

Chapter 10

Language Engineering for Mobile Software

Engineer Bainomugisha
Vrije Universiteit Brussel, Belgium

Alfredo Cádiz
Université catholique de Louvain, Belgium

Pascal Costanza
Vrije Universiteit Brussel, Belgium

Wolfgang De Meuter
Vrije Universiteit Brussel, Belgium

Sebastián González
Université catholique de Louvain, Belgium

Kim Mens
Université catholique de Louvain, Belgium

Jorge Vallejos
Vrije Universiteit Brussel, Belgium

Tom Van Cutsem
Vrije Universiteit Brussel, Belgium

ABSTRACT

Mobile systems offer the possibility of delivering software services that tightly match user needs, thanks to their availability right at the moment and place where they are needed, and their ability to take advantage of local resources and self-adapt to their environment of use. Alas, writing software for mobile systems is not an easy endeavour. Mobile software construction imposes a number challenges that render existing programming technology insufficient to write such software conveniently. To improve this situation, the authors have taken a language engineering approach. In this chapter they identify the main challenges encountered in mobile software construction and the requirements that rise in the design of programming languages. By way of illustration, the authors present the result of their language engineering experiments—four programming models to ease the construction of software that can cope gracefully with the challenges brought about by mobility.

INTRODUCTION

Hardware technology is ripe for the construction of mobile applications that run ubiquitously, foster peer-to-peer communication and seamlessly adapt their services to changing environments. With

respect to these contemporary hardware phenomena, we observe that programming technology is lagging behind in enabling the construction of applications that naturally support concurrency, decentralised and unreliable distribution, context-awareness and dynamic self-adaptability. These

DOI: 10.4018/978-1-61520-655-1.ch010

aspects are key to fully exploiting the potential that will set mobile systems apart from traditional desktop and server systems. To consolidate the emerging field of mobile software engineering, new programming languages, methodologies and development tools need to be co-designed. In this chapter we concentrate on one piece of the puzzle, namely language engineering for mobile applications. We give a rendition of the experience we have gathered so far in shaping the design space of mobile languages, and describe possible designs along our two principal axes of expertise: Ambient-Oriented Programming (AmOP) (Dedecker et al., 2006; Van Cutsem et al., 2007; Vallejos et al., 2009) and Context-Oriented Programming (COP) (Costanza, 2008; González et al., 2008). AmOP languages ease the construction of software deployed in mobile ad hoc networks because they provide dedicated language features that help the programmer in dealing with the hardware characteristics inherent to those networks. Example features include connections that tolerate temporary network failures (required because of the volatility of wireless network links) and primitives to spontaneously discover nearby services in the local network. COP languages ease the construction of adaptive applications by providing features to support context-dependent behavioural variations. COP treats context explicitly, and provides mechanisms to dynamically adapt application behaviour in reaction to changes in context at run time. The context encompasses all computationally accessible information that describes the current situation, such as device location, battery charge level, and user activity. We use the most representative languages we have developed –AmbientTalk, Ambience, ContextL and Lambic– as case studies on language design, highlighting their strong points and discussing their pitfalls by way of illustration.

CHALLENGES IN MOBILE LANGUAGE ENGINEERING

The hardware properties of the devices constituting a mobile network engender a number of phenomena that have to be dealt with when constructing mobile software. Next we summarise these phenomena, which are inherent to mobile networks and which shape the design space of mobile programming languages.

Volatile Connections

Mobile devices featuring wireless connectivity possess only a limited communication range, such that two communicating devices may move out of earshot unannounced. The resulting disconnections are not always permanent: the two devices may meet again, requiring their connection to be re-established. Quite often, such transient network partitions should not affect an application, allowing both parties to continue their collaboration where they left off. These more frequent transient disconnections expose applications to a much higher rate of partial failure than that for which most distributed languages have been designed. In mobile networks, disconnections become so omnipresent that they should be considered the rule, rather than an exceptional case.

Zero Infrastructure

In a mobile network, devices that offer services spontaneously join and leave the network. Moreover, a mobile ad hoc network is often not manually administered. As a result, in contrast to stationary networks where applications usually know where to find collaborating services via URLs or similar designators, applications in mobile networks have to find their required services dynamically in the

15 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/language-engineering-mobile-software/66466

Related Content

Evaluation of Kernel Based Atanassov's Intuitionistic Fuzzy Clustering for Network Forensics and Intrusion Detection

Anupam Panwar (2016). *International Journal of Software Innovation* (pp. 1-15).

www.irma-international.org/article/evaluation-of-kernel-based-atanassovs-intuitionistic-fuzzy-clustering-for-network-forensics-and-intrusion-detection/144138

Context Inference Engine (CiE): Classifying Activity of Context using Minkowski Distance and Standard Deviation-Based Ranks

Umar Mahmudand Muhammad Younus Javed (2014). *Systems and Software Development, Modeling, and Analysis: New Perspectives and Methodologies* (pp. 65-112).

www.irma-international.org/chapter/context-inference-engine-cie/108811

SQL Self-Adaptive System's Quality Assurance Language

Esmat Maatougui, Chafia Bouanakaand Nadia Zeghib (2020). *International Journal of Information System Modeling and Design* (pp. 78-104).

www.irma-international.org/article/sql-self-adaptive-systems-quality-assurance-language/255113

Landslide Dynamic Monitoring System Based on Unmanned Aerial Vehicle Oblique Photogrammetry and 3D Modeling Technology

Nianhong Wang, Jun Zhangand Meijun Wang (2025). *International Journal of Information System Modeling and Design* (pp. 1-19).

www.irma-international.org/article/landslide-dynamic-monitoring-system-based-on-unmanned-aerial-vehicle-oblique-photogrammetry-and-3d-modeling-technology/392934

Efficient Cloudlet Allocation to Virtual Machine to Impact Cloud System Performance

Lizia Sahkhar, Bunil Kumar Balabantarayand Satyendra Singh Yadav (2022). *International Journal of Information System Modeling and Design* (pp. 1-21).

www.irma-international.org/article/efficient-cloudlet-allocation-to-virtual-machine-to-impact-cloud-system-performance/297630