Event Recognition: Challenges and Current Techniques

Alexander Artikis, NCSR "Demokritos", Greece
Avigdor Gal, Technion – Israel Institute of Technology, Israel
Vana Kalogeraki, Athens University of Ecenomics and Business, Greece
Matthias Weidlich, Imperial College London, UK

1. INTRODUCTION

The concept of event processing is established as a generic computational paradigm in various application fields, ranging from data processing in Web environments, over logistics and networking, to finance and medicine [Cugola and Margara 2012]. Events report on state changes of a system and its environment. Event recognition (event pattern matching [Luckham 2002]), in turn, refers to the detection of events that are considered relevant for processing, thereby providing the opportunity to implement reactive measures. Examples consist of the recognition of attacks in computer network nodes [Dousson and Maigat 2007], human activities on video content [Brendel et al. 2011], emerging stories and trends on the Social Web[1], traffic and transport incidents in smart cities [Artikis et al. 2014b], fraud in electronic marketplaces [Schultz-Møller et al. 2009], cardiac arrhythmias [Callens et al. 2008], and epidemic spread [Chaudet 2006]. In each scenario, event recognition allows to make sense of large data streams and react accordingly.

Event recognition systems become increasingly important as we move from an information economy to an 'intelligent economy,' where it is not only the accessibility to information that matters but also the ability to analyse and act upon information, creating competitive advantage in commercial transactions, enabling sustainable management of communities, and promoting appropriate distribution of social, healthcare and educational services [Vesset et al. 2011]. Current businesses tend to be unable to make sense of the amounts of data that are generated by the increasing number of distributed data sources that are becoming available daily [Manyika et al. 2011] and will need to rely more and more on automated event recognition. As an example, consider traffic management in smart cities that needs to make use of data from an increasing number and variety of sensors.

An intelligent economy that makes use of Big Data can extract actionable knowledge from it by employing event recognition systems that detect events/activities of special significance given extremely large amounts of data spreading over various geographical locations. The goal of this paper is to provide an overview of the open research issues of event recognition as an introduction to this special issue. The rest of this paper is structured as follows. Section 2 presents a number of research challenges of event recognition, Section 3 briefly introduces the papers in this special issue, and Section 4 concludes the paper.

2. EVENT RECOGNITION RESEARCH CHALLENGES

Fig. 1 outlines the key aspects of the event recognition input, in terms of the event stream and the event patterns. On the one hand, the characteristics of the event stream, as they are described by the four "V"s and the distribution of events, impose challenges. That is, velocity (number of events per time unit), volume (overall amount of events), variety (differently structured events), lack of veracity (uncertainty of event

[1] `https://www.recordedfuture.com/`

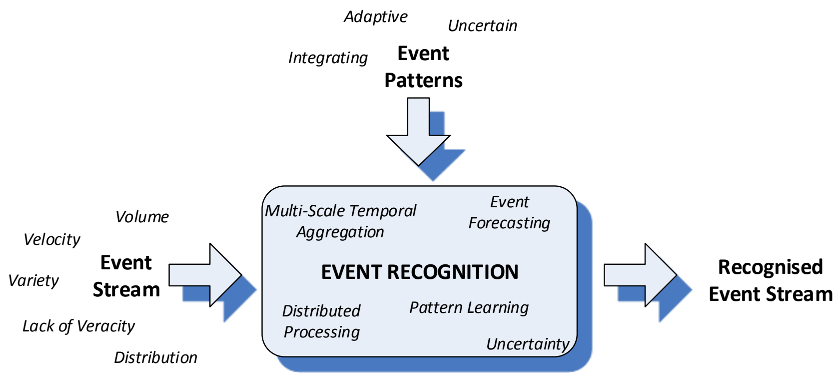

Fig. 1. Event recognition.

occurrence), and the distribution of event sources potentially involving mobility, complicate event recognition.

On the other hand, the properties of the event patterns are largely orthogonal to the characteristics of an event stream, and also add to the complexity of the event recognition task. Patterns may be required to adapt to dynamic environments. Further, patterns may integrate various event sources. Finally, patterns may be inherently uncertain in the sense that certain events can be recognised only within the bounds of a confidence interval.

Taking the dimensions outlined in Fig. 1 as a starting point, five research challenges of event recognition are discussed in this section, namely multi-scale temporal aggregation, uncertainty, distribution, pattern learning and event forecasting.

2.1. Multi-Scale Temporal Aggregation of Events

Composite events evolve over multiple scales of time and space [Vespier et al. 2013]. The variety of the event stream may be reflected by sources that report events at different time scales ranging from (milli-)seconds to days. For example, videos depict events at a small scale, whereas tweets may describe events at a much larger scale. Moreover, it is often the case that historical data spanning over long periods of time need to be taken into consideration [Dindar et al. 2011]. Consider, for instance, the recognition of traffic incidents using streaming bus probe data as well as several months of archived data [Bacon et al. 2011], and credit card fraud detection using streaming and archived transactions [Artikis et al. 2014a]. Once event patterns integrate events from sources that greatly vary in their adopted scales for time and space, event recognition becomes challenging. A recognition system should be adaptable, computing dynamically the appropriate lengths of multi-granular windows of varying levels of detail, and remain accurate, being able to recognise composite events from lower-level events of varying spatio-temporal granularity, without of course compromising efficiency [Maier et al. 2012; Patroumpas 2013; Lijffijt et al. 2012].

2.2. Event Recognition under Uncertainty

Event recognition applications exhibit various types of uncertainty. Input event streams are often incomplete and include erroneous information. Sensor networks introduce uncertainty due to reasons that range from inaccurate measurements through local network failures to unexpected interference of mediators. For all of these reasons, input event streams lack veracity.

Furthermore, the rules expressing an event pattern can be imprecise, meaning that there is also uncertainty related to the events that shall be recognised. In many application domains, we only have imprecise knowledge about the definition of a composite event, or the available events and context information are insufficient for expressing a composite event. Consider, for example, the recognition of a fight between two people in a system of event sources that cannot distinguish between abrupt and non-abrupt human movement.

Surprisingly, uncertainty has been largely overlooked by the event processing community [Cugola and Margara 2012]. [Gal et al. 2011; Wasserkrug et al. 2012] are but a few exceptions. On the other hand, in computer vision several approaches that deal with uncertainty in event recognition have been suggested. (An overview of event recognition under uncertainty may be found in [Skarlatidis et al. 2014].) Since event recognition requires the processing of streams of time-stamped events, numerous approaches are based on sequential variants of probabilistic graphical models, such as Hidden Markov Models, Dynamic Bayesian Networks and linear-chain Conditional Random Fields. Such models can naturally handle uncertainty but their propositional structure provides limited representation capabilities. To overcome this limitation, probabilistic graphical models have been extended to model interactions between multiple entities [Vail et al. 2007], capture long-term dependencies between states [Hongeng and Nevatia 2003], and model the hierarchical composition of events [Natarajan and Nevatia 2007]. However, the lack of a formal representation language makes the definition of composite events complicated and the use of background knowledge very hard.

Recently, statistical relational learning methods have been applied to event recognition. These methods combine logic with probabilistic models, in order to represent complex relational structures and perform reasoning under uncertainty. Markov Logic Networks (MLN)s [Domingos and Lowd 2009], a generic statistical relational learning framework that subsumes various probabilistic graphical models, have been attracting attention. For example, Tran and Davis [2008] and Kembhavi et al. [2010] use MLNs to take into account the confidence values of the input events. Morariu and Davis [2011] propose an MLN-based method that uses interval relations from Allen's algebra [Allen 1983]. The method determines the most consistent sequence of composite events, based on the observations of low-level classifiers. Sadilek and Kautz [2012] employ hybrid-MLNs [Wang and Domingos 2008] in order to recognise successful and failed interactions between humans, using noisy location data from GPS devices. Skarlatidis et al. [2014] express the Event Calculus [Kowalski and Sergot 1986] in MLNs to perform event recognition.

Although there is considerable work on optimising probabilistic reasoning techniques, the imposed overhead does not allow for real-time performance in a wide range of applications. This is then a key challenge of event recognition in uncertain environments.

2.3. Distributed Event Recognition

The volume and velocity of event streams is continuously growing and the increasing scale at which events are required to be processed poses challenges both in terms of computational resources and in terms of communication resources. Computational scalability issues are addressed by distributing event recognition tasks among multiple nodes, while communication scalability issues are addressed by algorithms that perform as much of the processing as possible on the event sources, thus minimising the amount of information transferred between nodes (a discussion about these issues may be found in [Artikis et al. 2014a]).

Several approaches have been proposed for distributing an event recognition task among multiple nodes. Brenna et al. [2009], for example, evaluate empirically strategies for distributing the execution of automata expressing queries in the Cayuga event language [Brenna et al. 2007]. Semantic dependencies between event queries are used by Lakshmanan et al. [2009] to identify strata of independent queries that are deployed on different (sets of) processing units. Then, profiling is applied to guide the assignment of the processing units in a stratum to the respective queries in order to maximise throughput. Schultz-Møller et al. [2009] employ query rewriting for efficient query execution, and distribute the recognition process to enable the system to scale to the rate of incoming events. Queries are compiled into detection automata, and the system deploys new automata in a greedy manner. Balkesen et al. [2013] propose a method for distributing input events that belong to individual run instances of the finite state machine of an event pattern to different processing units, thereby providing fine-grained partitioned data parallelism [Hirzel 2012] that is independent from the event pattern.

As mentioned in Section 2.2, the task of event recognition is inherently uncertain and, therefore, deterministic techniques are often unsuitable. The challenge thus is to develop methods for distributing *probabilistic* event recognition tasks, which are fundamentally different from deterministic event recognition tasks.

In addition to the computational scalability issues discussed above, the increasing distribution of event sources requires that network resources are utilised efficiently. Managing bandwidth usage is also a key requirement for mobility-aware event recognition, where the events of interest are detected in sensor networks including mobile devices such as smartphones and tablets. For instance, it is necessary to deal with highly dynamic event consumers, whose interests change with their location. Since communication efficiency reduces the volume of data sent to a data centre for processing, it also supports computational efficiency. Moreover, communication efficiency supports the privacy of event sources.

Communication-efficient distributed recognition has been an active research field. Proposed methods reduce communication by decomposing the recognition task into a set of local constraints on the data generated at the event sources. The constraints are such that as long as all of them are satisfied, it is guaranteed that the event of interest has not occurred. Consequently, as long as all constraints are satisfied, no communication is required. The event to be recognised is usually defined using a function over aggregate values derived at the event sources. Research on recognising such types of event includes sketching [Papapetrou et al. 2012], in which data summaries are sent, thus reducing communication overhead, and geometric methods for expressing constraints at the event sources [Sharfman et al. 2006; Giatrakos et al. 2014; Keren et al. 2014].

Most of the communication minimisation literature has focused on events defined as functions over aggregate values. Event recognition also requires matching logical, temporal and spatial event patterns. A key challenge thus lies in developing communication minimisation techniques covering all types of event pattern.

2.4. Event Pattern Learning

Machine learning techniques may be used for constructing and/or adapting event patterns in a dynamic and evolving environment. Both supervised and unsupervised techniques have been employed to automatically adapt and construct event patterns. A widely used unsupervised learning technique is the frequency-based analysis of sequences of events (e.g. [Yu et al. 2004; Lee and Lee 2005; Vautier et al. 2007; Álvarez et al. 2010; Calders et al. 2014]). Frequency-based analysis is a promising approach for discovering unknown events in databases or logs, but is limited to propositional learning. Moreover, this technique may not be adapted to learning the structure of patterns of composite events that are not frequent in data—in some applications, including credit card fraud management, the recognition of such events (fraud) is of utmost importance.

A common technique for learning the structure of composite event patterns in a supervised manner involves the use of Inductive Logic Programming (ILP) [Muggleton and Raedt 1994] (for example, [Callens et al. 2008]). ILP may construct patterns that capture exceptional cases in an event stream. On the other hand, ILP does not handle numerical reasoning, which is quintessential in the representation of event patterns. In the case of partial supervision, ILP is used in combination with abduction [Denecker and Kakas 2002] in order to learn an event pattern [Ray 2009; Corapi et al. 2011; Athakravi et al. 2013]. This combination of techniques, however, does not scale to the volume and velocity of event streams observed in practice.

In addition to learning the structure of an event pattern, the confidence values/weights attached to the pattern can be learned from data [Domingos and Lowd 2009]. Usually the tasks of structured learning and weight learning are separated; that is, first the structure of an event pattern is learnt and then the weights of the pattern are estimated. Separating the two learning tasks in this way, however, may lead to suboptimal results, as the first optimisation step (structure learning) needs to make assumptions about the weight values, which have not yet been optimised.

2.5. Event Forecasting

Rapid social, economic and political changes are leading organisations to shift their thinking from reactive to proactive in order to detect opportunities and threats that could affect their business [Burton et al. 2010]. Changing traffic light policies and speed limits to avoid forecast traffic congestions, for example, will reduce carbon emissions, optimise public transportation and increase the quality of life and productivity of commuters. In energy management, there is a need for real-time optimisation of power consumption in individual houses and buildings equipped with renewable energy sources. This requirement may be addressed by forecasting energy consumption and production, say for the next 30 minutes, and making decisions about load adjustments and/or rescheduling.

Proactive event-driven computing systems exhibit the ability to eliminate or mitigating anticipated problems, or capitalise on forecast opportunities, by event forecasting and decision-making [Engel and Etzion 2011; Engel et al. 2012; Feldman et al. 2013]. Event forecasting may be achieved by 'forward' event recognition—the ability of an event processing system to recognise events incrementally. The system reports partially recognised events, that is, events for which a subset of the constraints expressing the event pattern are satisfied. Such events may or may not be completely recognised in the future.

Perhaps the most successful system for forward event recognition is the Chronicle Recognition System (CRS) [Dousson and Maigat 2007]. CRS has proven to be very efficient and scalable. It is a purely temporal reasoning system, however, and therefore, cannot be directly applied to any application requiring spatial reasoning. Furthermore, CRS does not deal with uncertainty. Consequently, CRS is insufficient for applications that lack veracity. Note also that event forecasting needs to indicate the probability of a forecast event, as well as the probability of when an event will happen—a probability distribution over the event occurrence time must be provided.

3. IN THIS SPECIAL ISSUE

This section briefly introduces each of the six papers that were selected for this special issue.

3.1. Approximate Semantic Matching of Events for The Internet of Things

Souleiman Hasan and Edward Curry propose an approach for deriving semantic similarity and relatedness of events that do not require a prior agreement between event providers and event consumers on the schema of events. The paper is motivated by the increasing amount of events created through the Internet-of-Things. In the absence of an agreement on event schema, participants may loosely agree on topics represented in large corpora of texts, which in turn are used for approximate event matching. The authors tackle the challenges of event recognition under uncertainty, using top-K matchers along with a probability model, and event pattern learning. The paper offers a formal framework and empirical validation for time efficiency, and presents insights on the effect of the degree of approximation on the model.

3.2. PADUA: Parallel Architecture to Detect Unexplained Activities

Cristian Molinaro, Vincenzo Moscato, Antonio Picariello, Andrea Pugliese, Antonino Rullo and V. S. Subrahmanian present an approach for identifying situations that cannot be satisfactorily explained by any of the known event patterns. Their motivation stems from the fact that, in many applications, such as public space surveillance, credit card transactions and computer networks, there is a need to recognise irregularities, that is, behaviours for which there is no known pattern. Molinaro et al. first propose probabilistic penalty graphs, an extension of stochastic automata, to deal with the inherent uncertainty of event recognition. Then, they propose parallel coordination algorithms for distributed event recognition using probabilistic penalty graphs. The empirical evaluation consists of recognition over 160 CPUs using real-world datasets from video surveillance and network traffic.

3.3. Adaptive Speculative Processing of Out-of-Order Event Streams

Christopher Mutschler and Michael Phlippsen propose an adaptive speculative processing technique for out-of-order event streams in distributed event recognition. Out-of-order event arrival has been handled by using buffering techniques to delay events. This work combines buffering with speculative processing, and adapts the degree of speculation at run-time to fit the available system resources so that detection latency becomes minimal. The empirical evaluation shows improvement over existing approaches both on synthetic data and real data from a real-time locating system with several thousands of out-of-order sensor events per second.

3.4. Decentralised Fault Tolerant Event Correlation

Gregory Aaron Wilkin, Patrick Eugster and K. R. Jayaram argue that many use cases for event recognition require fault-tolerant processing in terms of guarantees of event delivery. Providing these guarantees in a decentralised infrastructure is challenging, though. The paper addresses this issue of distributed event recognition by means of FAIDECS, a model and system that provides guarantees of event delivery. In particular, the authors present a formal analysis of the interplay of the choices of event matching and disposal semantics, and the guarantees on safety, liveness, agreement and order properties of event recognition. In addition, the paper links these findings to the StreamSQL, EQL, CEL and TESLA languages, making it possible to assess the guarantees provided by these languages. Finally, FAIDECS is experimentally shown to scale better than centralised processing or existing solutions for decentralised, fault-tolerant processing.

3.5. MCEP: A Mobility-aware Complex Event Processing System

Beate Ottenwälder, Boris Koldehofe, Kirak Hong, David Lilethun, Umakishore Ramachandran and Kurt Rothermel present a middleware that handles efficiently highly dynamic mobile consumers, whose interests change with their location. The motivation of this work stems from the proliferation of mobile devices and sensors that offer increasing amounts of information. The proposed distributed Mobile Complex Event Processing (MCEP) system automatically adapts the processing of events according to a consumer's location. MCEP reduces latency, network utilisation and processing overhead by providing on-demand and opportunistic adaptation algorithms to dynamically assign event streams and computing resources to the MCEP operators. MCEP is evaluated on traffic accident detection, video friend finder and vehicle speed detection.

3.6. Efficient Stream Provenance via Operator Instrumentation

Boris Glavic, Kyumars Sheykh Esmaili, Peter M. Fischer and Nesime Tatbul introduce an approach that uses operator instrumentation, that is, modification of the behaviour of operators, to generate and propagate fine-grained provenance through several operators of a query network. The motivation of this work stems from the diagnostic and assurance requirements of a wide range of applications. In addition to computing provenance eagerly during query execution, the authors study how to decouple provenance computation from query processing in order to reduce run-time overhead and avoid unnecessary provenance retrieval. Ariadne, the provenance-aware extension of the Borealis stream management system, implements the proposed techniques, and manages provenance with minor overhead.

4. CONCLUSIONS

The new era of Big Data brings with it opportunities with a potential of advancing the world technology to new heights. One can envision a world where one wakes up in a smart home, networked with green technology, dresses up in clothes that can monitor day-to-day health, and uses a smartly monitored public transportation to get safely and quickly to work or a leisure place. Growing older has the promise of careful health monitoring, improved medical devices, avoiding a helpless embarrassing last years on earth. This special issue provides a few small steps in acquiring the technology for this world, and unlock the potential of Big Data, which undoubtedly awaits us.

Acknowledgments

This work is funded by the EU FP7 projects SPEEDD (619435) and INSIGHT (318225).

We have benefited from the description of the event recognition challenges of Elias Alevizos.

REFERENCES

ALLEN, J. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM 26,* 11, 832–843.

ÁLVAREZ, M. R., FÉLIX, P., CARIÑENA, P., AND OTERO, A. 2010. A data mining algorithm for inducing temporal constraint networks. In *International Conference on Information Processing and Management of Uncertainty (IPMU)*. 300–309.

ARTIKIS, A., BABER, C., BIZARRO, P., DE WIT, C. C., ETZION, O., FOURNIER, F., GOULART, P., HOWES, A., LYGEROS, J., PALIOURAS, G., SCHUSTER, A., AND SHARFMAN, I. 2014a. Scalable proactive event-driven decision-making. *IEEE Technology and Society Magazine*.

ARTIKIS, A., WEIDLICH, M., SCHNITZLER, F., BOUTSIS, I., LIEBIG, T., PIATKOWSKI, N., BOCKERMANN, C., MORIK, K., KALOGERAKI, V., MARECEK, J., GAL, A., MANNOR, S., GUNOPULOS, D., AND KINANE,

D. 2014b. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *International Conference on Extending Database Technology (EDBT)*. 712–723.

ATHAKRAVI, D., CORAPI, D., BRODA, K., AND RUSSO, A. 2013. Learning through hypothesis re nement using answer set programming. In *Proceedings of International Conference of Inductive Logic Programming (ILP)*.

BACON, J., BEJAN, A. I., BERESFORD, A. R., EVANS, D., GIBBENS, R. J., AND MOODY, K. 2011. Using real-time road traffic data to evaluate congestion. In *Dependable and Historic Computing*. 93–117.

BALKESEN, C., DINDAR, N., WETTER, M., AND TATBUL, N. 2013. Rip: run-based intra-query parallelism for scalable complex event processing. In *DEBS*. 3–14.

BRENDEL, W., FERN, A., AND TODOROVIC, S. 2011. Probabilistic event logic for interval-based event recognition. In *CVPR*. 3329–3336.

BRENNA, L., DEMERS, A. J., GEHRKE, J., HONG, M., OSSHER, J., PANDA, B., RIEDEWALD, M., THATTE, M., AND WHITE, W. M. 2007. Cayuga: a high-performance event processing engine. In *SIGMOD Conference*. 1100–1102.

BRENNA, L., GEHRKE, J., HONG, M., AND JOHANSEN, D. 2009. Distributed event stream processing with non-deterministic finite automata. In *DEBS*.

BURTON, B., GENOVESE, Y., RAYNER, N., CASONATO, R., SMITH, M., BEYER, M. A., AUSTIN, T., GASSMAN, B., AND SOMMER, D. 2010. Pattern-based strategy technologies and business practices gain momentum. Gartner Report G00208030.

CALDERS, T., DEXTERS, N., GILLIS, J. J. M., AND GOETHALS, B. 2014. Mining frequent itemsets in a stream. *Inf. Syst. 39*, 233–255.

CALLENS, L., CARRAULT, G., CORDIER, M.-O., FROMONT, É., PORTET, F., AND QUINIOU, R. 2008. Intelligent adaptive monitoring for cardiac surveillance. In *Proceedings of European Conference on Artificial Intelligence (ECAI)*. 653–657.

CHAUDET, H. 2006. Extending the event calculus for tracking epidemic spread. *Artificial Intelligence in Medicine 38,* 2.

CORAPI, D., RUSSO, A., AND LUPU, E. 2011. Inductive logic programming in answer set programming. In *ILP*. 91–97.

CUGOLA, G. AND MARGARA, A. 2012. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys 44,* 3, 15.

DENECKER, M. AND KAKAS, A. 2002. Abduction in logic programming. In *Computational Logic: Logic Programming and Beyond*, A. Kakas and F. Sadri, Eds. Lecture Notes in Computer Science Series, vol. 2407. Springer, 99–134.

DINDAR, N., FISCHER, P. M., AND TATBUL, N. 2011. Dejavu: a complex event processing system for pattern matching over live and historical data streams. In *DEBS*. 399–400.

DOMINGOS, P. AND LOWD, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool Publishers.

DOUSSON, C. AND MAIGAT, P. L. 2007. Chronicle recognition improvement using temporal focusing and hierarchisation. In *IJCAI*. 324–329.

ENGEL, Y. AND ETZION, O. 2011. Towards proactive event-driven computing. In *DEBS*. 125–136.

ENGEL, Y., ETZION, O., AND FELDMAN, Z. 2012. A basic model for proactive event-driven computing. In *DEBS*. 107–118.

FELDMAN, Z., FOURNIER, F., FRANKLIN, R., AND METZGER, A. 2013. Proactive event processing in action: a case study on the proactive management of transport processes (industry article). In *DEBS*. 97–106.

GAL, A., WASSERKRUG, S., AND ETZION, O. 2011. Event processing over uncertain data. In *Reasoning in Event-Based Distributed Systems*, S. Helmer, A. Poulovassilis, and F. Xhafa, Eds. Springer, 279–304.

GIATRAKOS, N., DELIGIANNAKIS, A., GAROFALAKIS, M., SHARFMAN, I., AND SCHUSTER, A. 2014. Distributed geometric query monitoring using prediction models. *ACM TODS*.

HIRZEL, M. 2012. Partition and compose: parallel complex event processing. In *DEBS*. 191–200.

HONGENG, S. AND NEVATIA, R. 2003. Large-scale event detection using semi-hidden markov models. In *ICCV*. 1455–1462.

KEMBHAVI, A., YEH, T., AND DAVIS, L. S. 2010. Why did the person cross the road (there)? scene understanding using probabilistic logic models and common sense reasoning. In *ECCV (2)*. 693–706.

KEREN, D., SAGY, G., ABBOUD, A., BEN-DAVID, D., SCHUSTER, A., SHARFMAN, I., AND DELIGIANNAKIS, A. 2014. Geometric monitoring of heterogeneous streams. *IEEE TKDE*.

KOWALSKI, R. AND SERGOT, M. 1986. A logic-based calculus of events. *New Generation Computing 4,* 1, 67–96.

LAKSHMANAN, G. T., RABINOVICH, Y. G., AND ETZION, O. 2009. A stratified approach for supporting high throughput event processing applications. In *DEBS*, A. S. Gokhale and D. C. Schmidt, Eds. ACM.

LEE, D. AND LEE, W. 2005. Finding maximal frequent itemsets over online data streams adaptively. In *ICDM*. IEEE Computer Society, 266–273.

LIJFFIJT, J., PAPAPETROU, P., AND PUOLAMÄKI, K. 2012. Size matters: Finding the most informative set of window lengths. In *ECML/PKDD (2)*. 451–466.

LUCKHAM, D. 2002. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley.

MAIER, D., GROSSNIKLAUS, M., MOORTHY, S., AND TUFTE, K. 2012. Capturing episodes: may the frame be with you. In *DEBS*. 1–11.

MANYIKA, J., CHUI, M., BROWN, B., BUGHIN, J., DOBBS, R., ROXBURGH, C., AND BYERS, A. H. 2011. Big data: The next frontier for innovation, competition, and productivity.

MORARIU, V. I. AND DAVIS, L. S. 2011. Multi-agent event recognition in structured scenarios. In *CVPR*. 3289–3296.

MUGGLETON, S. AND RAEDT, L. D. 1994. Inductive logic programming: Theory and methods. *Journal of Logic Programming 19/20*, 629–679.

NATARAJAN, P. AND NEVATIA, R. 2007. Hierarchical multi-channel hidden semi markov models. In *IJCAI*. 2562–2567.

PAPAPETROU, O., GAROFALAKIS, M. N., AND DELIGIANNAKIS, A. 2012. Sketch-based querying of distributed sliding-window data streams. *PVLDB 5,* 10, 992–1003.

PATROUMPAS, K. 2013. Multi-scale window specification over streaming trajectories. *J. Spatial Information Science 7,* 1, 45–75.

RAY, O. 2009. Nonmonotonic abductive inductive learning. *Journal of Applied Logic 7,* 3, 329–340.

SADILEK, A. AND KAUTZ, H. A. 2012. Location-based reasoning about complex multi-agent behavior. *J. Artif. Intell. Res. (JAIR) 43*, 87–133.

SCHULTZ-MØLLER, N. P., MIGLIAVACCA, M., AND PIETZUCH, P. R. 2009. Distributed complex event processing with query rewriting. In *DEBS*.

SHARFMAN, I., SCHUSTER, A., AND KEREN, D. 2006. A geometric approach to monitoring threshold functions over distributed data streams. In *SIGMOD Conference*. 301–312.

SKARLATIDIS, A., PALIOURAS, G., ARTIKIS, A., AND VOUROS, G. 2014. Probabilistic event calculus for event recognition. *ACM Transactions on Computational Logic*. Preprint available from `http://arxiv.org/abs/1207.3270`.

TRAN, S. D. AND DAVIS, L. S. 2008. Event modeling and recognition using markov logic networks. In *ECCV (2)*. 610–623.

VAIL, D. L., VELOSO, M. M., AND LAFFERTY, J. D. 2007. Conditional random fields for activity recognition. In *AAMAS*. 235.

VAUTIER, A., CORDIER, M.-O., AND QUINIOU, R. 2007. Towards data mining without information on knowledge structure. In *PKDD*. 300–311.

VESPIER, U., NIJSSEN, S., AND KNOBBE, A. J. 2013. Mining characteristic multi-scale motifs in sensor-based time series. In *CIKM*. 2393–2398.

VESSET, D., FLEMMING, M., AND SHIRER, M. 2011. Worldwide decision management software 2010–2014 forecast: A fast-growing opportunity to drive the intelligent economy. IDC report 226244.

WANG, J. AND DOMINGOS, P. 2008. Hybrid markov logic networks. In *AAAI*. 1106–1111.

WASSERKRUG, S., GAL, A., ETZION, O., AND TURCHIN, Y. 2012. Efficient processing of uncertain events in rule-based systems. *IEEE Trans. Knowl. Data Eng. 24,* 1, 45–58.

YU, J. X., CHONG, Z., LU, H., AND ZHOU, A. 2004. False positive or false negative: Mining frequent itemsets from high speed transactional data streams. In *VLDB*, M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, Eds. Morgan Kaufmann, 204–215.