Research Statement for Neil Klingensmith

My research focuses on building intelligent, connected cyber-physical and autonomous and devices for a world in which computers have become an integral part of daily life. My work focuses on encapsulating computers in everyday devices and designing the algorithms to control and coordinate actions that were previously not possible or required human intervention. In collaboration with other graduate students, I have developed real-time and power efficient software for mobile and IoT devices that has increased the efficiency and reliability of deployed cyber-physical systems. This work has applicability to a broad range of deployment scenarios including IoT, mobile, and building management devices. My work on distributed systems for building management has made it possible to coordinate control actions within homes and commercial buildings, conserving resources and making the environment more hospitable for the people who use it. As a graduate student, I have received nearly $200,000 in external funding for my work.

As a graduate student, I've engaged with the campus community at UW-Madison to design systems that solve real-world building management problems, saving the school tens of thousands of dollars per year on operations costs. This same project was also used as the basis of a startup called Emonix to implement these ideas on a larger scale. In that context, many other non-university buildings have been outfitted with Emonix controllers throughout the United States.

1 Current Research: IoT Systems and Software

My current research focuses on building connected IoT systems, algorithms, and software for automation and control. My approach has been to use multiple data streams from different sources to infer context and attempt to take global optimization actions.

IoT Software: The path of innovation in IoT software has been forged largely by server and workstation developers. Code developed for larger systems is downsized and optimized to accommodate embedded computing platforms. While this technique can be useful, it also has the potential disadvantage of producing software that is not well-suited to the end application. The IoT software I have developed has been designed with the unique challenges and objectives of IoT and embedded systems in mind.

My work on energy-aware real-time sampling (eEnergy 2018) allows IoT systems to conserve energy by reducing sampling frequency during predicted periods of inactivity. My algorithm, designed specifically for resource-constrained embedded devices, allows the nonuniformly sampled signal to be transformed to the frequency domain for processing with standard convolution filters, just as if it had been acquired in uniform sampling mode. This technique reduces the overall energy consumption of the sampling and conversion process by more than 18%, and it is widely applicable for IoT devices.

My lightweight real-time hypervisor (Hermes[1], HotMobile 2018) allows embedded time-sharing operating systems to coexist with fast, responsive bare-metal code on the same CPU. This permits multiple compute and I/O tasks to share resources without compromising latency-sensitive data acquisition and actuation software. Hermes is the first hypervisor ever written for computers without an MMU, and it has to deal with the unique challenges of running on hardware that was not designed for use with a hypervisor. While hypervisors were originally developed for use on servers and workstations, Hermes was built to address the unique challenges of embedded and IoT applications.

Building Automation Systems: Automated building resource management, traditionally available only in large commercial facilities, has recently become a domain of interest for IoT engineers. The goal of such systems is to design practical hardware and algorithms to coordinate and control disparate building systems such as heating and cooling (HVAC), electrical, lighting, plumbing, etc. The challenge is to share data collected in buildings and use it to synergistically improve building function, efficiency, and comfort.

[1] http://hermes.wings.cs.wisc.edu

My work has focused on making these systems available not just on a large expensive commercial scale, but also for residential and small office settings. I designed a low-cost energy monitoring platform (BuildSys 13) that could be installed cheaply and quickly in any building to gather information about electricity consumption. From that platform, my research group gained many useful insights about resource consumption patterns within a building.

I designed an automated zone control system for residential heating and cooling systems (eEnergy 14). While widely available in large commercial buildings, zoned HVAC is not common in individual residences, largely because it is normally implemented by replicating the heating and cooling equipment (i.e. one furnace for each zone). This makes zoned HVAC prohibitively expensive for all but the largest homes. My zoned system allows individual rooms to be independently controlled with low-cost internet-connected dampers and fans. Not only does it improve occupant comfort by making sure all spaces are heated and cooled to the appropriate temperature, it also can reduce energy consumption by only heating and cooling the occupied spaces within a home. My zoned HVAC system was connected to the low-cost electricity monitoring platform to measure and globally optimize energy consumption of the HVAC system.

I designed a water quality management system (BuildSys 14, BuildSys 15, BuildSys 16) that can monitor metrics of water quality and respond in real time. This was the first cloud-connected water quality management system. By outfitting water treatment systems with sensors and using the data to control them, my research group saved many buildings on the University of Wisconsin campus thousands of dollars per year in labor and chemicals. We also were able to identify failures in the water treatment system, and respond quickly, which was not possible before.

I spun the water quality management project out into a startup called Emonix[2] that collectively raised $75k in seed funding and has sold many units throughout the country.

2 Future Research Agenda

Toward User-Controlled Privacy in IoT Devices: Data from IoT devices is particularly vulnerable to intrusion because it cannot be curated by users in the same way as a social media feed. Furthermore, because IoT devices often operate autonomously, we may forget that they are transacting in troves of our personal data, much of which is of a far more intimate nature than the cat pictures and status updates we share on social networks. This problem is compounded by the fact that users are increasingly interacting with multiple mobile and IoT devices. When, as many predict, we are routinely interacting with hundreds of autonomous devices, the problem of managing the privacy settings for each will be intractable.

In this work we will think about how users can specify high-level privacy policy for all IoT devices that transact with their data. We will think about ways that users can express policy in simple, general terms, and how to translate a simple user-defined privacy policy into specific privacy settings on individual devices. We will define ways of implementing the privacy settings on devices that may not have a trusted API or for controlling privacy settings.

We will also investigate ways of analyzing data handling in IoT devices and their cloud-based services. Our goal here is to understand how data collected in IoT devices is processed within the device and transmitted to the cloud service. Data handling analysis would help us detect the misuse of user data and automatically adjust the user's privacy settings to correspond with the privacy policy. For example, suppose a user prefers that video data colletced by his IoT devices is not shared with cloud services. Using dynamic taint analysis, we could write software to track video frames as they pass from I/O device to main memory to the network.

Understanding and Improving Performance in Real-Time Virtualized Systems: The Hermes hypervisor for IoT systems presents several interesting research questions. Performance characteristics, sources of latency, bottlenecks, etc. are well-understood in server-based virtualization systems (although this knowledge is largely held by the developers of virtualization software). Since real-time virtualization is such a new domain, we do not know much about how various software design choices affect performance. For instance, if we have one or more real-time operating systems running as guests in a hypervisor, how do the various scheduling policies of the RTOSes interact with the scheduling policy of the hypervisor? Can we design scheduling policies for the hypervisor that optimize performance of application-level code by some metric (fairness, latency, etc)? What metrics should we use to characterize performance in such an environment?

I also aim to develop some applications to run on top of Hermes to understand the benefits and drawbacks of using a hypervisor in a real-time environment. A few of these are already understood (somewhat). The hypervisor, for example, can permit applications that require low-latency to run in tandem with others that have less stringent requirements more efficiently than a real-time OS. There are likely other advantages (and disadvantages) to using a hypervisor. I would like to study the Hermes hypervisor in the wild to learn about some of these tradeoffs and how to use them in real applications.

[2] `http://emonix.io`