A Quantitative Measurement and Validation of Granularity in Service Oriented Architecture

Dr. T. Karthikeyan[1] and J. Geetha[2]

[1] **Department of Computer Science, P.S.G. College of Arts and Science**
Coimbatore, Tamil Nadu, India

[2] **Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College**
Pondicherry, India

Abstract

The prominent principle of designing service in Service Oriented Architecture is service granularity. The granularity is a crucial design decision. The service should be neither too coarse nor too fine grained. Therefore it should be evaluated at the design phase itself to measure its level of appropriateness. This paper proposes a suite of metrics for measuring service granularity quantitatively. Although many metrics have already been defined to measure service granularity, their attention in measuring service interface granularity is very less which affects the other design principles coupling and reusability. This paper proposes a metric for measuring granularity of a service by considering its composite level, functional richness and its interface granularity. The paper also validates the proposed metrics theoretically and a case study is performed to analyze the proposed metrics.

Keywords: *Service Oriented Architecture, Granularity Metrics, Granularity, Validation of Granularity Metrics*

1. Introduction

Service Oriented Architecture (SOA) is a modern architectural style in the context of Information Technology. In SOA, services are used as building blocks to organize and architect the applications. The approaches and ways are different to build these services for different applications. There is no simple solution for designing and implementing service oriented architecture. However the service should be adhered design characteristics and principles of SOA while it is designed. One of the prominent principle of designing service is the matter of how abstract services should be i.e., service granularity. Proper granularity of a service leads to increase of service's reusability and performance which are the main goal of service oriented architecture. The "granularity" is the communication level associated with some aspect of program design. In the context of service design, the granularity of the service is the primary concern.

Naveen Kulkarni et al [1] identified some key issues arising out of improper service granularity. If the granularity is not properly designed, there is a chance of different services for similar tasks. This redundancy imperils business and technology alignment within enterprises. It is extremely difficult in service governance and maintenance when the service is either too large as too coarse grained or number of services are more as more number of fine grained services.

In order to understand the granularity of a service, first we should understand how a service has been defined and what a service is made of. A service consists of a contract, an implementation, and an interface Fig. (1)[2]. The contract contains the informal specification of the service, i.e. its purpose, functionality, constraints and usage. One element of this is a formal interface definition in for example WSDL (web services description language) or IDL (interface description language). The description of the interface is specified in the service contract. The implementation of the service physically provides the required business logic and appropriate data [2].

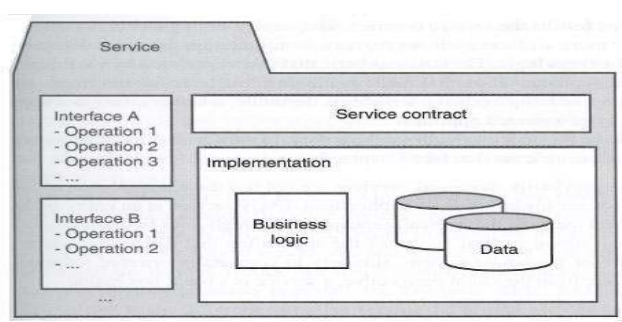


Figure-1 Service

All these three elements influence the service granularity. The description of the interface gives clear guidance in how the service is implemented and reduces complexity

and enhancing reusability in the further implementation. Therefore, it adds to granularity [2].
The rest of the paper is organized as follows. In section 2, we survey and analyze previously proposed metrics and research work related to granularity of services. Section 3 discuss about service granularity. In section 4 we propose a suite of metrics for evaluating granularity of services. The proposed metrics are theoretically validated using Weyuker's properties in section 5. In section 6 a case study is performed to analyze the proposed metrics. Finally in section 7 concludes the paper.

2. Related Work

Steghusis [3] proposes granularity framework and four service granularity patterns include: flexibility, reusability, generality and performance which are mapped to the framework. Each pattern accompanies some guide lines for choosing suitable level of service granularity. Even though, the work provides proper insight into service granularity trade-offs, but it limits to high-level guidelines which are both general and qualitative. Renuka Sindhgatta et al [4] propose a metric for measuring granularity of a service. They have taken the number of services involved and the number of operations involved in the service. Wang Xiao-jun [5] proposes for evaluating service granularity in SOA. In this paper common use of service operation which constitutes a service is taken for measurement. Bingu Shim et al [6] proposed SOA design metrics. In that they proposed a metric for service granularity which considers the type of the operations, i.e., it is synchronous or asynchronous as well as the parameter of the operations. In [7], a capability granularity analysis framework is presented which can find a trace of fine grained capabilities of web services operations by using association rules and apriori algorithm to analyze client invocation behavior. In [8], service granularity can be quantified as a combination of the number of components/services composed through a given operations on a service interface as well as the number of database tables. However this approach neither covers important granularity attributes nor recommends any criteria to evaluate appropriateness of service granularity.

3. Service Granularity

Service granularity generally refers to the size and functionality of a service. It means that the amount of business function performed in a single request/response exchange of messages. Erl [9] classifies the granularity as capability granularity and data granularity. Capability granularity refers to the functional scope of the service and data granularity refers to the amount of data transferred to provide the functionality. The service can be either coarse grained or fine grained. The characteristics of these granularities are [10]:

- Fine grained services are simple business logic, transact small volume of data, a large coupling among services and may be atomic service.
- Coarse grained services are complex business logic, transact large volume of data, a small coupling among services and may be composite.

But the fundamental principle of Service Oriented Architecture states that the service should be large sized or coarse grained service, which are invoked less frequently and exchange large amount of data.

All fine grained services are not atomic and coarse grained services are not composite in all the cases. Atomicity is a measure of process decomposition. Since granularity can be measured with respect to business context, it is possible to have coarse grained atomic services and fine grained composite services [11]. For example single database query may return large volume of data which is coarse grained atomic service. Sometimes more functional service may provide few information i.e. it may be a small part of large process which is fine grained composite service. If the composition of the service increases, the level of coupling and complexity will be increased. But the service should be loosely coupled. Therefore the composite level alone not matters for measuring service granularity.

The service should neither be too coarse nor be too fine grained. If it is too coarse it may exchange unnecessary data when it is accessed. If it is too fine grained, it may tend to have too narrow use; thereby it reduces its reusability [11]. Service granularity cannot be too large or too small but should be the right size [12][13]. A good SOA architecture design must maintain a balance in service granularity to obtain benefits of lower costs and flexibility responds [14][17][18]. The granularity is crucial design decision. If it is incorrectly predicted, consumers will have access to more functionality than they need. This can be a problem for security at the service level [15].

The concept of granularity is a relative measure of how broad the interaction between a service consumer and service provider to satisfy the need of service consumer [11]. The interaction style describes how information is passed into and out of the service. Since the type, size, functionality of the services are varying, the interaction style is also varying. In the interface the size of input and output may also be varied from small to large i.e. from single value to large size document. So when we measuring granularity of the service we have to also consider granularity of the interface.

The input to the service which is passed via the service interface reflects how much data is passed on to that service by a service consumer. If the input data is more, the granularity of the service is coarse grained [16]. If it is few or no input parameter is then the granularity of the service is fine grained. The output of the service indicates how much data is returned to the service consumer. If the output is one or more business objects or documents then the service is said to be coarse grained whereas fine grained service returns few or small data values to the service consumers [16].

By the above discussions, we conclude that the granularity of a service can be evaluated by quantifying the composite level, functional richness and interface of the service.

4. Proposed Metric

The services may be broadly classified as Business services, Information or Domain services and Utility or Task services according to the type and scope of the function they performed. Different service types can have different granularities.

Business services primarily manage and access to business entities. Business entities are customers, policies, claims and so on. They correspond to major business information concepts. This type of services and entities are usually large in size. Information or Domain services execute business rules to provide business decisions or information. It provides simple to complex information to support frequently changing business rules. Usually these services are composed into other services and are small to medium in size. Utility or task services are more general in purpose which does a small task to single function. They are small in size.

The optimal level of granularity cannot be a single measure. Different attributes have to be considered. The functionality, atomicity, interface input granularity and interface output granularity of a service should be considered when we measure the optimal level of granularity for different types of services. In this section we present a suite of metrics for measuring granularity of a service.

Let us consider S is a service which is composed of set of services $\{S_1, S_2, S_3 \ldots Sn\}$ and are invoked on a given operation. N is the total number of services in S. O is the set of operations $\{O_1, O_2, O_3, \ldots O_n\}$ of each service Si.

The composite service may be coarse grained service. A service is composed of many atomic and composite services. As the number of composite service increases, the granularity may also increase. Therefore we have to consider the level of composite services and it can be measured as,

$$CL(S) = 1 - \frac{NAS}{N}$$

Where, NAS is number of atomic services in S and N is the total number of services. The composite level value lies between 0 and 1. Lower the value lesser the composite level and granularity, higher the value higher the composite level and granularity.

As already discussed not all atomic services are fine grained and composite services are coarse grained. The functionality is also speaks the granularity of the service. The functions of a service include both business logic function and CRUD function. Therefore we have to consider both in measuring the functional richness of the service. We know that the functionality can be measured by measuring the function point. From this we built a metric for the functional richness of a service. The Functional Richness of service (*FR(S)*) can be measured as,

$$FR(S) = \left[\left(0.001 * \sum_{i=1}^{N} FC_i \right) + \left(\frac{\sum_{i=1}^{N} CRUD(O_i)}{\sum_{i=1}^{N} O_n(S_i)} \right) \right]$$

Where, N is total number of services in S,

FC_i is function point count of each invoked service on the given operation,

CRUD (O_i) is 1 if it is the CRUD operation in service S_i,

$O_n(S_i)$ is the total number of operations in each service S_i.

The value lies between 0 and ≥1. Closer to and above the value 1, the service is functionally rich.

The granularity of the service is exposed through its interface. The evaluation of granularity will be completed only if measuring the granularity of the service interface. The interface signature of the service may contain different input and output varying from simple data/ data set to complex data or documents. By classifying the input and output, we can assign weight values to each category. Based on the assigned values in the table (Table-1) granularity of the interface can be measured.

Table – 1 Weight Values

Input / Output data	Weight Value
Void	0.0
Primitive	0.25
User defined	0.5
Complex	1.0

The Interface granularity (IG(*S*)) can be measured as,

$$IG(S) = \left[\frac{\sum_{i=1}^{nip} W_i}{nip} + \frac{\sum_{i=1}^{nop} W_i}{nop} \right]$$

Where nip is number of input parameters,
nop is number of output parameters,
W_i is weight value of parameters.

Interface granularity of service value lies between 0 and 1. Lesser the value the granularity in the interface is less. More the value, the granularity of interface will be more.

The granularity of service S depends on its composite level, functional richness and its granularity of the interface and the granularity of service *G(S)* can be measured as,

$$G(S) = 0.33 * [CL(S) + FR(S) + IG(S)]$$

The value lies between 0 and ≥ 1. Lesser the value means the granularity level is low. Higher the value the granularity level of given service is more. The optimal level of granularity varies in different types of services (Table-2).

Table -2 Optimal Level

Type of Services	Optimal level of Granularity
Business Service	>= 0.7
Domain Service	0.31 to 0.69
Task Service	0 to 0.30

For Business service, the granularity should be high. Because it is composed of many services, should functionally rich as well as interface granularity should also be very high. Even the granularity level is medium; it is optimal for domain and small for task services because of its sizes and functionality.

5. Theoretical Validation of Proposed Metric using Weyuker's properties

Weyuker (1988) proposed an axiomatic framework for evaluating software metric. The proposed metrics can be evaluated against these properties. The properties and evaluations are:

Property 1: There are programs P and Q for which M (P) ≠ M (Q).
There may be two different services with different granularities satisfying the first property.

Property 2: If c is non-negative number, then there are finitely many programs P for which M (P) =c
As service will have at least one interface structure and business method with some functionality, therefore its granularity will always have some positive value. It confirms the second property.

Property 3: There are distinct programs P and Q for which M (P) =M (Q)
For two different services with different functionality and interface structure, granularity metric value may be same. Property 3 is satisfied on the proposed metric.

Property 4: There are functionally equivalent programs P and Q for which M (P) ≠M (Q)
Even if the functionality of the two services is same, both may have different granularities as these services may be designed by using different technologies and programming concepts. It confirms 4th property.

Property 5: For any program bodies P and Q, we have M (P) <= M (P; Q) and M (Q) <=M (P; Q).
If a service is composed of another to get an assembly for enhanced functionality, the granularity of these two individual services will be lesser than the granularity of the assembly, which satisfies 5th Weyuker property.

Property 6: There exist program bodies P, Q and R such that M (P) =M (Q) and M (P; R) ≠M (Q; R).
Two services with the same granularity means both will have same no. of interface methods with same arguments and return types and same functionality. However, they may be developed by using different programming methodologies and therefore when integrating in the service, both may have different integration code and implementation thus resulting in different granularities in both the cases. Therefore 6th property is also satisfied.

Property 7: There are program bodies P and Q such that Q is formed by permuting the order of statements of P and M(P) ≠M(Q). The ordering of interface methods and return type and arguments of a service will not change the granularity of the service. Thus this property is not satisfied.

Property 8: If P is a renaming of Q, then M (P) = M (Q).
The renaming of a method or a service will not affect the granularity of that interface or the service thus satisfying this property.

Property 9: There exist program bodies P and Q such that M (P) +M (Q) < M (P; Q).
When two services are composed, we may have to write some more methods related with the integration also. This

will increase the granularity of the composed service. This satisfies the last property.
Out of nine properties, eight properties are satisfied by the proposed metrics.

6. A Case Study

A granularity is the most important characteristics of a service. The more service granularity will increase the flexibility and reusability of the service. If we find any flaw in granularity, we can select the alternative design at the early stage itself. This can be done by quantitative assessment method over the design structure. This paper proposed a suite of metrics for measuring granularity. To show the applicability of the proposed metrics, three different versions of product catalog service design structure has been taken. The values of composite level, functional richness, interface granularity and the service granularity of these services are calculated and tabulated in Table-3.

Table-3 Metric Values

Version / Metric	V1	V2	V3
CL	0.73	0.73	0.73
FR	0.57	0.74	0.61
IG	0.41	0.43	0.46
GL(S)	0.56	0.63	0.59

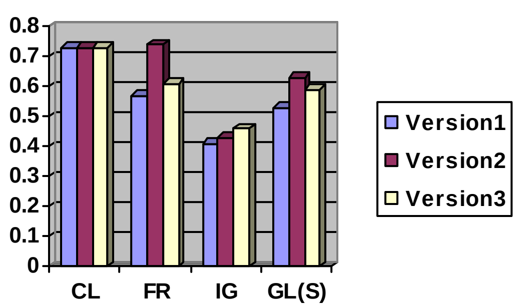


Figure - 2

The granularity of the version V2 is high and we can compare the primitive values such as composite level, functional richness and interface granularity of this service can be compared with the values of other versions (Fig.-2). Lower these values, the granularity is low. Though the CL value is same for the three versions, the other values are high for V2 comparing with other versions thereby the granularity is also high.

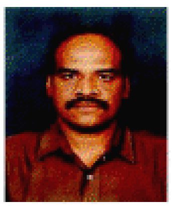
7. Conclusions

Evaluating service oriented architecture needs a suite of metrics. Our aim is to define metrics for service and thereby proposing a standard for service oriented architecture [19]. This paper proposed a suite of metrics for evaluating granularity of a service which is an important characteristic which affects the flexibility and reusability of the service. Our future work is to proposing metrics for other characteristics of the service.

8. References

[1] Kulkarni N., Dwivedi, "The Role of Service granularity in a successful SOA Realization A Case Study", IEEE Congress on Services – Part-I, vol.no. pp 423-430, 6-11, 2008.
[2]Krafzing D., Banke K., and Salma D. "Enterprise SOA, Service Oriented Architecture Best Practices", Prentice Hall, New Jersey, 2005.
[3] Steghuis C., "Service Granularity in SOA Projects: A trade off Analysis", M.Sc. Thesis, Business Information Technology, University of Twente, 2006.
[4] Renuka Sendhgatta, Bikaram Sengupta, and Karthikeyan Ponnalagu, "Measuring the Quality of Service Oriented Design", ICSOC – Service Wave 2009, LNCS 5900 PP-485-499, 2009.
[5] Wnag Xiao-jun, "Metrics for Evaluating Coupling and Service Granularity in Service Oriented Architecture", 978-1-4244-4994-1/09, 2009, IEEE.
[6]Binghushim, S. Choe, S.Kim, S.Park. "A Design quality model for Service Oriented Architecture", Proc. 2008 15th Asia Pacific Software Engineering Conference, IEEE, 2008, pp. 403-410.
[7]T. Senivongse, N. Phacharintanakul, C. Ngamnitiporn, M. Tangtrongchit. "A Capability Granularity Analysis on Web services Invocations". Proceedings of the world congress on Engineering and computer science 2010 Vol. I WCECS 2010, October 20-22, 2010, San Francisco, USA.
[8] A. Khoshbarforoushha, R. Tabein, P. Jamshidi, F. Shams, "Towards a Metrics Suite for Measuring Composite Service Granularity Level Appropriateness", IEEE 6th World Congress on Services", 2010.
[9] Thomas Erl, "SOA principles of Service Design", Prentice Hall, 2009.
[10] Xie Zhengyu, Dong Baotian, and Wang Li, "Research of Service Granularity Base on SOA in Railway Information Sharing Platform", Information Symposium on Information Processing, 2009.
[11] Jason Boomberg "How to Define a Business Service the Art and Science of Service Granularity", Zap Think White paper, 2007 available at www.zapthink.com.
[12] James Mc Govern, Sameer Tyagi, Michael E Stevens and Sunil Mathew, "Java Web Services Architecture", Morgan Kaufmann Publishers, An imprint of Elsevier, 2005.
[13] Hong YING, Yu WU,Fuming LIU, "Research on The SOA – based Service Granularity Control", Second International Conference on Information Technology and Computer Science, IEEE, 2010.

[14] Pierre Reldin and Peter Sundling "Explaining SOA Service Granularity – How IT-Strategy shapes services", Master Thesis, Linkoping University, 2007.
[15]Spyridon Antakis, "Security Service Granularity", 2008, available at www.*spyros.loonydesk.com/whitepapers/ssg.pdf*
[16] Michael Rosen, Boris Lublinsky, Kevi T. Smith and Marc J. Balcer, "Applied SOA", Wiley India Edition, 2008.
[17]]Qian Ma, Nianjun Zhou, Yanfeng Zhu, Hao Wang, "Evaluating Service Identification with Design Metrics on Business Process Decomposition", IEEE International Conferences on Services Computing, 2009.
[18] Alahmari S., and Zaluska E., "Optimal Granularity for Service Oriented Systems", The 3rd Saudi International Conference (SIC09), 5-6 June, 2009, University of Surrey, Guildford, U.K.
[19]Dr. T. Karthikeyan, J.Geetha, "A Software standard and Metric based Framework for Evaluating Service oriented Architecture", International Journal of Engineering Research and Applications. Vol.-1, issue 4, pp. 1740-1743.



Prof. Thirunavukarasu Karthikeyan received his graduate degree in Mathematics from Madras University in 1982. Post graduate degree in Applied Mathematics from Bharathidasan University in 1984. Received Ph.D. in Computer Science from Bharathiyar University in 2009. Presently he is working as an Associate Professor in Computer Science Department of P.S.G. College of Arts and Science, Coimbatore. His research interests are Image Coding, Medical Image Processing, Data Mining and Software Engineering. He has published many papers in national and international conferences and journals. He has completed many funded projects with excellent comments. He has contributed as a program committee member for a number of international conferences. He is the review board member of various reputed journals. He is board of studies member for various autonomous institutions and universities.

Mrs. J. Geetha received her Post graduate degree in Computer Applications from Madras University in 1996. Received Master of Technology Degree in IT from Allahabad Agricultural Institute Deemed University in 2005. She is now doing her research work in Bharathiyar University in the area of software metrics. Her research areas of interests are Software metrics and Data mining. Presently she is working as an Assistant Professor in the department of Computer Science and Engineering of Sri Manakula Vinayagar Engineering College, Pondicherry. She has published many papers in national and international conferences.