



StreamClean: Near real-time RFID data cleaning

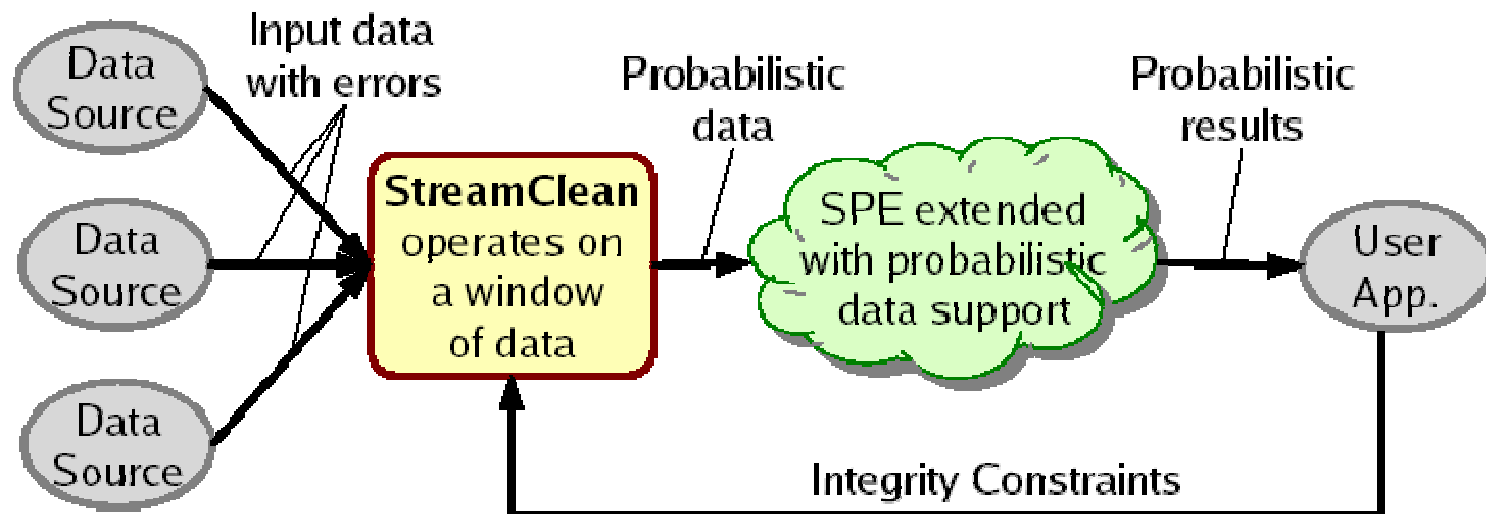
Nodira Khoussainova
Magdalena Balazinska
Dan Suciu
Ting-You Wang
and the RFID Ecosystem team

Introduction

- Applications frequently rely on devices such as:
 - RFID antennas
 - Light/motion/temperature sensors
- However, these devices are unreliable
 - Readings can be missed, duplicated, or simply erroneous.

Our Approach

- User or application specifies integrity constraints
- Constraint violations → input data errors
- StreamClean cleans input data probabilistically



[Outline]

- Constraint Language
- Constraint Taxonomy
- Detecting and Handling Errors
- Higher Level Events
- Probabilistic Constraints
- Future Work

[Constraint Language]

FORALL	INPUT1 as I1,..., INPUTn as In
WHERE	EXPR1
CHECK	EXPR2
CONF	C

[Constraints – Example 1]

Example from RFID-based tracking application

Constraint: “An object can not appear in more than one location at any time”

```
FORALL Sightings S
```

```
CHECK NOT EXISTS Sightings S1
```

```
    WHERE S.obj_id = S1.obj_id
```

```
    AND S.antenna_id <> S1.antenna_id
```

Constraints – Example 2

Example from RFID-based tracking application

Constraint: “If an object is sighted at antenna A, and then later at C they should have been sighted at B at some point in between.”

```
FORALL Sightings S1, Sightings S3
WHERE SEQ(S1, S3) AND S1.ant-id = 'A' AND
        S3.ant-id = 'C' AND S1.obj-id = S3.obj-id
CHECK EXISTS Sightings S2
        WHERE S2.obj-id = S1.obj-id
        AND S2.ant-id = 'B' AND ISLATER(S2, S1)
        AND ISLATER(S3, S2)
```

Constraint Taxonomy

	Stateless	Stateful
	Stateless	Stateful
Inclusion	Each object must appear in at least one location.	A person returning to the office must have previously passed by the elevator
Exclusion	Each object can appear in at most one location	A person who left the building cannot appear in the building

Hard constraints
must always hold

Soft constraints
usually hold

[Detecting and Handling Errors]

- Example 1: An object appears in three locations
 - Constraint enables error detection
 - StreamClean assigns probability $1/3$ to each sighting
- Example 2: An object is not sighted
 - Constraint enables error detection
 - StreamClean generates all N possible sightings
 - In the absence of other information, all locations are possible
 - Additional, possibly soft, constraints help reduce space (e.g. Ex 2)
 - StreamClean assigns probability $1/N$ to each sighting

Higher Level Events

- Use constraint language to define higher level events

E.g. “Seen at elevator immediately followed by at the door.”

```
FORALL Sightings S1, !Sightings S, Sightings S2
WHERE SEQ(S1, S, S2) AND S1.obj-id = S.obj-id
AND S1.obj-id = S2.obj-id AND
S1.ant-id = 'elevator1' AND
S2.ant-id = 'door3'
CHECK EXISTS LeftBldg L(obj-id)
```

- Write constraints on these higher level events to clean the data further.

Probabilistic Constraints

FORALL INPUT1 as I1,..., INPUTn as In

WHERE EXPR1

CHECK EXPR2

CONF C

Probabilistic Constraints

- Confidence supplied for constraints
 - Can be specified by user/application
 - Can be learned from history
 - How often is this constraint correct?
 - Parameters still need to be specified by user/application

[Future Work]

- Preliminary results
 - Assigned probabilities match intuition
 - Entropy maximization sufficiently fast
 - Example: 100 equations, 20 var/equ. solved in 200ms
 - But integrity constraints can get quite complex
- StreamClean Jr – on traditional database
- Extend StreamClean Jr to StreamClean
- Integrate with the RFID eco system

A decorative graphic consisting of a blue circle on the left, a grey horizontal bar in the center, and a blue bracket on the right. The text "Thank you!" is centered within the grey bar.

Thank you!

Any questions?