



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

Acknowledgments	iii
Abstract	iv
1 Granularity	1
1.1 Introduction	1
1.2 Related Work	2
Acronyms	3
List of Figures	4
Bibliography	5

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. [1][2]

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 transition 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.

Nevertheless, such abstraction is a good approach towards development simplicity and understanding, 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. [1]

Related Work

1.2 Related Work

Hazem and Sims have also supported the hierarchical nature of components granularity. A component functionality is composed of various fine grained components and thus leads to a hierarchy level of granularity. [3] According to [4], the granularity of a component is inversely related to various non functional qualities such as customization and maintainability.

The following section provides some views regarding the granularity based on various research papers. 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. [3] 2. A good candidate for a service should be independent upon the implementation but depends upon the understandability of domain experts. [2, 3] 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). [2, 3] 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. [2] 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. [2,5] 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. [2, 6] 7. The level of abstraction of the services should reflect the level of real world business activities. [7]

Service Hierarchy

Acronyms

D2WORM Distributed Data-centric Workflow On-line Resource Management.

WMS Workflow Management System.

List of Figures

Bibliography

- [Alo+np] G. Alonso, D. Agrawal, A. E. Abbadi, and C. Mohan. *Functionality and Limitations of Current Workow Management Systems*. Tech. rep. IBM Almaden Research Center and Institute of Information Systems, np.
- [CHnp] D. Cohn and R. Hull. *Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes*. Tech. rep. IBM T.J. Watson Research Laboratory, np.
- [CMGnp] S. Chakraborty, S. Majumder, and D. Goswami. *A Model for Load Balancing in Distributed System using -Congestion Game*. Tech. rep. Indian Institute of Technology, np.
- [Dog+np] A. Dogac, E. Gokkoca, S. Arpinar, P. Koksall, I. Cingil, B. Arpinar, N. Tatbul, P. Karagoz, U. Halici, and M. Altinel. *Design and Implementation of a Distributed Workow Management System METUFlow*. Tech. rep. Middle East Technical University, np.
- [JSJnp] M. Jergler, M. Sadoghi, and H.-A. Jacobsen. *D2WORM: A Management Infrastructure for Distributed Data-centric Workflows*. Tech. rep. Technische Universitaet Munich, np.
- [KCH08] I. Kuz, M. M. T. Chakravarty, and G. Heiser. *COMP9243 — Week 1*. Tech. rep. The University of New South Wales, 2008.
- [LHL00] S. I. Lee, D. Han, and D. Lee. *A Pattern for Managing Distributed Workflows*. Tech. rep. Information and Communications University, 2000.
- [Lud+05] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. *Scientific Workflow Management and the Kepler System*. Tech. rep. 2005.
- [SGH95] A. SHETH, D. GEORGAKOPOULOS, and M. HORNICK. *An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure*. Tech. rep. Kluwer Academic Publishers, 1995.
- [SLBnp] J. Schneider, B. Linnert, and L.-O. Burchard. *Distributed Workflow Management for Large-Scale Grid Environments*. Tech. rep. Technische Universitaet Berlin, np.

Bibliography

- [SPV11] M. van Steen, G. Pierre, and S. Voulgaris. *Challenges in very large distributed systems*. Tech. rep. np, 2011.
- [SSnp] K. P. Stoilova and T. A. Stoilov. *Evolution of the workflow management systems*. Tech. rep. Institute of Computer and Communication Systems, np.