

Using Sensor Networks to Measure Intensity in Sporting Activities*

Mark Roantree¹, Michael Whelan², Jie Shi¹, and Niall Moyna²

¹ Interoperable Systems Group, Dublin City University
mark, jshi@computing.dcu.ie,

² School of Health and Human Performance, Dublin City University
niall.moyna, michael.whelan@dcu.ie

Abstract. The deployment of sensor networks is both widespread and varied with more niche applications based on these networks. In the case study provided in this work, the network is provided by two football teams with sensors generating continuous heart rate values for the duration of the activity. In wireless networks such as these, the requirement is for complex methods of data management in order to deliver more and more powerful query results. In effect, what is required is a traditional database-style query interface where domain experts can continue to probe for the answers required in more specialised environments. This paper describes a system and series of experiments that requires powerful data management capabilities to meet the requirements of sports scientists.

Key words: Wireless Sensor Network, Data Synchronisation, Calibration, Query Service

1 Introduction

Sensor networks have become more varied with many niche applications based on a wide variety of application areas. Unobtrusive sensors are now commonly used to assess the physiological responses during individual and teams sports. The measurement of heart rate to assess the physiological load during individual and team sports is widely accepted within the sporting community [13, 2]. Ambulatory telemetric equipment such as the wireless Polar Team Heart monitor [12] used in this study has made it possible to innocuously monitor heart rates during team sports. Relative exercise intensity can be estimated [1] by processing and manipulating the output from heart rate monitors as this is commonly used as a measure of exercise intensity during a game of soccer [2, 13].

Gaelic football [14] is the most popular sport in Ireland. It is a hybrid of Rugby and Australian Rules football. This project assessed heart rate responses during small sided and regular Gaelic football games in young players. To achieve this, we created a wireless sensor network that has multiple configurations and

* Partially Funded by Enterprise Ireland Grant CFTD-2008-231

requires a sophisticated data management layer to process, normalise and query the data streams.

This paper is structured as follows: in the remainder of this section we provide the motivation and contribution for our research; in §2, we describe the sensor network in terms of components and different configurations; in §3, we introduce a DataSpace architecture that provides the platform for data management in the sensor network; in §4, we describe an application to harvest data from sensor networks; in §5, a set of experiments with analysis of the results are given; in §6, we provide details of related research; while §7 offers conclusions.

Table 1. Heart Rate Training Zones

Perc.	Zone	Description	Typical Range
Rest to 60%	Resting	Walking Pace	RHR to 120
60%-70%	Recovery	Develops basic endurance and aerobic capacity.	120 to 140
70%-80%	Aerobic	Develops the cardiovascular system.	140 to 160
80%-90%	Anaerobic	Develops the lactic acid system.	160 to 180
90%-100%	Maximal	Training in this zone is optimal for development of players' aerobic capacity but is possible only for short periods.	180 to 200

1.1 Requirements and Motivation

To accurately determine the intensity at which each player is working during a game or training session, a calculation of each player's maximum heart rate (MHR) and resting heart rate (RHR) must be determined. Resting heart rate was determined following a 5 minute rest period and Maximal heart rate was determined using a field based test.

Once again, heart rate data can also indicate the amount of time spent in different intensity zones (table 1) but only with adequate data management techniques and a flexible query interface. Such a query interface will require user interaction with the sensor network and not a series of 'built-in' queries. Specifically, the development of a quick accurate measurement of the amount of time spent in each training zone is an important factor in determining the primary energy source utilised during games and training. In addition, it may also facilitate coaches in developing individual physiological profiles for each player. This will allow coaches to design and implement appropriate individual training programmes.

The requirement for an infrastructure to monitor and optimise players' performances led to the development of a wireless sensor network that was configured for each experiment as described in section §2. While the network provided the participants and hardware, it was then necessary to provide the data management layer in order to process and calibrate data generated by the networks. The motivation is to provide a traditional query interface for the low level data generated by the wireless network. The research described in this paper is a result of a collaboration between the Interoperable Systems Group (ISG) and the

School of Health and Human Performance, both at Dublin City University in Ireland. While the usage of XML to provide interoperability for sensor networks is gaining in popularity, XML has well known performance issues [6] [11] and the manipulation of this sensor network requires normalisation and continuous recalibration of the sensor data. Thus, any approach that uses XML as part of the data management level must demonstrate that queries times are within an acceptable level of performance.

1.2 Contribution

The construction of a sensor network comprising athletes and heart rate monitors, and configured in different ways provides a strong foundation for analysing the performance characteristics of athletes and in building personal and team-based physiological profiles. However, this represents the physical layer in the solution and a software layer comprising all of the data management and query performance aspects is still required. In this paper, we describe those data management components that facilitate user manipulation and analysis. Our classification and calibration services are the key enablers in the provision of a robust query service and these will be discussed in detail, providing algorithms for locating only meaningful data. Furthermore, we believe our usage of a DataSpace Architecture [5] facilitates the required heterogeneity in data management for sensors networks where data arrives in multiple formats, requiring multiple processors and different processing logic. Our prototype and experiments will demonstrate the speed at which data becomes available to users and the response times they can expect from queries.

2 Wireless Sensor Network Configurations

In this section, we describe the various configurations of the wireless sensor network and outline the underlying scenarios giving rise to these configurations. The actual experiments involved a number of school-aged football teams playing Irish Gaelic football where each player wore a heart monitor that monitored and broadcasted their heart-rate values, every 5 seconds, to a base station. The configuration of the sensor network was determined by the context for each experiment. These are the contexts in which players wore the heart monitors.

- 15-a-side games. Each team consists of one goalkeeper, six defenders, two midfielders and six attackers with the likely configuration displayed in figure 1. The network operates for the lifetime of a single football match with sensing commencing just before, and terminating just after, the match itself. Data outside the First Half and Second Half is to be identified and eliminated from user queries.
- 9-a-side games. Each team consists of one goalkeeper, three defenders, two midfielders and three attackers. As for 15-a-side matches, only data for the First and Second halves should be used in analysis.

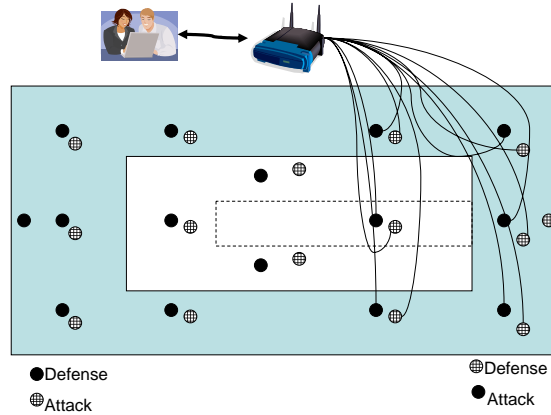


Fig. 1. Network Configurations

- Bangsbo test. The Bangsbo endurance test [1] has a series of levels of increasing intensity. Participants progress through the levels until fatigue forces them to drop out. When the last participant is eliminated, the sensor network terminates.

From the descriptions above, it is clear that networks may broadcast for set durations or may run for an unknown duration in the case of the Bangsbo test. In addition, the playing surface area differs for all three configurations as illustrated in figure 1. This diagram shows 3 different surface areas: the outside (largest) area is for 15 a side games and will contain 30 nodes for the 15 players on each team; the middle area is for 9-a-side and will contain 18 nodes while the smallest (inner strip) can contain up to 50 nodes as a larger number of players may participate in the training activity in this small area. The user scenario expressed in figure 1 is for an analysis of the 6 attacking players and the defenders who are closely tracking them.

3 Sensor Web Architecture

The concept of the DataSpace system was introduced in [5] as a solution to organisations such as healthcare or sport scientists who have a requirement for large numbers of diverse but interrelated data sources. In this section we describe the major components of the HealthSense DataSpace system and how each component contributes to the information management process.

3.1 Data Capture

The Data Capture Component comprises both data sources and a metabase that is used for understanding the content and semantics of the actual data sources.

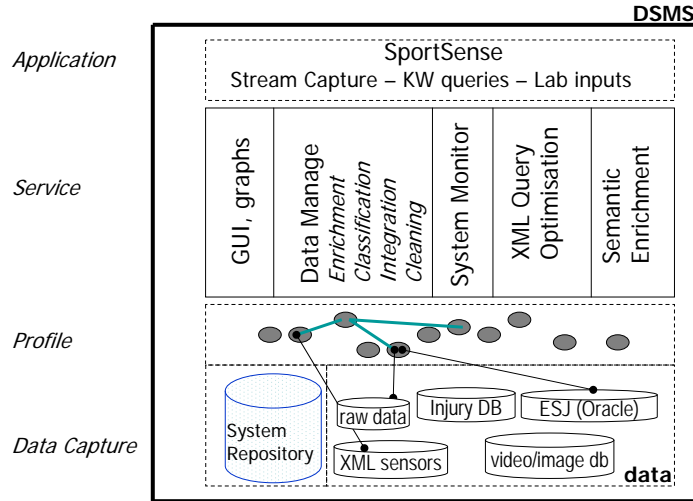


Fig. 2. SportSense DataSpace Architecture

The key difference between the DataSpace architecture and more traditional distributed architectures is that the data is *subject* oriented, similar to a Data Warehouse. In this DataSpace, sensor data exists in both raw format (binary or textual files) and in an enriched XML format. The raw format is necessary for live queries as the converted files are not available quickly enough for this purpose [11]. An Object-relational model (Oracle 10g) is used for creating subject (athlete or patient) profiles and is also used for managing injury data. A separate system is used to store video data while the video metadata is stored in a relational database. Unlike federated database systems that are created from existing application databases, this is a green-field architecture with some level of control to make integration of data easier. However, there will always be situations where unplanned integration will be necessary. For example, combining the output from new sensor devices with previously generated data.

DataSpace Repository The repository for the HealthSense DataSpace provides the engine for the DataSpace system and has a complex metamodel. As with the DataSpace System itself, the System Repository (or metabase) also adopts a hybrid storage model structure. This is necessary as some of the constructs involved are not suited to traditional storage systems. While a full description of the repository and its Metadata Service form part of a separate body of work [11], we provide a brief description of the major components now.

– **Integrations.** Relationships across separate data sources with semantics for integration.

- **Profiles.** For each user or user type, a profile is created that links the user to specified data sources. It may provide a link to Integration objects where users are managing data from multiple sources.
- **Templates.** Templates are used to describe raw sensor sources. Together with a structural enrichment process, they form XML schemas and are used to populate these schemas with raw data so that they can be queried using XPath or XQuery [15]. Their goal is to ensure that the service components are never required to change.
- **Contexts.** While template objects provide for the creation of XML data from raw sensor output, this provides only a structural enrichment of the sensor data. With Context objects, it is possible to semantically enrich the file. Contexts provide the necessary background to understand the situation in which each sensor was used. For example, a Heart Rate monitor can be used in a match situation: football, tennis or athletics; or it may be used in testing scenarios such as the Bangsbo test [2].
- **Schemas.** Schema objects are stored for XML databases only. They provide the user with a storage model version of raw sensor data and enable the user to formulate queries. There is a strict one-to-one mapping between templates and schemas.
- **Replicas.** There are many examples of data replication in the DataSpace system and these are modelled in the system repository. For example, the optimiser will create a relational index of an XML database; multimedia metadata is created for video files; XML views are created from object-relational databases for the purpose of sharing data.

3.2 Profile Component

HealthSense is a Web Information System in that web browsers provide the interface to multiple sources of data, and HTML or XML is used as an interface between heterogeneous data collections and users. What connects user types with different requirements, to one or more data sources are the profiles. Many profiles are simple to construct: a physiotherapist will only query and manage data from a single database (Injury DB) or a knowledge worker who wants to average team heart rate data after each match. However, some of the profiles are more complex: knowledge workers wishing to mine larger data volumes are searching for relationships between training regimes (Electronic Sports journal), maximal heart rate and injuries sustained.

Semantic Enrichment. This service creates the XML version for all sensor streams using a template approach [3] that requires no modification to system components when new sensors are introduced. Initially, raw sensor streams are structurally enhanced to produce basic XML files and subsequently, these files are mined to generate additional semantics for every sensor reading. In all sporting experiments, we apply *state* information to each sensor reading, where a state refers to some interval in either a sporting event (eg. football game or tennis match) or a lab-based training activity.

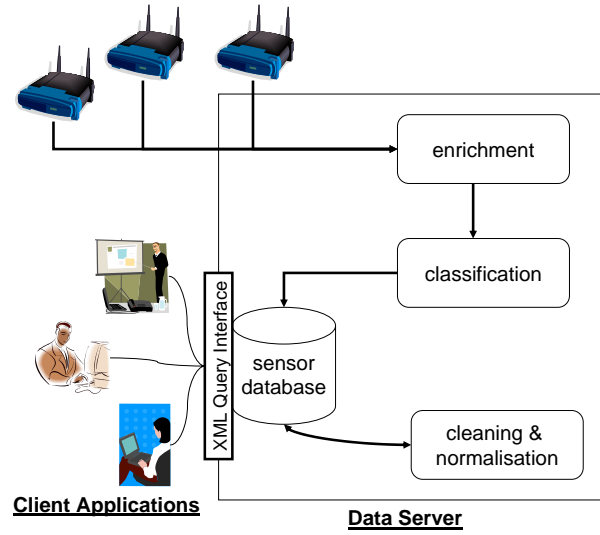


Fig. 3. Data Capture and Management

4 SportsSense System

In this section we describe the infrastructure that facilitates the provision of a data management and query service for a sensor network using the standard web languages XPath and XQuery. While this paper focuses mainly on the cleaning and normalisation of data, we will provide a brief overview of the entire infrastructure.

4.1 Data Enrichment

The role of the enrichment process is to convert the raw sensor data into XML format, providing both structure and additional semantics. This is motivated by the need to use a high level query language rather than write low level primitives every time the user modifies or has a new query requirement. Each sensor device has associated with it a template file (an XML schema document) that facilitates the transformation of the raw data into XML. In previous work [3], we described a system where all sensor streams use an XML template to make themselves readable and queryable by the system. The benefit of this approach is that the system requires no modification when new sensors are introduced to the wireless network. The enriched XML file contains a header section detailing the user information, session parameters that describe the current experiment or activity, and sensor device ID information. The body of the XML file contains the readings recorded by the sensor device.

In figure 4, we have a small extract from an enriched sensor stream which previously generated some header data, followed by a stream of heart rate values and time stamps. Meaningful queries (see §5) are not possible without the

```

<user>murphy</user>
<session>080503ME_Bangsbo</session>
<sessiontype>Under 14</sessiontype>
<sensorData candidate="candidate">
  <device>HRM</device>
  <startTime>1209826519000</startTime>
  <interval>5000</interval>
  <sections>
    <section name="Params">
      <parameter><key>Version</key>
      <value>106</value>
    </parameter>
    ... (parameter element repeats)
  </section>
  <section name="HRData">
    <measurement offset="0"
state="" stateoffset="" time="1209826519000">
      <reading ordinal="">
        <key>HeartRate</key>
        <raw-value>80</raw-value>
        <outlier-value>80</outlier-value>
        <padded-value>80</padded-value>
        <value>80</value>
        <averages>
          <average>
            <time/>
            <value/>
          </average>
        </averages>
      </reading>
    </measurement>
    ... (measurement element repeats)
  </section>
</sensorData>

```

Fig. 4. An Enriched Sensor Stream

descriptive attributes that are added from the sensor’s template file. Other than structural markup, additional semantics such as outlier information, rolling averages (to that timestamp) and athlete details are included to facilitate more complex queries.

4.2 Classification

The goal of this process is the dynamic classification of sensor streams as different groups or clusters of players (in this sensor network) based on various characteristics, such as age, team, field position and experiment type.

One of the requirements of databases or data warehouses is appropriate classification of all stored objects. This facilitates the user when expressing the query if for example, all related experiments are located in the same section of the database. The benefit of the enrichment process is that sensor data streams now have a number of classifying attributes as shown in figure 4. Currently **user**, **session** and **sessiontype** are used to sort each of the sensor streams as they arrive.

This is an important feature when the system scales to include experiments at a National level. As each Gaelic football club rolls out their own sensor network experiments, it will be necessary to distribute data across multiple sites but

have an integration parameter when queries are expressed across distributed information sources. For example, it would be possible to query the results (across the country) for all Bangsbo experiments, involving players under 14 years of age, between a specific range of dates.

4.3 Cleaning and Normalisation

The goal of this process is the calibration of the sensor data so that domain experts can extract and query only appropriate sensor readings. The difficulty with this process is that the network generates data that will corrupt the results of user queries and analyses. This is caused by heart rate monitors sending heart rate data outside the prescribing activities periods. Secondly, the necessary calibration of results requires a number of updates to the sensor streams that can be time consuming using XML databases. Thus, there are two challenges: identifying only relevant sensor data and performing a fast calibration process.

The calibration process involves the use of *states*. By imposing a state value on every sensor reading, we can use the states to categorise and ultimately synchronise sensor streams. All sporting events conform to rigid structures and the training activities employed by the Sports Scientists also adhere to a set structure [1]. Irish Gaelic football has 15 players with the game played over two halves. Scientists require that only heart rates for the duration of the First and Second halves should be used to generate the results of queries. Our algorithm identifies three principle states: First-Half (FH), Half-Time (HT) and Second-Half (SH). An illustration of a typical player heart rate stream over a full game is presented in figure 5. A heart rate sensor begins to monitor as soon as the electrodes touch the skin, and generates a value every 5 seconds. As this takes place in a random and fairly chaotic manner, it is not easy to synchronise sensors at the start of the match. In some cases the monitor is attached 10 minutes before the start and in others up to 40 minutes before the start. However, figure 5 provides some indication as to how the problem was approached as FH and SH states are clearly visible (although only for this participant). Furthermore, all devices were removed within 5 minutes of the end of each match, effectively closing down the sensor network.

The goal of the cleaning process is to remove heart rate values that are not in the First Half (FH) or Second Half (SH) states. For a 15-a-side game, the sports scientists agreed that obtaining a fixed duration of 30 minutes activity for both halves would provide the necessary data to determine the zonal information described in table 1. Thus, we have two fixed functions $T(\text{FH}) = 1800$ and $T(\text{SH}) = 1800$ ($30 * 60$ seconds). It was also agreed that the interval between halves (half-time HT) would be set at 12 minutes, providing $T(\text{HT}) = 720$. These functions will return different values for the 9-a-side networks and also have no meaning for tests such as Bangsbo which has a large number of states, but the principle remains the same, with only the number of functions and their durations changing. It also allows us to parameterise the process where it is clear that playing times were shorter or longer than expected.

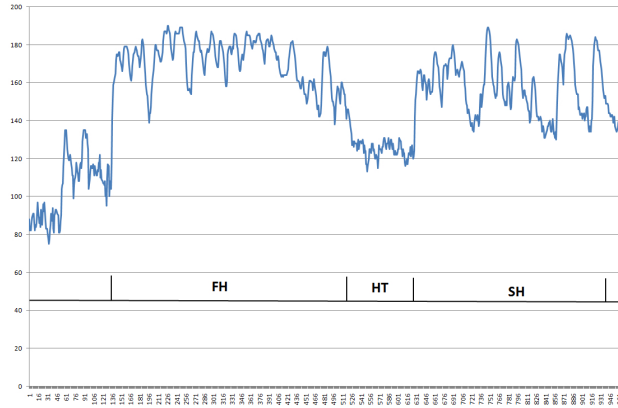


Fig. 5. Player Sensor Data Stream

The process to clean the data has three core algorithms. The first is `detectHT` which returns the start time for Half Time. Once found this is used to segment the sensor stream in order to reduce the search space for the `detectFH` and `detectSH` algorithms. This step is very important as we discovered that these algorithms return many false positives if the entire stream is searched.

The `detectHT` algorithm reads from the end of the file and calculates all candidate Half Times by computing rolling averages for every 5-second interval. This must be computed for at least the length of the half, $T(\text{SH})$ and the length of half time, $T(\text{HT})$, and a surplus, $T(\text{surplus})$ to ensure that we always read enough of the data stream. The lowest average heart rate for each 12 minute duration is regarded as half time: the segment of least activity during the search period.

Definition 1. *detectHT*

```

Start from End(Stream)
Calculate each 12 minute average AVG(HT) for T(SH)+T(HT)+T(surplus)
for all AVG(i)
locate smallest
return begin_time for AVG(smallest)

```

The `detectSH` algorithm reads from the end of half time until the end of the sensor stream, and generates each 30 minute average of heart rate values. Upon finding the largest 30 minute average, this is designated as the SH state. A similar process is used to detect the FH state.

Definition 2. *detectSH*

```

Start from End(HT)
Calculate each 30 minute average AVG(SH) until the end of stream
for all AVG(i)
locate largest
return begin_time for AVG(biggest)

```

As the length for each half and half time will almost never correspond exactly to the predefined durations, there will be gaps between each of the states

which are ignored for analysis purposes. However, we present these intervals to the sports scientists in the event that they wish to recalibrate by changing the state durations. For a subsequent iteration, they are used to modify parameters (lengths of playing halves or half time) and create a new range for the sensor values.

We have deliberately kept the algorithms simple to ensure fast query response times, but we have found accuracy to be above 92% with this purely automated approach. Furthermore, the small percentage of streams where we discovered false positives were due to irregular heart rate values. For example, the goal keeper does not have the high activity exhibited by out field players. Additionally, there were a very small number of players who had relatively low activity during playing times, making the true location of half time difficult. Our current approach will employ a Bayesian model to use data from multiple streams (players) to highlight what are impossible Half Time locations.

Query	Time
1 Return all HR values for player number 1 in game time	63 ms
2 Return the Avg HR value of player number 1 in second half game time	94 ms
3 Return the Max HR value of player number 5 in game time	78 ms
4 Return the Avg HR value of Team Scotstoun in selected game	141 ms
5 Return Avg HR value for each Scotstoun defender in selected game	563 ms
6 Return Max HR value for each Scotstoun defender in selected game	531 ms

Table 2. Query execution times.

5 User Queries and Experiments

After enrichment, data is stored in the MonetDB XQuery server [9] where it is then calibrated until ready for user queries. All experiments ran on an Intel Core 2 Duo processor PC with 4GB of RAM running Windows XP professional. The system was implemented using the Sun Java Virtual Machine version 1.6; the MonetDB XQuery server is version 4.30.0. The current size of the dataset is just under 0.5Gb in its XML format in the MonetDb database. All experiments were executed four times with the average of the last three runs times recorded. The first run was treated as a cold run and thus ignored.

5.1 Sample Queries

For the experiments described in this section, our queries focused on the sensor data generated from a single 15-a-side game. On average, each sensor recorded between 1100 and 1200 readings at 5 second intervals over a period of between 95 and 105 minutes. Table 2 provides a list of queries (in English) with the times

to compute the results. It provides the overall time for delivery of results to end users. These queries represent the base queries upon which more complex queries are then expressed. For example, to determine those players that exhibit heart rate readings that are, on average, higher than the team average, across a number of matches, it is necessary to build upon the set of basic queries presented in Table 2. While these base queries may appear simplistic, it is not possible to execute these user requests on raw data streams. In other words, without the data management layer, it would require identifying a particular stream (from a potentially large number of data streams), applying some context information to determine the phase "in game time", and writing low-level query primitives to detect values or collect sequences to compute min, max, avg etc. In our system, we can apply XQuery to resolve all user requirements. Additionally, we provide the XQuery expression for each query in Table 3 to demonstrate that even the most basic sensor queries require fairly complex XQuery expressions.

Query	Time
1 let \$c := collection('db/GAA_football/Club01') return \$c/healthSense[user[text()='mmccar']/sensorData/sections/section[@name='HRData']/measurement/reading[key[text()='HeartRate']/value/text()]	63 ms
2 let \$c := collection('db/GAA_football/Club01') return fn:avg(\$c/healthSense[user[text()='mmccar']/sensorData/sections/section[@name='HRData']/measurement[@offset >= 3600000]/reading[key[text()='HeartRate']/value/text()])	94 ms
3 let \$c := collection('db/GAA_football/Club01') return fn:max(\$c/healthSense[user[text()='bharra']/sensorData/sections/section[@name='HRData']/measurement/reading[key[text()='HeartRate']/value/text()])	78 ms
4 fn:avg(let \$c := collection('db/GAA_football/Club01') for \$p in \$c//healthSense, \$q in \$c//Players/Player where \$p/user = \$q/Name/Code order by \$q/@Id return if (\$q/TeamId="ST" then \$p//measurement/reading[key[text()='HeartRate']/value/text()] else())	141 ms
5 let \$c := collection('db/GAA_football/Club01') for \$p in \$c//healthSense, \$q in \$c//Players/Player, \$r in \$c//Players where \$p/user = \$q/Name/Code and \$p/session = \$r/Game/Id order by \$q/@Id return if ((\$q/Position/Role="DE" and (\$r/Game/Type="15aside") and (\$q/TeamId="ST")) then fn:avg(\$p//measurement/reading[key[text()='HeartRate']/value/text()] else())	563 ms
6 let \$c := collection('db/GAA_football/Club01') for \$p in \$c//healthSense, \$q in \$c//Players/Player, \$r in \$c//Players where \$p/user = \$q/Name/Code and \$p/session = \$r/Game/Id order by \$q/@Id return if ((\$q/Position/Role="DE" and (\$r/Game/Type="15aside") and (\$q/TeamId="ST")) then fn:max(\$p//measurement/reading[key[text()='HeartRate']/value/text()] else())	531 ms

Table 3. Full Query Expressions.

5.2 Query Evaluation

The first 3 queries are processed in less than 100ms with the reason being that they query a single player data file (or small XML document). In other words, simple queries based on individuals will always execute quickly, even where the database grows quite large. Even in a database of 0.5Gb, the tree pruning capabilities of MonetDB are quite effective.

It takes a little longer to run Query 4, approximately 150ms, because it calculates the team average heart rate in a specific game and requires access to 15 player files. However, the speed is good as the query locates the match data quickly. This is likely to decrease in speed as large numbers of match data is added. There is currently data from almost 200 matches in the database of which 80 are 15-a-side matches. We have determined the correct placement and classification of sensor data to improve query response times.

Queries 5 and 6 both require in excess of 500ms to generate results. This is due to the fact that it must compute averages and maximums for all players in a given team.

Table 3 provides the full XQuery expressions for each query to further emphasise the complexity of query expressions and to help explain the query times. Queries 5 and 6 require iterations not needed in other queries. What this table also demonstrates is the level of detail than can be expressed by building an appropriate data management layer that supports standard query languages such as XPath.

6 Related Research

A recent research project that focused on sensor networks in the sporting domain is presented in [4]. They have a similar contextual platform to their experiment in that they treat a single sporting event as the basis for generating the sensor data. Similar to our work, they provide an experimental prototype with fast response times for built-in queries. However, in our approach we provide a high level and flexible query language to allow domain specialists to probe and refine their requirements.

In [16], the authors present a platform for incorporating non-XML sources into an XML system. As with our approach, they use a Description Language to generate the XML representation of data. On the positive side, no conversion of sensor data is necessary as they create a view definition to interpret the raw data. However, they provide only a template system that has not been applied to any domain (instead they provide some use-case descriptions), and no query response times are possible. Our experience with real world data has shown that unexpected data values such as outliers can require system manipulation that affects response times and different networks will provide different problems when trying to synchronise or integrate the sensor data.

In [17], the authors process and query streams of raw sensor data *without* conversion to XML. Their approach is to enrich raw data into semantic streams

and process those streams as they are generated. Their usage of constraints on the data streams provides a useful query mechanism with possibilities for optimisation. However, this work is still theoretical and has yet to provide experiments or an indication of query performance.

The researchers in [8] also process raw sensor data without conversion to XML. Here they employ the concept of proximity queries where network nodes monitor and record ‘interesting’ events in their locality. While their results are positive in terms of cost, queries are still at a relatively low level (no common format for query expression), and it is difficult to see how this type of proximity network can be applied in general terms due to the complexity of the technologies involved.

In [7], they provide semantic clusters within their sensor network. This is a similar approach to our approach as we classify related groups of sensor outputs. They adopt a semi-automated approach and are capable of generating metadata to describe sensors and thus, support query processing. However, their object-oriented approach is likely to lead to problems with interoperability and this could be exacerbated through the lack of common query language. While this can be addressed with a canonical layer (probably using XML) for interoperability, it is likely to have performance related issues. Furthermore, our approach is fully automated with templates used to provide conversion to XML and processes that employ different rule logic for different sensor configurations and contexts.

7 Conclusions

In this paper, we describe our use of sensor networks to measure the work rate intensity during sports training and matches. As the research focused on school-age players, the networks were configured in terms of size, duration and format to assess the physiological impact of participants. Polar’s team heart rate monitors were used to generate data and this research describes the data management components that were used to deliver data, through a high-level query interface to domain experts. Data and results presented in this paper were gathered from the sensor networks over a 3-month period in the Summer 2008, and research was jointly carried out by both sports scientists and data management specialists. In our experiments, the slowest query executed in a little over half a second, the system is now in daily use. In effect, sports scientists have the full power of the XQuery language to extract results that are not possible with existing vendor software such as Polar or Garmin where fixed, built-in queries generate tables and graphs.

On the sports science side, current research is focused on providing heterogeneity to the sensor network by incorporating different sensor types which must be integrated with the current data streams. This allows for comparisons between sensor outputs, thus enabling an evaluation of new devices before being deployed outside laboratory environments. From an information management perspective, we are developing a framework that provides the same level of query interface, but for streaming data, so that sports sensor networks can be queried in real

time [11]. Furthermore, as all query expressions require somebody with the appropriate IT skills to generate (see Table 3), we are building a parameterised user interface to provide sports scientists with the ability to execute queries without the need for specialist IT involvement.

References

1. Bangsbo J. *Fitness Training in Football: A Scientific Approach*, HO+Storm, ISBN 87-983350-7-3, 1994.
2. Bangsbo J., The Physiology of Soccer - With special reference to Intense intermittent exercise, In *Acta Physiologica Scandinavica*, 151 (supp. 619), 1994.
3. Camous F., McCann D., and Roantree M. Capturing Personal Health Data from Wearable Sensors, *International Symposium on Applications and the Internet (SAINT)*, IEEE Computer Society Press, pp. 153-156, 2008.
4. Devlic A., Koziuk M., and Horsman W. Synthesizing Context for a Sports Domain on a Mobile Device, *3rd European Conference on Smart Sensing and Context (EuroSSC)*, LNCS vol. 5279, pp. 206-219, 2008.
5. Franklin M., Halevy A., and Maier D. From Databases to Dataspaces: A New Abstraction for Information Management, *SIGMOD Record*, 34(4), pp. 27-33, 2005.
6. Grust T. Accelerating XPath Location Steps. *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pp.109-120, ACM Press, 2002.
7. Kawashima H., Hirota Y., Satake S., and Imai M. Met: A Real World Oriented Metadata Management System for Semantic Sensor Networks. *3rd International Workshop on Data Management for Sensor Networks (DMSN)*, pp. 13-18, 2006.
8. Kotidis Y. Processing Proximity Queries in Sensor Networks. *3rd International Workshop on Data Management for Sensor Networks (DMSN)*, pp. 1-6, 2006.
9. MonetDB - open source XML database.
<http://monetdb.cwi.nl/>, 2008.
10. Marks G., and Roantree M. Metamodel-Based Optimisation of XPath Queries. *Proceedings of 26th BNCOD*, LNCS vol. 5588, Springer, 2009.
11. McCann D. and Roantree M. A Query Service for Raw Sensor Data. To appear in: *4th European Conference on Smart Sensing and Context (EuroSSC)*, Lecture Notes in Computer Science, Springer, September 2009.
12. Polar, www.polar.fi, 2008.
13. Reilly T., Energetics of high intensity exercise (soccer) with particular reference to fatigue, In *Journal of Sports Sciences*, vol. 15, pp. 257-263, 1997.
14. Reilly T., and Doran D. Science and Gaelic Football: A Review, *Journal of Sports Sciences*, vol. 19, pp 181-193, 2001.
15. Roantree M., McCann D., and Moyna N. Integrating Sensor streams in pHealth Networks, *14th Intl. Conf. on Parallel and Distributed Systems*, IEEE Computer Society Press, pp. 320-327, 2008.
16. Rose K., Malaika S., and Schloss R. Virtual XML: A toolbox and use cases for the XML World View. *IBM Systems Journal* 45:2, pp. 411-424, 2006.
17. Whitehouse K., Zhao F., and Liu J. Semantic Streams: a Framework for Composable Semantic Interpretation of Sensor Data. *3rd European Workshop on Wireless Sensor Networks (EWSN)*, LNCS 3868, pp. 5-20, 2006.