Poster Abstract: Active Rules for Wireless Networks of Sensors & Actuators

Michael Zoumboulakis, George Roussos, Alex Poulovassilis
Dept. of Computer Science
Birkbeck College, University of London
Malet Street, London WC1 7HX, UK
{mz, gr, ap}@dcs.bbk.ac.uk

ABSTRACT

The last few years have witnessed a flurry of research in the field of query processing for networks of sensors and actuators. It is widely accepted that query processing is the method of choice for acquiring data from a sensor field. Although query processing offers a very good computational model for a variety of applications such as environmental monitoring, it is a poor match for application scenarios where a timely response to an event is required by the system. With this in mind, we propose a mature database technology, namely active rules, that provides a natural computational paradigm for sensor network applications that require reactive behavior, such as rapid forest fire response and security management.

For the remainder of this paper we will outline the implications of active rules for sensor networks and contrast these against query processing. We will then proceed to discuss work in progress carried out by project *Asene* (Active SEnsor NEtworks) that aims to address these implications. We conclude by introducing our architecture for a decentralised event broker based on the publish/subscribe paradigm and our early design of an Event-Condition-Action (ECA) language for sensor networks.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Distributed databases, Rule-based databases*

General Terms

Management, Performance, Design

Keywords

Sensor Databases, Active Databases, ECA rules, Event Management, Publish/Subscribe.

1. INTRODUCTION

In recent years query processing has attracted considerable interest and is rapidly becoming a popular computational paradigm for a variety of applications. Prototype sensor network query processors have been implemented in the TinyDB [1] and Cougar [2] systems.

SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA.
ACM 1-58113-879-2/04/0011.

In this paper we argue that another database technology may provide an appropriate computational model for a distinct set of sensor and actuator network applications where the system is expected to react rapidly to the occurrence of some event. This technology, namely ECA or active rules [3], offers a reactive computational model that can be applied to a variety of application scenarios that fall under the event-driven category. Although it is possible to apply a query processing system for this class of applications, its deployment would burden the sensor network and unnecessarily consume the limited resources by regularly checking for events that may not have occurred.

The aim of active rules is to program the application so that when specific events are observed and certain conditions are met the network reacts in a predetermined way, for example when the concentration of particular chemical factors are observed and their concentration exceeds a threshold within a small area, then an alarm is activated. Such an active rule may look like this:

```
on UPDATE toxicity
if AVG(toxicity) > threshold WITHIN radius r1
do ACTIVATE alarm WITHIN radius r1, r2
```

Specifying the reactive capacity of a sensor network in rules offers several advantages to the end user of the system. Namely, the reactive functionality can be specified and managed within a rule base rather that being encoded in application code, the rules can be easily analyzed and optimized according to the constraints and requirements of the particular sensor network, and last but not least active rules offer a generic mechanism that can abstract a wide range of reactive behaviors.

2. CHALLENGES

While sensor network query processors (SNQP) have proved very successful in providing appropriate abstractions for user interaction, active rules address the problem of unattended system behavior and can effectively model application logic in autonomous situations [1] In the context of reactive applications, the system is required to provide a timely

[1]The scope of active functionality as described here should not be confused with the so-called event queries supported by TinyDB. Event queries aim to provide user control over data acquisition so that users can register their interest for specific query results returned by the acquisitional query processor. Hence, supporting generic reactive functionality is well beyond the scope of event queries.

response to an event at the lowest communication and computational cost. In contrast with SNQP, active rules aim to support reactive behavior by localizing control and by providing a mechanism to react to events rather than continuously polling the network to test whether a particular event has occurred.

SNQP and active rules have very different execution profiles which also means that they have very different requirements. We will attempt to outline the most critical differences between the two approaches and then discuss our current work in trying to address the novel requirements of ECA execution within project *Asene*.

- **Vantage Points.** SNQPs assume that queries are initiated at a small number of vantage points. In active rules, any sensor node in the network may generate an event that may be used by any actuator node also potentially placed at any network location.
- **Communication Pattern.** SNQPs collect data at regular intervals making it simple to synchronize node wake-up/sleep cycles. Active rules may fire at any time and their timing is largely unpredictable. Hence, we need wake-up/sleep strategies that can support this irregular pattern.
- **Data Model.** SNQPs view the sensor network as a single data space. Active rules require a data model that distinguishes between the different types of objects that are being monitored and can generate events.
- **Aggregation & In-network Storage.** Aggregation in active rules is carried out at the signal layer (e.g. collaborative signal processing) rather than the query layer in SNQPs. In addition, both SNQP and active rules systems can benefit from in-network storage. Whilst storage points are created on top of the tree-based hierarchy in SNQP, in active rule systems they tend to be decentralized-flat and at the event topic channel level.
- **Network Segmentation.** Active rules are more resilient to failure since a rule can execute locally despite isolation from a sink controller. SNQPs require reconstruction of the routing tree in order to accommodate segmentation.

3. SYSTEM DESIGN

Asene is built on top of event channels which are viewed as data object primitives. An event channel has two elements: a collection of nodes that monitor the same attribute and associated algorithmic mechanisms that coordinate node operation. Within an event channel nodes carry out collaborative signal processing, data aggregation and are responsible for in-network storage and event generation.

Event channels are also responsible for a computationally efficient distribution of event following the publish/subscribe (P/S) paradigm. P/S systems are commonly used to bring together data sources and information consumers by transparently delivering events from the first to the second. In *Asene*, event channels are responsible for maintaining a shared list of subscribers and for delivering event notifications. Thus, subscribers may move freely and re-attach to the channel at alternative locations. Effectively, an event channel functions as a decentralised event broker following the P/S jargon.

The particular characteristics of sensor and actuator networks make them especially compatible with the P/S paradigm, in particular with the need for in-network storage and processing. Some of these desirable P/S characteristics are: (i.) anonymity — no need to specify a node ID to subscribe to an event channel, (ii.) decentralized operation of event management and delivery — this makes the system resilient and matches well with the asymmetric computational patterns of sensor networks, (iii.) group communication — P/S systems deliver notifications to multicast groups, a communication mode that fits well with the need for aggregation in order to reduce power consumption.

4. DISCUSSION AND CONCLUSIONS

We have argued that in addition to query processing, active rules is another database technology that may provide an appropriate computational model for a distinct set of sensor and actuator applications. We have presented the challenges in the context of active rules and briefly introduced *Asene*, an ongoing research project that aims to establish ECA rules as the common mechanism for the description of reactive functionality in sensor and actuator networks. The current version of *Asene* supports simple event channels built on top of TinyOS primitives and a simple ECA language. We are currently further developing our algorithms for the efficient construction of event channels in sensor networks. Our focus is on a single-step approach that identifies all members of all registered event channels in a particular network and thus removes the need for duplication of the bootstrap phase. Our work aims to balance the need for low communication between nodes and the asynchronous nature of event generation with regard to the wake-up/sleep node cycles. We intend to conduct extensive experiments with the prototype implementation to better understand the trade-offs involved.

In addition to the development of efficient and effective event channel management techniques, a second major objective of the *Asene* project is the definition of an appropriate lightweight ECA language that satisfies the requirements of the application domain. Finally, the next step for *Asene* is the integration of advanced aggregation algorithms and the study of localized routing algorithms for event dissemination. In doing so we favor a multi-resolution approach similar to the aggregation schemes discussed in [4], but more appropriate for the structure of our event channel construction algorithms. We anticipate — and aim to prove — that our approach will offer significant reduction in resource demands from the network in comparison to query processing.

5. REFERENCES

[1] S. Madden, M. Franklin, J. Hellerstein and W. Hong. 'The Design of an Acquisitional Query Processor for Sensor Networks', *SIGMOD*, San Diego, CA, 2003.

[2] Y. Yao and J. Gehrke, 'The Cougar Approach to In-Network Query Processing in Sensor Networks', *SIGMOD*, Vol. 31, No. 3, 2001.

[3] N. Paton and O. Diaz, 'Active Database Systems', *ACM Comp. Surveys*, Vol. 31, No. 1, pp. 63-103, 1999.

[4] J. M. Hellerstein, W. Hong, S. Madden, K. Stanek, 'Beyond Average: toward sophisticated sensing with queries', *Proc. of IPSN 2003*, Palo Alto, CA, 2003.