Raimondas Lencevicius

Statement of Research

My primary research interests are in the area of runtime analysis and transformations of software systems, as well as software tools and advanced user interfaces. I believe that runtime program analysis can provide significant insight into program execution patterns and can be used to understand and transform the behavior of programs. Efficient implementation of such analyses and transformations is a challenging problem.

As a Ph.D. thesis project, I have developed a query-based debugger that allows programmers to ask questions about object relationships during program runtime. Finding groups of objects satisfying given constraints helps users to understand the program’s runtime behavior and to find objects used incorrectly. For example, assume that a graphical widget object in a GUI program references its parent window, and that this parent window must in turn reference the enclosed widget. With my system, programmers can find faults in this data structure using a simple query. To determine exactly where and when errors occur, the system reevaluates query results after every assignment or object creation that can change the query result. Consequently, this dynamic query-based debugger indicates object relationship failures as soon as the error occurs. Even when a program such as javac Java compiler modifies its data structure (AST) in several different places, the debugger will find the exact instruction that violates the user constraint. To speed up the dynamic query evaluation, the debugger (implemented in portable Java) uses a combination of program instrumentation, load-time code generation, query optimization, and incremental reevaluation. Experiments and a query cost model show that most queries are practical especially when query evaluations are infrequent or when query domains are small.

Query-based debugging could be extended for debugging distributed and parallel object-oriented systems. Though a distributed query can be evaluated by atomically collecting the state of all nodes in the system, the evaluation efficiency can be improved by relaxing atomicity requirements and finding the consistency criteria for obtaining correct answers to queries involving distributed, concurrently changing objects.

I believe that insights gained while implementing the dynamic query-based debugger can be extended to a wide range of applications. A natural extension of the debugger would be the analysis of mobile Internet programs. Here, in addition to handling distributed objects, the program would maintain a connection between the debugger and the migrating program. Different strategies to achieve this connection can be explored.

Since the debugger implementation provides an effective way to gather runtime program information, my research could be extended to incrementally change large data collections and programs handling such collections, such as digital libraries, data warehouses, Internet commerce services, and virtual communities. For example, a distributed commerce server could be reconfigured depending on usage patterns and network conditions.

I believe that runtime analysis of the systems described above allows us to better understand them, to correct errors and inefficiencies in their implementations, and to suggest new ways to improve them.

The research in these areas would greatly benefit from a collaborative study of user interfaces of runtime support tools. My experience shows that interfaces have to interact closely with the underlying tool to provide efficient and clear presentation for the user. Information filtering,
abstraction, compression and visualization would increase the utility of the underlying tools and would provide innovative user interfaces.

With my experience in the areas of debugging, program runtime analysis and software tools, I am interested in pursuing research in related fields such as advanced software tools, intelligent user interfaces, and program analysis, transformation and visualization.