Chapter 6
Engineering Spatial Services:
Concepts, Architecture, and Execution Models

Giovanna Di Marzo Serugendo
University of Geneva, Switzerland

Jose Luis Fernandez-Marquez
University of Geneva, Switzerland

Francesco Luca De Angelis
University of Geneva, Switzerland

ABSTRACT

A new category of services based on data propagation among mobile devices is evolving. Traditionally, distributed applications are engineered on top of a mobile infrastructure in an ad hoc manner that does not conform to standard software engineering practices, such as modularization and reuse. Furthermore, current service-oriented approaches are not suited to this new generation of services that are dispersed on diffused Clouds, are subjected to dynamicity in the underlying mobile and computing infrastructures, and are required to handle time-related and space-related data. This chapter describes the notion of Spatial Services and delineates the design of these services and applications using self-organizing services that implement the active tuple space model inspired by chemical reactions. It explores the spontaneous self-composition of services, self-management of non-functional aspects, and separation of concerns to guarantee specified quality of service. The chapter presents an architectural style and describes execution models to implement such services.

INTRODUCTION

Mobile phones, laptops, and tablets locally connected to each other form a large computing and storage infrastructure - currently under-exploited - but available on motorways, city-centers, inside buildings, etc. These infrastructures pave the way for a new category of services based on data propagation among devices, such as car traffic control through propagation of data from one car to another, information dissemination in a crowd to better steer the crowd towards points of interest or emergency exits, and alternative communication in case of environmental disasters. Such services are time-related, i.e., they may last just for a very short time for a specific purpose exploiting current contextual data; as well as space-related, because the data they rely on or the data they disseminate is spatially distributed over a geographic area.

Traditional service-oriented approaches allow developers to combine together diverse functionalities provided by different services into a new service. Typical approaches involve Web services or API mashups. Composition of services is generally foreseen at design-time; adaptation happens by choosing the actual services at runtime. Current service-oriented approaches are not suited for the new generation of services that are dispersed on diffused Clouds, enable smart environments, or exploit Internet of Things (IoT) scenarios. They do not cope with the dynamicity demanded by the underlying mobile and changing computing infrastructures, the spatiality of the considered data, or time-related issues. There is a need for

1. **Spatial Distribution of Services:** The functionality of the service is provided as the result of the interactions among multiple entities, possibly distributed across several computational nodes.
2. **Services Built On-Demand by Collaboration:** Services are built dynamically through collaboration with other

services. For example, when users query to retrieve the closest vacant parking place in a smart city, sensors, IoT, and services spontaneously collaborate to query the diffused data Clouds to provide the answer.

We call such services *Spatial Services*. The development of these spatial services faces multiple challenges. One challenge is their engineering: from the physical ad hoc connection of devices, to the reliability and real-time adaptation of services and applications hosted on those mobile and dynamic infrastructures, to availability at a large scale to thousands of users simultaneously.

Self-organization, as exhibited by natural systems, provides an appealing approach to engineer spatial services where the functionality and coordination arises from the interactions among several autonomous entities; no central control is required; and robustness properties and adaptation in dynamic large-scale environments are naturally provided. Self-organization is achieved by the use of *self-organizing mechanisms*, i.e., rules that autonomous entities employ to coordinate their behavior, usually following information gathered from their local environment.

Current proposals for engineering systems using self-organizing principles rely on ad hoc techniques which are far from standard software engineering practices, where applications are developed in a modular and systematic way by using (and reusing) libraries and lower-level services. Our proposal is to engineer spatial services and applications, as discussed above, using services that we call *self-organizing services* that provide self-organizing mechanisms which abstract away the implementation of low-level operations from the developer and favor the reuse of code. Self-organizing services implement self-organizing mechanisms which higher-level spatial services and applications can reuse. We complete our proposal with the specification of a *layered architectural style* organized around the notion of a

Related Content

Text-Dependent and Text-Independent Writer Identification Approaches: Challenges and Future Directions
Rajandeep Kaur, Rajneesh Raniand Roop Pahuja (2022). *International Journal of Software Innovation (pp. 1-23).*
www.irma-international.org/article/text-dependent-and-text-independent-writer-identification-approaches/297514

Admission Control in the Cloud: Algorithms for SLA-Based Service Model
Jose Luis Vazquez-Poletti, Rafael Moreno-Vozmedianoand Ignacio M. Llorente (2014). *Handbook of Research on Architectural Trends in Service-Driven Computing (pp. 701-717).*
www.irma-international.org/chapter/admission-control-in-the-cloud/115450

Multiagent Based Product Data Communication System for Computer Supported Collaborative Design
Bernadetta Kwintiana Ane, Dieter Rollerand Ajith Abraham (2013). *Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences (pp. 208-241).*
www.irma-international.org/chapter/multiagent-based-product-data-communication/70611

Software Architecture Framework of Healthcare Using Nonfunctional Artifacts
Umesh Banodhaand Kanak Saxena (2020). *Software Engineering for Agile Application Development (pp. 31-50).*
www.irma-international.org/chapter/software-architecture-framework-of-healthcare-using-nonfunctional-artifacts/250436

Design and Evaluation of Automated Scoring: Java Programming Assignments
Yuki Akahane, Hiroki Kitayaand Ushio Inoue (2015). *International Journal of Software Innovation (pp. 18-32).*
www.irma-international.org/article/design-and-evaluation-of-automated-scoring/133112