Chapter 40

Model Driven Integration of Heterogeneous Software Artifacts in Service Oriented Computing

Eric Simon
Laboratoire Informatique de Grenoble, France

Jacky Estublier
Laboratoire Informatique de Grenoble, France

ABSTRACT

Systems evolutivity requires complex operations on services, including migration, duplication, updating, and a number of administration-related actions. However, current environments are heterogeneous and require integration to manage services. It is a complex problem, because it implies a transformation of life cycle related concepts. This integration does not fit very well in the service-oriented approach: indeed this approach is consumer-centered and considers that services are hosted by third parties, while administration is a provider view. Therefore, there is a gap between technologies used to compose applications and technologies that provide them. In the context of system adaptability, this gap becomes a major challenge to be solved. The authors propose an execution environment, which provides a homogeneous service representation used to integrate: their functionalities, their life-cycle and management operations, and lifecycle related concerns, like deployment. Their approach includes two integration mechanisms: the technologies integration supported by wrappers and concerns integration supported by the run-times.

INTRODUCTION

Current trend in software engineering aims to integrate a patchwork of artifacts like devices or software components in a global and common universe. We observe this trend for example in the Ubiquitous Computing and Cloud Computing domains. It is interesting to observe that, despite of their opposite administration logic, these domains share a number of requirements related

to distribution, dynamic discovery, composition and interoperability. On one hand, the Ubiquitous Computing domain aims to interconnect computing devices to provide added-value services to people. In this domain, the interaction between a consumer and devices has to be transparent. On the other hand, the Cloud Computing domain aims to virtualize the infrastructure and to modularize the framework, in order to execute a distributed application according to technical and financial requirements. The localization of the artifacts, presented as services, is not relevant. In these domains, the purpose is to focus on Quality of Service (QoS) like availability for instances. Here, it is the topology of the infrastructure, which is transparent.

Both domains require the integration of data and functions and use Service-Oriented Computing (SOC) to implement them. Data integration is related to data acquisition and transmission from a source system to a target system. Devices in the Ubiquitous Computing, and software components in the Cloud, are both developed to be reused. Consequently, they are not directly compatible and require some sort of alignment. In Service-Oriented Computing, this integration is generally made possible through abstraction (common format) or on the fly transformation. Functionalities integration aims to reuse the functionalities of an existing system. Similarly to data integration, being in the Cloud Computing or in the Ubiquitous Computing, functionalities are presented as Services to be shared and reused between systems. Indeed the Service-Oriented approach allows this integration type because it allows abstracting a legacy system to focus on its functionalities; it is the goal of the Enterprise Service Bus (ESB) technologies (Chappell, 2004). In this type of integration, the syntax, and the grammar of a description is transformed upon operation calls.

However, the administration logic of these two domains is really different. Indeed, in the Ubiquitous Computing domain, the adaptation

and the administration of applications are driven by the changes of the devices availability. In the Ubiquitous Computing, the administration is made in a bottom-up way. In the Cloud Computing, the administration of the application is driven by the requirements of the application Quality of Service. In the Cloud, the administration is made in a top-down way. In both cases, devices and services in the Cloud are often implemented using different technologies. The multiplication of technologies in a same system involves the integration of the administration operations to automatize the administration or to be understandable and usable by a human administrator. However, administration integration is much more a complex problem than data and functionalities integration. It is no longer a data transformation or operation alignment issue but the transformation of life-cycle related concepts. Furthermore, contrarily to data and functionalities integration, administration integration does not fit very well in the Service-Oriented approach: indeed this approach is consumer-centered and considers that services are hosted by third parties, while the administration is based on a provider view. So there is a gap between the technologies used to compose applications (e.g.: BPEL [IBM, 2003]) and service consumers, and technologies that provide them.

However, the current needs in terms of adaptability and evolutivity of systems require the services migration, their duplication, updating, and a set of actions related to administration. Therefore, in the context of adaptability, this gap becomes a major challenge to be solved.

In the first section, we present the Cloud and Ubiquitous domains according to the administration needs for adaptation. This section extracts the essential information characterizing our problem.

In the second section, we propose an approach to design a platform for the Model Driven Integration of Heterogeneous Software Artifacts in Service Oriented Computing. This platform can

28 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/model-driven-integration-of-heterogeneous-software-artifacts-in-service-oriented-computing/127883

Related Content

A Secure Electronic Voting
Wamedh Wadood Abdoul Hameed (2011). *Handbook of Research on E-Services in the Public Sector: E-Government Strategies and Advancements (pp. 431-449).*
www.irma-international.org/chapter/secure-electronic-voting/46280

Actor Network Theory (ANT)
(2019). *Social Partnership and Governance Under Crises (pp. 100-120).*
www.irma-international.org/chapter/actor-network-theory-ant/227518

Science and the Morality of Weapons Research
John Forge (2021). *Research Anthology on Military and Defense Applications, Utilization, Education, and Ethics (pp. 258-270).*
www.irma-international.org/chapter/science-and-the-morality-of-weapons-research/284319

Addressing Big Issues in Small Towns: Graduate Students' Leadership Projects in Rural Areas
Meredith Jonesand Susan Catapano (2020). *Handbook of Research on Leadership and Advocacy for Children and Families in Rural Poverty (pp. 242-258).*
www.irma-international.org/chapter/addressing-big-issues-in-small-towns/253065

Pharmaceutical Product Complaint Handling: Evolution of A Systematic Mechanism
Nirmal Kumarand Ajeya Jha (2019). *International Journal of Public and Private Perspectives on Healthcare, Culture, and the Environment (pp. 43-58).*
www.irma-international.org/article/pharmaceutical-product-complaint-handling/231618