

A Pipelined Framework for Online Cleaning of Sensor Data Streams*

Shawn R. Jeffery^{1†}

¹ UC Berkeley

Gustavo Alonso^{2,1‡}

² ETH Zurich

Michael J. Franklin¹

³ Arched Rock Corporation

Wei Hong^{3†}

⁴ Stanford University

Jennifer Widom⁴

Abstract

Data captured from the physical world through sensor devices tends to be noisy and unreliable. The data cleaning process for such data is not easily handled by standard data warehouse-oriented techniques, which do not take into account the strong temporal and spatial components of receptor data. We present Extensible receptor Stream Processing (ESP), a declarative query-based framework designed to clean the data streams produced by sensor devices.

1 Introduction

Physical receptor devices such as wireless sensor networks and RFID technologies are enabling new classes of applications that use sensor readings to gain insight into the physical world. Examples include real-time supply chain management [3] and environmental monitoring [6]. A limitation of current systems is the unreliability of the data produced by physical devices. This “dirty data” manifests itself in two general forms: missed readings and unreliable readings. Missed readings occur when a sensor fails to report a value, through dropped messages or other errors. Unreliable readings, such as outlier values, are due to inaccurate sensors.

Applications attempting to use raw receptor data directly will have great difficulty functioning correctly without significant preprocessing of the data to account for its unreliability. As this data cleaning will be common across most applications, one possible solution would be to use standard data warehouse-style cleaning. Data cleaning in warehouses is usually an offline, centralized, iterative, and sometimes interactive process that focuses on tasks such as transformations, matching, and duplicate elimination [5]. Such an approach, however, is ill-suited for receptor data.

*This work was funded in part by NSF under ITR grants IIS-0086057 and SI-0122599, by the IBM Faculty Partnership Award program, and by research funds from Intel, Microsoft, and the UC MICRO program.

[†]This work was done while the author was at Intel Research Berkeley.

[‡]This work was done while the author was at UC Berkeley as a Stonebraker Fellow.

Receptor data demands different cleaning techniques that address the type of errors it exhibits (i.e., missed and unreliable readings). Receptor data tends to be strongly correlated in both time and space; the readings observed at one time instant are highly indicative of the readings observed at the next time instant, as are readings at nearby devices. We introduce the concepts of *temporal* and *spatial granule* to capture these correlations and to clean the data streams using temporal and spatial windowing.

We propose an extensible framework for online cleaning of receptor data streams, *Extensible receptor Stream Processing* (or *ESP*). ESP consists of a programmable pipeline of declarative query-based stages designed to operate on-the-fly as the data is streamed through the system.

In this short paper, we present the basic concepts behind ESP processing and show the high-level results of an experiment using ESP to clean RFID data. A more detailed study can be found in our technical report [4].

2 The ESP Framework

In this section, we introduce *Extensible receptor Stream Processing* (ESP), our pipelined data processing framework for online cleaning of receptor data streams. ESP cleans raw physical data by processing multiple receptor streams, exploiting the temporal and spatial aspects of receptor data to produce an improved stream that more accurately reflects the physical world.

Before discussing the ESP processing model, we first define the temporal and spatial abstractions that ESP uses to drive many of its cleaning mechanisms.

2.1 Temporal and Spatial Granules

In general, receptor-based applications are not interested in individual readings in time or individual devices in space, but rather in an application-level concept of *temporal* and *spatial granules*. These granules define the lowest-level, atomic unit of both time and space in which an application is interested.

2.1.1 Temporal Granules

Although many receptor devices are capable of producing data at a very high rate, applications are usually concerned with data from a larger time period, or *temporal granule*. For instance, a sensor network environmental monitoring application that builds models of micro-climates in a red-wood tree needs data at a 5 minute granularity to capture variations in micro-climate [6].

To support this notion of temporal granules, ESP uses time-based sliding windows to group readings within a granule. Within a window, readings can be aggregated or compared with each other to detect obvious outliers.

2.1.2 Spatial Granules

Similarly, receptor-based applications usually are not interested in the data from individual devices, but rather in an application-level notion of a *spatial granule*, such as a shelf in a retail scenario or a room in a digital home application. Spatial granules are the lowest level spatial unit on which an application operates.

To support this application-level view of spatial granules, ESP organizes receptors into *proximity groups*. A proximity group defines a set of receptors of the same type that are monitoring the same spatial granule, such as a set of motes monitoring the temperature in the same room, or two RFID readers monitoring the same warehouse shelf. Just as a window is the unit of processing for a temporal granule, a proximity group is the processing unit for a spatial granule. ESP processes the readings from devices in the same proximity group in a similar manner to readings within a time window.

2.2 ESP Cleaning Stages

Having described the fundamental abstractions underlying ESP, we now outline ESP's processing stages. Through an analysis of typical receptor-based applications, we distilled a set of logically distinct operations that occur in a large class of applications to clean data produced by a wide range of receptor devices. Using our observations, ESP organizes receptor stream cleaning into a cascade of five programmable stages: *Point* - *Smooth* - *Merge* - *Arbitrate* - *Virtualize*.¹ These stages can be programmed using declarative continuous queries.

Stage 1, *Point*: The *Point* stage operates over a single reading in a receptor stream. The purpose of this stage is to filter individual readings (e.g., obvious outliers) or to manipulate fields within a reading.

Stage 2, *Smooth*: In *Smooth*, ESP uses the temporal granule to correct for missed readings and detect outliers in a single receptor stream. It does this by interpolating or aggregating readings in a sliding window corresponding the size of the temporal granule.

¹Not all stages are necessary for a given deployment.

Stage 3, *Merge*: Analogous to the temporal processing in the *Smooth* stage, *Merge* uses the spatial granule to correct for missed readings and remove outliers spatially. *Merge* does this using windowed processing over receptor streams within a proximity group.

Stage 4, *Arbitrate*: Spatial granules may not map directly to receptor detection fields, leading to conflicts among the readings from different proximity groups that are physically close to one another. The *Arbitrate* stage deals with conflicts, such as duplicate readings, between data streams from different spatial granules.

Stage 5, *Virtualize*: Finally, some types of data cleaning utilize readings from different types of receptors or stored data for improved effectiveness. The *Virtualize* stage combines readings from different types of devices and data sources.

These stages are generally applicable across many types of receptors and applications. They are easy to program independently, in many cases through declarative queries.

3 RFID Data Cleaning

Here we present an example of receptor data cleaning based on ESP through a retail scenario using RFID technology.² Such technology is notoriously error-prone: tags that exist are frequently missed while tags that are not in a reader's normal view are sometimes read. In our retail scenario, the application continuously monitors the count of items on each shelf using the query shown below. In this query, the window clause indicates the temporal granule (5 seconds) and the GROUP BY clause denotes the spatial granule (a shelf).

```
SELECT shelf, count(distinct tag_id)
FROM rfid_data [Range '5 sec']
GROUP BY shelf
```

We ran an experiment emulating a retail scenario. Our experimental setup is depicted in Figure 1. We used two 915 MHz RFID readers from Alien Technology [2], each responsible for one shelf and thus each forming a proximity group. Each shelf was stocked with 10 tagged items using Alien "I2" tags [1]. Additionally, to introduce a dynamic component into the experiment, we relocated 5 items between the two shelves every 40 seconds.

The results of this experiment are shown in Figure 2. Figure 2(a) depicts the trace of the actual count of items on each shelf over the course of the experiment. Figure 2(b) shows the results of running the application's query over the raw data. If the application were to use the output of the RFID readers directly, the results would be near-meaningless. For instance, if an application wants to be

²We present more detailed examples in our technical report [4].

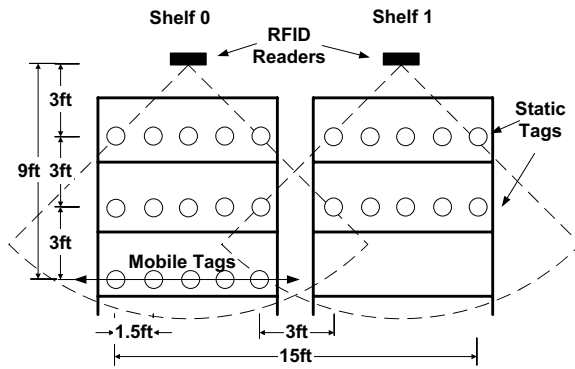


Figure 1. Shelf scenario setup with 2 shelves, each with an RFID reader and 10 tags statically placed within 6 feet of the antenna (5 tags at 3 feet, 5 tags at 6 feet). Additionally, 5 tags were relocated every 40 seconds.

notified when the number of items on a shelf drops below 5, then the query using the raw data would report that a shelf is in need of restocking 2.3 times per second, on average.

3.1 ESP Cleaning

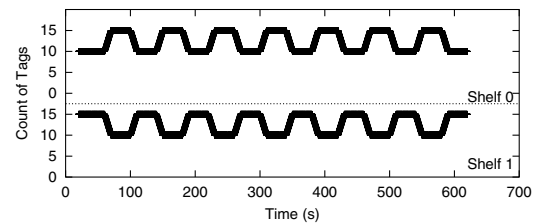
We use an ESP pipeline to clean this data by implementing the *Smooth* and *Arbitrate* stages using declarative continuous queries as discussed above. We show the queries for this scenario and others in our technical report [4].

The results of the application's query over the data produced by each successive stage are shown in Figures 2(c) and 2(d). Observe that in this scenario ESP is able to correct for the unreliabilities of RFID data to provide a substantially more accurate count of the items on each shelf to the application.

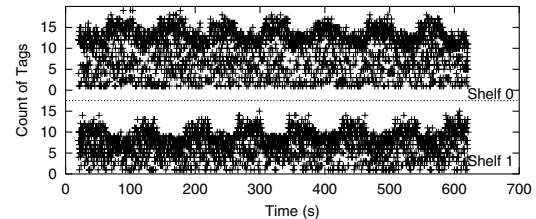
4 Conclusions

Data produced by physical receptor devices is notoriously dirty: readings are frequently either missed or dropped and individual readings are unreliable. This data must be cleaned before being used by any application. Traditional data cleaning techniques, however, cannot easily correct such errors.

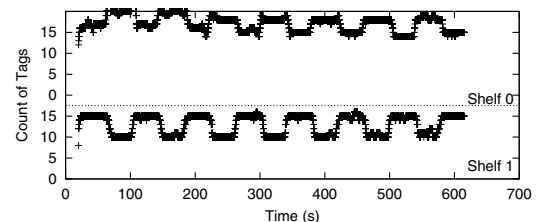
Our approach, ESP, uses declarative continuous query processing in a pipeline of processing stages to clean receptor data as it streams through the system. ESP recognizes the strong temporal and spatial components of receptor data and uses the concepts of temporal and spatial granule to drive many of its cleaning processes. In a real world scenario, we have shown that ESP provides significant improvement over raw sensor data in terms of accurately reflecting the physical world.



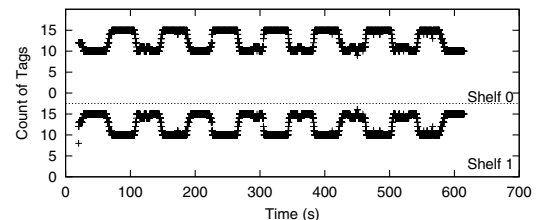
(a) Reality



(b) Application query results using raw data



(c) Application query results after *Smooth*



(d) Application query results after *Smooth* followed by *Arbitrate*

Figure 2. Shelf-monitoring application query results after different stages of cleaning

References

- [1] Alien ALL-9250 12 RFID tag. <http://www.alientechnology.com/products/rfid-tags>.
- [2] Alien ALR-9780 915 MHz RFID Reader. <http://www.alientechnology.com/products/rfid-readers/alr9780.php>.
- [3] EPCGlobal, Inc. <http://www.epcglobalinc.org/>.
- [4] S. R. Jeffery, *et al.* A Pipelined Framework for Online Cleaning of Sensor Data Streams. Technical Report UCB/CSD-05-1413, UC Berkeley, 2005. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/6474.html>.
- [5] E. Rahm *et al.* Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.
- [6] G. Tolle, *et al.* A macroscope in the redwoods. In *SenSys*. 2005.