Tutorial: Formal Methods for Event Processing

Alexander Artikis and Georgios Paliouras
Institute of Informatics & Telecommunications, NCSR Demokritos, Athens, Greece
{a.artikis, paliourg}@iit.demokritos.gr

Organisations require techniques for automated transformation of the Big Data they collect into operational knowledge. This requirement may be addressed by employing event processing systems that detect activities/events of special significance within an organisation, given streams of low-level information that are difficult to be utilised by humans [4].

Systems for event processing and in particular event recognition ('event pattern matching') accept as input a stream of time-stamped, simple or low-level events. A low-level event is the result of applying a computational derivation process to some other event, such as an event coming from a sensor. Using low-level events as input, event processing systems identify composite or high-level events of interest — collections of events that satisfy some pattern. Consider, for example, the recognition of attacks on nodes of a computer network given the TCP/IP messages, the recognition of suspicious trader behaviour given the transactions in a financial market, and the recognition of whale songs given a symbolic representation of whale sounds.

Numerous event processing systems have been proposed in the literature [3]. Systems with a logic-based representation of event structures, for example, have been attracting considerable attention. They exhibit a formal, declarative semantics, allowing for verification and a code maintenance, they have proven to be efficient and scalable, and they are supported by machine learning tools, minimising human effort in the system development. In this tutorial, we review formal event processing systems.

High-level event 'definitions' impose temporal and, possibly, atemporal constraints on subevents, that is, low-level events or other high-level events. We will review a Chronicle Recognition System, the Event Calculus, ProbLog and Markov Logic Networks. The Chronicle Recognition System is a purely temporal reasoning system that allows for efficient event processing. It has been used in various domains, ranging from medical applications to computer network management. The Event Calculus allows for the representation of temporal, as well as atemporal constraints. Consequently, the Event Calculus may be used in applications requiring spatial reasoning, for example. ProbLog and Markov Logic Networks, unlike the Event Calculus and the Chronicle Recognition System, allow for uncertainty representation and are thus suitable for event processing in noisy environments.

The manual development of event definitions is a tedious, time-consuming and error-prone process. Moreover, it is often necessary to update such definitions due to new information about the application under consideration. For this reason, we review machine learning techniques automating the construction and refinement of event definitions.

To illustrate the reviewed approaches we use a real-world application from the PRONTO[1] and SPEEDD[2] projects: event processing for city transport and traffic management. In this domain, public transport vehicles, such as buses and trams, are equipped with sensors that report on position, in-vehicle temperature, noise level and acceleration. Moreover, fixed sensors are mounted on intersections and motorways to report on traffic flow and density. Given such low-level events, the task is to inform the decision-making of transport officials by recognising high-level events related to traffic congestion, the punctuality of a vehicle, passenger and driver comfort, passenger and driver safety, and passenger satisfaction.

The content of the tutorial is documented in two tutorial-level articles [2, 1].

Acknowledgements

This work is supported by the EU FP7 SPEEDD project (619435).

1. REFERENCES

[1] A. Artikis, O. Etzion, Z. Feldman, and F. Fournier. Event processing under uncertainty. In *Proceedings of International Conference on Distributed Event-Based Systems (DEBS)*, pages 32–43. ACM, 2012.

[2] A. Artikis, A. Skarlatidis, F. Portet, and G. Paliouras. Logic-based event recognition. *Knowledge Engineering Review*, 27(4):469–506, 2012.

[3] G. Cugola and A. Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys*, 44(3):15, 2012.

[4] O. Etzion and P. Niblett. *Event Processing in Action.* Manning Publications Co., 2010.

[1] `http://www.ict-pronto.org/`
[2] `http://speedd-project.eu/`

10.5441/002/edbt.2014.73