Chapter IX

Composing Systems of Systems: Requirements for the Integration of Autonomous Computer Systems

Panayiotis Periorellis, University of Newcastle upon Tyne, UK

Abstract

Information Systems in general carry or have embedded in their structure, elements that stem from the organization’s strategic, tactical, and operational goals. Finding elements of an organization’s strategic, tactical, or operational goals embedded in computer systems is not at all surprising, since most developers and programmers were taught how to successfully map such goals into the Information System. We are, however, in an era where technology allows us to develop systems that are composed of smaller autonomous parts (sometimes complete systems themselves) that are integrated together despite being bound by their corresponding organizational boundaries. Therefore integration is not only a technical challenge but an organizational one, too. In this chapter we address a number of issues, namely system composition, regulation, evolution, and dependability, using examples from the two case studies we worked on for three years.
Introduction

Information Systems in general carry or have embedded in their structure, elements that stem from the organization’s strategic, tactical, and operational goals. One can easily “see” the level of trust an airline system has in its clients when asking for a credit card prior making the booking (and therefore updating the database), as opposed to another that updates the database and asks the user to pay over the phone within a time limit. Finding elements of an organization’s strategic, tactical, or operational goals embedded in computer systems is not at all surprising since most developers and programmers were taught how to successfully map such goals into the Information System. We are, however, in an era where technology allows us to develop systems that are composed of smaller autonomous parts (sometimes complete systems themselves) that are integrated together despite being bound by their corresponding organizational boundaries.

Many of the technical challenges have indeed been solved to allow the deployment of purpose-built modular pieces of code that provide some valuable service. Architectures such as Web services, which various organizations have started exploiting by exposing parts of their services over the Web, will reach commercial maturity or critical mass in the forthcoming years. The benefit such architectures add is that systems can be broken into smaller pieces not only for the purpose of analyzing their complexity, but also for the purpose of exploiting the value of the individual components. A company, for example, that runs a Web site for selling books could now “lease” its shopping cart component while at the same time maintaining the component as part of the overall business.

There is, however, an additional level of complexity that has not been addressed yet. In order to understand and consequently tackle the organizational challenges of building systems, we need to distinguish between distributed processing and distributed control. Distributed processing implies distribution of a task over a number of resources, whereas distributed control implies control of a task from a number of unrelated parties. This is what has not been addressed so far and where the originality of this work lies.

The conclusions of this chapter have been drawn by the two case studies that we developed throughout the course of our three-year project. The first case study involves a travel agency composed of individual autonomous Web-based booking services (that is, car rentals, hotels, airlines, insurance companies) to provide to the user via a Web front-end the ability to book a full trip. The case study Periorellis & Dobson (2001) was a technical one in nature. It involved the integration of autonomous booking systems of airlines, car rental systems, and hotels to provide an integrated travel agency. At times we refer to it as the TA case study in the chapter. The second case study, Ferrante & Diu (2002), involves a pan-European electrical power distribution system composed of national electricity grids. Sometimes we refer to it as the EXAMINE case study. It was a rather rich case study given the political, economic, and cultural diversity that has to be reflected on the software that will carry the actual distribution. We use examples from these case studies to emphasize certain points throughout the chapter.

The chapter raises a number of issues (which have so far been overlooked), and they all stem from one main question: “What happens when we integrate autonomous systems (that is, systems of systems) when the organizational elements embedded in them have conflicting interests.” In a nutshell three possible outcomes can happen: good things
A Model-Driven Approach for the Design and Implementation of Software Development Methods
Mario Cervera, Manoli Albert, Victoria Torres and Vicente Pelechano (2012). 
*International Journal of Information System Modeling and Design* (pp. 86-103). 
www.igi-global.com/article/model-driven-approach-design-implementation/70927?camid=4v1a