CIMSA 2006 – IEEE International Conference on
Computational Intelligence for Measurement Systems and Applications
La Coruna - Spain, 12-14 July 2006

A Novel Ontology for Sensor Networks Data

Mohamad Eid, Ramiro Liscano, Abdulmotaleb El Saddik
Multimedia Communications Research Laboratory - MCRLab
School of Information Technology and Engineering - University of Ottawa
Ottawa, Ontario, K1N 6N5, Canada
{eid095, rliscano, elsaddik } @ site.uottawa.ca

Abstract *– Sensor networks have seen an exponential growth in the last few years. They involve deploying a large number of small sensing nodes for capturing environmental data. Searching such networks is limited by two major constraints: scalability and precision. We argue that the key to enabling scalable and precise sensor information search is to define an ontology that associates sensor information taxonomy for searching and interpreting raw data streams. We present the motivation and description of the development of the proposed ontology, partial evaluation of the early prototype ontology, a discussion of design and implementation issues, and directions for future research works.*

Keywords *– sensor networks, ontology design, IEEE 1451, semantic representation.*

I. INTRODUCTION

Sensor networks are dense wired or wireless networks for collecting and disseminating environmental data. They consist of a large number of sensor nodes that are connected to central processing nodes called gateways. These networks are characterized by two main features. First, they are highly dense so that hundreds or thousands nodes may be deployed in limited geographical areas. These nodes return huge amount of data that must be efficiently searched to answer user queries. Unfortunately, classical information retrieval techniques showed poor performance in searching sensor networks data as they return many false positives/negatives.

Second, many of the captured data are analogous in nature making the chance of finding a specific term quite good. Most sensors are characterized by similar calibration mechanisms that can be described using different terms. String-matching search techniques may not retrieve all relevant data because different words/terms were used that did not match directly the term. This compromises the performance of the search engine. A big improvement in search engine performance could be achieved if these relationships are captured and utilized, and this is exactly what an ontology can do. This was demonstrated in some recent work on the use of process ontologies [1] that showed an increase in the precision of service discovery queries when semantic representations were used over syntactic representations.

The objective of this paper is to design and implement an initial ontology to retrieve ALL-and-ONLY relevant sensor data. Equipped with term representations and relationships definitions defined by the ontology, the search engine will have information about the meaning of terms. Moreover, these relationships can be used to capture synonyms of a term to retrieve all information available for particular concept. The rest of the paper is organized as follows. Section 2 introduces the basics of ontology design and section 3 highlights related work in semantic sensors data. Section 4 describes the initial taxonomy for sensors data and details the development stages. Section 5 presents the validation and consistency check of the proposed ontology. Finally, we conclude in section 6 by summarizing the preliminary validation of the proposed ontology and recommending directions for future work.

II. BACKGROUND

The term ontology can be defined as “an explicit formal specification of a shared conceptualization” [2]. An ontology comprises three components: (1) classes or concepts that may have subclasses to represent more specific concepts than in super-classes, (2) properties or relationships that describe various features and properties of the concepts, also named slots or roles, and (3) restrictions on slots (facets) that are superimposed on the defined classes and/or properties to define allowed values (domain and range). Individuals can be defined simply as instances of the classes and properties. The ontology together with a set of instances of classes and slots constitute the knowledge base. Reference [3] presents a detailed description of the development stages of ontologies.

Many advantages of ontology design are explained in [3], including: (1) sharing common understanding of the structure of information among people or software agents, (2) enabling reuse of domain knowledge, (3) making domain assumptions explicit, (4) separating the domain knowledge from the operational knowledge, and (5) analyzing domain knowledge. On the other hand, there exist several arguments and challenges, among which are the lack of an agreed-upon taxonomy and quantitative evaluation procedures.

III. RELATED WORK

Despite the amount of research devoted to ontology design and development, very little attention has been paid to semantic representation of sensor networks data. The idea of using ontology-driven information system for sensor networks is not entirely new. The work in [4] presents an attempt to capture the most important features of a sensor node that describes its functionality and its current state. The ontology describes the main components of a sensor node such as processor CPU and memory, power supply, and radio and sensor modules.

A step further in ontology-based sensor nodes is presented in [5] and [6]. The researchers in [5] define an ontology that integrates high level features that characterizes sensor networks for customizing routing behavior. The proposed ontology describes the network topology and settings, sensor description, and data flow. Again, there is no mention of sensor data. Subsequent work like [7] is an effort in the direction of facilitating semantic-service oriented sensor information systems. The notion of ontology used in this research is to capture the information about physical entities that sensors sense and their relationships.

The IEEE 1451 is a family of proposed standards that provide a single generic interface between a transducer and external network protocol in use [8]. The IEEE 1451 standard family uses Transducer Electronic Data Sheet (TEDS) to capture sensor characteristics, such as transducer identification, calibration, correction data, and manufacturer-related information. Consequently, much of the knowledge captured by the ontology describes the widely accepted IEEE 1451 TEDS templates.

IV. THE PROPOSED ONTOLOGY

In this section we overview the way the sensor-data ontology is built as a means to better organize sensor information and assist users and/or search engines in retrieving relevant information. We describe the development stages followed to build the sensor data ontology.

A. Ontology development life cycle

The ontology development follows an evolving prototype life cycle rather than a waterfall or an iterative one. This implies that one can go back from one stage to another stage in the development process as long as the ontology does not satisfy or meet all the desired requirements. The usually accepted stages through which ontology is built are: collecting vocabulary commonly used, identifying an initial taxonomy, adding restrictions and axioms, consistency checking, incremental modifications, and evaluation [10].

B. Obtaining an initial vocabulary list

Our main source for collecting commonly used terms in sensors domain was the IEEE 1451.4 smart transducers template description language [9]. Moreover, the 1451.4 TEDS templates provide the raw data for defining the taxonomic class diagram which in turn forms the foundation of the ontology. Some of the properties are electrical while others are physical. This implies a sort of classification of the taxonomy tree (electrical versus physical data). Moreover, any sensor must have a unique manufacturer ID, thus yielding a functional one to-one relationship between the sensor and the manufacturer ID subclass.

C. Identifying an initial taxonomy

The next step is to take the list of concepts as described by the identified terms and form the initial class taxonomy. This implies looking at whether a concept is a sub-concept of another one or not. Figure 1 shows our initial taxonomy after adding a few dozen concepts. Concepts were added one at a time, structuring the taxonomy as needed to accommodate each concept. Notice that the links from classes to their sub-classes represent properties that are listed in Table 1. For instance, the link from `Data` class to `Calibration` class represents the property "`Data_Can_Be`".

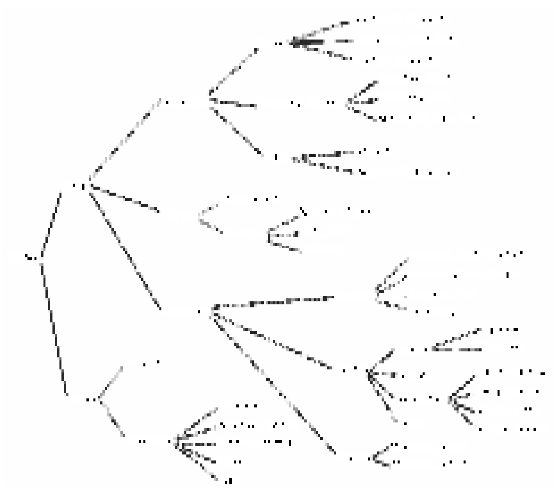

Fig. 1. Initial taxonomy for sensor data ontology.

D. Properties and Restriction

Relationships among classes are usually referred to as properties (for further description of properties classifications, refer to [10]). A property links an individual from its domain to an individual of its range. For example, the `Calibration_Of_Type` property links the `Calibration` class to either `Curve` or `Frequency_Response` or `Table` classes.

Concepts in the taxonomy can be further refined by superimposing constraints and axioms expression relationships. For example, every individual of the "`Accuracy`" class must have a relationship with individual in the "`Format`" class, this is called universal restriction.

Another restriction is called the existential restriction. For instance, there exists at least one electrical parameter, thus forming an existential relationship from the "`Parameter`" to the "`Electrical`" classes. Axioms can also be defined to restrict individual behavior. For example, the disjoint axiom can explicitly imply that a subclass can not inherit from distinct super classes ("`Electrical`" and "`Physical`" classes must be disjointed).

E. Consistency Checking

After building the ontology, the next step is to check whether the ontology is error free or not. Basically, two major ontology tests are conducted: the subsumption test and the consistency check. The first involves testing class hierarchy and whether a class is a subclass of another class or not. The second test is the logical consistency check test. Based on the description (conditions) of a class the reasoner can check whether or not it is possible for a class to have any instances. A class is deemed inconsistent if it can not have any instance. For example if two classes are defined to be disjointed and a third class is supposed to be subclass of both, then it is impossible to have an instance of this subclass thus resulting in logical consistency error.

F. Illustrating Example

Figure 2 shows three individuals of the `Thermocouple` class - namely X, Y, and Z – that are linked to instances of classes `Accuracy`, `Range`, and `Type`. Then the name of only one individual (for instance X) will suffice for retrieving ALL and ONLY thermocouple sensors in the network. The following reasoning takes place:

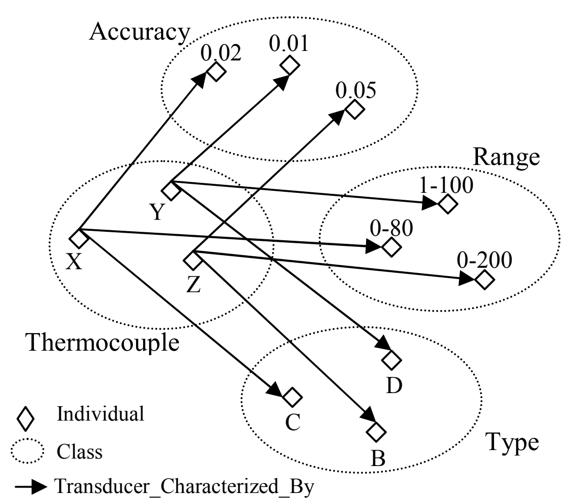

Fig. 2. Three Thermocouple individuals with their individual properties

1. X is an individual of Thermocouple sensor class then Y and Z are also thermocouple sensors and thus should be retrieved.

Table1: Properties list along with their domains and ranges.

Property Name	Domain	Range
Calibration_Of_Type	Calibration	Curve, Frequency_Response, Table
Data_Can_Be	Data	Calibration, Format, Parameter
electric_Has	Electrical	Maximum_Electric_Output, Minimum_Electric_Output, Table, Sensitivity
Format_Has	Format	Physical_Unit, Prototype
has	Prototype	Number_Of_Bits, start_Value, Tolerance
hasdataType	Format	Prototype
HasUnit	Format	Physical_Unit
identity_Has	Identity	Location, Manufacturer, Operator, Owner
location_Has	Location	Latitude and longitude
manufacturer_Has	Manufacturer	Manufacturer_ID, Serial_Number, Version_Letter, Version_Number
parameter_CanBe	Parameters	Electrical, Identity, and Physical
physical_Has	Physical	Maximum_Output and Minimum_Output
Prototype_Characterized_By	Prototype	Maximum_Output, start_Value, Tolerance
Sensor_Can_Be	Sensor	Transducer and Actuator
transducer_Characterized_By	Transducer	Accuracy, Negative_Material, Positive_Material, Range, Type

2. Sensors whose names are X or Y or Z and have the three properties (Accuracy, Range, and Type) are retrieved (only red sensors are retrieved as shown in Figure 3 assuming they have such properties)
3. Other sensors (such as the ones shown in green in Figure 3) which may have the same names as X or Y or Z but don't have such properties are not retrieved. Therefore, only thermocouple sensors are retrieved. Notice that such data will be retrieved if conventional search has been used.
4. Therefore, ALL and ONLY thermocouple sensors are retrieved (red ones in Figure 3)

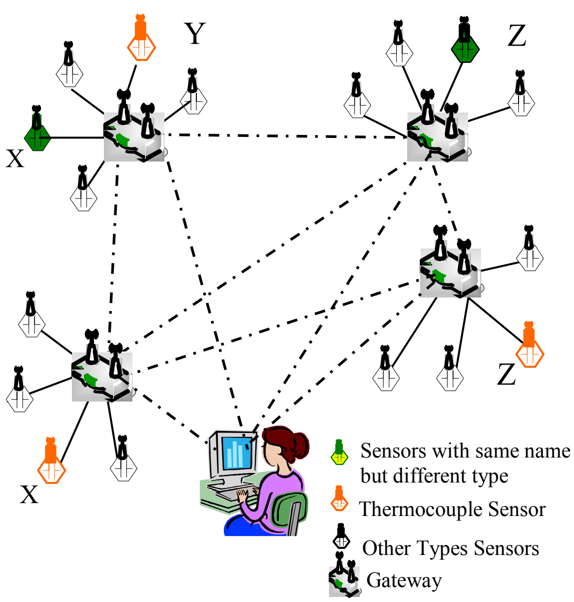

Fig. 3. Searching for ALL and ONLY thermocouple instances

V. IMPLEMENTATION AND VALIDATION

In this section, we present our technical judge of the designed ontology by performing the tests mentioned in section VI: E. The experimental evaluation is limited to validating the ontology (checking for logical inconsistencies). Eventually, comparing the performance parameters of a search engine (such as precision, recall, and response time) when utilizing the ontology versus traditional searching (such as databases) is a vital part of the performance analysis. Therefore this performance testing is our immediate future work.

A. Protégé and RACER

To implement the constructed taxonomy an ontology development tool, called protégé [11] is used to build and edit the ontology. The knowledge representation language for modeling the various data types of sensor data is OWL-DL. We manually add classes to the ontology by creating `Data` and `Sensor` classes and all their sub-classes. The constructed class hierarchy is called the manually created classification hierarchy and is shown in Figure 4.

As a validation tool, we used RacerPro because of its strong reasoning capabilities and interoperability with protégé. The manually created class hierarchy is fed to RacerPro whose main responsibility is to automatically compute the inferred class hierarchy (called asserted ontology) based on the description of classes and relationships. To perform the subsumption test, both Protégé and RacerPro should be up and running.

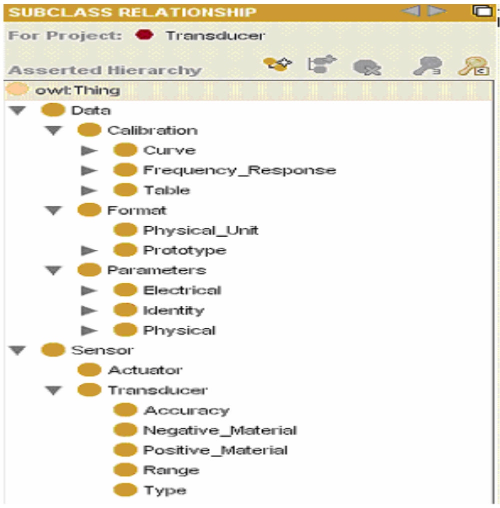

Fig. 4. Asserted hierarchy.

B. Check Consistency

Having started RACER, the ontology now can be sent to the reasoner to automatically compute the classification hierarchy (called taxonomy classification), and also to check the logical consistency of the ontology. We should distinguish two ontologies: the manually constructed class hierarchy (developed according to previous section) and the automatically computed one, both must be identical if the subsumption classification is error free. On the other hand, if the ontology has inconsistencies, the logical consistency check test must be able to detect them. A snapshot of Protégé is shown in Figure 5 where the inconsistent class is marked red.

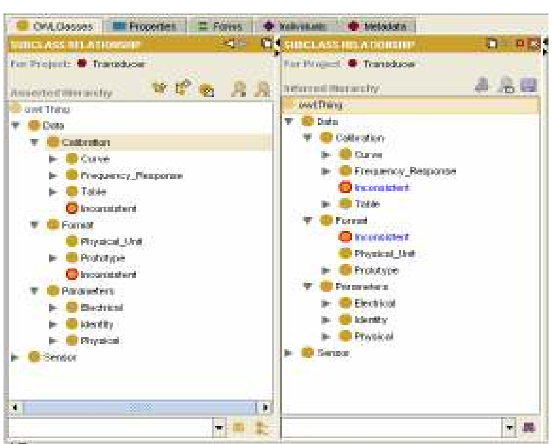

Fig. 5. Inconsistent class is marked red and shown as subclass of both disjoint classes.

VI. CONCLUSION AND FUTURE WORK

The semantic representation of sensor networks data is an exciting vision that enables structured information to be interpreted unambiguously. Precise interpretation is a necessary prerequisite for automatic search, retrieval, and processing of sensor data. This paper is the first attempt to define an ontology for describing concepts and relationships of the sensor network data. The benefits of our work are to improve the precision of searching sensor data by utilizing the ontology.

As for future work, we are considering extending the ontology so that it describes all the TEDS templates defined by the IEEE 1451 standard [8]; including calibration templates. Moreover, we plan to test the effectiveness of the ontology approach by quantitatively measuring the improvements in the precision and recall rates of a search engine when utilizing the ontology against traditional string-based searching approaches. Finally, in order to support semantic web services, we plan to investigate building a functional ontology that describes operations on sensor data. This effort will be a further step in the direction towards enabling semantic web services to access and process sensors data.

ACKNOWLEDGMENT

This research has been supported by an Ontario Centres of Excellence Grant (OT-ST-140282-01) and the industrial partner Instantel Corporation.

REFERENCES

[1] M. Klien and A. Bernstein, “Searching for services on the Semantic Web using process ontologies,” 1st Semantic Web Working Symposium (SWWS-1), Stanford, CA, 2001.

[2] N. Guarino, M. Carrara, and P. Giaretta, “An Ontology of Meta-Level Categories”, Proceedings of the Fourth International Conference (KR94), Morgan Kaufmann, San Mateo, CA, 1994.

[3] N. F. Noy and D. L. McGuinness. “Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[4] S. Avancha, C. Patel, and A. Joshi, “Ontology-Driven Adaptive Sensor Networks”, Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04).

[5] R. Jurdak, C. V. Lopes, and P. Baldi. “A Framework for Modeling Sensor Networks”, Proceedings of the Building

Software for Pervasive Computing Workshop at OOPSLA'04, Vancouver, Canada, 2004.
[6] G. V. Cybenko, G. Jiang, and W. Chung, "Semantic Agent Technologies for Tactical Sensor Networks", Proceedings of the SPIE Conference on Unattended Ground Sensor Technologies and Applications V, pages 311-320, Orlando, FL, September, 2003.
[7] J. Liu and F. Zhao, "Towards Semantic Services for Sensor-Rich Information Systems", proceedings of the 2nd IEEE/CreateNet International Workshop on Broadband Advanced Sensor Networks (Basenets 2005), Boston, MA, Oct. 3, 2005.
[8] IEEE P1451 – Project official web site - http://ieee1451.nist.gov, accessed May 23, 2006.
[9] C. H. Jones. "IEEE 1451.4 smart transducers template description language", http://standards.ieee.org/regauth/1451/Tutorials.html, accessed April 22, 2006.
[10] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe, "A Practical Guide to Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools, Edition 1.0", Tutorial, http://www.co-ode.org, accessed May 11, 2006.
[11] Protégé – The official web site – sitehttp://protege.stanford.edu/, accessed April 28, 2006.