Third Eye — Specification-Based Analysis of Software Execution Traces

Raimondas Lencevicius Alexander Ran Rahav Yairi
Nokia Research Center, 5 Wayside Road, Burlington, MA 01803, USA
Raimondas.Lencevicius@nokia.com Alexander.Ran@nokia.com Rahav.Yairi@nokia.com

In the Third Eye project we have defined a methodology for tracing execution of software by reporting events meaningful in the application domain or essential from the implementation point of view. Many ideas incorporated in the Third Eye framework were inspired by the Logic Assurance system [1]. The implemented portable prototype of the Third Eye framework includes reusable software components for event definition and reporting and stand-alone tools for storage and query of event traces, constraint specification and trace analysis.

A central decision of Third Eye framework is what information from the execution state of the program is traced. We decided to trace occurrences of events. “Event” in this case is a qualitative change in the state of an entity either meaningful in the application domain or significant architecturally. In Third Eye framework, events are typed objects. Event type has a name, a list of named and typed properties, and the type constructor. Third Eye event types are similar to classes in programming languages although the only method associated with the event type is its constructor. We allow event type inheritance. Events in Third Eye are characterized by the time and location of their occurrence.

Correct behavior specifications define constraints on the properties of the events, their sequence, location, and timing. We use formal descriptions of the constraints between events to identify violations in execution traces.

Another concept of Third Eye is the tracing state. Tracing state is a set of event types generated in that state, other event types are filtered out and not reported. The system is always in a specific tracing state. Tracing states correspond to specifications. A program specification describes a set of constraints on events. The event types used in a specification have to be monitored to validate a trace against this specification. All event types contained in a specification and monitored for this specification form a tracing state. Tracing states also control the overhead of tracing on the executing system.

The Third Eye framework includes modules for event type definition, event generation and reporting, tracing state definition and management, trace logging, query and browsing interfaces. Modules of event type definition, event reporting facility and tracing state controller are integrated with the software of the system under trace (SUT). The rest of the modules are independent from the SUT and can be deployed on a different execution platform to minimize the influence on system performance. Trace delivery for logging and analysis uses alternative interfaces to accommodate devices with different data storage and connectivity capabilities.

We have implemented Third Eye framework prototype currently used by the Third Eye project team in collaboration with product development teams in Nokia's business units. We used Third Eye to test a number of software systems: the memory subsystem of one of Nokia's handsets, Apache Web Server, and WAP (Wireless Application Protocol) client. WAP is an industrial standard for applications and services that operate over wireless communication networks. We validated message sequences in this protocol by adding events in the functions that correspond to the protocol primitives and then checking whether the event sequence corresponds to the protocol message sequence. Events are mapped to Prolog facts and constraints are expressed as Prolog rules.

Third Eye can be used for debugging, monitoring, specification validation, and performance measurements. These scenarios use typed events—a concept simple and yet expressive enough to be shared by product designers and developers. The Third Eye has an open architecture allowing easy replacement of third-party tools, including databases, analysis and validation tools. Third Eye is a practical framework for specification-based analysis and adaptive execution tracing of software systems.