Detecting Anomalies in Data Streams using Statecharts

Vasile-Marian Scuturici[1], Dan-Mircea Suciu[2], Romain Vuillemot[1], Aris Ouksel[3], Lionel Brunie[1]

[1] INSA Lyon
[2] Babes-Bolyai University, Cluj-Napoca
[3] Northwestern University, Illinois

Abstract. The environment around us is progressively equipped with various sensors, producing data continuously. The applications using these data face many challenges, such as data stream integration over an attribute (such as time) and knowledge extraction from raw data. In this paper we propose one approach to face those two challenges. First, data streams integration is performed using statecharts which represents a resume of data produced by the corresponding data producer. Second, we detect anomalous events over temporal relations among statecharts. We describe our approach in a demonstration scenario, that is using a visual tool called *Patternator*.

1 Introduction

The physical space surrounding us is progressively equipped with different sensors, giving us a digital view/projection of the real world. These sensors produce raw data streams, such as temperature values, images, and badge readings. Still, it remains difficult for humans to perceive interesting information from those raw data which are increasing in frequency and precision. In this demonstration we propose a mechanism to describe the behavior of data streams based on statecharts. We use the temporal attribute associated with a stream in order to detect relations between statecharts corresponding to non-homogenous datasources. The events non-holding these relations are considered as anomalies.

2 Statecharts modeling datasources

In our vision, a **datasource** is an abstraction for a fountain/fabric of data entities in the environment. Examples of datasources are the values read by a temperature sensor or the events produced by a badge reader. We associate the entities produces by a datasource with the datasource itself. These entities share the same structure, associated with the datasource.

The time plays a role in the description of a datasource *DS*. At every (discrete) moment *t*, *DS* is in a state defined by the produced entity at the instant *t*. The entities produced by a datasource, associated with a timestamp, form a **data stream**.

A *datasource DS* at a time *t* is a sequence of timestamped data entities sharing the same structure (header). Each data entity has values from the cartesian product $R^+ \times A^1 \times A^2 \times ... \times A^n$, where R^+ denotes the set of all positive real numbers (the time

domain) and A^i corresponds to the domain of datasource attributes. We consider the particular case where each domain A^i is categorical.

We consider a partition $\{S_1, S_2, \ldots, S_n\}$ of the space $A^1 \times A^2 \times \ldots \times A^n$. Each set S_i will be called a **state** of the datasource *DS*. The states $S_1, S_2, \ldots, S_n$ forms the **statechart** associated to the datasource *DS*. Each entity produced by *DS* will generate a transition in the associated statechart. If the last produced entity is called e_a and the new produced entity is e_b, then the added transition connects the two corresponding states containing these entities.

2.1 Relations between statecharts

The problem is the integration of datasources described by different structures. The only information useful to link these datasources is the timestamp associated to each produced entity. We use the timestamp to find define an inclusion relation between two states. $DS_1.S_1$ denotes the state S_1 of the statechart corresponding to the datasource DS_1. $DS_1.S_1$ is included in the state $DS_2.S_2$ if for each time when DS_1 is in S_1, DS_2 is in the state S_2:

For a predefined amount of data we detect all relations between the states composing the statecharts corresponding to studied datasources. All new events non-respecting these relations are qualified as anomalies.

3 Prototype and demonstration scenario

Our demonstration scenario aims at detecting anomalies in continuous data streams. The streams have different structures, and the timestamp is the single information used to link statecharts resuming these streams. Comparing to other approaches we use an integration of metadata (statecharts) automatically built on data streams. Another advantage of this approach: the statecharts have a powerful visual expression, useful for real-time monitoring applications.

The proposed prototype reads raw data from different datasources over HTTP (Scuturici, 2009). It can also load timestamped data from a relational database. *Patternator* automatically finds the relations between statecharts and suggests as anomalous events the events not respecting these relations.

As dataset, we used a subset of the dataset presented in (VAST, 2009). The subset contains one month of badges readers traces and computer network communications in a building. The goal is to detect suspect user behaviors, and since the ground truth is known we can evaluate our results and compare them to other published results.

References

Varun Chandola, Arindam Banerjee, Vipin Kumar (2009). *Anomaly detection: A survey*. ACM Comput. Surv. 41(3).

Scuturici, M. (2009) *Dataspace API*. Technical Report. LIRIS.

IEEE Symposium on Visual Analytics Science and Technology – VAST (2009), Atlantic City, New Jersey.