Wireless Sensor Network Data Description and Encoding in Heterogeneous Building Systems

Chris Conway, Matthew Barnes
Centre for Speckled Computing
School of Informatics, University of Edinburgh
Edinburgh EH8 9AB, Scotland, UK
{v1conw1, mbarnes}@inf.ed.ac.uk

Abstract—**This paper presents initial results and future directions for data description and payload encodings for heterogeneous networks in IP based building intelligence systems. The efficiency of several open standards for sensor data representation are evaluated along with a consideration of the complexity of implementation and suitability for use in development of 6lowPAN based intelligent building systems. A combination of a simplified generic XML schema for data description combined with Google Protobuffer encoding is suggested as the basis for further development of the architecture as this combines simple, platform independent implementation for developers with efficient compression and payload encoding for limited bandwidth 6lowPAN networks.**

I. Introduction

The HOBNET project[1] is developing an open IPv6 based architecture for building intelligence systems. The goal of the HOBNET architecture is to use open standards where possible, following a web-service resource oriented model. The architecture is divided into two abstract domains as shown in Figure 1; the building domain and application domain. This allows for high level building control and management components such to be implemented in the application domain, independently from the networks and sensors of the building domain. The interface between the two domains is via a Building Proxy, a software application handling relevant gateway devices to the building domain systems.

The primary component of the building domain are 6lowPAN sensor networks, although the architecture will support a widely heterogeneous system to incorporate legacy Building Management System (BMS) equipment and other wireless networks such as Zigbee via the Building Proxy. The HOBNET architecture will specify a set of standard protocols and formats for communication between heterogeneous systems.

It is intended that 6lowPAN networks will implement the standard set directly while the Building proxy will provide a mapping between BMS and proprietary systems to the selected standards. This paper considers the description and encoding of sensor data in the wireless sensor networks of the building domain. In the rest of the paper, Section II reviews existing efforts in development of WSN based building management. Section III presents initial developments in the specification of payload encoding. Finally Section IV concludes the paper.

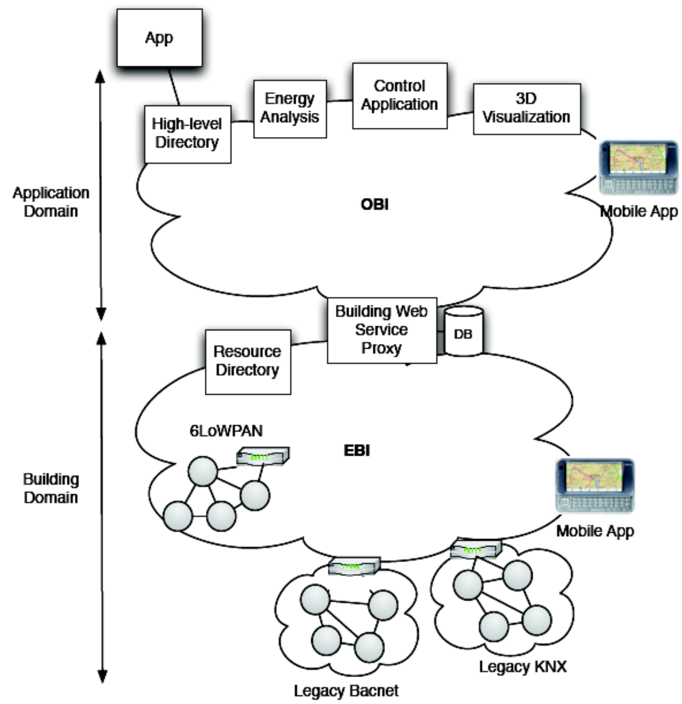

Fig. 1. HOBNET Architecture from (HOBNET D1.3 System Architecture Definition)

II. Related Work

The SENSEI project[2] has developed a resource based architecture to integrate large scale heterogeneous sensor and actuator networks[1], the SENSEI Framework is intended to provide contextual information about the sensor network data providing a platform for a wide range of services and applications. The resource model that is used throughout the framework provides a platform for web based sensor actuator networks however significant alteration would be required to use the SENSEI framework as the basis for a BMS.

[1] www.hobnet-project.eu

[2] www.ict-sensei.org

978-1-4577-0513-7/11/$26.00 ©2011 Crown

Some early work on the use of WSNs for building management is described in [2]. An architecture for WSNs in building intelligence systems was implemented. The interaction between WSN and legacy BMS systems is addressed as it is proposed that WSN based HVAC systems are connected to an existing BMS and the suitability of LonWorks and BACnet in providing an open BMS system are evaluated.

A sensor network architecture for monitoring the energy efficiency of buildings is presented in [3]. The focus of the architecture is on the integration of WSNs with IT systems, the requirements for sensor network communication are assessed. Connection with existing BMS communication buses such as KNX is proposed, however the requirements for this are not addressed in detail.

A sensor network architecture for health monitoring applications is presented in [4]. Although the focus of the proposed architecture is on the integration of heterogeneous Body Sensor Networks (BSN). The design of the architecture satisfies some comparable requirements to those required in a building management architecture as it uses of formal sensor descriptions in a standardised mark-up language (Sensor Ml) and openness, allowing heterogeneous sensors to be added dynamically.

The WSN-BMDS platform [5] is a closed architecture including only sensor networks to implement a BMS. The WSN uses the Global Sensor Network (GSN) middle-ware and EmNetS Network Management Protocol. Integration with legacy BMS is not addressed by the platform.

Global Sensor Networks (GSN)[6] is a middle-ware designed to provide a žero-programmingäpproach for rapid sensor network deployment, the middle-ware uses virtualisation to separate users from low-level device dependant programming. The middle-ware is implemented for specific WSN platforms, the interaction between the platform and sensors is specified using IEEE 1451-compliant Transducer Electronic Data Sheets (TEDS). Users provide an XML specification describing the required operation of the sensor device and the relevant TEDS for the sensor hardware. The GSN middle-ware implements the required sensor network functionality, creating a data stream of time stamped tuples for each sensor that can be processed using SQL commands.

A common feature of these WSN architectures is the use of strongly defined sensor descriptions. A number of description languages have been defined to model heterogeneous sensor and acutator networks including custom XML schema (GSN, SENSEI) and emerging standards such as SensorML and Open Building Information Exchange (oBIX). These languages are used for description of the configuration and offer a firm basis for advancement of future architectures, however, few details are typically provided on the format of data transfer in the network.

Closed platforms often specify their own payload formats and encodings and the interface between sensor networks and other systems such as data management software often also forms part of the platform and so standards for data formats are not specified.

Data representation and encoding should be considered in development of standards based architecture particularly when support for heterogeneity of networks and devices is required. The encoding format will have the greatest impact on the most constrained devices. Wireless sensor network platforms are often constrained in network bandwidth, memory and processing power and although data encoding will not affect performance to the same extent as the operation of communication protocols, some benefit may still be gained and as such some consideration of data representation should be made.

III. Packet Payload Encoding

The aim of the HOBNET architecture is to support heterogeneity through the use of open standards and wherever possible to make use of available resources. Standard protocols such as IP and UDP define packet header formats, however, a standard for cross-platform payload formatting and encoding is required.

A. Data-type Definition

Typically a WSN will provide only minimal information in data transmissions to reduce bandwidth requirements, a simple data description format named *CompactDataRecord* (CDR) is used here as an example. CDRs are used to represent sensor or application data from a node in the sensor network. It is intended that CDRs are generic so that data from multiple sensor types and applications can be encapsulated without modification, although extension or addition of further data types is possible.

The CDR encapsulates a timestamped data value and includes some identification values named sensor-type and sensor-id. The use of two levels of identifier allows each sensor to identified based on a concatenation of *device-address/sensor-type/sensor-id*, this can then be used as a URI for mapping to the resource directory. This format has been used effectively in previous sensor network deployments.

A radio packet payload can contain multiple CDRs and a optional single Time Record, specifying the generation time of the data. Although generic, the XML schema of a radio packet payload is shown in Listing 1; a concrete realisation in XML format is displayed in Listing 5.

To utilise this payload format in heterogeneous network communication a platform and language-independent library is required to provide serialisation and de-serialisation mechanisms. Two candidate encoding schemes are Google Protocol Buffer and Efficient XML Interchange, both are readily available for a variety of platforms and languages and are suited to the requirements of packet payload encoding.

B. Google Protocol Buffer

Google Protocol Buffers [7] are used to encode structured data into an efficient and extensible format for cross-platform communication and storage. Protocol Buffers provide platform-independent, clean access to the underlying messages

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/
    XMLSchema" elementFormDefault="qualified">

<xs:element name="payload">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="time" type="xs:dateTime"
          minOccurs="0" maxOccurs="1" />
      <xs:element name="datarecord" maxOccurs="
          unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="time" type="xs:
                dateTime" minOccurs="0" maxOccurs
                ="1" />
            <xs:element name="raw-value" type="xs:
                short" maxOccurs="1" />
            <xs:element name="sensor-id" type="xs:
                short" maxOccurs="1" />
            <xs:element name="sensor-type" type="xs:
                short" maxOccurs="1" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Listing 1. XML CompactDataRecord schema

to provide a small-scale communication solution for wireless sensor network devices.

Protocol buffer messages are defined by a data description language (DDL) and then compiled into language specific code for message manipulation specifying the native data structures, accessor methods and (de)serialisation mechanisms for the messages. Messages are then encoded into a binary format for communication or compression. For each data item encoded, the binary format contains an indicator relating to the expected data type, the length of the data, and an identifier for the data item.

Implementations of the Protocol Buffer compiler exist for all common programming languages and the native code can be cross-compiled to multiple hardware implementations.

Protocol Buffer's DDL provides multiple scalar data types, each with a language-specific mapping, and can additionally contain multiple nested message types. Furthermore, messages can be extended in a hierarchical manner and can import other messages.

C. Protocol Buffer Description Language Example

Protocol Buffer messages are described using a DDL. A realisation of the CompactDataRecord data definition described in Section III-A is shown in Listing 2.

```
// define the main payload
message DataRecordPayload {
  optional TimeRecord time = 1;
  repeated DataRecord records = 10;

  message DataRecord {
    required int32 raw_value = 1;
    required int32 sensor_id = 2;
    required int32 sensor_type = 3;
  }
}

// describe the reusable time record
message TimeRecord {
  required bytes calendar = 1;
}
```

Listing 2. Example Protocol Buffer message description

D. Protocol Buffer Encoding Example

Listing 3 details the Protocol Buffer encoding of the XML payload shown in Listing 5.

```
0xa 0x8 0xa 0x6 0x0 0x14 0x10 0x3 0x3 0xb
0x52 0x7 0x8 0xa 0x10 0xb 0x18 0xff 0x10x52
0x7 0x8 0x16 0x10 0xa 0x18 0xff 0x1 0x52 0x7
0x8 0x63 0x10 0x12 0x18 0xab 0x1
```

Listing 3. Example Protocol Buffer message encoding

E. EXI

Efficient XML Interchange (EXI) [8] is a compact representation of the XML Information Set that encodes XML documents from plain text into a binary data format. EXI is intended to simultaneously optimise performance and utilisation of computational resources on constrained devices.

EXI uses a simple algorithm and a small set of data types to reliably produce encodings of XML event streams. The EXI specification allows for two modes: schema-informed and schema-less. Schema-less requires the structure of the data to be included, much like XML. Schema-less uses a reduced grammar and state-machine limited to the schema enabling an extremely efficient encoding.

F. Example EXI Payload

Listing 4 details the EXI encoding of the XML payload shown in Listing 5 using schema-informed, byte-aligned mode.

```
0x90 0x0 0x59 0xf9 0xcb 0x60 0x0 0x28 0x2
0xc3 0xfc 0x4 0x0 0xb0 0x5 0x7 0xf8 0x8 0x6
0x30 0x12 0xa 0xb0 0x11
```

Listing 4. EXI Payload of XML implementation

G. Compression Results

Using the XML schema from Listing 1, a payload was encoded using the aforementioned techniques. The results from these encodings are presented in Table I.

As expected, XML encoding produced the largest sized payload and will therefore be used as the base for determining the compression ability of each technique.

EXI was examined twice: once in schema-less and again in schema-informed mode. The schema-less encoding produced a payload that was 27% of the size of the XML payload. Schema-informed encoding produced the smallest payload - 6% of the size of XML.

Protocol Buffers produced a payload 9% of the size of the XML payload. Due to Protocol Buffers including some basic information regarding the data, such as the unique identifier of the data item in the payload to allow out of order decoding, there is a slight overhead compared to schema-informed EXI.

```xml
<?xml version="1.0" encoding="utf-8"?>
<payload>
  <time>2011-03-30T16:20:00</time>
  <datarecord>
    <raw-value>10</raw-value>
    <sensor-id>11</sensor-id>
    <sensor-type>255</sensor-type>
  </datarecord>
  <datarecord>
    <raw-value>22</raw-value>
    <sensor-id>10</sensor-id>
    <sensor-type>255</sensor-type>
  </datarecord>
  <datarecord>
    <raw-value>99</raw-value>
    <sensor-id>18</sensor-id>
    <sensor-type>171</sensor-type>
  </datarecord>
</payload>
```

Listing 5. XML Compact DataRecord implementation

TABLE I
ENCODING TECHNIQUE RESULTS

Encoding Technique	**Payload Size**
XML	404B
EXI (Schema-less)	111B (27%)
EXI (Schema-informed)	24B (6%)
Protocol Bufffer	37B (9%)

H. Implementation Complexity

Further to encoding a payload into the smallest amount of data, the complexity of implementing a technique must be considered due to the constrained nature of the target devices.

EXI requires a four-stage process to convert an XML document into EXI: a finite state machine is built using a Simple API for XML (SAX) parser, with events as transactions; using the XML schema the finite state machine is extended to include additional information about elements; the finite state machine is used to convert the input XML file into a SAX event stream; the event stream is processed and the finite state machine outputs the encoded values. A SAX parser implementation is therefore required for the sensor node, this requires significant program memory and RAM to implement.

Protocol Buffer uses a pseudo-code like schema, as shown in Section III-F. These messages are compiled into language specific code using the Protocol Buffer compiler, providing helper and accessor methods for the messages as well as marshalling and unmarshalling mechanisms. The generated code is compiled and linked with the application to provide efficient, clean access to the records. Protocol Buffer therefore requires platform-specific code to be generated for each sensor node platform and linked to the firmware. Whilst this requires an additional step in compilation, the memory and processing requirement is reduced compared to EXI.

IV. CONCLUSION

This paper has presented ongoing work in the development of a data description and encoding scheme for use in a resource oriented architecture for heterogeneous building intelligence systems. Current evaluation has focussed on the requirements of radio packet payload encoding for transmissions in wireless sensor networks. A simple XML schema was used to trial EXI and Protocol Buffers as a suitable encoding format. It is suggested that whilst EXI produces a smaller sized encoding, Protocol Buffers may be more appropriate due to lower complexity in the implementation of the encoding software.

ACKNOWLEDGEMENTS

This work was supported by the EU/FIRE HOBNET project - STREP ICT-257466.

REFERENCES

[1] C. Villalonga, M. Bauer, V. Huang, J. Bernat, and P. Barnaghi, "Modeling of sensor data and context for the real world internet," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 1–6, IEEE, 2010.

[2] A. Sharma, "Design of wireless sensor networks for building management systems," Master's thesis, University of California at Berkeley, USA, 2003.

[3] K. Menzel, D. Pesch, B. O'Flynn, M. Keane, and C. O'Mathuna, "Towards a wireless sensor platform for energy efficient building operation," *Tsinghua Science & Technology*, vol. 13, pp. 381–386, 2008.

[4] A. Triantafyllidis, V. Koutkias, I. Chouvarda, and N. Maglaveras, "An open and reconfigurable wireless sensor network for pervasive health monitoring," in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pp. 112–115, IEEE, 2008.

[5] R. Spinar, P. Muthukumaran, R. de Paz, D. Pesch, W. Song, S. Chaudhry, C. Sreenan, E. Jafer, B. OFlynn, J. ODonnell, *et al.*, "Demo Abstract: Efficient Building Management with IP-based Wireless Sensor Network," in *6th European Conference on Wireless Sensor Networks, Cork*, 2009.

[6] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *Proceedings of the 32nd international conference on Very large data bases*, pp. 1199–1202, VLDB Endowment, 2006.

[7] R. Pike, S. Dorward, R. Griesemer, S. Quinlan, and G. Inc, "Interpreting the data: Parallel analysis with sawzall," in *Scientific Programming Journal*, 2005.

[8] W3C , "Efficient XML Interchange (EXI) Format 1.0, W3C Recommendation," 2011.