JOT JOURNAL OF OBJECT TECHNOLOGY

Journal of Object Technology | RESEARCH ARTICLE

Model-Driven Engineering for Complex Event Processing: A Survey

Jesús Rosa-Bilbao* **and Juan Boubeta-Puig***
*(UCASE Software Engineering Research Group, Department of Computer Science and Engineering, University of Cadiz, Spain)

ABSTRACT Complex Event Processing (CEP) is a powerful technology for analyzing and correlating large amounts of data coming from different application domains to automatically detect situations of interest (event patterns) in real time. However, extensive knowledge on CEP is required to be able to implement CEP applications. To alleviate this situation, in recent years, several works have proposed the use of Model-Driven Engineering (MDE) to facilitate the development of such CEP applications for domain experts. In this paper, we propose a systematic literature review of existing approaches, frameworks, systems and languages that integrate MDE with CEP, along with the application domains and maturity levels with which these proposals have been successfully adopted. Based on our findings, future research challenges in the CEP field are also discussed.

KEYWORDS Complex Event Processing, Domain-Specific Language, Model-Driven Development, Model-Driven Engineering.

1. Introduction

Complex Event Processing (CEP) (Luckham 2012) is a cutting-edge technology that allows event streams to be correlated and processed continuously to identify situations of interest in real time according to a set of rules (event patterns) previously defined.

However, CEP technology requires a lot of knowledge to be able to create applications that make use of this technology (Boubeta-Puig et al. 2014; Burgueño, Boubeta-Puig, & Vallecillo 2018). For this reason, in recent years several authors have proposed solutions, such as (Boubeta-Puig et al. 2015a; Corral-Plaza et al. 2021), that integrate CEP with Model-Driven Engineering (MDE) paradigm to facilitate the development of CEP applications for non-expert users of this technology.

MDE (Brambilla et al. 2017) is an increasingly used field of software engineering that allows to raise the level of abstraction by using models that focus only on the features of interest thus alleviating complexity (de Lara & Guerra 2021). These models can be aimed at code generation or proof of correctness testing of a solution, among other things. The application of MDE increases the automation of software development and allows problems to be detected early in the software development cycle (Burgueño, Vallecillo, & Gogolla 2018).

The aim of this paper is to conduct a Systematic Literature Review (SLR) of existing approaches, frameworks, systems and languages that integrate MDE with CEP, along with the application domains and maturity levels with which these proposals have been successfully adopted. Specifically, a SLR is used to structure a research area, collect information and make a synthesis of the analyzed data. To do this, it is necessary to evaluate existing proposals and identify how they should be updated in order to mark a way forward (Petersen et al. 2015).

To achieve this, five research questions have been formulated that seek to analyze the main advantages of the integration of both technologies, the main characteristics of model-based solutions for CEP, the application areas in which these solutions have been adopted, limitations or difficulties of the MDE and CEP integration and, finally, future research challenges in the field of CEP.

The remainder of the paper is structured as follows. Section 2 introduces the technologies used in this work. Section 3 describes the methodology used for the SLR. Section 4 presents the results after the selection of the contributions, and Section 5 discusses the research questions proposed. Finally, Section 6 draws conclusions of our work.

JOT reference format:
Jesús Rosa-Bilbao and Juan Boubeta-Puig. *Model-Driven Engineering for Complex Event Processing: A Survey.* Journal of Object Technology. Vol. 21, No. 4, 2022.
http://dx.doi.org/10.5381/jot.2022.21.4.a10

2. Background

This section describes the MDE and CEP technologies.

2.1. Model-Driven Engineering

MDE promotes the development of models, in the broad sense of the word, at different levels of abstraction. Models that are at higher levels are transformed into lower level models; this is an iterative process until finally transforming the models to code (Burgueño et al. 2019). One of the main advantages of this paradigm is that it allows the development of software in an efficient way due to its level of abstraction, where the amount of programming is reduced considerably (Rožanc & Mernik 2021).

The main purpose of MDE is to increase productivity by reusing standardized models, thus maximizing compatibility between systems. MDE enables simplification in the software design process by using models with design patterns for a specific domain. It also improves communication by standardizing best practices and terminologies used in the application domain (France & Rumpe 2007).

In this context, a Domain-Specific Language (DSL) (Kosar et al. 2016) is a kind of programming language that is focused on a particular domain. A DSL is not general-purpose, so it will have greater usability only among users who are experts in a particular domain and usually not programmers. DSLs are developed by language experts together with a domain expert. Once the DSL is implemented, it will facilitate the development of solutions and reduce the possibility of errors (Castañeda et al. 2022).

Thus, models can be expressed through DSLs (Frank 2013), whose definition consists of three distinct parts: (i) the abstract syntax, which is composed of both a metamodel (a model that describes the language concepts and the relationships between them) and the validation rules that determine whether a model is well formed; (ii) the concrete syntax or notation of the DSL (the set of graphical symbols needed to draw the diagrams); and (iii) the transformations between models and from model to code to conduct the software automation. It should be noted that the same model, i.e. an instance of a metamodel, may have different graphical notations. A DSL provides the following benefits (Fowler & Parsons 2011):

- It improves the development productivity: the problem to be addressed for a specific domain is represented in a more abstract form, which facilitates the specification of what the system has to do; in addition to avoiding the appearance of errors and duplication of code, since it is automatically generated from models.
- It improves the communication with domain experts: as the language is clear, concise and close to the users, it fosters communication between users and software system, who actively participate in modeling what they need to define.
- It facilitates adaptation to changes: the use of models independent of their implementation makes it possible to transform them into executable code in the specific environment that needs to be used at any given time.
- It specifies what, not how: this type of language helps the user to specify what the system should do, but not how it should be done.

Particularly, Model-Driven Development (MDD) is a software development methodology that allows users to build complex applications through simple abstractions (models) of already built components. In addition, it reduces human process intervention through automation since the code can be automatically generated from these models.

Therefore, MDE can be considered as a superset of MDD since MDE goes beyond development activities, including also other software engineering processes such as system evolution or model-driven reverse engineering, always based on the use of models (Mens & Van Gorp 2006).

2.2. Complex Event Processing

CEP is one of the most powerful technologies in distributed real-time environments as it provides a fast and efficient way to correlate and infer conclusions about events occurring in real time (Barquero et al. 2018). This technology can be applied in a multitude of areas such as logistics, monitoring critical infrastructure and financial applications (Hinze et al. 2009), business processes (Soffer et al. 2019), e-health (Caballero et al. 2021), air quality monitoring (Boudriki Semlali et al. 2021; Ortiz, Boubeta-Puig, et al. 2022), cybersecurity (Roldán et al. 2020; Roldán-Gómez et al. 2021), Internet of things (Ortiz, Castillo, et al. 2022), autonomous unmanned aerial vehicles (Boubeta-Puig et al. 2018), self-adaptive systems (Parra-Ullauri et al. 2022; Romero-Garcés et al. 2022), and intelligent transportation systems (Brazález et al. 2022).

One of the most important features is the ability to express event patterns by defining rules. These rules can be implemented using different Event Processing Languages (EPLs) such as Esper EPL (EsperTech 2022) and SiddhiQL (WSO2 2022).

In the field of CEP, simple events are indivisible events that occur at a specific moment in time. When several simple events are correlated, a complex event can be produced, providing meaningful and valuable information (Luckham 2012; Event Processing Technical Society 2011). Specifically, these complex events are automatically generated by a CEP engine when a series of conditions defined in an event pattern are satisfied. A CEP engine is a software component that allows programmers to implement event patterns through the use of EPLs (Burgueño, Boubeta-Puig, & Vallecillo 2018). Once an event pattern is detected, the CEP engine is able to take actions in real time.

The operation of CEP technology consists of three stages (see Figure 1):

1. **Event capture**: it receives the simple events to be analyzed and correlated in real time.

2. **Analysis**: it detects situations of interest when the conditions previously defined in an event patterns are satisfied.

3. **Response**: the actions to be taken in response to the situations detected, notifying them to the interested parties.

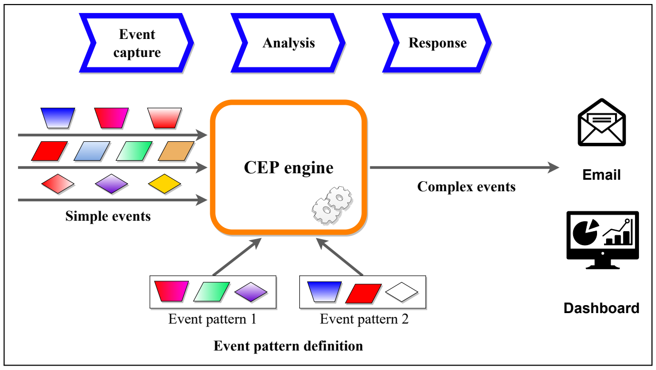

Figure 1 CEP stages.

The use of CEP technology brings many advantages, including fast and automatic response, reduction of human workload, improvement of decision quality and prevention of information overload (Chandy & Schulte 2010). The time required for the decision making process is considerably reduced compared to traditional event analysis techniques as these situations of interest can be detected and reported in real time (García-López et al. 2018).

3. Methods

This section presents the methodology of Kitchenham et al. (Kitchenham et al. 2009), which was used for the SLR conducted in this work. It follows the guidelines of the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) (Moher et al. 2009). In order to perform the SLR, we defined a review protocol with the following steps (Xiao & Watson 2017): (i) research questions; (ii) search strategies; (iii) inclusion and exclusion criteria; (iv) screening procedures; (v) data extraction; (vi) methodological quality assessment; and (vii) synthesis and reporting.

3.1. Research questions

The purpose of the SLR presented in this paper is to review and analyze the state of the art of existing proposals combining MDE and CEP technologies. To this end, it was broken down into the following Research Questions (RQ):

- RQ1: Which are the main advantages/benefits of integrating MDE and CEP?
- RQ2: Which are the main characteristics/functionalities of the existing model-driven solutions for CEP?
- RQ3: Which are the application domains in which these solutions have been successfully adopted?
- RQ4: Which are the limitations or difficulties of the MDE and CEP integration?
- RQ5: Which are the future research challenges in the CEP field?

3.2. Search strategies

To carry out the resource search, we considered the four most well-known databases in computer science: IEEE Xplore, ACM Digital Library, Scopus and Web of Science. IEEE Xplore is a scientific database that allows us to access journal articles, conference proceedings, technical standards and other resources related to the scope of this paper. ACM Digital Library is one of the most complete collections containing articles and bibliographic records on computing, informatics and related areas. Finally, Scopus and Web of Science are two of the main existing multidisciplinary databases because they index quality research; in addition, they have a large coverage of scientific journals and conference proceedings.

We determined three Search Queries (SQ) that allowed us to include all articles that have relevance in their titles, abstracts or keywords according to the aim of this paper:

- **SQ1:** ("domain-specific language" OR "domain-specific modeling language" OR "DSL" OR "DSML") AND ("complex event processing" OR "CEP")
- **SQ2:** ("model-driven development" OR "MDD") AND ("complex event processing" OR "CEP")
- **SQ3:** ("model-driven engineering" OR "MDE") AND ("complex event processing" OR "CEP")

3.3. Inclusion and exclusion criteria

As inclusion criteria, we decided to include all contributions that integrate CEP technology through the use of models, either because they use MDE technology or because they include DSL. Excluded articles are those that (i) have no authors; (ii) have no abstract; (iii) contributions are not related to the integration of CEP with MDE or DSL; (iv) contributions that do not adequately integrate CEP technology with MDE or DSL; (v) contributions that are not relevant to the objective of this SLR.

3.4. Screening procedures

The articles in this SLR were included according to the following steps:

- First step: duplicate contributions, references without authors or without abstracts were removed.
- Second step: all the titles and abstracts of the contributions were evaluated and those that did not comply with the inclusion and exclusion criteria were removed.
- Third step: the full texts of the remaining contributions were evaluated according to the inclusion and exclusion criteria, removing those that did not meet them.

Throughout this process, all articles were reviewed by two authors and any disagreements between them were discussed and resolved by consensus. Finally, as a result of this screening procedure, a list of contributions was obtained which were included in this SLR.

3.5. Data extraction

With respect to the extraction of data of interest for this SLR, we analyzed the following information for each of the contributions: (i) demographic data of the study (authors, year and source of publication); (ii) scope of the contribution and its objectives; (iii) purpose of the contribution with respect to this SLR; (iv) results of the contribution; (v) limitations of the contribution; and (vi) technologies used.

3.6. Methodological quality assessment

All contributions included in this SLR were evaluated a second time (in addition to the inclusion and exclusion criteria) against a set of six Quality Questions (QQ) (see Table 1) which were adopted from studies by (Shahin et al. 2014; Yang et al. 2021). Please note that each question was answered according to whether it does or does not apply.

For each contribution, two authors independently rank the six proposed quality questions. In case of disagreement, these were resolved by discussion until a consensus was reached.

3.7. Synthesis and reporting

From the demographic data obtained, we elaborated a synthesis of the characteristics of the selected contributions. Specifically, this synthesis includes: (i) number of contributions published in conferences, scientific journals or book chapters; (ii) distribution of contributions according to the year of publication; (iii) distribution of contributions by geographical area, taking the affiliation of the first author into account.

The objectives of each of the contributions were summarized. In addition, these contributions were classified according to their objectives and contributions in 5 different fields: (i) approaches; (ii) frameworks; (iii) systems; (iv) DSLs; and (v) other languages.

To evaluate the maturity level of the contributions included, we distinguished different stages of development in which it can be found: (i) requirements: the contribution is in an initial requirements analysis phase; (ii) design: the contribution focuses on an overview of an application architecture or some of its components; (iii) technical testing: the contribution includes a tool with no or insufficient evaluation results; (iv) prototype testing: the contribution includes performance evaluations and/or end-user evaluations with a working version of the tool; (v) pilot testing: the contribution includes a real-world evaluation in a specific context or domain; and (vi) mature: the contribution includes a final application, tested by end-users, ready for deployment in a final production environment. Additionally, there is a target analysis and end-user experiments.

4. Results

This section presents the selection of studies, methodological quality assessment, demographics, characteristics, case studies and maturity level of the contributions.

4.1. Selection of the studies

Figure 2 presents the PRISMA flow diagram of the SLR presented in this paper. The search for bibliographic references was carried out in April 2022 and included all types of contributions published in the last 10 years (2011 – 2021). A total of 211 articles were retrieved from the different queries executed in the chosen databases (see Table 2).

First, we eliminated duplicate articles, articles without abstracts, or articles without authors, corresponding to a total of 69 excluded articles. After this first filtering, the results were reduced to 142 articles. Secondly, after a first reading of the title and abstract of the remaining 142 articles, we decided to eliminate 99 more articles that were outside the context analyzed in this paper, i.e. they do not cover both MDE and CEP technologies in any way. After this second filtering, the results were reduced to 43 articles. Finally, after analysis of the full text, 11 articles were eliminated because they did not meet the inclusion and exclusion criteria. Therefore, the number of articles selected for this SLR was 32.

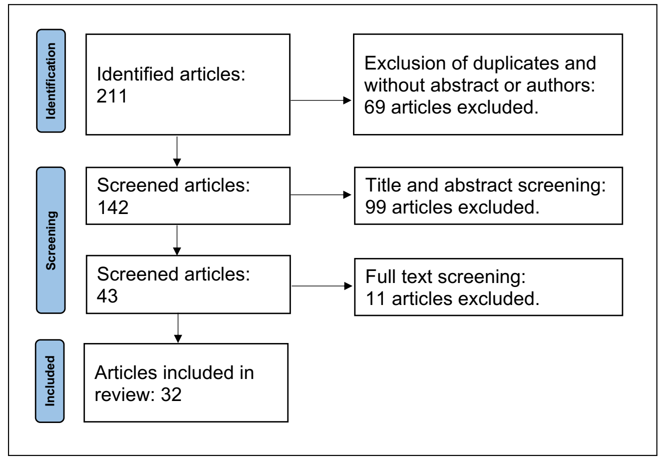

Figure 2 PRISMA flowchart of the SLR.

4.2. Demographics of the contributions

Table 3 shows the included contributions in terms of publication types. As it can be seen, there is a total of 32 contributions (18 journal articles, 13 conference articles and 1 book chapter).

Figure 3 illustrates the publication years of the included contributions. Specifically, the contributions selected for this SLR were published between 2011 and 2021, that is, the last 10 years. Moreover, as the figure shows, the year with the least number of publications was 2012 with 0 articles and the year with the most publications was 2018 with 6 articles.

Figure 4 shows the distribution of contributions by geographical area. Spain has the highest number of contributions with 9, followed by Germany with 8. The remaining countries have similar numbers, ranging from 1 to 3 contributions per country.

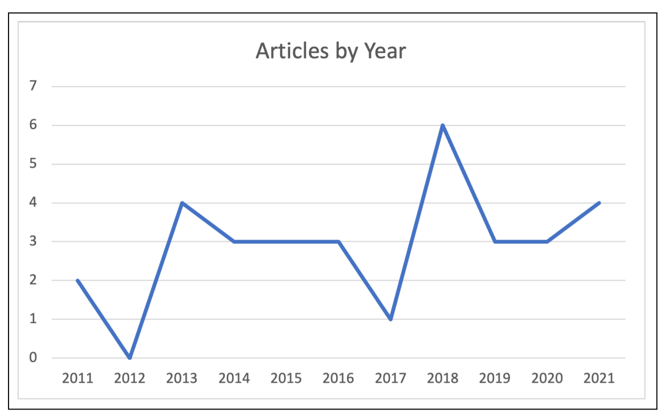

Figure 3 Distribution of articles by year of publication.

	Study quality assessment questions
QQ1	Are the objectives of the study clearly identifed?
QQ2	Is the context of the study clearly stated?
QQ3	Does the research methods support the aims of the study?
QQ4	Has the study an adequate description of the technologies being used?
QQ5	Is there a clear statement of the findings?
QQ6	Are limitations of the study discussed explicitly?

Table 1 Study quality assessment questions.

	Scopus	IEEE	ACM	Web of Science	Total
SQ1	20	4	77	11	112
SQ2	12	2	26	6	46
SQ3	7	5	36	5	53
Total	39	11	139	22	211

Table 2 Results obtained after querying the databases.

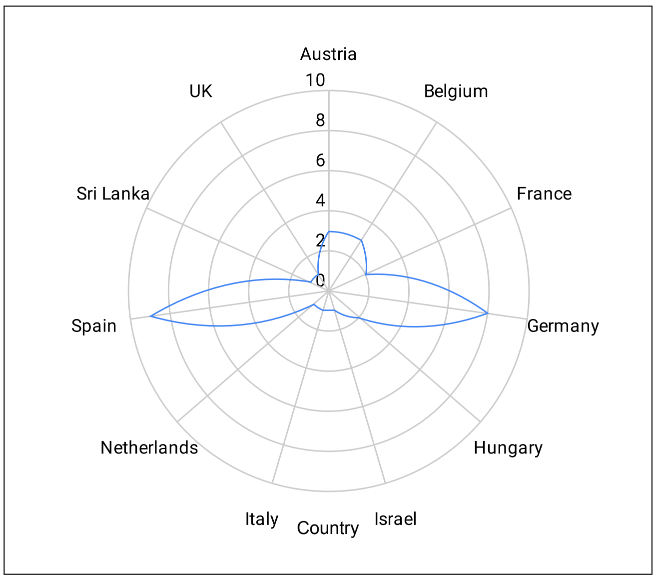

Figure 4 Distribution of articles by country.

4.3. Methodological quality assessment

Figure 5 shows the results of the evaluation of the methodological quality of the contributions included in this SLR: 31 contributions indicated the aims of the research conducted (*QQ1. Are the objectives of the study clearly identified?*), 30 contributions commented on the context to which the study was applied (*QQ2. Is the context of the study clearly stated?*), 22 contributions explicitly included the methods that supported their research (*QQ3. Does the research methods support the aims of the study?*), 31 contributions adequately described which technologies were used to carry out the study (*QQ4. Has the study an adequate description of the technologies being used?*), 31 contributions

Type of publication	Contributions
Journal article	(Amrani et al. 2018) (Boubeta-Puig et al. 2014) (Boubeta-Puig et al. 2015a) (Boubeta-Puig et al. 2015b) (Boubeta-Puig et al. 2019) (Boubeta-Puig et al. 2021) (Calderón et al. 2018) (Clemente & Lozano-Tello 2018) (Corral-Plaza et al. 2021) (Dávid et al. 2018) (Ehmes et al. 2020) (Garnier et al. 2016) (Humm & Hutter 2020) (Leroy et al. 2020) (Mulo et al. 2013) (Taher et al. 2013) (Valero et al. 2022) (Volanschi et al. 2018)
Conference article	(Bauer & Wolff 2014) (Baur et al. 2019) (Bertolino et al. 2011) (Bruns et al. 2014) (Etzion et al. 2016) (Kambona et al. 2015) (Kohler et al. 2018) (Majzik et al. 2019) (Parra-Ullauri et al. 2021) (Roledene et al. 2016) (Tragatschnig & Zdun 2013) (Vidackovic & Weisbecker 2011) (Weisenburger et al. 2017)
Book chapter	(Dávid & Gönczy 2013)

Table 3 Types of contributions included.

presented evaluations and results of their studies (*QQ5. Is there a clear statement of the findings?*) and, finally, 14 contributions discussed the limitations of their results (*QQ6. Are limitations of the study discussed explicitly?*).

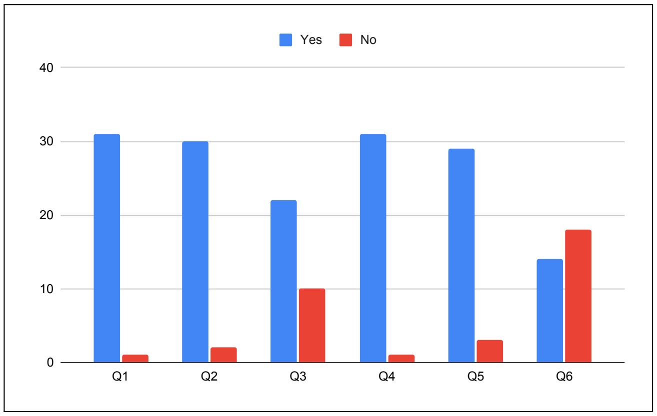

Figure 5 Methodological quality assessment of the included contributions.

4.4. Characteristics of the contributions

From the analysis of papers' aims, we identified the following types of contributions included in this SLR:

- **Approaches for automatic code generation**, 13 contributions for: (i) facilitating the design of event patterns (situations of interest to be detected) and then automatically transforming into EPL code (Boubeta-Puig et al. 2014); (ii) allowing domain experts to easily define both event patterns and alerts for real-time notification, and then automatically transforming them into EPL code, which is executable by a CEP engine, and XML code, which is executable by an enterprise service bus, hiding all implementation details through the proposed MEdit4CEP graphical modeling tool (Boubeta-Puig et al. 2015a); (iii) transforming event pattern models into PCPN models and then into XML, which is executable by the CPN Tools framework with the aim of checking the semantic behavior of the patterns, through the developed MEdit4CEP-CPN, an extension of MEdit4CEP that integrates CEP and PCPN (Boubeta-Puig et al. 2019); (iv) allowing domain experts to graphically define event patterns that interact with the Ethereum blockchain network without the need to write code through CEPchain, a graphical modeling tool that integrates CEP with blockchain (Boubeta-Puig et al. 2021); (v) facilitating the design of gamification strategies, their monitoring and code generation in CEP-based systems through MEdit4CEP-Gam, an extension of the MEdit4CEP graphical modeling tool that integrates CEP and gamification (Calderón et al. 2018); (vi) analyzing open data sources in near-real time through the integration of MDD and CEP capable of generating specific code from heterogeneous technologies (Clemente & Lozano-Tello 2018); (vii) consuming, processing and analyzing heterogeneous data in real time, providing domain experts with MEdit4CEP-SP, an extension of MEdit4CEP that integrates Stream Processing (SP) and CEP to infer and define heterogeneous data domains, and model the event patterns to be detected in these domains (Corral-Plaza et al. 2021); (viii) combining CEP techniques with live model queries and transformations (Dávid et al. 2018); (ix) achieving automatic transformations from computation independent model for mobile fraud detection (Etzion et al. 2016); (x) testing black-box systems for autonomous vehicles with guaranteed coverage by combining model-based techniques (Majzik et al. 2019); (xi) the creation of adapters of incompatible web service interfaces and transforming them into code through a graphical editor (Taher et al. 2013); (xii) allowing end users to easily define event patterns and obtain an automatic transformation into BPCPNs, i.e. PCPN models with black sequencing transitions, and then transforming them into their corresponding PCPNs, which are executed in CPN Tools (Valero et al. 2022); and (xiii) the integration of model-driven development with CEP and Business Process Modeling Notation (BPMN) for dynamic executable service compositions in enterprise environment (Vidackovic & Weisbecker 2011).
- **Frameworks**, 4 contributions for: (i) monitoring applications in the cloud through the combination of CEP and MDD (Baur et al. 2019); (ii) providing reusable and customized elements of a context-specific language to aid efficient development of DSLs (Dávid & Gönczy 2013); (iii) providing programmer constructs dedicated to distributed event composition and group coordination by the Mingo framework (Kambona et al. 2015); and (iv) modeling enterprise integration architectures that combine concepts of event-driven architecture, model-based development and architectural views (Tragatschnig & Zdun 2013).
- **Systems**, 5 contributions for: (i) processing text stream and filtering information through a reference domain model (Bauer & Wolff 2014); (ii) detecting composite events by using the publish/subscribe messaging pattern that is integrated in a monitoring infrastructure called Glimpse (Bertolino et al. 2011); (iii) describing error conditions during the automatic assembly of robot components by using learning patterns and training from a labeled data set so that employers are able to correctly classify optimization problems (Humm & Hutter 2020); (iv) empowering smart trading strategies based on events by the so-called Genibux system (Roledene et al. 2016); and (v) automatically adapting to changes in environmental conditions while guaranteeing a certain level of quality through the AdaptiveCEP system (Weisenburger et al. 2017).
- **DSLs**, 7 contributions for: (i) defining and manipulating IoT devices in domestic environments by using IoTDSL to deploy IoT infrastructures in a declarative and semantic event-driven way (Amrani et al. 2018); (ii) defining CEP domains and event patterns through a graphical DSL known as ModeL4CEP (Boubeta-Puig et al. 2015b); (iii) defining event patterns through a textual DSL known as DS-EPL (Bruns et al. 2014); (iv) filtering and identifying user-specific events by integrating CEP with multi-tree sen-

sor modeling (Garnier et al. 2016); (v) performing runtime verification by using a temporal property language with a semantics customized for this verification (Leroy et al. 2020); (vi) creating monitoring components for process-based Service-Oriented Architectures (SOAs) using development techniques (Mulo et al. 2013); and (vii) applying online states of binary sensors to ambient-assisted living (Volanschi et al. 2018).

– **Other languages**, 3 contributions for: (i) combining Incremental Graph Pattern Matching (IGPM) and CEP patterns by a tool built on a new specification language called GrapeL (Ehmes et al. 2020); (ii) integrating CEP with network computing through a tool to compile CEP operations formulated in P4 code through a rule specification language known as P4CEP (Kohler et al. 2018); and (iii) integrating CEP with event graph models to allow Self-Adaptive Systems (SAS) domain experts to describe global system behavior and represent global explanations in a human-understandable way (Parra-Ullauri et al. 2021).

4.5. Case studies of the contributions

Table 4 summarizes the case studies in which the different contributions included in this SLR have been applied. These case studies can be of two types: real or simulated. We consider a real case study if it has been applied to a real scenario, or the data used is real data. However, a simulated case study is one that has been applied to simulated (not real) scenarios or the data used is test data. Please note that some contributions have not been applied to any case studies, so they have been marked with a "-".

4.6. Maturity level of the contributions

Table 5 shows the different development phases of the contributions included in this SLR. No contribution is in the requirements phase and only two are in the design phase. The remaining contributions have been classified into technical testing (5 contributions), prototype testing (12 contributions) and finally pilot testing (13 contributions). Remarkably, none of the contributions included in this SLR has been classified as mature.

Figure 6 shows the number of contributions by maturity level over the years of publication. The most mature solutions are not necessarily associated with the most recent years. Figure 3 shows how in recent years the number of articles published has been increasing, this is why Figure 6 shows an increasing trend in the maturity of the contributions.

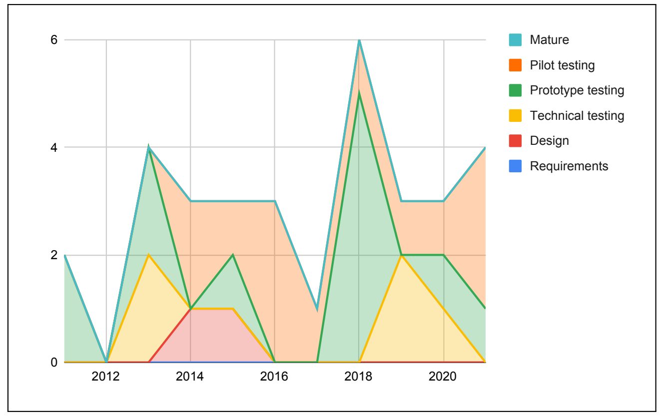

Figure 6 Number of contributions by maturity level over years.

5. Discussion

In this section, we will discuss and answer the research questions formulated in Section 3.1.

– **RQ1**: *Which are the main advantages/benefits of integrating MDE and CEP?*
From the contributions included and analyzed in this SLR, we can affirm that the integration of MDE and CEP brings several advantages. One of this advantages is that graphical modeling editors (Boubeta-Puig et al. 2015a,b), tools (Calderón et al. 2018) and solutions (Roledene et al. 2016) can be provided to users, who are not experts in certain technologies but in a specific domain, with the aim of facilitating them the detection of situations of interest in real time (Corral-Plaza et al. 2021). These real-time situations can be analyzed, processed and a series of actions can be performed according to the event patterns that have been previously defined. Other advantages are as follows: models are user-friendly and defined once and then automatically transformed into different required implementation languages like Esper EPL or SiddhiQL (Boubeta-Puig et al. 2015a), error-free implementations (Leroy et al. 2020), productivity improvement in development and easy adaptability to changes (Weisenburger et al. 2017). Additionally, it also allows integrating with other technologies such as blockchain (Boubeta-Puig et al. 2021), P4 (Kohler et al. 2018), IGPM (Ehmes et al. 2020) and PCPN (Boubeta-Puig et al. 2019). Moreover, most of the contributions can be used in different domains, i.e. they can be applied to different real-world scenarios (Taher et al. 2013).

– **RQ2:** *Which are the main characteristics/functionalities of the existing model-driven solutions for CEP?*
The main characteristics/functionalities of the contributions in the scope of the MDE for CEP applications are: (i) Graphical modeling editors to define CEP domains and patters (Boubeta-Puig et al. 2014); (ii) Automatic generation of CEP application models (Boubeta-Puig et al. 2015b); (iii) Automatic code generation for CEP applications (Calderón et al. 2018); (iv) Real-time decision making through models in CEP applications (Boubeta-Puig et al. 2015a); (v) Alert detection and notification through models for CEP applications (Boubeta-Puig et al. 2019); (vi) Graphical editor that allows the creation of CEP applications that interact with blockchain networks (Boubeta-Puig et al. 2021); (vii) Graphical editor that allows the creation of CEP applications in the field of gamification to implement, control and monitor gamification strategies (Calderón et al. 2018); (viii) Tools addressing the complexity of heterogeneous technologies in the context of open data sources and CEP through a MDD ap-

Contribution	Case study	Type
(Amrani et al. 2018)	Smart home	Simulated
(Bauer & Wolff 2014)	Processing text stream and filtering information	Real
(Baur et al. 2019)	Multi-cloud infrastructure	Simulated
(Bertolino et al. 2011)	Photo-sharing scenario	Simulated
(Boubeta-Puig et al. 2019)	Air quality	Real
(Boubeta-Puig et al. 2014)	Health-care scenario	Simulated
(Boubeta-Puig et al. 2015a)	Network security	Simulated
(Boubeta-Puig et al. 2015b)	-	-
(Boubeta-Puig et al. 2021)	Vaccine supply chain scenario	Real
(Bruns et al. 2014)	Machine-to-machine communication	Real
(Calderón et al. 2018)	Gamification	Simulated
(Clemente & Lozano-Tello 2018)	Air quality & earthquake data	Simulated
(Corral-Plaza et al. 2021)	Smart water network management	Real
(Dávid & Gönczy 2013)	IT infrastructure monitoring	Simulated
(Dávid et al. 2018)	On-the-fly gesture recognition	Simulated
(Ehmes et al. 2020)	Flight booking scenario	Simulated
(Etzion et al. 2016)	Mobile phone fraud detection	Real
(Garnier et al. 2016)	Rooms lighting scenario	Simulated
(Humm & Hutter 2020)	Assembly of electrical components by a robot	Simulated
(Kambona et al. 2015)	Mobile drawing application	Simulated
(Kohler et al. 2018)	Predicates on simple L3/L4-packets	Simulated
(Leroy et al. 2020)	Executable DSLs (xDSLs)	Simulated
(Majzik et al. 2019)	Autonomous vehicles	Simulated
(Mulo et al. 2013)	-	-
(Parra-Ullauri et al. 2021)	Airborne base stations SAS	Simulated
(Roledene et al. 2016)	Foreign currency exchange market (Forex)	Simulated
(Taher et al. 2013)	Web services with incompatible interfaces	Simulated
(Tragatschnig & Zdun 2013)	Telecom industry	Simulated
(Valero et al. 2022)	Monitoring of pregnant women	Simulated
(Vidackovic & Weisbecker 2011)	Logistics processes for aviation catering goods	Simulated
(Volanschi et al. 2018)	Ambient assisted living	Real
(Weisenburger et al. 2017)	-	-

Table 4 Case studies of the included contributions.

Maturity level	Contributions
Requirements	None
Design	(Bauer & Wolff 2014) (Boubeta-Puig et al. 2015b)
Technical testing	(Baur et al. 2019) (Dávid & Gönczy 2013) (Humm & Hutter 2020) (Majzik et al. 2019) (Tragatschnig & Zdun 2013)
Prototype testing	(Amrani et al. 2018) (Bertolino et al. 2011) (Clemente & Lozano-Tello 2018) (Dávid et al. 2018) (Ehmes et al. 2020) (Kambona et al. 2015) (Kohler et al. 2018) (Mulo et al. 2013) (Parra-Ullauri et al. 2021) (Taher et al. 2013) (Vidackovic & Weisbecker 2011) (Volanschi et al. 2018)
Pilot testing	(Boubeta-Puig et al. 2014) (Boubeta-Puig et al. 2015a) (Boubeta-Puig et al. 2019) (Boubeta-Puig et al. 2021) (Bruns et al. 2014) (Calderón et al. 2018) (Corral-Plaza et al. 2021) (Etzion et al. 2016) (Garnier et al. 2016) (Leroy et al. 2020) (Roledene et al. 2016) (Valero et al. 2022) (Weisenburger et al. 2017)
Mature	None

Table 5 Maturity levels of the included contributions.

proach (Clemente & Lozano-Tello 2018); (ix) CEP applications capable of consuming and processing heterogeneous data in real time (Corral-Plaza et al. 2021); (x) Integration of IGPM technology into CEP applications through specification languages (Ehmes et al. 2020); (xi) Specification language to integrate CEP with network computing (Kohler et al. 2018); (xii) Genibux Strategy Language (GSL) that allows for the improvement of the trading experience in the foreign exchange market thanks to a high-level editor and diagrams based on CEP applications (Roledene et al. 2016); (xiii) Model-driven approach to model enterprise integration architectures (Tragatschnig & Zdun 2013); (xiv) Graphical editor to transform event patterns into PCPN models (Valero et al. 2022); and (xv) Model-driven proposal that integrates CEP with BPMN (Vidackovic & Weisbecker 2011).

– **RQ3:** *Which are the application domains in which these solutions have been successfully adopted?*
The domains in which these contributions have been applied successfully include IoT in domestic environments (Amrani et al. 2018), text stream processing and information filtering (Bauer & Wolff 2014), monitoring (Mulo et al. 2013), automatic generation of CEP to PCPN domains (Valero et al. 2022), easy design of event patterns (Boubeta-Puig et al. 2014), detection of real-time notifications and alerts (Boubeta-Puig et al. 2015a), blockchain integration (Boubeta-Puig et al. 2021), gamification integration (Calderón et al. 2018), analysis of open data sources (Clemente & Lozano-Tello 2018), assistance in the efficient development (Dávid & Gönczy 2013), mobile phone fraud detection (Etzion et al. 2016), multi-tree sensor modeling (Garnier et al. 2016), binary sensors (Volanschi et al. 2018), automatic assembly of robot components (Humm & Hutter 2020), programmer constructions (Kambona et al. 2015), network computing (Kohler et al. 2018), autonomous vehicles (Majzik et al. 2019), human-understandable expressions (Parra-Ullauri et al. 2021), enterprise integration architectures (Tragatschnig & Zdun 2013), BPM (Vidackovic & Weisbecker 2011), dynamic environments (Weisenburger et al. 2017), smart cities (Clemente & Lozano-Tello 2018), air quality (Clemente & Lozano-Tello 2018), water quality (Corral-Plaza et al. 2021), heterogeneous data (Corral-Plaza et al. 2021), SOAs (Mulo et al. 2013) and trading systems (Roledene et al. 2016).

– **RQ4:** *Which are the limitations or difficulties of the MDE and CEP integration?*
The main limitations or difficulties presented by the authors of the selected papers are as follows: (i) Not being able to represent all types of patterns, only those related to Enterprise Application Integration (EAI) patterns (Tragatschnig & Zdun 2013); (ii) Lack of computational power for labeling when processing text streams as it requires a large amount of statistical calculations (Bauer & Wolff 2014); (iii) Necessity of knowing EPLs to be able to define complex event patterns to obtain better results (Clemente & Lozano-Tello 2018); (iv) There are not enough oper-

ators to be able to model any event pattern that can be implemented directly with an EPL (Boubeta-Puig et al. 2015b); (v) Inability to take into account the legal regulations and ethical issues of specific domains where such solutions are applied (Calderón et al. 2018); (vi) Differences in performance depending on the CEP engine used for integration (Kambona et al. 2015); and (vii) Latency problems may occur (Kohler et al. 2018).

– **RQ5:** *Which are the future research challenges in the CEP field?*
  Most of the authors agree on the lines and challenges of future work on their contributions in the field of CEP, which are as follows: (i) Extension of the solutions so that they can be used in any domain (Vidackovic & Weisbecker 2011); (ii) Extension of the solutions so that these can be used by anyone regardless of his/her knowledge about the technologies (Amrani et al. 2018); (iii) Possibility to run the solutions on lightweight devices, such as a micro controller (Volanschi et al. 2018); (iv) Optimizing and improving the performance of the solutions so that they can be scalable (Dávid et al. 2018); (v) Integrating the developed solutions with other unused and emerging technologies (Corral-Plaza et al. 2021); (vi) Generating reliability and user confidence in the solutions (Bertolino et al. 2011); (vii) Possibility of being able to define more complex patterns (Boubeta-Puig et al. 2019); and finally (viii) Improving usability of the proposals (Corral-Plaza et al. 2021).

6. Conclusions

In this paper, we presented a SLR with the objective of reviewing and analyzing the state of the art of the contributions that combine in some way the use of MDE to facilitate the development of CEP applications for domain experts. Relevant contributions were identified and classified into the following categories: approaches for automatic code generation, frameworks, systems, DSLs and other languages.

The contributions included in this paper were applied to multiple cutting-edge simulated and real-world scenarios, such as smart cities, autonomous vehicles, robots, IoT, blockchain, gamification and BPM. Most of the analyzed contributions clearly included technical details and results obtained after applying them to these types of scenarios.

The contributions usually provide a tool giving support to their proposals, which have different maturity levels between technical testing, prototype testing and pilot testing. There are no proposals that are in an initial stage of requirements analysis and only two works present proposals that are in a design phase. Likewise, none of the contributions presents enough data to determine that its proposal has a mature level. Therefore, based on the results obtained in this SLR, we can state that, in general, there is a lack of maturity in the identified contributions.

As a conclusion, this work shows that the use of MDE paradigm for the development of CEP applications allows users, who are not experts in technologies but are domain experts, to easily develop such applications, hiding the implementation details from them. Moreover, these contributions can be applied to most of the real-world scenarios of interest, but they may need to evolve to a level of maturity.

Dedication

This work is dedicated to Prof. Antonio Vallecillo on the occasion of his 60th birthday. We would like to thank Antonio for all he has contributed during his research career to the Software Engineering scientific community, especially in software modeling and analysis, with special focus on social computing applications and uncertainty modeling. We would also like to highlight the motivation, enthusiasm, dedication, rigor and constancy with which he has carried out the collaborations with his colleagues. Thank you so much, Antonio, for your attention, support and closeness to young researchers, always encouraging us to face new challenges in research!

Acknowledgments

This work was supported by the Spanish Ministry of Science and Innovation and the European Regional Development Fund (ERDF) under projects FAME [RTI2018-093608-B-C33] and AwESOMe [PID2021-122215NB-C33], the Andalusian Plan for Research, Development and Innovation and ERDF under project DECISION [P20_00865], and the Research Plan from the University of Cadiz and Grupo Energético de Puerto Real S.A. under project GANGES [IRTP03_UCA].

References

Amrani, M., Gilson, F., & Englebert, V. (2018). Complex event processing for user-centric management of IoT systems. *Communications in Computer and Information Science*, *880*, 426–448. doi: 10.1007/978-3-319-94764-8_18

Barquero, G., Burgueño, L., Troya, J., & Vallecillo, A. (2018). Extending complex event processing to graph-structured information. In *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* (p. 166–175). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3239372.3239402

Bauer, A., & Wolff, C. (2014). An Event Processing Approach to Text Stream Analysis: Basic Principles of Event Based Information Filtering. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems* (pp. 35–46). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2611286.2611288

Baur, D., Griesinger, F., Verginadis, Y., Stefanidis, V., & Patiniotakis, I. (2019). A model-driven engineering approach for flexible and distributed monitoring of cross-cloud applications. In *Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2018* (pp. 31–40). doi: 10.1109/UCC.2018.00012

Bertolino, A., Calabró, A., Lonetti, F., & Sabetta, A. (2011). GLIMPSE: A generic and flexible monitoring infrastructure. In *ACM International Conference Proceeding Series* (pp. 73–78). doi: 10.1145/1978582.1978598

Boubeta-Puig, J., Díaz, G., Macià, H., Valero, V., & Ortiz, G. (2019). MEdit4CEP-CPN: An approach for complex event processing modeling by prioritized colored Petri nets. *Information Systems*, *81*, 267–289. doi: 10.1016/j.is.2017.11.005

Boubeta-Puig, J., Moguel, E., Sánchez-Figueroa, F., Hernández, J., & Preciado, J. C. (2018). An Autonomous UAV Architecture for Remote Sensing and Intelligent Decision-making. *IEEE Internet Computing*, *22*(3), 6–15. doi: 10.1109/MIC.2018.032501511

Boubeta-Puig, J., Ortiz, G., & Medina-Bulo, I. (2014). A model-driven approach for facilitating user-friendly design of complex event patterns. *Expert Systems with Applications*, *41*(2), 445–456. doi: 10.1016/j.eswa.2013.07.070

Boubeta-Puig, J., Ortiz, G., & Medina-Bulo, I. (2015a). MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0. *Knowledge-Based Systems*, *89*, 97–112. doi: 10.1016/j.knosys.2015.06.021

Boubeta-Puig, J., Ortiz, G., & Medina-Bulo, I. (2015b). ModeL4CEP: Graphical domain-specific modeling languages for CEP domains and event patterns. *Expert Systems with Applications*, *42*(21), 8095–8110. doi: 10.1016/j.eswa.2015.06.045

Boubeta-Puig, J., Rosa-Bilbao, J., & Mendling, J. (2021). CEPchain: A graphical model-driven solution for integrating complex event processing and blockchain. *Expert Systems with Applications*, *184*, 115578. doi: 10.1016/j.eswa.2021.115578

Boudriki Semlali, B.-E., El Amrani, C., Ortiz, G., Boubeta-Puig, J., & Garcia-de Prado, A. (2021). SAT-CEP-monitor: An air quality monitoring software architecture combining complex event processing with satellite remote sensing. *Computers & Electrical Engineering*, *93*, 107257. doi: 10.1016/j.compeleceng.2021.107257

Brambilla, M., Cabot, J., & Wimmer, M. (2017). Model-driven software engineering in practice, second edition. *Synthesis Lectures on Software Engineering*, *3*(1), 1-207. doi: 10.2200/S00751ED2V01Y201701SWE004

Brazález, E., Macià, H., Díaz, G., Valero, V., & Boubeta-Puig, J. (2022). PITS: An Intelligent Transportation System in pandemic times. *Engineering Applications of Artificial Intelligence*, *114*, 105154. doi: 10.1016/j.engappai.2022.105154

Bruns, R., Dunkel, J., Lier, S., & Masbruch, H. (2014). DS-EPL: Domain-specific event processing language. In *DEBS 2014 - Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems* (pp. 83–94). doi: 10.1145/2611286.2611296

Burgueño, L., Boubeta-Puig, J., & Vallecillo, A. (2018). Formalizing Complex Event Processing Systems in Maude. *IEEE Access*, *6*, 23222–23241. doi: 10.1109/ACCESS.2018.2831185

Burgueño, L., Mayerhofer, T., Wimmer, M., & Vallecillo, A. (2019). Specifying quantities in software models. *Information and Software Technology*, *113*, 82-97. doi: https://doi.org/10.1016/j.infsof.2019.05.006

Burgueño, L., Vallecillo, A., & Gogolla, M. (2018). Teaching UML and OCL models and their validation to software engineering students: an experience report. *Computer Science Education*, *28*(1), 23-41. doi: 10.1080/08993408.2018.1462000

Caballero, P., Ortiz, G., Garcia-de Prado, A., & Boubeta-Puig, J. (2021). Paving the way to collaborative context-aware mobile applications: A case study on preventing worsening of allergy symptoms. *Multimedia Tools and Applications*, *80*(14), 21101–21133. doi: 10.1007/s11042-021-10759-6

Calderón, A., Boubeta-Puig, J., & Ruiz, M. (2018). MEdit4CEP-Gam: A model-driven approach for user-friendly gamification design, monitoring and code generation in CEP-based systems. *Information and Software Technology*, *95*, 238–264. doi: 10.1016/j.infsof.2017.11.009

Castañeda, M., Merayo, M. G., Boubeta-Puig, J., & Calvo, I. (2022). MODELFY: A Model-driven Solution for Decision Making based on Fuzzy Information. *Journal of Universal Computer Science*, *28*(5), 445–474. doi: 10.3897/jucs.76030

Chandy, K., & Schulte, W. (2010). *Event processing: Designing IT systems for agile companies* (1st ed.). New York, USA: McGraw-Hill, Inc.

Clemente, P., & Lozano-Tello, A. (2018). Model Driven Development Applied to Complex Event Processing for Near Real-Time Open Data. *Sensors*, *18*(12), 4125. doi: 10.3390/S18124125

Corral-Plaza, D., Ortiz, G., Medina-Bulo, I., & Boubeta-Puig, J. (2021). MEdit4CEP-SP: A model-driven solution to improve decision-making through user-friendly management and real-time processing of heterogeneous data streams. *Knowledge-Based Systems*, *213*, 106682. doi: 10.1016/j.knosys.2020.106682

Dávid, I., & Gönczy, L. (2013). Ontology-supported design of domain-specific languages: A complex event processing case study. In *Advances and applications in model-driven engineering.* IGI Global. Retrieved from http://www.igi-global.com/chapter/ontology-supported-design-domain-specific/78613/

de Lara, J., & Guerra, E. (2021). Language family engineering with product lines of multi-level models. *Formal Aspects of Computing*, *33*(6), 1173–1208. doi: 10.1007/s00165-021-00554-3

Dávid, I., Ráth, I., & Varró, D. (2018). Foundations for Streaming Model Transformations by Complex Event Processing. *Software and Systems Modeling*, *17*(1), 135–162. doi: 10.1007/s10270-016-0533-1

Ehmes, S., Fritsche, L., & Altenhofen, K. (2020). GrapeL: Combining Graph Pattern Matching and Complex Event Processing. *Communications in Computer and Information Science*, *1262 CCIS*, 180–196. doi: 10.1007/978-3-030-58167-1_13

EsperTech. (2022). *Esper.* http://www.espertech.com/esper/. (Accessed April 18, 2022)

Etzion, O., Fournier, F., Skarbovsky, I., & von Halle, B. (2016). A Model Driven Approach for Event Processing Applications. In *Proceedings of the 10th ACM International Conference on Distributed and Event-Based Systems* (pp. 81–92). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2933267.2933268

Event Processing Technical Society. (2011). *Event Processing Glossary - Version 2.0.* http://www.complexevents.com/wp-content/uploads/2011/08/EPTS_Event_Processing

_Glossary_v2.pdf. (Accessed April 18, 2022)
Fowler, M., & Parsons, R. (2011). *Domain-specific languages*. Upper Saddle River, NJ: Addison-Wesley.
France, R., & Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. In *Future of software engineering (FOSE '07)* (p. 37-54). doi: 10.1109/FOSE.2007.14
Frank, U. (2013). Domain-specific modeling languages: Requirements analysis and design guidelines. In I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, & J. Bettin (Eds.), *Domain engineering: Product lines, languages, and conceptual models* (pp. 133–157). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-36654-3_6
García-López, A., Burgueño, L., & Vallecillo, A. (2018). Static analysis of complex event processing programs. In R. Hebig & T. Berger (Eds.), *Proceedings of MODELS 2018 Workshops: ModComp, MRT, OCL, FlexMDE, EXE, COMMitMDE, MDETools, GEMOC, MORSE, MDE4IoT, MDEbug, MoDeVVa, ME, MULTI, HuFaMo, AMMoRe, PAINS co-located with ACM/IEEE 21st International Conference on Model Driven Engineering Languages and Systems (MODELS 2018), Copenhagen, Denmark, October, 14, 2018* (Vol. 2245, pp. 498–502). CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-2245/mde4iot_paper_1.pdf
Garnier, A., Menaud, J.-M., & Montavont, N. (2016). Bringing complex event processing into multitree modelling of sensors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9687*, 196–210. doi: 10.1007/978-3-319-39577-7_16
Hinze, A., Sachs, K., & Buchmann, A. (2009). Event-based applications and enabling technologies. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems* (pp. 1–15). Nashville, Tennessee: Association for Computing Machinery. doi: 10.1145/1619258.1619260
Humm, B., & Hutter, M. (2020). Learning patterns for complex event detection in robot sensor data. *Communications in Computer and Information Science*, *1173 CCIS*, 138–149. doi: 10.1007/978-3-030-41913-4_12
Kambona, K., Hoste, L., Gonzalez Boix, E., & De Meuter, W. (2015). Coordinating Collaborative Interactions in Web-Based Mobile Applications. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (pp. 181–190). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2817721.2817742
Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, *51*(1), 7-15. doi: 10.1016/j.infsof.2008.09.009
Kohler, T., Mayer, R., Dürr, F., Maaß, M., Bhowmik, S., & Rothermel, K. (2018). P4CEP: Towards In-Network Complex Event Processing. In *Proceedings of the 2018 Morning Workshop on In-Network Computing* (pp. 33–38). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3229591.3229593
Kosar, T., Bohra, S., & Mernik, M. (2016). Domain-specific languages: A systematic mapping study. *Information and Software Technology*, *71*, 77-91. doi: https://doi.org/10.1016/j.infsof.2015.11.001
Leroy, D., Jeanjean, P., Bousse, E., Wimmer, M., & Combemale, B. (2020). Runtime monitoring for executable DSLs. *Journal of Object Technology*, *19*(2), 1–23. doi: 10.5381/jot.2020.19.2.a6
Luckham, D. (2012). *Event Processing for Business: Organizing the Real-Time Enterprise*. New Jersey, USA: John Wiley & Sons.
Majzik, I., Semeráth, O., Hajdu, C., Marussy, K., Szatmári, Z., Micskei, Z., … Varró, D. (2019). Towards System-Level Testing with Coverage Guarantees for Autonomous Vehicles. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)* (pp. 89–94). doi: 10.1109/MODELS.2019.00-12
Mens, T., & Van Gorp, P. (2006). A taxonomy of model transformation. *Electronic Notes in Theoretical Computer Science*, *152*, 125-142. (Proceedings of the International Workshop on Graph and Model Transformation (GraMoT 2005)) doi: 10.1016/j.entcs.2005.10.021
Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *BMJ*, *339*. doi: 10.1136/bmj.b2535
Mulo, E., Zdun, U., & Dustdar, S. (2013). Domain-specific language for event-based compliance monitoring in process-driven SOAs. *Service Oriented Computing and Applications*, *7*(1), 59–73. doi: 10.1007/s11761-012-0121-3
Ortiz, G., Boubeta-Puig, J., Criado, J., Corral-Plaza, D., Garcia-de Prado, A., Medina-Bulo, I., & Iribarne, L. (2022). A microservice architecture for real-time IoT data processing: A reusable Web of things approach for smart ports. *Computer Standards & Interfaces*, *81*, 103604. doi: 10.1016/j.csi.2021.103604
Ortiz, G., Castillo, I., Garcia-de Prado, A., & Boubeta-Puig, J. (2022). Evaluating a Flow-Based Programming Approach as an Alternative for Developing CEP Applications in IoT. *IEEE Internet of Things Journal*, *9*(13), 11489–11499. doi: 10.1109/JIOT.2021.3130498
Parra-Ullauri, J. M., García-Domínguez, A., & Bencomo, N. (2021). From a Series of (Un)fortunate Events to Global Explainability of Runtime Model-Based Self-Adaptive Systems. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)* (pp. 807–816). doi: 10.1109/MODELS-C53483.2021.00127
Parra-Ullauri, J. M., García-Domínguez, A., Bencomo, N., Zheng, C., Zhen, C., Boubeta-Puig, J., … Yang, S. (2022). Event-driven temporal models for explanations - ETeMoX: explaining reinforcement learning. *Software and Systems Modeling*, *21*(3), 1091–1113. doi: 10.1007/s10270-021-00952-4
Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, *64*, 1–18. doi: 10.1016/j.infsof.2015.03.007

Roldán, J., Boubeta-Puig, J., Martínez, J. L., & Ortiz, G. (2020). Integrating Complex Event Processing and Machine Learning: an Intelligent Architecture for Detecting IoT Security Attacks. *Expert Systems with Applications*, *149*, 113251. doi: 10.1016/j.eswa.2020.113251

Roldán-Gómez, J., Boubeta-Puig, J., Castelo Gómez, J. M., Carrillo-Mondéjar, J., & Martínez Martínez, J. L. (2021). Attack Pattern Recognition in the Internet of Things using Complex Event Processing and Machine Learning. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 1919–1926). Melbourne, Australia: IEEE. doi: 10.1109/SMC52423.2021.9658711

Roledene, S., Ariyathilaka, L., Liyanage, N., Lakmal, P., & Bamunusinghe, J. (2016). GeniBux-event based intelligent Forex trading strategy enhancer. In *2016 IEEE International Conference on Information and Automation for Sustainability: Interoperable Sustainable Smart Systems for Next Generation, ICIAfS 2016* (pp. 1–6). doi: 10.1109/ICIAFS.2016.7946562

Romero-Garcés, A., Salles De Freitas, R., Marfil, R., Vicente-Chicote, C., Martínez, J., Inglés-Romero, J. F., & Bandera, A. (2022). QoS metrics-in-the-loop for endowing runtime self-adaptation to robotic software architectures. *Multimedia Tools and Applications*, *81*(3), 3603–3628. doi: 10.1007/s11042-021-11603-7

Rožanc, I., & Mernik, M. (2021). Chapter Three - The screening phase in systematic reviews: Can we speed up the process? In A. R. Hurson (Ed.), *Advances in Computers* (Vol. 123, pp. 115–191). Elsevier. doi: 10.1016/bs.adcom.2021.01.006

Shahin, M., Liang, P., & Babar, M. A. (2014). A systematic review of software architecture visualization techniques. *Journal of Systems and Software*, *94*, 161-185. doi: https://doi.org/10.1016/j.jss.2014.03.071

Soffer, P., Hinze, A., Koschmider, A., Ziekow, H., Di Ciccio, C., Koldehofe, B., … Song, W. (2019). From event streams to process models and back: Challenges and opportunities. *Information Systems*, *81*, 181–200. doi: 10.1016/j.is.2017.11.002

Taher, Y., Boubeta-Puig, J., van den Heuvel, W.-J., Ortiz, G., & Medina-Bulo, I. (2013). A Model-Driven Approach for Web Service Adaptation Using Complex Event Processing. *Communications in Computer and Information Science*, *393 CCIS*, 346–359. doi: 10.1007/978-3-642-45364-9_28

Tragatschnig, S., & Zdun, U. (2013). Enterprise Integration Using Event Actor Based Event Transformations. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (pp. 1434–1440). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2480362.2480630

Valero, V., Díaz, G., Boubeta-Puig, J., Macià, H., & Brazález, E. (2022). A Compositional Approach for Complex Event Pattern Modeling and Transformation to Colored Petri Nets with Black Sequencing Transitions. *IEEE Transactions on Software Engineering*, *48*(7), 2584–2605. doi: 10.1109/TSE.2021.3065584

Vidackovic, K., & Weisbecker, A. (2011). A Methodology for Dynamic Service Compositions Based on an Event-Driven Approach. In *2011 Annual SRII Global Conference* (p. 484-494). Los Alamitos, CA, USA: IEEE Computer Society. doi: 10.1109/SRII.2011.56

Volanschi, N., Serpette, B., Carteron, A., & Consel, C. (2018). A Language for Online State Processing of Binary Sensors, Applied to Ambient Assisted Living. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*(4), 192:1–192:26. doi: 10.1145/3287070

Weisenburger, P., Luthra, M., Koldehofe, B., & Salvaneschi, G. (2017). Quality-Aware Runtime Adaptation in Complex Event Processing. In *Proceedings of the 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (pp. 140–151). IEEE Press. doi: 10.1109/SEAMS.2017.10

WSO2. (2022). *Siddhi.* https://siddhi.io/. (Accessed April 18, 2022)

Xiao, Y., & Watson, M. (2017). Guidance on conducting a systematic literature review. *Journal of Planning Education and Research*, *39*, 0739456X1772397. doi: 10.1177/0739456X17723971

Yang, L., Zhang, H., Shen, H., Huang, X., Zhou, X., Rong, G., & Shao, D. (2021). Quality assessment in systematic literature reviews: A software engineering perspective. *Information and Software Technology*, *130*, 106397. doi: https://doi.org/10.1016/j.infsof.2020.106397

About the authors

Jesús Rosa-Bilbao is a Ph.D. student in the UCASE Software Engineering Research Group at the University of Cadiz (UCA), Spain. He received his B.Sc. degree in Computer Science specialized in Software Engineering from UCA in 2019. In addition, he received his M.Sc. degree in Cybersecurity from UCA, his M.Sc. degree in Project Management from the European Business School in Barcelona (ENEB) and his M.Sc. degree in Business Administration from ENEB in 2020. He also received his M.Sc. degree in Big Data and Business Intelligence from ENEB in 2021. His research interests include complex event processing, event-driven service-oriented architecture, model-driven development, blockchain and cybersecurity. You can contact him at jesus.rosabilbao@alum.uca.es.

Juan Boubeta-Puig received the Ph.D. degree in Computer Science and Engineering from the University of Cadiz (UCA), Cádiz, Spain, in 2014. He is an Associate Professor with the Department of Computer Science and Engineering, UCA. His research interests include real-time big data analytics through complex event processing, event-driven service-oriented architecture, Internet of things, blockchain and model-driven development of advanced user interfaces, and their application to smart cities, industry 4.0, e-health, and cybersecurity. Dr. Boubeta-Puig was honored with the Extraordinary Ph.D. Award from UCA and the Best Ph.D. Thesis Award from the Spanish Society of Software Engineering and Software Development Technologies. You can contact him at juan.boubeta@uca.es.