ thị thể hiện mối quan hệ của các thực thể để đánh giá độ tương đồng của các thực thể.

Một số tác giả khác như Li và các cộng sự [10] (2007), Umer và Mundy [14] (2012), đưa ra các giải pháp lai, sử dụng kết hợp các chiến lược như dựa vào khoảng cách chỉnh sửa (edit distance), phương pháp học thống kê (statiscal learning), ... để tạo ra kết quả cuối cùng.

Theo quan điểm của chúng tôi, các cách tiếp cận trên có một số nhược điểm. Việc căn cứ vào phép so sánh chuỗi trên các tên thực thể, hoặc thậm chí chi tiết hơn, so sánh chuỗi trên các tập thuật ngữ được trích rút từ các ghi chú kèm theo mô tả thực thể (thông qua các kỹ thuật xử lý ngôn ngữ tự nhiên) là chưa đủ để đánh giá toàn diện mức độ tương đồng của hai thực thể. Lý do là có thể có nhiều cặp từ đồng âm – khác nghĩa, hoặc đồng nghĩa – khác âm, hoặc phụ thuộc vào quan điểm độc lập của người xây dựng cơ sở tri thức. Mâu thuẫn trong đặc tả mối quan hệ $isTaughtBy$ ở phần đầu của bài báo này là một ví dụ. So khớp theo tên thực thể chỉ nên đóng vai trò tiền xử lý cho các bước tiếp theo của bài toán tích hợp tri thức. Căn cứ vào cấu trúc của đồ thị có thể cho kết quả chính xác hơn, nhưng cũng đồng nghĩa với việc làm gia tăng độ phức tạp của bài toán, đặc biệt là đối với số lượng lớn các ontology cũng như số lượng lớn các thực thể trong mỗi ontology thành phần. Một khó khăn nữa, sau khi xác định được các độ tương đồng giữa các thực thể (với một mức độ chính xác nào đó), cần phải có chiến lược cụ thể để đưa ra thực thể tổng hợp cuối cùng. Khó khăn này khiến hầu hết các giải pháp hiện nay chỉ đưa ra được lời giải cho một số ứng dụng cụ thể.

3. TÍCH HỢP ONTOLOGY MỨC KHÁI NIỆM THEO LÝ THUYẾT ĐỒNG THUẬN

3.1. Lý thuyết đồng thuận

Lý thuyết đồng thuận (consensus theory) [2] là một công cụ thích hợp để xây dựng trí tuệ tổng hợp (collective intelligence). Một số kết quả và hướng áp dụng của lý thuyết đồng thuận

cho bài toán xử lý tri thức được trình bày trong [13]. Trong phần này của bài báo, chúng tôi giới thiệu một số khái niệm cơ bản của lý thuyết đồng thuận được sử dụng cho bài toán tích hợp ontology.

Gọi U là tập hợp hữu hạn các đối tượng, biểu diễn các giá trị có thể có cho một trạng thái tri thức (knowledge state). Người ta ký hiệu:

- 2^U là tập hợp tất cả các tập hợp con lập được từ U.
- $\prod\limits_k(U)$ là tập hợp tất cả các bộ có lặp gồm k phần tử lập được từ U (k là một số tự nhiên).
- $\prod(U) = \cup_{k\in N}\prod\limits_k(U)$ được gọi là tập hợp tất cả các bộ có lặp khác rỗng lập được từ U. Mỗi phần tử thuộc $\prod(U)$ được gọi là một hồ sơ xung đột, hoặc gọi ngắn gọn là một hồ sơ.

Một hồ sơ xung đột có thể được xem là một tập hợp các ý kiến của các chuyên gia về một chủ đề nào đó. Các ý kiến của các chuyên gia có thể giống nhau hoặc không giống nhau. Ví dụ: Tập các ý kiến của chuyên gia về dự báo thời tiết theo các tiêu chí như mã vùng, ngày dự báo, nhiệt độ (°C), có mưa, có nắng như sau:

$$X = \left\{\left\{HU, 12.07.2013, 25°\text{C} \div 35°\text{C}, \text{có}, \text{có}\right\}, \left\{HU, 12.07.2013, 29°\text{C} \div 34°\text{C}, \text{có}, \text{không}\right\}\right\}$$

Từ các ý kiến của các chuyên gia, người ta cần xác định phương án lựa chọn phù hợp nhất có thể đại diện cho các phương án của các chuyên gia.

Khi xử lý các bộ có lặp, ta thường sử dụng các phép toán và ký hiệu thuộc đại số tập hợp có lặp như các ví dụ sau:

- $X = \{x, x, y, y, y, z\}$ là hồ sơ gồm 6 phần tử, trong đó có 2 phần tử có giá trị x, 3 phần tử có giá trị y, 1 phần tử có giá trị z. Ta viết $|X| = 6$.
- Người ta có thể viết tương đương $X = \{2 * x, 3 * y, z\}$.
- Hồ sơ X được gọi là bội của hồ sơ Y, ký hiệu $X = n * Y$ nếu $Y = \{x_1, x_2, \ldots, x_k\}$ và $X = \{n * x_1, n * x_2, \ldots, n * x_k\}$.
- Hồ sơ X được gọi là đồng nhất nếu mọi phần tử của nó đều giống nhau, tức là $X = \{n * x\}$ với $n \in N,\ x \in U$. Ngược lại n, ta nói X là không đồng nhất.
- Hồ sơ X được gọi là phân biệt được nếu các phần tử của nó là khác nhau từng đôi một.
- Hồ sơ X được gọi là chính quy nếu nó là không phân biệt được hoặc là bội của một hồ sơ không phân biệt được.
- Tổng ($\dot\cup$) của hai hồ sơ là một hồ sơ được thành lập theo quy tắc sau: Nếu x xuất hiện trong hồ sơ X và hồ sơ Y tương ứng n và n' lần thì trong hồ sơ tổng, x xuất hiện $n + n'$ lần.
- Hiệu (-) của hai hồ sơ là một hồ sơ được thành lập theo quy tắc sau: Nếu x xuất hiện trong hồ sơ X và hồ sơ Y tương ứng và n' lần thì trong hồ sơ hiệu, x xuất hiện $n - n'$ lần nếu $n \geqslant n'$, xuất hiện 0 lần nếu ngược lại.

- Hồ sơ X được gọi là con của hồ sơ Y, ký hiệu $X \subseteq Y$ nếu mỗi phần tử trong X có số lần xuất hiện không lớn hơn số lần xuất hiện trong hồ sơ Y.

3.1.1. Hàm khoảng cách và một số biểu thức trên hàm khoảng cách

Hàm khoảng cách $d : U \times U \to [0, 1]$ được định nghĩa để đảm bảo các tính chất sau:

- Tính không âm: $\forall x, y \in U : d(x, y) \geq 0$,
- Tính phản xạ: $\forall x, y \in U : d(x, y) = 0 \Leftrightarrow x = y$,
- Tính đối xứng: $\forall x, y \in U : d(x, y) = d(y, x)$.

Người ta gọi (U, d) là một không gian khoảng cách và định nghĩa một số biểu thức với hàm khoảng cách như sau:

Với $X \in \prod(U)$, $i = 1, 2$:

- $d^i(x, X) = \sum_{y \in X} d^i(x, y)$, với $x \in U$.
- $d^i_{t\_mean}(X) = \frac{1}{k(k+1)} \sum_{x,y \in X} d^i(x, y)$, với $k = |X|$.
- $d^i_{min}(X) = \min\{d^i(x, X) : x \in U\}$
- $d^i_{max}(X) = \max\{d^i(x, X) : x \in U\}$

Trong trường hợp $i = 1$, chỉ số i có thể được bỏ qua, chẳng hạn ta có thể viết $d(x, X)$ thay cho $d^1(x, X)$.

3.1.2. Hàm chọn đồng thuận và các tiêu chuẩn cho hàm chọn đồng thuận

Hàm chọn đồng thuận $C : \prod(U) \to 2^U$ được định nghĩa trong không gian khoảng cách (U, d) biểu diễn lựa chọn đồng thuận cho một hồ sơ xung đột.

Như vậy, với một hồ sơ xung đột $X \in \prod(U)$, $C(X)$ là một tập hợp (không lặp) chứa các phương án đồng thuận đồng thuận của hồ sơ xung đột X; mỗi phần tử của $C(X)$ được gọi là một phần tử đồng thuận của hồ sơ X.

Người ta ký hiệu $Con(U)$ là tập tất cả hàm chọn đồng thuận trong không gian khoảng cách (U, d). Một hàm chọn đồng thuận $C \in Con(U)$ được đánh giá qua các tính chất sau:

1) Tin cậy (Reliability), ký hiệu là Re, nếu và chỉ nếu $C(X) \neq \emptyset$.

2) Đồng nhất (Unanimity), ký hiệu là Un, nếu và chỉ nếu $C(\{n * x\}) = \{x\}$ với mỗi $n \in N$ và $x \in U$.

3) Đơn giản (Simplification), ký hiệu là Si, nếu và chỉ nếu

$$(X \text{ là bội của} Y) \Rightarrow (C(X) = C(Y)).$$

4) Gần-nhất quán (Quasi-unanimity), ký hiệu là Qu, nếu và chỉ nếu

$$(x \notin C(X)) \Rightarrow (\exists n \in N : x \in C(X \cup \{n^* x\})).$$

5) Nhất quán (Consistency), ký hiệu là Co, nếu và chỉ nếu

$$(x \in C(X)) \Rightarrow x \in C(X \dot{\cup} \{x\})$$

6) Nhất quán Condorcet (Condorcet consistency), ký hiệu là Cc, nếu và chỉ nếu

$$(C(X_1) \cap C(X_2) \neq \emptyset) \Rightarrow (C(X_1 \dot{\cup} X_2) = C(X_1) \cap C(X_2)) \text{ với mọi } X_1, X_2 \in \prod(U).$$

7) Nhất quán tổng quát (General consistency), ký hiệu là Gc nếu và chỉ nếu

$$C(X_1) \cap C(X_2) \subseteq C(X_1 \dot{\cup} X_2) \subseteq C(X_1) \cup C(X_2) \text{ với mọi } X_1, X_2 \in \prod(U)$$

8) Đồng biến (Proporiton), ký hiệu là Pr, nếu và chỉ nếu

$$(X_1 \subseteq X_2 \wedge x \in C(X_1) \wedge y \in C(X_2)) \Rightarrow (d(x, X_1) \leq d(y, X_2)) \text{ với mọi } X_1, X_2 \in \prod(U).$$

9) Tối ưu-1 (1-Optimality), ký hiệu O_1, nếu và chỉ nếu

$$x \in C(X) \Rightarrow d(x, X) = min_{y \in U} d(y, X) \text{ với mọi } X \in \prod(U).$$

10) Tối ưu-2 (2-Optimality), ký hiệu O_2, nếu và chỉ nếu

$$x \in C(X) \Rightarrow d^2(x, X) = min_{y \in U} d^2(y, X) \text{ với mọi } X \in \prod(U).$$

Tuỳ theo tính chất của từng bài toán lựa chọn đồng thuận, người ta sẽ xây dựng hàm chọn đồng thuận cụ thể nhằm thoả mãn các tiêu chuẩn trên. Hàm chọn đồng thuận càng thoả mãn nhiều tiêu chuẩn thì càng có giá trị. Trong [13] đã chứng minh rằng không có hàm chọn đồng thuận nào thoả mãn cả 10 tiêu chuẩn nói trên. Tuy nhiên [13] cũng đã chỉ ra một số phụ thuộc lẫn nhau của các tiêu chuẩn này. Những phụ thuộc quan trọng nhất là:

a) $(O_1 \wedge Re) \Leftrightarrow (Pr \wedge Qu \wedge Re \wedge Co \wedge Si)$

b) $(Pr \wedge Qu \wedge Re) \Rightarrow Un$.

c) $(O_2 \wedge Re) \Leftrightarrow (Co \wedge Qu \wedge Un \wedge Si)$.

Kết quả này được dùng làm cơ sở để xây dựng các hàm đồng thuận cho hai lớp bài toán chính sau đây. Giả sử cần đưa ra phương án hợp lý từ một bộ các giải pháp được cho bởi các thành viên (tức là cần chọn ra phương án đồng thuận từ một hồ sơ xung đột):

- Nếu phương án hợp lý phải phụ thuộc vào các phương án của các thành viên, theo nghĩa, phương án đồng thuận phải là đại diện tốt nhất cho các phương án đã được đề xuất trong hồ sơ, chúng ta phải dùng tiêu chuẩn O_1 để xây dựng hàm chọn đồng thuận.

- Nếu phương án hợp lý là độc lập với các phương án được đưa ra bởi các thành viên, theo nghĩa, phương án đồng thuận phải phản ánh tất cả các khía cạnh của các phương án đã được đề xuất trong hồ sơ (ở mức có thể thoả hiệp được), chúng ta phải dùng tiêu chuẩn O_2 để xây dựng hàm chọn đồng thuận.

3.1.3. Tính khả đồng thuận của hồ sơ xung đột

Hàm chọn đồng thuận thoả mãn tiêu chuẩn O_1 (tương ứng, O_2) được gọi là hàm-O_1 (tương ứng, hàm-O_2). Phương án đồng thuận được xác định bằng hàm-O1 (tương ứng, hàm-O_2) được gọi là đồng thuận- O_1 (tương ứng, đồng thuận O_2).

Không phải từ hồ sơ xung đột nào cũng chọn ra được phương án đồng thuận nói chung và đồng thuận O_1 hay O_2 nói riêng. Người ta đã chỉ ra tính khả đồng thuận đối với các hàm đồng thuận được xây dựng theo tiêu chuẩn O_1 và O_2 như sau: Trong không gian khoảng cách (U, d), hồ sơ $X \in \prod(U)$ là khả đồng thuận theo tiêu chuẩn O_i $(i = 1, 2)$ nếu và chỉ nếu $d^i_{t\_mean}(X) \geqslant d^i_{min}(X)$.

3.2. Tích hợp ontology mức khái niệm theo tiếp cận lý thuyết đồng thuận

Định nghĩa 3.1 (Ontology) Ontology là một bộ bốn $\langle C, I, R, Z\rangle$, trong đó:

- C là tập hợp các khái niệm (lớp).
- I là tập hợp các thể hiện (instance) của các lớp.
- R là tập hợp các quan hệ nhị phân định nghĩa trên C.
- Z là tập các tiên đề, là các công thức logic bậc nhất và có thể được diễn giải dưới dạng ràng buộc toàn vẹn hoặc các mối quan hệ giữa các thể hiện và các khái niệm, mà không thể được biểu diễn được bằng các quan hệ trong R.

Định nghĩa 3.2 (Thế giới thực) Gọi A là một tập hữu hạn các thuộc tính. Mỗi thuộc tính $a \in A$ có một miền V_a.
Với $V = \cup_{a\in A} V_a$, ta nói (A, V) mô tả một thế giới thực.
Một ontology tham chiếu đến thế giới thực (A, V) được gọi là ontology dựa trên (A, V).

Định nghĩa 3.3 (Cấu trúc khái niệm trong ontology) Một khái niệm của ontology dựa trên (A, V) được định nghĩa dưới dạng bộ ba (c, A^c, V^c), trong đó:

- c là tên của khái niệm,
- $A^c \subseteq A$ là tập thuộc tính mô tả khái niệm c,
- $V^c = \cup_{a\in A^c} V_a$ là miền của các thuộc tính $(V^c \subseteq V)$.

Cặp (A^c, V^c) được gọi là cấu trúc của khái niệm c.

Định nghĩa 3.4 (Quan hệ giữa các thuộc tính) Cặp thuộc tính a, b trong định nghĩa cấu trúc của một khái niệm có thể có quan hệ sau:

- tương đương: thuộc tính a được gọi là tương đương với thuộc tính b, viết là a $\leftrightarrow$ b, nếu a và b cùng phản ánh một đặc trưng cho các thể hiện của khái niệm. Nói cách khác, chúng là các tên khác nhau của một đặc trưng của khái niệm. Ví dụ: ngheNghiep $\leftrightarrow$ job.
- tổng quát hơn: thuộc tính a được gọi là tổng quát hơn thuộc tính b, viết là, a $\rightarrow$ b, khi thông tin được cho bởi thuộc tính a có chứa thông tin được cho bởi thuộc tính b. Ví dụ: dayOfBirth $\rightarrow$ age.

- trái ngược: thuộc tính a được gọi là trái ngược với thuộc tính b, viết là a $\downarrow$ b, nếu miền của chúng cùng là tập hợp 2 giá trị và giá trị mô tả của hai thuộc tính này cho cùng một thể hiện là trái ngược nhau. Ví dụ: isFree $\downarrow$ isLent, với $\mathrm{V_{isFree}} = \mathrm{V_{isLent}} = \{\text{true}, \text{false}\}$ giúp mô tả, chẳng hạn, các thực thể thuộc khái niệm sách là còn rảnh (isFree) hay đã được cho ai đó mượn rồi (isLent).

3.2.1. Phát biểu bài toán tích hợp ontology ở cấp độ khái niệm

Gọi $O_1, O_2, \ldots, O_n$ $(n \in N)$ là các ontology dựa trên (A, V). Khái niệm c được mô tả trong O_i là $\left(c, A^i, V^i\right)$, $i = 1, 2, \ldots, n$. Ta phát biểu bài toán tích hợp ontology mức khái niệm như sau:

Cho một bộ các cặp: $X = \left\{\left(A^i, V^i\right) : i = 1, 2, \ldots, n\right\}$ trong đó $\left(A^i, V^i\right)$ là cấu trúc của khái niệm c trong ontology O_i. Cần tìm bộ tích hợp (A^*, V^*) đại diện tốt nhất các cặp đã cho để mô tả cấu trúc của khái niệm c.

3.2.2. Các quy tắc để xác định bộ tích hợp tối ưu (A^*, V^*)

[13] đề xuất các tiêu chuẩn R1-R7 dưới đây để xây dựng thuật toán tìm bộ tích hợp tối ưu (A^*, V^*):

R1. Với mọi $a, b \in A = \cup_{i=1}^{n} A^i$, $a \leftrightarrow b$ thực hiện thay thế thuộc tính a trong mọi tập hợp A^i bởi thuộc tính b hoặc ngược lại.

R2. Nếu trong tập thuộc tính bất kỳ A^i xuất hiện đồng thời a và b mà $a \rightarrow b$ thì có thể loại bỏ thuộc tính b.

R3. Với mọi $a, b \in A = \cup_{i=1}^{n} A^i, a \downarrow b$, thực hiện thay thế thuộc tính a trong mọi tập hợp A^i bởi thuộc tính b hoặc ngược lại.

R4. Sự xuất hiện của một thuộc tính trong A^* phải chỉ phụ thuộc vào sự xuất hiện của thuộc tính này trong các tập hợp A^i.

R5. Một thuộc tính a xuất hiện trong A^* nếu nó xuất hiện trong quá nửa tổng số lần xuất hiện trong tập hợp các A^i.

R6. Tập A^* là bằng với tập A sau khi áp dụng các quy tắc R1-R3.

R7. Với mỗi thuộc tính $a \in A^*$, miền của nó là V_a (từ thế giới thực (A, V)).

Tuỳ theo tiêu chí chọn lựa tập thuộc tính của khái niệm tích hợp là "*càng nhiều thuộc tính càng tốt*" hay "*chỉ gồm những thuộc tính xuất hiện quá nửa*", chúng ta sẽ có các thuật toán tương ứng thoả các tiêu chuẩn {R1-R4, R6, R7}, {R1-R5, R7}.

Chúng tôi nhận thấy: trên thực tế, không phải lúc nào miền của thuộc tính a trong các ontology $O_1, O_2, \ldots, O_n$ cũng là giống nhau. Do đó, cần phải xác định một cách tường minh miền cho thuộc tính này. Tiêu chuẩn R7 ở trên có thể được điều chỉnh lại như sau:

Với một thuộc tính $a \in A^*$, miền V_a^* được xác định bằng cách tìm đồng thuận từ hồ sơ $X_a = \left\{V_a^1, V_a^2, \ldots, V_a^k\right\}$. Ở đây, X_a là hồ sơ xung đột thành lập từ các miền của thuộc tính a trong các ontology $O_1, O_2, \ldots, O_n$.

Phần còn lại của bài báo sẽ mô tả cách thức xây dựng không gian khoảng cách (U, d) và thuật toán để tìm cấu trúc tích hợp thoả các tiêu chuẩn {R1-R4, R6, R7}.

3.2.3. Hàm khoảng cách ngữ nghĩa giữa hai khái niệm trên cây phân cấp

Sử dụng ý tưởng từ [7] (Jike Ge và Yuhui Qiu, 2008), ta có thể tính khoảng cách ngữ nghĩa giữa hai khái niệm c_1, c_2 trên cây phân cấp. Ý tưởng bắt đầu từ việc gán trọng số cho các cạnh nối thể hiện quan hệ kế thừa trực tiếp trên cây phân cấp:

$$w\left(parent, children\right) = 1 + \frac{1}{2^{depth(child)}}$$

trong đó, $depth(child)$ biểu thị độ sâu từ khái niệm $child$ đến khái niệm gốc của cây phân cấp. Với hai khái niệm bất kỳ c_1, c_2 trên cây phân cấp, ta tính khoảng cách ngữ nghĩa giữa chúng theo thuật toán sau [7]:

Đầu vào: hai khái niệm c_1, c_2 thuộc cây phân cấp.

Đầu ra: giá trị khoảng cách ngữ nghĩa $Sem\_Dis\left(c_1, c_2\right)$.

Thủ tục:

if (c_1, c_2 là cùng một khái niệm)

$Sem\_Disc(c_1, c_2) := 0$ else if (tồn tại đường đi trực tiếp từ c_1 đến c_2 trên cây phân cấp)

$Sem\_Disc(c_1, c_2) := w(c_1, c_2)$;

else if (tồn tại đường đi gián tiếp từ c_1 đến c_2 trên cây phân cấp)

{

Xác định $shortestPath\left(c_1, c_2\right)$ là đường đi ngắn nhất từ c_1 đến c_2 trên cây phân cấp;

$$Sem\_Disc(c_1, c_2) := \sum\nolimits_{(c_i, c_j) \in shortestPath(c_1, c_2)} w(c_i, c_j);$$

}

else

{

Xác định *cpp* là khái niệm cha chung gần nhất của c_1, c_2 trên cây phân cấp;

$$Sem\_Disc(c_1, c_2) := \min\left\{Sem\_Disc(c_1, cpp)\right\} + \min\left\{Sem\_Disc(c_2, cpp)\right\};$$

}

Rõ ràng, hàm Sem_Disc là chưa được chuẩn hoá. Chúng ta có thể chuẩn hoá nó để định nghĩa một không gian khoảng cách (U, d) dựa trên cây phân cấp khái niệm như sau:

- U: tập các khái niệm của cây phân cấp khái niệm.
- d: $U \times U \to [0, 1]$

$$d(c_1, c_2) \mapsto 1 - \frac{1}{Sem\_Disc(c_1, c_2) + 1}$$

3.2.4. Thuật toán tích hợp ontology mức khái niệm dựa trên lý thuyết đồng thuận

Trên cơ sở lý thuyết đồng thuận, chúng tôi đề xuất thuật toán xác định cấu trúc tích hợp cho khái niệm c từ các ontology thành phần $O_1, O_2, \ldots, O_n$ như sau.

Đầu vào:

- Hồ sơ $X = \left\{\left(A^i, V^i\right), i = 1, \ldots, n\right\}$, với $\left(A^i, V^i\right)$ là cấu trúc mô tả khái niệm c trong ontology O_i.

- Cây phân cấp khái niệm REF-TREE dùng để tham chiếu. $C_{REF-TREE}$ là tập các khái niệm của cây phân cấp này.

- Không gian khoảng cách (U, d) được định nghĩa theo cây phân cấp khái niệm REF-TREE như mô tả ở phần 3.2.3.

Đầu ra: Cặp (A^*, V^*) đại diện tốt nhất lấy từ X để mô tả khái niệm c.

Thủ tục:

Bước 1: Đặt $A^* := \cup_{i=1}^{n} A^i$;

Bước 2: Với mỗi cặp thuộc tính $a, b \in A^*$

- Nếu $(a \leftrightarrow b)$ thì $A^* := A^* \backslash \{a\}$ với điều kiện a không xuất hiện trong các mối quan hệ với các thuộc tính khác của A^*;
- Nếu $(a \downarrow b)$ thì $A^* := A^* \setminus \{b\}$ với điều kiện không xuất hiện trong các mối quan hệ với các thuộc tính khác của A^*;
- Nếu $(a \rightarrow b)$ thì $A^* := A^* \setminus \{b\}$ với điều kiện b không xuất hiện trong các mối quan hệ với các thuộc tính khác của A^*;

Bước 3: Với mỗi thuộc tính $a \in A^*$

{

- Đặt $X_a = \left\{V_a^1, V_a^2, \ldots, V_a^k\right\}$ là hồ sơ chứa các miền của thuộc tính a được đặc tả trong các cặp và V_a^j là các khái niệm trên cây phân cấp REF-TREE $(i = 1, \ldots, n;\ j = 1, \ldots, k)$;
- Nếu X_a là khả đồng thuận theo tiêu chuẩn tối ưu O_1 thì

{
  - Xác định V_a^* là lựa chọn đồng thuận theo tiêu chuẩn tối ưu O_1 trên không gian khoảng cách (U, d);
  - Gán V_a^* là miền cho thuộc tính a trong tập A^*;

}

Ngược lại thì gán $A^* := A^* \setminus \{a\}$;

}

Bước 4: Với mỗi thuộc tính a từ A^*, bổ sung trở lại các thuộc tính b nếu có mối quan hệ $a \leftrightarrow b$ hoặc $a \downarrow b$;

Nhận xét:

- Độ phức tạp của thuật toán trên là $O\left(m^3\right)$ với $m = \left|\cup_{i=1}^{n} A^i\right|$ (m là số lượng thuộc tính khác nhau lấy từ các tập hợp A^i, $i = 1..n$).

- Thuật toán trên chỉ mô tả việc thực hiện tích hợp các thuộc tính có miền là các khái niệm thuộc cây phân cấp tham chiếu REF-TREE. Đối với các thuộc tính có miền không phải là khái niệm mà là các giá trị sơ cấp (số, chuỗi), hoặc các khoảng giá trị, theo [12] chúng ta vẫn có thể xác định miền tích hợp phù hợp cho thuộc tính bằng phương pháp đồng thuận.

- Thuật toán xác định cấu trúc đồng thuận cho khái niệm ở cả 2 thành phần: thuộc tính và miền của thuộc tính. Tập thuộc tính này thoả các tiêu chuẩn R1-R4, R6, R7 ở phần 3.2.2.

Dưới đây là ví dụ đơn giản minh hoạ cho thuật toán này.
Cho thế giới thực (A, V) được định nghĩa như sau:

- $A = \{cid, isTaughtBy, isFinish, isActive, sched, tkb\}$.
- $V_{cid} = [1, 1000]$.
- $V_{isTaughtBy} = \{AscProf, Prof, AssiProf, AcademicStaffMember\}$.
- $V_{isFinish} = \{Yes, No\}$.
- $V_{isActive} = \{Yes, No\}$.
- $V_{sched} = \{Mon, Tue, Wed, Thurs, Fri, Sat, Sun\}$.
- $V_{tkb} = \{Hai, Ba, Tu, Nam, Sau, Bay, CN\}$.

Mối quan hệ giữa các thuộc tính này là: $\{thoiKhoaBieu \leftrightarrow sched, isFinish \downarrow isActive\}$.
Các khái niệm của các ontology có tham chiếu đến cây phân cấp $O_{REF-TREE}$ ở Hình 1. Trước hết, ta xây dựng không gian khoảng cách (U, d) từ cây phân cấp khái niệm này:
Trọng số của các cạnh nối trên cây phân cấp:

- $w\,[Person,AcademicStaffMember] = 1 + \frac{1}{2} = 1.5$
- $w\,[AcademicStaffMember,AscProf] = 1 + \frac{1}{2^2} = 1.25$
- $w\,[AcademicStaffMember,Prof] = 1 + \frac{1}{2^2} = 1.25$
- $w\,[AcademicStaffMember,AssiProf] = 1 + \frac{1}{2^2} = 1.25$.

Bảng mô tả cấu trúc của khái niệm *course* từ 5 ontology:

Ontology	Cấu trúc của khái niệm course
O_1	$\{(cid, [1, 1000]), (isActive, V_{isActive}), (sched, V_{sched}), (isTaughtBy, AssiProf)\}$
O_2	$\{(cid, [1, 1000]), (isFinish, V_{isFinish})\}$
O_3	$\{(isActive, V_{isActive}), (tkb, V_{isFinish}), (cid, [1, 1000])\}$
O_4	$\{(cid, [1, 1000]), (isTaughtBy, Prof)\}$
O_5	$\{(cid, [1, 1000]), (isTaughtBy, AssiProf)\}$

Kết quả thực hiện thuật toán theo từng bước là như sau:

- Bước 1:
  Khởi gán bộ cấu trúc tích hợp cho khái niệm *course*:

$$A^* = \{cid, isActive, sched, isTaughtBy, isFinish, tkb\}$$

- Bước 2:
  Loại bỏ 2 thuộc tính $isFinish$ và tkb. Sau bước này, ta có:

$$A^* = \{cid, isActive, sched, isTaughtBy\}$$

- Bước 3:
  Xét thuộc tính cid: Miền của cid được xác định theo [12] là $V^*_{cid} = [1, 1000]$.
  Xét thuộc tính $isActive$: Miền của $isActive$ sẽ là $V^*_{isActive} = \{Yes, No\}$.
  Xét thuộc tính $sched$: Miền của $sched$ sẽ là

$$V_{sched} = \{Mon, Tue, Wed, Thurs, Fri, Sat, Sun\}.$$

  Xét thuộc tính $isTaughtBy$. Thuộc tính này có các miền được tham chiếu từ cây phân cấp OREF-TREE.
  Lập hồ sơ xung đột $X_{isTaughtBy} = \{2 * AssiProf, Prof\}$.

  - $d(Person, Prof) = \frac{11}{15} = 0.73$
  - $d(AcademicStaffMember, AscProf) =$
  - $d(AcademicStaffMember, Prof) =$
  - $d(AcademicStaffMember, AssiProf) = \frac{5}{9} = 0.56$
  - $d(Prof, AssiProf) = \frac{5}{7} = 0.71$
  - $d(Person, X) = \frac{11}{20} = 0.55$
  - $d(Prof, X) = \frac{5}{14} = 0.36$
  - $d(AssiProf, X) = \frac{5}{28} = 0.18$ $d(AssocProf, X) = \frac{5}{21} = 0.238$
  - $d_{t\_mean}(X_{isTaughtBy}) = \frac{5}{21} = 0.238$
  - $d_{\min(X_{isTaughtBy})} = \min\{d(Person, X), d(AcademicStaffmember, X)\}$,
  - $d(Prof, X), d(AssiProf, X), d(AscProf, X)\} = 0.18 = d(AssiProf, X)$.

  Như vậy ta có $d_{t\_mean}(X_{isTaughtBy}) \geqslant d_{min}(X_{isTaughtBy})$. Do đó hồ sơ X là khả đồng thuận theo tiêu chuẩn tối ưu O_1. Ta cũng xác định được, $V^*_{isTaughtBy} = AssiProf$.

- Bước 4:
  Bổ sung trở lại A^*các thuộc tính $isFinish$ và tkb. Kết quả cuối cùng, ta có cấu trúc tích hợp để mô tả khái niệm $course$ là như sau:
  $(shed, \{Mon, Tue, Wed, Thurs, Fri, Sat, Sun\}), (isTaughtBy, AssiProf)\}$.

4. KẾT LUẬN

Bài báo đã trình bày một cách sử dụng lý thuyết đồng thuận kết hợp với độ đo tương tự ngữ nghĩa giữa các khái niệm trên cây phân cấp khái niệm cho trước để tích hợp các ontology có xung đột ở cấp độ khái niệm.

Trong những công trình tiếp theo, chúng tôi sẽ phân tích khả năng áp dụng tiêu chuẩn tối ưu O_2 trong bước 3 của thuật toán cũng như áp dụng kết quả của bài báo cho việc tích hợp các ontology có xung đột ở các cấp độ khác.

TÀI LIỆU THAM KHẢO

[1] I. Akbari, and M. Fathian, "A novel algorithm for ontology matching", *Journal of Information Science*, v. 36, pp. 324-334, 2010

[2] J. P. Barthélemy, and M. F. Janowitz, "A formal theory of consensus", *SIAM Journal on Discrete Mathematics*, vol. 4, no. 3, pp. 305-322, 1991.

[3] D. Beneventano, S. Bergamaschi, and F. Guerra, *Semantic Annotation of Web Documents and Ontology evolution with the MOMIS System*, 2001.

[4] S. Castano, A. Ferrara, and S. Montanelli, "H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems", in *SWDB*, pp. 231-250, September, 2003.

[5] Jérôme Euzenat and Pavel Shvaiko, *Ontology matching, Second edition*, Heidelberg: Springer, 2013.

[6] Jérôme Euzenat *et al*, "Ontology alignment with OLA", in *Proc. 3rd ISWC2004 workshop on Evaluation of Ontology-based tools (EON)*, 2004.

[7] J. Ge, and Y. Qiu, "Concept similarity matching based on semantic distance", *In Semantics, Knowledge and Grid, 2008. SKG'08. IEEE*, pp. 380-383, December, 2008.

[8] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, *S-Match: an algorithm and an implementation of semantic matching*, Springer Berlin Heidelberg, pp. 61-75, 2004.

[9] T. Gruber, *What is an Ontology*, 1993.

[10] Y. Li, Q. Zhong, J. Li, and J. Tang, "Results of ontology alignment with RiMOM", in *Proceedings of International Workshop on Ontology Matching (OM)*, Busan, Korea, pp. 227–235, 2007.

[11] P. Mitra and G. Wiederhold, "Resolving terminological heterogeneity in ontologies", in *Proceedings of the ECAI workshop on Ontologies and Semantic Interoperability*, July, 2002.

[12] N. T. Nguyen, "Representation choice methods as the tool for solving uncertainty in distributed temporal database systems with indeterminate valid time", *In Engineering of Intelligent Systems*, Springer Berlin Heidelberg, pp. 445-454, 2002.

[13] N. T. Nguyen, *Advanced Information and Knowledge Processing*, Springer, pp. 1-362, 2008.

[14] Q. Umer, and D. Mundy, "Semantically intelligent semiautomated ontology integration", in *Proceedings of the World Congress on Engineering 2012* Vol II, WCE 2012, 4–6 July, London, UK, 2012.

Ngày nhận bài 18 - 4 - 2014
Nhận lại sau sửa 28 - 8- 2014