# Dagstuhl Seminar on the Foundations of Composite Event Recognition

Alexander Artikis[1,2] Thomas Eiter[3] Alessandro Margara[4]
Stijn Vansummeren[5]
[1]University of Piraeus, GR, [2]NCSR Demokritos, GR, [3]TU Wien, AT
[4]Polytechnic University of Milan, IT, [5]Hasselt University and transnational University of Limburg, BE
a.artikis@unipi.gr, eiter@kr.tuwien.ac.at, alessandro.margara@polimi.it
stijn.vansummeren@uhasselt.be

## ABSTRACT

Composite event recognition (CER) is concerned with continuously matching patterns in streams of 'event' data over (geographically) distributed sources. This paper reports the results of the Dagstuhl Seminar "Foundations of Composite Event Recognition" held in 2020.

## 1. INTRODUCTION

Composite Event Recognition (CER) refers to the activity of matching patterns in streams of continuously arriving 'event' data over (geographically) distributed sources. CER is a key ingredient of many modern Big Data applications that require the processing of such event streams to obtain timely insights and implement reactive and proactive measures. Traffic management in smart cities, for instance, requires the analysis of data from an increasing number of sensors, both mobile (e.g. mounted on public transport vehicles and private cars) and stationary (installed at intersections). Using such data streams, CER systems detect or even forecast traffic congestions, thus enabling one to proactively change traffic light policies and speed limits, with the aim of reducing carbon emissions, optimising public transportation, and improving the quality of life and productivity of commuters [3].

Numerous CER systems and languages have been proposed in the literature, cf. [1, 7, 9]. While these systems have a common goal, they differ in their architectures, data models, pattern languages and processing mechanisms, resulting in heterogeneous implementations with sometimes fundamentally different capabilities. Because the research focus in the established literature has been on practical system aspects, and less on formal foundations, CER can be difficult to understand, extend and generalise. As a concrete example, so-called *selection strategies* [7, 9] are supported by numerous systems, but with sometimes incompatible implementations. In this respect, it helps to model the semantics formally, so that these differences and the trade-offs they entail become clear, demonstrating the benefits of formal models compared to others [10].

To start addressing these issues, a seminar on the Foundations of Composite Event Recognition was held at Schloss Dagstuhl, Leibniz Center for Informatics during February 9-14, 2020.[1] The seminar gathered 39 researchers and practitioners working in diverse domains strictly related to CER. The first days put a focus on tutorials and talks that gave an overview of the approaches, techniques, methodologies and vocabularies used in different communities to refer to CER problems. Subsequently, the seminar continued by alternating sessions with focused research talks and working group discussions, on the topics that the participants identified as the most relevant for future investigations and research efforts. This paper gives an overview of the tutorials and outcomes of the discussions. Due to space limitations, the exposition is necessarily brief; more information is available in the Dagstuhl report [4].

## 2. TUTORIALS

Six tutorials aimed at introducing CER-related research in different communities.

**Applications & Requirements of CER.** Sabri Skhiri presented key requirements of CER systems from an applications perspective, focusing on four questions: (1) Which industrial applications does CER have? (2) What are the key requirements of CER concerning data models, recognition language expressiveness and performance (latency, throughput, predictive accuracy)? (3) How do existing approaches address the requirements? (4) Which classes of applications can benefit from CER techniques? To answer these questions, a typical

[1]https://www.dagstuhl.de/en/program/calendar/semhp/?semnr=20071

streaming architecture where CER systems may be deployed was presented. The CER challenges were then illustrated on industrial applications in crowd management, banking, telecommunication, security & surveillance, and microservice architectures. The key requirements for them are as follows. CER systems must scale to streams of millions of events per second while handling states or partial matches lasting from a few days to weeks. Next, the CER language should support temporal iteration, negation and sub-patterns. Moreover, imprecise patterns should be supported, as experts cannot always define target composite events precisely.

**CER in Data Management.** Martin Ugarte and Cristian Riveros presented a theoretical perspective of the most common features of CER languages. They started with a basic setting for CER that served to discuss the fundamental properties from a Data Management perspective: well-defined syntax and semantics, composability, and denotational, declarative semantics. These properties were then exemplified on *complex event logic* [10], which was also used to present the main operations in CER systems. Next, they discussed the challenges of formally defining the CER operations while satisfying the aforementioned properties. Moving on to evaluation of CER languages, they discussed the relevant notions of efficiency and complexity, and presented the types of lower bounds obtainable for evaluating CER patterns. Finally, they outlined CER challenges on concrete examples: What are the relevant complexity classes? What are the classes of queries that may be evaluated efficiently? How does the change from push-based to pull-based semantics affect complexity?

**Distributed Event-Based Systems.** Avigdor Gal and Ruben Mayer introduced Distributed Event-Based Systems (DEBS), which may be viewed as pipelines from sources of low-level events to sinks with applications, via an operator graph (event processing middleware). They presented event recognition languages and pointed out the differences between composite event processing, stream processing and rule-based event processing. Then they considered windowing in depth, parallelism and time aspects, such as event and processing time, late arrival of events, and the trade-off between latency and accuracy. Finally, they addressed uncertainty associated to event occurrence and attribute values and discussed approaches to deal with uncertainty in the matching process.

**Stream Reasoning.** Stream reasoning deals with incremental reasoning over rapidly changing information [8]. Jacopo Urbani and Fredrik Heintz gave an overview of the area, first presenting some application scenarios. The key ingredients for stream reasoning are temporal data (time management), various sources (data integration), intelligent decision making (AI), and scalability and efficiency (data management). Urbani and Heintz presented requirements for query answering over streams and matched them against the four V's in Big Data (volume, velocity, variety and veracity). A variety of approaches were then briefly discussed, viz. CSPARQL, CQELS, EP-SPARQL, LARS, Laser, Ticker, BigSR and metric temporal logic (MTL) based reasoning. Open challenges mentioned are handling massive data and uncertainty, combining symbolic and sub-symbolic knowledge, and benchmarking stream reasoning systems.

**CER in Logic and AI.** Diego Calvanese presented formalisms relevant for CER that have been developed in the areas of knowledge representation and reasoning, formal verification and database theory. Such formalisms typically rely on combining variants of temporal logics with logics used in knowledge representation and reasoning, which poses challenges for semantics and computability. The challenges have to a certain extent been addressed by a variety of techniques and under various assumptions; however, the area is fragmented and there is no unifying or consolidated framework.

**CER in Business Process Management.** In many application scenarios, business processes may be viewed as consumers as well as producers of events. Common process modelling languages, therefore, contain constructs to incorporate events. At the same time, event-based systems can be used as a basis for process execution and analysis. Against this background, Matthias Weidlich reviewed the relation between the fields of business process management and CER. Furthermore, he outlined opportunities for research at the intersection of the two fields, as regards CER integration with process modelling, event abstraction for process analysis, event pattern derivation from process models, and event-based process execution infrastructures.

## 3. WORKING GROUPS

The working groups were devoted to five topics.

**Expressiveness & Common Model.** As CER systems and languages have originated from many different communities, the CER field is broad and diverse, and this in turn makes understanding the relationships between various approaches difficult. The discussions in this working group aimed to clarify whether a "core" of existing systems and lan-

guages can be captured in a common formal model. Such a model could ease the comparison of the expressiveness and capabilities of different approaches as well as improve system interoperability.

The first session revealed sometimes widely different views about many essential CER aspects. For example: what kind of problem is CER, abstractly speaking? Is it a model checking, a monitoring, or a synthesis problem? What kind of object do CER systems produce as output? Is it a sequence of time-annotated facts from the input, a sequence of sets of such facts, or an arbitrary sequence of tuples? Is a notion of time essential in CER? Arguments in favor and against all proposed views were discussed; we refer the interested reader to [4].

As finding a single common model seemed to be difficult, the participants considered then establishing an abstract *meta-model* for CER. Ideally, such a meta-model incorporates key elements of CER systems and focuses on *what* CER does rather than *how* this is done. By introducing conceptual components that can be instantiated in different ways, the meta-model could allow a *common way of thinking* about CER, thus facilitating the discussions in the community. A first abstract candidate meta-model was proposed during the seminar [4]. A natural next step is to investigate how it can be instantiated to capture the existing CER literature, and to identify commonalities among these instantiations.

**Uncertainty in CER.** CER systems must deal with various types of uncertainty [1]. The events of the input stream may be noisy, e.g. due to inaccuracy of sensors or distortion in a communication channel. Moreover, a sensor may fail to report certain events, due to e.g. hardware malfunction. Even if the hardware infrastructure works as expected, the characteristics of the environment could prevent events from being recorded; consider, for instance, an occluded object in video monitoring. Data uncertainty may also be by intention, e.g. by an event publisher to prevent complete disclosure of an event stream to its subscribers.

Besides uncertainty in the input data, event patterns can be imprecise or incomplete, as identified in the 'Applications & Requirements' tutorial; due to lack of knowledge or inherent complexity of a domain, it is sometimes impossible to capture exactly all the conditions that a pattern should satisfy.

The participants outlined the following three open challenges. (1) Identify possible sources of uncertainty and classify them according to the impact they can have on the recognition process. (2) Define suitable models to represent different types of uncertainty. (3) Define a conceptual framework and algorithms to consider and propagate uncertainty in the composite event recognition process.

**Benchmarking.** While use-cases, key performance indicators and relevant benchmarking challenges have been identified [2, 6, 11], CER performance evaluation is still not homogeneous. In the absence of sufficient real-world event streams, researchers adapted analytic benchmarks like Linear Road [2], or benchmarks for Message-Oriented Middleware. Such hand-crafted approaches limit experiment reproducibility. Moreover, maintaining these benchmarks is a burden for individual research groups with long-term support hard to guarantee.

The working group focused on identifying a sustainable path to the design of a domain-specific benchmark for CER maintainable by the community. First, interesting types of benchmarks were discussed. As major ones, macro- (aka use-case driven) benchmarks and micro-benchmarks emerged, which focus on evaluating systems w.r.t. specific workloads, typically inspired by real-world scenarios, resp. the performance of single operators. Macro-benchmarks directly relate with the ongoing effort behind the DEBS Grand Challenges, which yearly provide interesting use-cases and workloads. Micro-benchmarks relate to a common CER model via a core algebra of operations that CER engines must support. Second, the discussion highlighted the lack of standard data and query models (and formats) for CER, which are crucial to develop and maintain a benchmark suite for the community. (Streaming extensions of SQL are towards the right direction [5].) Finally, the discussion focused on technical supports, emphasizing the need for a FAIR (i.e., *findable*, *accessible*, *interoperable* and *reusable*) community benchmark.

Towards establishing a CER benchmark, a two-step approach was proposed: (1) Provide a systematic review of existing benchmarks and systems experiments to identify their dimensions of interest, and a repeatability study that tries to replicate existing results. The insights gained provide input for a new benchmark. (2) Form a working group to create the CER benchmark.

**Process strategies, parallelization, and distribution.** CER applies to heterogeneous scenarios, with different requirements and deployment settings. A CER framework should adapt its processing and deployment strategies to optimise the use of resources for achieving the application goals. The participants identified here several research and engineering challenges: (1) Identify suitable metrics and constraints to express application requirements in terms of performance (e.g. latency and through-

put) as well as use of resources, security, privacy and fault tolerance. (2) Identify suitable models to capture the relevant characteristics of the deployment infrastructure, e.g. in terms of computation power, hardware architecture, memory, network connections, geographical locations and ownership. (3) Understand the trade-offs that exist between expressiveness and optimisation opportunities, e.g. to identify functionalities that limit the applicability of parallel processing. This could lead to the design of various language fragments that offer the best balance between generality, expressiveness, and performance in a given scenario. (4) Define flexible process and deployment mechanisms to allocate operators to physical nodes, and adopt the most suitable processing algorithms and communication techniques for a given deployment infrastructure. (5) Define monitoring mechanisms to promptly detect critical situations, such as failures and node overloads. Design and implement adaptation algorithms to change the deployment at runtime and restore from such critical situations.

**Event pattern induction and composite event forecasting.** Manual event pattern authoring is error-prone and time-consuming, as is manual fine-tuning of event patterns to optimise their predictive performance, which should be done whenever it deteriorates, e.g. when the statistical properties of a stream are modified. As machine learning techniques support event pattern construction and refinement, they start attracting attention by the CER community.

Composite Event Forecasting (CEF) refers to the ability of a system to provide forecasts about the possible occurrence of composite events in the future [3]. Notably, CEF is less mature than and *orthogonal* to pattern induction, as the underlying patterns for CEF may be manually constructed or automatically extracted from data.

Pattern induction and CEF have several challenges. (1) Enhance machine learning techniques with domain knowledge curated by experts, to reduce the search space and produce patterns with higher predictive accuracy. (2) Provide automatically constructable models that humans can understand, thus supporting explainability, and expressive enough to effectively capture the temporal phenomena of an application. (3) Provide *online* learning algorithms for CER systems that can construct an event pattern set in a single-pass over the input stream, while efficiently dealing with stream changes; to achieve decent performance, distributed learning may be necessary. (4) As for CEF, identify ways for online accurate forecasting of composite events that may take place (far) in the future, e.g. by combinations of probabilistic reasoning and (extended symbolic) automata. (5) Identify ways to effectively inform proactive decision-making as a result of CEF. For instance, if a traffic congestion is forecast for an intersection, re-direct traffic trying to avoid traffic congestion in other intersections.

# 4. CONCLUSION

Complex Event Recognition (CER) is an area of growing interest that draws from diverse communities. The Dagstuhl seminar served to share their views and identify the relevant topics with future research challenges on the foundations of CER; establishing a common view and (meta-)model of CER is the biggest among them. First steps have been made, but more efforts are necessary. A workshop on reasoning about actions and events over streams (RACES) at KR 2020 and a planned workshop on CER benchmarking are on the agenda.

# 5. REFERENCES


[1] E. Alevizos, A. Skarlatidis, A. Artikis, and G. Paliouras. Probabilistic complex event recognition: A survey. *ACM Comp. Surv.*, 50(5):71:1–71:31, 2017.

[2] A. Arasu, M. Cherniack, E. F. Galvez, D. Maier, A. Maskey, E. Ryvkina, M. Stonebraker, and R. Tibbetts. Linear road: A stream data management benchmark. In *VLDB*, pp. 480–491, 2004.

[3] A. Artikis, C. Baber, P. Bizarro, C. Canudas-de-Wit, O. Etzion, F. Fournier, P. Goulart, A. Howes, J. Lygeros, G. Paliouras, A. Schuster, and I. Sharfman. Scalable proactive event-driven decision making. *IEEE Technol. Soc. Mag.*, 33(3):35–41, 2014.

[4] A. Artikis, T. Eiter, A. Margara, and S. Vansummeren, editors. *Foundations of Composite Event Recognition: Report from Dagstuhl Seminar 20071*. Dagstuhl Reports. 2020.

[5] E. Begoli, T. Akidau, F. Hueske, J. Hyde, K. Knight, and K. Knowles. One SQL to rule them all - an efficient and syntactically idiomatic approach to management of streams and tables. In *SIGMOD*, pp. 1757–1772. ACM, 2019.

[6] P. Bizarro. Bicep - benchmarking complex event processing systems. In *Event Processing*, *Dagstuhl Seminar Proc.* 07191. IBFI, Schloss Dagstuhl, 2007.

[7] G. Cugola and A. Margara. Processing flows of information: From data stream to complex event processing. *ACM Comp. Surv.*, 44(3):15:1–15:62, 2012.

[8] D. Dell'Aglio, E. D. Valle, F. van Harmelen, and A. Bernstein. Stream reasoning: A survey and outlook. *Data Sci.*, 1(1-2):59–83, 2017.

[9] N. Giatrakos, E. Alevizos, A. Artikis, A. Deligiannakis, and M. Garofalakis. Complex event recognition in the big data era: a survey. *VLDB J.*, 29(1):313–352, 2020.

[10] A. Grez, C. Riveros, and M. Ugarte. A formal framework for complex event processing. In P. Barceló and M. Calautti, editors, *ICDT*, *LIPIcs* vol. 127, pp. 5:1–5:18, 2019.

[11] T. Scharrenbach, J. Urbani, A. Margara, E. D. Valle, and A. Bernstein. Seven commandments for benchmarking semantic flow processing systems. In *ESWC*, *LNCS* 7882, pp. 305–319. Springer, 2013.