PhenomenaBases: Querying and Processing Sensor-network Data at the Phenomenon Level[1]

(Position Paper)

Walid G. Aref
Department of Computer Science
Purdue University
305 N. University Street
West Lafayette, Indiana 47907
Phone: 765 494 1997
E-mail: aref@cs.purdue.edu
Home Page: www.cs.purdue.edu/faculty/aref.html

In this position paper, I take the position that in sensor networks, processing should be at the phenomenon-level and not at the data signal level. Initially, a user would express in extended SQL the definition of the phenomena or phenomena template that the user is interested in. The system would detect and track all the phenomena of interest. User queries would operate on the detected phenomena and not on the raw sensor data signals. Now the user views the sensor network data at a higher level, mainly at the phenomana level. Phenomenon definition is very similar to standard view definitions that operate on sensor network data. The collection of phenomena of interest form a phenomena database (phenomenaBase). A phenomenaBase is a database of registered phenomena that are of interest to some users. Data not contributing to any registered phenomena is not processed. Users may issue queries against specific registered phenomena or against all phenomena.

Motivation for phenomena bases include: high-level description of the sensor field, phenomena-guided data acquisition (e.g., collect data with higher frequency at the boundary of a phenomenon region in contrast to its inner region), sensor data compression (e.g., maintain only sensor readings that contribute to phenomena, maintain phenomena, their boundaries, and their trajectories), prediction (e.g., based on existing phenomena, their boundaries, and their trajectories, predict the next state of the space), phenomena-guided query processing (e.g., query processing is guided to regions with phenomena that satisfy query predicates, and hence reduce the query space based on the knowledge of existing phenomena – similar to the view matching problem).

A phenomenon is characterized by a group of sensors persisting to show “similar” behavior over a period of time. Possible notions of similarity are: sensors with the same value (i.e., discrete phenomena) at the same window of time, sensors with values within ξ of each other at the same window of time, sensors with similar summaries (e.g., count sketches), sensors with similar trends (e.g., correlated increase or decrease in values), etc. Several restrictions can be added to the phenomena definition. For example, we may only want to consider as phenomena ones that show in at least N streams (the “spread” of phenomenon). Other restrictions can be enforced in relation to the number of times a value or a pattern has to appear (persist) before it is considered a phenomenon. Below, I outline a possible interface/language for defining stream bundles on a

[1] Joint work with Ph.D. Candidate: M. H. Ali. This research is partly supported by the NSF under Grant No. 0093116.

sensor field, defining phenomena of interest, and querying the sensor field based on the phenomena definitions.

Defining and querying a stream/sensor:

```
CREATE STREAM EntranceSensor            SELECT EntranceSensor.Temperature
  (int Temperature, int Humidity)       FROM STREAM EntranceSensor
FROM IP:128.10.9.155 PORT 5600          WHERE EntranceSensor.Humidity>30
                                        WINDOW 60
```

Defining and querying a stream bundle (sensor field):

```
CREATE STREAM BUNDLE SB[200]            SELECT SB.id, SB.Temperature
(int Temperature,  int Humidity)        FROM STREAM BUNDLE SB
FROM IP:128.10.9.155 PORT 5600          WHERE SB.Humidity>30
                                        WINDOW 60
```

Defining a phenomenon over a stream bundle:

```
CREATE PHENOMENON Heat_Zones ON           STREAM BUNDLE SB
PATTERN |SB[i].Temperature-SB[j].Temperature| < 3
                PERSISTENCY 5       // No. of times pattern appears within window w
                                    // Persistent vs intermittent phenomena
                SPREAD 4            // No. of sensors pattern appears in
                                    // Stretch vs. confined phenomena
                TIME SPAN 60 ω      // Sliding window w
                                    // Durable vs. impulse phenomena
WHERE SB.Temperature > 90
```

The example above defines phenomena of interest to be ones that have a spread of at least 4 sensors and each sensor has to have at least 5 readings of temperature over 90 degrees in a time span of 60 seconds.

A PhenomenaBase is a database of phenomena that has the following functionality: Track the propagation of phenomena in the space, and based on detected phenomena, control the acquisition of stream data, and optimize subsequent user queries based on the detected phenomena.

Phenomena-based querying in PhenomenaBases:

```
LIST PHENOMENA;            // List the phenomena defined in the PhenomenaBase
SELECT * FROM … WHERE …WITHIN PHENOMENA <list-of-ph-ids>;
SELECT … FROM … WHERE …WITHIN PHENOMENA * ;
```

In contrast to disseminating a query to the entire sensor field, the queries above are directed only to regions in the sensor field that exhibit the phenomena of interest to the query. Challenges in PhenomenaBases include phenomena detection and tracking, scalable query processing, forwarding and routing of queries to where phenomena of interest are, indexing of phenomena, and phenomena containment and hierarchy.