Research Statement
Kristian G. Kvilekval

My research is focused on mobile distributed and adaptive systems. Beginning with an initial interest in how software could be made to adapt to runtime conditions, I was drawn to the area of mobile systems where very real problems were being experienced due to finicky communication and resource starved platforms.

Advancements in wireless communication and hardware miniaturization have made mobile computing a reality. However, our software systems are still rooted in fixed computing. Platform mobility has also enabled new styles of ad-hoc collaborative applications. While the hardware has made these applications possible, our software systems for building collaborative applications have remained pitifully behind. In particular, ad-hoc groups form what may be seen as small dynamic distributed systems. Example applications for this technology include mobile GIS for researchers and emergency services where new information needs to integrated and propagated to a changing set of users. Building robust distributed systems for computers over fixed networks has been the focus of years of study. These mobile dynamic distributed systems with rapid configuration changes and uncertain communication make designing these applications out of reach. My research has focused on enabling software technologies for these new systems.

My primary focus was on the problem of disconnected data access and prefetching for mobile systems. Small systems with wireless LAN links can often become disconnected from the network as they are moved. These disconnections can last anywhere from fractions to millions of seconds. While disconnected, remote data becomes completely unavailable to the mobile platform, often stopping progress or making the device unusable. Smoothing these bumps has been one area of my thesis.

In my research, I combined program analysis with runtime systems to estimate resource requirements of programs. I applied shape-analysis which had previously been used to remove unnecessary locking from programs to prefetching of data. Shape-analysis produces a set of graphs summarizing the actions of a method and all of its submethods. This summarization can be used to follow the expected data manipulations of program. I also applied these summarizing graphs to the problem of lock and lease scheduling. In databases, lock scheduling is an important area of current systems research. Even with complete future knowledge, optimal resource scheduling is NP-hard. However, techniques exist that can approximate to a large degree optimal scheduling. In most real systems, future knowl-
edge is difficult if not impossible to discover. My research investigated how shape-summarization graphs could be used to determine the future locking behavior and the future execution time of object-based programs.

I demonstrated these techniques in an object-oriented middleware system for building distributed applications. The system was based on Java and written in Java, but provided extensions to make distributed applications less daunting. At the language level, we included nested transactions, asynchronous methods, and guarded (atomic) actions. At the system level, we provided prefetching, replica distribution, and optimistic(strict) consistency.

Future Work

In disconnected database systems, determining the best way to resolve conflicting changes is an extremely difficult problem. In a general system, the number of possible conflicts rises exponentially with the number of participants. However, preliminary work suggests that program analysis can be used again to assist in automatic reconciliation of conflicting versions.

I am also interested in studying foundations of peer-to-peer computing. Large distributed systems have become a reality in the peer-to-peer movement. While current systems provide simple read-only data models and very few guarantees, future applications will include large scale gaming, virtual communities, and information distribution. I would like to explore and improve P2P architectures to provide database services. Also to that end, event-based computing is an attractive communication model for programs that are disconnected both in time (events are delivered possibly out of order) and space (events are passed over overlay networks). I am interested to see whether the model can be applied to database systems and what basic properties these systems will have.

Eventually, I would like to return to work in compilers and runtime systems, blurring the edges surrounding both. Programs might eventually optimize for current work load and parameters while maintaining enough high level program intent to de-optimize and re-optimize when parameters have changed.

Finally, I am convinced that some form the Whorf hypothesis ("Language shapes the way we think, and determines what we can think about") is evident in programming languages. I am interested in language design and the trade-offs that must be addressed. While the domain specific language have achieved some measure of success, new technologies such as Metaobject protocols (open compilers) and Aspect-orient programming also promise greater flexibility.