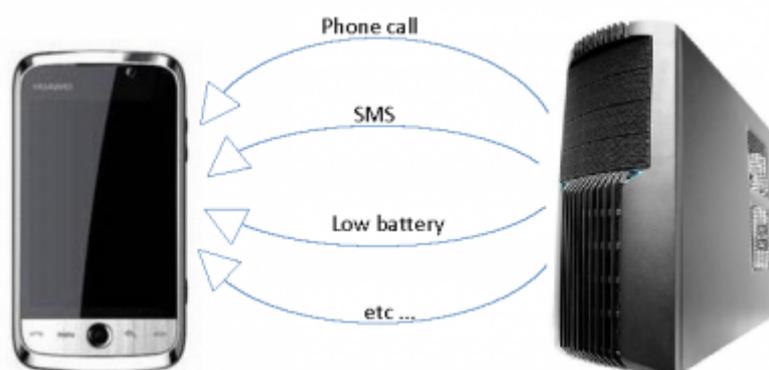


Testing Lifecycle Implementations of Mobile Applications

The increasing performance of modern mobile devices (e.g. mobile phones or tablets) leads to more powerful and complex mobile applications. With the growing complexity of these applications also the complexity of the corresponding application lifecycle grows. The lifecycle of a mobile application covers the period of time from initializing to killing the application. In a lifecycle model the application states (running, paused, stopped, ...) as well as all possible transitions between the states are specified. The transitions are labeled with methods, which the developer can override, to react on state changes of the application (e.g. application gets paused by an incoming call). In practice you often see applications that do not implement the application lifecycle correctly. Data loss and inconsistencies are possible results. To avoid such problems, vendors of mobile platforms (Google, Apple, Sun/Oracle, ...) try to communicate their different lifecycles through a detailed documentation. But there exists no tool to test the lifecycle implementation of a mobile application.

Within the framework of the UMIC Software Quality Initiative this work tries to improve the software quality of mobile applications. In the theoretical part a concept shall be created, how the lifecycle implementation of mobile applications can be tested efficiently. This part of the work is not platform dependent. Based on the lifecycle models resulting from the diploma thesis Analysis of Mobile Application Lifecycles by Corinna Elsemann platform independent (iOS, Android, J2ME, ...) test cases shall be derived (e.g. incoming call or SMS, shutting down due to low battery, ...). For these generic test cases tests shall be specified for the mobile platform Android. In the practical part of this work a test framework has to be implemented, which can execute the specified tests on an Android Emulator as well as on a real Android device. For the realization of this framework, different interfaces to the Android platform have to be used: The Telnet interface shall be used to simulate incoming phone calls, SMS, low battery and connection type switches. The Android Debug Bridge can be used to install an application or to start (concurrently) an existing application. The Android Testing API and the Android Instrumentation Framework shall be used to define JUnit Tests, to log data from the tested application and to simulate user interactions.



The goal of this work is threefold. The first result is to have a methodology how to derive efficiently test cases from an application lifecycle of an arbitrary mobile platform. The second result is to have a catalogue of executable tests for the Android platform. The third step contains the implemented test cases, with which an Android application can be tested.

Status

N. Prakobkosol currently works on this thesis.

Contact

- [Dr. rer. nat. Dominik Franke](#)

From: <https://embedded.rwth-aachen.de/> - **Lehrstuhl Informatik 11 - Embedded Software Laboratory**

Permanent link: https://embedded.rwth-aachen.de/doku.php?id=lehre:abschlussarbeiten:franke_testinglifecyleimplementation

Last update: **2011/05/05 10:44**

