SAMA: Static Analysis for Mobile Applications

Vasanth Rajamani and Christine Julien

The Mobile and Pervasive Computing Group, The Center for Excellence in Distributed Global Environments,
The University of Texas at Austin

TR-UTEDGE-2010-013

SAMA: Static Analysis for Mobile Applications

Vasanth Rajamani and Christine Julien
The Mobile and Pervasive Computing Group
The EDGE Center
The University of Texas at Austin
{vasanthrajamani, c.julien}@mail.utexas.edu

New classes of mobile computing applications have been enabled by a tight embedding of devices in the environment. A key characteristic of these applications is that they need to constantly monitor for frequent changes in the environment and rapidly modify their behavior accordingly. Examples of these applications can be found in a variety of domains such as intelligent construction sites, real time traffic monitoring, smart homes and military applications. For example, consider a traffic jam detection application where sensors are placed on cars and traffic lights. Information from these sensors is gathered through an ad hoc network, and the collected information is used to detect traffic jams. A traffic jam can be defined as seeing the same cars stranded for a period of five minutes. If a traffic jam is in fact suspected, the application might want to get more information from the environment and perform other actions (such as finding an alternate route to send to the stranded car).

A key observation in these applications is that they tend to be *reactive*. Application developers typically think in terms of rules that indicate a state change of interest (like stalled cars in the traffic jam example) and write code that adapts behavior if the predicate triggering the rules is valid. Such event-driven applications, where events potentially trigger rules, will become increasingly common in applications developed for mobile environments. The benefit of this style of interaction is that a variety of rules can be defined in a rule engine, and these rules can be used by a large number of applications that employ the same sensing infrastructure.

A significant challenge in realizing such applications lies in the development of the correct rules that can be used by the application. Rules are triggered by events in the environment, and the result of executing a rule can in turn trigger more events. If proper care is not taken, the system could oscillate between a series of rules that prevent the mobile system from performing any other operation. We argue that, for a large class of mobile applications, the ability to verify that the software rules are written in a robust fashion is of critical importance. In previous work [1] we have developed a middleware for mobile ad hoc networks that adapts the mechanism used to collect data in response to environmental changes. For example, our traffic jam application might change the collection protocol from random sampling the network to flooding if there is a suspicion of a traffic jam. Once the suspicion has been confirmed, the protocol can be changed to a location-based one to get relevant information at a reduced cost.

Our poster will highlight the key features of our tool, Static Analysis for Mobile Applications (SAMA), that can be employed to statically analyze the behavior of rules specified for an event-based application. SAMA builds on static analysis techniques from software engineering and active databases to ensure that the code running in a mobile application is safe. A poorly written application can end up requesting information continuously from the mobile devices draining them of their precious resources. The key features of SAMA are as follows:

- Termination Detection: SAMA contains algorithms to automatically ensure that the set of specified rules does not conflict. Consider a rule specifying that on detecting a traffic jam, the middleware should switch to a location-based protocol. If another rule specifies that the presence of a particular number of cars in a location should ensure that the middleware continue with a random sampling protocol, the middleware may potentially never switch from the location-based protocol. Other rules that rely on collecting information from other areas may never get triggered as a result.
- Ordering Errors: Rules can be written such that the order of their execution can produce non-deterministic behavior. SAMA contains algorithms that automatically flags to the user those rules where the order of precedence matters. The user can then decide the precedence relationships. If the order of execution does not introduce non-determinism, the programmer can avoid spending time analyzing the relationship between the rules and focus on the nature of the rules themselves.

Using these and other verification techniques borrowed from software engineering, SAMA will ensure that the code running on mobile applications is safe and correct.

HotMobile '10, Feb. 22-23, 2010, Annapolis, M.D.

1. REFERENCES

[1] V. Rajamani and C. Julien and J. Payton and G.-C. Roman, "PAQ: Persistent Query Middleware for Dynamic Environments", Middleware, 2009.