IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

EVENT-DRIVEN ARCHITECTURES FOR REAL-TIME ANALYTICS AND DECISION-MAKING

[1]SANKUL SETH

[1] AVP, Data, Analytics & Science, Pensacola, Florida, United States

Abstract: Event-driven architectures (EDAs) allow real-time responses to state changes in a software system. EDAs allow for proactive responses to issues, while also generating data that organizations can use to make predictions and decisions. Event-driven architectures and decision systems have applications across sectors as diverse as retail, finance, crisis management, and manufacturing. There are costs and challenges associated with event-driven architectures; however, in the proper use cases and with proper implementation, the benefits of real-time decision making outweigh the costs. An increasing number of software tools and frameworks combine EDAs with predictive analytics and mining of historical data to make accurate, real-time decisions.

Key Word: Event-driven architectures, events, decision-support system, big data, internet of things

I. INTRODUCTION

As data volume and throughput continues to expand across domains, software systems face an increased need to quickly respond to a constant influx of events. In software development, an event can refer to any notable occurrence or state change, whether a problem, potential problem, opportunity, or deviation. [1] Alternatively, an event can be a digitized version of a real-life incident. [2] Events originate from sources including sensors, applications, social media, and other data streams. [1]

Traditional batch processing and request/response architectures, in which events are processed in groups once a day, are often insufficient to meet the growing needs of organizations. Event-driven architectures (EDAs) have emerged as a powerful alternative for enabling real-time analytics and decision-making. In an EDA, after being emitted by event sources, events are then routed to appropriate event consumers or processors, where they trigger various actions, such as updating databases, generating alerts, or triggering downstream processes. This architectural style emphasizes the use of events as the primary means of communication and coordination between different components of a system. Ideal use-cases for EDAs include integrating applications, sharing data across applications, and analyzing data generated by the Internet of Things (IoT).

Many organizations utilize decision-support systems, in which large amounts of data are analyzed with software models to help decide on optimal courses of action. Event-driven architectures can provide the backbone for these decision systems. This is known as event-driven decision making and has applications ranging from traffic management to financial fraud detection. [6]

II. BENEFITS OF EVENT-DRIVEN ARCHITECTURES

The main appeal of event-driven architectures is that they allow proactive, near-real time responses to critical events. Additionally, the data collected from EDAs allows key insights to be made and opens the potential for predictive analytics. Event-driven architectures are scalable and flexible, able to utilize parallel processing to handle large volumes of data from different sources. They are also able to integrate different systems together.

The components of EDA systems are loosely coupled, sending, and receiving information without awareness of its source or destination. This means that parts can be easily added, removed, or updated without affecting the rest of the system. This independence allows faster development, as components can be worked on concurrently, and increases the resiliency of the system by preventing cascading failures.

III. COMPONENTS OF EDA SYSTEM

EDA systems vary across implementations, but key components include:

- **Event sources**: An event source is the trigger for the event - for example, a customer clicking on an offer. As EDA systems are loosely coupled, the event source is unaware of and acts independently from the rest of the EDA system. Depending on the source, events are generated in a variety of formats and may require transformation before they can be processed. [1]
- **Event brokers**: An event broker, also known as an event bus or message broker, acts as a mediator between the event producer and the event consumer. It can also process and analyze data in real time and perform event routing, queuing, logging, and auditing. For example, an e-commerce platform which experiences a high volume of orders from customers might utilize an event broker to efficiently process these orders while keeping track of the inventory in real-time. A network of interconnected event brokers is called an event mesh.
- **Event consumers**: Event consumers, which receive messages from the event broker, provide temporary storage in the user's browser. This storage is then cleared when the session ends. Event consumers speed operations by eliminating the need to communicate with the source for all operations.
- **Event processors**: Event processors establish a connection with the source to be maintained for the duration of a session. They carry out processes within the user's space. These processes are then killed when the session is ended.
- **Engines**: Engines process events and trigger actions. Simple engines process events individually, while complex engines can place them in a larger context. Engines can also collect and preserve event data for use in analytics. [1]

IV. COMBINING EDAS WITH PREDICTIVE ANALYTICS

Predictive event-driven process analytics is an emerging field. Rather than use only historical data to make predictions, predictive event-driven process analytics uses real-time data generated by events. EDAs can be combined with predictive analytics to create intelligent and proactive systems. The goal is to make fast, accurate predictions and reduce the distance between observation and action. [4]

For example, in the field of supply chain management, event streams from various points in the supply chain can be used to gather data about topics such as inventory levels, shipment status, and demand forecast. Predictive analytics can then be used to manage inventories, anticipate demand fluctuations, and proactively respond to supply chain disruptions.

Challenges remain in implementation, including standardization of event formatting, process log events, and optimization. [4] However, software tools such as preCEP, which uses a CEP engine to link real-time business activity monitoring with mining of historical data for predictive analytics, have been developed to help address these challenges. [5]

IV. APPLICATIONS OF EDA IN MANUFACTURING

In the sector of manufacturing, event-driven architectures can be used to manage and respond to the large amounts of data generated by the Internet of Things (IoT), as well as the data gathered by physical and digital sensors during the manufacturing and shipping processes. Combined with event monitoring and data processing, EDAs can allow for proactive maintenance of machines and systems. Sensors and related devices are capable of detecting state changes. These events can then be processed and acted upon by EDAs. [3] Analyzing sensor data and using it to make predictions is a concept known as "e-maintenance." Harnessing EDAs with e-maintenance has the potential to optimize proactive maintenance. [3]

An EDA-driven workflow in manufacturing would consist of the following steps: first, sensor data is collected and enriched with background information. Unusual events trigger the generation of predictions through predictive analytics. Finally, these predictions are used to make decisions and act. For example, imagine an oil and gas company that implements an EDA to oversee their equipment maintenance. One day, sensors might detect a loss of friction in a drilling machine's grill box. A predictive model might then estimate that based on this observation, it is likely the gearbox will break down soon. The model would then recommend the optimal maintenance action, as well as the ideal time frame for that action. A system of this nature could significantly reduce downtimes and improve performance, giving the company that adopted it a competitive advantage.[3]

EDAs have been successfully used by large corporations such as Unilever to modernize their supply chains and manage their shipping lines. By enabling organizations to collect information across ports and shipping lines and respond to it in real time, EDAs can enable flexible, interconnected, and resilient supply chains.

VI. APPLICATIONS OF EDA IN RETAIL

In retail, EDAs can be used to create a tailored experience for customers. By analyzing real-time customer interactions and behaviors, organizations can use predictive models to offer personalized recommendations and targeted marketing in the moment. For example, the Home Shopping Network's website successfully combined predictive analytics with EDAs to create its advertising sidebar. EDAs collected data about user behavior, including clicks and purchases, and used this to update the sidebar with products that the user was likely to be interested in purchasing next - increasing the likelihood of a sale.

VII. CHALLENGES AND ESSENTIAL CONSIDERATIONS FOR IMPLEMENTING EDA

While beneficial in a wide variety of use cases, event-driven architectures do come with some drawbacks and inherent challenges. Therefore, they are not ideal for all circumstances. Because data from user sessions is stored in event consumers, EDAs typically do not have access to historical data unless combined with other systems. Responding to events continuously does increase the load on software systems, and there are hardware limitations that can make scaling these architectures challenging. Therefore, EDAs must be carefully designed. The tasks triggered by events in EDAs should be simple and streamlined. Attempting to trigger complex tasks can cause long lag times and crashes, resulting in a frustrating user experience.

The number of events generated by the architecture must also be monitored and managed. Before implementing EDAs, businesses should review their workflow and examine if the use-case is one that would truly benefit from the implementation of an EDA.

Dealing with large volumes of data inherently brings challenges. Careful data management is necessary to reap the full benefits of EDAs. Timely decision-making also requires that unnecessary movement of data is eliminated. Distributing event-recognition tasks among nodes may help with dealing with large volumes of data. [6] Inductive logic programming (ILP) is another technique that can help identify anomalies in large data streams, which is especially helpful in use cases such as financial fraud identification. [6]

Additionally, when it comes to event-driven decision-making, uncertainty is impossible to fully eliminate. Systems will rarely be able to access all the information that might be helpful in making a sound decision. Therefore, it is important that event-driven decision-making systems be able to make decisions and forecasts in the face of significant uncertainty and incomplete information. Incorporating frameworks that utilize probabilistic reasoning and symbolic methods, like Markov Logic Networks, can help handle these uncertainties. [6]

VIII. CONCLUSION

Event-driven architectures can provide real-time responses to user actions, creating a more dynamic and personalized experience. They can also generate data that provides useful insight into user behavior and can be used for predictive analytics. EDAs can bring a variety of benefits to organizations and users, but they cannot be deployed indiscriminately. Improper management, such as including too many events or attempting to trigger overly complex actions, can cause system lags or crashes. However, as storage and hardware continue to improve, these limitations are decreasing. The future of EDAs likely involves a combination with predictive analytics, in which historical events are captured and analyzed in order to optimize decisions.

REFERENCES

[1] Michelson, B. (2011). Event-Driven Architecture Overview, Elemental Links Research.

[2] Fertier, A. et al. (2020). Real-time data exploitation supported by model- and event-driven architecture to enhance situation awareness, application to crisis management. Enterprise Information Systems, 14(6): 769-796.

[3] Bousdekis, A. et al. (2015). A Real-Time Architecture for Proactive Decision Making in Manufacturing Enterprises. On the Move to Meaningful Internet Systems: OTM 2015 Workshops. 9416: 137-146.

[4] Schwegmann, B. (2013). A Method and Tool for Predictive Event-Driven Process Analytics. Wirtschaftsinformatik Proceedings. 46.

[5] Schwegmann, B. (2013). preCEP: Facilitating Predictive Event-Driven Process Analytics. Design Science at the Intersection of Physical and Virtual Design. 7939.

[6] A. Artikis et al. (2014). Scalable Proactive Event-Driven Decision Making. IEEE Technology and Society Magazine, 33(3): 35-41.