A dynamic program analysis provides essential information during later phases of an application development. It helps with debugging, profiling, performance optimizations or vulnerability detection. Despite that, support for creating custom dynamic analysis tools, especially in the domain of managed languages, is rather limited.

In this thesis, we present two systems to help improve application observability on the Java platform. DiSL is a language accompanied with a framework allowing simple and flexible instrumentation for the dynamic program analysis. DiSL provides high level abstractions to enable quick prototyping even for programmers not possessing a knowledge of Java internals. A skilled analysis developer gains full control over the instrumentation process, thus does not have to worry about unwanted allocations or hidden execution overhead.

ShadowVM is a platform that provides isolation between the observed application and the analysis environment. To reduce the amount of possible interactions between the analysis and the application, ShadowVM offloads analysis events out of the context of the application. Even though the isolation is the primary focus of the platform, ShadowVM introduces a number of techniques to stay performance comparable and provide a similar programming model as existing dynamic analysis frameworks.