Chapter 8.3
Activity–Oriented Computing

João Pedro Sousa
George Mason University, USA

Bradley Schmerl
Carnegie Mellon University, USA

Peter Steenkiste
Carnegie Mellon University, USA

David Garlan
Carnegie Mellon University, USA

ABSTRACT

This chapter introduces a new way of thinking about software systems for supporting the activities of end-users. In this approach, models of user activities are promoted to first class entities, and software systems are assembled and configured dynamically based on activity models. This constitutes a fundamental change of perspective over traditional applications; activities take the main stage and may be long-lived, whereas the agents that carry them out are plentiful and interchangeable. The core of the chapter describes a closed-loop control design that enables activity-oriented systems to become self-aware and self-configurable, and to adapt to dynamic changes both in the requirements of user activities and in the environment resources. The chapter discusses how that design addresses challenges such as user mobility, resolving conflicts in accessing scarce resources, and robustness in the broad sense of responding adequately to user expectations, even in unpredictable situations, such as random failures, erroneous user input, and continuously changing resources. The chapter further summarizes challenges and ongoing work related to managing activities where humans and automated agents collaborate, human-computer interactions for managing activities, and privacy and security aspects.

INTRODUCTION

Over the past few years, considerable effort has been put into developing networking and middle-

ware infrastructures for ubiquitous computing, as well as in novel human-computer interfaces based on speech, vision, and gesture. These efforts tackle ubiquitous computing from two different perspectives—systems research and HCI research—hoping to converge and result in software that can support a rich variety of successful ubiquitous computing applications. However, although examples of successful applications exist, a good understanding of frameworks for designing ubiquitous computing applications is still largely missing.

A key reason for the lack of a broadly applicable framework is that many research efforts are based on an obsolete application model. This model assumes that ubiquitous computing applications can support user activities by packaging, at design time, a set of related functionalities within a specific domain, such as spatial navigation, finding information on the Web, or online chatting. However, user **activities** may require much diverse functionality, often spanning different domains. Which functionalities are required to support an activity can only be determined at runtime, depending on the user needs, and may need to evolve in response to changes in those needs. Examples of user activities targeted by ubiquitous computing are navigating spaces such as museums, assisting debilitated people in their daily living, activities at the office such as producing reports, as well as activities in the home such as watching a TV show, answering the doorbell, or enhancing house security.

This chapter introduces activity-oriented computing (AoC) as a basis for developing more comprehensive and dynamic applications for ubiquitous computing. Activity-oriented computing brings user activities to the foreground by promoting models of such activities to first class primitives in computing systems.

In the remainder of this chapter, the section on background presents our vision for activity-oriented computing and compares it with related work. Next we discuss the main challenges of this approach to ubiquitous computing. Specifically, we discuss user mobility (as opposed to mobile computing), conflict resolution and robustness, mixed-initiative control, human-computer interaction, and security and privacy.

The main body of the chapter presents our work towards a solution. Specifically, we discuss software architectures for activity-oriented computing and how to address the challenges of mobility and robustness, as well as the options to model user activities. The chapter ends with a discussion of future directions concerning human-computer interactions, and the tradeoff between ubiquity and security and privacy.

BACKGROUND

The vision of AoC is to make the computing environment aware of *user activities* so that resources can be autonomously managed to optimally assist the user. Activities are everyday actions that users wish to accomplish and that may be assisted in various ways by computing resources in the environment. Done right, AoC will allow users to focus on pursuing their activities rather than on configuring and managing the computing environment. For example, an AoC system could reduce overhead by automatically customizing the environment each time the user wishes to resume a previously interrupted long-lived activity, such as preparing a monthly report, or organizing a party.

To help make this vision concrete, the following examples illustrate possible applications of AoC:

- **Elderly care:** Rather than relying on hardcoded solutions, AoC enables domain experts such as doctors and nurses to "write prescriptions" for the activities of monitoring the health of the elderly or outpatients. Such descriptions enable smart homes to take charge of those activities, collaborating

Related Content

Metrics for Project Management Methodologies Elicitation

Patricia R. Cristaldo, Daniela Lopez De Luiseand Lucas La Pietra (2023). *Perspectives and Considerations on the Evolution of Smart Systems (pp. 187-212).*

www.irma-international.org/chapter/metrics-for-project-management-methodologies-elicitation/327531

Business Model for Mobile Payment in China

Jie Guo, Shahrokh Nikouand Harry Bouwman (2015). *International Journal of Systems and Service-Oriented Engineering (pp. 20-43).*

www.irma-international.org/article/business-model-for-mobile-payment-in-china/126636

Generation and Verification of a Safety-Aware Virtual Prototype in the Automotive Domain

Ralph Weissnegger, Martin Schachner, Markus Pistauer, Christian Kreiner, Kay Römerand Christian Steger (2018). *Solutions for Cyber-Physical Systems Ubiquity (pp. 195-222).*

www.irma-international.org/chapter/generation-and-verification-of-a-safety-aware-virtual-prototype-in-the-automotive-domain/186907

Efficient Utilization of Patterns in Mobile Application Development: An Architectural Model Approach

Jouni Markkulaand Oleksiy Mazhelis (2012). *Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications (pp. 167-192).*

www.irma-international.org/chapter/efficient-utilization-patterns-mobile-application/66467

Open Source Software: A Developing Country View

Jennifer Papin-Ramcharan (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications (pp. 1925-1933).*

www.irma-international.org/chapter/open-source-software/29487