Model Checking Mobile User Interface Constraints

Kyungmin Lee* Jason Flinn T.J. Giuli Brian Noble Christopher Peplin
kyminlee@umich.edu jflinn@umich.edu tgiuli@ford.com bnoble@umich.edu cpeplin@ford.com
University of Michigan University of Michigan Ford Motor Company University of Michigan Ford Motor
Company

Designing a good user interface (UI) for truly mobile applications is challenging. Such applications can be used while the user is walking, running, or driving. Therefore, the mobile application’s UI must not be intrusive and distracting. Checking UI constraints is difficult and time consuming, because it requires human UI experts to manually examine the application. As third-party applications become more prevalent, there is a need for a more scalable method for checking UI constraints. We present the idea of an automated tool that reports where the UI constraints have been violated. With this tool, third-party developers can easily test their applications without the need for a human expert. Furthermore, these human experts can use the tool to screen out poorly designed applications early and user their time more efficiently on other applications.

In this poster, we focus on model checking UI constraints for applications in a vehicular setting. Many recent vehicles have a human-machine interface (HMI) device, which is an on-board touch screen system that allows the driver to interact with its various applications using touch actions or voice commands. Since, many vehicle companies have recently introduced HMI devices that mirror the smartphone application’s screen and allow the driver to control the application [1, 2], we have decided to create a tool, AMC (Android Model Checker) that model checks Android applications for vehicular UI constraints.

AMC automatically explores Android applications and pinpoints where the vehicle UI constraint is violated. AMC checks for seven UI constraints (e.g., button-size, button-closeness, and contrast ratio), which are common UI constraints specified in the vehicular industry UI guidelines [3, 5]. AMC is designed to be used by application developers and automobile companies. The application developers can use AMC to quickly check and fix violations in their applications during the application development stage. The automobile companies can use AMC to screen out non-guideline conforming third-party applications quickly, which the current manual testing method is incapable of.

To systematically explore Android applications, we employ the model checking technique and infer the finite-state model of the application [4]. A state represents the view on the Android screen that a user would encounter in the application. Each state has various user actions that one can perform, such as back key press, menu key press, button click, etc. When each of these user actions is performed, the Android screen changes from one view to another. In our finite-state model, we model this change as a state transition. A state transition contains the information about the source state, the user action performed in the source state, and the destination state. To avoid the state explosion problem, we use heuristics to collapse multiple isomorphic states into a single state.

Using the model, AMC determines whether an application has violated any of the vehicular UI constraints. AMC terminates when either all user actions have been explored or there are no more reachable states that have unexplored user actions. Due to possible nondeterminism, AMC cannot always revisit its previous explored states, but it makes its best effort to reach there.

We have tested five popular non-vehicular applications across several different categories as well as seven popular vehicular applications. From our experiments, we have found that AMC covers on average 86% of the application’s state space and reports correct violations most of the time. Although there are few false negatives and positives in AMC’s reported violations, verifying them is a much smaller task than verifying the entire application for the violation. By significantly reducing the time and the effort needed to test an application, AMC provides a new highly scalable method for validating Android applications for vehicle UI constraints. Furthermore, we believe that AMC’s techniques can be applied to other applications to test different sets of UI constraints.

References

* Student and will be presenting the poster