CHESS

CENTER FOR HYBRID & EMBEDDED SOFTWARE SYSTEMS

COLLEGE OF ENGINEERING, UC BERKELEY

Spine: Software Framework for Wireless Body Sensor Networks

Sameer Iyengar
Alberto Sangiovanni-Vincentelli

http://chess.eecs.berkeley.edu/

Background

Wireless Body Sensor Networks have the potential to **revolutionize healthcare**.

They **reduce cost** and improve **quality of care** by reducing physical barriers to **prevention**, **detailed monitoring** and **continuous real-time reporting**.

However, the difficultly of application development is a large barrier to adoption. We aim to remedy this by providing tools for BSN application developers.

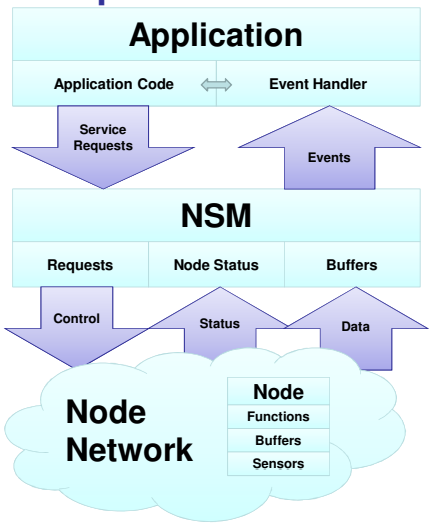
What is Spine?

Spine is a **software framework** that **automates** and **simplifies** common tasks in BSN application creation. It is based on a Java and TinyOS software stack.

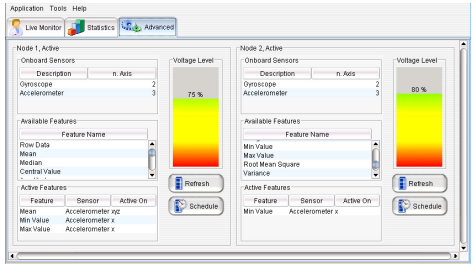
Extensible: Developers can define custom functions to extend node capability.
Developer-focused: Common tasks are automated to allow for rapid application deployment.
Communication Management: A TDMA schedule is dynamically assigned based on application requirements.
Power Management: To save power, nodes are duty-cycled according to the communication schedule.

We have developed and tested many sample applications including a real-time posture classification system and a tool to monitor range of motion in post-operative patients.

Spine Overview



Reporting of real-time local function activation



Network Service Manager (NSM)

Service Requests

- Query node capabilities
- Query request progress
- Request data
- Request to push data

A request for data follows the following generic format:

Node Id	Function Id	Parameters (determined by function definition)

For example, to request the mean value of sensor s on node n over the last 10 seconds computed every second (t is the current time):

n	meanValue	s	t-10	t	1

Events

NSM returns requested data asynchronously when it is ready. Data is identified by a unique identifier.

Error Handling

An application may pre-specify alternate actions in case a request cannot be completed. If unspecified, error notifications allow the application to choose actions based on current node status.

Local Sensing and Processing

The sensor nodes sample and buffer data according to the constraints of current data requests.

All node data requests are abstracted via **functions.**

Functions can simply return raw data or perform local processing.

Developers can define functions that implement:
- Processing algorithms
- Logic to control communication
- Local storage

Return data from a sensor

```
sensorValue (id)
    return getSensorValue(id)
```

Return sensor1 value only if it is greater than sensor2

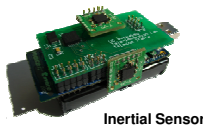
```
compare (id1, id2)
    if (getSensorValue(id1) > getSensorValue(id2))
        return getSensorValue(id1)
```

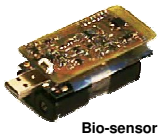
Return the mean value of a data buffer

```
meanValue (id, start, end, interval)
    counter = 0, total = 0
    for (time = start; time <= end; time += interval)
        total += getSensorValue(id, time)
        counter++
    return total/count
```

Hardware

Sensors
Compatible with any 802.15.4 device running TinyOS. We have tested two sensor devices built on the Tmote Sky.


Inertial Sensor


Bio-sensor

Gateway
Compatible with any Java-capable device. We have tested various PCs and the Motorola E680i.

Acknowledgements

Spine is being developed as an open-source project in collaboration with research labs at six institutions:

Berkeley
Cornell University
TELECOM ITALIA
UTD
UNIVERSITÀ DELLA CALABRIA
VANDERBILT

UC Berkeley:
Victor Shia, Ruzena Bajcsy
Telecom Italia:
Roberta Giannantonio, Filippo Tempia, Marco Sgroi
University of Calabria:
Raffaele Gravina, Antonio Guerreri, Giancarlo Fortino
Cornell University:
Phil Kuryloski
University of Texas, Dallas:
Roozbeh Jafari