Child Safety System in a Shopping Center

Chan-Ching

Purpose

Design a system for helping monitor the location of young children in a shopping center with the help of appropriate sensors. The task is to design the backend data processing system using IBM Streams.

Setting

A shopping center has a new feature to improve the safety of young children within its premises. Each child entering the shopping center is provided with an ID with a chip on it (such as an RFID tag), and each ID has a unique identification number that can be mapped to the identity of the child. The child carries this ID as long as he/she is within the center.

The center has 27 rooms, 0 to 26, (we consider corridors as rooms too) and an additional room number "E" to mark the exit of the child from shopping center. A door connects one room to another. On each door, there is a sensor; if any child with an ID passes through this door, the sensor records the timestamp, the ID of the child passing through, the ID of the sensor itself, the room number from where the child is coming from, and the room number where child is going to. Each sensor produces a continuous stream of events with all this information, and sends it to a central analyzer. While in real-life this will be a continuous stream of events that has to be monitored in real-time, in this experiment, these events are stored in files for analysis.

Task

I. Every 5 minutes, output the location of each child in the shopping center. The update is of the form: <timestamp, child ID, current room>.
   If the current room of a child cannot be determined for some reason, "1000" is used for the current room. The timestamps provided in the event stream are used, not the system time within the operator. The output is a file "LocationLog.txt".
II. If a child has not been tracked by any sensor for more than 15 minutes, an alert is generated on a "missing child" stream, with the following
   <current time, child ID, time last seen, room last seen, parent phone number>.
   This stream is written to the output file 'MissingChildAlert.txt".
III. Since it is unsafe for kids to spend a long time in room 26, if a kid spends more than 5 minutes in room 26, an alert is generated and a phone call is made to the parent

of the kid. The alert has the following information <timestamp, child ID, parent phone number>, and is written to the output file “UnsafeRoomAlert.txt”.

A. In addition, the following statistics is output every hour. Note the “one hour” is relative to the timestamps on the events, and does not refer to the system time within the operator.

   For each room, the number of distinct children who visited the room over the last hour. This is written to the file “RoomStats.txt”, in the format <timestamp, room number, number of distinct children>

   Note that if a room already has a child “A” in it at the beginning of the hour, “A” counts as a child that visited the room during the hour.

B. For each child, the number of distinct rooms the child visited over the last hour. This is written to the file “ChildStats.txt”, in the format <timestamp, child number, number of distinct rooms>.

   Note that if a child was already in room “X” at the beginning of the hour, “X” counts as a room that the child visited using the hour.

Input Format

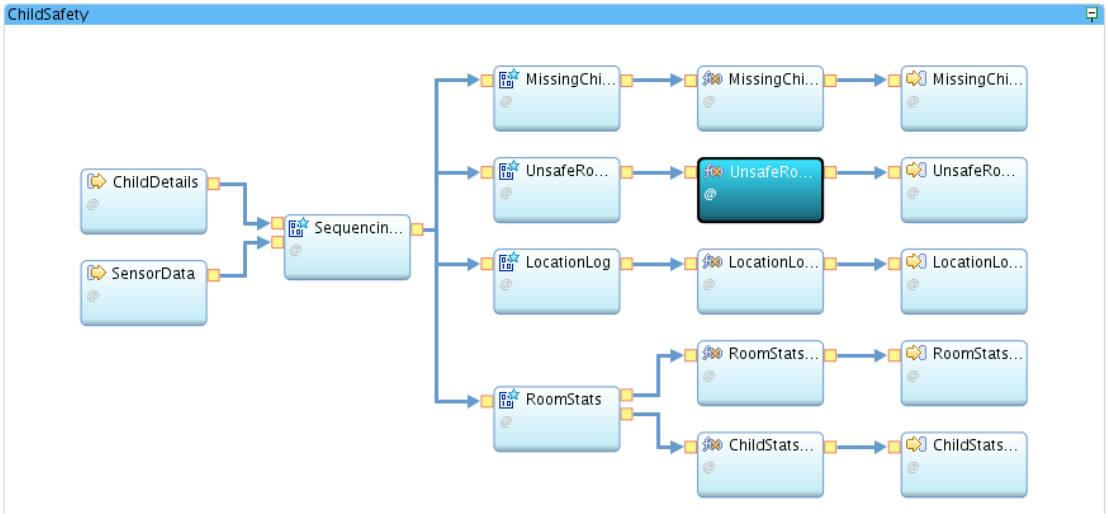
“ChildDetails.csv” has the following information for each child.

- Time when child entered
- Child ID
- Parent’s name
- Child Name
- Parent’s phone number

“SensorData.csv” contains the readings from all sensors.

- Timestamp
- Sensor ID
- Child ID
- From room
- To room

The snapshot of the data flow between the operators is as follows.



In a big picture, the streams coming out of the files, ChildDetails and SensorData (also the name of the respective operators in the figure), are integrated by the operator, SequencingStream. This operator takes data items arriving at the inputs, stores them in separate queues and output the data item whose timestamp is the earlies in the queues. It serves the purpose of simulating the real situation where events arrive one by one with system timestamp. Here the event order is based on the timestamps available in the available files.

To SequencingStream operator's right, there are four sequences of queue, each of which is designed for one of the tasks specified above. In other words, the queue starting with the LocationLog operator reports the last seen room of a child who is still in the shopping center for every 5 minutes. The queue of MissingChildAlert reports the children whose locations haven't been updated for 15 minutes from any sensor. UnsafeRoomAlert queue reports the children who has been in room 26 for more than 5 minutes, seeing no location update. RoomStats and ChildStats hourly report how many distinct children were in a room in the past one hour and how many distinct children were visited by a child in the previous hour, respectively.

The same stream is generated and fed to all the queues for different tasks, for which different logic need to be applied. For example, the Missing Child Alert will continue to be on until there is a location update for those children and children who have left the shopping center do not count (i.e. room 'E'). Similarly, Unsafe Room Alert will repeat the existing alert when a new stream arrives until those children leave room 26. For Room Stats and Child Stats, Room 1000 is considered and therefore included since children may only have entered the shopping center but not yet visited any other room.