Rapid Prototyping for Urban Sensing

Dawud Gordon, Matthias Berning, Hedda R. Schmidtke and Michael Beigl
Karlsruhe Institute of Technology
[*firstname.lastname*]@kit.edu

ABSTRACT
Sensing applications with blanket coverage over urban areas can provide vital information for scientists looking to research the effects of specific phenomena on urban environments. Unfortunately there are many large hurdles which have to be overcome in order to implement such sensing networks, for example little or no access to nodes for maintenance operations and system failures due to theft and vandalism, and the irreversible nature of the deployments. One solution is to implement iterative prototypes in the urban environments to gather experience, although wireless sensor network prototyping brings with it its own set of problems such as rising costs due to hardware and development investments and the level of expertise required by the developer. This paper looks at a possible solution for this complex problem set based on a wireless sensor network platform. The platform is modeled after the dinam concept which may prove to be a solution for many problems encountered in the field of blanket-coverage urban sensing applications.

Author Keywords
Wireless sensor networks, Prototyping, Urban Sensing, BASIC, Programmable Reality

ACM Classification Keywords
C.2.1 Network Architecture and Design: Wireless communication; C.2.4 Computer Systems Organization: Distributed Systems.

General Terms
Design, Human Factors, Languages.

INTRODUCTION AND CHALLENGES
When prototyping distributed sensing applications for urban scenarios, several unexpected challenges arise due to the unpredictable environment. Previous urban prototyping and application approaches such as [7],[9] and [5] uncovered challenges in urban computing which were not anticipated and are unique to the field, several of which have been documented in [11]. These issues included but were not limited to the following problem areas: *limited access* for maintenance purposes[11], damage through *environmental conditions*[7] and damage or loss due to *vandalism*[9, 5].

Rapid prototyping brings with it its own set of challenges which need to be addressed, regardless of the application area. On the one side, prototypes must be *disposable*, as they are often subjected to unexpected situations and can be easily destroyed. On the other side they must be *dynamic* enough to be adapted as new insight is acquired during deployment. Failure to meet this criteria incurs long turnaround times between iterations, making prototypes unnecessarily costly in terms of human resources and funding.

When considering urban deployments of wireless sensor network prototyping both of these sets of challenges come into play. This work examines a specific wireless sensor network development platform as a basis for development of urban sensing applications. First the technical details of the platform will be discussed as well as the development and prototyping process. The system is then evaluated based on its suitability for prototyping high-density high-surface-area (blanket coverage) sensing applications in urban environments along the lines of [7] based on the afore-mentioned challenges as well as the general challenges that accompany the prototyping process.

In [6], researchers from the Karlsruhe Institute of Technology presented a concept for wireless sensor network rapid prototyping called dinam, which allows researchers as well as non-experts to easily create wireless sensor network applications and prototypes. The system presents a novel approach to application development which reduces the total cost of ownership by streamlining development and reducing the price of the development platform. This work will conceptually explore the effects of this platform on the field of urban computing, specifically in prototyping blanket-coverage urban sensing applications. First we will discuss the challenges of urban prototyping, followed by an introduction to the dinam platform and its applications for prototyping of urban sensing applications. We will then discuss the advantages using a hypothetical deployment and conclude with a brief discussion and summary.

APPLICATION PROTOTYPING AND DEVELOPMENT
Development of wireless sensor network applications in the past has been a time-consuming and effort-intensive task. Normally, assuming pre-existing hardware, the process involves several components such as sensor nodes, a development environment, libraries, drivers, compiler, linker, assembler, debugger, programming setup (e.g. a hardware programmer device, cable, programming software, or an over-the-air programming (OTAP) approach) and design, docu-

Copyright is held by the author/owner(s).
UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

mentation and code revision platform. A typical development process works as follows: 1) Design application, 2) Set up development environment, 3) Connect hardware, 4) Prepare libraries and previous code, 5) Write code, 6) Compile code and link binaries, 7) Write binary to hardware, 8) Evaluate, 9) Repeat from 5.

This overhead, while being advantageous for large-scale projects and deployments, is a hindrance in rapid prototyping. The dinam platform in [6] addresses these problems with an integrated, interactive, bottom-up development approach: a dinam concept sensor node itself contains the program, development environment, program interpreter and debugger as well as all documents in a single device. The result is a single fluid development process which does not distinguish between run-time and program-time, allowing the developer to dynamically modify system behavior without physically touching the system or taking it temporarily offline. This gives developers the opportunity to compose system behavior piece by piece from the bottom up. While [6] discusses rapid prototyping, it does not address urban computing applications, which is the focus of this work.

There are numerous examples of mote-type sensor nodes. Other examples of rather small, simplistic WSN development platforms are the uPart Sensor nodes [2] from the TecO group at Karlsruhe Institute of Technology (KIT) and the MITes [12] from House_n at Massachusetts Institute of Technology (MIT). An example of a simplified application prototyping tool is the Arduino development platform for embedded electronics[1]. The Arduino platform is delivered with its own intuitive development environment which allows developers to quickly create applications. The developer only requires a basic knowledge of the ANSI C programming language, allowing easy prototyping of complex embedded applications with low overhead. The mbed™ project from ARM and NXP [8] uses a browser-based development environment provided via an externally hosted decentralized server environment. While removing the installation time of the development environment is a step in the positive direction, the discrete development process is still an unnecessary hindrance when creating applications.

THE DINAM PLATFORM

The dinam platform consists of two different units: the dinam-mite and the dinam-mote. The dinam-mite provides an interface to the outside world over a TCP/IP connection for prototype application development and modification. The dinam-mote is a small configurable low power sensor node whose only task is to deliver remote sensor data to a dinam-mite in a fashion specified by the application designer over the dinam-mite.

The dinam-mite

The dinam-mite (≈30 USD) is an application development and prototyping platform embedded in a wireless sensor node. The node itself is based on the D-Bridge [6] but has been modified for communication using WLAN technology instead of Ethernet communication. The dinam-mite simplifies the prototyping process by integrating the development

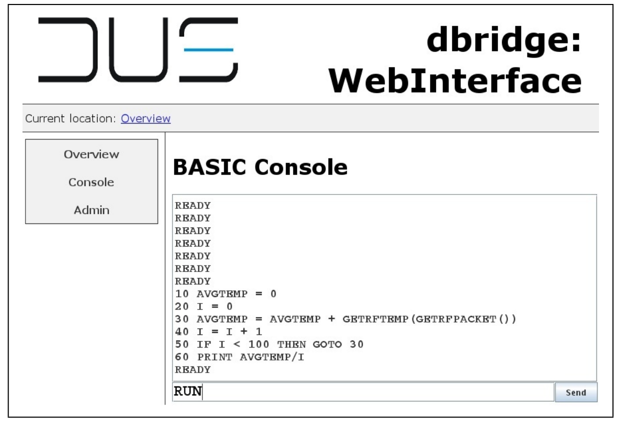

Figure 1. A Screenshot of the dinam-mite programming interface

environment – including the data – into the wireless sensor system. This means that the dinam platform removes the need for development software at the application programmer's computer to write programs for the sensor node – not even an editor is required. All code and the development environment is integrated into the node itself.

Access to the dinam-mite and the development environment is provided through a web-server embedded on the WSN system, thus the only tool a developer requires is a web-browser. Fig. 1 shows the developer's view of the IDE running on the dinam-mite: the web-interface provides simplified access for easy operation. Because the development environment is integrated into the sensor system, hardware and software versions of all development components are always synchronized and incompatibilities do not occur. This approach also removes some of the time consuming steps in the development process, such as writing the binary to the hardware, as well as reduces the initial static set-up time and effort overhead.

The online code development process creates a single constant run/development time. During this operational period, the developer is able to receive immediate feedback to his/her actions so that the evaluation step can be pipelined as well. The resulting development platform contains an interactive run-time programming interface which allows the programmer to change parameters, execute actions and evaluate the results simultaneously. The command interface is opened by pointing the browser to the interface webpage located at the IP address which device acquires via DHCP. Once the page has been loaded, commands can be inputted directly to the interpreter using the Java-based web application shown in fig. 1, whereby the interpreter provides immediate evaluative feedback of previously issued commands. Along with the BASIC command line interface, it also provides an editor to implement BASIC scripts.

The dinam-mote

The dinam-mote (≈20 USD) was implemented on the Akiba wireless sensor nodes. The Akiba node is based on the PIC18F14K22 8-bit micro-controller from Microchip, and was developed at the Technische Universität Braunschweig.

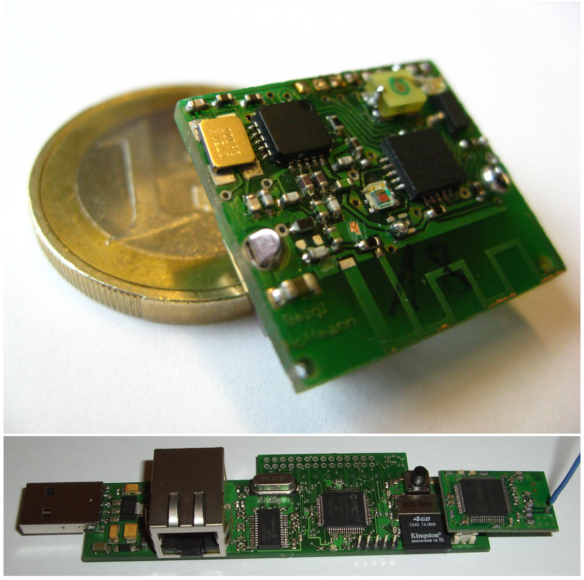

Figure 2. The dinam platform: dinam-mote (above) and dinam-mite

The processor runs at up to 16 MIPS with 512 B RAM, 256 B EEPROM and 16 kB ROM. Wireless communication occurs at 2.4 GHz over the ChipCon CC2500 from Texas Instruments with a printed PCB antenna. The system is on a single-sided PCB measuring 20mm x 18mm with a CR2032 coin cell battery on the back. Due to low-power technology such as wake-on-radio, standby receive modes have consumption rates in the low double digit microwatt range with transmission and reception consumption rates between 10 and 30 mW. This yields a lifetime of many months, or even years, on a single coin-cell battery depending on the how often data is received or transmitted.

The Akiba sensor node is equipped with light, temperature and vibration sensors as well as connection points for up to 4 analog external sensors and sensor/processor boards. On top of the RF communication interface, the nodes implement an over-the-air-configuration protocol that allows for remote configuration of parameters such as duty cycle length and sensing rate comparable to those found in uParts [2]. The node is well adapted to post-hoc computing applications along the lines of Smart-Its [4] which make them specifically suitable for the dinam platform.

The wireless communication and configuration protocol is based on the AwareCon [3] network stack which has been modified to implement the application specific functions of the dinam platform. In addition, the dinam-motes implement a version of CentRoute [10] to allow power-aware, low- computation routing for multi-hop situations. This, combined with a maximum range 50m per dinam-mote as well as multi-hop routing reduces the mite-to-mote ratio required for blanket coverage of large areas.

EVALUATION

In order to initially evaluate the dinam platform as a prototyping tool for urban sensing applications, a conceptual

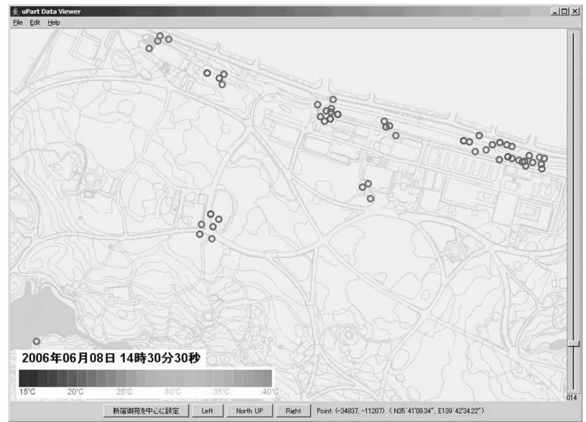

Figure 3. A Screen Shot of the Airy Notes Urban Sensing Project [7]

analysis will be conducted of a successful urban sensing application. The Airy Notes [7] project used wireless sensor nodes to conduct a climate analysis in a public park called Shinjuku-Gyoen Gardens in Tokyo, Japan. The motivation behind the project was that a wireless sensor network distributed within the garden would be able to retrieve information of interest for scientists who study the microclimate of that particular area. In Fig. 3 the distribution of sensor nodes within a section of the park can be seen.

During the Airy Notes project several difficulties were encountered. The relatively high cost of the hardware used (200 uParts á 30 USD and 10 XBridges á 200 USD = 8,000 USD total) was the limiting factor in the coverage which was achievable. Furthermore, node losses due to weather damage or theft were serious risks to project success due to funding. When observing Fig. 3 one can clearly see the effects of the monetary constraints as node clusters are scarcely distributed throughout the park, rather than the intended blanket coverage. A further problem was the static nature of the firmware on the nodes that were deployed. This prevented researchers from implementing changes to the system after deployment, as that would have involved reflashing or reconfiguring each node. This further limited the deployment as each node had to be located without technical assistance for maintenance purposes or to reconfigure the prototype, making turnaround between iterations difficult.

In this scenario, a similar approach using the dinam platform would have had several benefits. Firstly, the cost of the hardware would have been reduced by almost 50% from 8,000 USD to 4,300 USD (200 dinam-motes and 10 dinam-mites), indicating that the developers would have been able to double sensory coverage. This cost reduction is calculated over a system that was already using cost-concious hardware, where applications using standard nodes (for example, typical costs for nodes from www.xbow.com are over 100 USD) would incur exorbitant costs which normal research grants would not be able to support.

The remote reconfiguration capabilities of the dinam-motes and logic definition capabilities of the dinam-mites would

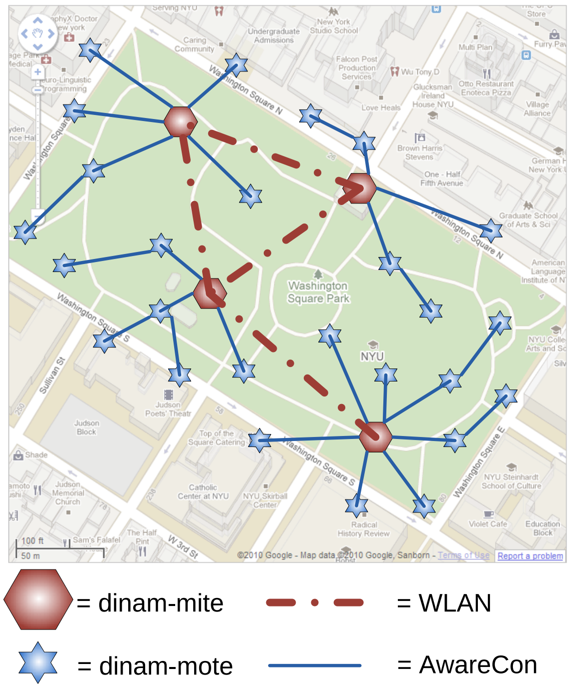

Figure 4. A Hypothetical Node Topology for a Blanket-Coverage Urban Sensing Application

have allowed for remote alterations to the applications, making multiple iterations possible. Furthermore, the printed antenna of the dinam-motes allows for malleable casings such as coatings or bags, and reduces susceptibility to weather-related antenna problems. Additional savings would also have been realized due to the multi-hop capabilities of the dinam-motes as they would reduce the number of more costly dinam-mites required for this application. This is demonstrated in Fig 4, which models how a similar approach would appear in Washington Square Park in New York City.

CONCLUSION

We presented some of the problems which one encounters when developing blanket-coverage urban sensing applications. We also discussed how prototyping can help solve many of these issues, but introduces other challenges of its own. We then introduced a platform based on the dinam concept and discussed how this platform would address the issues encountered when prototyping large-scale urban sensing applications. The indications are that the dinam platform, while only addressing a subset of the problem set, would greatly improve ease of use and developer-friendliness of the prototypes by reducing the effort required for iterative prototyping steps, and also by reducing the cost of the prototypes and applications.

ACKNOWLEDGEMENTS

The authors would like to acknowledge funding by the European Commission for the ICT project CHOSeN "Cooperative Hybrid Objects Sensor Networks" (Project number 224327, FP7-ICT-2007-2) within the 7th Framework Programme.

REFERENCES

1. A. an Open-Source Electronics Prototyping Platform. http://www.arduino.cc/, 2010.
2. M. Beigl, A. Krohn, T. Riedel, T. Zimmer, C. Decker, and M. Isomura. The upart experience: Building a wireless sensor network. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, pages 366–373, New York, NY, USA, 2006. ACM.
3. M. Beigl, A. Krohn, T. Zimmer, C. Decker, and P. Robinson. Awarecon: Situation aware context communication. In *Proceedings of Ubicomp 2003*, pages 12–15, 2003.
4. H. Gellersen, G. Kortuem, A. Schmidt, and M. Beigl. Physical prototyping with smart-its. *IEEE Pervasive Computing*, 3(3):74–82, 2004.
5. D. Gordon, M. Beigl, and M. Iwai. A study on the use of wireless sensor networks in a retail store. In *Workshop on Pervasive Shopping at the Conference on Pervasive Computing 2009 (Pervasive09)*, Nara, Japan, 2009.
6. D. Gordon, M. Beigl, and M. A. Neumann. dinam: A wireless sensor network concept and platform for rapid development. In *Proceedings of the Seventh Internation Conference on Networked Sensing Systems*, pages 57 – 60, Kassel, Germany, 2010. IEEE.
7. M. Ito, Y. Katagiri, M. Ishikawa, and H. Tokuda. Airy notes: An experiment of microclimate monitoring in shinjuku gyoen garden. In *Proceedings of IEEE Fourth International Conference on Networked Sensing Systems (INSS 2007)*, pages 260–266, 6 2007.
8. A. R. M. Ltd. The MBED Rapid Prototyping Platform for Microcontrollers. www.mbed.org, 2010.
9. T. Ojala, H. Kukka, T. Lind andn, T. Heikkinen, M. Jurmu, S. Hosio, and F. Kruger. Ubi-hotspot 1.0: Large-scale long-term deployment of interactive public displays in a city center. In *Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on*, pages 285 –294, 9-15 2010.
10. T. Stathopoulos, L. Girod, J. Heideman, and K. W. Deborah Estrin. Centralized routing for resource-constrained wireless sensor networks (sys 5), 2006.
11. O. Storz, A. Friday, N. Davies, J. Finney, C. Sas, and J. Sheridan. Public ubiquitous computing systems: Lessons from the e-campus display deployments. *Pervasive Computing, IEEE*, 5(3):40 –47, july-sept. 2006.
12. E. M. Tapia, S. S. Intille, L. Lopez, and K. Larson. The design of a portable kit of wireless sensors for naturalistic data collection. In *Pervasive Computing, 4th International Conference, PERVASIVE 2006*, volume 3968 of *Lecture Notes in Computer Science*, pages 117–134, Dublin, Ireland, May 2006. LNCS.