Knowledge Acquisition from Sensor Data in an Equine Environment*

Kenneth Conroy[1], Gregory May[1], Mark Roantree[2], Giles Warrington[1], Sarah Jane Cullen[3], and Adrian McGoldrick[4]

[1] CLARITY: Centre for Sensor Web Technologies
[2] Interoperable Systems Group, School of Computing
[3] School of Health & Human Performance, Dublin City University, Dublin 9, Ireland
[4] The Turf Club, The Curragh, Co. Kildare, Ireland

Abstract. Recent advances in sensor technology have led to a rapid growth in the availability of accurate, portable and low-cost sensors. In the Sport and Health Science domains, this has been used to deploy multiple sensors in a variety of situations in order to monitor participant and environmental factors of an activity or sport. As these sensors often output their data in a raw, proprietary or unstructured format, it is difficult to identify periods of interest, such as events or actions of interest to the Sport and Exercise Physiologists. In our research, we deploy multiple sensors on horses and jockeys while they engage in horse-racing training exercises. The Exercise Physiologists aim to identify events which contribute most to energy expenditure, and classify both the horse and jockey movement using basic accelerometer sensors. We propose a metadata driven approach to enriching the raw sensor data using a series of Profiles. This data then forms the basis of user defined algorithms to detect events using an Event-Condition-Action approach. We provide an Event Definition interface which is used to construct algorithms based on sensor measurements both before and after integration. The result enables the end user to express high level queries to meet their information needs.

1 Introduction

Given the widespread nature of sensor networks and sensor technology, and the high volumes of data generated on an ongoing basis, it is inevitable that data warehousing and knowledge discovery will be adopted as key technologies for end users and domain experts to improve their abilities in data analysis, decision support, and the automatic extraction of knowledge from data. In many application areas, the volume of data gathered in a single experiment is too great to extract any meaningful knowledge as end users must manually extract data from spreadsheets for simple queries. Where database type solutions may have been used in the past [2], sensor network data management will demand the type of functionality available in data warehouses [6, 8]. The continuous

* This work is supported by Science Foundation Ireland under grant 07/CE/I1147

refinement of query expressions that is taken for granted in data mining is not possible when using the raw data generated by sensors. However, the wide range of sensor devices together with the low level nature of data offers new challenges to data warehouse researchers as they seek to build *generic* solutions to the issues provided by sensor networks. Different domains will bring different requirements and associated issues. In all cases, one feature is common: the significant gap between the abstract queries of domain specialists and the data which they are processing. This paper presents a system to narrow this gap in order that specialist users can fully exploit the data gathered by their sensor networks.

Paper Structure. The paper is structured as follows: in the remainder of this section, we provide the background which motivates this work and provide a statement of our contribution; in §2, we provide a discussion on state of the art; §3 details the profile-based Semantic Enrichment process; in §4, we present our modular approach to complex event detection; in §5 we present our experiments and an evaluation in terms of high level user queries; and finally in §6, we conclude the paper.

1.1 Background and Motivation

A recent collaboration with the Irish Turf Club [16], the regulatory body for horse-racing in Ireland, provided us with an extensive set of data from sensors deployed on multiple horses and jockeys in training. The sensors include multiple accelerometers (Actilife GT3X, Crossbow) mounted on a horse simulator and jockey. These sensors measure: rates of change in direction in uniaxial and triaxial (x,y,z) planes; a Cosmed K4b2 metabolic system [5] measuring a variety of physiological factors from the jockeys breathing; a SenseWear [12] armband, which estimates energy expenditure; and a Garmin GPS system for outdoor trials. Each of these devices has its own format, ranging from plaintext to XML compliant sensor output.

Depending on the distance that the jockey is racing, there are many different factors that can predict competitive performance with predictors that can be physiological, environmental, or equipment specific. By sensing changes in physiological factors, environmental conditions, and equipment and how they affect each other, it is possible to gain a greater understanding of the demands of both racing and training. This could potentially allow for the development of targeted training sessions to investigate aspects of race performance. By capturing race specific data and comparing it to data generated using the horse simulator, it may be possible to see if training addresses the needs of racing and competing. This is not possible without a warehouse-style system that is capable of combining data from each sensor used to monitor the event as well as utilising user defined information on the event and participant, and measuring over a period of time.

The sports physiologists involved in this research have identified key events that must be determined for knowledge acquisition. These events are then incorporated into high level queries that extract the new knowledge and are now described.

- **SimSpeed.** This event classifies the speed at which the horse-simulator moves. It changes the energy demands of jockeys as they try to maintain a race position under great speeds. It has five movement stages, each representing change in gait of the horse; walking, trotting, cantering, fast-cantering and galloping. A combination of GT3X Accelerometer and Activity Profile are used to identify these stages.
- **Energy Expenditure(EE)-Est.** Estimates the amount of energy expended for the amount of time spent in each of the simulator stages. This is a simplified estimation, used when specialist equipment is not available and is based on participant anthropometrics and accelerometer data.
- **Energy Expenditure(EE)-Calc.** Calculates the amount of energy expended during each of the simulator stages based on physiological data captured at the same time using other sensors. Data can be more accurately calculated based on breath-by-breath data from a portable metabolic gas analysis system (Cosmed K4b2 metabolic system). Data can also be calculated from existing algorithms on portable heart-rate monitoring systems (Garmin, Polar, Cardiosport).
- **Whipping.** During the final stages of a race, jockeys use a hand held whip to drive the horse to greater speeds. Although jockeys are predominately one handed they need to be able to do this with both hands under different racing conditions. Sensing these events is based on three GT3X Accelerometers, one on each of the jockeys wrists, and another located on the saddle.
- **Jockey Pushing Out.** Usually occurring towards the end of a race, the jockey is in dynamic imbalance, positioned in a state of forward propulsion, crouching in order to minimise wind resistance, and encourage the horse to maintain speed. This is discovered using information from GT3X Accelerometers located on the chest, and the sacroiliac joint, and by ensuring the corresponding values are also associated with a level 4 Speed - fast canter.

The advances in sensor technology have resulted in significant changes in the ways in which scientists can gather data. In the horseracing domain, the focus is primarily on the health and condition of the horse. However, decreasing the energy expended by the jockey during the early parts of a race may result in gaining a competitive edge when *pushing out* at the end of a race. No standard way of measuring the energy expenditure of jockeys during horse-racing exists, and thus, no specialised systems to understand this data are in place. As a result, the domain experts seek to calculate energy expended and define other horse-related events from a deployment of multiple sensors. Due to a lack of a common standard amongst the sensors deployed, a data management framework for defining events and acquiring knowledge is required.

Contribution. In this paper, we extend the EventSense framework presented in [3] with a new process to extract knowledge from multiple sensor sources. We begin with a metadata driven approach to structural and semantic enrichment using a series of Profiles. We then expand our event detection mechanism to a 3-Tier format: basic event detection; event detection based on results of other defined events; and events definitions based on data integrated from multiple sources and new event definitions. The modular nature of the event definition allows the end user greater control and flexibility in defining events and thus, acquiring different forms of knowledge.

2 Related Research

In [15], the authors present the Semantic Sensor Web (SSW), an approach to annotating sensor data with semantic metadata to improve interoperability and provide contextual information required for knowledge discovery. They leverage Sensor Web Enablement (SWE) [14] and Semantic web standards to do so. Metadata referring to time, space and theme is included as they extend SWE to have more expressive information based on ontological (OWL)[17] representations. Semantics are defined using RDFa [10] with SWRL [11] based rules defined to deduce new ontological *assertations*. The resulting rule-based assertations allow for extended query and reasoning within the sensor domain. While we also use the SOS[13] and O&M[9] components, our approach is more lightweight, with our event definitions not requiring substantial knowledge of programming or complex specification language. In [7], the SSW approach is extended to illustrate the advantages of semantic annotation of SOS services, focusing on a deep analysis of sensor data to discover important environmental events.

The authors of [20] present a framework for sensor data collection, management and exchange conforming to the SWE standard. They have deployed their system for an environmental monitoring purpose, which involves the integration of multiple sensors. Unlike our approach, context applied to the data is limited to location, with additional context requiring the development of applications that access the data. While they support remote access to multiple sensors, it is not designed to be deployed in an environment with legacy sensors transmitting in various formats, or storing information locally. Their approach contains no facility for defining rules for detecting events, other than cross-correlation of multiple sensors measuring similar properties.

In [1], they present an approach to sensor discovery and fusion by semantically annotating sensor services with terms from a defined ontology representing an environmental monitoring setup. Their main goal is to aid in the detection of natural disasters. The sensors used are static and have relationships defined by the ontology. A Geosensor Discovery Ontology (GDO) is defined, specifying a taxonomy of sensor observations, geographic objects and substances. Like our approach, they use a lightweight method to provide added meaning, keeping complexity low in order to maintain usability by end-users from a non-computing background. Information is discovered based on rules defining semantic requirements, location and timepoints. Usability is provided using a GUI. Sensor fusion is performed by a Joint Server Engine (JSE) which takes user input and translates it into SOS requests. The data is then merged, removing duplicates and normalisation is performed during the process. However, the system structure cannot be altered by the creation of interrelated event definitions to define and detect more interesting events, a necessary requirement in our system.

3 Context and Knowledge Representation

In [3], we introduced the EventSense framework with Profiles used to automate the imposition of structural semantics to raw sensor data. Here, we will show how the same concept can be easily extended to the new domain of horse racing. The goal is to demonstrate how basic knowledge is represented and this is fundamental to our mining activities, presented in the next section.

3.1 Context Data

The task of defining the context is split into two constructs, an Activity Profile which is built for each activity (such as horse-racing) and consists of standard information such as the start and end time and the location, and some non-standard activity specific information. The sports physiologists are interested in activity-based effects on energy expended by the jockey. This requirement can involve complex calculations and algorithms to detect these events, as well as the inclusion of some external contextual information. For instance, knowledge of the weather at the time of deployment and the terrain is useful information to determining why performance was not optimal for a certain deployment. While there exist sensors to identify this information, it is often not feasible to do so. As a result, a broad range of manually recorded information is observed by scientists as the deployment of sensors is ongoing. It is this information which is included in an Activity Profile as optional context. A sample Activity Profile for an indoor deployment of a jockey on a simulator is shown in Example 1.

Example 1. Sample Activity Profile: Horse-Racing (Simulator)

```
<HorseRacing-Sim>
    <aid>1</aid>
    <start_time>12:30:00</start_time>
    <end_time>13:30:00</end_time>
    <date>2010-03-10</date>
    <location>indoor</location>
    <jockey>subject1</jockey>
    ...
</HorseRacing-Sim>
```

Further knowledge is encoded in a Participant Profile. This information is primarily anthropometric data measured infrequently by Sport and Exercise Physiologists as they typically do not alter greatly over time. In addition to these standard values, common across all domains, domain specific information is included where necessary, such as a specific multiplier for some algorithm measuring energy expenditure. In Example 2, we show the anthropometric data for 'Participant ID (pid) 1', the EE-Est multiplier figure, and other domain-based information such as jockey class (trainee). As with the Activity Profile, queries can be made on this information following integration, and they can by used as parameters in the formation of event detection rules.

Example 2. Sample Participant Profile: Jockey

```
<Jockey>
    <pid>1</pid>
    <gender>male</gender>
    <height>170.6</height>
    <weight>68</weight>
    <age>22</age>
    <BMI>23.53</BMI>
    <jockey_type>trainee</jockey_type>
    <horse>Sim3</horse>
    <horse_weight></horse_weight>
    <horse_height>15</horse_height>
    ...
</Jockey>
```

3.2 Sensor Representation

A Sensor Profile must be defined for each sensor type, to model the structure of the sensor data. Each sensor is assigned a Profile detailing the fields corresponding to sensor values and instructions to standardise the data format. This includes information relating to which timing protocol is used, and how this is converted to a system standard. For example, some sensors record their timestamps as a fraction of a minute, others in milliseconds. These must be standardised in order to aid in the process of merging multiple data sources.

Example 3. Sample Sensor Profile: GT3X Accelerometer

```
<GT3XAccelerometer>
    <sid>2</sid>
    <time_format>ms</time_format>
    <sample_rate>30</sample_rate>
    <Granularity_min>0.033333</Granularity_min>
    <Granulatity_max>0.033333</Granularity_max>
    <field_formats>int, int, int</field_formats>
    ...
</GT3XAccelerometer>
```

Example 3 shows a sample Sensor Profile for a GT3X Accelerometer. This shows the fields recorded, assigns them tag names, and details the sample rate and timing format used. It provides the basic structure and layout for a sensors output, but in order to make sense of the information, we must use contextual information. We do this by merging the sensor data with the Activity and Participant Profile information, discussed next.

3.3 Imposing Context on Sensor Data

The process of merging static context with dynamic sensor data uses a combination of Java and the XQuery Update Facility [19]. Currently, we perform integration based on the sensor timestamps and granularity constraints and Contextual Profile information, but this research is ongoing. Due to the different sampling rates of devices, there are often many more records for one device over some interval as for another device. For instance, the GT3X Accelerometer monitors the environment at 30Hz, whereas the Heart Rate monitor samples once per second. We take the approach of averaging values where appropriate and leaving blank spaces where averaged values do not correspond to real world conditions. For instance, averaging the following and preceding values of Heart Rate is appropriate in all experiments. However, averages for accelerometers cannot be used. We identify these constraints with the Sensor Profile, where the granularity `min` and `max` ensure the system does not create data outside acceptable limits. Our motivation for this paper was to determine if key events could be accurately detected, and if these events could be used in query expressions. A more holistic integration process will be presented as part of future work.

4 Knowledge Acquisition

Knowledge acquisition in EventSense is based on events defined by the specialist end user events. EventSense provides the ability to build event detection algorithms using sensor data, context profiles, functions and nested events. Events are modular in nature, and we classify them as Tier 1, 2 or 3 depending on their structure. The Tier classification corresponds to the inclusion of pre-condition requirements for some events prior to definition.

- The most basic events are Tier 3, which consist of a single sensor whose values match a specified condition, and may include Activity and Participant Profile knowledge.
- A Tier 2 event can contain other events (ie. their results) within its condition component and therefore, must also explicitly state the pre-condition required to execute the current event detection. This pre-condition is the event definition for detecting the property involved in the condition.
- A Tier 1 event definition can include both the results of previous events and any number of sensor data (i.e. all sensor data available after integration).

The use of a 3-tier system allows us to define a number of relatively basic events which can be combined to form more complex events. To allow Tier 1 and 2 event definitions, it was necessary to extend our original architecture [3]. It is now possible to specify additional operators, standardise the pre-condition element, perform integration of information sources and model how events relate to each other. The remainder of this section details the structure of the basic event detection module (Tier 3), describes the grammar and operators of the system and details how we use the results obtained from these modules to build up more advanced event detection modules (Tier 1 and 2), thus illustrating the power integrated data can provide the user.

4.1 Pre-Integration Event Detection

Tier 3 events, which are the building blocks for more advanced events, are generally defined to discover a large amount of single-sensor based events on very large sources of information. For instance, the GT3X Accelerometer accessed in Example 4 has 108,000 data values for each hour of deployment. Pre-processing this information to detect some event (a 'fast-cantering' horse in this case) allows subsequent queries for a fast-canter be executed promptly and ensures additional events (Tier 1 and 2) can be defined using these events. In the example shown, the GT3X Accelerometer located on the saddle is accessed for each entry, and the (x,y,z) values are evaluated accordingly. If the condition is satisfied, this new knowledge is added to the data warehouse, by encoded this value with the `fast-canter` tag.

Example 4. Event Definition: Fast-Cantering Horse

```
<event fast-cantering>
    <condition>
        <GT3XAccelerometer location="saddle">
            <entry>
                <x ge 65222>
                <y ge 65222>
                <z ge 65222>
            </entry>
        </GT3XAccelerometer>
    </condition>
    <action>
        UPDATE <GT3XAccelerometer location="saddle"><entry> WITH <fast-canter>
    </action>
<\event>
```

4.2 Post-Integration Event Detection

Example 5. Event Definition: Left-Handed Whip

```
<event Left-Handed-Whip>
    <precondition>
        <event fast-cantering>
    </precondition>
    <condition>
        <GT3XAccelerometer location="LHWrist">
           <entry>
             <x gt 65655>
             <y gt 65655>
             <z gt 65655>
           </entry>
        </GT3XAccelerometer>
        <Logical operator= "AND">
        <GT3XAccelerometer location="saddle">
             <entry><fast-canter></entry>
        </GT3XAccelerometer>
    </condition>
    <action>
        UPDATE <GT3XAccelerometer location="LHWrist"><entry> WITH <LHWhip>
    </action>
</event>
```

To demonstrate Tier 2 events, we introduce the event of *whipping*, as described in the Introduction. This event is defined as: *all three axes of a GT3X Accelerometer located on the left or right wrist reaching their upper threshold at the same time*. Both the left and right wrist values are taken for each jockey as whips are alternated between left and right side towards the end of the race. In addition, whips occur only when the horse is 'fast-cantering', and this constraint is built into the algorithm to improve accuracy.

End users can define this type of knowledge in a step-by-step manner. Firstly, they define the `fast-canter`, as shown in Example 4. Then, they define a whip occurring on either side (a definition for *Left-handed-whip* and *Right-handed-whip*), and finally a generic *whipping* definition which combines the results of left/right whip. Both the left and right whip events involve knowledge previously discovered by prior events. These are the <`fast-canter`> tags included in updates. It is therefore necessary for the left and right whip event definitions to specify the fast-canter event definition as a pre-condition. This means the `fast-cantering` event definition must be defined and executed prior to executing either the left or right whip event detection. The structure of the left handed whip event is shown in Example 5. Similarly, in the generic *whipping* event definition, shown in Example 6, the left and right whip events are pre-conditions.

Example 6. Event Definition: Whipping

```
<event Whipping>
    <precondition>
        <event Left-Handed-Whip>
        <event Right-Handed-Whip>
    </precondition>
    <condition>
        <GT3XAccelerometer location="LHWrist">
            <entry><LHWhip></entry>
        </GT3XAccelerometer>
        <Logical operator = "OR">
        <GT3XAccelerometer location="RHWrist">
            <entry><RHWhip></entry>
        </GT3XAccelerometer>
    </condition>
    <action>
        UPDATE <GT3XAccelerometer location="saddle"><entry> WITH <whip>
    </action>
</event>
```

The Tier 2 definitions illustrate the combination of event results required to evaluate more complex events. We extend the functionality of our previous system to include the NOT and XOR functions, in addition to the AND and OR previously defined.

Tier 1 events can contain multiple sensor output, each of a different type, and algorithms can contain functions combining values from these sensors to extract more complex knowledge. In Example 7, we show a prototype calculation for energy expenditure (EE-Calc). In this example, data from a Cosmed metabolic system and an accelerometer are used to compute a new measure and update an entry with the computed value. This mining process is ongoing: the team of domain experts are now in a position to refine threshold values as their analytical procedures progress.

Example 7. Event Definition: Energy Expenditure Calculation (Prototype)

```
<event EE-Calc>
    <precondtion>
        <event SimSpeed>
        <event fast-cantering>
    </precondition>
    <condition>
         <GT3XAccelerometer location="saddle">
             <entry><SimSpeed eq 3></entry>
         </GT3XAccelerometer>
```

```
        <Logical operator= "AND">
        <Cosmed><entry><EEm gt 0></entry></Cosmed>
        <Logical operator= "AND">
        <FnCalc-EE>
            <&result gt 0>
        </FnCalc-EE>
    </conditon>
    <action>
        UPDATE <Cosmed><entry><EE-Calc-Sum> WITH <value>&result</value>
    </action>
</event>
```

5 Experiments and Evaluation

Experiments were run on a 2.66GHz Intel Core2 Duo CPU server with 4GB of RAM. The sensors were deployed on a jockey while on a Horse Simulator, as shown in Figure 1. As part of our evaluation, we measure query times for identifying the events pre-defined by the sport scientists. We also measure the time taken to enrich the information both structurally and with the event-definition context.

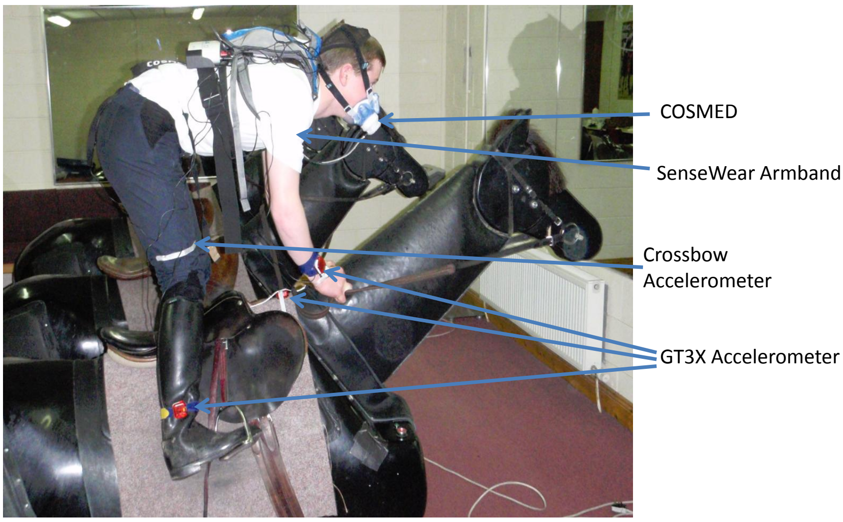

Fig. 1. Indoor Simulator Training

Table 1 shows sample event detection times, run following the execution of the event definitions. The instances of *whipping* represent entries matching the criteria given in the event definition. The numbers correspond to entries, which in the case of the GT3X Accelerometer is 0.03333 of a second. After experiments, it was discovered that the average duration of a `whip` is 0.5 seconds (or 15 instances). Analysing a ground-truth for the data concerned (ground

	Events	Filename	Size	Values	Query Time	Results
1	Fast-Cantering	GT3Xsaddle.xml	8.12MB	108,009	56ms	13,353
2	Left Whip	leftwrist.xml	7.22MB	108,009	51ms	210
3	Right Whip	rightwrist.xml	7.08MB	108,009	53ms	45
4	Whipping	GT3Xsaddle.xml	8.12MB	108,009	55ms	13,353

Table 1. Sample Event Queries and times

	Event	Filename	Enrichment Time
1	Fast-Cantering	GT3Xsaddle.xml	13,450ms
2	Left Whip	leftwrist.xml	12,221ms
3	Right Whip	rightwrist.xml	11,902ms
4	EE-Est > 90%	Cosmed.xml & HeartRate.xml	1,498ms

Table 2. Sample Enrichment Times

truth analysis was performed using video in as many circumstances as possible), confirmed that there were 3 right `whip` events, and 14 left `whip` events, as shown in the query result set.

A prior experimental run resulted in the (incorrect) discovery of 4 right `whip` events and 20 left `whip` events, a total of seven false positives. It was realised that *other* jockey movements were identifed to be `whip` movements. This event detection evolved to include the use a new `fast-cantering` constraint. Including this constraint and the knowledge that a `whip` can only occur during a `fast-canter`, resulted in removing these false positives (one right hand whip, and six left hand whips). This illustrates how the Sports Physiologists are now in a position to alter their needs using event definition modules to improve accuracy. As the tested jockeys were trainees, they are not allowed to gallop. For any other jockey, this event would be replaced with a `Gallop` event.

For completion, we show a sample of contextual enrichment times in Table 2. The time is the total accumulated from converting raw sensor data to low-level structured information and then to high-level event-rich information. As yet, we have not performed any optimisation on the transformation process, as the motivation was to enable users to define and detect complex requirements from semantically poor information. The main evaluation comes from our collaborators who can now define their requirements in the form of events and extract new knowledge.

6 Conclusions

In this paper, our goal was to reduce the gap between the requirements of our collaborators and the sensors recording movement data on Horse Simulators. We extended the EventSense framework with a new process to extract knowledge from multiple sensor sources. We described our metadata driven approach to structural and semantic enrichment using Sensor and Contextual Profiles. We then introduced our new event detection mechanism in its 3-Tier format: basic event detection; event detection based on results of other defined events; and events definitions based on data integrated from multiple sources and new event definitions. The modular nature of the event definition allows the end user greater control and flexibility in defining events and thus, acquiring different forms of knowledge. Our experiments have shown how this approach is evaluated and is providing benefit to the end user. Our current work is based on integration, utilising the timing and granularity constraints along with synchronisation techniques and algorithms, to extend the knowledge acquisition capabilities even further.

References

1. Babitski, G., Bergweiler, S., Hoffmann, J., Schon, D., Stasch, C., Walkowski, A. (2009), Ontology-Based Integration of Sensor Web Services in Disaster Management in *Proc. of the 3rd International Conference on GeoSpatial Semantics (GeoS)*, Springer-Verlag, pp. 103-121.
2. Bonnet, P., Gehrke, J., Seshadri, P. (2001), Towards Sensor Database Systems in *Mobile Data Management (MDM '01), Vol. 1987*, Springer-Verlag, pp. 3-14.
3. Conroy, K., May, G., Roantree, M., Warrington, G. (2011), Expanding Sensor Networks to Automate Knowledge Acquisition. *To Appear in British National Conference on Databases (BNCOD)*, LNCS, Springer-Verlag.
4. Corrales, J. A., Candelas, F. A., Torres, F. (2010) Sensor data integration for indoor human tracking in *Robotics and Autonomous Systems, Vol. 58, Issue 8*, pp. 931-939.
5. Cosmed (2011), http://www.cosmed.it/
6. Da Costa, R. A. G., Cugnasca, A. E. (2010), Use of Data Warehouse to Manage Data from Wireless Sensors Networks That Monitor Pollinators in *11th International Conference on Mobile Data Management (MDM)*, IEEE Computer Society, pp.402-406
7. Henson, C. A., Pschorr, J. K., Sheth, A. P., Thirunarayan, K. (2009), SemSOS: Semantic sensor Observation Service in *International Symposium on Collaborative Technologies and Systems (CTS)*, pp. 44-53
8. Marks, G., Roantree, M., Smyth, D. (2011), Optimizing Queries for Web Generated Sensor Data. in *Australasian Database Conference (ADC)*, Australian Computer Society, Inc., pp. 151-159
9. Observations and Measurements (2011), http://www.opengeospatial.org/standards/om
10. Resource Description Framework in attributes (RDFa) (2011), http://www.w3.org/TR/xhtml-rdfa-primer/
11. Semantic Web Rule Language (2011), http://www.w3.org/Submission/SWRL/
12. SenseWear System (BodyMedia) (2011), http://sensewear.bodymedia.com/
13. Sensor Observation Service (2011), http://www.opengeospatial.org/standards/sos
14. Sensor Web Enablement (2011), http://www.opengeospatial.org/projects/groups/sensorweb
15. Sheth, A. P., Henson, C. A., Sahoo, S. S. (2008), Semantic Sensor Web. in *IEEE Internet Computing Vol. 12*, IEEE Computer Society, pp. 78-83
16. The Irish Turf Club (2011), http://www.turfclub.ie/site/
17. Web Ontology Language (2011), http://www.w3.org/TR/owl-features/
18. XQuery (2011), http://www.w3.org/TR/xquery/
19. XQuery Update Facility (2011), http://www.w3.org/TR/xquery-update-10/
20. Yang, J., Zhang, C., Li, X. Huang, Y., Fu, S. Acevedo, M. F. (2010), Integration of wireless sensor networks in environmental monitoring cyber infrastructure in *Wireless Networks Vol. 16 Issue 4*, Springer Netherlands, pp. 1091-1108