

Programming Environment and Architecture for Situational Awareness and Response: the R³SAR Project

Robert Fowler, University of North Carolina, Chapel Hill

The objective of this research is to investigate and implement a software architecture to improve productivity in the development of rapidly deployable, robust, real-time situational awareness and response (R³SAR) applications. The approach is based on a modular cross-layered architecture that combines a data-centric descriptive programming model with an overlay-based communication model. By improving the economics of deploying sensor and response systems, this research will benefit society in several ways. In particular, it will facilitate the upgrading and re-deployment of systems during an emergency to cope with unanticipated issues. The productivity gains in deploying sensors and mobile devices will benefit other domains, such as scientific field research using sensor networks.

Applications such emergency response (in the face of both natural and human disasters), are distinguished by the need to be rapidly deployable, to maintain robustness under potential disruptions in the network, and to provide real-time communication guarantees. The system, R³SAR, is intended for a domain in which sources are capable of constantly streaming data and that users require frequent and timely updates from critical data sources such as field sensors, remote sensors (optical and radar), and simulation-driven forecasts.

The R3SAR project is currently exploring a model that provides programming abstractions similar to the MapReduce model coupled with user-defined goals and constraints. The properties of classes include desired sampling/update rates, delivery latencies, and useful lifetimes. The user then defines (selects) a list of predicated filter functions (map and reduce operators) that can be applied to data streams. An initial prefix of the list of filters describes the “normal” data manipulation and flow through the system. To maintain levels of service, additional filters are applied to the data streams to relax the service goals while continuing to satisfy the constraints. For example, to deal with a degradation of network quality, the system may first apply a user-defined rule that decreases sampling frequency. If this is insufficient, the system may apply a rule that introduces lossy compression.

This past year has seen a revolutionary transition in the technologies available for sense and respond systems. Inexpensive Android-based mobile phones and tablets have gotten a significant share of the commercial market. Android provides pre-packaged versions of many of the software components needed by R3SAR including an on-platform relational data base and an “intents” mechanism that implements a publish-subscribe abstraction suitable for composing filter pipelines. Another development has been the rapid expansion of the open-source ecosystem around the Arduino microcontroller and related devices. Interfaces between the Android, Arduino, and USB-based sensors are now available, including libraries supported by Google. We are therefore shifting our attention to building the R3SAR infrastructure as layer upon the Android API. Our primary driving problem is to build an extensible infrastructure for vehicle-based sensors connected to a central geo-analytics platform. The first use will be for mobile road condition (temperature, precipitation, ice) sensing for use by public safety and school system officials.