MEASURING REUSABILITY OF WEBSERVICES USING COUPLING METRICS

Dr K Venkatesh Sharma, Asst Professor, Jimma University,venkateshsharma.cse@gmail.com
Behailu Shewandagn, Asst Professor, school of computing,jimma university, bepoweru@gmail.com
Shankar Nayak Bhukya, Asst Professor, SVIT, Hyderabad, bsnaik546@gmail.com

ABSTRACT-A service-oriented architecture (SOA) is a set of principles and methodologies for designing and developing software in the form of interoperable services. These services are well defined business functionalities built as software component that can be reused. Reusability is the most demanding core principle of Service Oriented Architecture (SOA). Service reusability is the measure of ease with which one can use previously developed services in the new applications. The Existing System tells about each research and publication focuses on different qualities of service to propose a set of metrics for service reusability. Many coupling metrics have already been proposed but it is recognized that they considered for measuring coupling are insufficient. In all these the IO dependency factors, delayed dependency and various indirect dependency are not considered. The proposed system measures service coupling based on five metrics as follows: direct dependency, indirect dependency, IO dependency, delayed message dependency, state dependency .Since the metrics are going to be applied on each individual service, coupling level of the service will be more appropriate. The evaluation of correct coupling level of service will lead to better performance, reusability and efficiency of service.

Keywords- Service Oriented Architecture, Service Reusability, Coupling, UML, Class Diagram, Dependencies.

I.INTRODUCTION

Web services can implement a service-oriented architecture. Web services make functional building-blocks accessible over standard Internet protocols independent of platforms and programming languages.A Web service is a software application which is identified by a URI**,** and whose interfaces and binding are capable of being defined, described and discovered by XML artefacts and supports direct interactions with other software applications using XML based messages via internet-based protocols.Characteristics Of Web Services are XML-based, Loosely coupled, Coarse-grained**.** Loose coupling induces reusability of services.

A.Characteristics of service reusability

For making the service as a reusable one, the emphasis should be given to both design and quality characteristics (Fig.-1).

At the design stage the service should be adhere other design characteristics which enhance service reusability.

B. Design Characteristics of Service Reusability

- **Loose Coupling:** Service loose coupling enhances the service reusability. The lower the dependency with other services, the more easily it can be reused.
- **Composability:** Service composability is the key principle for reusability. The composable service can easily aggregate with other services. Therefore the service composability offers higher degree of service reusability.
- **Abstraction:** Service abstraction hides the unnecessary information from the service consumers. Also it reduces the needless coupling between the service consumer and service provider thereby increases the service reusability.
- **Statelessness:** Statelessness encourages service reusability. Lesser the amount of state management responsibilities increases its scalability and availability which are the required qualities to enhance service reusability.
- **Discoverability:** Service discoverability promotes service reusability. If and only if the service consumer can easily find the required service, the service can be more reusable.
- **Granularity:** The service granularity may be fine grained or coarse grained. Depending upon the type of service, the granularity level may vary. The correct granularity level of the service enhances the service re-use.

C.Coupling:

A principle characteristics of a service is Loose Coupling .Coupling means that the existent and the extent of dependency between modules, components, or services. If dependencies are known and few then the coupling is said to be loose coupling. In contrast, the dependencies are more and unknown for tight coupling. If dependency, exist between services, then the change in one service may cause changes in dependent services which affect the flexibility of the system.

Coupling is the extent to which the various subcomponents interact. If they highly interdependent then changes to one are likely to have significant effects on the behavior of others [4]. Hence loose coupling between its subcomponents is desirable characteristics of a component.

II.RELATED WORK

Reusability is the key paradigm for increasing software quality in the software development. Reusability was the core concept of object oriented systems which there on evolved into component based systems and service oriented systems. Various metrics like Modularity, Mismatch, Service Granularity, Discoverability, Service Cohesion and Service Coupling were identified [6].

A Study and Critical Survey on Service Reusability Metrics provides service design attributes for reusability such as functional richness, easy understandability, good exception handling and portability, independent of other services. The design characteristics of service reusability are loose coupling, comparability, autonomy, abstraction, statelessness, discoverability, granularity, etc[7].

The coupling is an important aspect in the evaluation of reusability of components or services. In this paper we propose a suite of metrics for measuring the degree of coupling in services at the design phase and show how reusability can be evaluated from those coupling metrics. The dependencies between services cause coupling. Many coupling metrics have already been proposed but it is recognized that the dependencies they considered for measuring coupling are insufficient. This paper takes five kinds of dependencies viz. direct dependency, indirect dependency, IO dependency, and State dependency and Delayed Message dependency to evaluate the degree of coupling in a service[5].

III.PROPOSED SYSTEM

The Proposed system defines the coupling dependencies of the given service. The service is represented based on the standardized modeling languages UML.

First stage of the system is conversion of UML notation of the given service to meta data. The meta data is the skeleton code of the UML notation of the given service. The meta data is the code which contains the member variables and member functions with its return type of the corresponding class of the given service

Second stage of the system is analysis of service from meta data. From the meta data the following data are grouped. Total number of services in the system, number of services calls from the given services to others and vice versa, number of transitive connection between the services, state of the web service, number and type of input parameters and return values, total number of asynchronous messages, total number of messages. Those data is given as input for next stage for evaluation of coupling.

Third stage of the system is evaluation of coupling level from the analysis of the service from Meta data. For evaluating the coupling level of the service the following parameters have been focused

- direct dependency
- indirect dependency
- IO dependency
- delayed message dependency
- state dependency.

From the above five metrics the coupling level of service have been evaluated. Since the metrics are going to be applied on each individual service, coupling level of the service will be more appropriate. By using the Meta data information the coupling of the service can be quantified.

Fourth stage of the system is design comparison and report generation. In this stage the report is generated for specifying the coupling level of the given service.

1) Direct Dependency

The direct dependency is direct connection between services. This kind of dependency may exist between services directly when a service calls other services or a service is called by other services. Therefore the direct dependency measure may be the summation of dependencies on other services and dependencies of other services on this service and can be measured as,

$$\boldsymbol{DD(S)} = \frac{\mathbf{1}}{\boldsymbol{n(n-1)}}\left[\sum_{\boldsymbol{i=j=1,i\neq j}}^{\boldsymbol{n}} \boldsymbol{Sij} + \sum_{\boldsymbol{i=j=1,i\neq j}}^{\boldsymbol{n}} \boldsymbol{Sji}\right]$$

Where,

n is the total number of services in the system an Si ,j=1 when there is a connection between the service Si and Sj, otherwise 0.

Sj i=1 when there is a connection between the services Si and Sj , otherwise 0.

DD(S) value lies between 0 and 1, if the value is 0 there is low direct dependency. If the value is 1, the direct dependency is high .

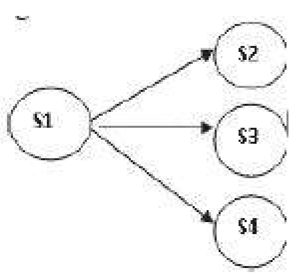

Fig 3.1 Dependencies on other services

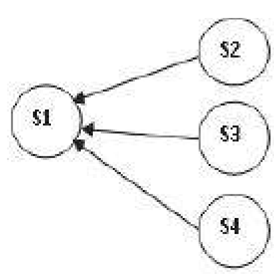

Fig 3.2 Dependencies of other services on this service

2) Indirect Dependency:

The indirect dependency between services may occur at:

(a) When there is indirect connection or transitive connection between services and

(b) When there is a sharing of global data between services.

(a) When service A depends on service B which in turn depends on service C, transitively there is indirect dependence between service A and service C (Fig.-3). This kind of indirect dependency can be measured as

$$IDD(S) = \frac{1}{n(n-2)} \sum_{i=1}^{n} Si$$

Si = 1 whenSi is indirectly connected with service S, otherwise 0.

Where,n is the total number of services.

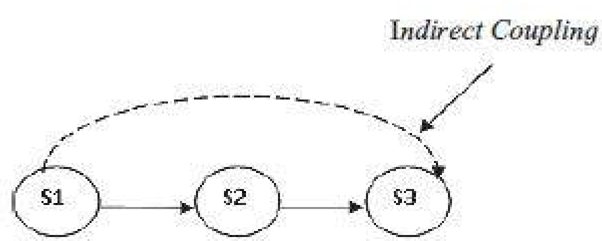

Fig 3.3 Indirect Coupling

(b) Indirect dependence may also occur when there is a global data sharing between the services. When global data in database is shared by many services, there exist coupling between the services. Therefore the indirect dependency due to shared database can be measured as,

$$IDDSD(S) = \frac{1}{(n \times m)} \sum_{i=1}^{n} TSi$$

Where, TSiis total number of services shares data in ith repository,

n is the total number of services and m is total number of repositories.

Hence the Indirect Dependency (IDD) can be measured using both equations we get,

IDD (S) = ½ [IDDT(S) + IDDSD(S)]

Where, the value of IDD lies between 0 and 1. Lesser the value, low indirect dependency, higher the value, there is high in direct dependency.

3) State Dependency

The service should be as stateless as possible. If any services maintain its state, then the coupling arises. Therefore while measuring the coupling we have to measure the state of service. The CRUD operations make the service to maintain its state. The state dependency of service can be measured as,

$$SD(S) = \frac{1}{nf} \sum_{i=1}^{nf} fi$$

Where,

fi is 1 when the function calls in S which create or read or update or delete a database otherwise 0.

Nfis total number of function calls in S and SD(S) is the coupling due to service statefulness where value lies between 0 and 1.

Lower the value the service is stateless and state dependency is less and higher the value the service is stateful and state dependency is more.

4) IO Dependency

Service coupling also depends upon both the number and type of input as well as output data items i.e., the number of input parameters and output data of the service. In order to avoid unnecessary dependencies on the input format, a service should accept required input data items to perform the service being requested. In the same way a service should return a data that is related to its internal operation. Exposing such data as part of the response data will create unnecessary dependencies on the internal implementation of the service.

For measuring the coupling due to this cause, we considered the type of input and output data items. For each type of data item, the coupling will be considered: void as very low, primitive as low, built-in or user defined as medium and complex data type as high. For quantifying the coupling, we can assign dependency weight value for each category of data items as given in the Table-5.1.

Data item type	Dependency weight value
Void	0.0
Primitive	0.33
Built-in	0.66
Complex	1.00

Table-5.1 -Dependency Weight Value

After finding the dependency weight value for each data item, the coupling due to I/O dependency can be measured as,

$$\boldsymbol{IOD(S) = \frac{1}{nd}\sum_{i=1}^{nd} Wi}$$

Where ,

*Wi*is the weight value of each input and output data items of the service,

*nd*is the total number of input and output data items of the service and

IOD(S) is the coupling due to IO dependency where value lies between 0 and 1.

Lower the dependency value lower the IO dependency and higher the value, higher io dependency and coupling. For the given input and output data items of the service, with the help of above tables we can measure the coupling.

5) Delayed Dependency

It is not necessary that all service requests need a immediate response. Often the response may be delayed. This delayed message returning may cause unnecessary dependency between called and calling services till there is a response. These synchronous messages may be used to reduce this kind of dependencies. The dependency due to delayed message can be measured as,

$$DMsgD(S) = \frac{NMsyn}{TNM}$$

Where,

*NMsyn*is total number of synchronous messages and

TNM is total number of messages in S. *DMsgD(S*) value lies between 0 and 1.

Lower the value service is loosely coupled.

IV.CONCLUSION

The Existing System focuses on different qualities of service to propose a set of metrics for service reusability. Many coupling metrics have already been proposed but it is recognized that they are insufficient. In all these the IO dependency factors, delayed dependency and various indirect dependency are not considered.

Thus the proposed system measures service coupling based on the metrics that have not been considered. The five metrics taken into consideration are direct dependency, indirect dependency, IO dependency, delayed message dependency, state dependency. Since the metrics are going to be applied on each individual service, coupling level of the service will be more appropriate. The evaluation of correct coupling level of service will lead to better performance, reusability and efficiency of service.

Other properties which affect service reusability are cohesion and granularity. The future enhancement is to include cohesion and granularity to evaluate reusability of services.

REFERENCES

[1] Peter Henderson and JingtaoYang(2004) ,"Reusable Web Services",Springer (ICSR 2004).

[2] SaadAlahmari , Ed Zaluska(2009),"Optimal Granularity for Service-Oriented Systems",The third Saudi International Conference (SIC09).

[3] George Feuerlicht and JosipLozina(2007), "Understanding Service Reusability",International Conference.

[4] GuiGui and Paul D. Scott(2008), "New Coupling and Cohesion Metrics for Evaluation of Software Component Reusability", IEEE

International Conference(ICYCS 2008).

[5] T.Karthikeyan, J.Geetha(2012),”A Metrics Suite and Fuzzy Model for Measuring Coupling in Service Oriented Architecture”IEEE International Conference.

[6] G.Shanmugasundaram, V. PrasannaVenkatesan and C. Punitha Devi(2011), ” Reusability metrics – An evolution based study on Object Oriented System, Component based System and Service Oriented System”,Journal of computing, Volume 3,Issue 9.

[7] T.Karthikeyan, J.Geetha(2012), ”A Study and Critical Survey on Service Reusability Metrics”International Journal of Information Technology & Computer Science, Vol. 4, No.5.