



FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Designing a business platform using microservices.

Rajendra Kharbuja





FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Designing a business platform using microservices.

Entwerfen einer Business-Plattform mit microservices.

Author:	Rajendra Kharbuja
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Manoj Mahabaleshwar
Submission Date:	



I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich,

Rajendra Kharbuja

Acknowledgments

Abstract

Contents

Acknowledgments	iii
Abstract	iv
1 Quality of Service	1
1.1 Introduction	1
1.2 Quality Attributes	1
1.3 Quality Metrics	2
1.3.1 Context and Notations	3
1.3.2 Coupling Metrics	4
1.3.3 Cohesion Metrics	5
1.3.4 Granularity Metrics	7
1.3.5 Complexity Metrics	8
1.3.6 Autonomy Metrics	9
1.3.7 Reusability Metrics	10
Acronyms	11
List of Figures	12
List of Tables	13
Bibliography	14

1 Quality of Service

1.1 Introduction

In addition to align with the business requirements, an important goal of software engineering is to provide high quality. The quality assessment in the context of service oriented product becomes more crucial as the complexity of the system is getting higher by time. [ZL09; GL11; Nem+14] Quality models have been devised over time to evaluate quality of a software. A quality model is defined by quality attributes and quality metrics. Quality attributes are the expected properties of software artifacts defined by the chosen quality model. And, quality metrics give the techniques to measure the quality attributes. [Man+np] The software quality attributes can again be categorized into two types: internal and external attributes. [Man+np; BMB96] The internal quality attributes are the design time attributes which should be fulfilled during the design of the software. Some of the external quality attributes are loose coupling, cohesion, granularity, autonomy etc.[Ros+11; SSPnp; EM14] On the other hand, the external quality attributes are the traits of the software artifacts produced at the end of Software Development Life Cycle. Some of them are reusability, maintainability etc. [EM14; Man+np; FL07; Feu13] For that reason, the external quality attributes can only be measure after the end of development. However, it has been evident that internal quality attributes have huge impact upon the value of external quality attributes and thus can be used to predict them. [HS90; Bri+np; AL03; Shi+08]The evaluation of both internal and external quality attributes are valuable in order to produce high quality software.[Man+np; PRF07; Per+07]

1.2 Quality Attributes

As already mentioned in section 1.1, the internal and external qualities determine the overall value of the service composed. There are different researches and studies which have been performed to find the features affecting the service qualities. Based on the published research papers, a comprehensive table 1.1 has been created. The table provides a minimum list of quality attributes which have been considered in various research papers.

#	Attribute	[SSPnp]	[Xianp]	[AZR11]	[Shi+08]	[Ma+09]	[FL07]
1	Coupling	✓	✓	✓	✓	✓	✓
2	Cohesion	✓	X	✓	✓	✓	✓
3	Autonomy	✓	X	X	X	X	X
4	Granularity	✓	✓	✓	✓	✓	✓
5	Reusability	✓	X	X	✓	X	✓
6	Abstraction	✓	X	X	X	X	X
7	Complexity	X	X	X	✓	X	X

Table 1.1: Quality Attributes

The table 1.1 gives a picture of the studies done so far around service quality attributes. It can be deduced that most of the papers focus on coupling and granularity of the service and only few of them focus on other attributes such as complexity and autonomy.

Coupling refers to the dependency and interaction among services. The interaction becomes inevitable when a service requires a functionality provided by another service in order to accomplish its own goals. Similarly, Cohesion of any system is the extent of appropriate arrangement of elements to perform the functionalities. It affects the degree of understandability and sensibility of its interface for its consumers. Whereas, the complexity attribute provides the way to evaluate the difficulty in understanding and reasoning the context of the service or component.[EM14]

The reusability attribute for a service measures the degree to which it can be used by multiple consumer services and can undergo multiple composition to accomplish various high order functionalities. [FL07]

Autonomy is a broad term which refers to the ability of a service for self-containment, self-controlling, self-governance. Autonomy defines the boundary and context of the service. [MLS07] Finally, granularity is described in chapter ?? in detail. The basic significance of granularity is the diversity and size of the functionalities offered by the service. [EM14]

1.3 Quality Metrics

There have been many studies made to define metrics for quality components. A major portion of the research have been done for object oriented and component based development. These metrics are refined to be used in service oriented systems.[Xianp; SSPnp] Firstly, various papers related to quality metrics of service oriented systems are collected. The papers which conducted their findings or proposition based on quality

amount of scientific studies and have produced convincing result are only selected. Additionally, the evaluation presented in the research papers were only done using the case studies of their own and not real life scenarios. Furthermore, they have used diversity of equations to define the quality metrics. On the basis of case studies made in the papers only, a fair comparison and choice of only one way to evaluate the quality metrics cannot be made. This is further added by the fact that the few papers cover most of the quality metrics, most of them focus on only few quality metrics such as coupling but not all of them focus on all the quality metrics. Nevertheless, they do not discuss about the relationship between all quality metrics. [EM14] This creates difficulty to map all the quality metrics into a common calculation family.

With such situation, this section will focus on facilitating the understanding of the metrics. Despite the case that there is no idea of threshold value of metrics discussed to define the optimal quality level, the discussed method can still be used to evaluate the quality along various metrics and compare the various design artifacts. This will definitely help to choose the optimum design based on the criteria. Also, there is complexity in understanding and confusion to follow a single proposed metrics procedure. But, the existing procedures can be broken down further to the simple understandable terms. By doing so, the basic terms which are the driving factors for the quality attributes and at the same time followed by most of the procedures as defined in the papers, can be identified.

The remaining part of this section will attempt to collaborate the metrics definitions proposed in various papers, analyze them to identify their conceptual base of measurement in simplest form.

1.3.1 Context and Notations

Before looking into the definitions of metrics, it will help understanding, to know related terms, assumptions and their respective notations.

- the business domain is realized with various processes defined as $P = \{p_1, p_2 \dots p_s\}$
- a set of services realizing the application is defined as $S = \{s_1, s_2 \dots s_s\}$; s_s is the total number of services in the application
- for any service $s \in S$, the set of operations provided by s is given by $O(s) = \{o_1, o_2, \dots o_o\}$ and $|O(s)| = O$
- if an operation $o \in O(s)$ has a set of input and output messages given by $M(o)$. Then the set of messages of all operations of the service is given by $M(s) = \bigcup_{o \in O(s)} M(o)$

- the consumers of the service $s \in S$ is given by $S_{consumer}(s) = \{S_{c1}, S_2, \dots, S_{n_c}\}$; n_c gives the number of consumer services
- the producers for the service or the number of services to which the given service $s \in S$ is dependent upon is given by $S_{producer}(s) = \{S_{c1}, S_2, \dots, S_{n_p}\}$; n_p gives the number of producer services

1.3.2 Coupling Metrics

[SSPnp] defines following metrics to determine coupling

$$SOI(s) = |\{o_i \in S_i : \exists o \in s, calls(o, o_i) \wedge s \neq s_i\}|$$

$$ISCI(s) = |\{s_i : \exists o \in s, \exists o_i \in s_i, calls(o, o_i) \wedge s \neq s_i\}|$$

$$SMCI(s) = |\cup M(o_i) : (o_i \in S_i) \vee (\exists o \in s, \exists o_i \in s_i, calls(o, o_i) \wedge s \neq s_i)|$$

where,

- $calls(o, o_i)$ represents the call made from service 'o' to service 'o_i'

[Xianp] defines

$$\begin{aligned} Coupling(S_i) &= p \sum_{j=1}^{n_s} (-\log(P_L(j))) \\ &= \frac{1}{n} \sum_{j=1}^{n_s} (-\log(P_L(j))) \end{aligned}$$

where,

- n_s is the total number of services connected
- 'n' is the total number of services in the entire application
- $p = \frac{1}{n}$ gives the probability of a service participating in any connection

[Kaz+11] defines

$$Coupling = \frac{\sum_{i \in D_s} \sum_{k \in O(s)} CCO(i, k)}{|O(s)| \cdot K}$$

where,

- $D_s = \{D_1, D_2, \dots, D_k\}$ is set of dependencies the service has on other services
- $k = |D_s|$
- CCO is Conceptual Coupling between service operations and obtained from CRUD matrix table constructed with business entities and business capabilities

[Shi+08] defines

$$coupling(s) = \frac{n_c + n_p}{n_s}$$

[AZR11] defines

$$coupling = \frac{\sum_{i=1}^{|O(s)|} (S_{i, sync} + S_{i, async})}{|S|}$$

where,

- $S_{i, sync}$ is the synchronous invocation in the operation O_i
- $S_{i, async}$ is the asynchronous invocation in the operation O_i

#	Figure	Metric Definition
1	[SSPnp]	<p>SCCI: number of operation of other services invoked by the given service</p> <p>SCI: number of services invoked by a given service</p> <p>SMCI: total number of messages from information model required by the operations</p>
2	[Damp]	<p>Coupling is evaluated as information entropy of a complex service components whose entropy is calculated using various data such as total number of atomic components connected, total number of links to atomic components and total number of atomic components.</p>
3	[Kav+11]	<p>The coupling is measured by using the dependency of a service operations with operations of other services. Additionally, the dependency is considered to have different weight value for each kind of operation such as Create, Read, Update, Delete (CRUD) and based on the type of business entity involved. The weight is referred from the CRUD matrix constructed in the previous.</p>
4	[Shi08]	<p>Service coupling is the average number of directly connected services.</p>
5	[AZR11]	<p>Service coupling is the average number of synchronous and asynchronous invocations in all the operations of the service.</p>

Table 1.2: Coupling Metrics

The table 1.2 shows simpler form of metrics proposed for coupling. The coupling of a service is defined in terms of the number of connections it creates with other services in order to fulfill its functionalities. The connections can be due to the dependency with operations of other services or input expectancy of the messages. The more dependency or connection a service has, the coupling increases as well.

1.3.3 Cohesion Metrics

[SSPnp] defines following metric for cohesion.

$$SFCI(s) = \frac{\max(\mu(m))}{|O(s)|}$$

where,

- the number of operations using a message 'm' is $\mu(m)$ such that $m \in M(s)$ and $|O(s)| > 0$

[Shi+08] defines

$$cohesion = \frac{n_s}{|M(s)|}$$

[PRF07] defines

$$SIDC = \frac{n_{op}}{n_{pt}}$$

$$SIUC = \frac{n_{oc}}{n_c * |O(s)|}$$

$$SIUC = \frac{n_{so}}{n_c * |O(s)|}$$

where,

where,

- n_{op} is the number of operations in the service which share the same parameter types n_{pt} is the total number of distinct parameter types in the service
- n_{oc} is the total number of operations in the service used by consumers
- n_{so} is the total number of sequentially accessed operations by the clients of the service itemize [AZR11] defines

$$cohesion = \frac{\max(\mu(OFG, ODG))}{|O(s)|}$$

where,

- OFG and ODG are Operation Gunctionality Granularity and Operation Data Granularity respectively as calculated in section 1.3.4
- $\mu(OFG, ODG)$ gives the number of operations with specific value of OFG and ODG

The table 1.3 gives simplified view for the cohesion metrics. In the basic way, the cohesion is defined in terms of the commonality of the messages consumed by the operations and similarity in the consumer of the operations.

1 Quality of Service

#	Papers	Metrics Definition
1	[SSPnp]	defines SFCI which measures the fraction of operations using similar messages out of total number of operations in the service operations of the service
2	[PRF07]	SIDC : defines cohesiveness as the fraction of operations based on commonality of the messages they operate on SUDC : defines the degree of consumption pattern of the service operations which is based on the similarity of consumers of the operations SILUC : defines the cohesion based on the sequential consumption behavior of more than one operations of a service by other services
3	[Shi+08]	cohesion is given by the inverse of average number of consumed messages by a service
4	[AZR11]	cohesion is defined as the consensus among the operations of the service regarding the functionality and data granularity which represents the type of parameters and the operations importance as calculated in the section 1.3.4

Table 1.3: Cohesion Metrics

1.3.4 Granularity Metrics

[SSPnp] defines

$$SCG = |O(s)|$$

$$SDG = |M(s)|$$

[Shi+08] defines

$$granularity = \frac{|O(s)|^2}{|M(s)|^2}$$

[AZR11] defines

$$ODG = 1 - \left(\frac{\sum_{i=1}^{n_{i0}} W_{pi}}{\sum_{i=1}^{n_{is}} W_{pi}} + \frac{\sum_{j=1}^{n_{j0}} W_{pj}}{\sum_{j=1}^{n_{js}} W_{pj}} \right)$$

$$OFG = \frac{W_i(o)}{\sum_{i=1} |O(S)| W_i(o)}$$

$$SOG = \sum_{i=1}^{|O(S)|} ODG(i) * OFG(i)$$

where,

- W_{pi} is the weight of input parameter
- W_{pj} is the weight of output parameter
- n_{i0} is the number of input parameters in an operation
- n_{j0} is the number of output parameters in an operation
- n_{is} is the total number of input parameters in the service
- n_{js} is the total number of output parameters in the service
- $W_i(o)$ is the weight of an operation of the service

1 Quality of Service

#	Papers	Metrics Definition
1	[SSPnp]	the service capability granularity is given by the number of operations in a service and the data granularity by the number of messages consumed by the operations of the service
2	[AZR11]	ODG : evaluated in terms of number of input and output parameters and their type such as simple, user-defined and complex OPG : defined as the level of logic provided by the operations of the services
3	[Shi+08]	SOG : defined by the sum of product of data granularity and functionality granularity for all operations in the service evaluated as the ratio of squared number of operations of the service to the squared number of messages consumed by the service

Table 1.4: Granularity Metrics

- $|O(S)|$ is the number operations in the service

The table 1.4 presents various view of granularity metrics based on different research papers. The chapter ?? discuss in detail regarding the topic. Here, the focus is made on the metrics. The granularity is evaluated by the number and type of the parameters, number of operations and number of messages consumed.

1.3.5 Complexity Metrics

[ZL09] defines

$$RCS = \frac{CS(s_i)}{|S|}$$

$$RIS = \frac{IS(s_i)}{|S|}$$

where,

- $CS(s_i)$ gives coupling value of a service
- $IS(s_i)$ gives importance weight of a service in an application
- $|S|$ is the number of services to realize the application

[AZR11] defines

$$Complexity(s) = \frac{\sum_{i=1}^{|O(s)|} (SOG(i))^2}{|S|}$$

where,

- $|S|$ is the number of services to realize the application
- $SOG(i)$ gives the granularity of ith service calculated as described in section 1.3.4

#	Papers	Metrics Definition
1	[ZL09]	RCS : complexity given by the degree of coupling for a service and evaluated as the fraction of its coupling to the total number of services RIS : measured as the fraction of total dependency weight of consumers upon the service to the total number of services
2	[AZR11]	the complexity is calculated using the granunarity of service operations

Table 1.5: Complexity Metrics

refer to the document for effect of RCS and RIS on complexity

1.5 provides the way to interpret complexity metrics. The complexity is highly dependent upon coupling and functionality granularity of the service.

1.3.6 Autonomy Metrics

[Ros+11] defines

$$SLC = \frac{1}{h_2 - l_2 + 1} \sum_{i=l_2}^{h_2} \sum_{sr \in SR} (BE_{i,sr} X V_{sr})$$

$$DEP = \frac{\sum_{j=L1_i}^{h1_i} \sum_{k=1}^{BE} V_{sr_{jk}} - \sum_{j=L1_i}^{h1_i} \sum_{k=L2_i}^{j2_i} V_{sr_{jk}}}{nc}$$

$$\text{autonomy} = \begin{cases} SLC - DEP & \text{if } SLC > DEP \\ 0 & \text{otherwise} \end{cases}$$

where,

- nc is the number of relations with other services
- $V_{sr_{jk}}$ is the corresponding value of the action in jkth element of CRUD matrix, it gives the weight of corresponding business capability affecting a business entity
- $l1_i, h1_i, l2_i, h2_i$ are bounding indices in CRUD matrix of ith service

#	Papers	Metrics Definition
1	[Ros+11]	SLC : defined as the degree of control of a service upon its operations to act on its Business entities only DEP : given by the degree of coupling of the given service with other services autonomy is given by the difference of SLC and DEP if SLC > DEP else it is taken as 0

Table 1.6: Autonomy Metrics

1.6 shows a way to interpret autonomy. It is calculated by the difference SLC-DEP when SLC is greater than DEP. In other cases it is taken as zero. So, autonomy increases as the operations of the services have full control upon its business entities but decreases if the service is dependent upon other services.

1.3.7 Reusability Metrics

[SSPnp] defines

$$SRI(s) = P + Q$$

where,

$$P = |\{s_i : \exists_{o \in s}, \exists_{o_i \in s_i} calls(o_i, o) \wedge s \neq s_i\}|$$

and

$$Q = |\{p \in P : s \in p\}|$$

[Shi+08] defines

$$Reusability = \frac{Cohesion - granularity + Consumability - coupling}{2}$$

#	Papers	Metrics Definition
1	[SSPnp]	SRI defines reusability as the number of existing consumers of the service
2	[Shi+08]	evaluated from coupling, cohesion, granularity and consumability of a service where consumability is the chance of the service being discovered and depends upon the fraction of operations in the service

Table 1.7: Reusability Metrics

The table 1.7 shows the reusability metrics evaluation. Reusability depends upon coupling, cohesion and granularity. It decreases as coupling and granularity increases.

Acronyms

CRUD create, read, update, delete.

DEP Dependency.

IFBS International Financial and Brokerage Services.

ISCI Inter Service Coupling Index.

ODC Operation Data Granularity.

RCS Relative Coupling of Services.

RIS Relative Importance of Services.

SDLC Software Development Life Cycle.

SFCI Service Functional Cohesion Index.

SIDC Service Interface Data Cohesion.

SIUC Service Interface Usage Cohesion.

SLC Self Containment.

SMCI Service Message Coupling Index.

SOAF Service Oriented Architecture Framework.

SOCI Service Operational Coupling Index.

SRI Service Reuse Index.

SWIFT Society for Worldwide Interbank Financial Telecommunication.

List of Figures

List of Tables

1.1	Quality Attributes	2
1.2	Coupling Metrics	5
1.3	Cohesion Metrics	7
1.4	Granularity Metrics	8
1.5	Complexity Metrics	9
1.6	Autonomy Metrics	9
1.7	Reusability Metrics	10

Bibliography

- [AL03] M. Alshayeb and W. Li. *An Empirical Validation of Object-Oriented Metrics in Two Different Iterative Software Processes*. Tech. rep. IEEE Computer Society, 2003.
- [AZR11] S. Alahmari, E. Zaluska, and D. C. D. Roure. *A Metrics Framework for Evaluating SOA Service Granularity*. Tech. rep. School of Electronics and Computer Science University Southampton, 2011.
- [BMB96] L. C. Briand, S. Morasca, and V. R. Basili. *Property-Based Software Engineering Measurement*. Tech. rep. IEEE Computer Society, 1996.
- [Bri+np] L. C. Briand, J. Daly, V. Porter, and J. Wüst. *A Comprehensive Empirical Validation of Design Measures for Object-Oriented Systems*. Tech. rep. Fraunhofer IESE, np.
- [EM14] A. A. M. Elhag and R. Mohamad. *Metrics for Evaluating the Quality of Service-Oriented Design*. Tech. rep. Universiti Teknologi Malaysia, 2014.
- [Feu13] G. Feuerlicht. *Evaluation of Quality of Design for Document-Centric Software Services*. Tech. rep. University of Economics and University of Technology, 2013.
- [FL07] G. Feuerlicht and J. Lozina. *Understanding Service Reusability*. Tech. rep. University of Technology, 2007.
- [GL11] A. Goeb and K. Lochmann. *A software quality model for SOA*. Tech. rep. Technische Universität München and SAP Research, 2011.
- [HS90] S. Henry and C. Selig. *Predicting Source=Code Complexity at the Design Stage*. Tech. rep. Virginia Polytechnic Institute, 1990.
- [Kaz+11] A. Kazemi, A. N. Azizkandi, A. Rostampour, H. Haghighi, P. Jamshidi, and F. Shams. *Measuring the Conceptual Coupling of Services Using Latent Semantic Indexing*. Tech. rep. Automated Software Engineering Research Group, 2011.

- [Ma+09] Q. Ma, N. Zhou, Y. Zhu, and H. Wang. *Evaluating Service Identification with Design Metrics on Business Process Decomposition*. Tech. rep. IBM China Research Laboratory and IBM T.J. Watson Research Center, 2009.
- [Man+np] M. Mancioppi, M. Pereplechikov, C. Ryan, W.-J. van den Heuvel, and M. P. Papazoglou. *Towards a Quality Model for Choreography*. Tech. rep. European Research Institute in Services Science, Tilburg University, np.
- [MLS07] Y.-F. Ma, H. X. Li, and P. Sun. *A Lightweight Agent Fabric for Service Autonomy*. Tech. rep. IBM China Research Lab and Bei Hang University, 2007.
- [Nem+14] H. Nematzadeh, H. Motameni, R. Mohamad, and Z. Nematzadeh. *QoS Measurement of Workflow-Based Web Service Compositions Using Colored Petri Net*. Tech. rep. Islamic Azad University Sari Branch and Universiti Teknologi Malaysia, 2014.
- [Per+07] M. Pereplechikov, C. Ryan, K. Frampton, and Z. Tari. *Coupling Metrics for Predicting Maintainability in Service-Oriented Designs*. Tech. rep. RMIT University, 2007.
- [PRF07] M. Pereplechikov, C. Ryan, and K. Frampton. *Cohesion Metrics for Predicting Maintainability of Service-Oriented Software*. Tech. rep. RMIT University, 2007.
- [Ros+11] A. Rostampour, A. Kazemi, F. Shams, P. Jamshidi, and A. Azizkandi. *Measures of Structural Complexity and Service Autonomy*. Tech. rep. Shahid Beheshti University GC, 2011.
- [Shi+08] B. Shim, S. Choue, S. Kim, and S. Park. *A Design Quality Model for Service-Oriented Architecture*. Tech. rep. Sogang University, 2008.
- [SSPnp] R. Sindhgatta, B. Sengupta, and K. Ponnalagu. *Measuring the Quality of Service Oriented Design*. Tech. rep. IBM India Research Laboratory, np.
- [Xianp] W. Xiao-jun. *Metrics for Evaluating Coupling and Service Granularity in Service Oriented Architecture*. Tech. rep. Nanjing University of Posts and Telecommunications, np.
- [ZL09] Q. Zhang and X. Li. *Complexity Metrics for Service-Oriented Systems*. Tech. rep. Hefei University of Technology, 2009.