Performance Analysis of two sensor data storages

Payam Porkar Rezaeiye, Mehdi Gheisari,
Department of computer, Damavand Branch, Islamic Azad University, Iran, Tehran

Abstract: Smart environments represent the next evolutionary development step in building, utilities, industrial, home, shipboard, and transportation systems automation. Like any sentient organism, the smart environment relies first and foremost on sensory data from the real world. Sensory data comes from multiple sensors of different modalities in distributed locations. The smart environment needs information about its surroundings as well as about its internal workings. Sensors generate data. Data should send to sink node for further information retrieval. In this paper we evaluate two of the proposed systems for sensor data storage in total data parameter received by sink node names SSW (Semantic Sensor Web) and SemSOS (Semantic Service Observation). They represent and store their data in semantically form. We show that SemSOS transmit more data through network so life time of network decreases rather than we use SSW framework.

Key-Words: Semantic Sensor Web, SSW, Sensor data storage, SemSOS, sensor services

1 Introduction

Wireless sensors are deployed in a growing number of applications where they perform a wide variety of tasks like pervasive computing, e.g., monitoring learning behavior of the children, senior care system, environment sensing, etc., generate a large amount of data continuously over a long period of time. Often, the large volumes of data have to be stored somewhere for future retrieval and data analysis. A big challenge is how to store data efficiently for future information retrieval.

There are some systems that were proposed for storing data like Minos that define a generic, Java-based tool that allows for collecting and storing data collected in wireless sensor networks [1]. ES3N uses Semantic Web techniques to manage and query data, collected from a minidome Sensor Network [2]. Sense and Sens'ability that describes a sensor data ontology which is created according to the Sensor Web Enablement and SensorML data component models[3].The next section describes background studies and the semantic Web technologies. Section 3 describes our evaluation and comparison with other methods. Section 4 present conclusions of paper and future work.

2 Background

The OGC has recently established the Sensor Web Enablement Group in order to address problem of, Lack of standardization is the primary barrier to the realization of a progressive Sensor Web, by developing a suite of specifications related to sensors, sensor data models, and sensor web services.

2.1 SWE

The main specifications defined by the group are described in the following [4].
- Observations & Measurements (O&M) which define standard models and XML Schema for encoding real-time and archived observations and measurements of sensor data.
- Sensor Model Language (SensorML) is a standard model to describe sensor systems and processes associated with sensor observations in an XML-based structure. The information provided by SensorML can be used for sensor discovery,
describing sensor data, and specifying sensor observations.
- Transducer Model Language (TransducerML or TML) provides a conceptual model to describe transducers and to support real-time data to and from sensor systems, sensors and actuators.
- Sensor Observations Service (SOS) is a standard Web service interface for requesting, filtering, and retrieving observations and sensor system information.

The models provided by SWE define a standard framework to deal with sensor data in heterogeneous sensor network applications. Although XML provides a remarkable solution for heterogeneous data representation, there are significant limitations in semantic interoperability and describing the semantics and relationships between different data element using XML representations [5].

Semantic Web Technologies
The Semantic Web is a mesh of information linked up in such a way as to be easily processable by machines, on a global scale. You can think of it as being an efficient way of representing data on the World Wide Web, or as a globally linked database. The Semantic Web was thought up by Tim Berners-Lee, inventor of the WWW, URIs, HTTP, and HTML. There is a dedicated team of people at the World Wide Web consortium (W3C) working to improve, extend and standardize the system, and many languages, publications, tools and so on have already been developed. However, Semantic Web technologies are still very much in their infancies, and although the future of the project in general appears to be bright, there seems to be little consensus about the likely direction and characteristics of the early Semantic Web in other word, The Semantic Web is envisioned as an extension of the current web where, in addition to being human-readable using WWW browsers, documents are annotated with meta-information. This meta-information defines what the information (documents) is about in a machine processable way. The explicit representation of meta-information, accompanied by domain theories (i.e. ontologies), will enable a web that provides a qualitatively new level of service [4].

An ontology defines a common vocabulary for researchers who need to share information in a domain[5]. It includes machine-interpretable definitions of basic concepts in the domain and relations among them.

Why would someone want to develop an ontology? Some of the reasons are:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies. For example, suppose several different Web sites contain medical information or provide medical e-commerce services. If these Web sites share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sites. The agents can use this aggregated information to answer user queries or as input data to other applications.

Enabling reuse of domain knowledge was one of the driving forces behind recent surge in ontology research. For example, models for many different domains need to represent the notion of time. This representation includes the notions of time intervals, points in time, relative measures of time, and so on. If one group of researchers develops such an ontology in detail, others can simply reuse it for their domains. Additionally, if we need to build a large ontology, we can integrate several existing ontologies describing portions of the large domain. We can also reuse a general ontology, such as the UNSPSC ontology, and extend it to describe our domain of interest.

Making explicit domain assumptions underlying an implementation makes it possible to change these assumptions easily if our knowledge about the domain changes. Hard-coding assumptions about the world in programming-language code makes these assumptions not only hard to find and understand but also hard to change, in particular for someone without programming expertise. In addition, explicit specifications of domain knowledge are useful for new users who must learn what terms in the domain mean.

Separating the domain knowledge from the operational knowledge is another common use of ontologies. We can describe a task of configuring a product from its components according to a required specification and implement a program that does this configuration independent of the products and components themselves. We can then develop an ontology of PC-components and characteristics and apply the algorithm to configure made-to-order PCs.
elevators if we “feed” an elevator component ontology to it. Analyzing domain knowledge is possible once a declarative specification of the terms is available. Formal analysis of terms is extremely valuable when both attempting to reuse existing ontologies and extending them. Often an ontology of the domain is not a goal in itself. Developing an ontology is akin to defining a set of data and their structure for other programs to use. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases built from ontologies as data. For example, in this paper we develop an ontology of wine and food and appropriate combinations of wine with meals. This ontology can then be used as a basis for some applications in a suite of restaurant-managing tools: One application could create wine suggestions for the menu of the day or answer queries of waiters and customers. Another application could analyze an inventory list of a wine cellar and suggest which wine categories to expand and which particular wines to purchase for upcoming in other words, Ontologies are a key enabling technology for the Semantic Web. They interweave human understanding of symbols with their machine process ability.

2.2 SSW
Sheth and Henson [6] Describes a frameworks that named semantic sensor Web (SSW) in which sensor data is annotated with semantic metadata to increase interoperability as well as provide contextual information essential for situational knowledge. In particular, this involves annotating sensor data with Spatial, temporal, and thematic semantic metadata. The spatial meta-data provides sensor location and data information in terms of a geographical reference system, location reference, or named locations. The temporal meta-data refers to the time interval duration whose sensor data has been captured. Thematic meta-data provides descriptive information about the sensor node which can be derived by sensor data analysis, and utilizing tagging and textual descriptions [7].

2.3 SemSOS
Henson[8-10] provide a system that are modeling the domain of sensors and sensor observations in a suite of ontologies, adding semantic annotations to the sensor data, using the ontology models to reason over sensor observations.

3 Evaluation and compare

We have evaluated two proposed framework named SSW and SemSOS in total data transmitted to sink node. For our purpose, we use same data once in Semantic annotation of SWE that proposed by SSW and then represent it in ontology form as proposed by SemSOS to evaluate them. The sample data used is shown below:

```
<swe:component rdfa:about="time_1"
               rdfa:instanceof="time:Instant">
<swe:Time rdfa:property="xs:date-time">
  2008-03-08T05:00:00
</swe:Time>
</swe:component>
<swe:value name="satellite-data"
          rdfa:about="Dayton"
          rdfa:instanceof="geo:City">
  0011000111001111 ... 
</swe:value>
```

This example generates two RDF triples. The first, time_1 rdf:type time:Instant, describes time_1 as an instance of time:Instant (subject is time_1, predicate is rdf:type, object is time:Instant). The second, time_1 xs:date-time “2008-03-08T05:00:00,” describes a data-type property of time_1 specifying the time as a literal value (subject is time_1, predicate is xs:date-time, object is “2008-03-08T05:00:00”).

We evaluate our simulation using j-sim[9] software. Figure 1 shows the amount of data received by sink in the each approach.

![Fig.1 amount of data received by sink](image-url)
Amount of data received by sink when we have 400 sensors, in SemSOS is 3134490 and in SSW framework is 2619135. As we can see increasing number of sensors, increase total data exchange in sensor network and in SemSOS more data exchange and we can response more queries but also we have more consuming of energy so that the life time of sensor network decreases[9,11,12].

The machine interpretable representation needs more amount of data to be transmitted to the sensor network. This would lead to an increase of sensor nodes’ power consumption. Most of the overhead consists of self-explanatory meta-data that helps the receiver of the information to interpret the data[13].

4 Conclusion

This paper evaluates two of known sensor data storage that storing data semantically in amount of data transmitted through sensor network to get sink node.

We have done our simulation when we have more sensors also. And show if data send in ontology form to sink, it should more amount of data to be transmitted to the sensor network. Increasing the power consumption means cutting the lifetime of a battery powered sensor. Such a trade-off between lifetime and machine interpretable data is very critical and needs to be addressed using other components in the sensor network architecture.

The future work will focus on the evaluation of other parameters, and comprise sensor data storage.

References:


