

2.8 Programming and Platforms (PRO)

CENS systems research strives to advance the state of the art in scalable, distributed observing systems. To this end, our research efforts have focused on two critical areas: the design and evaluation of **architectures and programming systems**, and the development of practical **tools and platforms**. Together, these two research directions will eventually (and, in some cases, already) enable the Center to field sophisticated, rapidly reconfigurable, multi-user observing systems that support advanced sensing modalities.

Architecture and Programming Systems

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 nimble and sustainable observing systems.

Visibility: Observing the internal execution of severely resource constrained wireless embedded devices remains a critical block to widespread adoption of embedded wireless sensing systems. Our LowLog system obtains semantically dense execution logs, provides triggered retrieval of these logs, and supports correlation of logs to help diagnose problems. **LowLog** automates and makes readily available standard logging techniques, such as inserting an arbitrary logging preamble to functions to log. At the same time, it enables exploring non-traditional logging techniques such as caller side logging, capturing compressed runtime control flow decision logs, and the impact of alternate log token encoding mechanisms.

Robust development: 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 aims to develop programming language tools and software that can certify the robustness of sensing devices and software. The project is developing a domain-specific language that will encourage 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. The certification tools will increase our confidence in sensing devices and help decrease the scope, duration, and cost of the testing effort.

Software architecture: Sensor network programmers must cope with a variety of concerns, including severe resource and energy constraints, consistency and synchronization among nodes, and node failure. Programmers currently address these concerns by interweaving the code for these concerns together with application logic.

Separation of concerns will enable these different constraints to be decoupled from application logic, and will allow applications to more easily cope with changes in the environment, or new functional or performance requirements. Building upon our prior work on the Pleiades programming language, we have been exploring new language mechanisms that enable application programmers to specify these concerns, and new compiler and runtime algorithms that address these concerns in an automated fashion. Our initial focus has been on robustness to, and recovery from, node failure.

Finally, our **Tenet** project revisits the architectural foundations of the sensor network systems built and deployed by CENS. The project has made considerable progress in the past year, having developed a robust duty-cycling mechanism called AEM that allows duty cycles of 1 to 3%, while maintaining the generality of Tenet. It also conducted a 3-month long Cyclops deployment at the James reserve, enabling biologists to observe avian breeding patterns over a relatively long time period at fine spatiotemporal scales.

Tools and Platforms for Observing Systems

A second systems research thrust is the development of mature tools and platforms for deployments of observing systems. Our efforts in this thrust revolve around three issues: low power, rapid deployment, and time services.

In **Low-power Energy Aware Processing (LEAP)** we have created platform support for deep, accurate, runtime energy consumption information. Building upon our previous years' research on fine-grained energy accounting instrumentation in embedded systems, we have now scaled it to desktops as well as datacenter servers with

emphasis on multi-core processors. We also fuse these direct measurements with indirect information obtained from a performance counter based behavioral model to provide unprecedented visibility of per-process CPU and RAM energy consumption information on multi-core systems. Evaluation with carefully designed experiments shows that our system is able to provide per-process energy information with an accuracy of at least 96%.

The **GeoNet** platform (see 2.03 Seismic; SEI 01) for rapid and distributed geophysical sensing builds upon the LEAP technology. It enables a rapidly installable wirelessly linked seismic network to measure earthquake or volcano sources in the near field to understand the underlying physics, or in buildings to understand earthquake damage. During this second year of the effort we partnered with one of the leading manufacturers of seismic recording systems, Refraction Technology, Dallas Texas, or Reftek, to construct several prototypes. We used the Mexico and Peru networks based on our older platform technology as testbeds for development of GeoNet systems software targeted at the seismological community. The software advances included an improved Disruption Tolerant Shell (DTS), measurement of radio link quality (ETT), network logging, an embedded web interface based on Emstar for deployment and maintenance, network timing, and a new routing protocol that caches the routes across sleep cycles for a fast startup.

Crucial for platforms such as GeoNet is high quality time information, which is the topic of our final activity, **Time Synchronization—Take 2**. We have explored two new directions. First, we have developed a new method of post facto time synchronization for broadband seismic arrays called data driven time synchronization (DDTS). We are implementing DDTS for seismic networks using microseisms as the underlying characteristic for synchronization and applying it to time offset data from the joint CENS and CalTech Meso American Subduction Experiment (MASE). Second, we have developed a new type of local clock source called Crystal Compensated Timer (XCT) that has frequency stability characteristics similar to timers based on temperature compensated crystal oscillators (TCXO) but is much cheaper and less power hungry because of its digital nature, and thus promises to enable hitherto impossible to achieve duty cycling and time synchronization capabilities. In collaboration with researchers at Berkeley, we are incorporating this technology into the new Quanto Testbed Mote platform with the iCount fine-grained energy monitoring software.