14th International Conference on Parallel and Distributed Systems (pp. 320-327),
IEEE Computer Society Press, 2008.

Integrating Sensor Streams in pHealth Networks

Mark Roantree and Donall McCann
Interoperable Systems Group
Dublin City University
Glasnevin, Dublin
Email: mark@Computing.DCU.ie

Niall Moyna
School of Health and Human Performance
Dublin City University
Glasnevin, Dublin
Email: niall.moyna@DCU.ie

Abstract

Personal Health (pHealth) sensor networks are generally used to monitor the wellbeing of both athletes and the general public to inform health specialists of future and often serious ailments. The problem facing these domain experts is the scale and quality of data they must search in order to extract meaningful results. By using peer-to-peer sensor architectures and a mechanism for reducing the search space, we can, to some extent, address the scalability issue. However, synchronisation and normalisation of distributed sensor streams remains a problem in many networks. In the case of pHealth sensor networks, it is crucial for experts to align multiple sensor readings before query or data mining activities can take place. This paper presents a system for clustering and synchronising sensor streams in preparation for user queries.

1. Introduction

A sensor is a device for quantifying some physical attribute and transforming the measurement into a signal. Sensor networks often involve large numbers of sensors where the signal data is transferred through wireless or wired networks (or often USB keys) to software managed storage devices. In effect, sensors provide a bridge between the phyiscal and digital worlds. With the continuing growth in ubiquitous sensing, which strives to develop and deploy sensing technology all around us, we can see the emergence of applications such as environmental and personal health monitoring. Most of the developments in sensor networks have focused on developing the sensing technologies, or the infrastructure (middleware) to gather data, with research on power consumption on the devices, security of data transmission, networking challenges in gathering and storing the data and fault tolerance. More recently, the focus has turned to data management, querying and mining of large and often distributed sensor networks.

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 XSense architecture, with processes for harvesting and integration of physical data from wearable sensors.

1.1. Background and Motivation

Until quite recently the evaluation of physiological responses during exercise has been limited to a controlled laboratory environment, which clearly lacks ecological validity. The ability to effortlessly interrogate data relating to the physiological responses during exercise will allow coaches:

1) To view the average heart rate every minute or every 5 min or 10 min etc., during the game. For example, coaches would be interested in comparing the heart rate response during the first 10 minutes compared to the last 10 minutes of a game.
2) To compare heart rate responses between games.
3) To determine the effect of environmental conditions of the heart rate response.
4) To determine how changes in the dimensions on the playing surface and the number of players per team can alter the physiological load (determined by heart rate).

5) Determining the period of time that individual players exercise above or below a predefined heart rate, or percentage of their maximal attainable heart rate (heart rate max).
6) To combine this information with laboratory based data to determine the percentage of maximal aerobic capacity. (VO_2max), and the percent heart rate corresponding to the ventilatory and/or lactate threshold.

Providing real-time automatic *alerting* to the coach/user on when a player reaches a pre-determined excessive level of physiological response (e.g. heart rate), either momentarily or for a pre-defined extended time period, will greatly assist in optimizing performance [5]. The coach can also, if necessary, view the corresponding video and movement (velocity) data to identify the tasks the athlete was undertaking to determine if it was an appropriate physiological response. This information could be used by the coach to design and implement an appropriate intervention. For example, an immediate response may be to remove the player from the activity/game, change the tactic or drill, or substitute the player. A post-event response by the coach may be to implement a (more) appropriate physical training programme to address the poor physiological response. Being able to monitor the information remotely in real-time, will allowing teams/athletes to gain feedback from experts without them having to travel to the site of training. Physiological results in similar experiments can be see in [4].

In this collaboration between data engineers and sports scientists, data was collected from a series of experiments conducted on teams playing Gaelic (Irish) football. We developed on a personal health (pHealth) sensor network where football players have heart rate monitors attached to their chests while they play competitive matches. Separately, we also have a number of different monitors reading separate measurements during personal exercise periods. The main device used to capture heart rate is the Polar Team System, used across a group of participants. Data is collected through USB connections with the entire set of sensors described in earlier work [3].

The motivation for this research project is to provide a data infrastructure and query management system for the listed requirements of sport scientists in scalable pHealth sensor network environment. The challenge lies in the fact that queries across multiple participants require that data must be synchronised across all players to ensure that all players are measured using the same criteria.

1.2. Contribution and Paper Structure

The contribution presented in this paper, is in the specification of an architecture for sensor network management. By clearly defining the boundaries between the different representations and interfaces to data, cross-cutting processors such as data mining and query optimisation can manage and optimise at various levels. Our primary focus will be on distributed management, and in particular, the integration of non-aligned sensor sources. We provide experiments to demonstrate the speed with which we can create dynamic clusters of sensors, and synchronise the streams within the cluster.

The paper is structured as follows: §2 describes the system architecture in order to provide the framework for this research; despite this framework for enriching and integrating data, we find that sensor streams are often unreliable, providing the issues described in §3; while in §4, we present our experiments and report on the output; in §5 we discuss related work; and finally in §6, we provide some conclusions.

2. The Sensor Network Architecture

In this section, we describe the XSense architecture, an infrastructure designed to provide a query interface to sensor data using the standard web query languages, XPath and XQuery. While this paper focuses mainly on Semantic Transformation and Integration (layers 3 and 4), for completeness we will provide a brief overview of the entire architecture.

Raw sensor data must be enriched, normalised and synchronised, and finally integrated into XML documents before queries or data mining activities can take place. In [3], we presented a system for generic enrichment of sensor devices that automatically transformed *any* sensor output to an XML file. This system demonstrated both our ability to handle all sensor streams, providing a short template was provided for the sensor, and the reasonably fast times for generating and integrating XML data. However, this system was largely centralised with little focus on scalability. Furthermore, it focused on individual sensor data and not on teams of participants. The latter requires a crucial synchronisation stage which we will later show to be a difficult problem. The original architecture presented in [3] has been enhanced with these new features.

2.1. Sensor Network

At this level, a potentially large number of sensors are generating raw data at intermittent or continuous

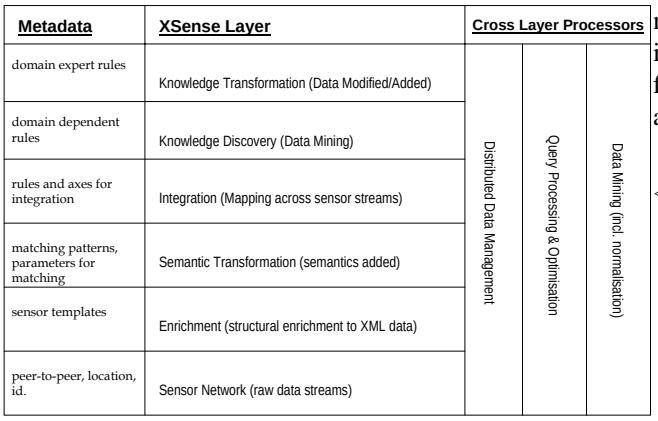


Figure 1. System Architecture

bursts. Querying information at these nodes is time consuming and data mining is very difficult due to the format of data generated. In [2], we described a clustering infrastructure which handles the large scale element of the problem. Peers are clustered based on criteria provided by the knowledge workers and queries will take place against selected clusters of peers, thus, reducing the search space for queries. For example, a cluster may be an entire team; or players from all teams who are over 35 years of age; or all players who played on a specific date.

Metadata. Peers are provided with unique identifiers but are otherwise regarded as autonomous entities which may come or go on a regular basis. Thus, the metadata repository contains no information on actual peers. However, clusters are created to group sensors together for the purpose of query optimisation. This is described in [2] where queries execute on clusters of peers rather than across the entire sensor network.

2.2. Enrichment

At this level, data is converted from its raw format into XML documents or streams. Each sensor device has a small XML template file associated with it. The enrichment processor combines the template and 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 its template. The template describes the sensor's output in terms of its structure and the location of 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 labelled measurement values. A more complete discussion on this process is provided in earlier work [3], while in this paper, small extracts from the enriched BodyMedia and heart rate sensors are provided in examples 1 and 2 respectively.

Example 1: BodyMedia Data

```
<user>
 <id>1</id>
 <session>
  <id>1</id>
  <sensorData deviceID="bsd">
   <startTime>1195226100000</startTime>
   <measurement time = "1195226160000"
         type="BodyMediaSenseWearData">
    <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>
```

Metadata. XML sensor templates are stored in the system repository.

2.3. Semantic Transformation

When integrating data from multiple sensors, one of the primary issues is coordination based on timing axes. It is often crucial when comparing sensor output from participants or events that we can identify and read output for the exact moment across all devices. This is discussed in depth in the section 3 as it provides one of the main contributions of this work.

Metadata. The system repository contains a Ground Truth template (also described in section 3) that is used to identify suitable sensors whose output can be used to calibrate all remaining sensors.

2.4. Integration

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. While the clustering process that employs predicates to group sensor streams, the integration process can combine data into a single stream based on a number of axes. In this paper, we focus on integrating using the temporal axis. In Example 2, a subset of the integrated sensor stream is illustrated.

Example 2: Integrated Data

```
<user>
 <id>1</id>
```

```
<session>
 <id>1</id>
 <sensorData
  <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>
```

Metadata. The system repository contains the attributes used for integration; rules for integrating multiple sensor streams and mappings that are created during the integration process.

2.5. Knowledge Discovery

This layer is domain dependent and employs rules, specified by knowledge workers and domain experts to perform query and data mining operations. XPath 2.0 is used to provide a standard query interface but we are currently extending XPath as not all data mining operations are possible. When changing domains (to environmental or food monitoring sensor networks for example) the knowledge and rules at this layer change completely.

Metadata. The system repository contains rules for navigating and mining data for each domain.

2.6. Knowledge Transformation

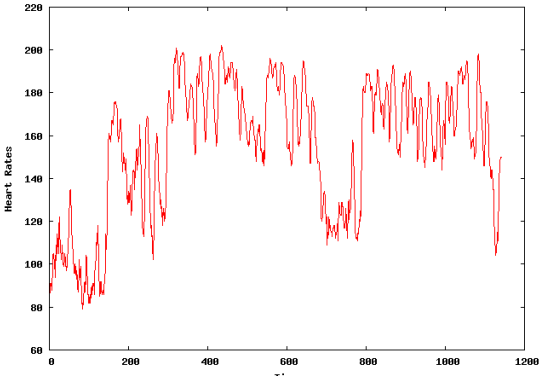
After the Knowledge Discovery process, data is once again transformed.

Metadata. The system repository contains rules for transformation of data, supplied by domain experts.

3. Integration Issues

Often queries involving teams of players will require near to precise times, for example, to compare all participants at the 15-minutes and 30-minute stages of a particular game. Other measures involve comparing their average heart rates across various intervals throughout the game. In order to query across multiple participants, this presents two challenges: dynamic integration of participants as different groups or clusters of players are chosen based on age, field position, weight, etc.; and the synchonisation of sensors to ensure that the starting time of the game is recorded on all sensors.

Figure 2. Midfielder Sensor Stream



From an abstract perspective, sensors can be regarded as generating values that correspond to various *states*. A *profile* is a combination of various states. For Irish field sports, we have identified six states: Pre-Game (Pre), Warm-Up (WU), First-Half (FH), Half-Time (HT), Second-Half (SH), and Warm-Down (WD). Each state occurs once and in the order specified. The goal is to semantically enrich sensor data with an additional field that identifies the state associated with every sensor reading.

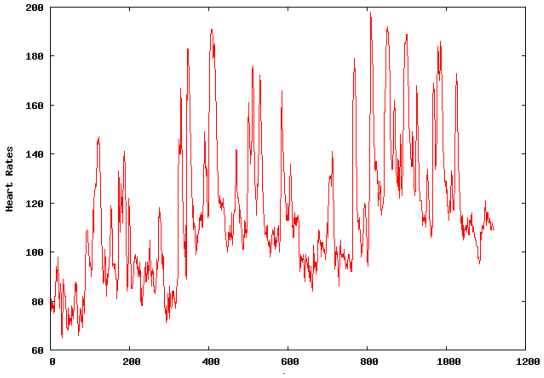
In Example 3, a sample valid profile with 24 values is illustrated (values not shown). Thus, if every sensor generates a set of 24 values and we apply the same profile of states to all players, then (by allocating a state to each sensor value) we can synchronise across players.

Example 3: Game Playing Profile

{Pre,Pre,Pre,WU,WU,WU,FH,FH,FH,FH,FH,FH,HT,HT,HT,
SH,SH,SH,SH,SH,SH,WD,WD,WD}

In figure 2, we see a sensor stream for one player for an entire game. Sensors record every 5 seconds and there are approximately 1,200 values generated while the device is worn. For this particular player, the graph demonstrates all six states albeit with some outlier data. Pre-Game is always the first state; by reading 150, we can see the beginning of Warm-Up; by reading 300, we can detect the beginning of the first half; at roughly reading 650, we see a break of activity for Half-Time; by reading 800, we can detect Second-Half; and finally, by 1100, we are in Warm-Down state. Thus, by applying some metrics of Heart Rate values, we can determine with an acceptable level of accuracy, the beginning of each state. However, this chart is for a midfield player, and one with a profile of gradually building activity through Pre-Game and Warm-Up, and remaining constantly active throughout

Figure 3. Forward Sensor Stream



each half. By analysing the graphs for all 30 players, we discovered that this *model* profile is unusual.

Furthermore, there are a number of issues involved in the enrichment process. Firstly, sensors begin sensing at the moment they are attached the the skin. Thus, each player's sensor will begin to generate values at different times. While there are certain rules (supplied by domain experts) that help to detect the beginning of certain states, they cannot be applied to all players due to their activity during the game. In simple terms, if we could identify (for example) the beginning value for the First-Half state, we could synchronise all players. As can be seen from figure 3, this is not possible. A player that plays up front appears to perform high bursts of activity both during warm up and during the game. At times during the game, this player appears to be at rest for intervals during the game. This provides a significant challenge to creating a generic process for normalising and synchronising sensor streams.

3.1. Matching Candidate Sensor Streams

Our solution to this problem was for the Enrichment Processor to examine all players in order to locate the player that most resembled a model profile. To do this we developed a template class for a player with "ground truth" data stored into our model profile. Each of the six states of the model has an average heart rate associated with it, expressed as a percentage of a maximum heart rate (maxHR).

The raw format of the data is a sequence of heart rate values, taken at an interval of 5 seconds. This results in a graph with a number of peaks and, potentially, a significant difference in standard deviation from one value to the next. In order to *smooth* the graph to ensure a better match against the model profile, we compute 60-second averages. Finally, it is necessary to normalise all heart rates in advance of the matching process as the distribution across players can be significant. For example, an 18 year old will have a different heart rate to a 35 year old player. The highest heart rate for each player is normalised to a value of 100, and all other values are adjusted accordingly.

The matching algorithm is initialised in the first state of the model and sequentially processes average heart rate values. For each value, it computes the difference between the current value and the model profile value of the current and next states (the states correspond to the periods of the model and are, therefore, ordered). If a number of consecutive values are found to be closer to the next state, then the state is changed. This is repeated until there are no more values or states to be processed. Then, the average difference between the heart rate averages and the archetypal average is calculated for each state. The mean of these values is then used to score each candidate. The lower the matching score, the closer the candidate is to the model profile.

In the case of players with profiles such as that in figure 3, the matching algorithm often cannot distinguish between one period and the next as there is no distinct boundary in the data. As a result of this, the program does not move from one state to the next when it should and some states are ignored completely. When this occurs, a high default value difference of 100 is accorded to the ignored states. This has the effect of higher scores for those players whose heart rate data does not fit the required profile, and thus, will never be selected. A value of 100 was chosen as an arbitrarily large value, although there are more elegant solutions, such as using the distance between the ignored state and the previous one, but as we are currently interested only in the lowest matching score, it is not important how the unsuitable candidates are weighted.

3.2. Enriching Sensor Data

Once the best sensor stream has been located, this data is used to discover the precise durations of each of the six states. Once the durations are determined, they are then imposed on all the other sensor streams for this activity (or cluster).

3.2.1. Identifying State Boundaries. The first step is to determine as precisely as possible, the beginning of each state, by searching for certain boundaries in the data. In the context of this experiment, we know that a game has two halves of roughly 30 minutes, with a half-time period of roughly 10 minutes. Based on this, the processor defines a window of 70 minutes of data

which *slides* across the data looking for the best match to the expected shape of the data graph.

The Identification Process comprises three phases.

1) The sliding window is composed of three contiguous sections of 30, 10 and 30 minutes respectively. The program computes the average heart rate values for each section of the window and the window with the maximum difference between the average values of the two 30 minute sections (which should be very high) and the middle 10 minute section (which should be moderate as it corresponds to the half time period) is selected as roughly corresponding to the game itself.
2) Once the correct window has been identified, and since the three states involved in the window (FH,HT,SH) have a fixed interval, these must be adjusted as games will never complete exactly on time.

   To achieve this, the standard deviation of each section is computed and the values on either boundary of the section are examined to see if they are within a standard deviation of value STD_{dur}. During experiments we set value of $STD_{dur}$to be between 1.5 and 3, and currently this is set at 2.8 as it gives the best results with the dataset that we have collected so far. The values that are in the same range as the rest of the section are thus included in it and those that are not, will form part of the *neighbour* state. This process then results in a more accurate picture of the durations of each period.

   As a result, the window states FH, HT and SH have been identified, and this also provides the last state WD, and the first two states, Pre and WU, although they must later be separated. In this experiment, the separation between these two states is not as important as other states.
3) Finally, each boundary value is checked to see whether it is at a peak, a trough or somewhere in between on the graph of heart rate values. Domain knowledge informs us that that each period begins at either a trough (no activity) or a peak (maximum activity). Thus, if we find that a boundary is not at the minimum or maximum in the curve, it is adjusted appropriately. For example, if the beginning of the First-Half state is on a rising curve, this boundary is moved back to the start of the curve.

3.2.2. Enriching All Streams. Once all state durations have been extracted from the data of the best candidate, they are then imposed on the rest of the players. The motivation for this is to have periods of uniform length across all participants, as computing the durations for each player would result in a number of conflicting values. By choosing an ideal candidate we are maximising our chance of getting accurate durations. Also, were we to attempt to extract durations from some players with poorly defined boundaries (e.g. players 9 and 13 from Team 1 in table 2 and the player profile in figure 3), results would be unreliable. Currently, a human inspection of the output graphs finds it difficult to discern the durations of periods in certain cases.

The remaining players' data is processed by a similar program to the one used for extracting durations from the best candidate. In this algorithm, a similar sliding window is used to locate the easiest state to identify (in these experiments, HT). The reason for choosing the beginning of the HT state is that we have observed empirically that the beginning and ending points of the game are often blurred, but the half time (HT) state is generally noticeable as all players rest during this state. Once the precise point has been located, the durations extracted from the best candidate are imposed on the data and it is enriched with state data for all sensor values.

The output of this algorithm is the generation of states with uniform length across all sensor streams. At this point, XQuery can be employed to query the distributed sensor data.

4. Experiments

A total of 60 datasets corresponding to 2 football games with 4 teams of 15 players is used to demonstrate the process of clustering and synchronisation of a relatively small set of data. Table 2 describes the result of the matching process where all 60 player streams were compared with the ground truth model. The experiments were executed 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.

4.1. Matching Results

In Gaelic (Irish) Football, they are 15 players on each side, with one goalkeeper, 6 defenders, 6 forwards and 2 midfielders. In a normal game, both midfielders (positions 7 and 8) are most likely to generate the profile that will help to identify the six states. During a prolonged offence, defensive players may find themselves relatively inactive, while the opposition forwards may also generate low levels of activity. This

Table 1. Candidate Matching Scores

Position	Team 1	Team 2	Team 3	Team 4
1	41.97	11.32	N/A	9.19
2	N/A	12.47	39.57	N/A
3	10.18	8.05	11.35	9.10
4	12.89	11.39	13.03	6.28
5	11.73	37.88	7.93	43.86
6	358.89	N/A	5.36	42.53
7	6.57	5.10	7.03	14.41
8	5.14	5.19	11.34	7.64
9	45.38	7.77	6.10	9.82
10	N/A	9.30	36.95	6.67
11	7.29	5.27	14.46	N/A
12	N/A	9.34	9.28	N/A
13	38.05	8.00	12.94	13.23
14	9.37	11.18	11.09	7.79
15	N/A	13.56	N/A	N/A

will have an effect on the heart rate data and can blur the boundaries between game periods. Despite this information, one cannot assume that specific sensors will always generate reliable data.

Table 2 illustrates the results of the matching process. Players with the lowest matching scores are those whose heartrate data provides the closest match with the model profile. As can be seen from the data, these players are generally numbers 7 and 8. As an example of unsuitable heart rate data, players 9 and 13 from Team 1 show quite high scores. These are the result of their data lacking clear boundaries and the program consequently failing to change state at the right time. They are then assigned a high score by default which makes them unlikely to be selected as the most suitable candidate.

The entries in the table labelled as "N/A" are due to unusable data caused by malfunctioning equipment. This is now less common as sensors are becoming more reliable.

The overall result supports our approach in terms of detecting the right sensor stream, for the subsequent process of enriching data with state information.

4.2. Cluster Timings

For a dynamic creation of a cluster of sensor streams, the matching process cannot be performed in advance. Thus, both matching and enrichment must be completed within a reasonable time frame for user queries. For this experiment with 60 sensors streams each with 1,200 values, it requires 205.2ms to locate the candidate, 198.8ms to extract the durations, and 1136.2ms to apply those durations to all remaining sensor streams.

Table 2. State start and durations

State	System Start	Actual Start	System Duration	Actual Duration	Diff
Pre	0	0	29:10	N/S	0
WU	350	N/S	14:40	N/S	0
FH	526	527	30:40	30:47	5s
HT	894	896	13:00	13:05	10s
SH	1050	1053	31:05	30:58	15s
WD	1423	1424	4:05	N/S	5s

4.3. Matching Precision

In order to evaluate the precision of our enrichment process, in one experiment the health specialists involved in running the trials, recorded the time at which the First-Half started, and the precise durations for the First-Half, Half-Time and Second-Half states. The actual time for the First-Half was then converted by the same person to an actual value. In this case, this value was 527 (43.9 minutes) which was a result of this player (the midfielder) being the first person to wear the sensor device. In general, there can be a long Pre state as it takes time for all players to be equipped with the devices. With this information, we calculate the Actual Start Values (column 3) for 4 states, while the first state will always be zero. The term N/S indicates that this data was not supplied to us.

By examining the System Start and Actual Start values (columns 2 and 3), we can see that our level of precision is very high. The time difference in the final column is based on the difference between columns 2 and 3. In other words, a difference of a single value is 5 seconds, while a difference of 3 sensor values, indicates an offset of 15 seconds. These results were well inside the boundaries for domain user queries.

5. Related Research

In [8], a template for incorporating non-XML sources into an XML environment is presented. Their approach is similar to ours in that they use their Data Format Description Language to generate the XML representation of data. Significantly, 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. In these systems, the query optimiser confronts problems when converting between the view definition and the physical data.

In both [10] and [7], the authors process and query raw streams of sensor data without conversion to XML.

This has its benefits as the construction times for XML repositories (both centralised and distributed) are often reported to be quite large, and we have reported similar issues in [9]. In [10], the approach is to enrich raw data into semantic streams and process these 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 contains no evidence of experiments or query times. In [7], 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 [6], they provide semantic clusters within their sensor network. This is a similar approach to our work, where we cluster related groups of sensors. They also 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.

6. Conclusions

In this paper, we described our architecture used in the management of distributed sensor streams. Two particular features of this architecture form the core of this paper: fast synchronisation of sensor streams, necessary for dynamic cluster creation [2], [9] and semantic enrichment of sensor streams necessary to allow domain expert queries and data mining procedures. We have also described our experiment which uses real-world data recorded over two months, using players from teams playing Gaelic football, the results and analysis of these experiments, and our method for measuring the precision of results.

Current research focus is on building an advanced query interface using a combination of XPath 2.0 and Java functions to facilitate complex query and data mining requirements. As data is distributed across a wide and potentially large sensor network, query primitives must employ distributed (peer-to-peer) metadata (and mappings) in the optimisation of user queries.

Acknowledgment

This research is funded by Irish Research Council for Science Engineering and Technology grant no. CNRS PICS/2006/04

References

[1] K. Aberer, P-Grid: A Self-Organizing Access Structure for P2P Information Systems, in: *Proc. 9th International Conference on Cooperative Information Systems*, LNCS vol. 2172, pp. 179-194, 2001.

[2] Z. Bellahsène and M. Roantree. Querying Distributed Data in a Super-Peer based Architecture, in: Proc. 15th International Conference on Database and Expert System Applications (DEXA), LNCS vol. 3180, pp. 296-305, 2004.

[3] Camous F., McCann D., and Roantree M. Capturing Personal Health Data from Wearable Sensors. *Proceedings of 2nd International Workshop on SensorWebs, Databases and Mining in Networked Sensing Systems*, IEEE Computer Society Press, pp. 153-156, 2008.

[4] Hill-Haas S, Coutts A, Rowsell G, Dawson B. Variability of acute physiological responses and performance profiles of youth soccer players in small-sided games. J Sci Med Sport. Sep 5. [Epub ahead of print], 2007.

[5] Impellizzeri FM, Marcora SM, Castagna C, Reilly T, Sassi A, Iaia FM, Rampinini E. Physiological and performance effects of generic versus specific aerobic training in soccer players. *Int J Sports Medicine*, Jun;27(6):483-92, 2006.

[6] 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.

[7] Kotidis Y. Processing Proximity Queries in Sensor Networks. 3rd International Workshop on Data Management for Sensor Networks (DMSN), pp. 1-6, 2006.

[8] 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.

[9] Roantree M., Smeaton A., Andrieu V., Legeay N., Camous F., and O'Connor N. Optimizing Queries for Mining Sensor Networks. To appear in Intelligent Techniques for Warehousing and Mining Sensor Network Data, Cuzzocrea (editor), IGI Global, 2008.

[10] 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.