

**PHƯƠNG PHÁP PHÁT HIỆN DỊCH VỤ WEB NGŨ NGHĨA  
VÀ ỨNG DỤNG TRONG CỘNG TÁC DOANH NGHIỆP**

*Mai Văn Mươi  
Trường Đại học Khoa học, Đại học Huế  
Hoàng Hữu Hạnh  
Đại học Huế*

**TÓM TẮT**

*Tích hợp doanh nghiệp (Business to Business integration - B2Bi) là kết nối các chức năng kinh doanh của doanh nghiệp thường bị phân tán trên các hệ khác nhau nhằm thực thi một tiến trình nghiệp vụ nào đó. Các hệ thống chức năng này được mô tả dưới dạng các dịch vụ dựa trên các chuẩn biểu diễn ngữ nghĩa nên giúp cho quá trình phát hiện các chức năng liên quan đến cùng một lĩnh vực kinh doanh có sự chính xác hơn [3]. Một trong những vấn đề quan trọng của quá trình tích hợp các doanh nghiệp ở mức thực thi của quy trình nghiệp vụ cộng tác là xây dựng các phương pháp phát hiện các dịch vụ Web (Web Services-WS) cùng mô tả về lĩnh vực kinh doanh có liên quan với nhau giữa các doanh nghiệp. Như vậy, mục tiêu của bài báo là xây dựng phương pháp phát hiện WS từ danh sách các dịch vụ Web quảng cáo mà tương đồng với WS mẫu với một ngưỡng  $\alpha$  cho trước.*

**1. Mở đầu**

Hiện nay, có nhiều giải pháp khác nhau được các nhà nghiên cứu đề xuất cho vấn đề đối sánh giữa các khái niệm. Mỗi giải pháp đều có những phương thức tiếp cận riêng. Tuy nhiên, mục tiêu cần thiết của các phương pháp đó là tính hiệu quả khi đối sánh. Trong quá trình đối sánh các khái niệm Ontology thì không chỉ xem xét ở mối quan hệ giữa các khái niệm mà còn cần nhắc về khoảng cách liên quan giữa các khái niệm, các ràng buộc giữa các khái niệm với nhau. Trên cơ sở của việc xem xét một cách tổng quan về mối quan hệ phân cấp và khoảng cách ngữ nghĩa giữa các khái niệm, chúng tôi đề xuất phương pháp phát hiện kết hợp với đối sánh tương đương dựa trên khoảng cách ngữ nghĩa.

**2. Phương pháp tính khoảng cách giữa hai khái niệm**

**2.1. Phương pháp đánh trọng số**

Trong quy trình đánh giá sự tương đồng, phương pháp đánh giá trị trọng số cho các khái niệm kề nhau được đưa ra để tính trọng số giữa các khái niệm trong một cây phân cấp Ontology phục vụ cho việc tính toán khoảng cách giữa các khái niệm [4]. Cho

hai khái niệm  $C_1$  và  $C_2$ , công thức để tính trọng số giữa hai khái niệm này được tính theo công thức như sau:

$$w[C_1, C_2] = 1 + \frac{1}{k^{\text{depth}(C_2)}} \quad (1)$$

Trong đó:  $\text{depth}(C)$  biểu diễn độ sâu của khái niệm  $C$  từ nút khái niệm gốc đến nút  $C$  trong cây phân cấp Ontology,  $k$  là một hằng số lớn hơn 1 chỉ ra tỉ lệ mà giá trị hàm  $w$  giảm theo cây phân cấp Ontology [4].

Công thức (1) có thể hiện hai đặc điểm: thứ nhất, sự khác biệt về ngữ nghĩa của mức trên thì cao hơn mức dưới. Thứ hai, khoảng cách giữa hai khái niệm ngang cấp (anh em) thì lớn hơn khoảng cách giữa hai khái niệm cha con. Thêm vào đó, nếu như tồn tại quan hệ đa kế thừa giữa các khái niệm (có nghĩa là một khái niệm con  $C_0$  mà có nhiều khái niệm cha  $C_1, C_2$ ) thì giá trị hàm  $w$  của nút khái niệm con của nút khái niệm  $C_0$  đó có nhiều giá trị.

## 2.2. Thuật toán tính khoảng cách giữa hai khái niệm

Thuật toán:

**Đầu vào:** hai khái niệm  $C_0, C_1, C_2$

**Đầu ra:** Khoảng cách ngữ nghĩa giữa  $C_1$  và  $C_2$

*Semantic\_Distance*( $C_1, C_2$ )

{

*If*  $C_1, C_2$  là hai khái niệm giống nhau

*Semantic\_Distance*( $C_1, C_2$ )=0

*Else If* tồn tại đường đi trực tiếp giữa  $C_1$  và  $C_2$

*Semantic\_Distance*( $C_1, C_2$ )= $w(C_1, C_2)$

*Else If* tồn tại đường đi gián tiếp giữa  $C_1$  và  $C_2$

*Semantic\_Distance*( $C_1, C_2$ )=  $\sum_{C \in \text{SPath}(C_1, C_2)} w_C(C_1, C_2)$

*Else*

*Semantic\_Distance*( $C_1, C_2$ ) =  $\min\{\text{Semantic\_Distance}(C_1, C_0)\} + \min\{\text{Semantic\_Distance}(C_2, C_0)\}$

}

### 3. Một số khái niệm cơ sở

#### 3.1. Ngữ nghĩa chức năng

Ngữ nghĩa chức năng của một phép toán của WS là sự kết hợp của ngữ nghĩa dữ liệu của nó với phân loại các chức năng phép toán cũng như tiền điều kiện và hậu điều kiện của nó [7]. Cho  $s$  là một dịch vụ và  $o$  là một trong những phép toán của nó thì ngữ nghĩa chức năng của một phép toán của WS được định nghĩa như sau:

$$F(s, o) = \langle F_c(s,o), I(s,o), O(s,o), E(s,o), P(s,o), Po(s,o) \rangle \quad (2)$$

Trong đó:

- $F_c(s,o)$ : phân loại chức năng của phép toán ‘ $o$ ’ trong các thuật ngữ (terms) của các khái niệm ontology.
- $I(s,o)$ : các giá trị input của phép toán ‘ $o$ ’ trong các thuật ngữ của các khái niệm ontology.
- $O(s,o)$ : giá trị output của phép toán ‘ $o$ ’ trong các thuật ngữ của các khái niệm ontology.
- $E(s,o)$ : cho phép bắt các ngoại lệ trong quá trình thực thi phép toán ‘ $o$ ’ trong các thuật ngữ của các khái niệm ontology.
- $P(s,o)$ : tiền điều kiện (Pre-conditions) của phép toán ‘ $o$ ’ trong các thuật ngữ của các khái niệm ontology.
- $Po(s,o)$ : hậu điều kiện (Post-conditions) của phép toán ‘ $o$ ’ trong các thuật ngữ của các khái niệm ontology.

#### 3.2. Phân loại chất lượng dịch vụ (QoS)

Phân loại chất lượng dịch vụ của WS mô tả sự thực thi và chất lượng hoặc sự định lượng của WS. Để cung cấp cho dịch vụ hiểu được mỗi thuật ngữ chất lượng dịch vụ khác nhau thì quan trọng nhất là phải nắm bắt được, hiểu được ý nghĩa của thuật ngữ. Ontology có thể được sử dụng để biểu diễn và phát triển ngữ nghĩa của các tham số này [7]. Chất lượng dịch vụ thường được mô tả dựa trên time (thời gian), cost (chi phí), tính hiệu lực (availability) và độ tin cậy (reliability). Chúng ta có thể định nghĩa chất lượng dịch vụ của một phép toán  $o_j$  của WS  $s_i$  như sau:

$$QoS(s_i, o_j) = \langle T(s_i, o_j), C(s_i, o_j), R(s_i, o_j), A(s_i, o_j), DS_1(s_i, o_j), DS_2(s_i, o_j), \dots, DS_n(s_i, o_j) \rangle \quad (3)$$

Trong đó:

- $T(s_i, o_j)$ : thời gian thực thi WS  $s_i$  khi phép toán  $o_j$  được gọi.
- $C(s_i, o_j)$ : Chi phí của WS  $s_i$  khi phép toán  $o_j$  được gọi.
- $R(s_i, o_j)$ : độ tin cậy của WS  $s_i$  khi phép toán  $o_j$  được gọi.

- $A(s_i, o_j)$ : tính hiệu lực của WS  $s_i$  khi phép toán  $o_j$  được gọi.
- $DSk(s_i, o_j)$ : mức độ miễn ứng dụng của phép toán hoặc service.

Từ ngữ nghĩa chức năng và phân loại chất lượng dịch vụ được định nghĩa như ở công thức (3.2) và (3.3). Ngữ nghĩa của một phép toán trong dịch vụ được định nghĩa như sau:

$$OP(s_i, o_j) = \langle F(s_i, o_j), QoS(s_i, o_j) \rangle \quad (4)$$

### 3.3. Dịch vụ mẫu (Service Template – ST)

Một dịch vụ mẫu được tạo ra bằng cách sử dụng các ngữ nghĩa chức năng cũng như phân loại chất lượng dịch vụ cho một phép toán phù hợp với WS [7]. Dịch vụ mẫu dùng để đối sánh các dịch vụ quảng cáo với nó để tìm ra được các dịch vụ tương đồng ngữ nghĩa thỏa mãn một ngưỡng  $\alpha$  cho trước. Dịch vụ mẫu được định nghĩa như sau:

$$ST = \langle SLP(ST), OP(ST, o_1), \dots, OP(ST, o_m) \rangle \quad (5)$$

Trong đó:

- $SLP(ST)$ : Các tham số mức độ dịch vụ
- $OP(ST, o_i)$ : các ngữ nghĩa phép toán.
- $SLP(ST)$  được mô tả như sau:
- $SLP(ST) = \langle B(ST), L(ST), D(ST) \rangle \quad (6)$

Trong đó:

- $B(ST)$ : tên doanh nghiệp (Business Name) của dịch vụ cung cấp.
- $L(ST)$ : vị trí địa lý (Geographic Location) của dịch vụ.
- $D(ST)$ : miền ứng dụng của dịch vụ.
- Dịch vụ quảng cáo (Service Advertisement - SA)

Một dịch vụ quảng cáo là biểu diễn một dịch vụ thực tế được xuất bản trong UDDI [6]. Dịch vụ quảng cáo được biểu diễn bởi dưới dạng file OWL-S [1]. Dịch vụ quảng cáo được định nghĩa như sau:

$$SA = \langle SLP(SA), OP(SA, o_1), \dots, OP(SA, o_m) \rangle \quad (7)$$

Trong đó:

$OP(SA, o_i)$ : các ngữ nghĩa phép toán.

## 4. Phương pháp tính mức độ tương đồng ngữ nghĩa

Ý tưởng cơ bản của phương pháp này khi tính mức độ tương đồng ngữ nghĩa (gọi tắt là mức độ tương đồng) giữa hai dịch vụ ST và SA được thực hiện theo lần lượt 4 bước sau:

Bước 1: So sánh xem hai dịch vụ có cùng miền ứng dụng hay không? Nếu hai dịch vụ cùng miền ứng dụng thì mức độ tương đồng về miền ứng dụng của hai dịch vụ bằng 1 và tiếp tục thực hiện các bước sau, ngược lại mức độ tương đồng về miền ứng dụng của hai dịch vụ bằng 0 và không thực hiện các bước kế tiếp. Giá trị 0 cũng chính là mức độ tương đồng ngữ nghĩa của hai dịch vụ ST và SA.

Bước 2: So sánh xem hai dịch vụ có cùng vị trí địa lý hay không? Trong trường hợp này sử dụng để loại trừ khả năng đa ngôn ngữ giữa các dịch vụ. Quy trình thực hiện và kết quả so sánh trong bước này cũng được thực hiện tương tự như bước 1 ở trên.

Bước 3: So sánh các khái niệm input, output và các phép toán của hai dịch vụ với nhau.

Bước 4: So sánh các ràng buộc ngữ nghĩa giữa hai dịch vụ.

Theo [6], gọi Sim là giá trị tương đồng ngữ nghĩa của dịch vụ mẫu ST và dịch vụ quảng cáo SA. Giá trị Sim  $\in [0, 1]$  và được tính bởi công thức (8) như sau:

$$Sim(ST, SA) = \{Match(SLP(ST), SLP(SA)) * Match(OP(ST), OP(SA))\} \quad (8)$$

Trong đó:

- Match(SLP(ST), SLP(SA)) là đối sánh giữa các tham số dịch vụ.
- Match(OP(ST), OP(SA)) là đối sánh giữa các tham số phép toán.

Match(SLP(ST), SLP(SA)) bằng tích ( $\prod$ ) của phạm vi so sánh các thành phần của các tham số gồm: tên doanh nghiệp, vị trí và miền ứng dụng:

$$Match(SLP(ST), SLP(SA)) = \prod_{SLP_i \in ST} [score(SLP_i(ST), SLP_i(SA))] \quad (9)$$

$score(SLP_i(ST), SLP_i(SA)) = 1$  nếu như hai tham số  $SLP_i(ST)$  và  $SLP_i(SA)$  bằng nhau, ngược lại  $score(SLP_i(ST), SLP_i(SA)) = 0$ .

Match(OP(ST), OP(SA)) bằng tích của các  $match_{eop}$  giữa mỗi phép toán trong ST với tất cả các phép toán trong SA. Với mỗi  $match_{eop}$  chính là giá trị  $match_{op}$  lớn nhất giữa phép toán trong ST với tất cả phép toán trong SA.

$$Match(OP(ST), OP(SA)) = \prod_{i=1 \text{ to } m} (Match_{eop}(OP(ST, o_i), OP(SA))) \quad (10)$$

$$Match_{eop}(OP(ST, o_i), OP(SA)) = Max[Match_{op}(OP(ST, o_i), OP(SA, o_1)), \dots, Match_{op}(OP(ST, o_i), OP(SA, o_j))] \quad (11)$$

Trong đó,  $Match_{op}$  giữa một phép toán trong dịch vụ ST với phép toán khác trong dịch vụ SA là giá trị trung bình trọng số của  $Match_{opf}$  giữa các tham số chức năng của ST và SA và  $Match_{opq}$  giữa các tham số QoS của ST và SA, được tính như sau:

$$Match_{op}(OP(ST, o_i), OP(SA, o_j)) = (w_a * Match_{opf}(OPF(ST, o_i), OPF(SA, o_j)) + (w_b * Match_{opq}(OPQ(ST, o_i), OPQ(SA, o_j)))/(w_a + w_b) \quad (12)$$

$w_a, w_b$  là trọng số để đối sánh tham số chức năng và tham số QoS của 2 dịch vụ ST và SA.  $Match_{opf}$  giữa các tham số chức năng của hai phép toán trong ST và SA là trung bình trọng số của đối sánh các khái niệm giữa các khái niệm operation, input, output.

$$Match_{opf}(OPF(ST,o_i),OPF(SA,o_j))=[w_1*Match_C(OP(ST,oper(o_i)),OP(SA,oper(o_j))) + w_2*Match_C(OP(ST,output(o_i)),OP(SA,output(o_j))) + w_3*Match_C(OP(ST,input(o_i)),OP(SA,input(o_j)))] / (w_1 + w_2 + w_3) \quad (13)$$

$w_1, w_2, w_3$  là trọng số để  $match_C$  giữa operation, output, input của 2 dịch vụ ST và SA.  $Match_{opq}$  giữa các tham số QoS của hai phép toán trong ST và SA là trung bình trọng số của đối sánh giữa time, cost, reliability và availability.

$$Match_{opq}(OPQ(ST,o_i),OPQ(SA,o_j)) = [w_1*Match_{QT}(OP(ST,T(o_i)),OP(SA,T(o_j))) + w_2*Match_{QC}(OP(ST,C(o_i)),OP(SA,C(o_j))) + w_3*Match_{QR}(OP(ST,R(o_i)),OP(SA,R(o_j))) + w_4*Match_{QA}(OP(ST,A(o_i)),OP(SA,A(o_j)))] / (w_1 + w_2 + w_3 + w_4) \quad (14)$$

$w_1, w_2, w_3, w_4$  là trọng số để  $match_{QX}$  giữa time, cost, reliability và availability của hai dịch vụ ST và SA.  $Match_C$  giữa các tham số dữ liệu và chức năng dùng để đối sánh hai khái niệm trong một Ontology.  $Match_C$  được tính như sau:

### Tính $Match_C$ :

Đặt  $OP(ST, output(o_i)) = A$

$OP(SA, output(o_j)) = B$

Đặt  $Match_C(A,B) = Y$  và  $Y \in [0, 1]$

Giá trị của Y được tính lệ thuộc vào bảng ràng buộc sau:

**Bảng 1.** Các giá trị ràng buộc của Y

<b>Y=1</b>	<b>nếu A = B</b>
Ngược lại, $Y = \text{pow}(x, d)$	nếu B là super-class của A và $Semantic\_Distance(A,B)=d$
Ngược lại, $Y = \text{pow}(x*x, d)$	nếu A là super-class của B và $Semantic\_Distance(A,B)=d$
Ngược lại, $Y = \text{pow}(x^3, d)$	Trong đó $d=\text{khoảng cách lớn nhất của khái niệm cha chung của A và B}$

Giá trị  $x = \text{PropertyMatch}(A, B) = (\text{số thuộc tính của A giống với thuộc tính trong B} * 2) / (\text{Tổng số thuộc tính của A và B})$

Trong phương pháp này cần đến sự đánh giá của ý kiến các chuyên gia đó là giá

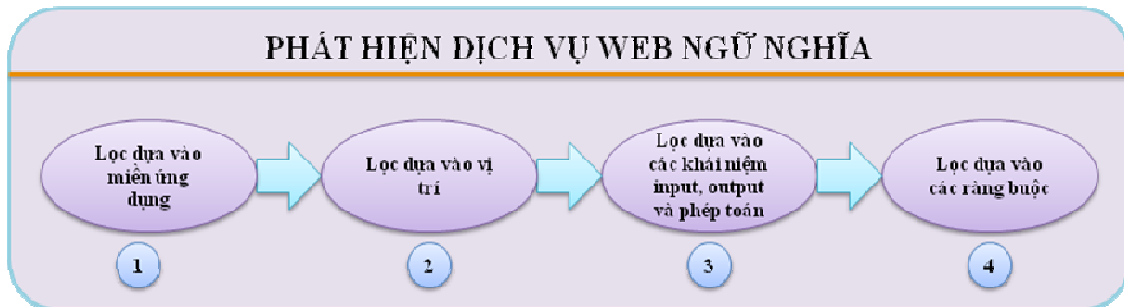
trị các giá trị trọng số  $w$ . Tùy vào từng ứng dụng cụ thể, mục tiêu của bài toán mà các chuyên gia có thể đưa ra các ý kiến cho các giá trị  $w$  khác nhau. Phương pháp này cũng đã sử dụng phương pháp tính khoảng cách giữa hai khái niệm để tính toán giá trị của  $Match_C$ .

## 5. Mô hình khung ứng dụng

Thông qua các nghiên cứu trên, chúng tôi đã xây dựng một khung ứng dụng dùng để phát hiện các dịch vụ Web ngữ nghĩa dựa trên các kết quả đối sánh ngữ nghĩa trong các ontology và làm đầu ra cho việc biên tập các dịch vụ web. Khung ứng dụng gồm ba thành phần chính: Ontology Matching, phát hiện dịch vụ Web ngữ nghĩa và Composition (biên tập) như được mô tả trong Hình 2.

*Thành phần “Ontology Matching”*: mô tả các kỹ thuật đối sánh giữa các ontology. Hiện tại có nhiều kỹ thuật, nhiều phương pháp đối sánh khác nhau. Tùy mục tiêu sử dụng mà có thể sử dụng kỹ thuật nào phù hợp với yêu cầu của ứng dụng. Trong bài báo này, chúng tôi đề xuất sử dụng phương pháp đối sánh dựa vào khoảng cách giữa các khái niệm [5].

*Thành phần “Phát hiện Dịch vụ Web ngữ nghĩa”*: là thành phần chính của khung ứng dụng, trong đó mô tả quy trình phát hiện dịch vụ Web ngữ nghĩa qua 4 giai đoạn chính như Hình 1:



**Hình 1.** Quy trình phát hiện Dịch vụ Web ngữ nghĩa

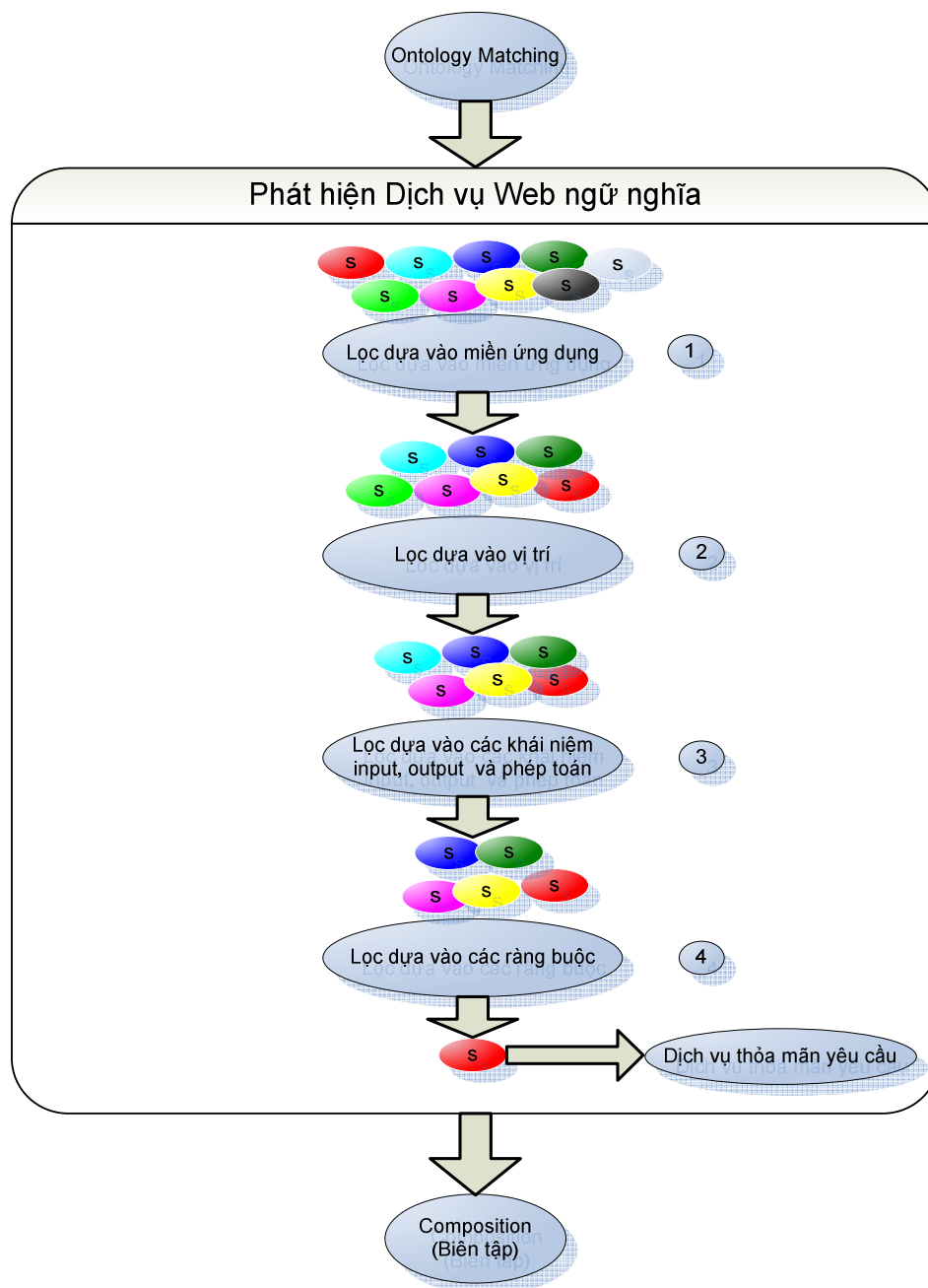
Giai đoạn lọc các dịch vụ dựa vào miền ứng dụng có nghĩa là thực hiện so sánh miền ứng dụng của dịch vụ quảng cáo với miền ứng dụng của dịch vụ yêu cầu. Trong trường hợp này, nếu dịch vụ quảng cáo không có cùng miền ứng dụng với dịch vụ yêu cầu thì sẽ không thực hiện các giai đoạn tiếp theo, ngược lại sẽ thực hiện so sánh tiếp giai đoạn (2).

Giai đoạn lọc các dịch vụ dựa vào vị trí là thực hiện so sánh vị trí của các dịch vụ quảng cáo có cùng miền ứng dụng với dịch vụ yêu cầu với vị trí của dịch vụ yêu cầu. Nếu dịch vụ quảng cáo có vị trí cùng với vị trí của dịch vụ yêu cầu thì sẽ tiếp tục thực hiện giai đoạn (3). Ngược lại, các dịch vụ quảng cáo này sẽ không tiếp tục so sánh với dịch vụ yêu cầu để tìm ra độ tương đồng.

Giai đoạn tiếp theo là lọc các dịch vụ quảng cáo kết quả của giai đoạn (2) theo các khái niệm input, output và các phép toán tương đồng với các khái niệm input, output của dịch vụ yêu cầu.

Cuối cùng hệ thống sẽ lọc các dịch vụ quảng cáo kết quả của giai đoạn (3) theo các ràng buộc ngữ nghĩa với dịch vụ yêu cầu để tìm ra dịch vụ thỏa mãn yêu cầu.

*Composition:* sau khi phát hiện ra các dịch vụ có độ tương đồng ngữ nghĩa với dịch vụ yêu cầu thì biên tập các dịch vụ để thực thi một tiến trình nghiệp vụ nào đó.



**Hình 2.** Khung ứng dụng phát hiện dịch vụ Web ngữ nghĩa.



## 6. Cài đặt thử nghiệm

Với mục đích minh họa và thử nghiệm, chúng tôi đã tiến hành cài đặt phương pháp phát hiện WS dựa vào ngữ nghĩa nhằm so sánh để tìm ra mức độ tương đồng của hai dịch vụ. Ngôn ngữ lập trình là ngôn ngữ Java, sử dụng bộ công cụ Eclipse [8] và một số thư viện trong [9] để tiến hành cài đặt.

Hệ thống đã tiến hành so sánh mức độ tương đồng của hai WS dựa trên các mô tả WS như OWL-S [1], WSDL [2]. Kết quả phát hiện ra là độ tương đồng của hai dịch vụ với nhau dựa trên độ tương đồng của các khái niệm.

Trên cơ sở đó, chúng ta có thể lọc ra các dịch vụ quảng cáo sau khi so sánh với dịch vụ mẫu mà có độ tương đồng lớn hơn hoặc bằng ngưỡng  $\alpha$  cho trước.

STT	Tên dịch vụ quảng cáo
1	C:/StockServices.wsdl
2	C:/xQuotes.wsdl
3	C:/StockQuoteService.wsdl
4	C:/156_StockQuoteNET.wsdl
5	C:/163_StockQuoteService.wsdl
6	C:/159_FastQuote.wsdl

STT	Tên dịch vụ phát hiện	Độ tương đồng
1	C:/xQuotes.wsdl	0.480
2	C:/156_StockQuoteNET.wsdl	0.452
3	C:/159_FastQuote.wsdl	0.467

Hình 3. Giao diện cài đặt thử nghiệm

## 7. Kết luận

Bài báo trình bày phương pháp phát hiện dịch vụ Web dựa vào ngữ nghĩa, sử dụng phương pháp phát tính khoảng cách ngữ nghĩa giữa các khái niệm trong Ontology phân cấp. Trên cơ sở đó, xây dựng mô hình khung ứng dụng phát hiện dịch vụ Web ngữ nghĩa. Bài báo cũng đã trình bày việc thiết kế và xây dựng công cụ thử nghiệm phương pháp phát hiện WS dựa vào ngữ nghĩa dựa trên các mô tả WS như OWL-S hay WSDL. Công cụ này đã hỗ trợ cho chúng ta biết được mức độ tương đồng giữa các khái niệm, mức độ tương đồng giữa hai dịch vụ với nhau.

## TÀI LIỆU THAM KHẢO

1. David Martin, Mark Burstein (2004). *OWL-S: Semantic Markup for Web Services*, W3C Member Submission, <http://www.w3.org>.
2. Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana (2001), *Web Services Description Language*, W3C, <http://www.w3.org>.
3. Hoàng Hữu Hạnh. *Web ngữ nghĩa: những thách thức và hướng tiếp cận mới*, Tạp chí Khoa học Đại học Huế, 48, (2008), 31-40.
4. J. Zhong, H. Zhu, J. Li and Y. Yu. *Conceptual graph matching for semantic search*, The 2002 International Conference on Computational Science (ICCS2002), Amsterdam, (2002), 92-106.
5. Jike Ge, Yuhui Qiu. *Concept Similarity Matching Based on Semantic Distance*, Proceedings of the 2008 Fourth International Conference on Semantics, Knowledge and Grid, 2008.
6. L. Clement, A. Hately, C. Riegen, T. Rogers (2004), *UDDI Spec Technical Committee Draft*, [http://www.uddi.org/pubs/uddi\\_v3.htm](http://www.uddi.org/pubs/uddi_v3.htm).
7. Rohit Aggarwal, Kunal Verma, John Mille. *Constraint Driven Web Service Composition in METEOR-S*, Proceedings of IEEE International Conference on Services Computing, 2004.
8. <http://www.eclipse.org/>
9. <http://jena.sourceforge.net/>

## SEMANTIC WEB SERVICES ISCOVERY APPROACHES FOR CROSS-ENTERPRISE COLLABORATION

*Mai Van Muoi*  
*College of Sciences, Hue University*  
*Hoang Huu Hanh*  
*Hue University*

### SUMMARY

*This paper presents discovery methods for the Semantic Web Services through the usage of the semantic distance between concepts in an ontology hierarchy. Based on the analysis, the application model of semantic web services discovery is developed. In addition, this paper proposes and implements a prototype application for Semantic Services Web discovery on the basis of Web Services descriptions, such as OWL-S and WSDL. This tool helps users know similarities between concepts and two Web services.*