FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Representation and Visualization of load consumption in D2WORM work units.

Rajendra Kharbuja

FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Representation and Visualization of load consumption in D2WORM work units.

Repräsentation und Visualisierung von Lastverbrauch in D2WORM Arbeitseinheiten.

Author:	Rajendra Kharbuja
Supervisor:	Prof. Dr. rer. pol. Hans-Arno Jacobsen
Advisor:	Martin Jergler
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

1 Granularity

1.1 Introduction

The granularity of a service is often ambiguous and has different interpretation. In simple term, it refers to the size of the service. However, the size itself can be vague. It can neither be defined as a single quantitative value nor it can be defined in terms of single dependent criterion. It is difficult to define granularity in terms of number because the concepts defining granularity are vague and subjective in nature. If we choose an activity supported by the service to determine its granularity then we cannot have one fixed value instead a hierarchical list of answers; where an activity can either refer to a simple state change, any action performed by an actor or a complete business process. [Linnp], [Hae+np]
Although, the interest upon the granularity of a component or service for the business users only depends upon their business value, there is no doubt that the granularity affects the architecture of a system. The honest granularity of a service should reflect upon both business perspective and should also consider the impact upon the overall architecture.
If we consider other units of software application, we come from object oriented to component based and then to service oriented development. Such a transistion has been considered with the increase in size of the individual unit. The increase in size is contributed by the interpretation or the choice of the abstraction used. For example, in case of object oriented paradigm, the abstraction is chosen to represent close impression of real world objects, each unit representing fine grained abstraction with some attributes and functionalities.
Such abstraction is a good approach towards development simplicity and understanding, however it is not sufficient when high order business goals have to be implemented. It indicates the necessity of coarser-grained units than units of object oriented paradigm. Moreover, component based development introduced the concept of business components which target the business problems and are coarser grained. The services provide access to application where each application is composed of various component services. [Linnp]

1.2 Related Work

A number of researches have been done in the area of service and component granularity. The focus of the researches includes various areas such as definition, effect, its dependent criteria and various measurement metrics.
According to [Pad04], the granularity of a component is inversely related to various non functional qualities such as customization and maintainaibility.
According to Hazen and Sims [HS00], component granularity is rather recursive as a component can be composed of various fine-grained components. They further classify recursion as of discrete or continuous nature but then prefer on discrete recursion. Thus, component granularity can be divide into three discrete layers: system, business and distributed where the composition happens from right to left order and granularity decreases in the opposite order. It is also useful to relate granularity with reusability. Coarse-grained components have high reuse efficiency because they are well focused to a specific business functionality. Whereas fine-grained components have high reusability since they focus on small functionality and thus can be used by other coarse-grained components to accomplish higher level business capabilities. [Mil+01], [WXZ05]
Additionally, Sims [Sim05] gave some insights to measure granularity. The granularity can be measured either a) by the number of components called from an operation on a service interface, b) by the number of components calling the service interface or c) by the number of database tables updated.
Service Oriented Architecture Framework (SOAF [EAK06] also gives some way to measure the granularity in terms of quantitative value. The value can be derived from a) the number of components that are invoked by the service interface and b) the number of changes of states in resources like the database tables updates influenced by the service. Again, the granularity can affect reusability and the network communication to accomplish any task. Coarse grained components have more volume of data exchanged where as fine grained components have high frequency of round trips to complete the full business functionality. It is recommended to balance them which can depend upon various factors such as abstraction level, change expectation rate, service complexity, desired coupling and cohesion. Furthermore, the granularity can be at different levels within an application. High-level business process can be realized as a coarse-grained services and each coarse-grained component can be composed of various fine-grained services.
The idea of multi-layered granularity is further supported by Security Trading case study [EAK06] and International Financial and Brokerage Services IFBS case study [Nav08]. The design of pilot application for Security Trading consists of various services across various levels such as process, application, shared data and infrastructures and containing services with granularity decreasing from left to the right in the order they

are listed. The application for IFBS is refactored from large number of fine-grained service to multi-layered granular services such as process services, business services, composite services or orchestration services, utility services etc.
A research performed by Steghuis [Ste06] talks about various aspects of granularity such as functionality, flexibility, reusability, complexity, context-independence, Performance, Genericity and sourcing. Some differenciating aspects in comparision to previous researches are context-independence, genericity and sourcing. Context-independence means independence and self sufficiency in terms of data as well as logic. It promotes loose coupling and insists lack of knowledge of state of surrounding and inessential maintainance of own state. Genericity is the quality of making generic services which can be used in different situations such as messaging service. Finally, sourcing is the process of identifying cohesive group of tasks which can be outsourced by considering the coupling associated.

1.3 Basic Principles Related to Service Granularity

In addition to quantitative perspective on granularity, it can be helpful to approach the granularity qualitatively. The points below are some basic principls derived from various scientific and research papers, which attempt to define the qualitative properties of granularity. Hopefully, these principles will be helpful to come with the service of right granularity.

1. The correct granularity of a component or a service is dependent upon the time. The various supporting technologies that evolve during time can also be an important factor to define the level of vertical decomposition. For eg: with the improvement in virtualization, containerization as well as platform as a service technologies, it is fast and easy for deployment automation which supports multiple fine-grained services creation.[HS00]

2. A good candidate for a service should be independent upon the implementation but should depend upon the understandability of domain experts. [Hae+np; HS00]

3. A service should be an autonomous reusable component and should support various cohesion such as functional (group similar functions), temporal(change in the service should not affect other services), run-time(allocate similar runtime environment for similar jobs; eg. provide same address space for jobs of similar computing intensity) and actor (a component should provide service to similar users). [Hae+np], [HS00]

4. A service should not support huge number of operations. If it happens, it will affect high number of customers on any change and there will be no unified view on the functionality. Furthermore, if the interface of the service is small, it will be easy to maintain and understand.[Hae+np], [RS07]

5. A service should provide transaction integrity and compensation. The activities supported by a service should be within the scope of one transaction. Additionally, the compensation should be provided when the transaction fails. [Hae+np], [Foo05]

6. The notion of right granularity is more important than that of fine or coarse. It depends upon the usage condition and moreover is about balancing various qualities such as reusability, network usage, completeness of context etc. [Hae+np], [WV04]

7. The level of abstraction of the services should reflect the level of real world business activities. [RS07]

8. There can be two better approaches for breaking down an abstraction. One way is to separate redundant data or data with different semanting meaning. The other approach is to divide services with limited control and scope. For example: A Customer Enrollment service which deals with registration of new customers and assignment of unique customer ID can be divided into two independent fine-grained services: Customer Registration and ID Generation, each service will have limited scope and separate context of Customer.[RS07]

Dimensions of Granularity The factors affecting the granularity of any service is presented by R3 model. The three dimensions of the model are as given below: Reach: It gives answer to "Who can use the functionality? ". It provides the levels the functionality of service is accessible from. It can be a customer, the customers within an organization, supplier etc. Range: It gives answer to "Which functionality is available? ". It provides the level of information shared with other systems. It can be a simple data access, a transaction, message transfer etc. Realm: It gives answer to "What kind of functionality? ". It focuses more on the functionality in terms of business value which is the value as perceived by the stakeholders. [RS07]

Fig: R3 model to present various level of granularity

Another interpretation but similar approach has also been published, which presents three different criteria to affect granularity of any service. Data granularity: It evaluates how much data is passed to the interface of the service or returned by the service. Business value granularity: It measures the inclination of the service towards the business. The business value can be predicted by e3 approach or i* framework. Moreover, it

considers contribution of service towards the achievement of business goals and values. Functionality granularity: It refers to the default functionality offered by the service as well as additional optional functionalities provided by the service. [Hae+np]

An additional contribution to defining the granularity has been made when the differentiating features of distributed system are taken into consideration. The granularity of a service depends upon the coupling and cohesion. In order to find the right granularity for a service, we need to balance between coupling and cohesion. The coupling can be determined from the dependency graph between a service and its consumers. The goal is to minimize the coupling of individual service and to the application as a whole. The cohesion is evaluated by the likeliness of the operations supported by the service. It considers that higher the similarities among the operations, higher the cohesion is. [Xianp], [Per+np], [Shi+08]

Service Hierarchy

Acronyms

IFBS International Financial and Brokerage Services.

SOAF Service Oriented Architecture Framework.

List of Figures

Bibliography

[EAK06] A. Erradi, S. Anand, and N. Kulkarni. *SOAF: An Architectural Framework for Service Definition and Realization*. Tech. rep. University of New South Wales, Infosys Technologies Ltd, 2006.

[Foo05] D. Foody. *Getting web service granularity right*. 2005.

[Hae+np] R. Haesen, M. Snoeck, W. Lemahieu, and S. Poelmans. *On the Definition of Service Granularity and Its Architectural Impact*. Tech. rep. Katholieke Universiteit Leuven, np.

[HS00] P. Herzum and O. Sims. *Business Components Factory: A Comprehensive Overview of Component-Based Development for the Enterprise*. John Wiley Sons, 2000.

[Linnp] D. Linthicum. *Service Oriented Architecture (SOA)*. np.

[Mil+01] H. Mili, A. Mili, S. Yacoub, and E. Addy. *Reuse-based software engineering: techniques, organization, and controls*. Wiley-Interscience New York, 2001.

[Nav08] V. D. Naveen Kulkarni. *The Role of Service Granularity in A Successful SOA Realization – A Case Study*. Tech. rep. SETLabs, Infosys Technologies Ltd, 2008.

[Pad04] F. Z. Padmal Vitharana Hemant Jain. *Strategy-Based Design of Reusable Business Components*. Tech. rep. IEEE, 2004.

[Per+np] M. Perepletchikov, C. Ryan, K. Frampton, and Z. Tari. *Coupling Metrics for Predicting Maintainability in Service-Oriented Designs*. Tech. rep. RMIT University, np.

[RS07] P. Reldin and P. Sundling. *Explaining SOA Service Granularity– How IT-strategy shapes services*. Tech. rep. Linköping University, 2007.

[Shi+08] B. Shim, S. Choue, S. Kim, and S. Park. *A Design Quality Model for Service-Oriented Architecture*. Tech. rep. Sogang University, 2008.

[Sim05] O. Sims. *Developing the architectural framework for SOA - part 2-service granularity and dependency management. CBDI Forum Journal*. Tech. rep. CBDI, 2005.

[Ste06] C. Steghuis. *Service Granularity in SOA Projects: A Trade-off Analysis*. Tech. rep. University of Twente, 2006.

[WV04] L. Wilkes and R. Veryard. "Service-Oriented Architecture: Considerations for Agile Systems." In: *np* (2004).

[WXZ05] Z. Wang, X. Xu, and D. Zhan. *A Survey of Business Component Identification Methods and Related Techniques*. Tech. rep. International Journal of Information Technology, 2005.

[Xianp] W. Xiao-jun. *Metrics for Evaluating Coupling and Service Granularity in Service Oriented Architecture*. Tech. rep. Nanjing University of Posts and Telecommunications, np.