Report from Dagstuhl Seminar 16341

Integrating Process-Oriented and Event-Based Systems

Edited by

David Eyers[1], Avigdor Gal[2], Hans-Arno Jacobsen[3], and Matthias Weidlich[4]

1 **University of Otago, NZ,** dme@cs.otago.ac.nz
2 **Technion – Haifa, IL,** avigal@ie.technion.ac.il
3 **TU München, DE,** arno.jacobsen@tum.de
4 **HU Berlin, DE,** matthias.weidlich@informatik.hu-berlin.de

Abstract

This report documents the programme and the outcomes of Dagstuhl Seminar 16341 on "Integrating Process-Oriented and Event-Based Systems", which took place August 21–26, 2016, at Schloss Dagstuhl – Leibniz Center for Informatics. The seminar brought together researchers and practitioners from the communities that have been established for research on process-oriented information systems on the one hand, and event-based systems on the other hand. By exploring the use of processes in event handling (from the distribution of event processing to the assessment of event data quality), the use of events in processes (from rich event semantics in processes to support for flexible BPM), and the role of events in process choreographies, the seminar identified the diverse connections between the scientific fields. This report summarises the outcomes of the seminar by reviewing the state-of-the-art and outlining research challenges on the intersection of process-oriented and event-based systems.

Seminar August 21–26, 2016 – http://www.dagstuhl.de/16341
1998 ACM Subject Classification C.2.4 Distributed Systems, H. Information Systems
Keywords and phrases distributed systems, event-based systems, process-aware information systems
Digital Object Identifier 10.4230/DagRep.6.8.21
Edited in cooperation with Jan Sürmeli

1 Executive Summary

David Eyers
Avigdor Gal
Hans-Arno Jacobsen
Matthias Weidlich

License Creative Commons BY 3.0 Unported license
© David Eyers, Avigdor Gal, Hans-Arno Jacobsen, and Matthias Weidlich

Background and Motivation

Process-oriented information systems are software systems that execute and manage a process, broadly defined as a coordinated execution of actions to achieve a certain goal. As such, they support Business Process Management (BPM) initiatives. Process-oriented systems have been traditionally used in domains such as business process automation, enterprise application integration, and collaborative work. Recently, there has also been a significant uptake of process-oriented information systems in transportation, logistics, and medical

Except where otherwise noted, content of this report is licensed
under a Creative Commons BY 3.0 Unported license

infrastructures – domains that impose new challenges in terms of system reactivity and adaptability. Here, trends such as sensing of data (e.g., based on RFID technology) and advancing system integration (driven by technical standards such as EPCglobal) represent opportunities to strengthen the event-perspective in process-oriented systems in order to achieve more flexible and comprehensive process control.

Event-based systems, in turn, have been put forward to integrate heterogeneous systems in a flexible and scalable manner by separating communication from application logic. These systems provide interaction models, mechanisms for routing events between components, and techniques for the detection of composite events, i.e., for Complex Event Processing (CEP). Although event-based systems are typically positioned as general-purpose technology, they have found their way into many applications where event generation is comparatively deterministic and follows structured behaviour. In domains such as transportation, logistics, and the medical sector, events handled by event-based systems stem from the execution of processes, which are partially supported by process-oriented information systems. Exploiting the process-perspective, therefore, promises to lead to advancements in the design, analysis, and optimisation of event-based systems.

The increasing overlap of application scenarios that involve concepts and techniques of process-oriented as well as event-based systems, however, is only marginally supported by exchange and convergence of the related research fields. Strong communities have been established for research on either type of system. Yet, due to the missing link between these communities, manifold opportunities for ground-breaking research and broad impact in industry are missed out. Research efforts related to the underlying theory as well as specific platforms are duplicated and similar approaches are developed in both communities.

Breaking this disconnect had been the goal that the seminar aimed to achieve by identifying the links between conceptual models, formal analysis methods, and engineering techniques developed for either type of system.

Seminar Structure

Given that seminar attendees came from two rather disconnected communities, the first day of the seminar featured four tutorials to establish a joint understanding of essential concepts and terminology. First, Alessandro Margara presented an overview of the basic techniques to manage streams of events. Mathias Weske then gave a primer on BPM, elaborating on the main concepts, models, and the role of events for process management. An advanced view on techniques for event processing was given by Alejandro Buchmann. Stefanie Rinderle-Ma closed this part of the seminar with a tutorial on management, utilisation, and analysis of instance data in distributed process management.

The remainder of the seminar week was centred on break-out sessions, in which participants worked on particular topics on the intersection of process-oriented and event-based systems. In these working groups, participants discussed the relevant state-of-the-art and identified the research challenges under a near-, mid-, or long-term perspective. In addition, there were two sessions in which seminar participants gave a very short overview of their recent research work.

Topics and Key Challenges

The working groups focussed on a diverse set of topics, highlighting the key challenges that need to be addressed:

Event Models for BPM: Semantics of Events and Patterns. Starting from the observation that event models are well-established in both BPM and CEP and that their coupling has obvious benefits, the challenge relates to the question of how events can guide the evolution or adaptation of process instances.

Towards Automatic Event-Based Monitoring of Processes. Event-based monitoring of processes is influenced by the availability of patterns, the consequences of monitoring results, and the integration of contextual information. These dimensions render it particularly challenging to comprehensively discover and utilise patterns for process monitoring.

Patterns and Models for Communication. The communication models underlying an event-based middleware have diverse implications for the interplay of processes and event patterns – and a major challenge is the identification of requirements that are imposed by process scenarios on communication models.

Choreographies and Inter-Process Correlation. Common languages for the description of interacting processes lack capabilities for the specification of event-based processing. The challenge is to develop a better grounding of choreography languages and enable analysis of the information flow between processes.

Abstraction Levels: Processes versus Events. Observing that methods in BPM mainly proceed top-down, whereas event processing is often approached bottom-up, a key challenge is the identification of the right abstraction level on which concepts and methods shall be integrated.

Context in Events and Processes. The context of a process may influence event processing, and the context as materialised in complex events impacts the execution of a process. Yet, a suitable representation and dynamic evolution of context information is an open research challenge.

Integrated Platforms for BPM & CEP. The integration of traditional BPM or CEP engines promises accelerated application development and lower maintenance cost. To attain this end, the challenge of developing a unified model for events and processes, enabling well-grounded architectural decisions, needs to be addressed.

(Highly) Distributed Processes & The Role of Events. Events and processes can both be handled in a centralised or distributed infrastructure and open challenges relate to the tradeoffs regarding trustworthiness, reliability, and scalability.

Event Data Quality. Event data may be uncertain, which needs to be reflected in processes that are influenced by these events. The challenge is how to capture such uncertainty and make explicit how it influences decision making on the level of the process.

From Event Streams to Process Models and Back. Event patterns and processes are typically concerned with events on different levels of abstractions, which can be bridged only on the basis of a unifying formal model. Further challenges arise from the imprecision of event definitions in processes and the expressiveness of CEP languages when capturing procedural behaviour.

Compliance, Audit, Privacy and Security. Compliance checking of business processes may benefit from CEP systems and BPM tools may be useful to express service level agreements in event-based systems. Challenges, however, are methods for a structured integration of BPM and CEP technology and their alignment with informal compliance requirements.

Main Recommendations

From the discussions and the exchange of ideas during the workshop, a set of recommendations was able to be distilled in order to materialise the benefits of integrating process-oriented and event-based systems.

Build a community around BPM and CEP. The topics on the intersection of process-oriented and event-based systems provide a rich field for high-impact research. The number and diversity of open research questions call for a long-term research initiative, so that a respective community needs to be built up. To achieve this, it is recommended that joint workshops be initiated at the flagship conferences in either field, the BPM conference and the DEBS conference, and to evaluate potential co-location of the conferences in future.

Start research on integrated models. For many of the aforementioned challenges, the lack of integrated models, in which processes and events are first-class citizens, turns out to be a major issue. Research shall be devoted to creating such models, clarifying which basic notions of events exist, and considering the semantics of distributed event generation.

Facilitate joint research. Joint research is currently hindered not only by the disconnect of the research communities, but also by a lack of a common set of standard concepts in either community. There is a need for concise overviews of the most important concepts and methods in either field, e.g., by means of standard textbooks. Researchers from one field need to be able to quickly gather the level of understanding of the other field that is required for joint research initiatives.

Engage industry. The integration of process-oriented and event-based systems is driven by particular domains, such as logistics, health, and mobility. The prioritisation of challenges and the evaluation of developed solutions critically depends on the involvement of industrial partners from these domains. As such, it is recommended to reach out to industry to develop evaluation scenarios and benchmark datasets. One viable means for this are the research proposals on the EU and national levels that involve BPM and CEP experts from both academia and industry.

2 Table of Contents

Executive Summary
David Eyers, Avigdor Gal, Hans-Arno Jacobsen, and Matthias Weidlich 21

Working groups

Event Models for BPM: Semantics of Events and Patterns and Formal Methods for Reasoning on Events in the BPM Context
Anne Baumgraß, Alexander Artikis, Annika M. Hinze, Ken Moody, Wolfgang Reisig, Stefanie Rinderle-Ma, Stijn Vansummeren, and Matthias Weidlich 28

Compliance, Audit, Privacy and Security
David Eyers, Jean Bacon, Martin Jergler, Ken Moody, Stefanie Rinderle-Ma, and Stijn Vansummeren 30

Towards Automatic Event-Based Monitoring of Processes
Annika M. Hinze, Alexander Artikis, Anne Baumgraß, Alejandro P. Buchmann, Claudio Di Ciccio, Hans-Arno Jacobsen, Boris Koldehofe, Alessandro Margara, Pnina Soffer, and Holger Ziekow 31

Patterns and Models for Communication
Boris Koldehofe, Oliver Kopp, Wolfgang Reisig, Martin Ugarte, and Roman Vitenberg 34

Choreographies and Inter-Process Correlation
Oliver Kopp, Wolfgang Reisig, Jatinder Singh, Sergey Smirnov, Jan Sürmeli, Roman Vitenberg, Matthias Weidlich, and Kaiwen Zhang 35

Abstraction Levels: Processes versus Events
Sankalita Mandal, David Eyers, Agnes Koschmider, Ken Moody, Cesare Pautasso, Mohammad Sadoghi Hamedani, Wei Song, Lucinéia Heloisa Thom, and Lijie Wen 38

Context in Events and Processes
Alessandro Margara, Alejandro P. Buchmann, Sankalita Mandal, Cesare Pautasso, Arik Senderovich, Sergey Smirnov, Matthias Weidlich, and Mathias Weske 39

Integrated Platforms for BPM & CEP
Mohammad Sadoghi Hamedani, Alejandro P. Buchmann, Hans-Arno Jacobsen, Martin Jergler, Sankalita Mandal, Cesare Pautasso, Stefan Schulte, Jatinder Singh, Sergey Smirnov, and Mathias Weske 41

(Highly) Distributed Processes & The Role of Events
Stefan Schulte, Jean Bacon, Avigdor Gal, Martin Jergler, Stefanie Rinderle-Ma, Arik Senderovich, Vinay Setty, Martin Ugarte, and Stijn Vansummeren 42

Optimisation opportunities
Roman Vitenberg, Avigdor Gal, Alessandro Margara, Vinay Setty, Martin Ugarte, Matthias Weidlich, Lijie Wen, and Kaiwen Zhang 45

Event Data Quality
Kaiwen Zhang, Alexander Artikis, Anne Baumgraß, Avigdor Gal, Mohammad Sadoghi Hamedani, and Stefan Schulte 49

From Event Streams to Process Models and Back
Holger Ziekow, Jean Bacon, Claudio Di Ciccio, David Eyers, Boris Koldehofe, Oliver Kopp, Agnes Koschmider, Pnina Soffer, Wei Song, and Jan Sürmeli 50

Tutorials

Managing Streams of Events: An Overview
Alessandro Margara 52

Business Process Management: Concepts, Models, Events
Mathias Weske 52

Management, Utilisation, and Analysis of Instance Data in Distributed Process Settings
Stefanie Rinderle-Ma 53

Event Processing
Alejandro P. Buchmann 53

Overview of Talks

Online Learning for Complex Event Recognition
Alexander Artikis 53

Smart Logistics in Practice – Using Event Processing for Comprehensive Transportation Monitoring
Anne Baumgraß 54

Predictive Task Monitoring: Processing Flight Events to Foresee Diversions
Claudio Di Ciccio 54

Smart Landscape: The Rugged Internet of Things
Annika M. Hinze 55

BPM in Cloud Architectures: Enabling the Internet of Things Through Effective Business Processes Management with Events
Hans-Arno Jacobsen 56

D2Worm – A Management Infrastructure for Distributed Data-centric Workflows
Martin Jergler 56

Explicit Subscription for Enabling Event Buffering
Sankalita Mandal and Jan Sürmeli 57

Associative Composition of Stream Processing Processes
Wolfgang Reisig 57

RichNote: Adaptive Selection and Delivery of Rich Media Notifications to Mobile Users
Roman Vitenberg 58

Real-time Explorative Event-based Systems
Mohammad Sadoghi Hamedani 58

The ROAD from Sensor Data to Process Instances via Interaction Mining
Arik Senderovich 59

Cost-Effective Resource Allocation for Deploying Pub/Sub on Cloud
Vinay Setty, Guido Urdaneta, Gunnar Kreitz, and Maarten van Steen 59

Challenges of Data Integration in Cross-Organisational Processes
Sergey Smirnov 60

Repairing Event Logs
Wei Song and Hans-Arno Jacobsen 60

Efficient Handling of Out-of-Order Events
Jan Sürmeli 61

Research Issues on the Extraction of Process Models from Natural Language Text
Lucinéia Heloisa Thom and Renato César Borges Ferreira 62

Scalable, Expressive Publish/Subscribe Systems
Kaiwen Zhang, Hans-Arno Jacobsen, Mohammad Sadoghi Hamedani, and Roman Vitenberg 62

Participants 64

3 Working groups

3.1 Event Models for BPM: Semantics of Events and Patterns and Formal Methods for Reasoning on Events in the BPM Context

Anne Baumgraß (Synfioo – Potsdam, DE), Alexander Artikis (NCSR Demokritos – Athens, GR), Annika M. Hinze (University of Waikato, NZ), Ken Moody (University of Cambridge, GB), Wolfgang Reisig (HU Berlin, DE), Stefanie Rinderle-Ma (Universität Wien, AT), Stijn Vansummeren (Free University of Brussels, BE), and Matthias Weidlich (HU Berlin, DE)

License Creative Commons BY 3.0 Unported license
© Anne Baumgraß, Alexander Artikis, Annika M. Hinze, Ken Moody, Wolfgang Reisig, Stefanie Rinderle-Ma, Stijn Vansummeren, and Matthias Weidlich

Event models are well known in the complex event processing (CEP) area, while process models are well known in the business process management (BPM) area. The combination of both is beneficial and promising but it is facing several challenges.

The benefits can be shown along three examples. First, in the field of logistics, phenomena such as missed connections, congestions, technical problems or strikes may impact on the timely arrival of trucks at a destination. Transportation itself is thus dealing with much uncertainty. Uncertainties are broadly investigated in the area of event processing, however not in process models or process instance definitions. A second example is that of discovering event patterns in order to predict hazardous situations (e.g., in forestry or mining) [1]. If no event logs exist that can be mined for event pattern discovery, patterns may have to be derived with incomplete information about adverse events (e.g., fatal accidents). Exploring the involved work processes may provide helpful information about events to monitor. Third, for care processes, uncertainties such as exceptional patient conditions might trigger therapy adaptations. Here, event models might be helpful in order to support nurses in deciding on which adaptation to apply.

From the CEP perspective the examples include two modelling topics: the modelling of time, and the modelling of uncertainty. Issues concerning the modelling of time include the meaning of time-points itself (do they represent application time, detection time, processing time?) [2] and the nature of time associated to events (e.g., is it a single time-point or an interval?) [3]. Uncertainty in events (e.g., due to noisy sensor readings or unreliable network transmission) can be managed and modelled by associating probabilities to events, and calculating derive probabilities for complex events [4]. In BPM, events are used to represent milestones, trigger instantiation, define deadlines, specify message exchange or communication and many more [5]. For this a simplistic definition of events is sufficient. It defines a discrete, atomic occurrence of a happening. However, we have to exclude events from the process model that can not be defined. The process model itself determines the exact execution, not allowing for executions that have not been defined. Furthermore, adaptive processes have been investigated in BPM [6, 7, 8]. User support for process adaptations has been addressed by few works [9, 10], but the treatment and utilisation of event models has not been considered yet.

We can benefit from the combination of both worlds in the following ways:

- Events trigger changes on the instance or process level
- Event classes in a process model are a means to abstract from the exact definition of an event and the production is moved to event pattern in CEP
- Probabilities may be introduced in process models to enrich them

- Temporal constraints allow for more flexible ways of handling event occurrences, e.g., through the use of intervals
- Consider rich event structures in process models
- Use process models as structured context for CEP and event pattern definitions

Derived from this we define the specific challenges:

- **Guided process instance adaptation:** How can events be used to guide process instance adaptation? Can we use event classes? How to identify them? How to identify the levels of abstraction?
- **Events change states of entities, while this state determines processing:** How can we understand the change of properties as an event which triggers/influences the process instances and determines the flow?
- **Event context through processes:** Can the process give more context that can be used to define pattern better? Can process knowledge give sense to situations that you discover with event processing? How does it help in the discovery of events?

References

1 J. Bowen, A. Hinze, and SJ Cunningham. Into the woods. In *Workshop on Human Work Interaction Design (HWID): Design for Challenging Work Environments, Interact conference, September 2015.*

2 Christian S. Jensen, James Clifford, Ramez Elmasri, Shashi K. Gadia, Patrick J. Hayes, and Sushil Jajodia. A consensus glossary of temporal database concepts. *SIGMOD Record*, 23(1):52–64, 1994.

3 Michael H. Böhlen, Renato Busatto, and Christian S. Jensen. Point-versus interval-based temporal data models. In *ICDE*, pages 192–200. IEEE Computer Society, 1998.

4 Gianpaolo Cugola, Alessandro Margara, Matteo Matteucci, and Giordano Tamburrelli. Introducing uncertainty in complex event processing: model, implementation, and validation. *Computing*, 97(2):103–144, 2015.

5 Mathias Weske. *Business Process Management – Concepts, Languages, Architectures, 2nd Edition.* Springer, 2012.

6 Stefanie Rinderle, Manfred Reichert, and Peter Dadam. Correctness criteria for dynamic changes in workflow systems – a survey. *Data Knowl. Eng.*, 50(1):9–34, 2004.

7 Barbara Weber, Manfred Reichert, and Stefanie Rinderle-Ma. Change patterns and change support features – enhancing flexibility in process-aware information systems. *Data Knowl. Eng.*, 66(3):438–466, 2008.

8 W. Song and H. A. Jacobsen. Static and dynamic process change. *IEEE Transactions on Services Computing*, PP(99):1–1, 2016.

9 Barbara Weber, Manfred Reichert, Stefanie Rinderle-Ma, and Werner Wild. Providing integrated life cycle support in process-aware information systems. *Int. J. Cooperative Inf. Syst.*, 18(1):115–165, 2009.

10 Georg Kaes and Stefanie Rinderle-Ma. Mining and querying process change information based on change trees. In *Service-Oriented Computing – 13th International Conference, ICSOC 2015, Goa, India, November 16-19, 2015*, pages 269–284, 2015.

3.2 Compliance, Audit, Privacy and Security

David Eyers (University of Otago, NZ), Jean Bacon (University of Cambridge, GB), Martin Jergler (TU München, DE), Ken Moody (University of Cambridge, GB), Stefanie Rinderle-Ma (Universität Wien, AT), and Stijn Vansummeren (Free University of Brussels, BE)

License Creative Commons BY 3.0 Unported license
© David Eyers, Jean Bacon, Martin Jergler, Ken Moody, Stefanie Rinderle-Ma, and Stijn Vansummeren

Applications must demonstrate compliance with policy, law and regulation. Audit is a means of achieving this. Regulations often concern privacy and security. In this space, there is a number of challenges and opportunities for the integration of process-oriented and event-based systems.

Reviewing the state-of-the-art, we observe that business process compliance has been concerned with checking/enforcing policies over business process models (design time, e.g., [4]) or process executions, mostly in the form of examination of event logs and traces (e.g., runtime monitoring, see survey in [3]). Security of business processes is often concerned with access control and anomaly detection, see survey in [2]. In event-based systems, there has been research on access control (AC) e.g., via parametrised RBAC [1] at the application level. AC is also well-established in the field of Business Process Management (BPM).

Near-term research challenges and opportunities in this space relate to the processing of audit logs, where engines for Complex Event Processing (CEP) may turn out to be a useful tool. BPM tools, in turn, may provide a useful means of expressing policies in event-based systems. Furthermore, Service Level Agreements (SLAs) are imposed on CEP engines, such as the requirement to provide composite event detection within a particular deadline, e.g., stock quote matching. There is the potential to impose similar SLAs on process-oriented systems.

Mid-term challenges relate to the observation that event-based systems tend to be developed *ad hoc*, and likewise are their application-level access control policies. BPM provides rich access control context that might be used to enrich access control within event-based systems. For example, principals' connections to process and task instances may carry across usefully to parameterised access control systems and enforcement of constraints such as Dynamic Separation of Duties (DSoD).

In the long run, a major challenge is that law is hard to translate into policy, both through being expressed in natural language, and interpreted within case law. There will be an increasing need for automation of enforcing and demonstrating compliance, given the emerging domains of Cloud Computing and the Internet of Things. Policies will need to be expressed and interpreted by machines at run-time, and be able to be validated against the law as it stands.

In terms of application logic the combination of different rules is a problem—policies may conflict at run-time. Conflict detection and resolution is difficult. Conflict may arise for many reasons, such as incorrect policy specification, an inability to meet obligations, or due to unexpected context.

Applications and audit logs are increasingly cloud hosted. While cloud tenants hold the legal responsibility, they do not have technical control over what cloud service providers are doing, and there is no transparency. For example, access to audit logs must be controlled. Also, they may be copied across international boundaries and jurisdictions.

References

1 Jean Bacon, David M. Eyers, Jatinder Singh, and Peter R. Pietzuch. Access control in publish/subscribe systems. In *DEBS'08: Proceedings of the second international conference on Distributed event-based systems*, pages 23–34, New York, NY, USA, 2008. ACM. doi:10.1145/1385989.1385993.

2 Maria Leitner and Stefanie Rinderle-Ma. A systematic review on security in process-aware information systems – constitution, challenges, and future directions. *Information and Software Technology*, 56(3):273–293, 2014. doi:10.1016/j.infsof.2013.12.004.

3 Linh Thao Ly, Fabrizio Maria Maggi, Marco Montali, Stefanie Rinderle-Ma, and Wil M.P. van der Aalst. Compliance monitoring in business processes: Functionalities, application, and tool-support. *Information Systems*, 54:209–234, 2015. doi:10.1016/j.is.2015.02.007.

4 Shazia Sadiq, Guido Governatori, and Kioumars Namiri. Modeling control objectives for business process compliance. In *Proceedings of the 5th International Conference on Business Process Management*, BPM'07, pages 149–164, Berlin, Heidelberg, 2007. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=1793114.1793130

3.3 Towards Automatic Event-Based Monitoring of Processes

Annika M. Hinze (University of Waikato, NZ), Alexander Artikis (NCSR Demokritos – Athens, GR), Anne Baumgraß (Synfioo – Potsdam, DE), Alejandro P. Buchmann (TU Darmstadt, DE), Claudio Di Ciccio (Wirtschaftsuniversität Wien, AT), Hans-Arno Jacobsen (TU München, DE), Boris Koldehofe (TU Darmstadt, DE), Alessandro Margara (University of Lugano, CH), Pnina Soffer (Haifa University, IL), and Holger Ziekow (FH Furtwangen, DE)

License Creative Commons BY 3.0 Unported license
© Annika M. Hinze, Alexander Artikis, Anne Baumgraß, Alejandro P. Buchmann, Claudio Di Ciccio, Hans-Arno Jacobsen, Boris Koldehofe, Alessandro Margara, Pnina Soffer, and Holger Ziekow

The dynamic interplay between event processing and process modelling was discussed along three dimensions, using a logistics example for illustration.

D1. Pattern discovery vs. Monitoring of known patterns

Pattern discovery is typically done once sufficient data is available (e.g., on event logs [18]) and rarely online while monitoring is done online. Online "event" discovery may be relevant in the case of many parallel event sources or in emergency situations that cannot rely on available event logs. Pattern discovery is typically done either via a process norm, i.e., detecting if a process deviates from expectation or via a process goal, i.e., prediction of success or failure of a process. For instance, the typical patterns of events about a delivery truck driving from A to B can be identified in an unsupervised manner, while supervised pattern learning requires labelling the event sequences to indicate whether the activity execution had positive or negative outcomes [10] (e.g., shipment activities that respectively delivered on time or delayed [9]). Once desired patterns are known, event streams are then evaluated online to identify the occurrence of these patterns. However, on-the-fly adaptation of the monitoring pattern may be necessary.

D2. Monitoring for adaptation vs. adaptive monitoring

Monitoring for adaptation aims to identify events that then trigger changes in the business process. While the announcement is on the process level, the adaptation is carried out on the instance level. For example, if a parcel delivery misses a shipping deadline, it may have to be redirected to go via train. For event processing, this kind of monitoring and action triggering is business as usual (e.g., change of flow depending on event sources and sinks). In contrast, monitoring adaptively is an event processing issue that can be guided by business process information. For example, the closer to the ETA of a delivery by truck, the more frequently the traffic situation needs to be monitored. Relevant information from processes could be: patterns to be monitored, constraints (deadlines, tolerance to false positives or negatives), utility functions, acceptable levels of monitoring, monitoring intervals, monitoring points, QoS as foundation to manage automatic adaptation. Changes in the monitoring may then be made, e.g., to the frequency or granularity of monitoring [5], the observed source of events and streams [2], the monitoring points within the process [1], and to variables and rules [6].

D3. Use of context for integration vs. context awareness for decision support

Context knowledge can drive the integration of heterogeneous sources. For example, instead of monitoring each parcel inside a container, the container ID is used to identify events of interest, and later the truck transporting the container. Without context knowledge, such integrations would not be possible. Context and situation awareness may enhance decision support, e.g., in the selecting between two alternative process paths [8]. When it is known that a thunderstorm may impair airplane landings, the shipment could be transported via truck instead.

Open Challenges

We thus envision the following open challenges, classified by their assessed time scope:
1. Using automatically discovered patterns in online monitoring of BPMSs (short-term)
2. Leveraging process background knowledge for pattern discovery (short-term)
3. Leveraging context to drive the integration of heterogeneous sources (short-term)
4. Monitoring events to guide process adaptation (mid-term)
5. Process information to guide the monitoring adaptation (mid-term)
6. Discovery of patterns at runtime (long-term)
7. Leveraging context and situation awareness to enhance decision support (long-term)

References

1 Anne Baumgrass, Cristina Cabanillas, and Claudio Di Ciccio. A conceptual architecture for an event-based information aggregation engine in smart logistics. In *EMISA*, pages 109–123. GI, September 2015.
2 Cristina Cabanillas, Claudio Di Ciccio, Jan Mendling, and Anne Baumgrass. Predictive task monitoring for business processes. In *BPM*, pages 424–432. Springer, September 2014. doi:10.1007/978-3-319-10172-9_31.
3 Tony Chau, Vinod Muthusamy, Hans-Arno Jacobsen, Elena Litani, Allen Chan, and Phil Coulthard. Automating SLA modelling. In Marsha Chechik, Mark R. Vigder, and Darlene A. Stewart, editors, *Proceedings of the 2008 conference of the Centre for Advanced Studies on Collaborative Research, October 27-30, 2008, Richmond Hill, Ontario, Canada*, page 10. IBM, 2008. doi:10.1145/1463788.1463802.

4 Alex King Yeung Cheung and Hans-Arno Jacobsen. Load balancing content-based publish/subscribe systems. *ACM Trans. Comput. Syst.*, 28(4):9, 2010. `doi:10.1145/1880018.1880020`.

5 Claudio Di Ciccio, Han van der Aa, Cristina Cabanillas, Jan Mendling, and Johannes Prescher. Detecting flight trajectory anomalies and predicting diversions in freight transportation. *Decision Support Systems*, 88:1–17, August 2016. `doi:10.1016/j.dss.2016.05.004`.

6 Sebastian Frischbier. *Runtime Support for Quality of Information Requirements in Event-based Systems.* PhD thesis, Technische Universität Darmstadt, 2016.

7 Guoli Li, Vinod Muthusamy, and Hans-Arno Jacobsen. A distributed service-oriented architecture for business process execution. *TWEB*, 4(1), 2010. `doi:10.1145/1658373.1658375`.

8 David C Luckham. *Event processing for business: organizing the real-time enterprise.* John Wiley & Sons, 2011.

9 Alessandro Margara, Gianpaolo Cugola, and Giordano Tamburrelli. Learning from the past: automated rule generation for complex event processing. In *DEBS*, pages 47–58. ACM, 2014.

10 Thomas M. Mitchell. *Machine Learning.* McGraw Hill series in computer science. McGraw-Hill, Inc., New York, NY, USA, first edition, 1997.

11 Vinod Muthusamy and Hans-Arno Jacobsen. BPM in cloud architectures: Business process management with SLAs and events. In Richard Hull, Jan Mendling, and Stefan Tai, editors, *Business Process Management – 8th International Conference, BPM 2010, Hoboken, NJ, USA, September 13-16, 2010. Proceedings*, volume 6336 of *Lecture Notes in Computer Science*, pages 5–10. Springer, 2010. `doi:10.1007/978-3-642-15618-2_2`.

12 Vinod Muthusamy, Hans-Arno Jacobsen, Tony Chau, Allen Chan, and Phil Coulthard. SLA-driven business process management in SOA. In Patrick Martin, Anatol W. Kark, and Darlene A. Stewart, editors, *Proceedings of the 2009 conference of the Centre for Advanced Studies on Collaborative Research, November 2-5, 2009, Toronto, Ontario, Canada*, pages 86–100. ACM, 2009. `doi:10.1145/1723028.1723040`.

13 Vinod Muthusamy, Haifeng Liu, and Hans-Arno Jacobsen. Predictive publish/subscribe matching. In Jean Bacon, Peter R. Pietzuch, Joe Sventek, and Ugur Çetintemel, editors, *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS 2010, Cambridge, United Kingdom, July 12-15, 2010*, pages 14–25. ACM, 2010. `doi:10.1145/1827418.1827423`.

14 Vinod Muthusamy, Young Yoon, Mohammad Sadoghi, and Hans-Arno Jacobsen. eqosystem: supporting fluid distributed service-oriented workflows. In David M. Eyers, Opher Etzion, Avigdor Gal, Stanley B. Zdonik, and Paul Vincent, editors, *Proceedings of the Fifth ACM International Conference on Distributed Event-Based Systems, DEBS 2011, New York, NY, USA, July 11-15, 2011*, pages 381–382. ACM, 2011. `doi:10.1145/2002259.2002320`.

15 Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Distributed event aggregation for content-based publish/subscribe systems. In Umesh Bellur and Ravi Kothari, editors, *The 8th ACM International Conference on Distributed Event-Based Systems, DEBS'14, Mumbai, India, May 26-29, 2014*, pages 95–106. ACM, 2014. `doi:10.1145/2611286.2611302`.

16 Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Minimizing the communication cost of aggregation in publish/subscribe systems. In *35th IEEE International Conference on Distributed Computing Systems, ICDCS 2015, Columbus, OH, USA, June 29 – July 2, 2015*, pages 462–473. IEEE Computer Society, 2015. `doi:10.1109/ICDCS.2015.54`.

17 Mohammad Sadoghi, Martin Jergler, Hans-Arno Jacobsen, Richard Hull, and Roman Vaculín. Safe distribution and parallel execution of data-centric workflows over the publish/subscribe abstraction. *IEEE Trans. Knowl. Data Eng.*, 27(10):2824–2838, 2015. doi:10.1109/TKDE.2015.2421331.
18 Wil M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011. doi:10.1007/978-3-642-19345-3.

3.4 Patterns and Models for Communication

Boris Koldehofe (TU Darmstadt, DE), Oliver Kopp (Universität Stuttgart, DE), Wolfgang Reisig (HU Berlin, DE), Martin Ugarte (Free University of Brussels, BE), and Roman Vitenberg (University of Oslo, NO)

License Creative Commons BY 3.0 Unported license
© Boris Koldehofe, Oliver Kopp, Wolfgang Reisig, Martin Ugarte, and Roman Vitenberg

Communication in processes, in particular by means of events, has many implicit effects on the modelled process, which are typically disregarded. Often many assumptions are implicit, e.g. it is assumed that the services are always on or all events are eventually received. Blocking communication may prevent processes from making progress if the expected event fails to occur. Counter measures like timeouts introduce assumptions on the timeliness of a communication system which may not be supported by a communication middleware. In general the behaviour of processes is affected by the behaviour of the middleware. The concrete properties of a middleware are often not exposed and are thus unclear for the process modeller. This poses an important source for badly specified processes and erroneous deviations of the process execution from the intended process behaviour.

For example, consider the process to collect votes for deciding on the public money to spend on a big project such as the Stuttgart 21 underground train station. Each admitted voter can send a vote and in addition update the vote within a given voting period. The process modelled in Figure 1 is underspecified in several ways: best effort delivery of events, reordered messages as well as timeouts may impose all changes to the number of votes collected.

Therefore it will be important to reflect these properties during process modelling. This will enable process modelling implementation/design to opt for a particular middleware, based

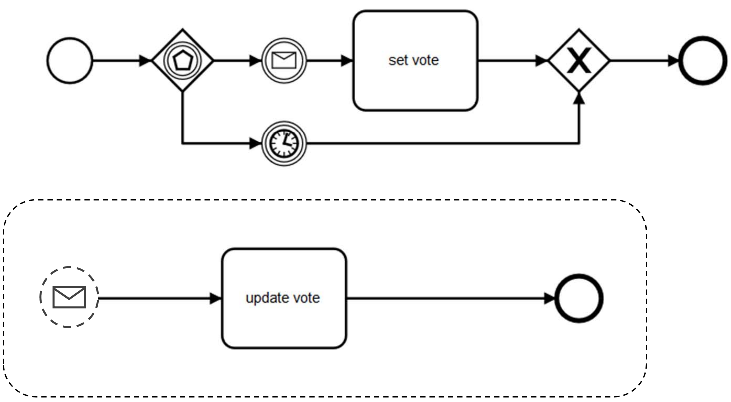

Figure 1 Patterns and Models for Communication.

on the provided properties. The requirement specification for the process model may need to be mapped to the lower level specification of the communication middleware. Although there are works addressing failure models and formal specifications of middleware properties, e.g. [3, 1], there seems to be a significant modelling gap in connecting the middleware and the process layer. In closing this gap, several mid- and long-term challenges are arising.

A first challenge detected is to collect the right terms to assert the requirements or conditions that are imposed over communication. This is identified as a short-term challenge, given that it should be a starting point for middleware requirement specification. Consequently, a mid-term challenge is to provide a way for exposing middleware properties to the process modeller. Tasks like reliability and latency are today left out of the process modelling, given that modellers do not take into account the characteristics of the middleware. Another mid-term challenge is to understand that these specifications may affect (positively or negatively) the verification of processes. For example, requiring that events arrive in order provides certain guarantees on consistency; whereas an out of order arrival of a process could impact the process (e.g. loss of information). Once the previous problems are understood, a long term challenge would be to automatically determine from process descriptions suitable middleware components that are compliant, provide a good (or optimal) set of QoS guarantees, and to minimise the cost for deployment and execution. Such components may be available at a marketplace that offers appropriate components, building tools, and methods for dynamic adaptations, e.g. transitions between middleware components [SHK+15].

References

1 Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33, January 2004. doi:10.1109/TDSC.2004.2.

2 Oliver Kopp, Frank Leymann, and Daniel Wutke. Fault handling in the web service stack. In *Service-Oriented Computing*, pages 303–317. Springer Science + Business Media, 2010. doi:10.1007/978-3-642-17358-5_21.

3 J. C. Laprie, editor. *Dependability: Basic Concepts and Terminology : In English, French, German, Italian and Japanese (Dependable Computing and Fault-Tolerant Systems)*. Springer, 1991.

4 Ralf Steinmetz, Melanie Holloway, Boris Koldehofe, Björn Richerzhagen, and Nils Richerzhagen. Towards future internet communications – role of scalable adaptive mechanisms. In *27th Annual Conference on Symbiosis – Synergy of Humans & Technology*, pages 59–61. Eurographics Association, 2016.

3.5 Choreographies and Inter-Process Correlation

Oliver Kopp (Universität Stuttgart, DE), Wolfgang Reisig (HU Berlin, DE), Jatinder Singh (University of Cambridge, GB), Sergey Smirnov (SAP SE – Walldorf, DE), Jan Sürmeli (HU Berlin, DE), Roman Vitenberg (University of Oslo, NO), Matthias Weidlich (HU Berlin, DE), and Kaiwen Zhang (TU München, DE)

License Ⓒ Creative Commons BY 3.0 Unported license
© Oliver Kopp, Wolfgang Reisig, Jatinder Singh, Sergey Smirnov, Jan Sürmeli, Roman Vitenberg, Matthias Weidlich, and Kaiwen Zhang

Choreographies enable modelling inter-organisational processes [9]. The idea is to abstract from the processes of each organisation and provide a global view. There are multiple choreography languages available, which offer two different modelling styles: interaction models and interconnection models [8]. In interaction models ('choreography diagrams' in BPMN 2.0), each message exchange is made atomic, whereas in interconnection models ('collaboration diagrams' in BPMN 2.0), the abstract process of each participant is shown. There are various studies on the expressiveness of the languages (e.g., [5, 2]). For implementing choreographies, each participant can be implemented using processes, web services, or other approaches [3]. We see advantages in also considering event-based systems [1] as they offer an established technology, especially with respect to performance, scalability, community size, and commercial support.

There exist approaches mapping (a subset of) orchestration languages [7] and declarative process modelling languages [6] to event-based systems. Choreography modelling languages lack advanced event-based concepts, such as subscription and correlation between matched events and process instances. The CASU framework [4] presents a discussion regarding processes and instantiation. This has to be extended to the whole life-cycle of choreographies. Moreover, the formal understanding about the relation between choreography models and event-based implementations is missing. Finally, a holistic, systematic top-down approach from (conceptual) choreography models to event-based (implementation) models is required. We thus propose the following research goal for the near and far future.

3.5.1 Short-Term Goals

Extension of existing choreography modelling methods

The goal is to facilitate a systematic approach to the transformation of conceptual choreography models to models for their event-based implementation. This requires the identification and addition of missing notions in choreography modelling languages, such as subscription, correlation, and non-functional properties. Likewise, it is necessary to extend the methodological concepts accordingly, such as roles, steps, best practices and deliverables.

Introduction of intermediate implementation models

We aim at bridging the gap between conceptual models and their implementation. To this end, we propose to introduce an intermediate model, establishing a clear relation between a conceptual model and its implementation.

Identification of evaluation goals

The developed concepts have to be evaluated and validated. We aim to bring up concrete evaluation scenarios, concepts, and performance indicators to ensure a scientifically grounded evaluation.

3.5.2 Mid-Term Goals

Analysis

Analysis aspects regarding information flow have to be developed. For example, it should be ensured by the analysis that partners that do not directly interact do not gain additional information. The goal is to identify relevant analysis questions and to answer them. This includes concrete classical correctness and completeness properties.

Crisp simplistic theoretical foundation

The semantics of both the extended choreography language and event-based systems have to be formally defined. This enables formal analysis techniques to answer the analysis questions raised above.

3.5.3 Long-Term Goals

Increased level of automation

The mapping between the choreography model, the intermediate model and the event-based implementation should be as automatic as possible. At least, a skeleton of the intermediate model should be derived from the choreography model.

Implementation of logic as event-based systems

An approach to move choreography logic into event-based systems should be developed, preserving BPM benefits such as monitoring and analysis. Both the internal logic and the logic of the called services itself should be moved into event-based systems.

References

1 Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3):15:1–15:62, June 2012. doi:10.1145/2187671.2187677.
2 Gero Decker, Alistair P. Barros, Frank Michael Kraft, and Niels Lohmann. Non-desynchronizable Service Choreographies. In Athman Bouguettaya, Ingolf Krüger, and Tiziana Margaria, editors, *ISCOC 2008: 6th International Conference on Service-Oriented Computing*, volume 5364 of *LNCS*, pages 331–346, 2008.
3 Gero Decker, Oliver Kopp, Frank Leymann, and Mathias Weske. Interacting services: From specification to execution. *Data & Knowledge Engineering*, 68(10):946–972, April 2009.
4 Gero Decker and Jan Mendling. Process Instantiation. *Data & Knowledge Engineering*, 68:777–792, 2009.
5 Gero Decker and Mathias Weske. Local enforceability in interaction Petri nets. In *Business Process Management*. Springer, 2007.
6 Martin Jergler, Mohammad Sadoghi, and Hans-Arno Jacobsen. D2WORM: A management infrastructure for distributed data-centric workflows. In *SIGMOD Conference*, pages 1427–1432. ACM, 2015.
7 Guoli Li, Vinod Muthusamy, and Hans-Arno Jacobsen. A distributed service-oriented architecture for business process execution. *ACM Trans. Web*, 4(1):1–33, Jan. 2010. doi:10.1145/1658373.1658375.
8 Andreas Schönberger. Do we need a refined choreography notion? In *3rd Central-European Workshop on Services and their Composition (ZEUS 2011)*. CEUR-WS.org, 2011.
9 Wil M. P. van der Aalst and Mathias Weske. The P2P approach to interorganizational workflows. In *Seminal Contributions to Information Systems Engineering*, pages 289–305. Springer, 2013. doi:10.1007/978-3-642-36926-1_23.

3.6 Abstraction Levels: Processes versus Events

Sankalita Mandal (Hasso-Plattner-Institut – Potsdam, DE), David Eyers (University of Otago, NZ), Agnes Koschmider (KIT – Karlsruher Institut für Technologie, DE), Ken Moody (University of Cambridge, GB), Cesare Pautasso (University of Lugano, CH), Mohammad Sadoghi Hamedani (Purdue University, US), Wei Song (Nanjing University of Science & Technology, CN), Lucinéia Heloisa Thom (Federal University of Rio Grande do Sul, BR), and Lijie Wen (Tsinghua University Beijing, CN)

License Creative Commons BY 3.0 Unported license
© Sankalita Mandal, David Eyers, Agnes Koschmider, Ken Moody, Cesare Pautasso, Mohammad Sadoghi Hamedani, Wei Song, Lucinéia Heloisa Thom, and Lijie Wen
URL https://docs.google.com/document/d/10SSdxSAmjT2WbLHJmO_RgqrP5_1Wewbi52N-JMR3oqw

3.6.1 How to characterise and compare processes versus events?

Modelling

In general, process management follows the top-down approach where we start from a business goal and then model and execute the activities required to achieve it. So the outcome of a process would be the actions that are taken and the goal that is fulfilled collectively by those actions. Event processing typically follows a bottom-up approach where we start with the raw events and aggregate them based on some specific patterns or rules. Thus, the outcome of event processing are higher level or complex events. Though this aggregation may take time due to event buffering, simple events occur instantaneously. Processes (and their activities) instead have a duration. Processes model complete state machines, while events focus on transitions. From a graph representation perspective, processes correspond to nodes while events are on the edges. Unlike the arbitrary (though directed) topology structure for processes, the graph topology for event aggregation can be very well depicted by a tree structure. So far, process models are targeted towards the business analysts and the CEP developers take care of specifying the event hierarchies.

Runtime execution

Both process and event processing performance is measured in terms of latency and throughput. Scalability may depend on: number of process instances, size of process models, number of users versus number of rules and number of event types and stream sources. Both facilitate building flexible systems as processes can be dynamically adapted and evolved. Likewise, the set of CEP rules can be changed on the fly. In terms of monitoring, processes are concerned with the progress of their execution and give an explicit and persistent representation of their state and execution history, which could also ease debugging and testing. While (some) process engines offer transactional guarantees to deal with outages during the execution of processes, it remains to be seen whether process or event engines can be safely introduced within a safety critical system.

3.6.2 How are processes and events connected?

Processes are triggered by events. Processes catch events, which will produce a transition of their execution state and trigger the execution of further activities and the emission of events following the control flow structure of the process model. Events are consumed or produced by processes. Process start, process completion, and many other intermediate events can be explicitly represented in a process model. Low-level events produced during the execution of a process and its activities can be logged and used for monitoring, auditing, etc. In general, the execution semantics of a process model corresponds to a partial order of events.

3.6.3 What are some challenges to integrate processes and events on a conceptual level?

How to find the suitable abstraction level for different modelling goals (time, space, cost)? How to deal with conflicting sources in large-scale systems integration? How to handle unexpected events? The "serious" Smart City scenario may provide a suitable challenging context to evaluate integrated event/process platforms.

3.7 Context in Events and Processes

Alessandro Margara (University of Lugano, CH), Alejandro P. Buchmann (TU Darmstadt, DE), Sankalita Mandal (Hasso-Plattner-Institut – Potsdam, DE), Cesare Pautasso (University of Lugano, CH), Arik Senderovich (Technion – Haifa, IL), Sergey Smirnov (SAP SE – Walldorf, DE), Matthias Weidlich (HU Berlin, DE), and Mathias Weske (Hasso-Plattner-Institut – Potsdam, DE)

License Creative Commons BY 3.0 Unported license
© Alessandro Margara, Alejandro P. Buchmann, Sankalita Mandal, Cesare Pautasso, Arik Senderovich, Sergey Smirnov, Matthias Weidlich, and Mathias Weske

Context-aware computing was discussed by Schilit and Theimer in 1994 to be software that "adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time". In other words, context is the model of the environment that provides shared knowledge useful for the processes or event patterns at hand. For instance, the currency of a financial transaction depends on the country in which a process or an event pattern is executed: abstracting the concept of currency and modelling it as part of the context can help to simplify the pattern or process definition, avoid duplicate definitions and terminology ambiguity and biases. Thus, modelling context separately with respect to the core event patterns and processes strengthens separation of concerns, eases maintainability, and facilitates the reuse of the shared context between different processes. Several models have been proposed in the literature that could be used as a starting point to represent the context of processes and event patterns.

When considering the integration of business processes and event based systems, events can provide and contribute to detect the context for the process analysis, execution and external interaction. Similarly, processes provide the context for event pattern evaluation, optimisation and filtering. More in detail:

- The process provides the context in which event patterns operate.
- The process determines which events are relevant (filtering).
- The process determines if events will no longer arrive (optimisation).
- The process determines validity windows for pattern detection (session windows).
- The events and the situations that can be inferred through event pattern detection define the context in which the processes operate.
- Context events influence how processes make decisions (control flow).
- Context events determine which process variant is selected for deployment (variability).
- Context events constrain the binding of resources with activities (dynamic binding).
- Context compatibility is required for event subscriptions (interop, external data flow).

- Context can provide a global state shared between multiple instances (internal data flow).
- Context may impact the granularity of the monitoring events that are collected from one or more process instances.

Challenges

Representation of context: Which formalisms and models can be adopted to represent the context for processes and for event patterns? Are existing models adequate or we need new formalisms? Is a single formalism sufficient to capture both the context of processes and the context of event patterns?

Scope of context: How to distinguish what is included in the process and what is context? Context alignment at design time – how to determine the points for integrating context into processes? Should process and context remain orthogonal/independently modelled? If the process and event do not share the context, how to convert from one to another?

Context alignment at runtime: How to relate the running process with correct (smart) context? Which events are relevant for the current instance? How to discover the hidden context which explains variations in process outcomes?

References

1 Cristiana Bolchini, Carlo Curino, Elisa Quintarelli, Fabio A. Schreiber, and Letizia Tanca. A data-oriented survey of context models. *SIGMOD Record*, 36(4):19–26, 2007. `doi:10.1145/1361348.1361353`.

2 Ludger Fiege, Mariano Cilia, Gero Mühl, and Alejandro P. Buchmann. Publish-subscribe grows up: Support for management, visibility control, and heterogeneity. *IEEE Internet Computing*, 10(1):48–55, 2006. `doi:10.1109/MIC.2006.17`.

3 Tobias Freudenreich. *Simplifying the use of event-based systems with context mediation and declarative descriptions.* PhD thesis, Technische Universität, Darmstadt, 2015. URL: `http://tuprints.ulb.tu-darmstadt.de/5131/`.

4 Cesare Pautasso and Gustavo Alonso. Flexible binding for reusable composition of web services. In Thomas Gschwind, Uwe Aßmann, and Oscar Nierstrasz, editors, *Software Composition, 4th International Workshop, SC 2005, Edinburgh, UK, April 9, 2005, Revised Selected Papers*, volume 3628 of *Lecture Notes in Computer Science*, pages 151–166. Springer, 2005. `doi:10.1007/11550679_12`.

5 Michael Rosemann and Jan Recker. Context-aware process design exploring the extrinsic drivers for process flexibility. In Gil Regev, Pnina Soffer, and Rainer Schmidt, editors, *Proceedings of the CAISE*06 Workshop on Business Process Modelling, Development, and Support BPMDS'06, Luxemburg, June 5-9, 2006*, volume 236 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006. URL: `http://ceur-ws.org/Vol-236/paper9.pdf`.

6 Oumaima Saidani and Selmin Nurcan. Towards context aware business process modelling. In *8th Workshop on Business Process Modeling, Development, and Support (BPMDS'07), CAiSE*, volume 7, page 1, 2007.

3.8 Integrated Platforms for BPM & CEP

Mohammad Sadoghi Hamedani (Purdue University, US), Alejandro P. Buchmann (TU Darmstadt, DE), Hans-Arno Jacobsen (TU München, DE), Martin Jergler (TU München, DE), Sankalita Mandal (Hasso-Plattner-Institut – Potsdam, DE), Cesare Pautasso (University of Lugano, CH), Stefan Schulte (TU Wien, AT), Jatinder Singh (University of Cambridge, GB), Sergey Smirnov (SAP SE – Walldorf, DE), and Mathias Weske (Hasso-Plattner-Institut – Potsdam, DE)

License Creative Commons BY 3.0 Unported license
© Mohammad Sadoghi Hamedani, Alejandro P. Buchmann, Hans-Arno Jacobsen, Martin Jergler, Sankalita Mandal, Cesare Pautasso, Stefan Schulte, Jatinder Singh, Sergey Smirnov, and Mathias Weske

In general, BPM platforms are intended for expressive and flexible modelling of (business) processes while CEP platforms are designed for formulation and efficient detection of patterns or composite events. One may further view processes as partial ordering of events (i.e., a pattern or sequence of events) thereby giving rise to exploiting CEP as an efficient backend for executing BPM. On the one hand, unlike specialised BPM platforms, CEP has a loosely coupled architecture using a decentralised messaging substrate that is governed by publish/subscribe paradigm. As a result, the global state of CEP is distributed in the decentralised fabric of publish/subscribe (i.e., indirect state), which may have the potential to provide better compliant support for private data. On the other hand, BPM platforms rely on central messaging architecture with a tighter coupling among processes that rely on a more complex interaction and interfaces in comparison to basic publish and subscribe primitives. As a result, the global state of BPM is directly captured, which further simplifies the governance of the execution and the quality of service such as message ordering guarantees and transactional support. These benefits are gained at the cost of central execution of processes using a specialised (and possibly an *ad hoc*) runtime.

Therefore, we envision significant opportunities in integrating BPM and CEP platforms to accelerate the development and reduce the maintenance cost of building the BPM (or even CEP) engine. At one extreme, the CEP can be used exclusively as an enabler of BPM by mapping processes into a chain of rule firings that is formulated as a set of subscriptions/publications (e.g., [1-4]). Thus, the process is now represented as a lightweight, distributed agent that simply issues a set of subscriptions and publications on behalf of one or more processes. A weaker form of integration is to exploit the CEP engine only as notification or messaging substrate while the process execution runtime remains within the BPM platform. Alternatively, the CEP engine could be used only for the dynamic monitoring health and progress of processes using the generated events in BPM or the offline mining of the event log to support functionalities such as compliance and auditing. At the other extreme, one may explore the role of BPM as an enabler of CEP in order to benefit from the visual and flexible modelling power of BPM.

The key research problem in this space is to develop a unified event-process model (such as representing a process as a partial order of events) in order to enable building a unified engine that can model and execute both events and processes in a scalable, decentralised, distributed, and secured architecture (possibly by exploiting the inherent decoupling of the publish/subscribe paradigm).

References

1 Martin Jergler, Hans-Arno Jacobsen, Mohammad Sadoghi, Richard Hull, and Roman Vaculín. Safe distribution and parallel execution of data-centric workflows over the publish/sub-

scribe abstraction. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 1498–1499. IEEE Computer Society, 2016. doi:10.1109/ICDE.2016.7498393.
2 Martin Jergler, Mohammad Sadoghi, and Hans-Arno Jacobsen. D2WORM: A management infrastructure for distributed data-centric workflows. In Timos K. Sellis, Susan B. Davidson, and Zachary G. Ives, editors, *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 – June 4, 2015*, pages 1427–1432. ACM, 2015. doi:10.1145/2723372.2735362.
3 Haifeng Liu and Hans-Arno Jacobsen. A-ToPSS: A publish/subscribe system supporting imperfect information processing. In Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors, *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 – September 3 2004*, pages 1281–1284. Morgan Kaufmann, 2004. URL: http://www.vldb.org/conf/2004/DEMP8.PDF.
4 H.-A. Jacobsen, M. Jergler, M. Sadoghi. Geo-distribution of flexible business processes over publish/subscribe paradigm. In *16th ACM/IFIP/USENIX International Middleware Conference (Middleware 2016)*, Trento, Italy, Dec. 2016.
5 Mohammad Sadoghi, Martin Jergler, Hans-Arno Jacobsen, Richard Hull, and Roman Vaculín. Safe distribution and parallel execution of data-centric workflows over the publish/subscribe abstraction. *IEEE Trans. Knowl. Data Eng.*, 27(10):2824–2838, 2015. doi:10.1109/TKDE.2015.2421331.

3.9 (Highly) Distributed Processes & The Role of Events

Stefan Schulte (TU Wien, AT), Jean Bacon (University of Cambridge, GB), Avigdor Gal (Technion – Haifa, IL), Martin Jergler (TU München, DE), Stefanie Rinderle-Ma (Universität Wien, AT), Arik Senderovich (Technion – Haifa, IL), Vinay Setty (MPI für Informatik – Saarbrücken, DE), Martin Ugarte (Free University of Brussels, BE), and Stijn Vansummeren (Free University of Brussels, BE)

License Creative Commons BY 3.0 Unported license
© Stefan Schulte, Jean Bacon, Avigdor Gal, Martin Jergler, Stefanie Rinderle-Ma, Arik Senderovich, Vinay Setty, Martin Ugarte, and Stijn Vansummeren

The discussion about the nature of process and event distribution yielded a two-dimensional table: Events can come from either a centralised source (e.g., a single patient monitor device) or distributed sources (e.g., surveillance cameras). Process management may be centralised (e.g., performed on a single machine or performed by a single organisational unit) or distributed, which may be done top-down, where processes are distributed for the purpose of efficiency, security, etc., or bottom-up. This classification leads to four combinations, namely centralised events-centralised processes (CECP), centralised events-distributed processes (CEDP), distributed events-centralised processes (DECP) and distributed events-distributed processes (DEDP).

An example that illustrates the four combinations involves a hospital that discharges patients to home monitoring. There is a process for each patient that continuously performs monitoring of vital signs: temperature, pulse rate, etc. If all is well, the data streams are stored locally. If a composite event detection indicates a possible emergency condition, an external event is sent to the hospital. The hospital process is responsible for responding to emergency events from all patients. This may involve invoking other emergency services such as sending an ambulance to the patient's home.

3.9.1 CECP

The classical scenario in BPM is when processes are executed in a centralised manner. Specifically, the activities, decisions, and information flow of the business process happen in a single location. Further, the current scenario considers a situation when events are gathered and processed in a single location. Following our example, patient data is collected from wearable devices that the patient carries around. Further, the monitoring process (which we consider to be a business process) is also executed on the same device. Related work includes the analysis of small data, i.e. individual event gathering, works by Deborah Estrin, e.g. [1], and application paper from machine learning on detecting machine breakdowns [2]. From the BPM perspective, most processes that are considered in the literature except previous work on distributed processes, are centralised.

Challenges

1. Event data collection (privacy (of patients), trust (how much patients trust us, how much do we trust machine/patient sensors), new sensors to collect new kinds of data (perhaps some things are not measurable yet), sample rates (under-sampling vs. over-sampling of the device)).
2. Event processing (local processing on tiny devices, transmission of data).
3. BPM (process mining in small data devices (e.g. time series analysis) – real-time and off-line, granularity in analysis of such data: low-level recordings vs. high-level activity (abstraction gap), workflow recognition (much harder than recognising a single activity), combining historical data with recently collected data for analytics (short-term vs. long-term)).

3.9.2 DECP

Events from distributed sources are collected and processed by a centralised process, which is typically a centralised CEP engine. In the hospital example this occurs in a patient monitoring process. Every patient is equipped with a sensor measuring certain body parameters. If a combination of parameter values indicates a possible emergency (e.g. internal bleeding: low temperature, low blood pressure, high heart rate) an alarm is raised. A centralised monitoring process collects alarm events from individual patients and considers whether to send notifications to call for more doctors. Related work includes centralised event processing engines, e.g. Esper, event ordering Pub/Sub [3], and event aggregation in Pub/Sub.

Challenges

1. Out-of-order arrival of events from different event sources causes misdetection of complex events and thereby corrupts process execution: How can the order of events be guaranteed or how can misordering be detected?
2. Event loss (identification of lost events, compensation of event loss, high volume of event data that needs to be processed, high burst rates of events).

3.9.3 DPCE

This scenario refers to a centralised event source but the processes are distributed for load-balancing purposes. This scenario is critical for applications where fast processing is necessary. With a high volume stream of events centralised processing becomes a bottleneck. For fast processing of events, distributed computing using platforms such as Apache Storm[1] and Spark streaming[2] is often necessary. For example, in a hospital, sensor data from patients' wearable devices is collected in a centralised location but for faster processing, such as scheduling logistics, inventory and global decision making, distributed processing is done. Other tasks like aggregation and top-k may require distribution as well [4, 5].

Challenges

Stock quote matching [6] is an example of a DPCE scenario. Such applications produce massive-scale streams of events from a centralised location. Analysing them in real-time in a centralised setting becomes a bottleneck. Faster response times are critical for many business processes in such scenarios. While distributed platforms such as Storm and Spark streaming can handle massive-scale streams, their response time needs to be improved. The event streams in such a scenario apart from being massive are also unpredictable and consequently require elastic provisioning of resources. Finally since the events are processed in a distributed fashion, correctness of results must be ensured as well.

3.9.4 DEDP

In this setting, both processes and events are distributed. In our example, the hospital may be monitoring multiple patients (generating distributed event streams) and has to coordinate (among its own departments) emergency services, where each department has its own process model and external services are independent (police, ambulance). This is a setting where observed event data is incomplete and uncertain (e.g. because of network delays or noisy sensors). Several approaches exist on how to establish choreographies taking care of criteria such as consistency between different partner processes, e.g. [7]. Less attention has been spent on aspects such as change in distributed processes, e.g. [8]. Distribution in CEP generally refers to parallelised techniques for detecting complex patterns. In this respect, some efforts have been devoted to generate, given a complex pattern, a query plan that allows for federation of sub-patterns (from RETE networks [9] and Information Flow Processing Systems [10] to Intrusion Detection Systems [11]).

Challenges

1. Methods to deal with distributed incompleteness and uncertainty in order to ensure reliability of process enactment.
2. Methods for resilient data integration are required to improve event completeness and accuracy.
3. Methods to deal with privacy and security are required.
4. Events may trigger the local processes to change; also affecting the global process, hence requiring management of ripple effects.

References

1 Deborah Estrin. *Small data, where n = me.* Communications of the ACM 57(4):32–34, 2014
2 Sohyung Cho, Shihab Asfour, Arzu Onar, Nandita Kaundinya. *Tool breakage detection using support vector machine learning in a milling process.* International Journal of Machine Tools and Manufacture 45(3):241–249, 2005

[1] http://storm.apache.org/
[2] http://spark.apache.org/streaming/

3 Kaiwen Zhang, Vinod Muthusamy, Hans-Arno Jacobsen. *Total order in content-based publish/subscribe systems*. IEEE 32nd International Conference on Distributed Computing Systems, 335–344, 2012
4 Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, Roman Vitenberg. *Distributed event aggregation for content-based publish/subscribe systems*. 8th ACM International Conference on Distributed Event-based Systems, 95–106, 2014
5 Brian Babcock, Chris Olston. *Distributed top-k monitoring*. 2003 ACM SIGMOD International Conference on Management of Data, 28–39, 2003
6 Matteo Migliavacca, Ioannis Papagiannis, David M. Eyers, Brian Shand, Jean Bacon, Peter Pietzuch. *DEFCON: High-Performance Event Processing with Information Security*. 2010 USENIX Annual Technical Conference, 2010.
7 Wil M. P. van der Aalst, Niels Lohmann, Peter Massuthe, Christian Stahl, Karsten Wolf. *Multiparty Contracts: Agreeing and Implementing Interorganizational Processes*. The Computer Journal 53(1):90–106, 2010
8 Walid Fdhila, Conrad Indiono, Stefanie Rinderle-Ma, Manfred Reichert. *Dealing with change in process choreographies: Design and implementation of propagation algorithms*. Information Systems 49:1–24, 2015
9 Ho Soo Lee, Marshall I. Schor. *Match algorithms for generalized Rete networks*. Artificial Intelligence 54(3):249–274, 1992
10 Michael Stonebraker, Ugur Cetintemel, Stan Zdonik. *The 8 requirements of real-time stream processing*. ACM SIGMOD Record 34(4): 42–47, 2005
11 Tim Bass. *Intrusion detection systems and multisensor data fusion*. Communications of the ACM 43(4):99–105, 2000

3.10 Optimisation opportunities

Roman Vitenberg (University of Oslo, NO), Avigdor Gal (Technion – Haifa, IL), Alessandro Margara (University of Lugano, CH), Vinay Setty (MPI für Informatik – Saarbrücken, DE), Martin Ugarte (Free University of Brussels, BE), Matthias Weidlich (HU Berlin, DE), Lijie Wen (Tsinghua University Beijing, CN), and Kaiwen Zhang (TU München, DE)

License Creative Commons BY 3.0 Unported license
© Roman Vitenberg, Avigdor Gal, Alessandro Margara, Vinay Setty, Martin Ugarte, Matthias Weidlich, Lijie Wen, and Kaiwen Zhang

The scope of optimisations in EP and BPM is large and diverse. A common use of EP is as a building block or component or a guiding paradigm within a BPM architecture. On the one hand, process improvements can be based on event data. On the other hand, application feedback and BPM insights can be exploited to optimise and configure implementation of the events processing component.

The state of the art covers a large area in a way that could be further systematised. Some representative works include [1, 18, 17, 6].

Challenge 1: Exploit BPM insight to improve event processing

A growing number of enterprises use complex event processing for monitoring and controlling their operations, while business process models are used to document working procedures. In a recent work [15], a method for optimising complex event processing using business process models was proposed, based on the extraction of behavioural constraints that are used, in turn, to rewrite patterns for event detection, and select and transform execution plans. This work can be extended, for instance, by exploiting constraints derived from data dependencies between activities. For that, the use of declarative process languages, such as DECLARE, can be applied. Also, mining techniques can be used to identify common event patterns, and probabilistic methods can be used to quantify the importance of their appearance as a further optimisation method.

Challenge 2: Exploit process information to improve Quality of Service (QoS) in event processing/monitoring

CEP engines can monitor the execution of processes. In this context, we can envision the exploitation of process-related information to guide the CEP engine and optimise the QoS it provides. For instance, by knowing which information is more relevant for the stakeholder, the CEP engine can prioritise the execution of the rules that derive such information. Similarly, the CEP engine can allocate more resources to the processors that are responsible for extracting the knowledge that is more critical for the process at hand. Finally, in the case of bursts, the monitoring engine can also use process information to selectively discard the data that is less relevant for the process (selective load shedding) to bring back the monitoring system to its normal operating conditions with minimum impact on the quality of the results produced.

This involves the following open subchallenges:

- Extract/Infer Quality of Service (QoS) requirements from the process specifications
- Map the inferred QoS requirements to the CEP queries
- Adapt the CEP engine behaviour online to optimise QoS requirements

Challenge 3: Using Complex Event Processing (CEP) methods to better detect/predict/improve processes at run-time

An important aspect in Business Process Management is the ability to detect anomalies, predict delays and improve process execution policies in real-time. For example, in hospitals where real-time data is available, we would like to detect a re-route of a patient, her waiting time for the next event and the best order in which the patients are to be served. Current techniques for such run-time analysis rely on models (e.g. process models or machine learning models) that are fitted to historical data. However, the updated streams of incoming events provide additional information (beyond the current state of the system). For example, we may find out that a doctor did not arrive to work on the day of the surgery (current state assessment), and that all surgeries so far took longer than usual. Using a set of rules based on CEP, we could invoke an online mechanism (e.g. a heuristic) that would aid with detection prediction and improvement of running process instances [11].

Challenge 4: Optimising distributed EP and business process execution

A distributed orchestration engine for business process execution can be realised using a distributed content-based publish/subscribe system [8]. A service discovery mechanism is provided to automatically perform service composition. By supplying SLAs into the workflows, the architecture can be monitored and elastically provisioned. In this model, different execution engines can decide whether to host a needed activity or not, based on bandwidth, energy, and response time.

A related challenge is to optimise a distributed event-based system informed by business process execution. By identifying common flows, distributed operators can be placed in optimised paths. For instance, multiple closely related processes can be co-located on the same machine. In addition, vertical scaling of overloaded CEP operators can be identified/predicted using workflow analysis.

Challenge 5: Use CEP techniques to diagnose/improve/optimise business process models in offline mode

CEP techniques are most suitable for monitoring, controlling and adapting of business process execution in online mode. This is very common in sensing or web service situations. Nonetheless, in industries, especially for manufacturing companies, the adaption of business process models is not done directly in real time. It is the responsibility of business managers to diagnose/improve/optimise business process models manually in offline mode. However, CEP techniques can still help to provide insights into improvement chances of business process models upon the huge amount of events recorded during the execution of business process instances. E.g., if a former task A was executed by Tom and the latter task B was executed by Peter, the duration of the process instance could be very large. If so, it should be recommended that one resource assignment rule should be added to prevent Tom and Peter from executing A and B successively. The most important issue in this context is generating such recommended assignment rules in a systematic and automatic way. For related work, see [5, 7, 13].

Challenge 6: Distributed query optimisation in BPM, efficient implementation of non-functional properties in decentralised BPM

A number of core issues in the event processing area such as query optimisation, performance, and non-functional properties (reliability [14], adaptivity [9], prioritisation) are currently beyond the mainstream focus of BPM. All those issues could be useful for BPM, however. For example, CEP considers the fact that event streams originate at different sources and tries to place different operators on different nodes. Such a distributed query optimisation could improve the performance of BPM.

Challenge 7: Resource allocation for event-based architectures

A number of business processes involve running event-based architectures in the cloud. An important optimisation challenge in this context is an adaptive allocation of servers and bandwidth in the cloud that optimises QoS and reduces the use of resources and monetary cost. It was proposed in [12] how to perform such an allocation for a pub/sub architecture supporting social notifications from Spotify.

Challenge 8: Align aspects of declarative languages found in CEP and in BPM

The task of defining a unified framework is still a work in progress for both BPM and CEP (see [16] and [3], respectively). At the core of this task lies the problem of having an expressive declarative language that at the same time allows for efficient evaluation. Although the concept of evaluation is different in the two contexts (event detection in CEP and model compliance in BPM), there are certainly shared aspects that have yet to be understood. For example, the study of CEP has derived some formal models that allow to understand the efficient evaluation of operators (e.g. [2] and [4]). These operators are also present in BPM (e.g. the "followed by" operator), and therefore a knowledge transfer could lead to new optimisations in Business Processes. Conversely, BPM models constantly have to deal with the problem of making model definition accessible to end users, and therefore the community has developed graphics representations (e.g. [10]) that could be used in CEP to allow for simple definition of patterns.

References

1 Irene Barba, Andreas Lanz, Barbara Weber, Manfred Reichert, and Carmelo Del Valle. *Optimized Time Management for Declarative Workflows*, pages 195–210. Springer Berlin Heidelberg, 2012.
2 Gianpaolo Cugola and Alessandro Margara. Tesla: A formally defined event specification language. In *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*, pages 50–61. ACM, 2010.
3 Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3):15:1–15:62, June 2012.
4 Yanlei Diao, Neil Immerman, and Daniel Gyllstrom. Sase+: an agile language for Kleene closure over event streams. 2007.
5 Dirk Fahland and Wil M. P. van der Aalst. Model repair – aligning process models to reality. *Inf. Syst.*, 47:220–243, 2015.
6 G. Hermosillo, L. Seinturier, and L. Duchien. Using complex event processing for dynamic business process adaptation. In *Services Computing (SCC), 2010 IEEE International Conference on*, pages 466–473, 2010.
7 T. Jin, J. Wang, Y. Yang, L. Wen, and K. Li. Refactor business process models with maximized parallelism. *IEEE Transactions on Services Computing*, 9(3):456–468, 2016.
8 Guoli Li, Vinod Muthusamy, and Hans-Arno Jacobsen. A distributed service-oriented architecture for business process execution. *ACM Trans. Web*, 4(1):2:1–2:33, January 2010.
9 Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Distributed event aggregation for content-based publish/subscribe systems. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS'14, pages 95–106. ACM, 2014.
10 Maja Pesic, Helen Schonenberg, and Wil M. P. van der Aalst. Declare: Full support for loosely-structured processes. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*, pages 287–, 2007.
11 Arik Senderovich, Andreas Rogge-Solti, Avigdor Gal, Jan Mendling, and Avishai Mandelbaum. The ROAD from sensor data to process instances via interaction mining. In *Advanced Information Systems Engineering – 28th International Conference, CAiSE 2016, Ljubljana, Slovenia, June 13-17, 2016. Proceedings*, pages 257–273, 2016.
12 Vinay Setty, Roman Vitenberg, Gunnar Kreitz, Guido Urdaneta, and Maarten van Steen. Cost-effective resource allocation for deploying pub/sub on cloud. In *IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014, Madrid, Spain, June 30 – July 3, 2014*, pages 555–566, 2014.
13 Rob J. B. Vanwersch, Khurram Shahzad, Irene Vanderfeesten, Kris Vanhaecht, Paul Grefen, Liliane Pintelon, Jan Mendling, Godefridus G. van Merode, and Hajo A. Reijers. A critical evaluation and framework of business process improvement methods. *Business & Information Systems Engineering*, 58(1):43–53, 2016.
14 Marco Volz, Boris Koldehofe, and Kurt Rothermel. Supporting strong reliability for distributed complex event processing systems. In *Proceedings of the 2011 IEEE International Conference on High Performance Computing and Communications*, HPCC'11, pages 477–486. IEEE Computer Society, 2011.
15 Matthias Weidlich, Holger Ziekow, Avigdor Gal, Jan Mendling, and Mathias Weske. Optimizing event pattern matching using business process models. *IEEE Trans. Knowl. Data Eng.*, 26(11):2759–2773, 2014.
16 Mathias Weske. *Business Process Management: Concepts, Languages, Architectures.* Springer, 2007.

17 Gregor Zellner. A structured evaluation of business process improvement approaches. *Business Process Management Journal*, 17(2):203–237, 2011.
18 Haopeng Zhang, Yanlei Diao, and Neil Immerman. On complexity and optimization of expensive queries in complex event processing. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD'14, pages 217–228. ACM, 2014.

3.11 Event Data Quality

Kaiwen Zhang (TU München, DE), Alexander Artikis (NCSR Demokritos – Athens, GR), Anne Baumgraß (Synfioo – Potsdam, DE), Avigdor Gal (Technion – Haifa, IL), Mohammad Sadoghi Hamedani (Purdue University, US), and Stefan Schulte (TU Wien, AT)

License Ⓒ Creative Commons BY 3.0 Unported license
© Kaiwen Zhang, Alexander Artikis, Anne Baumgraß, Avigdor Gal, Mohammad Sadoghi Hamedani, and Stefan Schulte

Traditional business process modelling assumes perfect knowledge of all process executions, as well as perfect sources of event data. This creates a mismatch between the plan and the execution, which is affected by sources of uncertainties, such as unreliable sensors, traffic patterns, weather, etc.

We seek to capture the complexity of data quality, integrate it in business process modelling, and extract insights for concrete decision-making.

Figure 2 illustrates our top-down approach to enriching business process modelling using data quality. Two components must be supplied: the standard business process model, with perfect knowledge and quality, and the specifications for data quality. Combining these together form a data quality-aware model, which can then be analysed to gain additional insights on the business.

The first challenge is defining sources of uncertainties, assessing their quality, and capturing this information in a complete specification format. Sources include the reliability of the event streams (e.g. ordering, missing events, corrupted events, duplication, latency), consistency of correlated events (e.g. the same vehicle reported at two different locations simultaneously), integration issues when fusing various sets of data, or dealing with subjective processes

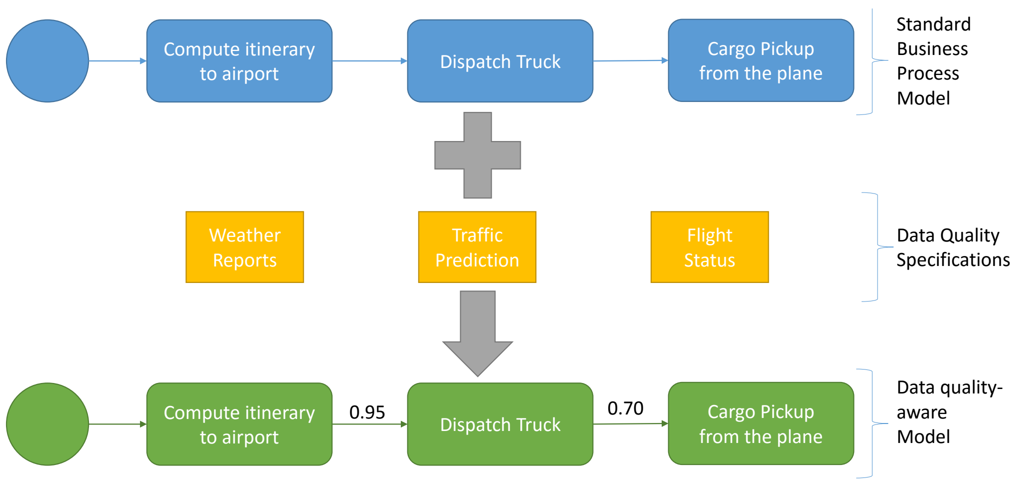

Figure 2 Event Data Quality.

(processes involving human factor). This type of information must be captured in a format which can be easily read and parsed. For instance, a possibility would be to provide contracts that each component abides to in XML.

The second challenge is to enrich the standard provided model with the data quality specifications into a transformed data quality-aware business process model. Currently, there are no such known models. We envision tools such as probabilistic and possibilistic graphical models to be adapted to represent processes with probabilistic transitions, depending on the level of uncertainty. Fuzzy set notation can be used to assign multiple properties to objects that handled by uncertain sources. For instance, a package inspected by customs could be 30% toy, 70% electronic, which would dictate its processing flow.

The third challenge is to study the quality-aware model and decision making. This includes worst-case/average-case analysis, model refinement, and improving the reliability of the event processing system employed (with the associated cost clearly defined).

3.12 From Event Streams to Process Models and Back

Holger Ziekow (FH Furtwangen, DE), Jean Bacon (University of Cambridge, GB), Claudio Di Ciccio (Wirtschaftsuniversität Wien, AT), David Eyers (University of Otago, NZ), Boris Koldehofe (TU Darmstadt, DE), Oliver Kopp (Universität Stuttgart, DE), Agnes Koschmider (KIT – Karlsruher Institut für Technologie, DE), Pnina Soffer (Haifa University, IL), Wei Song (Nanjing University of Science & Technology, CN), and Jan Sürmeli (HU Berlin, DE)

License Creative Commons BY 3.0 Unported license
© Holger Ziekow, Jean Bacon, Claudio Di Ciccio, David Eyers, Boris Koldehofe, Oliver Kopp, Agnes Koschmider, Pnina Soffer, Wei Song, and Jan Sürmeli
Joint work of Jean Bacon, Claudio Di Ciccio, David Eyers, Annika Hinze, Arno Jacobsen, Boris Koldehofe, Oliver Kopp, Agnes Koschmider, Pnina Soffer, Wei Song, Jan Sürmeli, Lucineia Heloisa Thom, Lijie Wen, Holger Ziekow

Business process models typically capture processes at a high level of abstraction. In contrast, the logic for processing event streams is typically defined at a lower and more technical level (e.g. in the form of CEP rules). Specifically, process models operate on the granularity of activities within a process (e.g. the shipment of a good), whereas event processing operates on low-level input events that relate to the process (e.g. observing an RFID tag by a certain reader). We addressed the gap that exists between these abstraction levels of business process models and complex event processing rules by highlighting challenge areas where a combination of CEP and BPM could be beneficial. We identified the four challenge areas depicted in Fig. 3. Our discussions revealed the need for establishing a common ground of formalisms that unify common concepts from the worlds of CEP and BPM. This common ground was seen as a key enabler for all discussed challenge areas of combining CEP and BPM, and for bridging the gap between conceptual models and implementation.

We identified the following main goals, key challenges and opportunities:

Deriving CEP Rules from Process Models (short-term):

Main goal: Automate generation of CEP rules for process monitoring and control
Key challenge: Process models do not provide all the information that may be exploited in rule generation
Opportunity: Augment process models with CEP constructs for more precise definitions of events and decision gates

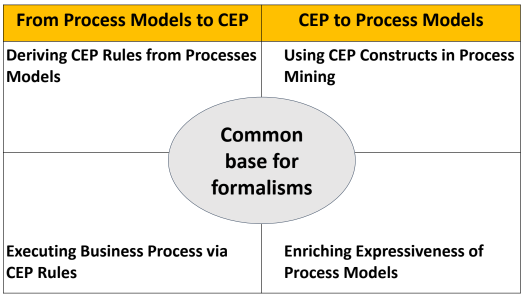

Figure 3 Key challenge areas.

Using CEP constructs for process mining (mid-term):

Main goal: Get from low level events (e.g. sensor data) to a process model
Key challenge: Applying process mining techniques on low-level events
Opportunity: Use CEP constructs to identify higher level activities from event logs

Enriching Expressiveness of Process Models (mid-term):

Main goal: Bridge the gap between conceptual models and implementation
Key challenge: Events that steer processes are too weakly defined in process models
Opportunity: Use CEP constructs to define events in process models more precisely

Executing Business Processes via CEP Rules (long-term):

Main goal: Seamless scalability, adaptiveness, context awareness, and distributed process execution
Key challenge: Transformation of imperative and conceptual process models into CEP rules
Opportunity: Support definitions of highly abstract activities in flexible processes with CEP constructs

Common basis for formalism (long-term):

Main goal: A meta-model for both processes and events
Key challenge: Find fundamental rules to express both control-flow and complex event patterns
Opportunity: ECA rules may inspire us

4 Tutorials

4.1 Managing Streams of Events: An Overview

Alessandro Margara (University of Lugano, CH)

License Creative Commons BY 3.0 Unported license
© Alessandro Margara

Many modern software applications involve processing, analysing, and reacting to potentially large volumes of streaming data. Examples of such applications include monitoring systems, decision support systems, financial analysis tools, and traffic control systems. Designing and implementing these applications is difficult. Indeed, the streaming nature of the data demands for efficient algorithms, techniques, and infrastructures to analyse the incoming information on the fly and extract relevant knowledge from it. To solve this problem, several event and stream processing systems have been proposed in the last years, both from the academia and from the industry. These systems typically provide high level languages to specify how to interpret and transform the input data to produce the relevant results, and hide data distribution and communication for efficient distributed computation. This tutorial overviews the state-of-the-art proposals for stream and event processing. Given the heterogeneity of the proposed solutions, the tutorial introduces a number of models that isolate and analyse the various design choices behind such systems, and discusses the implications of such choices.

4.2 Business Process Management: Concepts, Models, Events

Mathias Weske (Hasso-Plattner-Institut – Potsdam, DE)

License Creative Commons BY 3.0 Unported license
© Mathias Weske

Business process management covers a broad area, ranging from management topics like process improvement to the formal investigation of behavioural properties to implementation aspects related to process automation. In this talk, basic concepts in business process management are introduced. The BPM lifecycle organises major activities, including process design, automation, and enactment. Models play a key role in this field because they allow us to investigate processes, to argue about them, and to achieve a shared understanding. The syntax and semantics of process models are discussed using the BPMN industry standard. It is shown that events are a crucial aspect in the execution semantics of business processes. The talk concludes with a presentation of a process execution environment including a process engine that controls the execution of process instances.

4.3 Management, Utilisation, and Analysis of Instance Data in Distributed Process Settings

Stefanie Rinderle-Ma (Universität Wien, AT)

License Creative Commons BY 3.0 Unported license
© Stefanie Rinderle-Ma
Main reference W. Fdhila, C. Indiono, S. Rinderle-Ma, M. Reichert, "Dealing with change in process choreographies: Design and implementation of propagation algorithms", Inf. Syst., 49:1–24, 2015.
URL http://dx.doi.org/10.1016/j.is.2014.10.004

Distributed process settings are a means to describe, implement, and execute business networks in a process-oriented fashion. Describing and setting up such processes is already a challenging task. When enacting and executing them a multitude of additional challenges arise. This tutorial outlines a selection of challenges and possible solutions; they range from more basic and technical ones such as correlation to advanced utilisation and analysis of data emitted during choreography execution, for example, for distributed compliance checking and prediction of change effects in choreographies. The challenges and solutions are illustrated by means of use cases and projects from different domains such as manufacturing and energy.

4.4 Event Processing

Alejandro P. Buchmann (TU Darmstadt, DE)

License Creative Commons BY 3.0 Unported license
© Alejandro P. Buchmann

Event processing and push-based systems are at the core of a broad range of applications. At the same time the development of application systems has lagged because of the difficulty for users to deal with a new programming paradigm and several open issues regarding non-functional properties ranging from security, privacy and trust to the management of large distributed and decentralised systems.

Starting from some examples of interesting applications we will identify open issues that must be overcome, discuss ideas for integrating event processing with other architectures and programming styles, and will look at event processing as a key element for self-adaptive systems.

5 Overview of Talks

5.1 Online Learning for Complex Event Recognition

Alexander Artikis (NCSR Demokritos – Athens, GR)

License Creative Commons BY 3.0 Unported license
© Alexander Artikis
Joint work of A. Artikis, V. Micheloudakis, A. Skarlatidis, G.Paliouras

Complex event recognition is characterised by uncertainty and relational structure. Markov Logic Networks (MLN)s is a state-of-the-art Statistical Relational Learning framework that can naturally be applied to domains governed by these characteristics. We present OSLα – an online structure learner for MLNs that exploits an Event Calculus axiomatisation in order to constrain the space of possible structures. Learning MLNs from data streams is challenging, as their relational structure increases the complexity of the learning process. In addition, due to the dynamic nature of event recognition applications, it is desirable to incrementally learn or revise the complex event definitions' structure and parameters. Our empirical analysis on real and synthetic data showed that OSLα learns complex event definitions orders of magnitude faster than OSL, the structure learning algorithm that it extends. Moreover, OSLα outperforms event recognition based on manual rules, and, in some cases, weighted manual rules.

References

1 V. Micheloudakis, A. Skarlatidis, G. Paliouras and A. Artikis. *OSLa: Online Structure Learning using Background Knowledge Axiomatization*. Proceedings of European Conference on Machine Learning (ECML), 2016

5.2 Smart Logistics in Practice – Using Event Processing for Comprehensive Transportation Monitoring

Anne Baumgraß (Synfioo – Potsdam, DE)

License Creative Commons BY 3.0 Unported license
© Anne Baumgraß
Joint work of Anne Baumgraß, Andreas Meyer, Marian Pufahl
URL http://synfioo.com

Synfioo predicts and visualises delay times in intermodal transports due to external disruptions such as strikes, tunnel closures and bad weather, using sources like social media, open and closed API, websites and telematics. Synfioo gives shippers, truckers and transport planners hours of advance warning, leading to significant savings in an industry where minor supply chain disruptions can cost millions.

The Synfioo GmbH results from the EU project "GET Service" for green logistics with 9 partners from industry and research (2012–2015). The technical background are both business process management and complex event processing. Furthermore, Synfioo implements machine learning techniques for its predictions.

5.3 Predictive Task Monitoring: Processing Flight Events to Foresee Diversions

Claudio Di Ciccio (Wirtschaftsuniversität Wien, AT)

License Creative Commons BY 3.0 Unported license
© Claudio Di Ciccio
Joint work of Claudio Di Ciccio, Han van der Aa, Cristina Cabanillas, Jan Mendling, Johannes Prescher, Anne Baumgraß
Main reference C. Di Ciccio, H. van der Aa, C. Cabanillas, J. Mendling, J. Prescher, "Detecting flight trajectory anomalies and predicting diversions in freight transportation," Decision Support Systems, 88:1–17, 2016.
URL http://dx.doi.org/10.1016/j.dss.2016.05.004

Identifying flight diversions in a timely manner is a crucial aspect of efficient multi-modal transportation. When an aeroplane diverts, logistics providers must promptly adapt their transportation plans in order to ensure proper delivery despite such an unexpected event. In practice, the different parties in a logistics chain do not exchange real-time information related to flights. This calls for a means to detect diversions that just requires publicly available data, thus being independent of the communication between different parties. The dependence on public data results in a challenge to detect anomalous behaviour without knowing the planned flight trajectory. Our work addresses this challenge by introducing a prediction model that processes events bearing only information on an aeroplane's position, velocity, and intended destination. This information is used to distinguish between regular and anomalous behaviour. When an aeroplane displays anomalous behaviour for an extended period of time, the model predicts a diversion. A quantitative evaluation shows that this approach is able to detect diverting aeroplanes with excellent precision and recall even without knowing planned trajectories as required by related research. By utilising the proposed prediction model, logistics companies gain a significant amount of response time for these cases.

References

1 Claudio Di Ciccio, Han van der Aa, Cristina Cabanillas, Jan Mendling, and Johannes Prescher. Detecting flight trajectory anomalies and predicting diversions in freight transportation. *Decision Support Systems*, 88:1–17, August 2016.

2 Cristina Cabanillas, Claudio Di Ciccio, Jan Mendling, and Anne Baumgrass. Predictive task monitoring for business processes. In Shazia Wasim Sadiq, Pnina Soffer, and Hagen Völzer, editors, *BPM*, volume 8659 of *Lecture Notes in Computer Science*, pages 424–432. Springer, September 2014.

3 Anne Baumgraß, Mirela Botezatu, Claudio Di Ciccio, Remco Dijkman, Paul Grefen, Marcin Hewelt, Jan Mendling, Andreas Meyer, Shaya Pourmirza, and Hagen Völzer. Towards a methodology for the engineering of event-driven process applications. In Manfred Reichert and A. Hajo Reijers, editors, *BPM Workshops*, volume 256 of *Lecture Notes in Business Information Processing*, pages 501–514. Springer International Publishing, 2016.

4 Anne Baumgrass, Cristina Cabanillas, and Claudio Di Ciccio. A conceptual architecture for an event-based information aggregation engine in smart logitics. In Jens Kolb, Henrik Leopold, and Jan Mendling, editors, *EMISA*, volume 248 of *Lecture Notes in Informatics (LNI)*, pages 109–123. GI, September 2015.

5.4 Smart Landscape: The Rugged Internet of Things

Annika M. Hinze (University of Waikato, NZ)

License Ⓒ Creative Commons BY 3.0 Unported license
© Annika M. Hinze
Joint work of Annika Hinze, Chris Griffiths, Judy Bowen, Vimal Kumar, David Bainbridge

Different to smart city proposals, we target the harder problem of a 'smart landscape' in which disconnectedness due to remote locations and harsh operating conditions are common. This talk introduced the issues encountered when using IoT and event processing to help support safer working environments in hazardous work contexts. It particularly highlighted the need for identifying event patterns that are not known *a priori* as no historic event logs exist and dedicated observation of accidents is not a viable option. We have begun to explore the options of combining process approaches in combination with complex event processing techniques.

5.5 BPM in Cloud Architectures: Enabling the Internet of Things Through Effective Business Processes Management with Events

Hans-Arno Jacobsen (TU München, DE)

License Creative Commons BY 3.0 Unported license
© Hans-Arno Jacobsen
Main reference G. Li, V. Muthusamy, H.-A. Jacobsen., "A Distributed Service Oriented Architecture for Business Process Execution", ACM Transactions on the Web, 4(1)2:1–2:33, January 2010
URL http://dx.doi.org/10.1145/1658373.1658375

In today's cloud-based enterprise systems, many business processes rely on service-level agreements (SLAs) to manage interactions with partners and suppliers. SLAs determine revenue, cost and customer satisfaction, but implementing and monitoring SLAs is often a manual and error-prone effort. Companies struggle with how to express, track, verify, manage, and enforce SLAs. This is further exasperated by our rapidly increasing reliance on "everything connected" to track supply and demand across global supply chains.

This talk presents a powerful enterprise process management architecture that manages SLAs across the entire supply chain and enterprise system life-cycle. Our approach leverages events available at every layer of the enterprise software systems stack to efficiently manage business process and interactions. Questions such as the following are addressed: Where is the value in real-time process monitoring and how does it work? Which technologies and design patterns are most effective for monitoring SLAs in real-time? What run-time adaptation and performance optimisations are practical to implement in business processes? What architecture can enable the above?

This talk is based on findings resulting from our PADRES Events & Services Bus (http://padres.msrg.org) and eQoSystem (http://eQoSystem.msrg.org) research projects.

5.6 D2Worm – A Management Infrastructure for Distributed Data-centric Workflows

Martin Jergler (TU München, DE)

License Creative Commons BY 3.0 Unported license
© Martin Jergler
Joint work of Mohammad Sadoghi, Hans-Arno Jacobsen
Main reference M. Jergler, M. Sadoghi, H.-A. Jacobsen, "D2WORM: A Management Infrastructure for Distributed Data-centric Workflows", in Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD'15), pp. 1427–1432, ACM, 2015.
URL https://dx.doi.org/10.1145/2723372.2735362

In this talk, we revisit D2WORM, a Distributed Data-centric Workflow Management system. D2WORM allows users to (1) graphically model data-centric workflows based on the Guard-Stage-Milestone (GSM) meta-model, (2) automatically compile the modelled workflow into several fine-grained workflow units (WFUs), and (3) deploy these WFUs on distributed infrastructures. A WFU is a system component that manages a subset of the workflow's data model and, at the same time, represents part of the global control flow by evaluating conditions over the data. WFUs communicate with each other over a publish/subscribe messaging infrastructure that allows the architecture to scale from a single node to dozens of machines distributed over different data-centers.

5.7 Explicit Subscription for Enabling Event Buffering

Sankalita Mandal (Hasso-Plattner-Institut – Potsdam, DE) and Jan Sürmeli (HU Berlin, DE)

License Ⓒ Creative Commons BY 3.0 Unported license
© Sankalita Mandal and Jan Sürmeli

Catching an event from inside a business process requires the prior subscription to some event producer such as a CEP-engine or a pub/sub broker. Between the moment of subscription and the moment of catching, multiple (possibly complex) events can occur. Current modelling languages for business processes only allow an abstract specification of event catching, but lack the notions to specify:

1. The subscription to specific, possibly complex, events from specific event producers.
2. The way multiple occurrences are handled and the order in which events are caught.

We propose to extend BPMN in order to overcome these shortcomings, centred on the explicit notion of a buffer: A buffer holds the (possibly complex) events that occur between subscription and catching, and allows the retrieval of its contents based on a buffer policy.

5.8 Associative Composition of Stream Processing Processes

Wolfgang Reisig (HU Berlin, DE)

License Ⓒ Creative Commons BY 3.0 Unported license
© Wolfgang Reisig

Large stream based systems as well as (business) processes are usually composed of more elementary components. So, composition is a fundamental structuring principle of such systems and processes.

A system or a process is in general composed of many components. Examples include supply chains or systems with components that are composed along with an adapter. Composition is assumed to be associative in these cases, i.e. for components C_i it is assumed that $(C_1 + C_2) + C_3 = C_1 + (C_2 + C_3)$. This, however, requires a careful definition of components and their composition. We suggest quite a general such setting.

Based on the observation that a component frequently has two partner components in a composed system, we suggest the interface of a component to consist of two ports, with each port consisting of any set of labeled elements. Composition $C_1 + C_2$ is then defined by "gluing" elements of the right port of C_1 with equally labelled elements of the left port of C_2.

As an application we consider a more general version of van der Aalst' workflow nets, where in $C_1 + C_2$, some events of C_2 may already have occurred while some events of C_1 have not yet started. We show that this composition retains the important property of soundness.

5.9 RichNote: Adaptive Selection and Delivery of Rich Media Notifications to Mobile Users

Roman Vitenberg (University of Oslo, NO)

License Creative Commons BY 3.0 Unported license
© Roman Vitenberg

In recent years, notification services for social networks, mobile apps, messaging systems and other electronic services have become truly ubiquitous. When a new content item becomes available, the service sends an instant notification to the user. When the content is produced in massive quantities, and it includes both large-size media and a lot of meta-information, it gives rise to a major challenge of selecting content to notify about and information to include in such notifications. We tackle three important challenges in realising rich notification delivery: (1) content and presentation utility modelling, (2) notification selection and (3) scheduling of delivery. We consider a number of progressive presentation levels for the content. Since utility is subjective and hard to model, we rely on real data and user surveys. We model the content utility by learning from large-scale real world data collected from the Spotify music streaming service. For the utility of the presentation levels we rely on user surveys. Blending these two techniques together, we derive utility of notifications with different presentation levels. We then model the selection and delivery of rich notifications as an optimisation problem with a goal to maximise the utility of notifications under resource budget constraints. We validate our system with large-scale simulations driven by the real-world de-identified traces obtained from Spotify. With the help of several baseline approaches we show that our solution is adaptive and resource efficient.

5.10 Real-time Explorative Event-based Systems

Mohammad Sadoghi Hamedani (Purdue University, US)

License Creative Commons BY 3.0 Unported license
© Mohammad Sadoghi Hamedani

This talk covered formulating subscription as natural language, and subsequently, translating the imprecise natural language formulation to a precise query language (such as SQL or complex-events) followed by applying query/subscription expansion techniques to find all relevant (both historic/recent) publications in order to faithfully capture user's intention.

5.11 The ROAD from Sensor Data to Process Instances via Interaction Mining

Arik Senderovich (Technion – Haifa, IL)

License Creative Commons BY 3.0 Unported license
© Arik Senderovich
Joint work of Arik Senderovich, Andreas Rogge-Solti, Avigdor Gal, Jan Mendling, Avishai Mandelbaum
Main reference A. Senderovich, A. Rogge-Solti, A. Gal, J. Mendling, A. Mandelbaum, "The ROAD from Sensor Data to Process Instances via Interaction Mining", in Proc. of the 28th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'16), LNCS, Vol. 9694, pp. 257–273, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-319-39696-5_16

Process mining is a rapidly developing field that aims at automated modelling of business processes based on data coming from event logs. In recent years, advances in tracking technologies, e.g., Real-Time Locating Systems RTLS), put forward the ability to log business process events as location sensor data. To apply process mining techniques to such sensor data, one needs to overcome an abstraction gap, because location data recordings do not relate to the process directly. In this work, we solve the problem of mapping sensor data to event logs based on process knowledge. Specifically, we propose interactions as an intermediate knowledge layer between the sensor data and the event log.

We solve the mapping problem via optimal matching between interactions and process instances. An empirical evaluation of our approach shows its feasibility and provides insights into the relation between ambiguities and deviations from process knowledge, and accuracy of the resulting event log.

5.12 Cost-Effective Resource Allocation for Deploying Pub/Sub on Cloud

Vinay Setty (MPI für Informatik – Saarbrücken, DE), Guido Urdaneta, Gunnar Kreitz, and Maarten van Steen

License Creative Commons BY 3.0 Unported license
© Vinay Setty, Guido Urdaneta, Gunnar Kreitz, and Maarten van Steen
Main reference V. Setty, R. Vitenberg, G. Kreitz, G. Urdaneta, M. van Steen, "Cost-Effective Resource Allocation for Deploying Pub/Sub on Cloud," in Proc. of the IEEE 34th International Conference on Distributed Computing Systems (ICDCS'14), pp. 555–566, IEEE CS, 2014.
URL http://dx.doi.org/10.1109/ICDCS.2014.63

Publish/subscribe (pub/sub) is a popular communication paradigm in the design of large-scale distributed systems. A fundamental challenge in deploying pub/sub systems on a data center or a cloud infrastructure is efficient and cost-effective resource allocation that would allow delivery of notifications to all subscribers. In this work we addressed the answers to the following three fundamental questions: Given a pub/sub workload, (1) what is the minimum amount of resources needed to satisfy all the subscribers, (2) what is a cost-effective way to allocate resources for the given workload, and (3) what is the cost of hosting it on a public Infrastructure-as-a-Service (IaaS) provider like Amazon EC2.

We evaluate the solution experimentally using real traces from Spotify and Twitter along with a pricing model from Amazon. Using a variety of practical scenarios for each dataset, we also show that our solution scales well for millions of subscribers and runs fast.

5.13 Challenges of Data Integration in Cross-Organisational Processes

Sergey Smirnov (SAP SE – Walldorf, DE)

License Creative Commons BY 3.0 Unported license
© Sergey Smirnov

Modern logistic companies enjoy a broad choice of IT products that can automate their business processes. The product examples include telematics solutions, software applications enabling communication within a company, and applications enabling integration with partners. However, small and medium enterprises often remain very conservative: they run their processes on disintegrated proprietary IT solutions, if any. This talk described the technical and organisational challenges that logistics companies face when they integrate their processes with other businesses.

5.14 Repairing Event Logs

Wei Song (Nanjing University of Science & Technology, CN) and Hans-Arno Jacobsen (TU München, DE)

License Creative Commons BY 3.0 Unported license
© Wei Song and Hans-Arno Jacobsen
Main reference W. Song, X. Xia, H.-A. Jacobsen, P. Zhang, H. Hu, "Efficient alignment between event logs and process models", IEEE Transactions on Services Computing, PP(99):1–1, 2016.
URL http://dx.doi.org/10.1109/TSC.2016.2601094

The aligning of event logs with process models is of great significance for process mining to enable conformance checking, process enhancement, performance analysis, and trace repairing. Since process models are increasingly complex and event logs may deviate from process models by exhibiting redundant, missing, and dislocated events, it is challenging to determine the optimal alignment for each event sequence in the log, as this problem is NP-hard. Existing approaches utilise the cost-based A^* algorithm to address this problem. However, scalability is often not considered, which is especially important when dealing with industrial-sized problems. In this paper, by taking advantage of the structural and behavioural features of process models, we present an efficient approach which leverages effective heuristics and trace replaying to significantly reduce the overall search space for seeking the optimal alignment. We employ real-world business processes and their traces to evaluate the proposed approach. Experimental results demonstrate that our approach works well in most cases, and that it outperforms the state-of-the-art approach by up to 5 orders of magnitude in runtime efficiency.

References

1. Arya Adriansyah, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. Conformance checking using cost-based fitness analysis. In *Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOC'11, Helsinki, Finland, August 29 – September 2*, pages 55–64, 2011.
2. Massimiliano de Leoni, Fabrizio Maria Maggi, and Wil M. P. van der Aalst. Aligning event logs and declarative process models for conformance checking. In *Business Process Management – 10th International Conference, BPM'12, Tallinn, Estonia, September 3-6. Proceedings*, pages 82–97, 2012.
3. Massimiliano de Leoni and Wil M. P. van der Aalst. Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear

programming. In *Business Process Management – 11th International Conference, BPM'13, Beijing, China, August 26-30. Proceedings*, pages 113–129, 2013.
4 Jorge Munoz-Gama, Josep Carmona, and Wil M. P. van der Aalst. Conformance checking in the large: Partitioning and topology. In *Business Process Management – 11th International Conference, BPM'13, Beijing, China, August 26-30. Proceedings*, pages 130–145, 2013.
5 Andreas Rogge-Solti, Ronny Mans, Wil M. P. van der Aalst, and Mathias Weske. Improving documentation by repairing event logs. In *The Practice of Enterprise Modeling – 6th IFIP WG 8.1 Working Conference, PoEM'13, Riga, Latvia, November 6-7, Proceedings*, pages 129–144, 2013.
6 Wei Song, Xiaoxu Xia, Hans-Arno Jacobsen, Pengcheng Zhang, and Hao Hu. Heuristic recovery of missing events in process logs. In *IEEE International Conference on Web Services, ICWS'15, New York, USA, Jun 27-July 2*, pages 105–112, 2015.
7 Wil M. P. van der Aalst, Arya Adriansyah, and Boudewijn F. van Dongen. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 2(2):182–192, 2012.
8 Jianmin Wang, Shaoxu Song, Xiaochen Zhu, and Xuemin Lin. Efficient recovery of missing events. *PVLDB*, 6(10):841–852, 2013.

5.15 Efficient Handling of Out-of-Order Events

Jan Sürmeli (HU Berlin, DE)

License Creative Commons BY 3.0 Unported license
© Jan Sürmeli
Joint work of Dirk Fahland, Jan Sürmeli, Matthias Weidlich

A CEP system produces a stream of complex events from an input stream of events based on a query. The distribution of event producers raises the problem that the arrival order deviates from the production order of events: The input stream of a CEP system is thus not inherently ordered by time stamps. Purely adhering to the arrival order leads to a deviation between the computed and the desired output stream of complex events: Some complex events in the output stream are incorrect, others are missing. The problem is to achieve *correctness* and *completeness* while preserving low latency and feasibility for real time applications. One approach is that of *aggressive query evaluation*: Upon the detection of an out-of-order event, one invalidates and delivers the incorrect and missing complex events, respectively. Our research focuses on improving this approach by analysing the query at design time, and applying the analysis results to increase the performance at runtime.

5.16 Research Issues on the Extraction of Process Models from Natural Language Text

Lucinéia Heloisa Thom (Federal University of Rio Grande do Sul, BR) and Renato César Borges Ferreira

License Creative Commons BY 3.0 Unported license
© Lucinéia Heloisa Thom and Renato César Borges Ferreira
Main reference R. C. Borger Ferreira, L. H. Thom, "Uma Abordagem para Gerar Texto Orientado a Processo a partir de Texto em Linguagem Natural", in Proc. of the XII Brazilian Symp. on Information Systems, pp. 585–588. 2016.
URL http://sbsi2016.ufsc.br/anais/Proceedings%20of%20the%20XII%20SBSI.pdf

In organisations, business process modelling is very important to report, understand and automate processes. Organisations usually have unstructured documents related to their business processes, which can be very difficult to understand by process analysts and developers. The extraction of process models from natural language text may contribute to minimise the effort required during process modelling. This research proposes an approach to generate process–oriented text from text in natural language. In particular, to investigate the structure a text in natural language must present so that process models can be extracted from that. As practical result this proposes the development of an open-source tool to support: (i) the automatic selection of business process relevant information from text in natural language; (ii) the extraction of process models from business process-oriented text.

5.17 Scalable, Expressive Publish/Subscribe Systems

Kaiwen Zhang (TU München, DE), Hans-Arno Jacobsen (TU München, DE), Mohammad Sadoghi Hamedani (Purdue University, US), and Roman Vitenberg (University of Oslo, NO)

License Creative Commons BY 3.0 Unported license
© Kaiwen Zhang, Hans-Arno Jacobsen, Mohammad Sadoghi Hamedani, and Roman Vitenberg

The publish/subscribe paradigm is known for its loosely coupled interactions and event filtering capabilities. Traditional applications using pub/sub systems require large-scale deployment and high event throughput. Thus, pub/sub has always put the emphasis on scalability and performance, to the detriment of filtering expressiveness and quality of service. The matching language is usually limited to topic-based or content-based event filtering and does not allow complex stream-based subscriptions to be expressed. Messages are delivered on a best-effort basis without any ordering or reliability guarantees.

Recently, modern pub/sub applications such as online games, social networks, and sensor networks, have specifications which extend beyond the basic semantics provided by standard systems. Installing additional services and event processing systems at the endpoints can overcome these limitations. However, we argue that such solutions are inefficient and put an avoidable strain on the pub/sub layer itself. Therefore, the focus of our work is to develop integrated solutions to extend pub/sub language expressiveness and quality of service, as well as demonstrate that this approach results in better performance from a holistic perspective. The methodology and technical insights from this line of work can contribute to the integration of event processing and business process management.

References

1 César Cañas, Kaiwen Zhang, Bettina Kemme, Jörg Kienzle, and Hans-Arno Jacobsen. Publish/subscribe network designs for multiplayer games. In Laurent Réveillère, Lucy

Cherkasova, and François Taïani, editors, *Proceedings of the 15th International Middleware Conference, Bordeaux, France, December 8-12, 2014*, pages 241–252. ACM, 2014. doi:10.1145/2663165.2663337.

2 Christoph Doblander, Tanuj Ghinaiya, Kaiwen Zhang, and Hans-Arno Jacobsen. Shared dictionary compression in publish/subscribe systems. In *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*, DEBS'16, pages 117–124, New York, NY, USA, 2016. ACM. doi:10.1145/2933267.2933308.

3 Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Distributed event aggregation for content-based publish/subscribe systems. In Umesh Bellur and Ravi Kothari, editors, *The 8th ACM International Conference on Distributed Event-Based Systems, DEBS'14, Mumbai, India, May 26-29, 2014*, pages 95–106. ACM, 2014. doi:10.1145/2611286.2611302.

4 Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Minimizing the communication cost of aggregation in publish/subscribe systems. In *35th IEEE International Conference on Distributed Computing Systems, ICDCS 2015, Columbus, OH, USA, June 29 – July 2, 2015*, pages 462–473. IEEE Computer Society, 2015. doi:10.1109/ICDCS.2015.54.

5 Kaiwen Zhang, Vinod Muthusamy, and Hans-Arno Jacobsen. Total order in content-based publish/subscribe systems. In *2012 IEEE 32nd International Conference on Distributed Computing Systems, Macau, China, June 18-21, 2012*, pages 335–344. IEEE Computer Society, 2012. doi:10.1109/ICDCS.2012.17.

Participants

- Alexander Artikis
NCSR Demokritos – Athens, GR
- Jean Bacon
University of Cambridge, GB
- Anne Baumgraß
Synfioo – Potsdam, DE
- Alejandro P. Buchmann
TU Darmstadt, DE
- Claudio Di Ciccio
Wirtschaftsuniversität Wien, AT
- David Eyers
University of Otago, NZ
- Avigdor Gal
Technion – Haifa, IL
- Annika M. Hinze
University of Waikato, NZ
- Hans-Arno Jacobsen
TU München, DE
- Martin Jergler
TU München, DE
- Boris Koldehofe
TU Darmstadt, DE
- Oliver Kopp
Universität Stuttgart, DE
- Agnes Koschmider
KIT – Karlsruher Institut für Technologie, DE
- Sankalita Mandal
Hasso-Plattner-Institut – Potsdam, DE
- Alessandro Margara
University of Lugano, CH
- Ken Moody
University of Cambridge, GB
- Cesare Pautasso
University of Lugano, CH
- Wolfgang Reisig
HU Berlin, DE
- Stefanie Rinderle-Ma
Universität Wien, AT
- Mohammad Sadoghi Hamedani
Purdue University, US
- Stefan Schulte
TU Wien, AT
- Arik Senderovich
Technion – Haifa, IL
- Vinay Setty
MPI für Informatik – Saarbrücken, DE
- Jatinder Singh
University of Cambridge, GB
- Sergey Smirnov
SAP SE – Walldorf, DE
- Pnina Soffer
Haifa University, IL
- Wei Song
Nanjing University of Science & Technology, CN
- Jan Sürmeli
HU Berlin, DE
- Lucinéia Heloisa Thom
Federal University of Rio Grande do Sul, BR
- Martin Ugarte
Free University of Brussels, BE
- Stijn Vansummeren
Free University of Brussels, BE
- Roman Vitenberg
University of Oslo, NO
- Matthias Weidlich
HU Berlin, DE
- Lijie Wen
Tsinghua University Beijing, CN
- Mathias Weske
Hasso-Plattner-Institut – Potsdam, DE
- Kaiwen Zhang
TU München, DE
- Holger Ziekow
FH Furtwangen, DE