Converting Midlet Navigation Graphs into JML

Research Proposal

Daan de Jong (s0433799; d.dejong@student.ru.nl)
Radboud University Nijmegen, The Netherlands

December 31, 2006

Abstract

As a result of the fast growing mobile (application) market the need for reliable and trusted mobile applications is growing. Telecom operators need the guarantee that software can be trusted and behaves according to its specifications. This research mostly describes a solution for modeling software specifications in such a way that it can be checked using a formal verification tool. In particular investigation is done of the current informal use of flow charts, also known as midlet navigation graphs, as a basis for formal specification that enables rigorious testing and program verification.

Introduction

Mobile devices are developing fast and most of them can already be marked as full grown computers. Together with this development also mobile applications, e.g. mobile phones and PDAs, are becoming more advanced. Most applications (midlets) use Java source code based on MIDP 1.0 or the more recent MIDP 2.0. As a result of this, mobile applications are the next target for misuse of their functionalities. Some viruses and malicious mobile applications already are circulating and their number is growing.
Most applications running on mobile devices are downloaded from the Internet. These applications are running in a sandbox where only a minimal number of functionalities of the mobile device can be used. The sandbox restricts the permissions of the applications. These are quite harmless applications and only a bug in the operating system of the mobile device can be harmful. Mobile applications signed by the telecom operator are granted with all rights. This way these applications can use extra functionality, e.g. sending of sms, to profit from the functionality of the applications. Telecom operators buy these applications from external companies and these applications must be developed according to the specification of the telecom operator. It is obvious that this source code cannot be trusted. Testing of the application and the source code is done according to the international testing standard [tes06] and simply trying all scenarios most often to happen. These tests are rather time consuming and the result is not satisfying. There is a great need for a way to check a given

source code against a given specification in such a way that the source code meets its specification and can be marked as trusted.

Positioning

The department Security of Systems in Nijmegen is specialized in testing and formal verification of (mobile) Java applications. A lot of research has already been done which has resulted in some interesting papers. Because the department is involved in an European research team some interesting papers are provided by other universities and companies joining this team. The most important papers are described below and related to position this research.
Worldwide telecom operators have joined in an organization to set up a standard for testing of mobile Java-based applications. [tes06] describes the procedures for testing a mobile application using MIDP 1.0 and MIDP 2.0. One of the most concrete parts of this paper is the modeling of the properties of the mobile application in a flow chart. These flow charts are still informal and are described in an image. Paper [HO03] describes a prototype to generate JML and Java skeleton code from an UML model and verifying it with ESC/Java. Some problems with this model are the fact that is based on UML 1.3 where UML 2.0 is already available. This prototype does not check the UML against given Java source code but it generates correct JML and Java source code. To confirm the usefulness of the previous prototype paper [EHP03] describes the implementation of a security protocol. This paper gives an example of a security protocol and the generation of the skeleton Java source code. The used JML is not automatically generated from the specification but is done by hand. Paper [Cre06] describes a more formal way of defining formal state diagrams (FSAs) as a model for the specification of a mobile application. It mainly implements the security risks which can occur. This definition is far from finished and needs a lot more research for a complete formal specification of these FSAs.

Problem definition

Checking if a given Java program meets its specification can be done manually. Producing a formal specification (in JML) is most difficult and time consuming. A tool that automates this process and checks if the given source code meets its specification will solve these problems and will guarantee trusted source code. The flow of the target situation is shown in Figure 1. All knowledge of the existing research papers is embedded into this flow chart.

At the start of the flow chart there are the informal specs of the mobile applications. The specifications are not yet in a standard format and are informal represented as an image or in English text. Also at the start of the flow is the Java source code which can be labeled as untrusted. The specification must be modeled as a formal state diagram. This way it can be converted into JML. The next step is to convert the modeled specification into JML. This JML can then be embedded in the given Java source code. The external tool ESC/Java can formally check if the input source code meet the given specification. The output will simply be OK if the source code meets the specification and FAILED if not

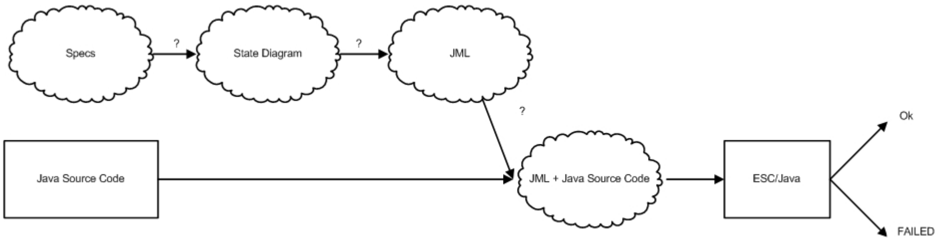

Figure 1: Flow chart

so. The focus of this research wil be on how the informal specs can be expressed in a formal state diagram that can be translated to JML.

Scope and side conditions

As a result of discussions with the supervisor and the currently available papers the scope and side conditions are defined. The most important conditions are:

- ESC/Java is used for formal verification of JML
  ESC/Java is a tool which has proven itself during practical research. Therefor this tool will be seen as trusted.
- JML is used for the formal specifications in the Java code
  JML is simple to use because its syntax is in Java style and therefor known to Java developers. JML in combination with ESC/Java has been proven a useful couple.
- No Java source code will be generated
  In [tes06] Java skeleton source code was generated from the specifications. Because in most practical situations Java source code is delivered from external companies the research will mainly target on given source code.

Research questions

The main research question:

- “Can a given Java midlet and its formal specification be converted to a JML integrated Java application which correctness can be verified by ESC/Java?”

The sub-questions raised from the main research question:

- “Which formal format of state diagrams can be used to express the specification?”
  How can specifications be converted to state diagrams? What is the definition of a state in the model and what means a state change in the model?

- “How can the formal specification be converted to JML in the Java source code?”
  The states of the formal specification must be connected to JML and the source code in such a way that code behavior is analyzed in each state or state change.

- “How can the the generated JML be integrated into the given Java source code?”
  Is the structure of the specification dependable of the method names and variable names or any other type in the Java source code?

Products

The products will be the final result of the research. Dependent on the progress and course of the research the following products are delivered:

- Master thesis
  A written justification of the research period with at least a summary, the problem definitions, used methods/models, fundamentals and the conclusions.

- Presentation
  During the presentation all interested parties will be informed of the results and course of the research.

- Proof of concept
  A working proof of concept of the principle. This will only be completed if enough time is available.

Global approach and planning

The following list gives the global steps which will proceed during the research. These steps and the order of the steps can change during the research period. If so, this list and its associated planning will be updated.

Task	Period
Study variants of state diagrams	October/November
Study JML and its use in Java	October/November
Study midlets and their known specifications	October/November
Specify specs for the state diagrams/JML conversion	December/January/February/March
Proof of concept	April, May
Write the master thesis	June
Presentation	July

Project conditions

During the research period Erik Poll from the department of Security Systems at the University of Nijmegen will be the supervisor and the only other project member. Useful input will be provided by other members of the department of Security Systems during the research. Meetings will be arranged every 2 weeks with the supervisor to discuss the progress and results of the current status of the research. The frequency of the meetings will be higher during the starting period, possibly every week, to prevent misdirecting. All feedback from the supervisor and other interested parties will be done during the meetings and by use of mail.

References

[Cre06] P. Cregut. Midlet navigation graphs. 2006. To appear.

[EHP03] M. Oostdijk E. Hubbers and E. Poll. Implementing a formally verifiable security protocol in java card. *In Proceedings of the 1st International Conference on Security in Pervasive Computing, LNCS*, 2003. To appear.

[HO03] E. Hubbers and M. Oostdijk. Generating jml specifications from uml state diagrams. *In Proc. Forum on specification and Design Languages (FDL '03')*, 2003.

[tes06] *Unified Testing Criteria for Java Technology-based Applications for Mobile Devices*, May 2006.