Capturing Personal Health Data From Wearable Sensors*

Fabrice Camous, Dónall McCann and Mark Roantree
Interoperable Systems Group
Dublin City University, Ireland
fcamous,dmccann,mark@computing.dcu.ie

1. Introduction

Recently, there has been a significant growth in pervasive computing and ubiquitous sensing which strives to develop and deploy sensing technology all around us. We are also seeing the emergence of applications such as environmental and personal health monitoring to leverage data from a *physical* world.

Most of the developments in this area have been concerned with either developing the sensing technologies, or the infrastructure (middleware) to gather this data and the issues which have been addressed include power consumption on the devices, security of data transmission, networking challenges in gathering and storing the data and fault tolerance in the event of network and/or device failure [12]. Research is focusing on harvesting and managing data and providing query capabilities.

1.1 Motivation and Contribution.

Monitoring devices are already used in Sport Science to monitor athletes. Additionally, they are becoming increasingly available to individuals who take a pro-active interest in managing their own health. Most of the time, sport and health monitoring devices are used with their own distinct proprietary software. Consequently, the user has to manually make sense of the data and integrate data from the different sources. From our collaboration with Sport Scientists, we believe that this is not taking place and data is lost or stored in a form that is unusable. In this paper, we present the HealthSense prototype for harvesting and integration of physical data from wearable sensors. Our contribution is in the automated enrichment of data from any sensor and integration and management of data to provide a query interface for knowledge workers.

The paper is structured as follows: §2 describes our architecture; §3 presents our experiments and reports on processing times; in §4 we discuss related work, and finally in §5, we provide conclusions.

*The RSS SENSE project is funded by Enterprise Ireland Ref. PC/2007/112.

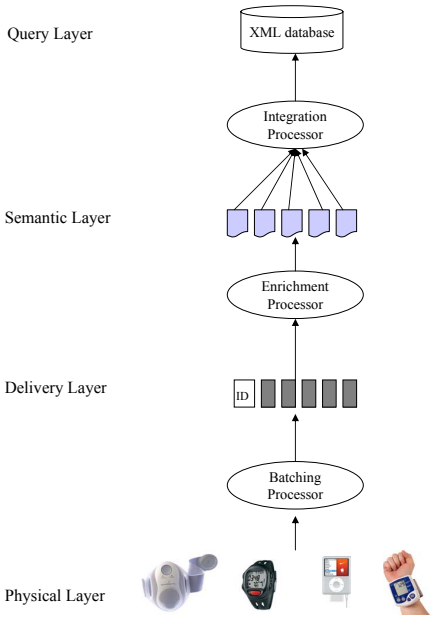

Figure 1. System Architecture

2. The HealthSense Architecture

In this section, we describe the HealthSense architecture which provides the platform for transformation from physical sensor readings recorded in the physical world to digital information in XML databases, through processes of enrichment and integration. There are 4 layers, representing data in different formats, separated by the 3 processors that transform them. We now describe the 3 processors in more details.

2.1 Batching Processor

The role of the batching processor is to ensure that data from multiple sensors that are used in the same activity are grouped together. The processor also associates each activity with an identifier corresponding to the individual being monitored. While there is little innovation in this processor, it is a necessary step to ensure data is harvested in the correct manner.

2.2 Enrichment Processor

The role of the enrichment processor is to convert sensor data to an XML format that contains the semantics to describe each of the values generated by the sensor.

Sensor data files are written and designed to be read by machines, and so there is a finite number of ways in which the data can be represented (e.g. a simple list of values or a csv file). The enrichment processor has defined a template approach to enrichment by examining data generated from a diverse set of sensor devices across environmental, food monitoring and personal health domains.

Each sensor device has a small XML template file associated with it. The enrichment processor combines the template and the output from the device to create an XML document that can be queried using the XPath query language. If the system encounters a new sensor, all that is required is the template file. The template describes the sensor's output in terms of its structure and the location of certain key elements, such as start time and measurements, as well as important data such as value delimiters. The output file contains user, session and device ID information, followed by sensor-specific information and, finally, a list of labeled measurement values. Figure 2 shows files that contain enriched data from the bodymedia and heart rate sensors.

2.3 Integration Processor

It is often necessary for the domain expert who monitors an activity to be interested in the combination of simultaneous measurements by various sensors. For example, medical staff may be alerted whenever the heart rate goes above a certain level and whenever the skin temperature goes above a certain level during the same session. In this paper, the integration process merges individual sensor files according to the temporal dimension. However, our prototype is open to multi-dimensional integration and on-going work is evaluating integration across different parameters (participant data, sensor types and sensor readings. In effect, by integrating on time, we create a fast integration that is easy to specify (as time is a parameter across all devices). Figure 3 shows a subset of the file generated by the integration process.

```
<user>
  <id>1</id>
  <session>
    <id>1</id>
    <sensorData deviceID="bsd">
      <startTime>1195226100000</startTime>
      <measurement time = "1195226160000"
                    type="BodyMedia SenseWear Data">
        <skin_temp_average_original_rate>30.126686096191406
        </skin_temp_average_original_rate>
        <energy_expenditure_per_minute>1.4301998615264893
        </energy_expenditure_per_minute>
      </measurement>
    </sensorData>
  </session>
</user>

<user>
  <id>1</id>
  <session>
    <id>1</id>
    <sensorData deviceID="hrm">
      <startTime>1195226100010</startTime>
      <measurement time="1195226130000" type="HRData">
        <HRValue>91</HRValue>
      </measurement>
      <measurement time="1195226160000" type="HRData">
        <HRValue>95</HRValue>
      </measurement>
      <measurement time="1195226190000" type="HRData">
        <HRValue>97</HRValue>
      </measurement>
    </sensorData>
  </session>
</user>
```

Figure 2. Bodymedia and HeartRate enriched data

3 Experiments

The datasets used in this paper are a subset of a dataset obtained from experiments running from October 2007 which simulate the health monitoring of a group of individuals with biometric sensors. The group comprises 12 people, aged 26 to 47, and includes an equal number of males and females. Data is collected through USB connections using the following sensors:

- **Polar S625X™ heart-rate monitor**. This consists of a fabric band which fits around a person's chest and detects and logs their *heart rate*. It also includes a foot pod that accurately captures speed/pace and the distance covered. We set the sensor to sample data every 5 (14h memory capacity) or 15 seconds (44h memory capacity).
- **BodyMedia SenseWear®**. This sensor array is worn around the upper arm and measures and logs the following: *galvanic skin response*, a measure of skin conductivity which is affected by perspiration; *skin tem-*

Table 1. Description of the datasets

Dataset Number	User ID	Sensor Files	Raw Size	**Enrichment Times ms**	**Enrichment Size**	**Integration Times ms**	**Integration Size**
1	1	1 Nike iPod (8KB) 1 Polar (3KB) 1 BodyMedia (17KB)	28k	175.2	141.7k	32	133.6k
2	12	1 Nike iPod (7k) 3 Polar (7k) 1 BodyMedia (35k) 1 Blood Pressure (1k)	50k	307.8	261.7k	56.4	252.7k
3	6	3 Polar (12k) 1 BodyMedia (144k) 1 Blood Pressure (2k)	158k	529	943.5k	188	914.1k
4	4	4 Nike iPod (34k) 2 Polar (3k) 1 BodyMedia (227k) 1 Blood Pressure (2k)	266k	499.2	1.3M	289.2	1.3M
5	9	3 Nike iPod (23k) 4 Polar (34k) 1 BodyMedia (250k) 1 Blood Pressure (7k)	314k	756.4	1.9M	417.4	1.8M
6	7	8 Nike iPod (83k) 6 Polar (63k) 1 BodyMedia (561k) 1 Blood Pressure (3k)	710k	1098.6	4.2M	845	4M
7	7	2 Polar (32k) 1 BodyMedia (1.36M) 1 Blood Pressure (3k)	1.37M	1334.6	7.5M	1239.2	7.3M
8	1	2 Polar (44k) 2 BodyMedia (11.04M)	11.09M	5392.6	67.4M	9946.8	63.7M

Figure 3. Example of integrated data

```
<user>
  <id>1</id>
  <session>
    <id>1</id>
    <sensorData deviceID="bsd">
      <startTime>1195226100000</startTime>
      <measurement time="1195226130000">
        <HRValue>91</HRValue>
      </measurement>
      <measurement time = "1195226160000">
        <skin_temp_average_original_rate>30.126686096191406
        </skin_temp_average_original_rate>
        <energy_expenditure_per_minute>1.4301998615264893
        </energy_expenditure_per_minute>
        <HRValue>95</HRValue>
      </measurement>
      <measurement time="1195226190000">
        <HRValue>97</HRValue>
      </measurement>
    </sensorData>
  </session>
</user>
```

perature, which is linearly reflective of the body's core temperature activities; *heat flux* which is the rate of heat being dissipated by the body; *subject motion* using an in-built accelerometer. We set the sensor to collect data once every minute (research configuration, 34h memory capacity), or 12 times a minute (channel configuration, 247h memory capacity).

- **Deluxe Wrist Blood Pressure Monitor HL168JC**. This device can store up to 90 blood pressure and pulse readings.
- **iPod Nano 4G with Nike®+ IPod Sport kit**. This sensor includes a foot pod and a detector connected to the IPod Nano. Distance covered during a walk or run and caloric consumption are collected along with the type of music listened to.

A total of 8 datasets corresponding to sessions (activity associated with several sensor files) from different users were selected. Table 1 describes the content of each dataset. The experiments were evaluated on a single workstation running the Windows XP Pro operating system on 2.66GHz Intel Core 2 Duo processors with 4 GB of RAM. The system was implemented using the Java Virtual Machine 6.0.

3.1 Result and Analysis

When enriched into XML format, the sizes of the files increase, on average, by a factor of just below 6. From the data, this appears to be largely independent of the file size, with the smallest file increasing by a factor of 5.45 and the largest increasing by a factor of 6.12. Furthermore, we found that our generic approach (loading template files rather than writing purpose-built drivers for every sensor device) did not cause a significant delay. For example, the smallest dataset is available in less than 0.2 seconds.

From our study, the exact amount by which a file increases when converted to XML is directly dependent on the complexity of the schema of the output file, as well as the size of the raw sensor data values. In general, times increased in a linear fashion as file sizes increased. However, at some point in file sizes (approx. 2MB), the enrichment process becomes faster than the integration process. Indeed, the enrichment process becomes more efficient when file sizes are larger. Looking at dataset 2 in Table 3, the enrichment process requires opening 6 sensor streams and 6 template files, and writing 6 new enriched files, a total of 18 file I/O operations. On the other hand, the integration process opens the 6 input files and only a single file for output. Thus, we discovered the enrichment process itself is faster than the integration process, but this is hidden for small files

by the times for disk I/O.

Overall, the time to transform sensor data to a queryable format is fast. For 11MB of sensor data, it requires a little over 15 seconds to transform to a single XML document.

4 Related Research

The work described in this paper is by nature interdisciplinary and aims at building bridges between various research communities (Sensors, ICT, Health/Medical Informatics). In this section we first mention related work in semantic enrichment. As we intend to include an XML stream query Engine in the future, we also present the latest approaches to XML stream querying.

The use of data semantic enrichment for interoperability can be found in the research literature[8, 13]. Nonetheless, the enrichment is addressed in the context of a traditional data management system. In contrast, we enrich data on-the-fly to accommodate the new data stream management system paradigm. In particular, our semantic enrichment process is building on previous work[11].

Most XML stream query engines focus on efficiently processing an XML stream with one or several queries, expressed in an XML query language such as XPath or XQuery, on a single machine or peer. AFilter[6], XFilter[1], YFilter[7], XMLTK[2] and SFilter[10] address the scenario of filtering a stream for multiple subscribers and are concerned with the scalability to the number of subscribers. On the other hand, XPA[4], $\chi\alpha o\zeta$[3], similarly to SPEX[5], concentrate on the challenges of implementing key features or remaining aspects of the XPath query language.

While there has been a considerable volume of work on developing XML stream query engines, more effort is needed to integrate XML stream filtering into a real-life infrastructure. Additionally, XML stream filtering needs to be approach in a distributed environment in order to manage scale and to improve query performance. In future work, we will address the problem of distributing the filtering of XML streams, similarly to [9].

5 Conclusions

In this paper, we described our framework for capturing sensor data and transforming this data into a queryable format by enriching and then integrating related sensor outputs. Sensor files were shown to grow 5 times in size after enrichment, but despite this, even over 60MB of enriched data is available for query in 15 seconds. This demonstrates a close to real time database creation. Our current research tackles two other issues: scalability and dynamic querying. To manage an increasing volume of sensors, we introduced a Peer-to-Peer system to manage scalability with multiple sensor databases. Our current research also places query filters throughout the network to quickly identify sensor output of interest to knowledge workers.

References

[1] M. Altinel and M. J. Franklin. Efficient Filtering of XML Documents for Selective Dissemination of Information. In *Proceedings*, pages 53–64. VDLB 2000, 2000.

[2] I. Avila-Campillo, T. J. Green, A. Gupta, M. Onizukaz, D. Raven, and D. Suciu. An XML Toolkit for Light-weight XML Stream Processing. In *Proceedings*, Pittsburgh, PA, October 2002. PLAN-X 2002.

[3] C. Barton, P. Charles, D. Goyal, M. Raghavachari, M. Fontoura, and V. Josifovski. Streaming XPath Processing with Forward and Backward Axes. In *Proceedings*, pages 455–466, Bangalore, India, March 2004. ICDE 2003.

[4] S. Bottcher and R. Steinmetz. Evaluating XPath Queries on XML Data Streams. In *Proceedings*, pages 101–113, Glasgow, Scotland, July 2007. BNCOD 2007.

[5] F. Bry, F. Coskun, S. Durmaz, T. Furche, D. Olteanu, and M. Spannagel. The XML Stream Query Processor SPEX. In *Proceedings*, pages 1120–1121, Tokyo, Japan, April 2005. ICDE 2005.

[6] K. S. Candan, W. Hsiung, S. Chen, J. Tatemura, and D. Agrawal. AFilter: Adaptable XML Filtering with Prefix-Caching and Suffix-Clustering. In *Proceedings*, pages 559–570, Seoul, Korea, September 2006. VLDB 2006.

[7] Y. Diao and M. J. Franklin. High-Performance XML Filtering: An Overview of YFilter. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 26(1):41–48, 2003.

[8] U. Hohenstein and V. Plesser. Semantic enrichment: A first step to provide database interoperability. In *Workshop Fderierte Datenbanken*, pages 3–17, Magdeburg, 1996.

[9] R. Kuntschke, B. Stegmaier, A. Kemper, and A. Reiser. StreamGlobe: Processing and Sharing Data Streams in Grid-Based P2P Infrastructures. In *Proceedings*, volume 2, pages 1259–1262, Trondheim, Norway, 2005. The 31st VLDB Conference.

[10] D. Lee, H. Shin, J. Kwon, W. Yang, and S. Lee. SFilter : Schema based Filtering System for XML Streams. In *Proceedings*, Seoul, Korea, 2007. 2007 International Conference on Multimedia and Ubiquitous Engineering(MUE'07).

[11] N. Legeay, M. Roantree, G. Jones, N. O'Connor, and A. F. Smeaton. *Semi-Automatic Semantic Enrichment of Raw Sensor Data*, volume 4805, pages 13–14. Springer, Berlin / Heidelberg, 2007.

[12] M. Stonebraker, U. Cetintemel, and S. Zdonik. The 8 Requirements of Real-Time Stream Processing. In *Proceedings*, Tokyo, Japan, April 2005. 21st International Conference on Data Engineering (ICDE'05).

[13] X. Su, S. Hakkarainen, and T. Brasethvik. Semantic enrichment for improving systems interoperability. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1634–1641, New York, NY, USA, 2004. ACM.