

Research Statement

Konrad Lorincz
School of Engineering and Applied Sciences
Harvard University
konrad@eecs.harvard.edu
<http://www.eecs.harvard.edu/~konrad/>

My research interests include several aspects of designing and building distributed systems. I am a strong advocate of applying good software engineering techniques, such as well defined APIs, modular design, and early end-to-end system integration and testing, which I believe are vital for the success of building large distributed systems. I have successfully applied these principles in my thesis research titled, *Resource Aware Programming in Sensor Networks*, which is focused on how to design sensor network applications that are data-intensive and able to gracefully adapt to fluctuations in resource availability and load. To address these challenges, I led the design of Pixie, a resource-aware OS for sensor networks in which resources are treated as a first-class entity. This work was motivated from my earlier projects, which include Volcano monitoring, CodeBlue, and Mercury, a wearable sensor network platform designed for high-resolution motion capture in a home setting. In addition to my thesis project, I also developed MoteTrack, an RF-based location tracking. This system has received considerable interest from companies and universities and has been deployed at several institutions, including Boston University, Crossbow, and Antotech.

Research Approach

My approach to research is to build real working systems, not just research prototypes, and evaluate them in a realistic setting. Research prototypes often make unrealistic assumptions, which often lead to incorrect or optimistic conclusions. When tackling a new research problem, I think it is vital to first identify the challenges and understand the requirements. The next step is to define the APIs as concretely as possible and then build and grow the system “organically”, i.e. build a bare-bones end-to-end prototype first. I believe in building a system out of self-contained modules, which should be individually tested. However, it is crucial that the entire system be integrated and tested early on and for the remainder of the development cycles.

I am an advocate of releasing source code to the public. Doing so fosters collaboration and feedback, which improves the system. Providing good documentation is key for the adaptation of the system and for it to have an impact in the real world. I have followed this philosophy of releasing software along with documentation for all my research projects at Harvard, including MoteTrack, CodeBlue, Pixie, and Mercury.

Resource Aware Programming

A large part of my work as a graduate student involved designing and building software for Wireless Sensor Networks (WSN). One of the main challenges with such systems is the fundamental resource-limited nature of sensor nodes. In addition, unlike conventional WSNs, which focused on low-data rate applications, most of my work has involved applications that are data-intensive and are subject to variation in load and resource availability (e.g. variation in bandwidth due to mobility and variation in lifetime due to data load). These challenges have motivated my colleagues and I to develop Pixie, a sensor network operating system that addresses these concerns.

Pixie is a new operating system for sensor networks that enables resource-aware programming, a model in which applications receive feedback on, and have explicit control over resource usage. Pixie is designed to support the needs of data-intensive applications. These applications, which include high-resolution monitoring of acoustic,

seismic, acceleration, and other signals, involve high data rates and extensive in-network processing. Given the fundamentally resource-limited nature of sensor networks, a pressing concern for such applications is their ability to receive feedback on, and adapt their behavior to, fluctuations in both resource availability and load.

The Pixie OS is based on a dataflow programming model and is based on the concept of resource tickets, a core abstraction for representing resource availability and reservations. By giving the system visibility and fine-grained control over resource management, a broad range of policies can be implemented. To shield application programmers from the burden of managing these details, Pixie provides a suite of resource brokers, which mediate between low-level physical resources and higher-level application demands. Pixie is implemented in NesC and supports limited backwards compatibility with TinyOS.

Wearable Sensor Platforms for Medical Monitoring

Mercury

A significant part of my dissertation work focused on designing a platform for long-term high-resolution capture of sensor data for patients in a home setting, called Mercury. Mercury faced many of the challenges that Pixie was designed to address, and, as a result, the latest version of Mercury is implemented in Pixie.

Mercury is a wearable, wireless sensor platform for motion analysis of patients being treated for neuromotor disorders, such as Parkinson's Disease, epilepsy, and stroke. In contrast to previous systems intended for short-term use in a laboratory, Mercury is designed to support long-term, longitudinal data collection on patients in home settings. Patients wear up to eight wireless sensors equipped with MEMS accelerometers and gyroscopes that record continuous high-resolution movement data. Individual nodes compute high-level features from the raw signal, and a base station in the patient's home performs opportunistic data collection based on energy availability and radio link quality. Mercury is designed to overcome the core challenges of long battery lifetime and high data fidelity for long-term studies where patients wear sensors continuously for 12 to 18 hours a day. This requires tuning sensor operation and data transfers based on energy consumption of each node, processing data under severe computational constraints, and autonomous operation and failure recovery.

An earlier version of the Mercury platform is currently being used in several patient studies at the Motion Analysis Laboratory at Spaulding Rehabilitation Hospital in Boston. To date, data has been collected from six patients over several months. Four of these patients are involved in a study of deep brain stimulation (DBS) for Parkinson's Disease. We have undertaken 8 data recording sessions for each patient over a 3-month period, in which each patient wears 9 SHIMMER sensors and performs a series of predetermined tasks while their DBS parameters are adjusted. The sensor data is being used to build predictors of the severity of Parkinsonian symptoms and to gain an understanding of how different DBS parameters affect those symptoms.

The two remaining patients are undergoing treatment for epilepsy. Mercury has been used to capture up to 12 h of accelerometer and electromyogram (EMG) data per day for a 5-day period for each patient. The specific number and placement of the sensors varies depending on the nature of the seizures.

CodeBlue

I have also been involved in a related project, in which wireless sensor network technology is used to improve medical care in pre-hospital, in-hospital, and disaster response. This platform, called CodeBlue, provides protocols for device discovery, publish/subscribe multihop routing, and a simple query interface. Several medical sensors may be used including pulse oximeter, ECG, EMG, blood pressure, temperature, and accelerometers for detecting motion. These sensors are attached to motes, which communicate with PDAs, PCs, and other devices that may be used to monitor and treat patients. My colleagues and I deployed CodeBlue in a disaster drill in August 2006 in Baltimore, MD, in collaboration with AID-N team at Johns Hopkins University and rescue workers from Montgomery County Fire and Rescue Services. In addition, I took part in a pilot study at the

Washington Hospital Center Burn Unit, which demonstrated that CodeBlue is capable of transmitting data inside a radio-interference-rich critical setting. Part of my work also involved integrating CodeBlue with MoteTrack, a system that I built earlier, which allows for tracking the location of patient devices indoors and outdoors.

Monitoring Volcanic Activity

Wireless sensor networks have the potential to greatly enhance our understanding and ability to monitor volcanic activity. Unlike traditional volcano monitoring equipment, which is heavy, expensive and power-hungry, wireless sensor networks are small, low cost, and have low power-requirements. This technology permits volcanologists to deploy a large number of sensors, which provide a much greater spatial resolution and richer dataset than previously feasible.

I was involved in the design and deployment of the second deployment, which took place at Vulcan Reventador, Ecuador, in July/August 2005. This deployment consisted of 16 nodes deployed for over a span of 3 km on the upper part of the volcano for a duration of 19 days. The sensor nodes collected both seismic and infrasonic signals with 24-bit resolution per channel at 100 Hz. In order to validate that our wireless sensor network can collect scientifically meaningful data, our colleagues from UNC, UNH, and IG-PEN deployed 3 existing volcano instrumentations. Our goal was to demonstrate that WSN are an effective scientific instrument, which we published in OSDI'06.

Location Tracking

My earliest work at Harvard, focused on developing a robust decentralized RF-based location tracking system targeted at mote-class devices. This work was motivated by the observation that localization is a key requirement to a large number of applications and at the time virtually no-such system for motes existed. The system, MoteTrack, runs entirely on motes and requires no wired infrastructure. With just 25 beacon nodes deployed throughout one floor, the system achieves a 50th-percentile and 80th-percentile location-tracking accuracy of 1 meters and 1.7 meters respectively when diversifying the radio signal over 16 frequencies. In addition, MoteTrack can tolerate the failure of up to 60% of the beacon nodes without severely degrading accuracy, making the system suitable for deployment in highly volatile conditions. This project has had a significant impact in the sensor network community and received a lot of interest from companies and other institutions, with daily downloads continuing to this day.

Future Agenda

One of my future research interests is network-wide resource awareness. So far, Pixie focuses on node-level resource management. An important next step is to extend Pixie to involve coordinated resource management of sensor nodes across the network. A range of algorithms and protocols have been proposed for congestion control, network topology adaptation, coordinated duty-cycling, and other behaviors. One approach is to allow such policies to be expressed through a unified programming model that can permit non expert WSN programmers, such as domain scientists, to build adaptive, efficient, and self-tuning sensor networks.

I am also interested in extending the Mercury platform to support a range of heterogeneous devices. For example, one goal is to support a wearable base station, such as a cell phone, iMote2, or PDA that can be used to collect and process sensor data on the body itself without requiring an external base station. This will require careful balancing of computation and communication within the network to ensure acceptable battery lifetimes. Another goal is to ensure that Mercury is suitable in related clinical applications in home and hospital settings for monitoring patients being treated for epilepsy, chronic obstructive pulmonary disease, and stroke. Experience gained through our work with Parkinson's disease patients will inform our future developments on the platform.