Complex Event Recognition under Uncertainty: A Short Survey

Elias Alevizos[1], Anastasios Skarlatidis[1], Alexander Artikis[2 1], Georgios Paliouras[1]
[1]National Centre for Scientific Research (NCSR) "Demokritos", Greece
[2]University of Piraeus, Greece
{alevizos.elias, anskarl, a.artikis, paliourg}@iit.demokritos.gr

ABSTRACT

Complex Event Recognition (CER) applications exhibit various types of uncertainty, ranging from incomplete and erroneous data streams to imperfect complex event patterns. We review CER techniques that handle, to some extent, uncertainty. We examine both automata-based techniques, which are the most often, and logic-based ones, which are less frequently used. A number of limitations are identified with respect to the employed languages, their probabilistic models and their performance, as compared to the purely deterministic cases.

1. INTRODUCTION

Systems for Complex Event Recognition (CER) accept as input a stream of time-stamped simple, derived events (SDE)s. A SDE ('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 SDEs as input, CER systems identify complex events (CE)s of interest—collections of events that satisfy some pattern. The 'definition' of a CE ('high-level event') imposes temporal and, possibly, atemporal constraints on its subevents, i.e. SDEs or other CEs. For example, consider the recognition of attacks on computer network nodes, given the TCP/IP messages. A CER system attempting to detect a DOS attack has to identify (as one possible scenario) both a forged IP address that fails to respond and that the rate of requests is unusually high.

Due to the complex nature of information sources, the input events arriving at a CER system almost always carry a certain degree of uncertainty and/or ambiguity. Sensor networks introduce uncertainty into the system due to reasons that range from inaccurate measurements through network local failures to unexpected interference of mediators. The latter is a new phenomenon that stems from the distribution of sensor sources. Sensor data may go through multiple mediators en route to the CER systems. Such mediators apply filtering and aggregation mechanisms, most of which are unknown to the system that receives the data. For example, a road sensor collecting traffic data may calculate the average speed of cars passing over it within a time period, but this calculation might not be accurate, it might be corrupted or it might even fail to reach the CER system, due to some network failure, unrelated to the sensor. Again, in the traffic management domain, it might not be possible to define all the possible situations which indicate the occurrence of an accident. Hence, the uncertainty that is inherent to sensor data is multiplied by the factor of unknown aggregation and filtering treatments [5]. Even if we assume perfectly accurate sensors, the domain under study might be difficult or impossible to model precisely, thereby leading to another type of uncertainty.

Until recently, most CER systems did not make any effort to handle uncertainty [9]. This need is gradually being acknowledged and it seems that this might constitute a significant line of research and development for CER. Almost all of the papers presented have appeared after 2008. The purpose of this paper is to present a short overview of existing approaches for performing CER under uncertainty. It should be noted that handling uncertainty in activity recognition (where SDEs come mainly from video streams or RFID tracks) is an active research field that has strong similarities with CER. However, in this short survey we have chosen to present only those methods that come directly from the field of CER.

The structure of the paper is as follows: In Section 2 we discuss the dimensions along which a proposed solution for handling uncertainty may be evaluated. Section 3 presents the reviewed approaches, summarizes them in a tabular form and comments on their limitations. Some open issues and lines of potential future work are identified in Section 4.

2. EVALUATION DIMENSIONS

We restrict attention to the following types of uncertainty. First, the rules defining a CE may be imperfect. Second, the SDE stream may be incomplete and/or include erroneous events. Detailed discussions about types and sources of uncertainty in CER may be found in [4, 22].

We follow the customary division between representation, inference and learning. In other words, we are interested in what kind of knowledge a system can encode (representation), what kind of queries it can answer (inference) and if/what parameters and models it can learn. However, although learning in general is a very active research area, we have decided not to include a detailed discussion about the learning capabilities of the examined approaches in our survey. The reason is quite simple. Almost none of the systems touches upon this subject. Instead, we draw some conclusions as far as the performance of each system is concerned.

2.1 Representation

Following the terminology of [15], we define an event as an object in the form of a tuple of data components, signifying an activity and holding certain relationships to other events by time, causality and aggregation. An event with N attributes can be represented as

$$E(Type, ID, Attribute1, \ldots, AttributeN, Time)$$

where $Time$ might be a point, in case of an instantaneous event, or an interval during which the event happens, if it is durative. In CER, we are interested in detecting patterns of events among the streams of SDEs. Therefore, we need a language for expressing such pattern detection rules.

Formalisms for reasoning about events and time have appeared in the past, such as the Event Calculus [6, 14] and Allen's Interval Algebra [2,3], and have already been used for defining event algebras (e.g. in [18]). With the help of the theory of descriptive complexity, recent work has also identified those constructs of an event algebra which strike a balance between expressive power and complexity [27]. Based on the capabilities of existing CER systems and on related theoretical work, the following list enumerates those operations that should be supported by a CER engine:

- $Sequence$: Two events following each other in time.
- $Disjunction$: Either of two events occurring, regardless of temporal relations. Conjunction (both events occurring) may be expressed by combining $Sequence$ and $Disjunction$.
- $Iteration$: An event occurring N times in sequence, where $N \geq 0$.
- $Negation$: Event not occurring at all.
- $Selection$: Select those events whose attributes satisfy a set of predicates/relations, temporal or otherwise.
- $Projection$: Return an event whose attribute values are a possibly transformed subset of the attribute values of its sub-events.
- $Windowing$: Apply pattern for events within a specified time window.

In a probabilistic setting, uncertain events are assigned an occurrence probability. More complex models also allow for probabilities on the attributes of the events as well. Furthermore, the rules for expressing CE definitions may also be probabilistic. The semantics for the probability space are usually those of possible worlds. A possible world is one of the possible SDE streams, as defined by the SDE probabilities. Thus, the probability space is understood as the set of all the alternative event streams that may have occurred and the distribution is defined over this set. Event attributes are usually discrete and the continuous case is outside the scope of most CER systems.

2.2 Inference

In probabilistic CER, the most basic inference task is to compute the probability of occurrence of a CE. In other words, the task is to compute the marginal probabilities of the CEs, given the SDEs. In some settings, we might also be interested in performing maximum a posteriori (MAP) inference, in which the task is to compute the most probable states of some CEs, given the evidence SDEs stream. A simple example from the domain of video recognition is the query in which the user asks about the most probable time interval during which a certain activity occurs.

Another dimension concerns the ability of a system to perform approximate inference. In the literature of statistical relational learning, it is widely believed that for all but the simplest cases, exact inference stumbles upon serious performance issues, unless several simplifying assumptions are made. For this reason, approximate inference is considered essential. When this capability is present, certain systems provide answers with confidence intervals and/or the option of setting a confidence threshold above which an answer may be accepted.

2.3 Performance

CER systems are usually evaluated for their performance in terms of throughput, measured as number of events processed per second. For some queries, the latency, as measured by the time required to process an event, is also important. Less often, the memory footprint is reported. Note that no standard benchmarks exist, although some work towards this direction has begun [12, 16, 17]. Reporting throughput figures is not enough by itself, since there are multiple factors which can affect performance, such as query selectivity (see [16] for a list of such factors). When uncertainty is introduced, the complexity of the problem grows and other performance-affecting factors enter the picture, such as the option of approximate inference. Moreover, systems need to be evaluated along another dimension, that of accuracy.

The issue of accuracy is of critical importance and is not orthogonal to that of performance. Precision and recall are the usual measures of accuracy, but neither one of them may be sufficient by itself. Therefore, a more appropriate measure would be that of the F-measure, i.e. the harmonic mean of precision and recall.

3. APPROACHES

Since many of the CER engines employ finite automata, either deterministic (DFA) or non-deterministic (NFA), it is not surprising that automata are one of the dominant approaches for handling uncertainty. Less frequently, logic-based approaches are preferred. In this section, we present both of these areas.

We summarize our results in Tables 1 - 3. The columns of Table 1 correspond to the list of operators presented in Section 2.1 and refer to the expressive power of the language employed. An extra column has been added to indicate whether a system supports event hierarchies, i.e. the ability to define CEs at various levels and reuse those intermediate inferred events in order to infer other higher-level events. In Table 2 we present the probabilistic properties of each method, with respect to the independence assumptions they make and to their capacity for assigning probabilities to the input data (SDEs) and/or the rules for CE definitions. Some systems

Language Expressivity										
Paper	σ	π	$\wedge$	$\vee$	$\neg$	;	*	W	H	Remarks
Kawashima et al [13]	✓	✓				✓	✓	✓		
Re et al [19]	✓					✓	✓			
Chuanfei et al [7]					✓	✓	✓	✓		Not enough details in paper about σ, π, $\wedge$, $\vee$, $\neg$.
Shen et al [20]	✓	✓				✓	✓	✓		
Wang et al [21]	✓		✓	✓	✓	✓		✓	✓	
Zhang et al [26, 27]	✓	✓	✓	✓	✓	✓	✓	✓		
Cugola et al [10]	✓	✓	✓		✓	✓		✓	✓	* implicit; Support for continuous event attributes.
Wasserkrug et al [23, 24, 25]	✓	✓	✓			✓			✓	Explicit time representation

Table 1: Expressive power of CER systems. Columns: σ: selection, π: projection, $\wedge$: conjunction, $\vee$: disjunction, $\neg$: negation, ;: sequence, *: iteration, W:windowing, H: hierarchies.

Probabilistic Expressivity				
Paper	Data (occurrence and/or attributes)	Rules	Independence Assumptions	Remarks
Kawashima et al [13]	Occurrence		All events independent	
Re et al [19]	Both		1st-order Markov for SDEs (different streams independent)	
Chuanfei et al [7]	Occurrence		1st-order Markov with extensions	
Shen et al [20]	Both		SDEs independent	
Wang et al [21]	Occurrence		SDEs independent or Markovian (different streams independent).	
Zhang et al [26, 27]	Occurrence		SDEs independent	Probability distribution on time attribute
Cugola et al [10]	Both	✓	Event attributes independent. SDEs independent. CEs dependent only on events immediately below in hierarchy.	Bayesian Networks
Wasserkrug et al [23, 24, 25]	Both	✓	SDEs independent	Bayesian Networks

Table 2: Expressive power of CER systems with respect to their probabilistic properties.

Inference					
Paper	Marginal / MAP	Confidence Thresholds	Approximate	Performance	Remarks
Kawashima et al [13]	Marginal	✓		0.8-1.1 K events/s with *Kleene+*	
Re et al [19]	Marginal	✓		> 10 points increase in accuracy. 100K tuples/s for Extended Regular Queries.	
Chuanfei et al [7]	Marginal	✓		4-8K events/s for pattern lengths 6-2	
Shen et al [20]	Marginal	✓		1000K events/s, almost constant for varying window size. 1000K-100K events/s for 10-1 alternatives.	
Wang et al [21]	Marginal	✓		8K-13K events/s for 2-6 nodes	Distributed
Zhang et al [26, 27]	Marginal	✓		Reduction from exponential to close-linear cost w.r.t to selectivity / window size	
Cugola et al [10]	Marginal	✓		50% overhead	
Wasserkrug et al [23, 24, 25]	Marginal	✓	✓	CEs within desired confidence interval. Sub-linear decay of event rate w.r.t possible worlds.	

Table 3: Inference capabilities of probabilistic CER systems

may allow only uncertainty with respect to the occurrence of an event, whereas others may allow uncertainty for the event attributes as well. Finally, Table 3 presents some of the systems' properties when performing inference, such as whether they perform marginal or MAP inference, whether they give the user the option to set minimum confidence thresholds and whether they can perform approximate inference. Some comments about their performance are also included.

3.1 Cayuga

The Lahar system of Re et al [19] constitutes one of the earliest proposals. It is based on the Cayuga [11] CER engine. The design goal behind the Lahar system is to develop an efficient inference mechanism for answering queries over probabilistic SDE streams, i.e. streams whose events are tagged with a probability value. It is assumed that events follow a first-order Markov process. The possible queries are categorized in three different classes. Regular queries are composed of subgoals which do not share any variables, can readily be transformed into regular expressions with a corresponding automaton and can be evaluated in time linear to the size of the event stream. Extended regular queries allow for shared variables which must be present in all of the subgoals. Therefore, the query can be broken into independent, regular "ground" queries (by substitution) and its success probability can be computed by combining the probabilities of its constituent "ground" queries. Finally, in safe queries, variables might not be shared among all subgoals. These queries are evaluated by using a version of the Probabilistic Relational Algebra with a complexity that is quadratic to the number of timestamps in the SDE stream. Lahar was tested on object tracking in which persons and objects were equipped with RFID tags and the persons' paths and/or locations had to be assessed. Significant improvements in precision and recall were observed against deterministic approaches, with only a relatively slight overhead on throughput, which reached hundreds of thousands of events per second. A method which attempts to overcome the strict markovian hypothesis and to apply certain optimizations, such as early pruning, may be found in [7].

3.2 SASE

A simple solution for handling uncertainty with automata was proposed by Kawashima et al [13], as an extension of the SASE+ event processing engine [1]. The system builds a deterministic automaton for every user query (CE definition) and detects patterns above a certain confidence threshold by developing a matching tree as new SDEs arrive until the time window of the query expires. Branches of the tree below the given threshold are pruned early for optimization purposes. The SDEs are assumed to be independent (therefore, probability values are calculated by multiplication) and are tagged with an occurrence probability. Neither probability values for the event attributes are allowed nor for the queries themselves. Throughput values can reach several hundreds of events per second, but these numbers correspond to experiments with a single query of low complexity – a sequence operator with equality selection on the attributes and no shared variables.

Another early, NFA-based approach to incorporate uncertainty within an existing CER system is presented in [20] by Shen et al. This work uses SASE+ as its starting point and amends it in order to handle probabilistic SDEs. Each SDE is defined as a set of alternatives, each with its occurrence probability, with all alternatives summing to a probability value of 1 or less than 1 if non-occurrence is considered. The probability space is therefore defined over the possible worlds, as determined by the different (mutually exclusive) alternatives of the SDEs. The CE definitions are encoded as NFAs, but, in order to avoid enumerating all possible worlds, a special data structure, called Active Instance Graph, is used. The Active Instance Graph is a Directed Acyclic Graph connecting events with previous candidate events, i.e. whose possible occurrence may lead to the recognition of the CE. By backward-traversing the AIG, the sequence(s) that satisfy the CE definition may be retrieved and this structure also allows for dynamic filtering of events when other constraints (besides temporal sequence) are present. Finally, each event is associated with its lineage, i.e. a function which captures "where the event came from", used for computing its probability.

Inspired yet again by SASE, the work recently proposed by Wang et al [21] attempts to address two important issues. The first, related to previous NFA-based methods, concerns their inability to express CE hierarchies. The second is a performance issue and, this work is the first one which develops a CER system which is both probabilistic and distributed. The CE recognition process depends on a data structure, called Active Instance Stack, which is an optimized version of the already mentioned Active Instance Graph. Probabilities may be assigned only to events and refer to occurrences (neither probabilities for CE definitions nor for event attributes are allowed). Events are also assumed to be either independent or to follow a first-order Markov process. A data partitioning scheme is used in order to distribute different parts of the streams to different nodes and the local results are later combined to produce a global result. Finally, CE hierarchies may be constructed by having different event processing agents producing different CE types and connecting them through channels (agents are pattern matching components which can be connected to form an event processing network).

In most of the automata-based methods (with the exception of [10], presented in Section 3.3), uncertainty concerns the occurrence of the event itself as a whole, but the event attributes, including timestamps, are certain. In Zhang et al [26], the issue of imprecise timestamps is addressed, while all the other attributes have crisp values. Due to sensors' sensitivity or time granularity differences between event sources, timestamps are assumed to follow a probability distribution (usually uniform). Each event may thus have several alternative occurrence timestamps and many possible worlds, i.e. event histories, are available to the system. The temporal relations between events may differ among the possible worlds and a CE recognized in one of them may not be recognized in another. One solution is to enforce an ordering of the events from all possible worlds and then leverage an existing CER engine, such as SASE, for the CE recognition task. However, the authors present another, more efficient method, which avoids a complete enumeration of all possible worlds by employing an incremental, three-pass algorithm through the events in order to construct event matches and their intervals. This method achieves high throughput but supports only *sequence* patterns with simple equality/inequality predicates. Moreover, it was extended in [27] by Zhang et al, which added *negation* and *Kleene plus* and allowed for user-defined predicates.

3.3 CEP2U

A more recent effort extends the TESLA [8] event specification language with probabilistic modelling, in order to handle the uncertainty both in input SDEs and in the definitions of CEs [10]. The semantics of the TESLA language are formally specified by using a first order logical representation with temporal constraints that express the length of time intervals. The CE recognition algorithm however employs automata. At the input level, the method supports uncertainty regarding the occurrence of the SDEs, as well as uncertainty regarding their content. In the former case, SDEs are associated with probabilities that indicate a degree of confidence. In the latter case, the attributes of an event are modelled as random variables with some measurement error. The probability distribution function of the measurement error is assumed to be known (e.g. Gaussian distribution). Since uncertainty also derives from incomplete or erroneous assumptions about the environment in which the system operates, the method also models the uncertainty of the CE definitions. In particular, the method automatically builds a Bayesian network for each rule. The probabilistic parameters of the network are manually estimated by domain experts.

3.4 Logic-based methods

Wasserkrug et al [23,24,25] employ the technique of knowledge based model construction (KBMC), whereby knowledge representation is separated from the inference process. Inference is preformed on a Bayesian network as needed (when new SDEs arrive), without constructing the whole network beforehand. Each event is assigned a probability, denoting how probable it is that the event occurred with specific values for its attributes. Uncertainty about the value of a single event attribute may be represented by multiple event instances with different probabilities and with the same values for all other attributes.

In turn, CE definitions are encoded in a two-fold way, with a selection operation (mostly based on event type) performing an initial filtering, followed by a pattern-detection schema for more complex operations, based on temporal relations and attribute equalities. The selection mechanism imposes certain independence properties on the Bayesian network. Inferred CEs are conditioned only on selectable lower-level events, preventing the network from being cluttered with many dependency edges. This framework is not limited to representing only propositional or first order knowledge. It could potentially handle higher-order knowledge, since this pattern-matching step could, in principle, be defined in any kind of language. However, the system presented in the evaluation experiments allows only predicates expressing temporal constraints on event timestamps or equality relations on event attributes.

Calculation of the probabilities for the inferred CEs is done by dynamically constructing a Bayesian network upon every new event arrival. The nodes of the network correspond to SDEs and CEs. First, SDEs are added. Nodes for CEs are inserted only when a rule defining the CE is satisfied, having as parents the events that triggered the rule, which might be SDEs or even other CEs, in case of hierarchical CE definitions. The probability and attribute values of the inferred CEs are determined by mapping expressions associated with the corresponding rule. In order to avoid the cost of exact inference, a form of sampling is followed, which allows for bypassing the construction of the network by sampling directly according to the rules for CE definitions.

3.5 Comments

In Table 1 we list the operators supported by each method. Table 2 presents their probabilistic properties: their independence assumptions and the support for data and/or rules uncertainty. Their properties with respect to inference are shown in Table 3 (marginal/MAP inference, support for confidence thresholds, approximate inference).

As shown in Table 2, all of the presented approaches have the ability to represent probabilistic SDEs, where uncertainty may refer to their occurrence or/and the content of their attributes. However, a feature which is lacking in most of the methods is the capacity to assign probabilities to rules expressing CE definitions. In this case, probabilistic graphical models, with their ability to represent all events as nodes in a homogeneous manner and encode the direction of causation, can prove useful. The two methods which allow rule probabilities, use such a model, namely Bayesian Networks.

The KBMC method of [23, 24, 25] and the CEP2U system of [10] allow for both hierarchies and probabilistic rules (see Table 2). Both of them use Bayesian Networks for inference, with the nodes of the network representing events, SDEs and CEs. CEP2U was designed from the very beginning with the goal of minimizing the performance overhead incurred by the introduction of uncertainty. Indeed, the maximum overhead mentioned in the experiments was almost always less than 50%, compared to the deterministic case. On the other hand, the KBMC technique is still far from achieving event rates comparable (say, within an order of magnitude) to those of purely deterministic models. This performance robustness of CEP2U against uncertainty comes at a price though, since some simplifying assumptions have to be made. CEP2U constructs only a single Bayesian Network for each rule (not for each grounding) and a simple solution is proposed for the problem of propagating probabilities from lower to higher level CEs. Occurrence probabilities of intermediate events are propagated to higher level events with a value of 1, essentially decomposing the total probability space into smaller and more manageable spaces. This means that these Bayesian Networks function more like look-up tables, hence the much lower cost of inference. The effects of this simplification on accuracy, however, are unclear.

A related issue is that of the independence assumptions made by each method. Automata-based methods tend to make a substantial number of simplifying assumptions about the independence of events or streams, resulting in simpler probabilistic models The most complex dependency models employed make the assumption that events may follow a first-order Markov process, as in [19, 21] (a slightly more complex model may be found in [7]). In domains characterized mostly by sequential patterns upon homogeneous streams, this assumption may be sufficient. When multiple streams with different event types are involved and hierarchies of CEs are required, which take into account lower-level CEs across a time window, more complex dependencies need to be encoded.

Bayesian Networks offer such a flexibility but they suffer from problems of high inference complexity. In order to keep the inference cost low, certain simplifications are introduced again. For example, CEP2U assumes that an inferred CE is the only cause for all of its sub-events (note that, in CEP2U, the direction of causation is from the higher level to the lower level events), i.e. one sub-event cannot be used to define other CEs and it is not possible to have multiple definitions for a CE. Although this obviously helps in making the Bayesian Networks (which can be manually edited by the user) the assumption of such a strict separation of rule conditions limits the expressive power of the system (and would presumably require tedious tuning to correct it).

4. CONCLUSIONS

Our short review of probabilistic CER systems identified the following limitations: In terms of language expressivity, the basic drawback of most systems is the absence of support for constructing hierarchies of CEs. Moreover, most systems do not support uncertainty in the rules defining CEs. Those that do support rule uncertainty either make too strong simplifying assumptions, thus possibly limiting accuracy in domains with complex dependencies, or face serious issues of under-performance, even when approximate inference is employed. Distributed processing of probabilistic SDE streams is still at its early stages, with only one method employing it. Notice also that none of the systems supports MAP inference, a feature which is useful in certain domains (e.g. in video recognition, where it is sometimes desirable to retrieve those time intervals during which it is most likely for an activity to have occurred). Those issues should act as indicators for possible directions of future work.

5. ACKNOWLEDGMENTS

This work has been funded by the EU SPEEDD project (FP7-ICT 619435).

6. REFERENCES

[1] J. Agrawal, Y. Diao, D. Gyllstrom, and N. Immerman. Efficient pattern matching over event streams. In *SIGMOD Conference*, pages 147–160, 2008.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[3] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, July 1984.

[4] A. Artikis, O. Etzion, Z. Feldman, and F. Fournier. Event processing under uncertainty. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 32–43, New York, NY, USA, 2012. ACM.

[5] A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek, and others. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *EDBT*, pages 712–723, 2014.

[6] I. Cervesato and A. Montanari. A calculus of macro-events: progress report. In *Seventh International Workshop on Temporal Representation and Reasoning, 2000. TIME 2000. Proceedings*, pages 47–58, 2000.

[7] X. Chuanfei, L. Shukuan, W. Lei, and Q. Jianzhong. Complex event detection in probabilistic stream. In *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pages 361–363, Apr. 2010.

[8] G. Cugola and A. Margara. TESLA: a formally defined event specification language. In *Proceedings of Conference on Distributed-Event Based Systems (DEBS)*, pages 50–61, 2010.

[9] G. Cugola and A. Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys*, 44(3), 2011.

[10] G. Cugola, A. Margara, M. Matteucci, and G. Tamburrelli. Introducing uncertainty in complex event processing: model, implementation, and validation. *Computing*, pages 1–42, 2014.

[11] A. Demers, J. Gehrke, M. Hong, M. Riedewald, and W. White. Towards expressive publish/subscribe systems. In Y. Ioannidis, M. H. Scholl, J. W. Schmidt, F. Matthes, M. Hatzopoulos, K. Boehm, A. Kemper, T. Grust, and C. Boehm, editors, *Advances in Database Technology - EDBT 2006*, number 3896 in Lecture Notes in Computer Science, pages 627–644. Springer Berlin Heidelberg, Jan. 2006.

[12] T. Grabs and M. Lu. Measuring performance of complex event processing systems. In *Topics in Performance Evaluation, Measurement and Characterization*, pages 83–96. Springer, 2012.

[13] H. Kawashima, H. Kitagawa, and X. Li. Complex event processing over uncertain data streams. In *2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 521–526, Nov. 2010.

[14] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.

[15] D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[16] M. R. Mendes, P. Bizarro, and P. Marques. A performance study of event processing systems. In *Performance Evaluation and Benchmarking*, pages 221–236. Springer, 2009.

[17] M. R. Mendes, P. Bizarro, and P. Marques. Towards a standard event processing benchmark. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, ICPE '13, pages 307–310, New York, NY, USA, 2013. ACM.

[18] A. Paschke and M. Bichler. Knowledge representation concepts for automated SLA management. *Decision Support Systems*, 46(1):187–205, Dec. 2008.

[19] C. Re, J. Letchner, M. Balazinksa, and D. Suciu. Event queries on correlated probabilistic streams. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 715–728, New York, NY, USA, 2008. ACM.

[20] Z. Shen, H. Kawashima, and H. Kitagawa. Probabilistic event stream processing with lineage. In *Proc. of Data Engineering Workshop*, 2008.

[21] Y. H. Wang, K. Cao, and X. M. Zhang. Complex event processing over distributed probabilistic event streams. *Computers & Mathematics with Applications*, 66(10):1808–1821, Dec. 2013.

[22] S. Wasserkrug, A. Gal, and O. Etzion. A taxonomy and representation of sources of uncertainty in active systems. In O. Etzion, T. Kuflik, and A. Motro, editors, *Next Generation Information Technologies and Systems*, number 4032 in Lecture Notes in Computer Science, pages 174–185. Springer Berlin Heidelberg, Jan. 2006.

[23] S. Wasserkrug, A. Gal, and O. Etzion. A model for reasoning with uncertain rules in event composition systems. *arXiv:1207.1427 [cs]*, July 2012.

[24] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Complex event processing over uncertain data. In *Proceedings of the second international conference on Distributed event-based systems*, pages 253–264. ACM, 2008.

[25] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Efficient processing of uncertain events in rule-based systems. *IEEE Transactions on Knowledge and Data Engineering*, 24(1):45–58, Jan. 2012.

[26] H. Zhang, Y. Diao, and N. Immerman. Recognizing patterns in streams with imprecise timestamps. *Proc. VLDB Endow.*, 3(1-2):244–255, Sept. 2010.

[27] H. Zhang, Y. Diao, and N. Immerman. On complexity and optimization of expensive queries in complex event processing. pages 217–228. ACM Press, 2014.