2.9 Programming and Platforms (PRO)

CENS systems research strives to advance the state of the art in innovative in situ observing systems. To this end, our research efforts have focused on two critical themes that cut across our broad range of observing systems: the development of practical tools and platforms, and the design and evaluation of architectures and programming systems. Together, these two research directions will enable the Center to field sophisticated, rapidly reconfigurable, multi-user observing systems that support advanced sensing modalities.

Tools and Platforms for Observing Systems

A primary systems research thrust is the development of mature tools and platforms for deployments of observing systems. Our platform efforts have revolved around three activities: participatory sensing mobile to web platform mechanisms, the power aware LEAP platform, and time synchronization.

Participatory Sensing Platforms: The first are a suite of building blocks needed for the construction of robust Participatory Sensing systems. The growing real-world application experience with PS systems has clarified a set of technology challenges that span the three major components of participatory sensing systems: (1) data collection on handsets and other devices, (2) cloud-based processing, and (3) user interaction and system management on the web, as well as the underlying data models and cross-cutting security and privacy issues.

On the mobile devices themselves, research continues to better understand *handset usage models*, especially with respect to power consumption, where improved knowledge will support the goal of being able to continually run participatory sensing applications on everyday handsets. Additionally, we continue to explore embedding local processing to tighten the feedback loop with users. In general, as our deployments expand, we expect to dedicate more effort to user interface improvements and usability study.

On the data campaign management side, approaches to analyzing spatiotemporal patterns of participants are being developed to support semi-supervised *campaign recruitment*—the matching of users' availability and interests with data collection needs, as well as *path planning and automated sample requests* once participants have signed up for a collection campaign, and *performance metrics and incentive mechanisms* for managing and improving long-term participation. *Rapid campaign deployment and management tools*, borrowing from the experience of mobile crisis response systems like Ushahidi, will enable new data collection campaigns to be quickly created and deployed through the assembly and customization of a pre-existing web services and user interface components. Additionally, through collaboration with other institutions, CENS is exploring how to integrate other emerging platforms and standards for data collection, such as Open Data Kit (ODK).

As noted in the project details of this report, many participatory sensing applications combine continuous time-location "traces" gathered using mobile handset GPS and/or cell tower data combined with intermittent samples in other modalities. *Activity and place classification*, the generation of higher-level, semantically meaningful features from this basic participatory sensing data will reduce the need for participants to continually classify their own data manually, enabling them instead to make use of semantically-meaningful location and mobility data that is automatically defined in terms of place and path instead of coordinate points and time series. Handling of time-location traces themselves is being improved in applications like PEIR by using a path model rather than simply storing a collection of points.

The Spotlight project aims to develop an affordable, easy-to-use resource monitoring system that monitors fine-grained resource consumption in buildings to provide feedback to individuals about their life habits and impacts. The main goal of the project is to provide general users with an easy means to monitor their own resource consumption in their spaces. This year we developed an affordable easy-to-use appliance level power monitoring system by exploiting the fact that appliances emit measurable signals when they operate. Since appliances typically emit measurable signals when they are consuming energy, we can estimate their consumption using indirect sensing (Figure 7). The project team developed a fine-grained power monitoring system that furnishes users with an economical, self-calibrating tool that provides power consumption of virtually every appliance in buildings. The principle of operation is a network of wireless distributed sensors monitoring signals that appliances emit and forwarding them to a personal computer acting as a back-end fusion center. The fusion center collects data from the indirect sensors and measurements from the main power meter, and runs a model-based machine learning algorithm that automatically learns and estimates power consumption of every appliance on-the-fly.

Low Power Energy Aware Processing Platform (LEAP): In LEAP we have created a state of the art seismic data acquisition system based upon the successful second-generation low power, energy aware processing (LEAP) platform. This technology is now incorporated in the Reftek RT-155 platform from Refraction Technology Inc. in the form of the RT619 module that includes a micropower FPGA with LEAP's Energy Management and Preprocessing (EMAP) module and a a new time synchronization module that provides sub-microsecond time accuracy from both

GPS and network timing sources. The LEAP enabled RT-155 is currently operating in field exercises with L-4 seismometers as well as acoustic microphones.

Time Synchronization for Sensing Platforms: High quality time information is crucial in embedded sensing networks, but is difficult to achieve in the presence of jitter in computational and communication latencies, time interval between resynchronizations, accuracy of time stamping network wireless packets, and quality of local clock source (quantization, frequency tolerance, aging, and drift). In Time Synchronization—Take 2 we have re-evaluated two aspects of time synchronization – sources of error in time synchronization and how they affect synchronization accuracy and energy consumption, and post-facto synchronization using sensor data. Based on that, we have developed three new mechanisms that collectively provide robust, stable, and high resolution time information at low energy costs: Temperature Compensated Time Synchronization (TCTS), Virtual High Resolution Time (VHT), and Data Driven Time Synchronization (DDTS).

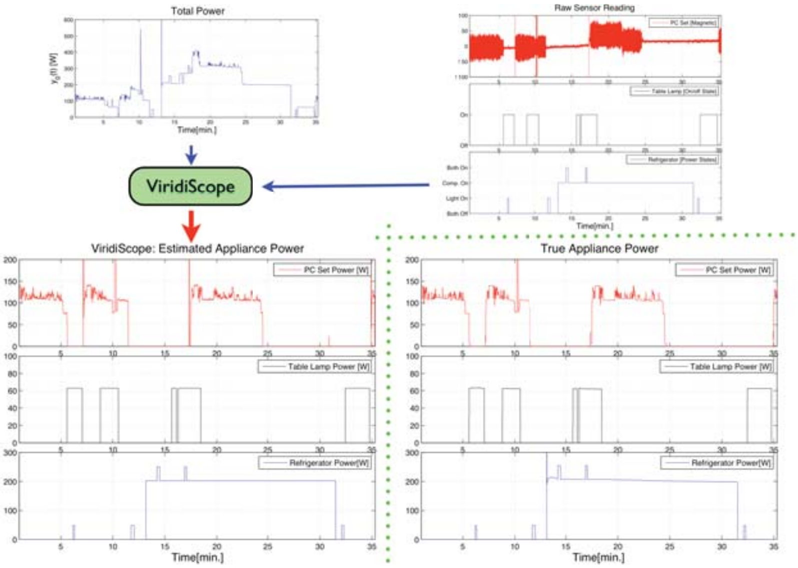

Figure 7. Non-intrusive Appliance-level power monitoring system evaluation

Architecture and Programming Systems

Taking a sensing system that works in a small scale lab setting out into the real world might seem to be a conceptually simple step. We have repeatedly found to that this is not so; the effort to deploy a long-running reliable ENS observing system is currently perhaps an order of magnitude more than that of writing the application itself. Our ongoing research will change this: by developing visibility into a deployed system, by developing languages and tools that enable robust application development, and by re-architecting ENS software to ensure the development of manageable software, our research will enable more nimble observing systems.

Lowlog: Observing the internal execution of severely resource constrained wireless embedded devices remains a critical block to widespread adoption of embedded wireless sensing systems. Bandwidth limitations, constraining both data transfer and data storage, hinder the ability to observe runtime state in a deployed distributed embedded system using traditional logging mechanisms. LowLog, a logging framework capturing runtime control flow traces, attempts to attack this visibility problem. Our work over the past year formalizes the benefits of LowLog, and takes from this formalization the new observation that local token scoping is important for compact logging. Specifically, our work exploits statically derived runtime program behavior to create small local name spaces from which token identifiers are assigned. This results in identifiers having very small bit widths and thus minimizes the resulting log bandwidth.

Virgil: The failure of sensing device software can sometimes have life-threatening consequences. Medical monitors, or sensors that monitor the infrastructure, must be certifiably robust. Our Virgil project investigates programming language tools and software that can certify the robustness of sensing devices and software. It has developed a domain-specific language that encourages design for certifiability and make certification easier, and domain-specific tools for certifying the four key properties of space bounds, soft-real-time response, life time, and meaningful results. A tool maps a Virgil program to a timed automaton and uses a real-time model checker to check properties of a timed automaton that represents all program variables via transitions, and can capture timings of most operations.

Tenet: Finally, our Tenet project revisits the architectural foundations of the sensor network systems built and deployed by CENS. Tenet leverages the fact that every network we deploy has masters: 32-bit CPU-class nodes for which power can be engineered. Its architectural principle constrains multi-node fusion functionality to these relatively less constrained nodes. This year, the project has focused on expanding the expressivity of its tasking language. Using a general-purpose threads package and a dynamic loader for TinyOS, the project has explored an extension of its tasking language beyond linear data-flow programs. Now, tasks can contain arbitrary loop constructs and conditionals, expanding the scope of supported applications. Furthermore, the new tasking language supports pre-emption and therefore greater concurrency of execution.