Incorporation of Machine Learning in the Selection of Event Patterns for Smart Contract Modelling

(RIncorporación de Machine Learning en la selección de Patrones de Eventos para el modelado de Contratos Inteligentes)

Oscar Carlos Medina[1], Brenda Elizabeth Meloni[2] , Ana María Strub[3] y Marelo Martín Marciszack[4]

Original material authorized for its first publication in the journal Ciencia y Tecnología of the Faculty of Engineering, Universidad de Palermo.
Subject area: Computer Science.
Received: 06/30/2025 | Accepted: 07/25/2025

Abstract

Event patterns are used to model interactions between smart contracts and their environment, improving the efficiency, security, and flexibility of applications based

[1] Doctor en Ingeniería, mención Sistemas de Información, egresado de la Universidad Tecnológica Nacional (UTN) - Facultad Regional Córdoba (FRC). Miembro de la carrera de Investigador Científico Categoría C UTN y del Programa de Incentivos del Ministerio de Educación de la Nación Categoría V. Docente adjunto, coordinador de la Diplomatura de Blockchain y finanzas digitales y Director del CIDS Blockchain Lab. Profesor asociado de la Universidad CAECE y coordinador de la Red Iberoamericana RIBCi-CYTED. oscarcmedina@gmail.com

[2] Magister en Docencia Universitaria y Especialista en Sistemas de Información, Universidad Tecnológica Nacional (UTN) - Facultad Regional Córdoba (FRC). Miembro de la carrera de Investigador Científico categoría D y del programa de Incentivos del Ministerio de Educación de la Nación categoría IV. Docente Adjunto/JTP de la carrera de Ingeniería en Sistemas de Información, en IUA-UNDEF. Docente Titular en Ingeniería en Informática. Colaboradora de la Red Iberoamericana RIBCi-CYTED.

[3] Especialista en Ingeniería en Sistemas de Información, egresada de la Universidad Tecnológica Nacional (UTN)- Facultad Regional Córdoba (FRC). Miembro de la carrera de Investigador Científico Categoría D UTN y del Programa de Incentivos del Ministerio de Educación de la Nación Categoría IV. Prof. Asociada, Coordinadora de la Cátedra de Probabilidad y Estadística y docente de la asignatura Análisis de Sistemas. Es Product Owner y Líder Funcional en proyectos del Centro de Investigación Desarrollo y Transferencia de Sistemas de Información (CIDS) de la UTN- FRC.

[4] Doctor en Ingeniería de Software por la Universidad de Vigo, Magíster en Ingeniería de Software - Universidad Nacional de la Plata e Ingeniero en Sistemas de Información- Universidad Tecnológica Nacional. Profesor Titular Ordinario en UTN-FRC. Miembro de la carrera del Investigador Científico Categoría "A" Univ. Tecnológica Nacional y Categoría I Carrera Investigador programa de Incentivos. Director del Centro de UTN de Investigación, Desarrollo y Transferencia de Sistemas de Información (CIDS) de la Facultad Regional Córdoba y Director del Doctorado en Ingeniería, mención Sistemas de Información. Coordinador del Programa Sistemas de Información e Informática de la Universidad Tecnológica Nacional.

on blockchain technologies. Event pattern-driven smart contract modelling is an emerging trend. Defining a new pattern or selecting a pre-existing one from a limited catalog is a fundamental activity for any analysis model that utilizes event patterns. This study presents a proposal to manage a catalog of event patterns that can be applied to the conceptual modelling of smart contracts and incorporates machine learning techniques to optimize pattern selection. To test this proposal, a prototype application called PatCat (Pattern Catalogue) was developed, using a decentralized electronic voting application as a case study. The incorporation of patterns at the beginning of the modelling process simplifies and clarifies the elicitation of requirements, among other benefits, while the use of machine learning accelerates the description of the problem situation. Consequently, a specialized application for managing a catalog of event patterns, supported by machine learning techniques, proves useful in standardizing and streamlining smart contract modelling tasks.

Keywords: Machine Learning; Smart Contract; Blockchain; Artificial Intelligence.

Resumen

Los patrones de eventos se utilizan para modelar las interacciones entre los contratos inteligentes y su entorno, mejorando así la eficiencia, la seguridad y la flexibilidad de las aplicaciones basadas en tecnologías blockchain. El modelado de contratos inteligentes basado en patrones de eventos es una tendencia emergente. Definir un nuevo patrón, o seleccionar uno preexistente de un catálogo determinado, es fundamental para cualquier modelo de análisis que utilice patrones de eventos. Este estudio presenta una propuesta para gestionar un catálogo de patrones de eventos aplicable al modelado conceptual de contratos inteligentes e incorpora técnicas de aprendizaje automático para optimizar la selección de patrones. Para evaluar esta propuesta, se desarrolló un prototipo de aplicación llamado PatCat (Catálogo de Patrones), utilizando como caso práctico una aplicación de votación electrónica descentralizada. La incorporación de patrones al inicio del proceso de modelado simplifica y clarifica la obtención de requisitos, entre otras ventajas, mientras que el uso del aprendizaje automático acelera la descripción del problema. En consecuencia, una aplicación especializada para la gestión de un catálogo de patrones de eventos, con el apoyo de técnicas de aprendizaje automático, resulta útil para estandarizar y optimizar las tareas de modelado de contratos inteligentes.

Palabras claves: Machine Learning; contrato inteligente; Blockchain; Inteligencia Artificial.

1 Introduction

This article explores the intersection of emerging technologies such as blockchain and artificial intelligence (AI), explaining their conceptual characterization and dimensions related to smart contracts and the patterns used for their modelling in blockchain, as well as the learning techniques employed in AI.

The most commonly used AI techniques today include Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL). These techniques create learning models that process and analyze data, perform tasks, or make predictions for various real-world problems. Until recently, such capabilities were considered beyond the scope of traditional computing. However, the use of AI has raised social concerns regarding data manipulation due to centralization, the proliferation of fake news, privacy invasion, and biases in data training (Fraga-Fernández, 2022).

Blockchain is a decentralized database technology in which stored records are not subject to a central authority that can decide their deletion or modification (Nakamoto, 2008). Various blockchain networks operate with modifications; the most well-known is the one supporting the Bitcoin cryptocurrency. However, beyond cryptocurrencies, smart contracts span e-government, supply chains, energy, and banking.

A smart contract is a programming code that is automatically executed when certain predefined conditions are met. These programs are created and stored on a blockchain platform, enabling their automatic execution and verification by multiple nodes on the network.

The use of event patterns in smart contract modelling is an emerging trend. This systematic and agile process allows knowledge and experience from previous solutions described with business patterns to be reused for similar use cases (Medina et al., 2021a).

In the context of smart contracts, event patterns define the conditions that activate the contract and specify the actions triggered by certain events, such as token transfers or the arrival of a new block in the blockchain. A pattern in software engineering provides a standardized, reusable model for describing a process clearly and precisely (Marciszack et al., 2018). Event patterns facilitate the description, analysis, and modelling of event-driven systems, where events occur over time and can be detected by sensors or devices (Zhao et al., 2021).

Blockchain enhances AI by ensuring the provenance, authenticity, and reliability of data sources (Tanwar et al., 2020), while decentralized technology enables secure decision-making without intermediaries. Autonomous systems leveraging smart contracts can learn over time and enable reliable decision-making (Marwala et al., 2018).

The convergence of AI and blockchain is expected to result in Blockchain-AI systems, fostering innovations in smart grids, Industry 4.0 automation, critical infrastructure, 6G networks, the Internet of Vehicles, and data security. Additionally, this integration may be enhanced through other Industry 4.0/5.0 technologies, such

as the Internet of Things (IoT), Edge Computing, and Augmented, Mixed, or Virtual Reality (Fraga-Fernández, 2022).

This paper presents a proposal for cataloging and selecting event patterns for application in smart contract modeling. Additionally, it explores the incorporation of ML techniques to optimize the initial pattern selection process.

The remainder of the paper is structured as follows: Section 2 introduces the background and key concepts of the theoretical framework. Section 3 describes the main features of the working methodology, while Section 4 explains the results obtained in a case study using the PatCat catalog. Finally, Section 6 presents conclusions and future work.

2 Background

Machine Learning (ML) is a subfield of computer science and a branch of artificial intelligence that aims to develop techniques allowing computers to learn autonomously, forming a fundamental pillar of large-scale data processing.

ML enables autonomous learning through three main types of algorithms:

- **Supervised Learning:** Labeled datasets are used to detect patterns and label new data sets.
- **Unsupervised Learning:** Applied when datasets are unlabeled, and categorization is achieved by identifying similarities or differences.
- **Reinforcement Learning:** The system refines itself over time through repeated actions and feedback-based updates (Ramírez, 2018).

ML and blockchain are rapidly evolving technologies with increasing applications across various industries. Blockchain ensures secure, transparent transaction recording, while ML facilitates data-driven decision-making by analyzing vast data volumes. The combination of these technologies has the potential to revolutionize industries by enhancing efficiency through automated, reliable processes, fostering data-driven decision-making, and strengthening security measures by reducing vulnerabilities and ensuring data integrity (Liu et al., 2020).

The synergy between ML and blockchain creates secure, decentralized, intelligent, and efficient network transaction and management systems. This union offers the following benefits:

- **Enhanced data** and information contribution.
- **Greater security** and confidentiality in decision-making.

Liu et al. (2020) assert that ML significantly advances blockchain in communication and networking systems by improving efficiency, scalability, and

security. The integration of these technologies facilitates the secure and transparent storage of large data volumes, enables ML models to be trained with reliable data, and enhances data privacy through decentralized ownership and controlled access.

Blockchain is a distributed ledger technology that enables the creation of immutable, tamper-proof records. Essentially, it is a decentralized and distributed database in which multiple parties can transact and share information without the need to trust a centralized intermediary.

At the end of 2013, Vitalik Buterin began developing a new blockchain network called (Ethereum), with the first proof of concept published in February 2014. The goal of this platform was to create a system that would allow the implementation of smart contracts on a blockchain network, whether public or private.

A smart contract is a set of algorithms or computer operations designed to eliminate intermediaries, simplifying processes while ensuring compliance with predefined conditions. A smart contract only proceeds to the next stage if a specific task is successfully completed. Since it operates on a blockchain, it becomes a transparent, auditable, and immutable agreement, which also saves time and money.

In this work, the use of event patterns is considered to facilitate the reuse of business knowledge in smart contract modelling, as detailed below.

A pattern is a description of a common solution to a recurring problem that can be applied to a specific context. Patterns help leverage the collective experience of specialized software, where engineers document existing and proven knowledge in system development.

Eriksson and Penker (2000) distinguish three types of patterns based on the problems they address:

- **Business Patterns:** Address problems in the organizational domain, including defining and relating business processes, rules, corporate visions, and objectives.
- **Architecture Patterns:** Focus on the architectural design of information systems.
- **Design Patterns:** Apply to situations where analysis is already defined, focusing on producing flexible and adaptable technical solutions.

Among architectural patterns, event patterns stand out. Event patterns provide a software engineering approach to describing, analyzing, and modeling event-based systems. In this approach, systems are considered to be composed of a series of events occurring over time, which can be observed and captured by sensors or devices.

In this research, event patterns are applied to the Conceptual Model alongside other patterns that address different facets of modeling, such as business patterns, analysis patterns, design patterns, and scenario patterns.

3 Work methodology

Conceptual modeling aims to identify and explain significant concepts within a problem domain, defining attributes and the relationships between them. This study proposes incorporating event patterns into the modeling process, as they serve as predefined analysis and design schemes. This approach optimizes the initial definition of smart contract behavior, formalizing its structure, facilitating the validation of functional requirements, and reusing previously acquired knowledge from distributed applications.

The rationale for using event patterns instead of design or business patterns is outlined below. Design patterns originated in the object-oriented programming paradigm and are typically applied during system construction. Business patterns, formulated after design patterns, focus on modeling business processes. Event patterns, however, are particularly useful in this context due to their specialization in modeling the autonomous behavior of smart contracts. The use of event patterns in smart contract modeling is an emerging trend. Event patterns enhance smart contract interactions with their environment, improving efficiency, security, and flexibility.

The most frequently used event patterns include time patterns, flow control patterns, and communication patterns. Additionally, specific tools and languages have been developed to facilitate smart contract modeling using event patterns.

Defining new patterns and selecting pre-existing ones from a limited catalog are fundamental tasks in any analysis model utilizing patterns. To experimentally validate an analysis model incorporating patterns, a system was developed to catalog and select Business Patterns for Conceptual Modeling, implemented through a prototype called PatCat (Pattern Catalogue).

PatCat is a web-based catalog of patterns applied to successful Electronic Government implementations. The application was developed under the university research project (Coauthor 1 et al., 2018). The project aimed to define best-practice patterns for E-Government that could be incorporated into the conceptual modeling of public sector websites, optimizing software quality dimensions such as reusability.

The PatCat algorithm employs natural language analysis to interpret the modeler's description of a problem. Text mining techniques analyze cataloged patterns and interact with the designer to recommend the most appropriate pattern for the described problem. Implemented within a Design Patterns catalog, PatCat introduces an interactive method for pattern selection. Initially, keyword searches are performed within the business model, allowing early application in conceptual modeling. The process involves identifying an Event Pattern that aligns with the system being modeled, as this pattern describes the core problems addressed. Next, the Business Pattern is derived in a unified graph that illustrates how related processes interact and exchange resources. The core process specification of this

pattern may also include associated patterns, such as Analysis Patterns and Scenario Patterns, which complement the software requirements specification.

Subsequently, a second version of the software was developed in a follow-up research project (Medina et al., 2021a; 2021b). This iteration refined the scope, limiting the catalog to business patterns specifically applied to smart contract modeling.

4 Results Obtained

The PatCat catalog allows the registration of Patterns, grouped by pattern type and business process. The search engine's data model represents each document stored in the database, in this case, a business pattern, as a vector in a k-dimensional vector space. Each coordinate is a number that indicates the value of a term identified as a keyword in the entered query for each document in the catalog. These input values are translated into input variables for a mathematical model that calculates the relevance of a document to a query. The value assigned to each document for a particular search allows an order of relevance to be established.

This ranking process is managed through an inverted index structure consisting of two elements:

- **Vocabulary:** A set of all the distinct terms registered in the database.
- **Publication List**: For each existing term, a list of documents associated with the searched terms and their relevance values.

Once both structures are obtained, a simple query resolves the search, returning the most relevant documents.

PatCat has two different views: the researcher view and the administrator view. The researcher view consists of a search engine where users can enter data related to the pattern they are looking for. The search engine scans all matching results and sorts them from most to least relevant.

Additionally, clicking on a result displays a detailed view with all relevant registry data. This view includes links for downloading files and images representing the registry structure (→ Figure 1).

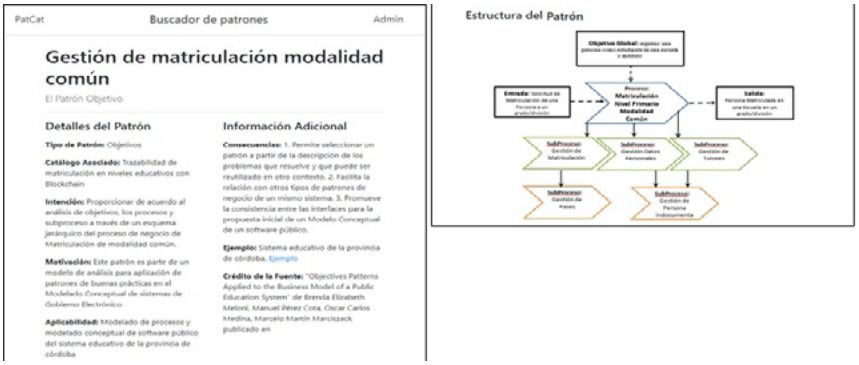

Figure 1. Detailed representation of a Pattern.

The administrator view utilizes the Django framework's administration module (Django Framework), which provides a standardized system for data visualization and modification, generating views based on object definitions. From this section, administrators can systematically manage the data hosted in the application, applying filters according to various parameters, thanks to the flexibility of the exposed interfaces.

In the administrator interface, users can access and modify all PatCat components (→ Figure 2).

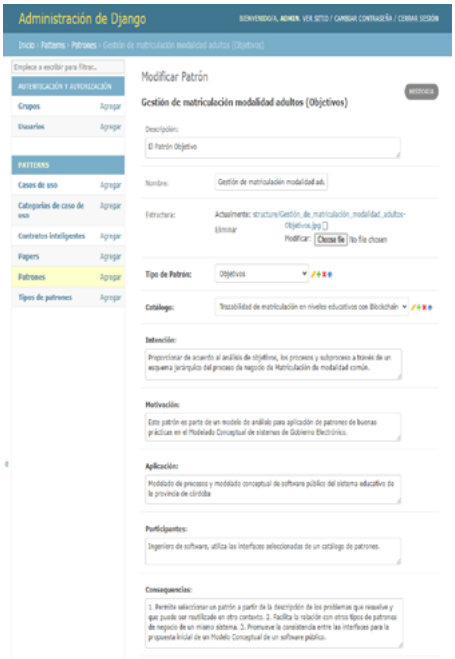

Figure 2. Django Administration main screen.

Each element represented in the Administration view corresponds to a table in the database. The schema (→ Figure 3) illustrates the fields of each table and their foreign key relationships. This schema is generated from the Python definition of the database model.

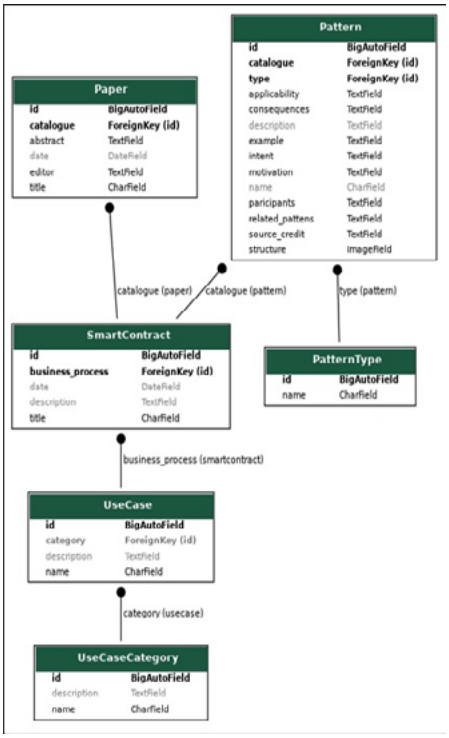

Figure 3. PatCat catalog database model.

This research explores the integration of Machine Learning (ML) and Reinforcement Learning (RL) techniques to enhance both the elicitation of requirements from the oral description provided by smart contract modelers and the selection of event patterns from the PatCat catalog.

1. **Speech AI for Problem Analysis:** To capture and analyze the modeler's narrative description of the smart contract problem, speech AI tools are being evaluated. These include:
    - Google Cloud AI Speech-to-Text API (Speech-to-Text)
    - Open-source frameworks such as (Parler TTS) and (MeloTTS)
    - ScopeMaster's analysis engine (ScopeMaster), which interprets descriptions and provides contextual visualizations and questions to enhance keyword extraction.

2. **Enhancing Pattern Selection with AI:** To improve the accuracy and efficiency of selecting event patterns—both within PatCat and external catalogs—ML tools are being explored, including:
    - ChatGPT prompts for data science (ChatGPT)
    - Google Cloud's generative AI product, Vertex AI Agent Builder (Vertex).

Second, to enhance the selection of event patterns in the catalog—or even to identify event patterns in external catalogs—ML software such as ChatGPT prompts for data science (ChatGPT) and Google Cloud's generative AI product, Vertex AI Agent Builder (Vertex), is being explored.

The event patterns identified by the modeler serve as the initial conceptual model for the smart contract. This approach enhances the efficiency of contract modeling, leverages knowledge from previous implementations, and optimizes reusability, which is a key dimension of software quality.

6. Conclusions

This proposal enhances a conceptual modeling method for systems incorporating patterns, with the latest version focusing on smart contract modeling based on use cases and event patterns.

It has been demonstrated that using patterns in system modeling facilitates reusability, ensuring software quality from the early stages of development. Additionally, it optimizes solutions by clearly and comprehensively addressing potential requirements essential for later stages of the development cycle.

In the case studies conducted, the incorporation of Machine Learning (ML) in problem description and pattern selection helped clarify the elicitation of requirements, identify problems, and discover solutions and similarities between scenarios, thereby simplifying the modeling process.

The method presented in this work facilitates the anticipation of problem-solution scenarios within the process circuit, enabling their management in the same language as the system user. Representing patterns through graphs provides a clearer business perspective. It also allows for the easy identification of processes and their activities, validating the reuse of patterns. Furthermore, it promotes the use of a precise and standardized vocabulary between the process owner and the analyst, eliminating ambiguities and refining requirements. The integration of two emerging technologies in this project—Blockchain and Artificial Intelligence—creates a synergy that enhances productivity in the repetitive tasks involved in modeling smart contracts.

This article provides an example of AI application in the development of Blockchain technology systems, specifically facilitating the selection of event patterns applicable to smart contract modeling.

The aforementioned AI tools could enable, in the first case, the analysis of voice-based narratives to identify information system requirements. In the second case, they could assist in the selection of event patterns that align with the identified requirements by precisely defining prompts composed of contextualized key concepts.

The implementation of AI in this field is planned for future work, aiming to further explore each tool presented. Additionally, an experimental evaluation of this modeling method and the PatCat catalog is planned to assess their effectiveness and efficiency by reusing the conceptual model, driven by event patterns, in new smart contracts within the same use case.

Finally, the transfer of this method and the PatCat catalog software to the private, public, and educational sectors is planned through its publication in an open-source repository.

References

ChatGPT. A repository of 60 useful data science prompts for ChatGPT, https://github.com/travistangvh/ChatGPT-Data-Science-Prompts

Django Framework, https://www.djangoproject.com/

Eriksson, H. E. y Penker, M. (2000). *Business modeling with UML: Business patterns at work*. OMG Press.

Ethereum, https://ethereum.org/en/

Fraga Lamas, P. y Fernández Caramez, T. (2022). *Introductory chapter: Opportunities and challenges on the convergence of blockchain and artificial intelligence*. En T. M. Fernández-Caramés y P. Fraga-Lamas (Eds), Advances in the convergence of blockchain and artificial intelligence. IntechOpen. https://www.intechopen.com/chapters/79564

Liu, Y., Yu, F. R., Li, X., Ji, H. y Leung, V. C. M. (2020). *Blockchain and machine learning for communications and networking systems*. *IEEE Communications Surveys & Tutorials, 22*(2), 1392-1431.

Marciszack, M. M., Moreno, J. C., Sánchez, C. E., Medina, O. C., Delgado, A. F. y Castro, C. S. (2018). *Patrones en la construcción del modelo conceptual para sistemas de información*. Editorial edUTecNe, Universidad Tecnológica Nacional.

Marwala, T. y Xing, B. (2018). *Blockchain and artificial intelligence*, https://arxiv.org/abs/1802.04451

Medina, O. C., Marciszack, M. M. y Groppo, M. A. (2018). *Proposal for the patterns definition based on good practices for the electronic government systems development*. In Proceedings of CISTI 2018 – 13th Iberian Conference on Information Systems and Technologies

Medina, O. C., Marciszack, M. M. y Groppo, M. A. (2021a). *Una aproximación a la incorporación de patrones de negocio en el modelado de aplicaciones de contratos inteligentes.* Jornadas de Ciencia y Tecnología 2021 de U.T.N. – F.R.S.F., (art. 871). Editorial edUTecNe, U.T.N.

Medina, O. C., Pérez Cota, M., Meloni, B. E., & Marciszack, M. M. (2021b). Business patterns catalogue and selection proposal for the conceptual model of a software product. *J.UCS Journal, 27*(2), 135-151. https://doi.org/10.3897/jucs.65083

MeloTTS, https://github.com/myshell-ai/MeloTTS

Nakamoto, S.: Bitcoin (2008). A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf

Parler TTS., https://github.com/huggingface/parler-tts

Ramírez, D. H. (2018). *El machine learning a través de los tiempos, y los aportes a la humanidad*. Universidad Libre - Seccional Pereira Facultad de Ingenierías, Colombia.

ScopeMaster, https://www.scopemaster.com/es/

Speech-to-Text. Convierte voz en texto con la IA de Google, https://cloud.google.com/speech-to-text?hl=es

Tanwar, S., Bhatia, Q., Patel, P., Kumari, A., Singh, P. K. y Hong, W. (2020). Machine learning adoption in blockchain-based smart applications: The challenges and a way forward. *IEEE Access, 8*, 474-488.

Vertex. AI Agent Builder, https://cloud.google.com/products/agent-builder?hl=es-419

Zhao, W., Huang, T., Chen, Y., Deng, Y. y Tan, W. (2021). A systematic literature review on event pattern-based modeling of smart contracts. Concurrency and Computation: Practice and Experience.