Chapter II

Beyond Application-Oriented Software Engineering: Service-Oriented Software Engineering

Jiehan Zhou
VTT Technical Research Centre of Finland, Embedded Software, Finland

Eila Niemelä
VTT Technical Research Centre of Finland, Embedded Software, Finland

Abstract

This chapter introduces SOSE (Service-Oriented Software Engineering) as an advanced software development. It argues that SOSE is characterized by small projects, existing software reuse, market changing and software evolution focusing, customer domination, and common middards in comparison with AOSE (Application-Oriented Software Engineering). It presents SOSE software development methodology involving the main processes of service extracting, service middard, service circulation, service evaluation, and service evolution with the middard service fundamental. Eventually, compared with other industries (for example, car manufacturing, construction, and electronics) with global standards and fine-granularity components, the software industry is immature in unified service standards, service marketplace, and service granularity.
evaluation. The authors hope that understanding the underlying fundamental SOSE middard service and SOSE methodology will make the software industry more productive and profitable.

Introduction

As the number of component services (for example, ActiveXs, DCOMs, and CORBAs) grows, e-business software development is coming into being. Concretely, over-engineered systems with redundant functionality are not required for the majority of customers. Software organizations are typically of a small size, in a state of continual process change, never arriving, and always in transition (Bennett et al., 2000). Software developers prefer exploiting the services available in the marketplace to produce the most effective software in the least time rather than programming from scratch. If needed, software will be produced as a particular service, instead of “a system,” conforming to a service standard technology. The system could be composed, executed, maintained, and evaluated in the way of online service procuring, engaging, and changing.

Currently, almost all commercial application software is sold on the basis of ownership (Bennett et al., 2002). Thus, a customer buys the object code with some form of license to use it. Any updates, however important to the customer, are the responsibility of the vendor. Any attempt by the customer to modify the application is likely to invalidate warranties as well as ongoing support. This form of marketing, known as supply-side, is facing the following challenges:

- Bringing together users and providers of software in a trusted marketplace.
- The continuously changing software market and customer needs. Today’s software development is in the way of e-business, in which customers are expecting and demanding various timely services from sites, not costly and time-consuming turnkey products.
- Reducing software development cost and time. Supply on customers’ demand is one of the most successful ways to reduce software development cost and time.
- Large-scale and complex software systems. The systems we need to build are likely to get more complex. Making service standards or specifications enables us to successfully develop large complex software systems.
- Evolution in Internet time. This challenge is to achieve very fast change yet provide very high quality software. Existing software maintenance processes are simply too slow to meet the needs of much faster implementation of software changes.

In recognition of these challenges, studies have been running all over the world, aiming at developing new approaches to software development for highly agile software systems, which design, implement, test, evaluate, and access services across the Web.
Related Content

Testing of Future Internet Applications Running in the Cloud
www.igi-global.com/chapter/testing-future-internet-applications-running/72237?camid=4v1a

Ethics and Social Networking: An Interdisciplinary Approach to Evaluating Online Information Disclosure
www.igi-global.com/article/ethics-and-social-networking/153679?camid=4v1a

A Software Tool for Reading DICOM Directory Files
www.igi-global.com/chapter/software-tool-reading-dicom-directory/29441?camid=4v1a

Building Secure Software Using XP
www.igi-global.com/article/building-secure-software-using/58508?camid=4v1a