Chapter 11
Decreasing Service Coupling to Increase Enterprise Agility

Jose Carlos Martins Delgado
University of Lisbon, Portugal

ABSTRACT

The agility with which an Enterprise Information Systems (EIS) can be changed is of primordial importance in today's fast paced, competitive market. This depends on the changeability of the EIS software resources, which in turn is heavily influenced by the coupling between the services implemented by these resources. This chapter presents a solution to the coupling problem based on the concepts of structural compliance and conformance, in which compatibility between interacting services does not rely on a shared schema. Instead, it checks resources component by component, recursively, until primitive elements are reached. However, coupling is not a single-faceted issue and involves several aspects and slants. To help in understanding and systematizing them, the chapter proposes a multidimensional framework that caters for EIS lifecycle stages, concreteness (with various levels of abstraction), interoperability (based on structural compliance and conformance), and concerns (to deal with non-functional aspects such as security, reliability and quality of service).

INTRODUCTION

Enterprise agility can be defined as the capacity of an enterprise to adapt (reactively and/or proactively) to changes in its environment in a timely and cost efficient manner (Ganguly, Nilchiani, & Farr, 2009). Agile software development methods (Dingsøyr, Nerur, Balijepally, & Moe, 2012) seek to improve productivity in the development of Enterprise Information Systems (EIS) by trying to provide a better match between human abilities (such as creativity and programming) and problem solving constraints, such as complexity and team management.

A complex, potentially distributed software system, such as an Enterprise Information Systems (EIS), typically involves many interacting *resources* (modules, subsystems) with many decisions to take and

DOI: 10.4018/978-1-4666-8510-9.ch011

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

many tradeoffs to endure, not only in each resource but also in the ways in which the various resources interact. In this chapter, "resource" is a generic designation that can refer to a simple module, an EIS subsystem, a complete EIS or even a set of EIS (in a multi-enterprise collaboration).

Ideally, resources should be completely decoupled, with no constraints on one another and completely independent lifecycles. This would allow separate development of each resource and elimination of agile programming inefficiencies due to interactions between the specifications of resources, which usually cause iterations and changes in their requirements.

However, resources do need to interact and to cooperate, to collectively fulfill the intended goals of the application. Therefore, a fundamental tenet of agile programming is to reduce resource coupling as much as possible without hindering the interaction capabilities necessary to support the required resource interoperability. Decoupling also translates into a higher:

- Changeability (a change in a resource is less likely to have a significant impact on other resources);
- Adaptability (less constraints require less effort to adapt to changes in other resources);
- Reusability (a resource with less requirements and constraints has an increased applicability range);
- Reliability (a smaller set of requirements simplifies the task of finding an alternative in case of failure).

Although decoupling may be deemed a fairly obvious goal, tuning it to the right degree in practice is not an easy task.

To start with, an EIS should be as independent as possible from technological solutions, but the fact is that existing architectural models, such as SOA (Service Oriented Architecture) (Erl, 2008) and REST (Representational State Transfer) (Fielding, 2000), impose much more constraints and coupling than those actually required by the specifications of the systems under programming.

SOA is good at modeling enterprise systems but involves rather complex and static software specifications and entails sharing schemas between consumer and provider, which is a heavy form of service coupling. Performing a change in an interaction between Web Services (the typical SOA instantiation) is not trivial. RESTful APIs (Application Programming Interfaces) are much simpler, justifying the increasing popularity of REST (Pautasso, Zimmermann & Leymann, 2008), but is rather low level and not the best match for general-purpose, state-oriented enterprise applications. It also hides a high level of coupling, since it requires that both interacting resources share the same media type specification.

In addition, coupling is not a monolithic issue and needs to be detailed in its various slants. The main goal of this chapter is to systematize the aspects pertinent to interoperability and coupling and how they can contribute to enterprise agility, taking a service-oriented perspective (each resource is modeled by the service its implements) and a coupling minimization approach (how should resources interact to minimize coupling). The topics discussed can be equally applied at intra-application and inter-application levels (interacting resources compiled and linked together or distributed, respectively), although the emphasis is on distributed resources, in which reducing coupling is more critical, since resources can be changed independently at any time.

The chapter is organized as follows. The Background section describes some of the existing technologies and frameworks relevant to the context of this chapter, followed by a discussion on the meaning of enterprise agility and on the coupling problem. To understand it better, and how it relates to the interoperability problem, a multidimensional framework is outlined, presenting several levels of abstraction

35 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/decreasing-service-coupling-to-increase-enterprise-agility/135230

Related Content

Runtime Verification of Distributed Programs
Eslam Al Maghayreh (2012). *Advanced Automated Software Testing: Frameworks for Refined Practice (pp. 49-67).*
www.irma-international.org/chapter/runtime-verification-distributed-programs/62150

Traffic Data Collection and Visualization Tool for Knowledge Discovery Using Google Maps
Iftekhar Hossainand Naushin Nower (2022). *International Journal of Software Innovation (pp. 1-12).*
www.irma-international.org/article/traffic-data-collection-and-visualization-tool-for-knowledge-discovery-using-google-maps/293270

A Robust and Lightweight Key Management Protocol for WSNs in Distributed IoT Applications
Muhammad Ranaand Quazi Mamun (2018). *International Journal of Systems and Software Security and Protection (pp. 1-16).*
www.irma-international.org/article/a-robust-and-lightweight-key-management-protocol-for-wsns-in-distributed-iot-applications/232746

Swing Weight Development for Software Platforms
(2023). *Adaptive Security and Cyber Assurance for Risk-Based Decision Making (pp. 86-114).*
www.irma-international.org/chapter/swing-weight-development-for-software-platforms/320459

Color Quantization Based Artificial Bee Colony And Training Tools
(2022). *International Journal of Software Innovation (pp. 0-0).*
www.irma-international.org/article//301217