

PHƯƠNG PHÁP XÂY DỰNG ĐỘ ĐO TÁI SỬ DỤNG CỦA WEB SERVICE

PHẠM THỊ QUỲNH, HUỖNH QUYẾT THẮNG, TẠ QUANG DŨNG

Viện Công nghệ thông tin - truyền thông, Đại học Bách khoa Hà Nội

Abstract. Reusing available services is one of the best advantages in service-oriented software development process. Therefore, evaluation of services reusability plays an important role. From reusability point, it gives indication for developers to choose the best service among group of services which provide the same function. However, it is difficult to apply reusability metrics of traditional software in service-oriented software because of its black-box nature. In this paper, we propose a new approach to measure the reusability of service based on its coupling and cohesion. Several experiments were performed to prove the reliability of our metrics.

Tóm tắt. Việc sử dụng lại các web service trong quá trình phát triển phần mềm hướng dịch vụ là một trong những ưu điểm lớn nhất của loại hình phần mềm này. Khả năng tái sử dụng của web service phụ thuộc chủ yếu vào tính kết nối (coupling) và tính gắn kết (cohesion). Trong phạm vi bài báo này, chúng tôi đề xuất một phương pháp xây dựng độ đo khả năng tái sử dụng của web service dựa trên tính kết nối và tính gắn kết của nó. Một số thực nghiệm cũng đã được tiến hành để chứng minh tính hợp lý của độ đo đã đề xuất.

1. GIỚI THIỆU

Dịch vụ phần mềm (web service), còn gọi tắt là dịch vụ, là một đơn vị phần mềm hoàn chỉnh với những giao tiếp được xác định từ trước. Trong kiến trúc hướng dịch vụ, các dịch vụ được tích hợp với nhau, cùng hoạt động để thực hiện các yêu cầu nghiệp vụ [1]. Những dịch vụ này có thể đã có sẵn. Việc tái sử dụng lại chúng giúp giảm thời gian, chi phí và tránh được rủi ro trong quá trình xây dựng phần mềm. Đây chính là một đặc điểm nổi trội của phương pháp phát triển phần mềm hướng dịch vụ.

Tái sử dụng là mức độ được sử dụng lại của một thứ gì đó [2]. Vì vậy, khả năng tái sử dụng dịch vụ là khả năng của dịch vụ có thể tham gia vào nhiều hệ thống khác nhau. Tức là dịch vụ có thể dễ dàng tích hợp vào nhiều hệ thống. Một cách đơn giản, chúng ta có thể hiểu nếu một dịch vụ có thể sử dụng hoặc hữu ích đối với nhiều hệ thống thì khả năng tái sử dụng của dịch vụ đó cao.

Tuy nhiên, vấn đề cốt lõi ở đây là làm thế nào mà người sử dụng nhận diện được dịch vụ nào có khả năng tái sử dụng cao nhất trong số các dịch vụ cùng đáp ứng các yêu cầu nghiệp vụ. Do vậy, việc đo lường khả năng tái sử dụng của dịch vụ để hiện thực hóa hoạt động tái sử dụng dịch vụ một cách hiệu quả đóng một vai trò hết sức quan trọng.

Bài báo đề xuất một tập độ đo khả năng tái sử dụng dịch vụ. Bố cục của bài báo như sau: Mục 2 trình bày các thuộc tính liên quan đến khả năng tái sử dụng, tập độ đo được đề xuất trong Mục 3, Mục 4 trình bày các thực nghiệm trên cơ sở độ đo đã đề xuất và cuối cùng là hướng phát triển.

2. CÁC THUỘC TÍNH LIÊN QUAN ĐẾN KHẢ NĂNG TÁI SỬ DỤNG

Khả năng tái sử dụng của phần mềm được quyết định bởi các thuộc tính như: dễ hiểu, hoàn thiện, đáng tin cậy, kết nối lỏng và gắn kết chặt, dễ tích hợp vào các môi trường khác nhau [3]. Tuy nhiên, các thuộc tính này lại ảnh hưởng và có tác động qua lại lẫn nhau. Do đó, theo [4] và [5], các yếu tố chính quyết định đến khả năng tái sử dụng của dịch vụ bao gồm: tính kết nối (coupling) và tính gắn kết (cohesion).

2.1. Tính kết nối

Tính kết nối đề cập tới mức độ độc lập giữa các đơn vị phần mềm. Giảm thiểu tính kết nối sẽ tạo ra các thành phần phần mềm kết nối lỏng với nhau; điều đó được coi là thuộc tính quan trọng của một phần mềm có chất lượng cao. Kết nối lỏng làm tăng khả năng tái sử dụng và dễ được chấp nhận trong các hệ thống phân tán.

Phần mềm hướng dịch vụ được xây dựng nhờ sự kết nối các dịch vụ đơn lẻ. Không giống như các kiến trúc phần mềm khác như CORBA, DCOM, liên kết các dịch vụ trong kiến trúc SOA hết sức lỏng lẻo. Các dịch vụ cần kết nối được biết thông qua địa chỉ URL, và kết nối này chỉ được duy trì khi cần thiết chứ không phải mọi lúc. Sự liên kết lỏng lẻo này giúp cho phần mềm hướng dịch vụ đơn giản hơn trong việc bảo trì và kiểm thử, bởi vì chúng được thực hiện trên các mô-đun độc lập, không ảnh hưởng tới các mô-đun khác trong phần mềm.

Có ba dạng kết nối phổ biến thường gặp trong kiến trúc hướng dịch vụ:

- Kết nối dữ liệu (data coupling): kết nối dữ liệu đề cập tới sự trao đổi dữ liệu giữa các mô-đun phần mềm; tức là việc truyền tham số dữ liệu tới một phương thức của dịch vụ. Chúng ta có thể tối thiểu hóa việc kết nối dữ liệu bằng cách chỉ trao đổi những dữ liệu cần thiết cho phương thức của dịch vụ để thực hiện nhiệm vụ.
- Kết nối dấu hiệu (stamp coupling): kết nối dấu hiệu là trường hợp dữ liệu được trao đổi giữa các dịch vụ có dạng một cấu trúc phức.
- Kết nối điều khiển (control coupling): kết nối điều khiển là trường hợp một tham số điều khiển việc thực thi logic của dịch vụ. Logic điều khiển nên được cài đặt trực tiếp trong một dịch vụ hoặc sử dụng BPEL để kết hợp các dịch vụ; không nên thông qua các tham số điều khiển.

Tóm lại, để tăng khả năng tái sử dụng của dịch vụ thì cần phải giảm tính kết nối dữ liệu, tránh kết nối dấu hiệu và không sử dụng kết nối điều khiển. Giảm kết nối sẽ làm tăng khả năng tái sử dụng bằng cách tạo ra các giao diện của dịch vụ đơn giản với ít tham số dữ liệu.

2.2. Tính gắn kết

Tính gắn kết được đề cập lần đầu tiên trong lĩnh vực thiết kế mô-đun bởi Stevens [6]. Tính gắn kết của một mô-đun thể hiện mối tương quan giữa các thành phần trong mô-đun này.

Nguyên tắc thiết kế một phần mềm tốt là tạo ra một mô-đun có tính gắn kết cao, đặc biệt là các mô-đun chức năng. Dịch vụ có tính gắn kết càng cao thì ứng dụng càng ổn định vì tính gắn kết có thể hạn chế các thay đổi gây ảnh hưởng đến một số lượng nhỏ các dịch vụ.

Sau đây là một số dạng gắn kết:

- Gắn kết chức năng: là trường hợp tất cả các phương thức của dịch vụ có quan hệ chặt chẽ với nhau và cùng đóng góp vào việc thực hiện một nhiệm vụ đã được định nghĩa rõ ràng [7].
- Gắn kết thông tin: là trường hợp tất cả các phương thức của dịch vụ sử dụng chung dữ liệu giống nhau. Gắn kết thông tin có quan hệ rất gần với gắn kết chức năng [5].

Tóm lại, tính gắn kết và kết nối của dịch vụ có một mối quan hệ khăng khít. Tính gắn kết yếu sẽ làm tăng kết nối. Ngược lại, liên kết yếu sẽ khiến các chức năng bị chông chéo.

3. ĐO LƯỜNG KHẢ NĂNG TÁI SỬ DỤNG DỊCH VỤ

Hiện nay, đã có nhiều đề xuất về những độ đo tái sử dụng khác nhau thích hợp với từng loại mô hình phần mềm. Ví dụ, Chi-damber và Kemerer (C&K) [8] đã đưa ra một tập độ đo được áp dụng trên mô hình hướng đối tượng. Washizaki [9] đã đề xuất một tập độ đo khả năng tái sử dụng của thành phần và xây dựng framework nhằm hướng dẫn cách sử dụng tập độ đo đó.

Tuy nhiên, các tập độ đo khả năng tái sử dụng này chỉ thích hợp với các mô hình phần mềm mà các tác giả tập trung đến. Chúng ta không thể sử dụng trực tiếp chúng trên kiến trúc phần mềm hướng dịch vụ vì đặc điểm khác biệt của loại phần mềm này. Ví dụ, chúng ta không thể phân tích mã lệnh như phép đo LCOM, CBO trong tập độ đo C&K [8] đã thực hiện vì kiến trúc hướng dịch vụ coi các dịch vụ là những hộp đen và chỉ cung cấp cách giao tiếp với dịch vụ đó. Hoặc độ đo EMI của Washizaki [9] là không cần thiết trong mô hình hướng dịch vụ vì các thông tin về dịch vụ là bắt buộc phải được cung cấp.

Tuy nhiên, chúng tôi vẫn tiếp tục thừa kế ý tưởng của các tác giả trên và đề xuất một tập độ đo mới sao cho thích hợp và có thể thực hiện được trên mô hình kiến trúc hướng dịch vụ.

Theo cách nhìn nhận trực tiếp, chúng ta có thể đánh giá khả năng tái sử dụng của một dịch vụ X bằng cách phân tích các thông tin thống kê sau:

- Số lượng các dịch vụ khác đã sử dụng dịch vụ X,
- Tấn suất mà các dịch vụ khác sử dụng dịch vụ X.

Những thông tin này cho phép chúng ta đánh giá mức độ thành công của một dịch vụ

trên khía cạnh đầu tư, chi phí và lợi nhuận. Tuy nhiên, phương pháp thống kê này không có ý nghĩa trong việc đánh giá trước khả năng tái sử dụng dịch vụ.

Như trong Mục 2 đã trình bày, hai yếu tố quyết định đến khả năng tái sử dụng của dịch vụ dựa là tính kết nối và tính gắn kết. Vì vậy, để đo lường khả năng tái sử dụng của dịch vụ, chúng ta có thể đánh giá gián tiếp thông qua hai yếu tố trên.

Chú ý rằng việc đánh giá các thuộc tính của một dịch vụ theo khía cạnh tĩnh (tức là không phải trong trường hợp đang thực thi dịch vụ) và chỉ có thể dựa vào một nguồn thông tin duy nhất đó là file định nghĩa dịch vụ WSDL. Vì vậy, tập phép đo mà chúng tôi đề xuất hoàn toàn là các phép đo hộp đen, không có sự phân tích hay can thiệp vào mã lệnh, mà chỉ đơn giản là sử dụng những tài liệu cho phép công bố công khai.

3.1. Tập độ đo tính kết nối

Tập hợp độ đo tính kết nối được xây dựng dựa trên từng loại kết nối đã được định nghĩa trong phần 2. Đồng thời, chúng tôi áp dụng phương pháp định nghĩa độ đo SCCp của Washizaki [9] lên tập độ đo tính kết nối.

Độ đo tính kết nối dữ liệu của dịch vụ s (Data Coupling Metric $DCM(s)$) là tỷ số giữa số lượng các phương thức không có tham số input trên tổng số các phương thức trong dịch vụ.

$$DCM(s) = \frac{\sum OP_{none_input}}{\sum OP}. \quad (1)$$

Độ đo tính kết nối dấu hiệu của dịch vụ s (Stamp Coupling Metric $SCM(s)$) là tỷ số giữa số lượng các phương thức không có tham số kiểu phức trên tổng số các phương thức trong dịch vụ.

$$SCM(s) = \frac{\sum OP_{none_complex_message}}{\sum OP}. \quad (2)$$

Tính kết nối điều khiển thường bị thay thế bằng việc sử dụng các quy trình BPEL. Vì vậy, người thiết kế có thể tránh được loại kết nối này. Vì vậy, chúng tôi không đề xuất độ đo kết nối điều khiển.

Từ định nghĩa (1) và (2), chúng ta thấy rằng giá trị của các độ đo chỉ nằm trong khoảng $[0, 1]$. Giá trị của các độ đo này càng nhỏ thì kết nối càng lỏng và khả năng tái sử dụng càng cao. Ví dụ sau mô tả cách tính độ đo tính kết nối của dịch vụ GoogleSearch.

Ví dụ: <http://api.google.com/GoogleSearch.wsdl> có 3 phương thức.

Xét phương thức 1

```
<operation name="doGetCachedPage">
  <input message="typens:doGetCachedPage" />
```

```
<output message="typens:doGetCachedPageResponse" />
</operation>
```

trong đó

```
<message name="doGetCachedPage">
  <part name="key" type="xsd:string" />
  <part name="url" type="xsd:string" />
</message>
```

là message không phải kiểu phức vì các thành phần của nó đều lấy từ xsd. Và

```
<message name="doGetCachedPageResponse">
  <part name="return" type="xsd:base64Binary" />
</message>
```

không những chỉ có một thành phần mà kiểu dữ liệu của thành phần đó cũng là kiểu lấy từ xsd.

Vì vậy phương thức 1 được coi là phương thức không có tham số kiểu phức.

Tương tự như vậy với phương thức thứ 2.

Với phương thức thứ 3

```
<operation name="doGoogleSearch">
  <input message="typens:doGoogleSearch" />
  <output message="typens:doGoogleSearchResponse" />
</operation>
```

Có message doGoogleSearchResponse là tham số kiểu phức. Vì vậy, phương thức này là phương thức có chứa tham số kiểu phức.

Vậy độ đo $DCM(s) = 0$ và $SCM(s) = 2/3$.

3.2. Tập độ đo tính gắn kết

Trong mô hình phần mềm hướng đối tượng, việc đánh giá tính gắn kết của một lớp thông thường dựa trên mức độ gắn kết giữa các phương thức của lớp đó. Ví dụ, độ đo LCOM trong tập độ đo C&K [8] được tính toán dựa trên số lượng biến dùng chung, độ đo PSI [10] dựa trên số lượng tham số dùng chung giữa các phương thức. Những nhận xét này giúp chúng tôi đề xuất ý tưởng xây dựng độ đo tính gắn kết của dịch vụ. Tính gắn kết của dịch vụ sẽ được đánh giá thông qua các phương thức mà dịch vụ đó cung cấp.

Trong mô hình phần mềm hướng dịch vụ, chúng tôi không thể phân tích được mã lệnh nên không thể biết được nội dung phương thức được cài đặt cụ thể như thế nào và do đó, không thể biết được số lượng biến dùng chung giữa các phương thức như cách mà LCOM thực hiện. Chúng tôi chỉ có thể dựa vào các tham số mà phương thức sử dụng như cách của PSI, nhưng ở đây chúng tôi nhấn mạnh đến các tham số đầu vào và đầu ra để phù hợp với đặc điểm định nghĩa của dịch vụ.

Để xây dựng các độ đo, chúng tôi cần phải định nghĩa một số kí hiệu hình thức. Gọi s là kí hiệu của dịch vụ cần đo. M là tập hợp các phương thức được khai báo trong s (giả sử gồm n phương thức). Mg là tập hợp các thông điệp đã được khai báo trong s (giả sử gồm m thông điệp).

Gọi P là tập hợp các tham số của các phương thức trong tập M . Mỗi phần tử trong P là một bộ gồm 2 thành phần (Mg_i, v) ; trong đó v có hai giá trị đầu vào (I) hoặc đầu ra (O) tương ứng với 2 kiểu truyền tham số và $Mg_i \subset Mg$.

Tiếp theo, chúng tôi xây dựng ma trận có các cột là các phương thức thuộc tập M và các hàng là các phần tử thuộc tập P . Gọi $p_{i,j}$ là giá trị của phần tử trong ma trận. Nếu phương thức M_i có chứa thông điệp M_j và có kiểu truyền là input thì $p_{i,j} = 1$, có kiểu truyền là output thì $p_{i,m+j} = 1$; trong các trường hợp còn lại $p_{i,j} = 0$.

Gọi $c_{i,j}$ là hệ số gắn kết giữa hai phương thức M_i và M_j , nhằm xác định hai phương thức này có sử dụng chung các tham số nào. Do đó, $c_{i,j}$ hay $c_{j,i}$ là như nhau. $c_{i,j}$ được tính theo công thức sau

$$c_{i,j} = \frac{\sum_{k=1}^{2m} p_{i,k} \times p_{j,k}}{\sum_{k=1}^{2m} p_{i,k} + \sum_{k=1}^{2m} p_{j,k} - \sum_{k=1}^{2m} p_{i,k} \times p_{j,k}}, \quad (3)$$

trong đó, giá trị $\sum_{k=1}^{2m} p_{i,k} \times p_{j,k}$ cho biết hai phương thức M_i và M_j có bao nhiêu tham số có

kiểu dữ liệu và kiểu truyền giống nhau. Và giá trị $\sum_{k=1}^{2m} p_{i,k} + \sum_{k=1}^{2m} p_{j,k} - \sum_{k=1}^{2m} p_{i,k} \times p_{j,k}$ cho biết tổng số tham số có trong cả hai phương thức M_i và M_j .

Gọi c_i là hệ số gắn kết giữa phương thức M_i với các phương thức khác thuộc tập M

$$c_i = \sum_{j=1, j \neq i}^n c_{i,j}. \quad (4)$$

Cuối cùng, độ đo tính gắn kết dựa theo chức năng của dịch vụ s (ChM(s) - Cohesion Metric) được tính theo công thức

$$ChM(s) = \frac{\sum_{i=1}^n c_i}{n}. \quad (5)$$

Giá trị của độ đo ChM càng lớn cho thấy tính gắn kết của dịch vụ s được đo càng chặt chẽ và khả năng tái sử dụng nó càng cao. Ví dụ sau đây minh họa cách tính độ đo ChM của một dịch vụ.

Ví dụ: <http://api.google.com/GoogleSearch.wsdl>.

M là tập các phương thức: `doGetCachedPage`, `doSpellingSuggestion`, `doGoogleSearch`.
 Như vậy, $n = 3$.

Mg là tập các thông điệp. Có thể thấy là có 6 thông điệp. Nhưng chúng ta đang xét đến kiểu dữ liệu. Vì có 2 thông điệp cùng kiểu nên $m = 5$.

P là tập hợp các tham số trong các phương thức M . Mỗi phần tử trong P có dạng (Mg_i, v) .

Để đơn giản, ký hiệu các phương thức là m_1, m_2, m_3 . Các kiểu thông điệp ký hiệu là $mg1, mg2, mg3, mg4, mg5$.

Phương thức m_1 có 2 phần tử trong P là $(mg1, I)$ và $(mg2, O)$

Phương thức m_2 có 2 phần tử trong P là $(mg1, I)$ và $(mg3, O)$

Phương thức m_3 có 2 phần tử trong P là $(mg4, I)$ và $(mg5, O)$

Từ đó ta có thể xây dựng ma trận 3×10 như sau:

	mg1,I	mg2,I	mg3,I	mg4,I	mg5,I	mg1,O	mg2,O	mg3,O	mg4,O	mg5,O
M1	1						1			
M2	1							1		
M3				1						1

Ngoài các giá trị 1 ở trên, còn lại là các giá trị 0. Sau đó tính $c_{i,j}$, ví dụ $c_{1,2} = \frac{1}{3}$.

$c_{1,3}$ và $c_{2,3}$ đều bằng 0 vì không có tham số nào giống nhau giữa hai cặp phương thức tương ứng (m_1, m_3) và (m_2, m_3) .

Sau đó tính c_i .

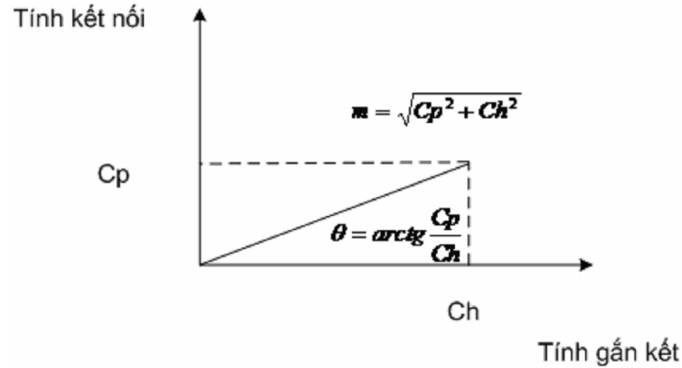
Ví dụ $c_1 = c_{1,2} + c_{1,3} = \frac{1}{3}$, $c_2 = c_{2,1} + c_{2,3} = c_{1,2} + c_{3,2} = \frac{1}{3}$, $c_3 = c_{3,1} + c_{3,2} = c_{1,3} + c_{2,3} = 0$.

Cuối cùng $ChM(s) = \frac{\frac{1}{3} + \frac{1}{3} + 0}{3} = \frac{2}{9}$.

3.3. Độ đo khả năng tái sử dụng

Xét một dịch vụ s có giá trị độ đo tính kết nối là $DCM(s)$ và $SCM(s)$ và giá trị độ đo tính gắn kết là $ChM(s)$. Theo định nghĩa của chúng, giá trị của độ đo tính kết nối càng nhỏ thì khả năng tái sử dụng càng cao và chú ý rằng giá trị của nó nằm trong khoảng từ 0 đến 1. Ngược lại, nếu độ đo tính gắn kết càng lớn thì khả năng tái sử dụng của dịch vụ càng cao. Vì vậy, để xây dựng một hàm đồng biến, chúng tôi đề xuất độ đo tính kết nối CpM (Coupling Metric) được tính theo công thức: $CpM(s) = 1 - DCM(s)$ hoặc $CpM(s) = 1 - SCM(s)$.

Theo [11], chúng ta sẽ xây dựng độ đo khả năng tái sử dụng dựa trên vector hai chiều - một chiều là $ChM(s)$ và một chiều là $CpM(s)$. Độ lớn m của vector này phản ánh giá trị của độ đo khả năng tái sử dụng. Độ lớn của góc $theta$ cho biết mối tương quan của hai thuộc tính kết nối và gắn kết tác động lên khả năng tái sử dụng.



Hình 1. Phương pháp xác định độ đo tái sử dụng của dịch vụ

Đơn vị tỷ lệ của hai trục trong trường hợp này được coi bằng 1. Tuy nhiên, trong trường hợp tổng quát có thể gán bằng i, j tương ứng với các trục Ch và Cp .

Như vậy, độ đo tái sử dụng của một dịch vụ SRM (Services Reusability Metric) được tính dựa trên các thuộc tính kết nối và gắn kết như sau

$$SRM(s) = \sqrt{CpM(s)^2 + ChM(s)^2}, \quad (6)$$

trong đó, $CpM(s)$ và $ChM(s)$ là độ đo tính kết nối và tính gắn kết của dịch vụ s .

Giá trị của độ đo SRM phản ánh khả năng tái sử dụng của dịch vụ được đo. Nếu giá trị này càng lớn thì khả năng tái sử dụng của dịch vụ đó càng cao.

4. THỬ NGHIỆM VÀ ĐÁNH GIÁ

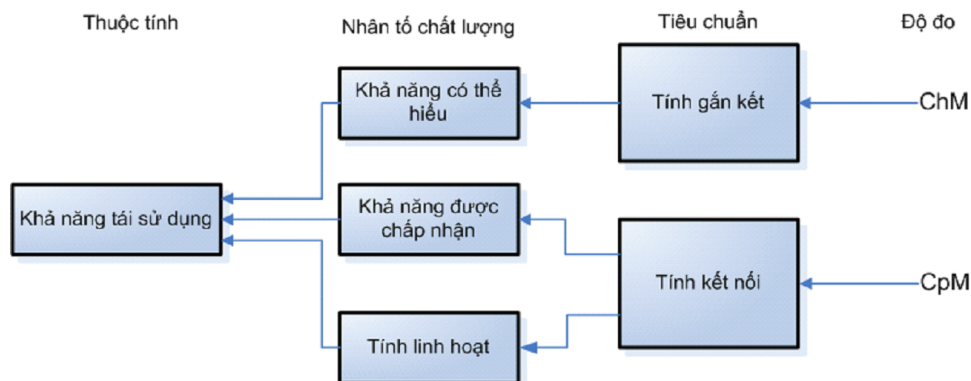
Theo mô hình chất lượng ISO-9126 [12], chất lượng được định nghĩa là một tập các thuộc tính, đặc điểm của sản phẩm đảm bảo khả năng thỏa mãn các yêu cầu đã đề ra. Tiêu chuẩn ISO-9126 đề xuất 6 thuộc tính chất lượng ở mức cao nhất và được xác định thông qua tập các phép đo. Bằng cách sử dụng các phép đo, người ta có thể xác định được các thuộc tính chất lượng của phần mềm.

Để thuận tiện hơn cho việc ước lượng và đánh giá cũng như làm tăng khả năng dễ hiểu cho các thuộc tính chất lượng, các thuộc tính chất lượng được chia thành các thuộc tính con. Các thuộc tính con lại được chia nhỏ, và quá trình này được thực hiện cho đến khi ở bước cuối cùng là các thuộc tính có thể tính toán được. Các thuộc tính có thể tính toán này được gọi là các phép đo.



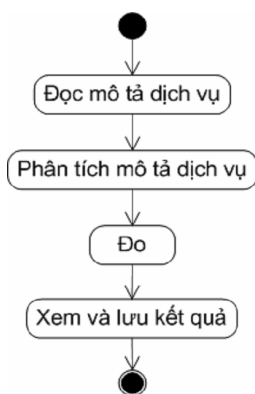
Hình 2. Quan hệ giữa các thành phần trong mô hình chất lượng ISO-9126

Dựa trên tiêu chuẩn ISO-9126 và mô hình FCM (Factor-Criteria-Metrics) [13], chúng tôi đề xuất mô hình nhằm định hướng cách sử dụng tập độ đo đã được đề xuất trong Mục 3.



Hình 3. Mô hình đánh giá khả năng tái sử dụng của dịch vụ

Tiếp theo, ta tiến hành xây dựng công cụ đo và thực hiện một số thử nghiệm. Đầu vào của quy trình đo là địa chỉ File mô tả web service WSDL. Sau đó, file này sẽ được xử lý, phân tích để thu được các dữ liệu cần thiết cho việc thực hiện các thuật toán tập các độ đo. Cuối cùng, kết quả đo được lưu trữ và hiển thị.



Hình 4. Quy trình tính toán độ đo

Tiến hành thực nghiệm qua các bước sau [14]:

1) Lựa chọn dữ liệu: hiện tại, có một số trang web cho phép đăng ký thông tin về các web service được công bố. Các thông tin này bao gồm tên, nhà cung cấp, mô tả, URL chỉ đến file WSDL của web service. Vì vậy, những trang web này sẽ là nơi chúng ta thu thập dữ liệu. Bao gồm các trang web sau:

- www.xmethod.com
- www.webservicex.com
- www.webservicelist.com
- www.seekda.com

2) Phân loại và loại bỏ dữ liệu không hợp lệ: theo thống kê của [19], các web service được phân loại theo miền ứng dụng. Vì vậy, chúng ta sẽ phân loại dữ liệu thu thập được trong bước 1 theo miền ứng dụng hoặc chức năng mà web service đảm nhận. Sau đó, chúng ta tiến hành loại bỏ các web service trùng lặp vì một web service có thể được đăng ký tại nhiều trang web nên có khả năng xảy ra hiện tượng này. Phương pháp loại bỏ là loại bỏ các web service có cùng chức năng và có cùng provider.

3) Đo lường: sử dụng các độ đo đã đề xuất để tiến hành đo lường độ phức tạp và khả năng tái sử dụng của chúng.

4) Thống kê và đánh giá: đã thống kê số lượng các web service được sử dụng làm mẫu, số lượng còn lại sau quá trình chọn lọc, số lượng theo chức năng. Hiển thị các đồ thị cột các giá trị của từng phép đo để so sánh kết quả thu được theo các chức năng. Và cuối cùng là nhận xét và đánh giá kết quả đo được.

Bảng sau đây mô tả kết quả đo khả năng tái sử dụng của các web service được phân loại theo miền chức năng từ các nhà cung cấp khác nhau.

Bảng 1. Một số kết quả thử nghiệm

Nhà cung c?p	Ch?c nang	Coupling Metric (CpM)		Cohesion Metric (ChM)	Service's Reusability Metric (SRM)
		DCM	SCM		
XMLLogic	Email Validation	0	0,5	0,167	0,527
Webservicex	Email Validation	0	0,667	0,667	0,745
StrikeIron	Email Validation	0	0,25	0,167	0,768
CDYNE	Email Validation	0	0,167	1,222	1,479
Xwebservice	Email Validation	0	0	0	1
ServiceObjects	Email Validation	0	0	0,667	1,202
Webservicex	Credit Card Validation	0	0,667	0,667	0,745
Fraudlabs	Credit Card Validation	0	0	0,667	1,202
Primerchants	Credit Card Validation	0	0	3,873	4
CDYNE	IP to geology	0	0	0,667	1,202
Webservicex	IP to geology	0	0	1,222	1,579
Fraudlabs	IP to geology	0	0	0,667	1,202

Kết quả thực nghiệm cho thấy giá trị độ đo DCM luôn bằng 0, tức là hầu hết các phương thức của web service đều có tham số đầu vào; vì vậy, nó không có ý nghĩa trong việc đánh giá khả năng kết nối của web service. Chúng tôi sử dụng độ đo SCR để đại diện cho khả năng kết nối. Cho nên, công thức tính của CpM là $CpM = 1 - SCM$.

Theo kết quả đo, trong 6 web service cùng thực hiện chức năng Email Validation thì CDYNE có khả năng tái sử dụng cao nhất và XMLLogic có khả năng tái sử dụng thấp nhất.

Ta có thể sử dụng các giá trị đo này để có một lựa chọn về các web service thuộc cùng chức năng nhưng có khả năng tái sử dụng tốt hơn.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, ta đã xây dựng tập độ đo khả năng tái sử dụng của dịch vụ dựa trên tính kết nối và tính gắn kết của nó. Một số thực nghiệm đã được tiến hành nhằm đánh giá độ tin cậy của độ đo này.

Tiếp tục xây dựng các tập độ đo khác, giúp đo lường các thuộc tính khác của dịch vụ như độ phức tạp, độ tin cậy... Tập hợp các độ đo này sẽ giúp đánh giá dịch vụ trên mọi khía cạnh một cách chính xác.

Trong quá trình thực nghiệm, ta chỉ sử dụng các dịch vụ được cung cấp miễn phí. Tới đây, sẽ triển khai công cụ đo lường này để nó có thể được sử dụng rộng rãi. Ngoài ra, các kết quả đo lường cần được so sánh với những đánh giá thực tiễn khi sử dụng dịch vụ trong thực tiễn.

TÀI LIỆU THAM KHẢO

- [1] Eric Newcomer, *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, Addison-Wesley Professional Publisher, 2002 (p.368).
- [2] Arun Sharma, Rajesh Kumar, P. S. Grover, A critical survey of reusability aspects for component-based systems, *Journal of World Academy of Science, Engineering and Technology* (33) (2007) 35–39.
- [3] Poulin, Jeffrey S., *Measuring Software Reuse: Principles, Practices, and Economic Models*, Addison-Wesley, Reading, MA, 1996 (p.224).
- [4] M. Aoyama et al., Software commerce broker over the internet, *Proc. of 22nd Annual International Computer Software and Applications Conference, IEEE CS*, Vienna, Austria, August 19-21, 1998 (430–435).
- [5] George Feuerlicht, Josip Lozina, “Understanding Service Reusability”, Faculty of Information Technology, University of Technology, Sydney, Australia,
<http://si.vse.cz/archive/proceedings/2007/understanding-service-reusability.pdf>
- [6] W. Stevens, G. Myers, L. Constantine, *Structured Design, IBM Systems Journal* **38** (1999) (2/3) 231–256.
- [7] M.P. Papazoglou, J. Yang, Design methodology for web services and business processes, *Proceedings of the 3rd VLDB-TEW Workshop*, Hong Kong, Springer, 2002 (54–64).
- [8] S. Chidamber and C. Kemerer, A metrics suite for object oriented design, *IEEE Transactions on Software Engineering* **20** (6) (1994) 476–493.

- [9] Hironori Washizaki, A metrics suite for measuring reusability of software components, *Proceedings of the 9th International Symposium on Software Metrics*, Washington, DC, USA, September 03 - 05, 2003 (p.211–225).
- [10] Cara Stein, Letha Etzkorn and others, A knowledge-based cohesion metric for object-oriented software, *Journal of Computer Science* **5** (4) (December 2006) 44–53.
- [11] T.E. Hastings, A.S.M. Sajeev, A vector-based approach to software size measurement and effort estimation, *IEEE Transactions on Software Engineering* **27** (4) (April 2001) 337–350.
- [12] Ho-Won Jung, Seung-Gweon Kim, Chang-Shin Chung, Measuring software product quality: A survey of ISO/IEC 9126, *IEEE Software* **21** (5) (Sep./Oct. 2004) 88–92.
- [13] J.A. McCall, P.K. Richards, G.F. Walters, Factors in Software Quality, US Rome Air Development Center Reports, Vol. I, II, III, RADC-TR-77-369, (1977), <http://www.dtic.mil/dtic/>.
- [14] Jianchun Fan, Subbarao Kambhampati, A snapshot of public Web services, *SIGMOD Record* **34** (1) (March 2005) 24–32.

Nhận bài ngày 29 - 12 - 2009

Nhận lại sau sửa ngày 22 - 6 - 2010