Research Repository UCD

Title	Simulating interaction with wireless sensor networks
Authors(s)	Brennan, Adam J., O'Grady, Michael J., O'Hare, G. M. P. (Greg M. P.)
Publication date	2010-10
Publication information	Brennan, Adam J., Michael J. O'Grady, and G. M. P. (Greg M. P.) O'Hare. "Simulating Interaction with Wireless Sensor Networks." IEEE, 2010.
Conference details	Presented at the International Conference on Computer Information Systems and Industrial Management Applications (CISIM 2010), Cracow, Poland, October 8-10 2010
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/2797
Publisher's statement	Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/CISIM.2010.5643650

Downloaded 2024-04-18 15:11:25

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

Simulating Interaction with Wireless Sensor Networks

Adam Brennan, M.J. O'Grady and G.M.P. O'Hare
CLARITY: Centre for Sensor Web Technologies,
University College Dublin,
Belfield, Dublin 4, Ireland.
Email: adam.brennan@ucdconnect.ie, {michael.j.ograd y, gregory.ohare}@ucd.ie

Abstract

Wireless Sensor Networks(WSNs) are primarily regarded as data gathering entities that route data to a central server for subsequent processing. However, as WSN technologies mature and their prevalence increases, it is envisaged that a range of diverse and innovative services for mobile subscribers will be launched, of which some may require remote access and interaction with individual WSNs. Such an eventuality will give rise to additional constraints in the WSN planning and deployment processes, and raise particular challenges in the practical engineering of mobile applications and services. In this paper, a simulation tool is described. This tool enables software engineers rapidly model WSN deployments, and in this way, explore how interactions between mobile users and WSNs may occur.

1. Introduction

Network planning is an indispensable part of any network deployment process. In the case of Wireless Sensor Networks (WSNs), a multiplicity of factors must be factored into the planning process to ensure the WSN performs in the manner envisaged. Presently, issues pertaining to maximising WSN operational longevity remain the focus of much WSN research. The conventional view of WSNs is as inherently distributed entities, capturing select data and returning such data for subsequent processing in central locations. While the advantages of this central approach are not contested, it is envisaged that scenarios may arise where the availability of data gathered by the WSN may be of use in the provision of services for mobile users while they are in the proximity of individual WSN nodes - Operation & Maintenance tasks being a particular case in point. Thus, application developers and service providers have a stake in the planning process. In this paper, a simulation tool aimed specifically at addressing the needs of such stakeholders is described.

2. Related Research

Given the inherent difficulties in deploying realistic test beds for WSNs, the use of WSN Simulators has been proposed as a means of modelling and evaluating performance of WSNs under a variety of configurations. Protocol analysis and energy performance are two aspects frequently investigated. A variety of simulation environments have been described in the literature - well documented examples include J-SIM [1], TOSSIM [2] and SensorSim [3] amongst others. A detailed description of the various simulation and emulation approaches is beyond the scope of this discussion; however, the interested reader is referred to Du et al [4] for a taxonomy and description of a range of simulation tools for WSN environments.

Invariably, current approaches focus on the lower layers of the stack. The needs of those operating at the application layer has not been a priority, though some prototypes demonstrating interaction have been described in the literature [5], [6], [7]. As the deployment of WSNs increases, as it is expected to in the coming years, this situation will change. WSNs may perform a multitude of tasks, and be harnessed by a range of diverse services. Thus more attention needs to be focused on the requirements of those working at the application layer, especially if adaptive distributed, mobile services [8] [9] are to be delivered in practice. As an initial step in this direction, an Interaction Simulator (iSim) is being developed that allows experimentation with various WSN deployment configurations, and enables the exploration of how

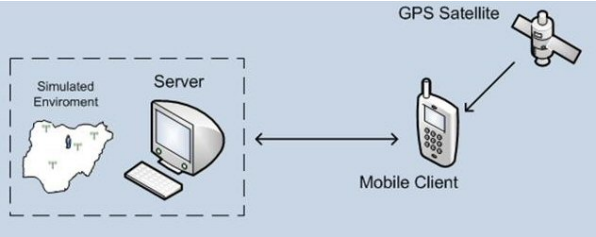

Figure 1. Overview of iSim.

third-party users might encounter and use services that require remote WSN access.

3. Design of iSim

iSim consist of two components, the core simulator and a mobile client (Figure 1).

The Simulator component is hosted on a fixed server. Its interface is constructed around an electronic map. The definition of virtual sensors, and the specification of specific routes, where designers wish to explore how interactions between mobile subscribers and sensors might evolve, is facilitated. A combination of routes and sensors define a scenario. Once scenarios have been defined, they can be stored and replayed at will.

The Mobile Client component hosts any application or service that can harness information from the WSN whilst in its vicinity. The client can be hosted on a mobile phone emulation environment, for example, the Sun Java Wireless Toolkit. Alternatively, it can be hosted on a real mobile device. In this case, access to the simulator may be facilitated via the internet.

4. Implementation of iSim

iSim is implemented in standard Java. Google Maps and OpenStreetMap can each be used as a basis for the user interface. To ensure interoperability, the lightweight data interchange format JavaScript Object Notation (JSON) [10] has been adopted.

It is assumed that interaction with the WSN will occur via a mobile phone. A sample client has been implemented using Java 2 Micro Edition (J2ME), as this is one of the most common platform available for mobile applications at present. However, there is no reason why Android or iPhone applications could not use the simulator provided they use an agreed port and can interpret JSON.

5. iSim in Practice

To define a scenario, the following steps are followed:

1) A number of virtual sensors are defined. These may be modelled on sensors that are commercially available, or may be configurations of the designer's choosing.
2) A geographic region where the WSN will be deployed is identified, either in Google Maps or OpenStreetMap.
3) A path is defined on the map, via a series of mouse clicks.
4) The location of sensors is then specified. The geographic point where the sensors will be located is selected and the broadcast radius of the sensor defined. Note that the issue of interference is not considered in this simulator as other simulators have effective mechanism for modelling interference and other phenomena. Figure 2 illustrates how the map appears after the path and sensors and have been defined. In this case, OpenStreeMap has been used. Google Maps may also be adopted and Figure 3 illustrates how this appears using a satellite image. In some cases, this may be preferable as the level of detail supported may better inform the designer of potential problems.
5) The scenario is then saved, and can be activated at any time.

On commencing to play a scenario, the simulator migrates a virtual user along the path every few seconds. The simulated GPS position is broadcast to the mobile client. Likewise at each iteration, the position of the virtual user relative to the sensors is calculated. Should the user be perceived to be within the range of a virtual sensor, the status of the sensor is communicated to the mobile client.

6. Remote access

Simulation is essentially an approximation process. Once the possibilities offered by the simulation have been exhausted, it may be necessary to explore how a service might operate in the corresponding physical environment. Thus remote access to iSim is supported. In this case, it assumed that the mobile application itself has access to GPS. As the designer follows the physical path, GPS updates are continuously sent to the simulator. When the simulator determines that, for a given scenario, the user is in the coverage area of a sensor, it broadcasts simulated values to the mobile devices. Figure 4 demonstrates this remote interface.

Figure 2. iSim interface using OpenStreetMap.

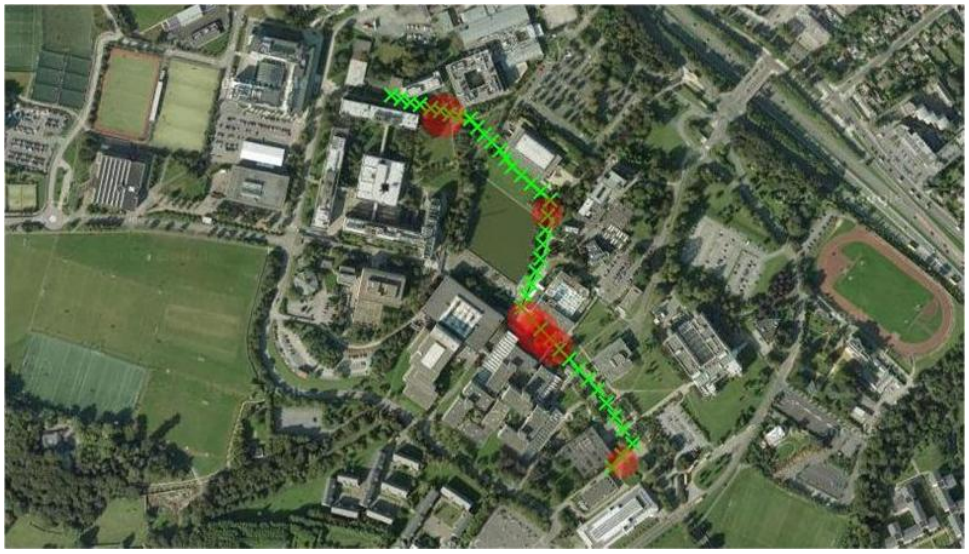

Figure 3. iSim interface using Google Maps.

7. Future Work

Though Google Maps and OpenStreetMap are excellent resources in themselves and invaluable as a basis for providing iSim functionality, they are compromised in that they are two dimensional (2D). A three dimensional (3D) environmental representation would contribute to a more meaningful analysis of any interaction possibilities. For urban environments, Google Street View offers significant potential; for rural environments, the situation is more complex though not as urgent due to the sparse population. The use of Digital Terrain Models (DTMs) may offer possibilities for rural scenarios.

8. Conclusion

This paper presents the design and implementation of a prototype simulator for aiding the design of WSN-based or WSN-augmented services for mobile subscribers. WSN technologies will increasingly offer possibilities for integration into third party services and applications. Currently, WSN planners and software designers have little to aid them in evaluating how WSNs may be remotely accessed and interacted with.

Figure 4. The interface to the iSim mobile client. In this case, the mobile user is about to encounter two sensors on their path.

The use of simulators offers one approach by which this aspect of the design process may be addressed.

Acknowledgment

This work is supported by Science Foundation Ireland (SFI) under grant 07/CE/I1147.

References

[1] A. Sobeih, W.-P. Chen, J. C. Hou, L.-C. Kung, N. Li, H. Lim, H.-Y. Tyan, and H. Zhang, "J-sim: A simulation environment for wireless sensor networks," in *ANSS '05: Proceedings of the 38th annual Symposium on Simulation*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 175–187.

[2] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 126–137.

[3] S. Park, A. Savvides, and M. B. Srivastava, "Sensorsim: a simulation framework for sensor networks," in *MSWIM '00: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2000, pp. 104–111.

[4] W. Du, D. Navarro, F. Mieyeville, and F. Gaffiot, "Towards a taxonomy of simulation tools for wireless sensor networks," in *SIMUTools '10: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. Brussels, Belgium: ICST, 2010, pp. 1–7.

[5] F. Tirkawi and S. Fischer, "Remote interaction tool for wireless sensor networks," *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pp. 360–364, May 2008.

[6] D. Merrill, J. Kalanithi, and P. Maes, "Siftables: towards sensor network user interfaces," in *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*. New York, NY, USA: ACM, 2007, pp. 75–78.

[7] M. Ringwald, M. Yücel, and K. Römer, "Demo abstract: Interactive in-field inspection of wsns," in *Adjunct Proceedings of the 3rd European Workshop on Wireless Sensor Networks (EWSN 2006)*, Zurich, Switzerland, Feb. 2006.

[8] M. J. O'Grady, G. M. P. O'Hare, J. Chen, and D. Phelan, "Distributed network intelligence: A prerequisite for adaptive & personalised service delivery," *Information System Frontiers*, vol. 11, pp. 61–73, March 2009.

[9] M. J. O'Grady, G. M. P. O'Hare, and C. Donaghey, "Delivering adaptivity through context-awareness," *Journal of Network and Computer Applications*, vol. 30, no. 3, pp. 1007–1033, August 2007.

[10] D. Crockford. (2006) Json: The fat-free alternative to xml. [Online]. Available: http://www.json.org/fatfree.html