Project Report

SensorML on SenseTile

Ciaran Palmer

A thesis submitted in part fulfilment of the degree of

Msc Advance Software Engineering

Supervisor: Your Supervisor Name

Moderator: Your Moderator Name

UCD School of Computer Science and Informatics
College of Engineering Mathematical and Physical Sciences
University College Dublin

January 27, 2010

Table of Contents

Abstract

SenseTileSensor Board

SensorML description.

WebServer to access sensor observations

SensorML Bon Mapping with a tool and method to develop sensorML descriptions

Acknowledgments

Chapter 1: **Introduction**

The aim of this thesis project was to prototype a Web Service for accessing sensor observations from the SenseTile Sensor Board. SenseTile is a sensor and processing package including motion sensors, RFID sensors, temperature sensor, audio sensors, pressure sensors, light level sensors video sensors among others. It is used as a replacement for standard ceiling tiles to provide smart building services as part of a Web Sensor Network.

Descriptions of the SenseTile Sensor Board were to be developed using Sensor Model Language (SensorML)[1]. SensorML is a language that describes the process of sensor measurement by providing standard models and XML encoding for such processes. SensorML was be used as it is becoming a standard for describing sensor data processing as part of the SWE[ref].

This description is executed on the processing unit as part of a webservice to generate data. SOS. Implemation added to allow the description to be used to generate observations. Processing of data.

A SensorML to BON Mapping was also to be explored . BON[2] is a notation and a method for Object Oriented systems analysis and design. Based on this mapping it was proposed to develop a tool/method to automatically generate SensorML descriptions. Generally SensorML is hand generated with all the shortcomings of such an approach. The proposed approach would allow sensor processing to defined formally in BON and then refined to JML/Java.

Chapter 2: **Background Research**

How was problem solved before.

2.0.1 Sensor Networks

2.0.2 BON

2.0.3 Sensorml

reference chapter

2.0.4 WebService

2.0.5 JIBX

Chapter 3: **SensorML**

3.1 SensorML Overview

The Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) [ref] is establishing interfaces and protocols that will enable a standardised Sensor Web. It is hoped that an easier integration and development of sensor applications across sensor networks will be facilitated by this standarising activity. Sensor Modelling Language (SensorML) is one part of the SWE initiative and is used to describe sensor systems and the processing of observations from sensor systems.

The purposes of SensorML are to: Provide descriptions of sensors and sensor systems for inventory management Provide sensor and process information in support of resource and observation discovery Support the processing and analysis of the sensor observations Support the geolocation of observed values (measured data) Provide performance characteristics (e.g., accuracy, threshold, etc.) Provide an explicit description of the process by which an observation was obtained (i.e., its lineage) Provide an executable process chain for deriving new data products on demand (i.e., derivable observation) Archive fundamental properties and assumptions regarding sensor systems

SensorML provides a functional model of sensors and an XML encoding to describe sensors and their observations. Sensors are modelled as processes that convert real phenomena to data. Processes take inputs, process them and result in one or several outputs. They can also define relevant metadata. The processes can be connected together in chains using SensorML ProcessChains or Systems.

including measurement by a sensor system, as well as post-measurement processing.

When Modelling a sensor system in SensorML the main entities used are Systems and Components. A SensorML System is used to group sensors that are related spatially to one another. A Component is an atomic process that generally converts a physical phenomen to a digital number. ProcessModels are another modelling that is used to describe a pure process that has no physicality

Quantity, Count, Boolean, Category, and Time provide the basic primitives. Within SensorML, these serve as the basis for specifying all inputs, outputs, and parameters within a Process

DataRecord or DataArray.

Detector model.

The SensorML description of the SenseTile Sensor Board is shown in the following section.

3.2 SensorML Description of SenseTile

SensorML System and Component were used to describe the SenseTile system and it's sensors. The sensor specifications were used to generate the metadata. The Sensors are modelled a detectors. They are process that take scalar values and can perform transformations on them.

System is providing the mapping of inputs to processes and the outputs.

Naming rules inputs and outputs - allow more generic handling of implementation classes.

Processed verus raw data.

It is further recognized that most sensor observations consist of at least two processes: a sampling process and a conversion process.

3.2.1 SenseTile System

3.2.2 Light Sensor

3.2.3 Thermistor

3.2.4 Pressure Sensor

3.2.5 Accelerometer

3.2.6 Acoustic Sensor

Types - SWE -

Count - BigInteger Quantity - Double

DataArray

* Carries an array of int primitives. * All data is casted to other types when requested.

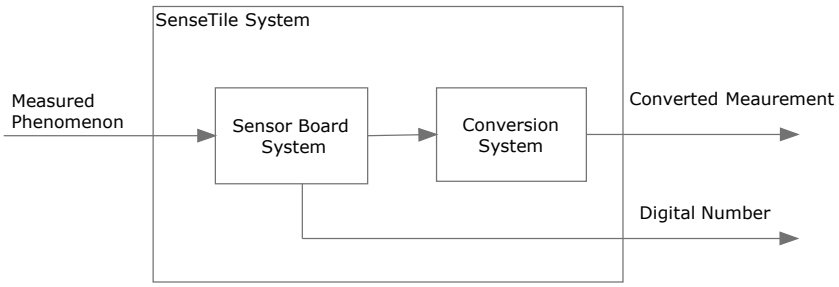

Figure 3.1: SenseTile System Overview

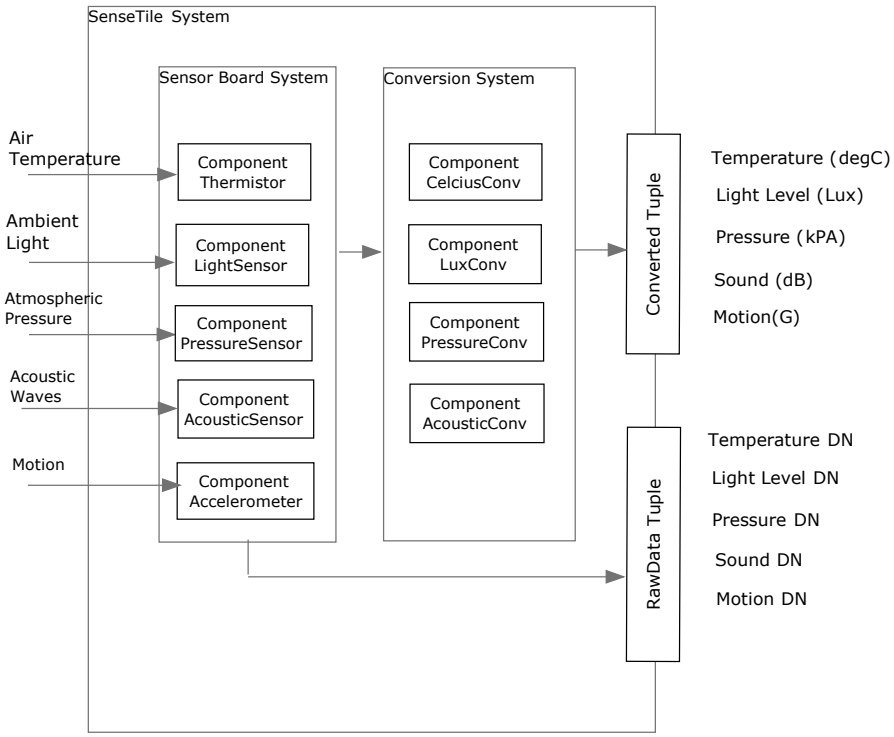

Figure 3.2: SenseTile System Components

3.3 BON to SensorML mapping

Features to be elements or attributes.

Name rule suffix "Attrib" to make it an attribute.

List handling - convention for mapping.

SML prefix to handled key word clashes between sensorML and BON.

How to display this in thesis?

Chapter 4: **Design**

4.1 SenseTile Web Service

4.1.1 Overview

Web Service - SOS -standard access to sensor observations

SOS diagram

DataProvider

DataProducer

Deployment Scenarios - diagram

Flexible Division - several dataproducers can register with one DataProvider.

Seperate processes rmi.

4.1.2 DataProducer

Use PacketInput stream to access SenseTile observation packets.

System reads packets and averages them for every configurable seconds. Default 1.

Thermistor components reads out the temperature value and its output is sent to both the raw data and the celcius converter component and on to the tuples.

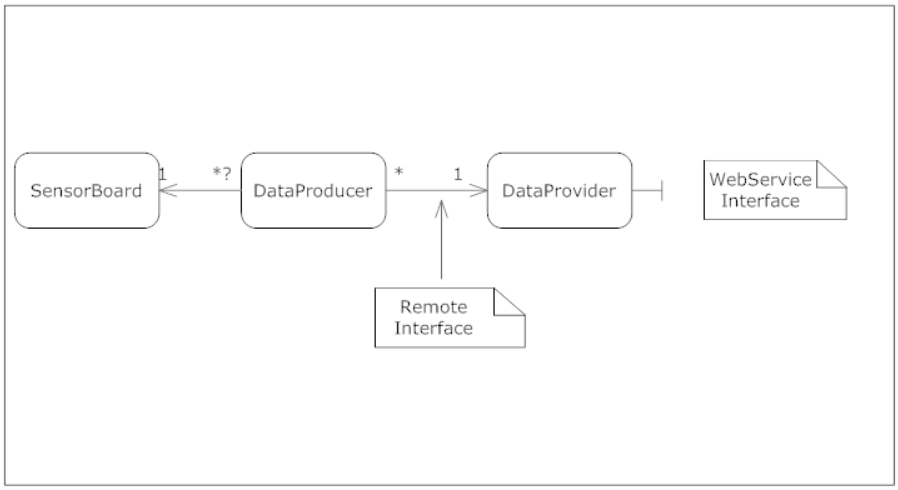

Figure 4.1: Overview

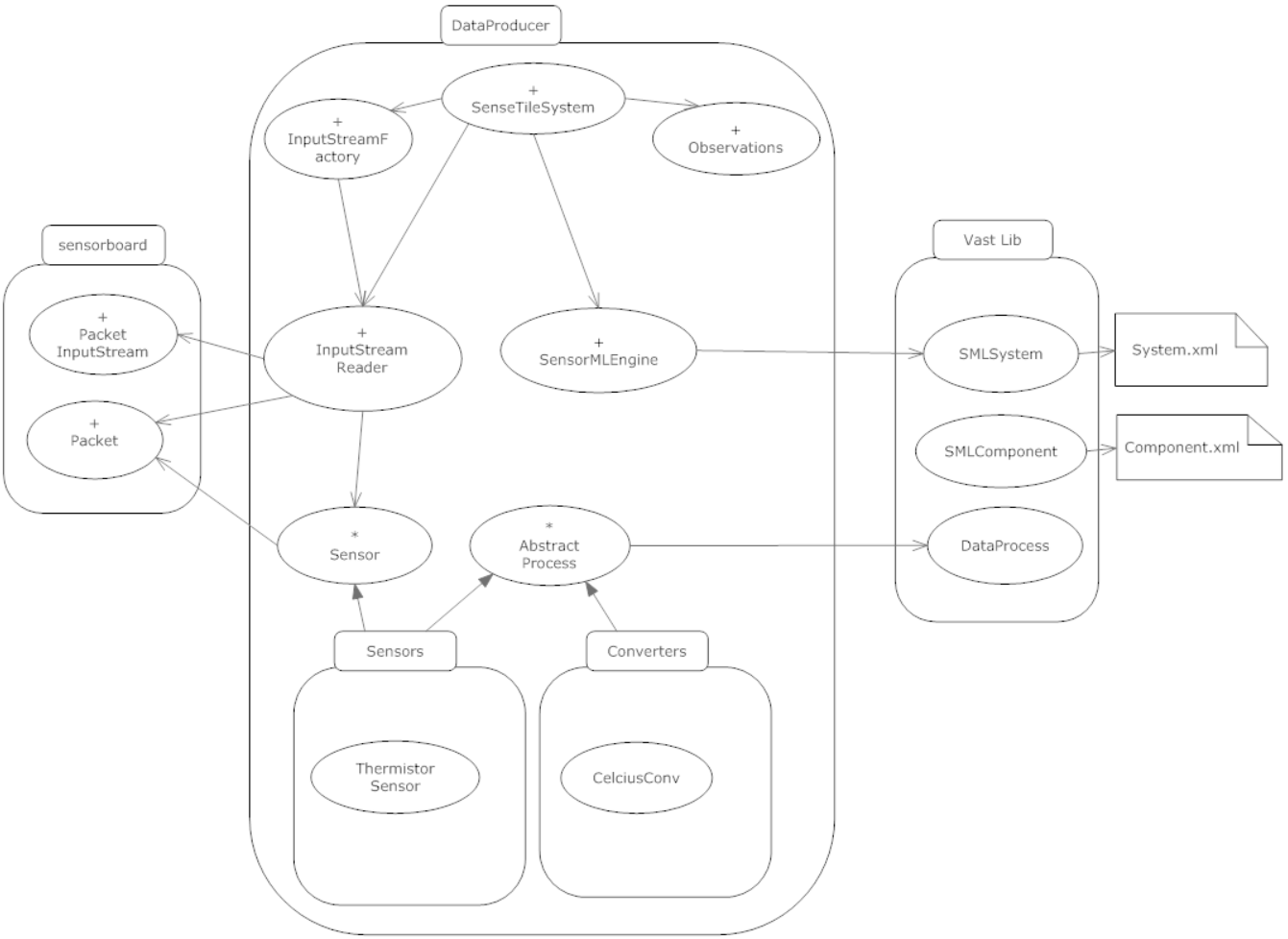

Figure 4.2: DataProducer Static Diagram

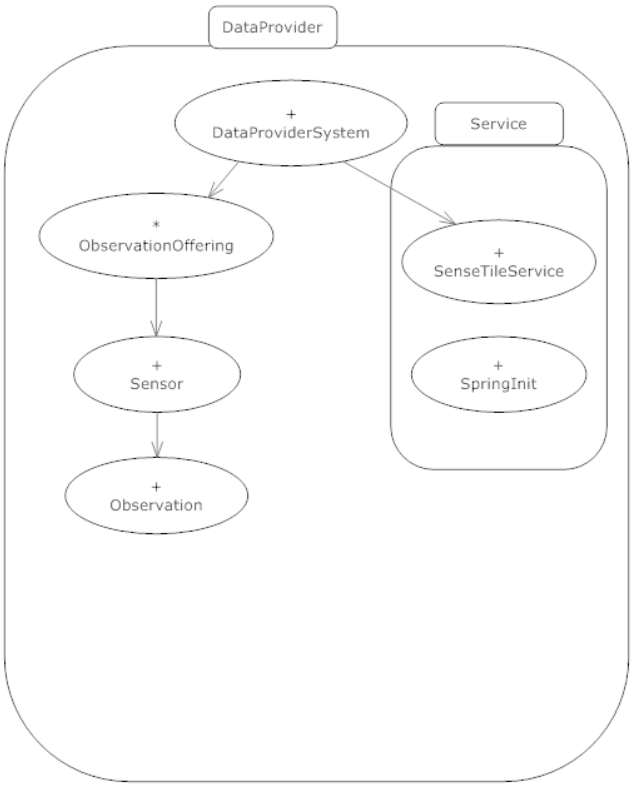

Figure 4.3: DataProvider Static Diagram

These tuples are sent to webservice provider as observations.

Observation as described in SOS.

4.1.3 DataProvider

Keeps track of sensorboards that register

provides an RMI interface to the DataProducer to update with observations.

GetObservaton - marshal Observation to xml string and send as response.

StreamObservation - registered clients get new observation sent over UDP connection?

Provides a Web Service interface based on Sensor Observation Service. Not fully as is a very complex interface.

4.2 SensorML generation

Java binding wth JIBX generate sensorml.

Simple xml file with Sensor metadata.

Naming rules allow connection of inputs and outputs in sensorML processes.

component id +Input

component id+Output

Chapter 5: **Detailed Design and Implementation**

5.1 SenseTile WebService

DataProvider is based on AXIS 2.0/Spring

DataProducer POJO/VastLib/UCD SenseTile Driver Lib

RMI used between DataProvider and DataProducer.

5.2 SensorML Generation

Chapter 6: **Results**

Tested against real SenseTile and sensor data retrieved by a testclient.

Generation of SensorML description of a Senstile Sensor

Chapter 7: **Conclusions and Future Work**

what has been achieved the weaknesses of your approach

Chapter 8: **References**

Bibliography

[1] Open Geospatial Consortium Inc., OpenGIS Sensor Model Language (SensorML) Implementation Specification, 2007

[2] Kim Waldn and Jean-Marc Nerson , "Seamless Object-Oriented Software Architecture", 1995

Chapter 9: **Appendices**