The Determination of Service Couplings for Non-disruptive Systems

Thunchira Thongmee, Hiroto Suzuki, Takahiro Ohno
Graduate School of Creative Science and Engineering, Waseda University, Tokyo, Japan
thunchira@moegi.waseda.jp, hiroto-suzuki@aoni.waseda.jp, ohno@waseda.jp

Udom Silparcha
School of Information Technology, King Mongkut's University of Technology Thonburi, Bangkok, Thailand
udom@sit.kmutt.ac.th

Keywords: Service Oriented Architecture, Degree of Coupling, Non-disruptive.

Abstract: Business continuity becomes an important feature for business these days. It is difficult to accept unplanned downtime due to the unavailability of the service, which mostly is unavoidable. The unavailable of the service is the situation such as service failure, unplanned downtime, etc. When such situation occurs, the system will need to look for some solution that will allow the system to continue. A simple but effective solution is to provide at least one alternative path of services that performs similar functions to the usual services which become unavailable. Therefore, switching between service paths is crucial in any reliable system. Service quality of a system or task can be determined with various factors. One of the most common factors is service coupling. This study will analyse the changes in couplings upon service switching, which in turn reflect the overall quality of the system.

1 INTRODUCTION

Service technology is widely adopted by business these days. One of the important features to determine the service quality is couplings among services or modules. Couplings refer to the dependencies between two or more services, which normally exist in software applications (Erl, 2008). There are loose couplings and tight couplings, which imply how much the modules are dependent on each other. Such levels of dependencies can be described as the "degree" of coupling between services, where a loose coupling contributes a lower degree of coupling than a tight coupling. The degree of coupling can be used to determine the availability, reliability, performance, usability, discoverability, and maintainability of the services within a system (Xu et al., 2006). The degree of coupling also identifies possible risks, and the estimated cost for development and maintenance (Yang et al., 2005). In general, the lower degree of coupling implies the better system design. Thus, there are many researches focus on the degree of coupling in order to be able to minimize the degree of coupling in their tasks.

In a service-oriented environment, any service interruption may cause some loss in business due to the business disruption. Therefore, for business continuity, a highly available system should consider providing at least one alternative service for each critical one (Wang et al., 2009). That means when a critical service becomes unavailable, there should be some other service that can provide the equivalent functionality. The unavailability of the service indicates the situations such as the service down, resource full (unable to allocated), or when the communication lost occurs. In this study, we discuss about the service switching between the critical service and the alternative services, by focusing on the analysis of its impact on the degree of coupling within the same task and among different tasks within a system. By analyzing such an impact, a system designer will have a more proper insight of how reliable the system truly is.

2 SERVICE COUPLINGS

There are a number of researches in the literature that have proposed techniques to determine the quality of a system design (Yang et al., 2005; Rich

Thongmee T., Suzuki H., Ohno T. and Silparcha U.
The Determination of Service Couplings for Non-disruptive Systems.
DOI: 10.5220/0004775602100214
In *Proceedings of the Third International Symposium on Business Modeling and Software Design (BMSD 2013)*, pages 210-214
ISBN: 978-989-8565-56-3

et al., 2012; Wang et al., 2009; Gui and Scott, 2006; Allen and Khoshgoftaar, 1999). Among the proposed techniques, system component coupling or dependency is one of the most widely adopted techniques (Yang et al., 2007). There are two kinds of dependencies among system components, inter-component and intra-component dependencies. The system components can be viewed at different levels. From the business point of view, a system component can be a single business process that interacts with other business processes. At the system architecture level, a system component is a service that provides specific functions for a business process. Such a service usually requires some interactions with other services. Finally, at the object-oriented implementation level, a system component is a class that works hand in hand with other classes. Therefore, the degree of coupling can be determined at different levels depending on the perspectives. In this paper, we focused at the system architecture level, where services are the major component of concern. How much a service is dependent on another service is described as the degree of coupling. The system with a higher degree of coupling, i.e. there are more dependencies among its services, is considered to be more complex than the system with lower degree of coupling.

There are a lot of researches in literature intended to measure the degree of coupling (Xu et al., 2006; Yang and Temporo, 2007; Wang et al., 2009; Gui and Scott, 2006; Allen and Khoshgoftaar, 1999; Saxena and Kumar, 2012). Our proposed concepts are applicable for any kind of couplings. However, for simplicity, we only concern with direct couplings (Yang et al., 2005). A direct coupling is a relationship, between two services. There are several kinds of such a relationship, including a method call, communication between services, etc. (Saxena and Kumar, 2012). The amount of dependency between two services can be obtained by the number of relationships between them. Therefore, the overall degree of coupling of a task or a system is the summation of the amounts of dependency of all possible pairs of services within its scope. The degree of coupling of a task is described in equation (1).

$$DC_{task} = \sum_{i,j=1}^{n} dp(i,j), i \neq j \quad (1)$$

where

DC_{task} is the degree of coupling of the task,

$dp(i,j)$ is the amount of dependency services i and j,

n is the number of services within the task.

3 SERVICE SWITCHING

For system continuity, a reliable system should be designed such that every critical service should have some "backup" or alternative service that can provide similar functionalities. In case of uncertain circumstances may occur with a critical service also called the primary service becomes unavailable, some business process will not be able to continue or have a difficulty to continue its execution. In this case, the service path will have to switch from the primary path to the alternative path. In the other words, the prepared alternative service will be used in place of the primary service. Such uncertain circumstances may be due to service unavailable such as failures, data corruption or human errors (Beecher et al., 2011). In turns, the alternative service may also be required by some other services which are possibly parts of the same task and even worse if it is required by services of some other tasks. In addition, the primary path is usually the best choice in a good system design. Being forced to execute an alternative path to maintain its function may result in an inevitable increasing complexity of the system. We define service switching into two types which are switching services within a task, and switching services across tasks. Before we discuss in greater details of such types of service switching, there are some conventions to be noted as follows.

- Primary and alternative services can provide equivalent functionalities but they are not exact the same service.
- Primary and alternative services may have different degrees of coupling associated with them.
- Any service may be served by one or more services. To complete a task a number of services may be utilized in parallel and/or sequential orders.

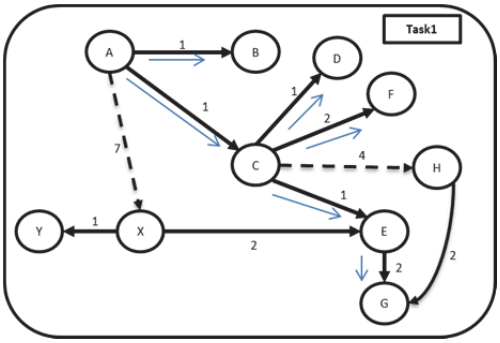

Figure 1: This primary path to complete Task1.

Table 1: The primary path to complete Task1.

Path	Degree of Coupling
A → B	1
A → C	1
C → D	1
C → E	1
C → F	2
E → G	2
Total Degree of Coupling	8

3.1 Switching Services within a Task

When a critical service needed for a task becomes unavailable, the task will switch to execute a predefined alternative service in order to complete the task. In this case, the service switching will affect only the specific task, and no others.

To demonstrate the service switching within a task, an example system is considered with an execution task, referred to as Taks1. The task has its primary path set to services A, B, C, D, E, F, and G. Services C and E are critical services with X and H being their alternative services, respectively. The overall service dependencies of Task1 are shown in Figure 1.

The overall degree of coupling of the primary path of Task1 is calculated as shown in Table 1.

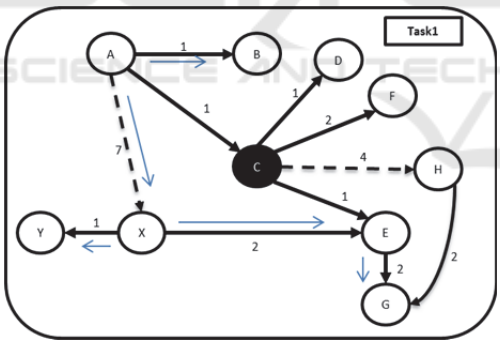

Figure 2: Task1 after service switching occurs.

Table 2: The alternative path to complete Task1.

Path	Degree of Coupling
A → B	1
A → X	7
X → E	2
X → Y	1
E → G	2
Total Degree of Coupling	13

The primary path's degree of coupling of Task1 is 1+1+1+1+2+2 = 8. We can assume for simplicity that the execution order of these services is not significant. Once service C becomes unavailable as shown in Figure 2, Service A has to execute service X in order to continue its work for Task1. By switching from service C to service X the degree of coupling has become 1+7+2+1+2 = 13 as shown in Table 2.

3.2 Switching Services across Tasks

It is quite common that different task may require same services. Switching services across tasks happens when a service component, that serves a usual task, is occupied by a different task from the usual one due to some circumstance that makes such a switching occurred. The result of such service switching can then cause a chain reaction on another task that usually utilizes the service. This phenomenal is demonstrated in Figure 3.

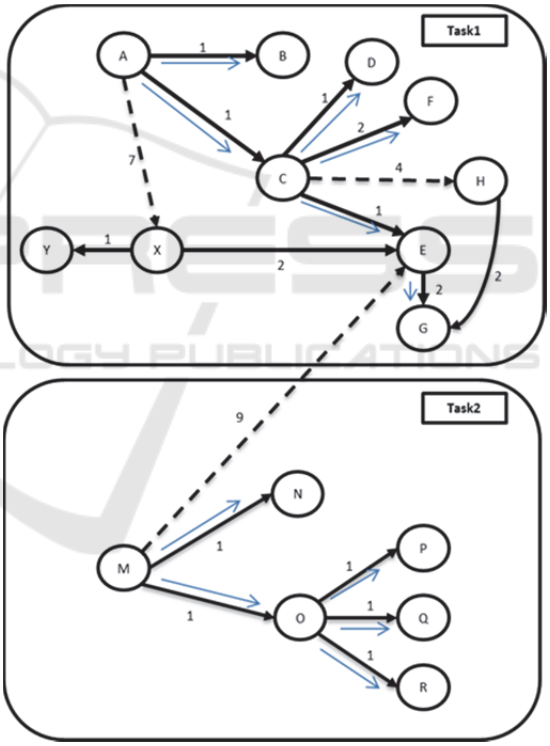

Figure 3: Service path of Task1 and Task2.

In this system, there are two separated tasks running simultaneously, i.e. Task1 and Task2. Task1 is similar to the one described in the previous discussion, such that it requires services A, B, C, D, E, F, and G with the degree of coupling 8 for its primary path. Task2 consists of services M, N, O, P, Q, and R. Task1's service E presents as an alternative of Task 2's service N. The degree of coupling of the primary path of Task2 is 1+1+1+1+1

= 5. In a normal situation, each task executes its service according to its primary path. Once, for some reason, service N becomes unavailable. In this case, the service switching occurs. Service N will be switched to its alternative which is service E, which depends on service G. Task1's degree of coupling has now increased from 8 to 13 as discussed earlier. Task2, at the same time, has increased its overall degree of coupling from 5 to 16 due to the switching across tasks. Since both Task1 and Task2 belong to the same system, the total degree of coupling of this system has increased from 8+5 = 13 to 13+16 = 29. The increased degree of coupling of a system implies that it becomes more vulnerable.

3.3 Determining the System True Degree of Coupling

In the two sections above, we discussed about the couplings within and across tasks. When a task within the system needs to switch from a primary service to the alternative services, the degree of coupling usually increases. That is because when we design software, we usually select the best path, i.e. the path that has the lowest overall degree of coupling, to be our primary path, and some other path with higher degree of coupling to be an alternative one. Therefore, once a service switching occurs, the degree of coupling usually increases.

Since the degree of coupling can be used for software quality measurement, considering only the degree of coupling of the primary path may no longer be accurate because it is not guaranteed that the primary path will always be available, i.e. a service switching occurs and the degree of coupling will change. On the other hand, considering only alternative path is equally unfair. Thus, the system designer should take account for the degrees of coupling in both primary and all alternative paths. Equations (2) - (4) suggest the proper degree of coupling of a system using the probability as the weight.

$$DC_{system} = \sum_{p=1}^{m} DC_{task_p} \quad (2)$$

$$DC_{task_p} = P(M_p)C(M_p) + C(A_p) \quad (3)$$

$$C(A_p) = \sum_{x=1}^{y} [P(alt_{px})C(alt_{px})] \quad (4)$$

,where

DC_{system} is the degree of coupling of the system,

DC_{task_p} is the direct coupling of $task_p$ within the system,

$P(M_p)$ is the probability that the primary path will be used in task p,

$C(M_p)$ is the total dependency along the primary path,

$C(A_p)$ is the total degree of coupling along the alternative path,

$P(alt_{px})$ is the probability that the alternative path x will be used in task p,

$C(alt_{px})$ is the total dependency along the alternative path x.

For a simplest case, a system consists of two separated tasks; say T_a and T_b, each of which relies on its main service path. Suppose the degrees of coupling primary paths of T_a and T_b are 8 and 11, respectively. An alternative path is provided for each main task, such that the alternative path of T_a has 14 degree of coupling, which the one for T_b has 17 degree of coupling. The probability that the main service path in T_a to become unavailable is 5%, and 10% for T_b. Then the overall degree of coupling of this system will be calculated as shown below. Then $P(M_a) = 0.95$, $P(M_b) = 0.90$, $P(A_a) = 0.05$, $P(A_b) = 0.10$, $C(M_a) = 8$, $C(M_b) = 11$, $C(A_a) = 14$, and $C(A_b) = 17$.

$$\begin{aligned} DC_a &= P(M_a)C(M_a) + P(A_a)C(A_a) \\ &= (0.95)(8) + (0.05)(14) \\ &= 8.3 \end{aligned}$$

$$\begin{aligned} DC_b &= P(M_b)C(M_b) + P(A_b)C(A_b) \\ &= (0.90)(11) + (0.1)(17) \\ &= 11.6 \end{aligned}$$

$$\begin{aligned} \text{Thus, } DC_{System} &= DC_a + DC_b \\ &= 8.3 + 11.6 \\ &= 19.9 \end{aligned}$$

From above example, we consider the degree of coupling from both primary and alternative paths.

4 CONCLUSIONS

This study proposes a technique that can be used to determine the quality of software systems through the overall degree of couplings, particularly for non-disruptive systems. The system continuity can be achieved by providing alternative service paths to critical ones, so that the system may switch the execution to the alternative services once the main services become unavailable.

It is quite natural that the degree of couplings of the alternative path is higher than the main path

because of the system design that should minimize the degree of coupling. Since switching the service paths may increase the overall degree of coupling of a system, it is essential that the system designer should look into this issue when determines the true quality of the system.

REFERENCES

Allen, E., Khoshgoftaar, T., 1999. Measuring coupling and cohesion: an information – theory approach. The 6th International Symposium on Software Metrics.

Beecher V. et al., 2011. Oracle database high availability overview 11g R2. Oracle Database Documentation Library.

Erl, T., 2008. SOA: principles of service design. Prentice Hall.

Gui, G., Scott, P., 2006. Coupling and cohesion measures for evaluation of component reusability. The 2006 International on Mining Software Repositories.

IBM Global Service, 2012. IBM index reveals key indicators of business continuity exposure and maturity. IBM Global Technology Services.

Rich, K. et al., 2012. Oracle data guard concepts and administration 11g R2. Oracle Database Document Library.

Saxena, V., Kumar, S., 2012. Impact of coupling and cohesion in object-oriented technology. Journal of Software Engineering and Application.

Wang, J. et al., 2009. Service evaluation in SOA: Towards business/IT alignment. The 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking, and Parallel/Distributed Computing.

Xu, T., et al., 2006. Service oriented dynamic decoupling metrics. The 2006 International Conference on Semantic Web and Web Service.

Yang, H. et al., 2005. Detecting indirect coupling. The 2005 Australian Software Engineering Conference.

Yang, H., Temporo, E., 2007. Measuring the strength of indirect coupling. The 2007 Australian Software Engineering Conference.