A System for Semantic Data Fusion in Sensor Networks

Alex Wun† wun@eecg.utoronto.ca
Milenko Petrovic† petrovi@eecg.utoronto.ca
Hans-Arno Jacobsen†‡ jacobsen@eecg.utoronto.ca

† Department of Electrical and Computer Engineering
‡ Department of Computer Science
University of Toronto

ABSTRACT

Emerging sensor network technologies are expected to substantially augment applications such as environmental monitoring, health-care, and home/commercial automation. However, much of the existing work focuses mainly on collecting and using sensor level data from isolated sensor networks directly, which still burdens applications with the task of interpreting the context and meaning of sensor data. In order to infer high-level phenomena, sensor data needs to be filtered, aggregated, correlated, and translated from many heterogeneous and dispersed sensor networks. In this paper, we present a novel system for decoupling the process of semantic data fusion from application logic based on semantic Content-based Publish/Subscribe techniques. Our main contribution is an integrated system that allows efficient semantic event detection to occur both within and across sensor networks by translating events using ontologies.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Management

Keywords

Publish/subscribe, Sensor networks, Ontologies

1. INTRODUCTION

Emerging sensor network technologies are expected to substantially augment applications such as environmental monitoring, health-care, and home/commercial automation [2]. By nature, sensor networks will be highly customized for specific applications and use highly customized protocols and data formats. As sensor networks become more pervasive, there will be a need to hide their internals from client applications. This requires more than simply providing a declarative interface to access sensor network data, which would still burden applications with the task of interpreting the context and meaning of sensor data. For example, an application may only be interested in weather condition reports and not the specific audio, temperature, and light sensor readings collected when monitoring the environment. However, much of the work today focuses mainly on posting sensor level data from isolated sensor networks to the Internet [8, 6]. There has been little emphasis on how this data should be presented to applications as *high-level phenomena*, thereby enabling the kind of utility envisioned by grid and pervasive sensor networks [3]. In order to infer these high-level events, sensor data needs to be filtered, aggregated, correlated, and translated from heterogeneous (and possibly geographically dispersed) sensor networks. This is especially true in large-scale collaborative sensor network deployments geared towards observing emergent phenomena. For instance, Neptune is one such joint project between the U.S. and Canada whose goal is to establish a long-term sensor network observatory in the northeast Pacific Ocean [1]. They are interested in identifying large-scale events such as submarine volcanic eruptions by fusing data from a multitude of physical, chemical, and biological sensor readings. In practice, distinguishing such events will require interpreting diverse sensor readings based on the knowledge of domain experts. It must consequently also be easy to define, redefine, and modify the semantics defining any given event in order to adapt to new and improving domain knowledge.

The challenges faced by these emerging sensor network applications are threefold: to aggregate sensor data and detect emergent events, to translate these events under different application domains, and to decouple the semantic data fusion process from application logic. Although some of these challenges have been addressed before with respect to sensor data aggregation [7] or web ontologies [9], we are not aware of any efforts to design an integrated system for semantic data fusion across independent sensor networks. Such an integration involves non-obvious interactions between many different system components. In this demonstration, we present a novel system for addressing these challenges. Our approach uses Content-based Publish/Subscribe (CPS) techniques to support the event-driven nature of these application scenarios and to achieve system flexibility through loose-coupling. Our integrated system builds upon and extends concepts studied in S-ToPSS[1] and Micro-ToPSS[2], two

DEBS '07 , June 20-22, 2007 Toronto, Ontario, Canada

---

[1] Semantic-Toronto Publish/Subscribe System

[2] http://microtopss.msrg.utoronto.ca

systems developed by our research group as part of a larger suite of various CPS middleware systems. The approach we take allows semantic events to be defined independently of application and sensor network interfaces. Our main contribution is an integrated system that allows efficient semantic event detection to occur both within and across sensor networks while supporting and decoupling flexible high-level application interfaces.

2. SYSTEM ARCHITECTURE

The system in this demonstration builds upon concepts developed through two of our prior projects studying semantic CPS systems and CPS middleware for sensor networks. We will first briefly overview the CPS paradigm and these two projects before detailing how they are integrated into the overall architecture of our solution.

2.1 Content-based Publish/Subscribe

Content-based Publish/Subscribe is an event-driven messaging model in which event *notifications* are delivered to clients based on their interests. There are three roles in a CPS system: subscribers, publishers, and brokers. Subscribers express interest by issuing *subscriptions* that describe the content of notifications they wish to receive and the subscriptions are stored by the brokers. Publishers issue *publications* to brokers, who are responsible for matching publications against known subscriptions and subsequently forwarding these publications to the appropriate subscribers as notifications. Publications are sets of attribute-value tuples while subscriptions are expressed as filtering constraints on an unordered set of tuples[3]. However, a limitation of current CPS systems is that the matching process is based on strict syntax. It is therefore difficult to address the challenges discussed in the introduction since they require semantic decoupling across application domains.

2.2 Semantic Content-based Publish/Subscribe

Suppose a subscriber expresses interest in being notified of earthquakes in North America by issuing the subscription: $S = \{\textit{(alert = earthquake)} \wedge \textit{(region = North America)} \wedge \textit{(magnitude} \geq \textit{5)}\}$. A publisher may issue a publication in the form: $P = \{\textit{(alert, earthquake),(country, U.S.),(richter, 6.5)}\}$. In traditional CPS systems, this publication would incorrectly fail to match the subscription filter above since neither the attributes *country* and *richter* nor the value *U.S.* are expressed using the same syntax as the subscription. S-ToPSS is a system that extends CPS matching with semantic awareness using three independent and increasingly powerful methods collectively defined in a domain *ontology*.

Synonym and taxonomy translation are two simple methods for expressing equivalence and hierarchical relationships between terms, respectively. Synonym translation allows the matching engine to map attributes and values into "canonical" terms within an application domain. Taxonomy translation enables classification of publication contents. Publications containing more specialized terms than those used in a subscription (from the same hierarchy) are considered to match. While publications containing more generalized terms than those used in a subscription do not match.

While synonyms and taxonomies capture common data representation techniques, the most powerful method involves arbitrary mapping functions that translate one or more attribute-value tuples to one or more semantically related attribute-value tuples. Mapping functions allow domain experts to express arbitrary relationships that otherwise cannot be accounted for by the previous two methods. For example, a mapping function could be defined to transform the publication $P = \{\textit{(station-A-seismic, 5.5),(station-B-seismic, 2),(station-C-seismic, 4)}\}$ into a semantically equivalent publication $P' = \{\textit{(alert, earthquake),(region, North America),(magnitude, 6.5)}\}$ that would match the high-level subscription issued above. In the context of sensor networks, mapping functions can also be used to perform calculations on tuple values for capturing concerns such as uncertainty or sensor noise. For instance, a publication could contain data readings obtained from multiple sensors around a phenomenon (such as a chemical hazard). A mapping function could be written by a domain expert to aggregate or transform these readings into a semantically equivalent publication.

2.3 CPS Middleware for Sensor Networks

Sensor networks are primarily event-based, either generating data at predefined intervals or upon detecting specific events. To our knowledge, current sensor network projects tend to "hard-code" sensor behaviour into their networks. However, as [11] observed, cooperating sensor networks will need to run flexible data-centric middleware capable of directing sensor behaviour and achieving higher-level application goals. Micro-ToPSS is sensor network middleware that facilitates sensor level CPS capabilities by extending the TinyScript [5] language and its associated VM.

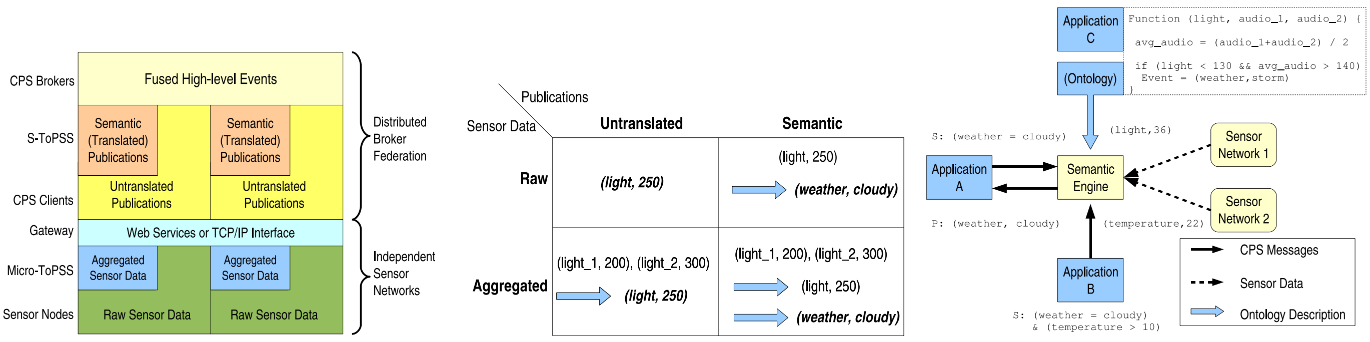
Applications in Micro-ToPSS can use either an unscripted or scripted API. The unscripted API is more coupled with the sensor nodes and provides a way to quickly develop sensor network applications with efficient data-centric routing. The scripted API provides a more flexible and dynamic way to develop CPS applications for sensor networks by decomposing applications into a collection of scripts called *handlers*. Optimization of in-network processing is facilitated by handler mobility and relocation. Both APIs provide CPS primitives for advertising, subscribing, and publishing data gathered on sensor nodes that filter unwanted data according to application interests.

2.4 Semantic Data Fusion System

Figure 1(a) shows the architecture of data processing in our overall system from low level sensor readings to high-level semantic events. In our system, we allow disparate sensor networks to be administrated independently. Each sensor network deploys our Micro-ToPSS middleware platform onto sensor nodes to facilitate application-aware, event-driven data collection within the network. Application-specific data collection requests can either be decomposed into a collection of subscriptions or Micro-ToPSS handlers. Micro-ToPSS performs data aggregation at the sensor level as required by applications and ensures that the sensor network only collects and transmits data that is relevant to applications. Sensor network gateways and CPS clients bridge between the sensor network and application domains, respectively. Sensor data passes from the sensor gateways to CPS clients using either TCP/IP or more flexible Web Ser-

---

[3] Examples of a subscription and matching publication are:
$S = \{\textit{(temperature} > \textit{10)} \wedge \textit{(light} < \textit{150)}\}$
$P = \{\textit{(temperature, 25),(light, 100)}\}$



```
Function (light, audio_1, audio_2) {

 avg_audio = (audio_1+audio_2) / 2

 if (light < 130 && avg_audio > 140)
  Event = (weather,storm)
}
```

(a) System components and data processing architecture.

(b) Both sensor and semantic publications are available.

(c) Decoupled interpretation of sensor network data.

Figure 1: Decoupling application semantics from sensor level data.

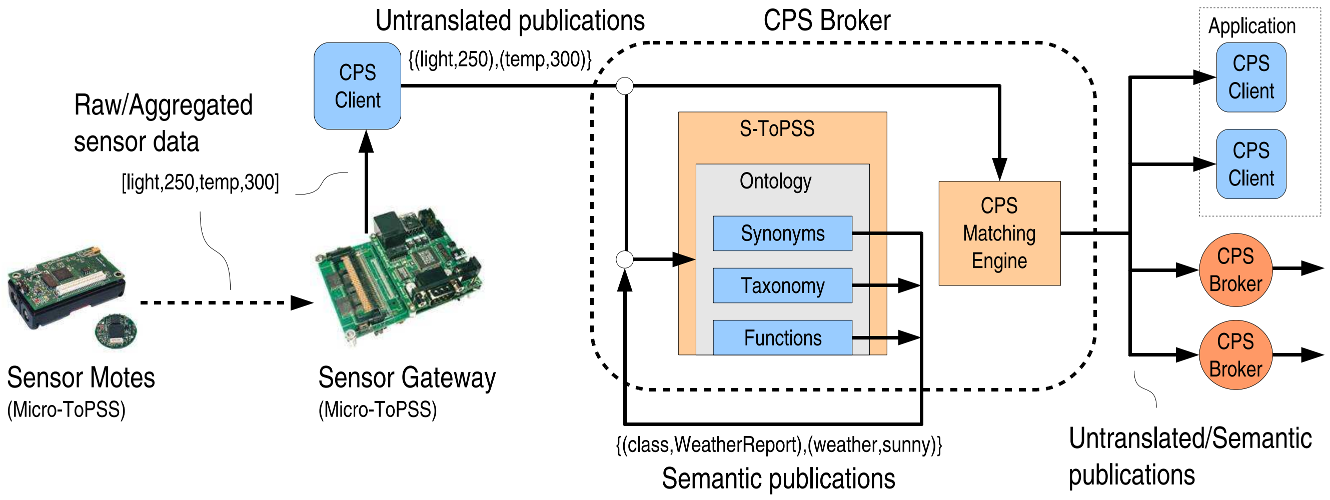


Figure 2: Data flow through our system from sensors to application.

vices interfaces. The CPS clients repackage sensor data into publications and issue the sensor publications to the broker overlay network.

Although the gateway interface abstracts sensor network protocol details, applications still receive sensor level data. This data may be processed (i.e., aggregated, filtered, averaged, etc.) but it is still expressed in the context of the sensor network. In other words, if the sensor network collects light and temperature readings, then the application will receive light and temperature readings in some form. As discussed in our introduction, applications may not be interested in specific sensor level data but rather, the semantic meaning of the data. It is possible for the sensor network to provide high-level interfaces directly, but it is difficult and cumbersome for the sensor network to predict how its data will be interpreted by applications and maintain such a high-level interface. By running a *semantic engine* (S-ToPSS) in our brokers, we give applications the ability to exploit the full utility of sensor level data over a distributed messaging layer. Applications can simply issue high-level subscriptions to the broker overlay, which is responsible for interpreting sensor data and semantically fusing data from disparate sensor networks as necessary. The semantics of the sensor data are defined entirely by ontologies as described earlier and changes to these ontologies are independent of application and sensor network interfaces. The CPS system itself deals with the management of ontologies and translation of semantic events.

In our system, both aggregated and non-aggregated (raw) sensor data is made available for semantic translation. Consequently, applications have the option of subscribing to any of four types of publications as shown in Figure 1(b). The greatest efficiency and semantic decoupling is achieved by subscribing to publications translated from aggregated sensor data. Furthermore, applications can subscribe to correlations between publications originating from independent sensor networks. This approach allows semantic data fusion to occur across sensor networks in addition to aggregation within each sensor network. Figure 1(c) shows how different applications can subscribe to different semantic events based on the same sensor network data through the semantic engine. Application A is issuing a high-level subscription directly specified in the existing ontology. Application B may choose to augment the high-level subscription with a sensor reading filter relevant to its own context. Note that Application B can still access fine-grained sensor data in addition to high-level publications if desired. More importantly, Application C may choose to supply additional domain expert knowledge to the semantic engine by describing new mapping functions, synonyms, or taxonomies. In Figure 1(c), Application C has defined a new mapping function that considers sensor noise by averaging two audio readings before mapping to a new semantic event. Ontologies may exist on a per-application basis or a single ontology may pull together semantic definitions from across application domains. In this approach, new ontologies contributed to the system can be shared by other applications. The new mapping function defined by Application C allows Applications A and B to subscribe to new high-level events without performing any additional sensor data processing themselves.

3. RELATED WORK

Mainwaring *et al.* [8] established a sensor network for monitoring seabird nesting environmental conditions on an island. Their approach makes environmental sensor data readings available through a standard Internet website. Our system is complementary to their project as we investigate allowing applications to semantically leverage sensor networks.

Similar to our project, Gaynor *et al.* [3] propose integrating geographically dispersed sensor networks. They investigate allowing applications to query sensor data through a Data Collection Network consisting of host systems on the Internet. Translation between application queries and sensor queries occurs at Sensor Entry Points, but the DCN itself still deals primarily with sensor data. Our approach complements their system design by providing the messaging layer with application semantic awareness that would greatly improve flexibility and decoupling.

Both Madden *et al.* [7] and Yao and Gehrke [10] allow users to make declarative queries into sensor networks by modelling the networks as databases. As such, they focus on database-style queries such as aggregation and averaging. However, the queries still primarily return sensor network relevant data that will need to be interpreted in an application context. Our integrated system performs this interpretation in a decoupled manner transparent to applications using ontologies. Our system is also based on a publish/subscribe rather than database model.

Intanagonwiwat *et al.* [4] also use publish/subscribe techniques for collecting data in sensor networks. However, their work is restricted to dealing with only sensor level data and does not consider data fusion across sensor networks at the application level, which is the focus of our work.

Ontologies have been studied extensively in contexts such as the semantic web and OWL [9]. However, our research goal is not to design a formalized ontology language or new ontology tools. Rather, we aim to demonstrate that given such ontologies, it is possible to decouple application semantics from sensor network data in a flexible manner. Our approach complements existing work by using general ontology concepts to semantically fuse sensor data across independent sensor networks.

4. DEMONSTRATION DESCRIPTION

Figure 2 illustrates the architecture and data flow of our system. Our demonstration involves a small overlay network of CPS brokers and CPS clients interfaced with the gateways of independent sensor networks. The brokers and clients are part of the PADRES[4] CPS middleware system developed by our research group. The brokers have been extended to run semantic engines and are loaded with different ontologies representing different application domains such as environmental monitoring and factory equipment monitoring. For the sensor networks, we use Mica2 and Mica2dot Berkeley mote hardware equipped with basic sensor boards that provide light, audio, and temperature readings[5]. Each mote in the sensor network runs our Micro-ToPSS middleware. Although we are limited by the environment of the actual demonstration area and sensor motes, this simple sensor network is sufficient since we focus on demonstrating the data fusion system itself rather than the application scenario or the sensor hardware.

Our demonstration uses the predefined ontologies to translate sensor publications into semantic publications. For example, the CPS clients issue publications of the form $P = \{$*(class, SensorData),(light, 250)*$\}$ upon receiving light readings from the sensor network. These publications are translated into semantically equivalent events by the brokers as specified in the loaded ontologies. For instance, the environmental ontology can translate P into $\{$*(class, WeatherReport),(condition, cloudy)*$\}$ while the factory ontology can translate P into $\{$*(class, EquipmentReport),(status, active)*$\}$. These high-level events can then be subscribed to directly or as part of a subscription correlating multiple events. Three key features of our system are highlighted in the demonstration: (1) Describing the same high-level event under different ontologies. For instance, a "lightning strike" may imply high light and temperature readings or high audio readings depending on the ontologies used. (2) Interpreting the same sensor data in different ontologies. For instance, the same light, temperature, and audio readings can be interpreted differently as "lightning strike" or "equipment failure" events under different ontologies. (3) Data fusion of sensor readings. For instance, the mapping function that defines an "equipment failure" event can aggregate audio readings from multiple sensors to achieve more accurate results as part of the event mapping process. These key features are demonstrated through a web interface over our CPS clients.

Acknowledgements

This research was funded in part by CFI, OIT, OCE, NSERC, CA and Sun. We would also like to thank Vinod Muthusamy and the entire Middleware Systems Research Group at the University of Toronto for their valuable feedback regarding this work.

5. REFERENCES

[1] Neptune. http://www.neptune.washington.edu/.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: a Survey. *Computer Networks*, 38(4):393–422, 2002.

[3] M. Gaynor, S. L. Moulton, M. Welsh, E. LaCombe, A. Rowan, and J. Wynne. Integrating Wireless Sensor Networks with the Grid. In *IEEE INTERNET COMPUTING*, 2004.

---

[4] http://padres.msrg.utoronto.ca

[5] http://xbow.com

[4] C. Intanagonwiwat, R. Govindan, and D. Estri. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *MOBICOM*, 2000.

[5] P. Levis, D. Gay, and D. Culler. Active Sensor Networks. In *NSDI*, 2005.

[6] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, and K. Kim. TinyREST: a Protocol for Integrating Sensor Networks into the Internet. In *REALWSN*, 2005.

[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. In *OSDI*, 2002.

[8] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *MOBICOM*, 2002.

[9] OWL. http://www.w3.org/TR/webont-req/, 2004.

[10] Y. Yao and J. Gehrke. The Cougar Approach to In-Network Query Processing in Sensor Networks. In *SIGMOD*, 2002.

[11] Y. Yu, B. Krishnamachari, and V. K. Prasanna. Issues in Designing Middleware for Wireless Sensor Networks. *IEEE Network*, Jan/Feb:15–21, 2004.