

# **Part II**

## **Software Methodologies and Technologies**

Networked enterprises cannot perform well without a software infrastructure supporting data exchange and coordination. In the current practice, as argued in Part I, these activities are often performed in an ad hoc manner, often, without a specific or with limited software support. This part focuses on some of the challenges that arise when defining a software infrastructure for the networked enterprise.

As a networked enterprise is typically a dynamic entity where specific organizations can enter and exit over time, the underlying software infrastructure has to self-adapt dynamically to these changes. For instance, while at a certain time the services offered by a company, let us call it *A*, are available, these same services could be unavailable at a later time, or they could become less convenient than others offered by a new company *B* that has just entered the market. Moreover, while in a situation of high demand for its products the organization could find convenient to rely on a personalized logistic service, this same service might not be convenient in a situation with a small size demand. In this last case, it could be more convenient to coordinate with other companies with the same needs to share the logistic support. As another example, if we consider the case of a small winery relying on its own production of grapes, this winery might decide to go for the grapes available on the market if the weather conditions appear to be critical for the in house production.

The need for self-adaptation of software infrastructures is one of the problems addressed by the autonomic computing field. Thus, in Chapter 6 we review this research field and discuss on how it addresses the needs of networked enterprises. Moreover, we propose our autonomic framework called the *SelfLets* that focuses on enabling self-adaptation in the cooperation among organizations.

The need for being adaptable is not only a requirement concerning the inter-companies infrastructure, but it also concerns the infrastructure in charge of managing the processes that are local to a specific company. In this case, processes may evolve to account for new situations that have occurred in the company and that are sensed by proper, physical or logical sensors. This problem is addressed by the autonomic workflow management system proposed in Chapter 7, while the issues concerned with the design, deployment, and management of physical sensors and actuators is the focus of Part IV, in this chapter we focus on the interaction between these sensors and actuators and the core processes of each company, as well as on the interaction between these same processes and the data that are acquired and conceptualized in the company repositories according to the techniques and guidelines described in Part III.

Autonomic infrastructures and workflows need to be verified as any other piece of software. While verifying traditional software is typically an off-line activity performed before the system is deployed in its operation environment, verifying self-adaptable software means being able to check it while it is running. This activity is called on-line verification and exploits specifically tailored analysis and testing techniques that do not significantly interfere with the execution of the system. A literature review on these aspects is conducted in Chapter 8. This chapter also offers an overview of the approach we have taken to verify the evolving workflow.