

Developing Instrumented Construction Site Applications using the CONSUL Middleware

Robert Goodrich, Kelly Chang, Dr. Jamie Payton
{rgoodri1, kchang5, payton}@uncc.edu



Robert Goodrich
College of Charleston

Introduction

In recent years, ad-hoc networking has become an interesting topic of research. An ad-hoc wireless network uses the same technology as a normal wireless network (802.11 protocol), however, it differs in one critical trait; it has no infrastructure. To ease development of applications on an ad-hoc network, we build on top of Dr. Payton's research into middleware for ad-hoc networks, CONtext Sensing User Library (CONSUL). CONSUL takes care of all of the nitty-gritty details of acquiring information in an ad-hoc network and encapsulates them in a nice package for application developers. We aim to prove that ad-hoc network application development using CONSUL is easier than starting from scratch. To do so, this project focuses uses CONSUL in the development of a complex application with practical use: supporting construction site management in an instrumented construction site.

Background

Some problems in sensor networks include mobility, uncertainty, power constraints, node discovery, and routing. [1] This creates a lot of work for the programmer when trying to develop an application. One idea is to separate all of this from an application developer who just wants to use the ad-hoc network to get some information without knowing all of the networking details. One approach to this problem is the use of middleware solutions and

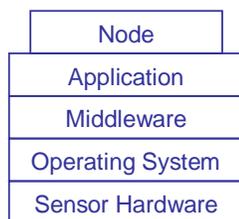
toolkits, which package well-known solutions to acquiring information in an ad hoc network and provide an easy to use API to the application developer. An example is the context toolkit [3], which abstracts the networked sensors into context widgets. The widgets provide an interface to various sensors or groups of sensors. The software developer then includes the widgets in his application, which allows for all sensors to be treated as if they provide the same interface, even if the network is made up of many different kinds of sensors with different methods for accessing data. While the context toolkit is decentralized and has good communication, it also carries with it a large amount of overhead and limited types of aggregation. Aggregation is the summary of data for the purpose of analysis, and is important because it is how the sensor data collected is presented to the application and ultimately the user of the application. [2] Using CONSUL provides us with a similar access to the sensors, but considerably less overhead and more data aggregation options.

Research

Our plan for this summer was to implement an application for to simulate a construction site using the Gumstix embedded computer mounted on the iRobot Create to serve as machinery or a worker, and to use sensor boards such as the Mica2 and Tmote Sky motes. Getting all of the hardware setup and software installed on these platforms is not an easy task, so we were only partially successful to date. Concerning the application itself, we were going to use a robotics abstraction program called player, to control the Create. However, this was determined to contain huge amounts of overhead and access to the robots themselves was hidden away in many

Research Continued

layers of software. Needless to say, we did not use player, and instead turned to Open Interface for controlling the iRobot Create. Next we had the issue of what language to program the application in. Since the middleware CONSUL was programmed in java, and the Gumstix computer did not support that initially, so setting up a java virtual machine on the Gumstix was another time-consuming task. Once all that was setup, we had the vision of implementing nodes with the structure of the figure below. The application level is what we REU students work on, which is the specific application designed to showcase ease of design of network applications using the CONSUL middleware and network discovery package. The middleware level includes both CONSUL, which was developed by Dr. Payton and some of her colleagues at Washington State University [2]. We used a gumstix and iRobot Create combination to represent a construction worker, while having laptops represent various equipment and materials on a construction site. The first step for using CONSUL was to create a monitor that interfaces with our hardware by extending the Abstract Monitor. The next step was writing the application that uses them along with a network discovery package, and this was split up into 3 categories; supervisor, worker, and sensors, of which sensors was divided up into 3 sub-categories; backhoe, steel pile, and concrete mixer. The sensor applications only differ in what local monitors they have, while the worker and supervisor applications each have an extra thing over providing and searching for monitors. The worker, since it is simulated, must also have control over the robot motors. The supervisor puts its data into a GUI for convenient display of important information.



Conclusions

To conclude, the application development time was relatively short when compared with the setup time. The setup was tedious and riddled with problems, but lead to large amounts of learning about robotics, embedded computers, and cross-compilation toolchains. The application itself was quite easy to develop once the monitors were implemented on the hardware we had. The summer was full of many trials and tribulations, but each obstacle overcome created great satisfaction.

Future Work

In the near future, the application can be tested using actual sensor motes to provide sensor data, instead of laptops running simulated sensors. After that, the application can be rewritten using QueryME, to provide a comparison between application development using CONSUL and the networking discovery package, or with using QueryME. QueryME is a middleware that wraps CONSUL, a network discovery package, as well as providing more advanced features and functions for the application developer. Another future venture would be to deploy the application in a real construction environment

References

- [1] J. Blumenthal, M. Handy, F. Golatowski, M. Haase, and D. Timmermann. Wireless Sensor Networks – New Challenges in Software Engineering. *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, Sept 2003.
- [2] G. Hackmann, C. Julien, J. Payton, and G.-C. Roman. Supporting Generalized Context Interactions. *Proceedings of the 4th International Workshop on Software Engineering and Middleware, co-located with ASE'04, Linz, Austria*, T. Gschwind and c. Mascolo (editors), Lecture Notes in Computer Science 3437, Springer-Verlag, March 2005, pp. 91-106.
- [3] D. Salber, A. K. Dey, and G. D. Abowd. The Context Toolkit: Aiding the Development of Context-Enabled Applications. *In Proceedings of CHI'99, Pittsburgh, PA*, May 1999.