A Big Data Perspective on Cyber-Physical Systems for Industry 4.0: Modernizing and Scaling Complex Event Processing

Carina Andrade[1][0000-0001-8783-9412]

[1] ALGORITMI Research Centre, University of Minho, Guimarães, Portugal
carina.andrade@dsi.uminho.pt

Abstract. Nowadays, organizations have devices integrated into their business processes and producing data that reflects the events happening in their systems. This data relevancy is widely recognized by the community, but there is no common system architecture that categorizes how and what can be done with this streaming data to maximize its usefulness. This document discusses the work proposed for a doctoral thesis in this research topic, presenting: i) the main goal, objectives and expected contributions; ii) the state-of-the-art that supports the identified problem, describing some system architectures and relating them with the architecture being proposed; iii) the research methodology; iv) and, the logical and technological system architecture and preliminary results of the demonstration case at Bosch Car Multimedia Portugal.

Keywords: Big Data, Complex Event Processing, Cyber-Physical Systems, Rules Engine, Machine Learning.

1 Introduction

Currently, several business sectors are trying to catch up with the Big Data era, namely dealing with multiple data sources, in different formats, and different velocities. With the Internet of Things (IoT) proliferation, organizations can have their business processes complemented with sensors that produce event data contributing to monitoring their processes. Considering the existence of these hardware capabilities, some technologies to handle all the data that is constantly being produced are required, such as Spark[1], Druid[2] or Storm[3], which are of major relevance for processing, aggregating or analyzing streaming data in real time. Besides the technologies needed to handle the streaming data, some technologies for Big Data Warehousing (e.g. Hive[4]) can be relevant to complement the streaming data analysis. Considering these technologies, the system's scalability regarding data processing and storage is guaranteed. However,

[1] https://spark.apache.org/
[2] http://druid.io/
[3] http://storm.apache.org/
[4] https://hive.apache.org/

CEUR-WS.org/Vol-2370/paper-01.pdf

concepts such as Complex Event Processing (CEP) and rule-based technologies (e.g., Drools[5]) are called into this context to allow the processing of different types of events, finding patterns between them and using rules for that, reflecting the business requirements identified in the context of each organization. The integration of these concepts and technologies is already mentioned in some identified works, but it is not considered that they can be complemented with Machine Learning (ML) techniques that will allow the system to make predictions or recommendations using pre-determined ML models over events that arrive at the system. In addition, no other work mentions the importance of a complete and innovative visualization component for monitoring this type of systems, which is being considered in this work.

In this context, the goal of this doctoral thesis supervised by Professors Maribel Yasmina Santos and Carlos Costa, is the proposal of a logical and technological system architecture that provides to organizations the capability of using all their event data in real-time fashion, considering: i) business requirements that should be easily integrated into the system; ii) powerful ways of doing predictions and recommendations to improve daily operations; iii) system self-management and monitoring to prevent uncontrolled growth of the system. As can be observed in Section 2, the idealized system architecture addressing the mentioned points was not identified in the literature review, reason why this work discusses the proposal of a CEP system for the Big Data era.

This document is divided as follows: Section 2 presents the state-of-the-art; Section 3 mentions the expected contributions; Section 4 dissects the research methodology; Section 5 highlights the proposed approach and current results; and, Section 6 summarizes the presented and future work.

2 State-of-the-Art

A CEP system can be described as a system that analyses events through different perspectives like pattern matching or inference. In these processes, the system can filter and aggregate the relevant information, complementing it with external data [1]. Besides the concept itself, [2] presents CEP systems as being a challenge for the Data Stream Management Systems, considering that besides processing the data in real-time, it is necessary to take action over it. The Rapide project [3, 4] is often considered as the first work to explore the CEP concept [1, 5]. This project started in the 90s and provided the capability of identifying the temporal and causal relationship among events. However, the current amount of available data requires improvements to this concept, adapting it to Big Data environments.

In this context, and analyzing the existing architectures that aim to integrate the CEP and Big Data concepts, the work of [6] uses the recognized Lambda and Kappa architectures as the base for the proposal of BiDCEP, an architecture that integrates CEP and Big Data Streaming concepts. A relevant point mentioned by the authors is the relevance of the IoT concept, which should be considered as an enabler for descriptive, predictive and prescriptive analytics, although it is not noticeable where these aspects

[5] https://www.drools.org/

are considered in the proposed architecture. Another architecture is highlighted in the FERARI project context [7], with a prototype for real-time CEP that process vast amounts of event data in a distributed way. These two architectures share some principles, such as the components responsible for the connection to the data sources; the components associated with the event processing; and, the components related to the data consumers.

The main properties that should be considered in the development of a Big Data CEP system are mentioned by [8]: parallelism, elasticity, multi-query and distributed resources. However, the authors also mention that, although there are some works that try to address these issues, the integration of CEP and Big Data technologies is still something significantly unexplored.

Nevertheless, more than integrating CEP in Big Data contexts, some works [9–11] emphasize the relevance of combining Big Data, CEP, and IoT to support the manufacturing industry through Cyber-Physical Systems (CPSs). The work of [9] proposes a framework for a Manufacturing CPS that considers the physical world (e.g., manufacturing facilities and shop-floor resources); the cyber world (e.g., simulation and prediction models); and, the interface between these two worlds (e.g., sensor networks and structured and unstructured data). In this case, the CEP system is a component in the cyber world, responsible for processing events and return results in (near)real-time that could provide operational visibility and awareness for the manufacturing system. In [10], the authors explore the use of event-based predictions for manufacturing planning and control. In this case, sensors installed in the manufacturing plant are the data source for the CEP system that, combining the events with historical data, will provide the possibility of achieving the event-based prediction models for production planning and control. The work of [11] proposes a framework that can be applied to monitor the status of a CPS through the use of IoT data. This framework presents a Publish-Subscribe Messaging System that receives the data for further identification of meaningful events in a rule-based CEP System (running in a distributed way). The processing results are published in the self-healing mechanism and predictive maintenance component for the execution of the actions previously defined.

3 Expected Contributions

Although the main concepts and components of the analyzed architectures are being considered in the architecture to be proposed in this doctoral thesis, current contributions: i) do not clearly and rigorously refer or detail how CEP and Big Data concepts and technologies can act together for distributed data, rules and events processing with (near)real-time aggregations and Key Performance Indicators (KPIs) at data ingestion time; ii) do not combine batch data arriving from a Big Data Warehouse [12], complementing the event data that arrives in a streaming way and bringing more value to the results and actions of the system; iii) do not discuss techniques similar to the ML models lake component (presented in section 5), which can be significantly helpful for patterns discovery and it is identified as a major gap in these systems [13]; and, iv) do not

consider the relevance of monitoring the functioning and evolution of this type of systems that can quickly become untraceable in Big Data contexts.

Beyond the current results presented in Section 5, this work reveals to be of significant relevance to several contexts considering that the proposed system architecture should be generic to be applied in different areas, such as industry, smart cities, agriculture, among others, where several data sources are involved (as it is intended to be shown during this thesis, using different demonstration cases). Consequently, this system can be considerably helpful for the monitoring of business processes and events, also using complementary data, to prevent possible problems through the capabilities of its *Predictors and Recommenders* component, which is directly linked to the actions that could be triggered, components not seen in other identified works. In industrial contexts, the role of this system can be easily identified in the manufacturing shop floor where machines and other interconnected devices are increasing, or even for the analysis of customers' reviews in social media, for example. In this context, this system can, for example, monitor the production data and, if some business rule is activated (e.g., for a defective product), predict if the next products will also be defective products and, if true, stop the production machine. Also, the system can predict and prevent a brand crisis related to an organization's publication or a defective product, which is generating negative comments in social media. In smart cities and agricultural contexts, the goal is fundamentally the same, i.e., use the data that is being produced by several sources and process it in (near)real-time, sending, for example, an accident warning to a street screen or changing the amount of water for irrigation due to a temperature change.

4 Research Methodology

This work follows a design science [14] research approach with the goal of extension of the boundaries of human or organizational capabilities through the creation of new and innovative artifacts, in this case, proposing a solution for organizations pursuing (near)real-time decisions and automated actions based on Big Data streams and CPSs. In this context, [14] presents a set of guidelines to conduct design science research in Information Systems: i) propose an artifact to address an organizational problem; ii) understand the relevance of the problem that is being solved; iii) evaluate the utility, quality and efficacy of the proposed artifact; iv) provide clear and verifiable research contributions; v) apply rigorous methods on the artifact development and evaluation; vi) consider the design of the artifact as a search process utilizing the available means to fulfil the goals and satisfy the constraints; and, vii) present the research to technology practitioners and researchers, as well as management-oriented audiences. Moreover, [14] considers that the Information Technology (IT) artifacts, resulting from a design science research process, can be: constructs (vocabulary and symbols); models (abstractions and representations); methods (algorithms and practices); and, instantiations (implementations and prototype systems).

Therefore, the widely recognized Design Science Research Methodology for Information Systems [15] is used in this doctoral thesis, considering an objective-centred

approach for the designing of a logical and technological system architecture that must meet the following objectives:

1. Handle Big Data produced by several sources inside and outside the organizations (e.g., data from production lines, cars, citizens, smartphones, among others);
   a. Consider several possible data sources and their interfaces' differences, in order to design a system that ensures that new data sources can be easily added;
   b. Consider the volume, variety and velocity of the data that could arrive at the system, in order to define its scalability, multi-query, parallelism and distributed resources characteristics.
2. Consider the business requirements and indicators defined by organizations, besides the data itself;
3. Process the data within the time frame needed for several decision makers, with organizations providing inputs regarding the latency for the different system use cases, as this can be used to evaluate the timeliness of the automated actions triggered by the system;
4. Provide predictions and recommendations for the organization's daily activities;
   a. Design and evaluate an adequate system architecture to efficiently integrate predictive and prescriptive ML models into the streams of data/event processed by the system, providing high throughput and low latency predictions/prescriptions.
5. Autonomously execute adequate actions to avoid considerable problems for the organization (e.g., stop a production machine);
   a. Design and evaluate the most adequate way to communicate with external systems, executing automated actions directly related to the business requirements (rules) and indicators defined by the organization.
6. Consider the relevance of self-management and monitoring, preventing the uncontrolled growth of the system with the constant monitoring and visualization of what happened in the system (e.g., Which producers are introducing more data into the system? What are the most triggered actions?).
   a. Design and evaluate the monitoring system and the visualization platform that should consider strategic endpoints in which data is collected and provide user-friendly analysis of the system status (e.g., immersive and drill-down visualization, virtual or augmented reality).

Considering these objectives, some metrics were identified as relevant for the system evaluation: scalability and complexity when integrating new data sources, producers and consumers in the system; number of events produced/consumed per second; number of rules, actions and ML models verified, triggered and applied per second, respectively; average response time of the application of ML models; and, usefulness of the analyses made available in the system's monitoring platform. These metrics can be evaluated considering the organizations' requirements and/or using baselines and guidelines identified in the literature review (e.g., throughput, latency and scalability benchmarks).

The results of a first iteration (after finishing the second of the four years of the doctoral program) on the Design and Development phase of the research methodology are presented in Section 5, already fulfilling the 1st, 2nd, 3rd and 5th objectives discussed

above through a real-world prototype implementation at Bosch Car Multimedia Portugal, although it still needs a rigorous evaluation.

5 Proposed Approach and Current Results

This section presents the first iteration of the Design and Development phase of the research methodology explained in Section 4, using a proof of concept based on the Active Lot Release application from Bosch Car Multimedia Portugal as a demonstration case. Taking into consideration the organization's needs and the gap found in the literature review, a system architecture is being proposed to fill these needs. This system, named Intelligent Event Broker, aims to represent a Big Data-oriented CEP system that combines a collection of software components and data engineering decisions, integrated to ensure their usefulness, efficiency and harmonious functioning, in order to process the events that arrive in the system.

The proposed architecture for the Intelligent Event Broker (**Fig. 1**) considers a vast number of components for dealing with the volume, variety and velocity of the data:

1. *Source Systems*: the system architecture should be prepared to receive data from several sources: relational, NoSQL or NewSQL databases, IoT Gateways or (Web)Servers, and even components of the Hadoop ecosystem, such as Hive Tables or HDFS files;
2. *Producers*: to ensure that all the *Source Systems* identified in 1. can be integrated into the system, regardless of their communication interfaces, the *Producers* component is proposed to standardize the collection of events entering the system. Kafka[6] (a distributed streaming platform) is proposed for the deployment of this component;
3. *Broker Beans*: the events collected in the *Source Systems* by the Kafka *Producers* are serialized into the form of classes that define the several business entities existing in the system – *Broker Beans*;
4. *Brokers*: events serialized into *Broker Beans,* can be published by the *Producers* into a Kafka topic that is stored in a cluster of Kafka *Brokers*;
5. *Event Processor*: events are subscribed by the *Event Processor* (Kafka Consumers that are embedded into Spark Applications) that are always waiting for processing the events arriving at the system, regardless of their frequency;
6. *Complementary Data*: in addition to the events published in the topics, the *Event Processor* can use *Complementary Data* from the *Source Systems,* if useful for the event processing, providing additional and relevant information;
7. *Rules Engine*: includes the defined rules that represent the Business Requirements (with Strategical, Tactical or Operational rules). This work is usually associated to a Data Engineer that creates the rules that represent the business needs. Here Drools is used to store the rules that will be then translated by the *Event Processor*, using a custom-made integration of Spark and Drools, based on previously explored paths by the technical community [16].

[6] https://kafka.apache.org/

8. *Triggers*: connectors to the *Destination Systems*, execute the actions previously defined for the rules when the condition is evaluated as being true.

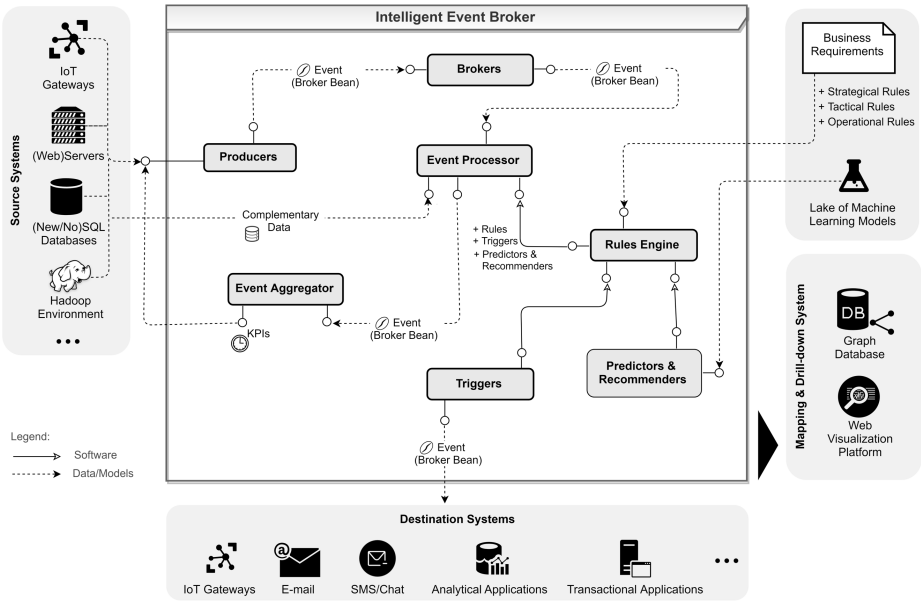

Fig. 1. System Architecture

9. *Destination Systems*: the results of processing the events can be sent to, for example, *IoT Gateways* that can activate an actuator, *Text* or *E-mails Messages,* or even *Transactional* or *Analytical Applications*;
10. *Predictors and Recommenders*: the concept of a *Lake of ML Models*, which are trained beforehand, is proposed in this system architecture as being of major relevance. Allow the application of those ML models to the data that is being processed, providing the capability to predict occurrences or recommend actions based on the events that are arriving at the system;
11. *Event Aggregator*: stores the raw event data (events that arrived at the system) or processed event data (the events processing result, such as results from the *Predictors and Recommenders* component) used to calculate the KPIs relevant to the business. This component is supported by Druid, a columnar storage system useful for aggregating event data at ingestion time [17, 18];
12. *Mapping and Drill-down System*: allows the constant monitoring of the Intelligent Event Broker and includes:
    a. *Graph Database*: stores the relevant metadata of the Intelligent Event Broker, allowing the exploration of the flows of the events in the Intelligent Event Broker;
    b. *Web Visualization Platform*: provides an interactive and immersive visualization regarding the Intelligent Event Broker metadata, stored in the *Graph Database,* taking into account the various implementation contexts of the system.

Currently, a demonstration case was already implemented using data from the Bosch Car Multimedia Portugal plant [19]. This data comes from its ALR System that supports the quality control used in the manufacturing and packaging processes. This system is based on rules that are applied to the products contained into lots before they are shipped to customers. The ALR system provides a stream of events that contain information about the quality control process, being considered, at this point, the lot identification, its packaging date, the production line, and the status ("*Valid*" or "*Invalid*" lot).

Therefore, for this demonstration case, one *Operational Rule* and two *Tactical Rules* were defined, as well as their own *Triggers* (store data into Cassandra for further analysis and send an *E-mail Message* to a stakeholder) that are activated if the result for the rule condition is true. Considering these two types of rules, two types of dashboards were created on the *Analytical Application*, one oriented for operational analysis and the other one oriented for a more tactical point of view. These dashboards, and more details about the proposed aproach and current results, can be seen in [19].

6 Conclusions and Future Work

Considering the current evolution of the industrial world, this document presents the status and main research goals of a doctoral thesis that is dedicated to exploring the Big Data and CEP concepts in the Industry 4.0 movement. The context for the emergence of this topic of interest, as well as the research agenda, were explained in this work, and the state-of-the-art was detailed to highlight the contributions that distinguish this work from the already existing contributions. From the methodological point of view, the research methodology was presented, and the current status of the proposed contribution was also highlighted.

At this moment, a first version of the system was already implemented with the Bosch Car Multimedia Portugal demonstration case, where the 1st, 2nd, 3rd and 5th objectives discussed in Section 4 were fulfilled. With this prototype, a CEP system in Big Data contexts that reveals its adequacy to the problem and contains several components already developed (e.g., Data Producers and Consumers, Rules and Triggers with the business requirements, an Event Aggregator and an Analytical Application as Destination System) is presented to practitioners and researchers.

As future work, the components of the architecture should be quantitatively evaluated (e.g., benchmarking), as mentioned in the methodology section. Moreover, three other relevant components will be developed and evaluated: Predictors and Recommenders based on ML; Graph Database for metadata management; and, Web Visualization Platform for the system's monitoring.

Acknowledgements. This work has been supported by FCT – Fundação para a Ciência e Tecnologia, Projects Scope UID/CEC/00319/2019 and PDE/00040/2013, and the Doctoral scholarship PD/BDE/135101/2017. This paper uses icons made by Freepik, from www.flaticon.com.

References

1. Leavitt, N.: Complex-Event Processing Poised for Growth. Computer. 42, 17–20 (2009). doi:10.1109/MC.2009.109
2. Chakravarthy, S., Qingchun, J.: Stream Data Processing: A Quality of Service Perspective: Modeling, Scheduling, Load Shedding, and Complex Event Processing. Springer US (2009)
3. Luckham, D.C.: Rapide: A Language and Toolset for Simulation of Distributed Systems by Partial Orderings of Events. Stanford University, Stanford, CA, USA (1996)
4. Luckham, D.C., Vera, J.: An event-based architecture definition language. IEEE Transactions on Software Engineering. 21, 717–734 (1995). doi:10.1109/32.464548
5. Cugola, G., Margara, A.: Processing Flows of Information: From Data Stream to Complex Event Processing. ACM Comput. Surv. 44, 15:1–15:62 (2012). doi:10.1145/2187671.2187677
6. Hadar, E.: BIDCEP: A vision of big data complex event processing for near real time data streaming position paper - A practitioner view. In: CEUR Workshop Proceedings (2016)
7. Flouris, I., Manikaki, V., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Mock, M., Bothe, S., Skarbovsky, I., Fournier, F., Stajcer, M., Krizan, T., Yom-Tov, J., Curin, T.: FERARI: A Prototype for Complex Event Processing over Streaming Multi-cloud Platforms. In: Proceedings of the 2016 International Conference on Management of Data. pp. 2093–2096. ACM, New York, NY, USA (2016)
8. Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M., Mock, M.: Issues in complex event processing: Status and prospects in the Big Data era. Journal of Systems and Software. 127, 217–236 (2017). doi:10.1016/j.jss.2016.06.011
9. Babiceanu, R.F., Seker, R.: Manufacturing Cyber-Physical Systems Enabled by Complex Event Processing and Big Data Environments: A Framework for Development. In: Service Orientation in Holonic and Multi-agent Manufacturing, Studies in Computational Intelligence. pp. 165–173. Springer International Publishing Switzerland 2015 (2015)
10. Krumeich, J., Jacobi, S., Werth, D., Loos, P.: Big Data Analytics for Predictive Manufacturing Control - A Case Study from Process Industry. In: 2014 IEEE International Congress on Big Data. pp. 530–537. , Anchorage, AK, USA (2014)
11. Dundar, B., Astekin, M., Aktas, M.S.: A Big Data Processing Framework for Self-Healing Internet of Things Applications. In: 12th International Conference on Semantics, Knowledge and Grids (SKG). pp. 62–68. , Beijing, China (2016)
12. Costa, C., Santos, M.Y.: Evaluating Several Design Patterns and Trends in Big Data Warehousing Systems. In: Krogstie, J. and Reijers, H.A. (eds.) Advanced Information Systems Engineering. pp. 459–473. Springer International Publishing (2018)
13. Tawsif, K., Hossen, J., Emerson Raja, J., Jesmeen, M.Z.H., Arif, E.M.H.: A Review on Complex Event Processing Systems for Big Data. In: 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP). , Kota Kinabalu, Malaysia (2018)
14. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Q. 28, 75–105 (2004)
15. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems. 24, 45–77 (2007). doi:10.2753/MIS0742-1222240302
16. Ganta, M.: How-to: Build a Complex Event Processing App on Apache Spark and Drools, https://blog.cloudera.com/blog/2015/11/how-to-build-a-complex-event-processing-app-on-apache-spark-and-drools/

17. Yang, F., Tschetter, E., Léauté, X., Ray, N., Merlino, G., Ganguli, D.: Druid: A real-time analytical data store. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. pp. 157–168. ACM, Utah, USA (2014)
18. Correia, J., Santos, M.Y., Costa, C., Andrade, C.: Fast Online Analytical Processing for Big Data Warehousing. Presented at the International Conference on Intelligent Systems, Madeira Island, Portugal September (2018)
19. Andrade, C., Correia, J., Costa, C., Santos, M.Y.: Intelligent Event Broker: A Complex Event Processing System in Big Data Contexts. In: AMCIS 2019 Proceedings. Cancun (2019). Manuscript accepted for publication.