

## Chapter 16

# Analyzing Mobile Application Software Power Consumption via Model-Driven Engineering

**Chris Thompson**

*Vanderbilt University, USA*

**Jules White**

*Vanderbilt University, USA*

**Douglas C. Schmidt**

*Vanderbilt University, USA*

### ABSTRACT

*Smartphones are mobile devices that travel with their owners and provide increasingly powerful services. The software implementing these services must conserve battery power since smartphones may operate for days without being recharged. It is hard, however, to design smartphone software that minimizes power consumption. For example, multiple layers of abstractions and middleware sit between an application and the hardware, which make it hard to predict the power consumption of a potential application design accurately. Application developers must therefore wait until after implementation (when changes are more expensive) to determine the power consumption characteristics of a design. This chapter provides three contributions to the study of applying model-driven engineering to analyze power consumption early in the lifecycle of smartphone applications. First, it presents a model-driven methodology for accurately emulating the power consumption of smartphone application architectures. Second, it describes the System Power Optimization Tool (SPOT), which is a model-driven tool that automates power consumption emulation code generation and simplifies analysis. Third, it empirically demonstrates how SPOT can estimate power consumption to within ~3-4% of actual power consumption for representative smartphone applications.*

DOI: 10.4018/978-1-4666-4494-6.ch016

## INTRODUCTION

### Emerging Trends and Challenges

Smart devices, such as smartphones and tablet computers have been steadily increasing in capabilities and performance. Modern devices commonly contain multi-core CPUs running at clock speeds well over 1GHz. For example, the Galaxy Nexus from Google contains a 1.2 GHz, dual-core processor with 1GB of RAM and other Android smartphones now have quad-core processors.

The increased performance of smart devices comes at a cost, however. Due to their mobility, smartphones and tablets run exclusively on battery power, so every increase in performance must be balanced with an increase in battery capacity to maximize the time between charges. The design of applications running on the device can also impact battery life, *e.g.*, frequently sampling the GPS or making network requests can consume a significant amount of power.

To balance functionality with power consumption effectively, developers must understand various trade-offs. A key trade-off is between performance and battery life, as well as the implications of the application and infrastructure software architecture on power consumption. Functional requirements, such as minimum application response time, can conflict with power consumption optimization needs.

For example, a traffic accident detection application (White, Clarke, Dougherty, Thompson, & Schmidt, 2010) must be able to detect sudden accelerations indicative of a car accident. To detect acceleration events that indicate accidents, the application must sample device sensors and perform numerous calculations at a high rate. Conflicts occur between the *functional requirements* (*e.g.*, the minimum sensor sampling rate needed to accurately detect accidents) and *non-functional*

*requirements* (*e.g.*, sustaining operations on the mobile device without frequent battery recharging).

Due to complex middleware, Operating System (OS), and networking layers, it is hard to predict the effects of application software architecture decisions on power consumption *a priori*, *i.e.*, without actually *implementing* a design, which makes it hard to analyze the power consumption of design until late in the development cycle, when changes are more expensive (Kang, Park, Seo, Choi, & Hong, 2008). For example, a developer may elect to use HTTPS instead of HTTP to satisfy a security requirement by making communication between the application and server more confidential. It is currently hard, however, to predict how much additional power is consumed by the added encryption and decryption of data without actually implementing the system.

It is also hard to quantify the trade-off between power consumption and security, as well as many other design decisions. Moreover, certain design decisions, such as data transmission policies (*e.g.*, should an application transmit immediately or wait for a specific medium like a Wi-Fi or 3G cellular connection) are hard to analyze early in the design cycle, due to their variability. For example, if an application only sends data intermittently, it may be beneficial to transmit small amounts of data over cellular connections due to the decreased power consumption of 3G cellular connection compared to Wi-Fi (Agarwal, Chandra, Wolman, Bahl, Chin, & Gupta, 2007). If a large amount of data must be sent, however, the total time required to transmit it over 3G may negate the benefit of using the less power consumptive connection. The cellular connection will take longer to transmit the data, therefore, which may in turn consume more total power than the Wi-Fi radio that can transmit the data faster.

24 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/analyzing-mobile-application-software-power/78623](http://www.igi-global.com/chapter/analyzing-mobile-application-software-power/78623)

## Related Content

---

### Towards A Semi-Automatic Transformation Process in MDA: Architecture, Methodology and First Experiments

Slimane Hammoudi, Wajih Alouini, Denivaldo Lopes and Marianne Huchard (2010). *International Journal of Information System Modeling and Design* (pp. 48-76).

[www.irma-international.org/article/towards-semi-automatic-transformation-process/47385](http://www.irma-international.org/article/towards-semi-automatic-transformation-process/47385)

### Analysis of the Evolution of Eight VSEs Using the ISO/IEC 29110 to Reinforce Their Agile Approaches

Mirna Muñoz, Jezreel Mejía and Claude Y. Laporte (2021). *Balancing Agile and Disciplined Engineering and Management Approaches for IT Services and Software Products* (pp. 28-51).

[www.irma-international.org/chapter/analysis-of-the-evolution-of-eight-vses-using-the-isoiec-29110-to-reinforce-their-agile-approaches/259170](http://www.irma-international.org/chapter/analysis-of-the-evolution-of-eight-vses-using-the-isoiec-29110-to-reinforce-their-agile-approaches/259170)

### The Impact of Ideology on the Organizational Adoption of Open Source Software

Kris Ven and Jan Verelst (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1849-1864).

[www.irma-international.org/chapter/impact-ideology-organizational-adoption-open/29482](http://www.irma-international.org/chapter/impact-ideology-organizational-adoption-open/29482)

### A Dynamic Threshold Based Energy Efficient Method for Cloud Datacenters

Shally, Sanjay Kumar Sharma and Sunil Kumar (2020). *International Journal of Software Innovation* (pp. 54-67).

[www.irma-international.org/article/a-dynamic-threshold-based-energy-efficient-method-for-cloud-datacenters/248530](http://www.irma-international.org/article/a-dynamic-threshold-based-energy-efficient-method-for-cloud-datacenters/248530)

### Recommendation System for Hairstyle Based on Face Recognition Using AI and Machine Learning

Yogesh M. Kamble and Raj B. Kulkarni (2024). *International Journal of Software Innovation* (pp. 1-10).

[www.irma-international.org/article/recommendation-system-for-hairstyle-based-on-face-recognition-using-ai-and-machine-learning/309960](http://www.irma-international.org/article/recommendation-system-for-hairstyle-based-on-face-recognition-using-ai-and-machine-learning/309960)