Chapter 3
Ubiquitous Integration
Architectural Issues

João Henriques
Instituto Politécnico de Viseu, Portugal

Paulo Tomé
Instituto Politécnico de Viseu, Portugal

ABSTRACT

The paradigm of heterogeneous systems has been demanding the implementation of new approaches to ensure the organizational operations as a whole. This chapter proposes a new integration model architecture, supported by research in an organization with great needs at integration systems level which can be applied to any situation, based on the execution of instructions transmitted between systems. The new architecture sustains the interoperability process between different systems that can be applied in an environment populated by heterogeneous systems and is supported by the use XML language. To this architecture contributed decisively different integration experiences and concepts provided by SAP R/3.

1. INTRODUCTION

In this chapter, the concepts of: ubiquitous computing and systems integration play an important role. In this section, these concepts will be properly explained in the context of computer science.

According York and Pendharkar (York & Pendharkar, 2004), it was Weiser (Weiser, 1999) that first used the concept of ubiquitous computing. Weiser considered that through ubiquitous computing we “have machines that fit the human environment instead of forcing humans to enter theirs . . .”. Usually ubiquitous computing refers to building a global computing environment where seamless and invisible access to computing resources is provided to the user (Singh, Puradkar, & Lee, 2005).

The concepts of persavive computing, ambient intelligence and eveywere are usually related with ubiquitous computing. Although, the three concepts mentioned have slightly different aspects. In the literature of the domain, such as (Singh, Puradkar, & Lee, 2005).and (Hansmann, 2003), it can be found definitions of these concepts.

DOI: 10.4018/978-1-4666-8368-6.ch003
Systems integration is an activity that requires interoperability between different entities, adding value and reducing costs maintenance, and an area that needs research and development (Land, 2006). The interoperability process contains various entities that intend to change some information to contribute with benefits to the involved business processes, optimizing its activities in the context of organizations, including information systems. Moreover, the evolution, migration and integration of the existing software to the software legacy is an enormous challenge to business (Bachmann, F., Bass, L., Buhman, C., Comella-Dorda, S., Long, F., Robert, J. E., & Seacord, R. C, 2000), (Britton & Bye, 2004).

2. SYSTEMS INTEGRATION

According to Reynolds, (Reynolds, 2011), fundamental integration concepts include Enterprise Integration and Enterprise Service Bus, middleware and messaging, to support data transfer.

The concept of Enterprise Integration was first introduced by Vernadt, (Vernadat, 1996), as a technical field of Enterprise Architecture, which focused on the study of topics such as system interconnection, electronic data interchange, product data exchange and distributed computing environments. Enterprise architecture was also defined by Brosey et al., (Brosey, Neal, & Marks, 2001), as a discipline and enabling technology that connects and combines people, processes, systems, and technologies to ensure that the right people and the right processes have the right information and the right resources at the right time to optimally perform their functions.

According to Ortiz, (Ortiz Jr., 2007), an Enterprise Service Bus is the middleware glue that holds an SOA together and enables communication between Web-based enterprise applications. The work in this domain started in 2002 in the Gartner Group (Schulte, 2002).

According to the McGraw-Hill encyclopedia of networking & telecommunications, (Sheldon, 2001), middleware is a layer of software or functionality that sits between one system and another, and provides a way for those systems to exchange information or connect with one another even though they have different interfaces. Messaging is one of the methods that has become integral to the way that middleware is implemented. Middleware and messaging may be employed within an organization to tie together its LAN and legacy systems, its diverse clients and back-end databases, and its local and remote systems. Middleware is also important for Web applications.

Recent integration architectures are based on concepts such as event-driven architecture, grid computing or extreme transaction processing (Poduval, A., Todd, D., Sarang, P., Gaur, H., Bolie, J., Geminiuc, K., & Pravin, L. 2011).

Michelson, (Michelson, 2006), says that in an event-driven architecture, a notable thing happens inside or outside your business, which disseminates immediately to all interested parties (human or automated). The interested parties evaluate the event, and optionally take action. The event-driven action may include the invocation of a service, the triggering of a business process, and/or further information publication/syndication. Yochem, (Yochem, 2008), says that an event-driven architecture consists of event creators and event consumers. The creator, which is the source of the event, only knows that the event has occurred. Consumers are entities that need to know the event has occurred; they may be involved in processing the event or they may simply be affected by the event.

Usually, (Puschmann & Alt, 2004), the following integration levels are considered: data level, object level and process level. As can easily be concluded, these types of integrations consider different types of elements.