sensors

MDPI

Systematic Review

Model-Driven Engineering Techniques and Tools for Machine Learning-Enabled IoT Applications: A Scoping Review

Zahra Mardani Korani [1,2], Armin Moin [3], Alberto Rodrigues da Silva [4,*] and João Carlos Ferreira [1,5]

[1] ISCTE, Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, 1649-026 Lisbon, Portugal
[2] Hydraulics and Environment Department, LNEC, 1700-066 Lisbon, Portugal
[3] School of Computation, Information, and Technology (CIT), Technical University of Munich, D-80333 Munich, Germany
[4] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, 1000-029 Lisbon, Portugal
[5] Inov - Inesc Inovação—Instituto de Novas Tecnologias, 1000-029 Lisbon, Portugal
[*] Correspondence: alberto.silva@tecnico.ulisboa.pt

Abstract: This paper reviews the literature on model-driven engineering (MDE) tools and languages for the internet of things (IoT). Due to the abundance of big data in the IoT, data analytics and machine learning (DAML) techniques play a key role in providing smart IoT applications. In particular, since a significant portion of the IoT data is sequential time series data, such as sensor data, time series analysis techniques are required. Therefore, IoT modeling languages and tools are expected to support DAML methods, including time series analysis techniques, out of the box. In this paper, we study and classify prior work in the literature through the mentioned lens and following the scoping review approach. Hence, the key underlying research questions are what MDE approaches, tools, and languages have been proposed and which ones have supported DAML techniques at the modeling level and in the scope of smart IoT services.

Keywords: model-driven engineering; internet of things; data analytics and machine learning; time series; literature review; scoping review

check for updates

Citation: Mardani Korani, Z.; Moin, A.; Rodrigues da Silva, A.; Ferreira, J.C. Model-Driven Engineering Techniques and Tools for Machine Learning-Enabled IoT Applications: A Scoping Review. *Sensors* **2023**, *23*, 1458. https://doi.org/10.3390/s23031458

Academic Editor: Raffaele Bruno

Received: 14 December 2022
Revised: 19 January 2023
Accepted: 24 January 2023
Published: 28 January 2023

1. Introduction

Internet of things (IoT) generally refers to many cyber-physical things, including objects and devices around us that are connected to the Internet and can be controlled, accessed, and managed by a myriad of applications running on smartphones and other computational environments [1]. Things can interact with each other and with the users of the digital services that are based on the IoT, thus, called IoT applications or services [2]. Sensors and devices are connected to a network through which they can interact with each other and their users. The IoT concept can be illustrated with a simple example, such as connecting a smartphone to a TV, or with more complex examples, such as monitoring urban infrastructures and traffic [3,4]. IoT-based applications monitor, control, or interact with the physical world. The most popular IoT application domains are healthcare, environment, smart cities, commercial, and industrial [2]. Furthermore, artificial intelligence techniques are being used in IoT applications to add intelligence to devices and related services [5].

IoT data analytics is the process of analyzing the data that are produced from IoT devices and services [6]. In this scope, data analytics and machine learning (DAML) techniques help to make IoT services smarter by finding patterns, hidden correlations, trends, inferences, and actionable insights that have not been seen before. This would lead to more intelligence, better decision-making, performance, automation, productivity, and accuracy. However, the huge increase in the amount and complexity of data poses new challenges for data analytics [7]. In particular, non-i.i.d data (i.e., data instances for which the usual assumption of independent and identically distributed data may not hold), such as sequential time series data requires reliable methods and techniques that should be supported in tools that are focused on IoT data analytics.

However, designing and developing IoT applications requires skills in various areas. Furthermore, several technologies, such as programming embedded systems, communication, networks, data science, and web engineering [8]. To successfully implement IoT applications that are often quite complex, efficient software development methodologies, approaches, and solutions have been provided to address various issues, including heterogeneity, collaborative development, reusability of software artifacts, and self-adaptation [9,10].

One such proposal is model-driven engineering (MDE), which is a general engineering approach that focuses on using models as the primary artifact for software development [10]. In MDE, models can be used to capture the structure and behavior of a system, and these models can then be transformed into the final implementation of the system using model transformations and code generators. MDE aims to improve the efficiency of the software development process by automating the transformation of models into code, and by providing domain-specific abstractions and notation that make it easier to reason about and design complex systems. MDE is often used with domain-specific languages (DSLs), as the models used in MDE are often expressed using these languages. DSLs are languages designed for a particular domain, such as a particular problem or a particular domain of expertise [11]. DSLs are designed to be more concise and more accessible to use than general-purpose languages; they often provide domain-specific abstractions and notation that make it easier to express solutions to problems in that domain [12].

According to the MDE paradigm, a key artifact is the “model”; thus, the whole software development cycle revolves around models. Many MDE approaches aim to generate code for various platforms automatically [10]. An MDE approach can follow two general stages. The first stage involves the creation of MDE artifacts such as languages and tool workbenches with features like authoring, model-to-model, and model-to-text transformations. In the second stage, people without advanced computer training can easily use these artefacts to design and develop their software systems [12].

However, because various MDE tools and languages are emerging, understanding and selecting the most suitable tools and languages are always challenging, especially for new developers or researchers in the area. From our knowledge, there is no general framework or guideline to help to select which tool(s) and language(s) should be used for supporting the various MDE activities in the development of IoT applications (or just “MDE4IoT” for short) [13,14].

Furthermore, the current literature that reviews MDE4IoT does not provide a comprehensive and complete classification and analysis of the languages, model-based transformations, application domains, and even the supported DAML techniques, as we discuss in this paper. For example, Salman et al. [15] analyzed only a few textual and graphic modeling languages and respective tools, but without an accurate classification, and they did not classify those tools and languages or identify their specific features like model-to-model or model-to-text transformations, as we propose in this paper. A further discussion of the related work is presented below in Section 3.

This paper presents a scoping review of research studies that have proposed or used MDE4IoT. This review includes an accurate classification of the used modeling languages, model-to-model, and model-to-text transformations, but also on identifying the application domains and even on the DAML techniques with a particular focus on time series data. This review can assist researchers and developers in selecting the best MDE tools, languages, and techniques for their IoT projects. Moreover, this paper provides a general discussion of future trends by arguing that more work on DAML, in the context of MDE4IoT, is a worthy research area. We select, classify and review 68 studies presenting the stat-of-the-art in MDE4IoT, a few of them focusing on DAML techniques. In addition, we identify the limitations of prior work and discuss some challenges that address these shortcomings.

This paper is organized into seven sections. Section 2 introduces the background by reviewing the basic concepts of MDE, IoT, and DAML. Section 3 presents the related work, i.e., other papers that have also addressed literature reviews on similar topics. Section 4 describes the followed research methodology based on the scoping review approach. Section 5 presents the results from the scoping review. Section 6 discusses the results obtained from the review and identifies the limitations. Finally, Section 7 presents the main conclusion and open issues.

2. Background

The key concepts considered in the scope of this research are MDE, IoT, and DAML, as introduced below.

2.1. Model-Driven Engineering (MDE)

Model-driven engineering (MDE) is a software development approach that emphasizes using models as a primary means for specifying, designing, and implementing software systems. A simple and popular formula used to describe MDE is the following [16]: Models + Transformations = Software, thus, we briefly introduce these essential concepts as follows [14,16,17]:

A *model* is an abstract representation of a system that eliminates unnecessary details to increase the users' understanding of the system under consideration. In addition, a metamodel is a particular model that describes the structure of a modeling language, i.e., a metamodel defines the abstract syntax of a modeling language [14,17]. *Transformations* are special programs that convert models to other models or to text (e.g., source code) according to specific rules previously defined. Model transformations are commonly classified as model-to-model (M2M) and model-to-text (M2T). In M2M, both the input and the output of the transformation are models, whereas in M2T, the input is a model, but the output is text, such as source code in a programming language like Java or Python.

MDE involves creating and manipulating abstract, high-level models that represent the different aspects of a system, such as its functionality, structure, and deployment. These models can then be analyzed, validated, and transformed using various tools and techniques to generate the actual software code and related artifacts [14,18]. A common goal of MDE is to improve the efficiency and reliability of software development by providing a more abstract and flexible way of specifying and designing systems. MDE also aims to reduce the gap between the conceptual models and the implementation by automating the transformation and code generation process. MDE is often used in complex, large-scale software projects where traditional development approaches can be very cumbersome, and error-prone [19]. MDE relies on using modeling languages, such as UML, SysML, and domain-specific languages (DSLs); and model-driven tools, such as model checkers, and M2M and M2T transformations. MDE is commonly specialized by more concrete approaches, such as model-driven architecture (MDA), model-driven development (MDD) [14], model-based testing (MBT) [20].

2.2. Internet of Things (IoT)

The Internet of things (IoT) refers to the interconnected network of physical devices, vehicles, buildings, and other objects embedded with sensors, software, and connectivity, allowing them to collect and exchange data over the Internet. These connected devices can range from smart appliances, and wearable devices to industrial equipment and infrastructure [21,22].

The main goal of IoT is to improve the efficiency, productivity, and safety of various industries and applications by enabling the seamless exchange of data and information between devices and systems. IoT also enables the creation of new services and business models by leveraging data analytics and machine learning techniques to extract insights and value from the collected data.

IoT relies on various technologies, such as sensors, communication protocols, and cloud computing, to enable the connectivity and interoperability of devices. It also involves using various frameworks, such as the IoT Reference Model, to define the architecture and interoperability of IoT systems.

IoT has the potential to transform various sectors, such as transport and mobility, healthcare, manufacturing, and agriculture, by providing real-time data and insights that can improve the decision-making process and optimize the performance of various systems and processes [21,23]. However, IoT also raises concerns about privacy, security, and sustainability, as the proliferation of connected devices can create new vulnerabilities and risks. Therefore, developing and deploying IoT applications requires careful consideration of these issues and adopting appropriate security measures and best practices.

2.3. Data Analytics and Machine Learning (DAML)

Data analytics and machine learning (DAML) refers to the use of techniques to extract insights and knowledge from data. Data analytics (DA) involves the use of statistical, computational, and visualization techniques to analyze and interpret data. DA can be used to identify patterns, trends, and relationships in data, and to make predictions and decisions based on this analysis. On the other hand, machine learning (ML) involves the use of algorithms and models that can learn and improve their performance over time without explicit programming. This is achieved by training the algorithms on data and allowing them to automatically discover patterns and relationships in the data.

Together, DAML can be used to solve a wide range of problems, including predictive modeling, classification, clustering, and optimization [24]. It can be applied to a variety of domains. These approaches are broadly classified into four types [25]: supervised, unsupervised, semi-supervised, and reinforcement learning.

Most of the data that IoT devices produce are sequential time series data. This means the order of data instances matters, and instances are not independent. The main goal of time series analysis is to *learn* patterns in this kind of data to find valuable features, predict future patterns, and find out how different data streams are related. Auto-correlation is a property of time series data. This means that the current value in the time series is linked to values from the past. In linear models, the current value is a linear function of the previous values. In nonlinear models, the current value is not a linear function of the previous values. If the properties of a stochastic process change over time, it is hard to predict its future values based on what has happened in the past, and this is called a non-stationary process. Auto-Regressive Integrated Moving Average (ARIMA), Hidden Markov Models (HMMs), and Recurrent Neural Networks (RNNs) are some of the well-established methods to model time series data [26].

3. Related Work

Since this research reviews the state-of-the-art of MDE tools and languages for IoT, this section introduces other related secondary studies, i.e., surveys, systematic mapping, and systematic review articles. Table 1 presents a summary of these related works.

Table 1. Related works.

Authors	Study	Paradigm	Target System	Analyzed Papers	Years
Sabin et al., 2018	[27]	MDE	IoT	26	N/A
Felicien et al., 2020	[28]	Low-code Engineering	IoT	16	2000–2020
Muzaffar et al., 2017	[29]	MDE	Cloud computing	25	2009–2016
Abshir Mohamed et al., 2021	[30]	MDE	CPS	140	2010–2018
Casalaro et al., 2022	[31]	MDE	Mobile Robot systems	69	2000–2022
Mashkoor et al., 2021	[32]	MDE	Safety and Security Systems	95	1992–2020
Salman et al., 2020	[15]	DSL	IoT	23	2014–2020
Edsonde et al., 2021	[33]	MDE	Robotics	63	2014–2022

Sabin et al. [27] analyzed the state of the art in MDE4IoT by reviewing 26 papers using Google Scholar and the snowballing method. These authors investigated MDE techniques for addressing IoT challenges. Furthermore, they examined IoT concepts related to MDE4IoT approaches, which included: device description, discovery, deployment, self-adaptation, service composition, cloud computing, fog, edge computing, and middle-ware frameworks. The most addressed IoT concept in this review was service composition, which was identified in 16 studies reviewed.

Felicien et al. [28] showed the current state of existing low-code platforms for developing IoT systems. They analyzed 16 platforms and defined a set of features that corresponded to the functions and services that each platform supports. In particular, they focused on languages and tools available in the MDE field and the emergent low-code development platforms for the IoT domain.

Muzaffar et al. [29] discussed the use of MDE tools in cloud computing scenarios. They defined research questions regarding MDE paradigms and techniques applied in cloud computing (e.g., SaaS, PaaS, and IaaS), and the leading cloud computing tools that include some MDE features.

Abshir Mohamed et al. [30] reviewed MDE techniques, tools, and languages used in cyber-physical systems (CPSs) and IoT components that are modeled. They raised four research questions, and to answer them, they analyzed 140 papers published between 2010–2018. The domain of their study was CPS, while ours is IoT services with a focus on DAML techniques.

Casalaro et al. [31] provided a framework for the software engineering research in MDE for the particular class of mobile robotic systems (MRS). That framework includes the following aspects: the types of robots supported by existing MDE approaches; the types and characteristics of MRS that are engineered using MDE approaches; description of how MDE approaches support the engineering of MRS; how existing MDE approaches are validated; and how tools support the existing MDE approaches.

Mashkoor et al. [32] presented a systematic mapping study on MDE in the context of safety and security systems. They analyzed key questions such as what are the commonly used methods and tools in this field, their development phases, and the most common application domains that have been evaluated?

Salman et al. [15] performed a systematic literature review on DSLs for the IoT by considering four research questions: what are the most common hardware and software platforms in each domain; what are the used DSLs; how are these DSLs evaluated; and, what are the scenarios in which these DSLs were applied.

Edsonde et al. [33] selected and analyzed 63 papers focusing on DSLs and concluded that robot self-adaptation is a well-established area. Still, with the advancements in machine learning techniques, it is possible to employ model-based approaches to make robotic software even more autonomous.

Differently, our review focuses on identifying and classifying the MDE tools and languages used to develop IoT services for different application domains. This review includes analyzing the involved tools, types of model-based transformations (e.g., M2M and M2T), the transformation-supported tools used, and even the DAML techniques provided by these tools.

4. Research Methodology

There are several MDE tools and languages for developing IoT applications; however, selecting the most suitable options is always challenging, especially for new developers or researchers. From our knowledge, there is no general framework or guideline to help select which tool(s) and language(s) should support MDE4IoT activities. For instance, the current reviews on MDE4IoT do not provide a comprehensive and complete classification and analysis of the languages, model-based transformations, application domains, and even DAML techniques, as proposed in this review.

This paper presents a scoping review of research studies that have proposed or used MDE technologies for IoT applications. A *scoping review* is a type of literature review used to map the key concepts, types of studies, and sources of information within a specific research area [34,35]. It is typically used to identify the breadth and depth of existing research on a particular topic and to identify gaps in the literature that may need to be addressed in future studies. The scoping review method involves a systematic search and analysis of the literature, using predefined inclusion and exclusion criteria to identify relevant studies. The findings of a scoping review are usually presented in a narrative or tabular format and can be used to inform the design and implementation of future research studies.

A scoping review and a systematic literature review (SLR) are both types of literature reviews, but they have some key differences in terms of purpose, methodology, and output [35,36]: *Purpose*. The main purpose of a scoping review is to map the key concepts, types of studies, and sources of information within a specific research area. It is used to identify the breadth and depth of existing research on a particular topic and to identify gaps in the literature that may need to be addressed in future studies. On the other hand, the main purpose of an SLR is to summarize and critically evaluate the existing evidence on a specific research question or topic. *Methodology*. A scoping review typically has a less rigorous methodology than an SLR. Scoping reviews often use more inclusive inclusion criteria, and the search and analysis of the literature are typically less formal and less comprehensive. On the other hand, SLRs follow a more rigorous methodology, including a comprehensive search strategy, predefined inclusion and exclusion criteria, and a formal data extraction and synthesis process. *Output*. The output of a scoping review is typically presented in a narrative or tabular format and can be used to inform the design and implementation of future research studies. On the other hand, the output of an SLR is typically presented in the form of a systematic review or a meta-analysis, which summarizes and critically evaluates the existing evidence on a specific research question or topic. In summary, a scoping review is a broad overview of the existing literature on a topic and is used to identify gaps and opportunities for future research, while an SLR is a more in-depth and rigorous examination of the existing literature on a specific research question or topic, to summarize and critically evaluate that topic.

A scoping review can follow a 5-stage process, as originally discussed by Arksey and O'Malley [37], which is a rigorous process of transparency that allows replication of the search technique and enhances the dependability of the study's conclusions. As shown in Figure 1, scoping review involves a sequence of the following five stages (or tasks): (1) Identify the initial research questions; (2) identify relevant studies; (3) select the relevant studies; (4) chart the data; and (5) summarize and report the results. We ran these stages, reacted to each stage, and repeated some stages, as needed, to ensure that all of the literature was covered and analyzed. This means that the process ran iterative (as suggested in the figure), i.e., ran several times until we got confident with the results.

The current section is organized around the presentation of these stages. Stages 1 to 4 are presented in the following subsections; Stage 5 is in Section 5.

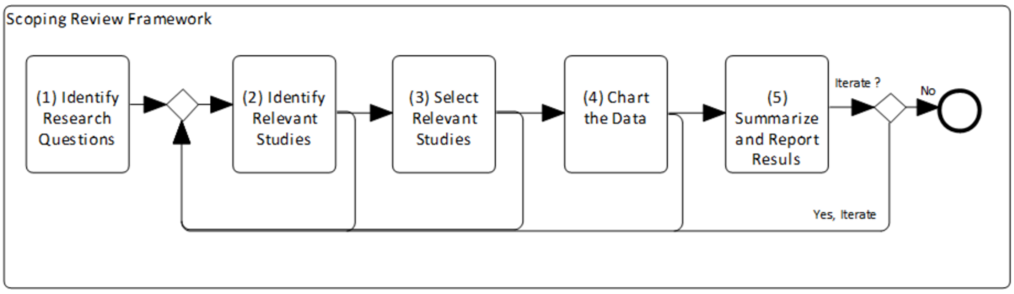
Scoping Review Framework
(1) Identify Research Questions
(2) Identify Relevant Studies
(3) Select Relevant Studies
(4) Chart the Data
(5) Summarize and Report Resuls
Iterate ?
No
Yes, Iterate

Figure 1. Followed stages of the scoping review framework.

4.1. Identify the Research Questions (Stage-1)

The first stage of the scoping review consisted of defining the initial research questions. These questions help frame and conduct the review and are the following:

RQ1. What is the current state of MDE languages and tools for developing IoT applications? This question is broken down into the following sub-questions: **RQ1.1**. What modeling languages are used? **RQ1.2**. What tools are used? **RQ1.3**. What model transformations and the respective supported languages and tools are employed? **RQ1.4**. What are the outputs of M2M and M2T transformations?

RQ2. What application domains are these MDE approaches mostly applied to?

RQ3. What DAML techniques are supported by these MDE approaches?

4.2. Identify Relevant Studies (Stage-2)

The second stage of the research aims to identify relevant studies in the field of MDE4IoT by conducting a comprehensive literature search. This stage consists of three activities: (1) defining the search expression or search terms, (2) discussing and defining the eligibility criteria, and (3) selecting databases or repositories to collect research papers.

Search expression: To obtain a broad coverage of the available literature, Arksey and O'Malley [37] recommend using a broad definition of keywords for search terms. Thus, key concepts and search terms were defined to capture most of the literature related to the MDE4IoT topic. Search tools and Boolean operators were used to narrow, broaden, and combine literature searches. The defined search expression for this study is *("model-driven development" OR "model-driven engineering" OR "model-driven architecture" OR "model-based approach" OR "model-driven approach" OR "domain specific model*" OR "metamodel") AND ("IoT*" OR "Internet of thing*").*

Eligibility criteria: To ensure the relevance of the selected papers, the following inclusion and exclusion criteria were defined:

Inclusion criteria: The study must propose at least one MDE tool, language, or technique for IoT; The study must be a peer-reviewed manuscript, i.e., a journal, conference, or workshop paper; The study must be published between 2010 and 2023.

Exclusion criteria: The study is a secondary manuscript, i.e., a survey, systematic mapping, or systematic review; The study is not applied or relevant to the IoT domains, or it does not follow an MDE approach; The study is a type of educational, editorial, tutorial, or other, i.e., it is not a scientific paper; The study is written in other languages than English.

Databases of research papers: To identify peer-reviewed literature, the following electronic databases were used: IEEE Explore (https://ieeexplore.ieee.org, accessed on 13 December 2022), ACM Digital Library (https://dl.acm.org, accessed on 13 December 2022), ScienceDirect (https://www.sciencedirect.com, accessed on 13 December 2022), Web of Science (https://clarivate.com, accessed on 13 December 2022), SCOPUS (https://www.scopus.com, accessed on 13 December 2022), and DBLP (https://dblp.org, accessed on 13 December 2022). These databases are considered the most popular and widely used in the IT domain.

4.3. Select Relevant Studies (Stage-3)

The third stage of the scoping review is the most critical and time-consuming stage of the research. This stage involves selecting the relevant studies that will be included in the review. Two research team members (ZM and AM) worked together to conduct an extensive selection of relevant studies. This stage was guided by the PRISMA framework [38] and involved the following activities:

Identifying papers: The research expression (previously defined) was used to search the selected databases, as well as Google Scholar (n = 621).

Screening the title and abstract of the papers: All the collected papers were screened based on their title and abstract, and duplicates were eliminated (n = 511). Papers that were secondary studies or not relevant to the MDE4IoT topic were also eliminated (n = 100).

Assessing the full papers: The remaining papers were assessed in full, and the eligibility criteria were applied to eliminate those papers that were not relevant (n = 68).

Final set of included papers: The final set was reached (n = 68).

Figure 2 summarizes the papers collected in each stage based on the PRISMA guideline [38]. This process of selecting relevant studies ensures that the review is comprehensive, relevant, and unbiased.

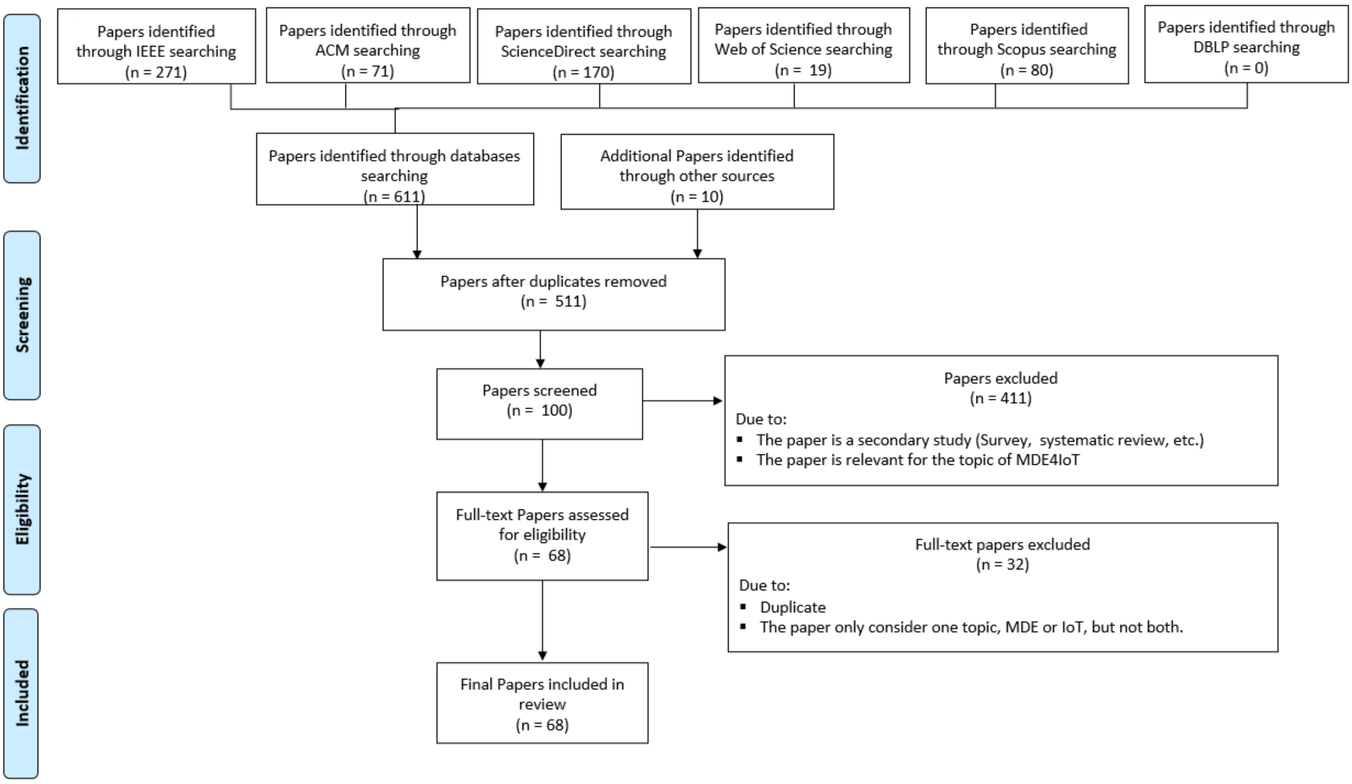
Identification
Screening
Eligibility
Included
Papers identified through IEEE searching (n = 271)
Papers identified through ACM searching (n = 71)
Papers identified through ScienceDirect searching (n = 170)
Papers identified through Web of Science searching (n = 19)
Papers identified through Scopus searching (n = 80)
Papers identified through DBLP searching (n = 0)
Papers identified through databases searching (n = 611)
Additional Papers identified through other sources (n = 10)
Papers after duplicates removed (n = 511)
Papers screened (n = 100)
Papers excluded (n = 411)
Due to:
▪ The paper is a secondary study (Survey, systematic review, etc.)
▪ The paper is relevant for the topic of MDE4IoT
Full-text Papers assessed for eligibility (n = 68)
Full-text papers excluded (n = 32)
Due to:
▪ Duplicate
▪ The paper only consider one topic, MDE or IoT, but not both.
Final Papers included in review (n = 68)

Figure 2. PRISMA flow diagram for paper selection.

4.4. Chart the Data (Stage-4)

The fourth stage of the scoping review involves charting the selected papers by creating summaries of each paper, including information on the author, year, location of study, study design, study methods, and sample size, as well as a brief comment on the limitations and recommendations of the individual study.

We reviewed the final 68 papers to confirm that the selection was appropriate. Table A1 provides an overview of these papers, including their titles, years, publication types, countries, names of journals/conferences, and the number of authors.

In terms of **distribution of papers by year and publication type** (see Figure 3), we found that 40 studies were published in conferences, 19 in journals, and 9 in workshops. Researchers in this field prefer conferences over journals, except for 2022, which showed increased journal papers.

Regarding the **distribution of papers by year and the number of authors**, Figure 4 illustrates the authorship pattern for the past thirteen years. The study classified the number of authors in each paper into six clusters: single authors, two authors, three authors, four authors, five authors, and six or more authors. It was found that 25 papers were published by four authors, 22 papers by three authors, and 7 papers by five. Additionally, 6 papers were authored by two authors and 1 paper by a single author, indicating a growing trend of collaboration among researchers.

The authors' affiliations were considered to determine the **country of origin for each publication**. As shown in Figure 5, the majority of the 68 papers were from Spain (11), Germany (9), France (8), Belgium (6), and Italy (5). Figure 5 shows the distribution of papers from the remaining 32 countries.

Regarding the **distribution of papers by year and journal**, 21 papers were published in 14 journals. Figure 6 displays the distribution of papers per journal from 2010 to 2022, showing an increase in journal publications in recent years.

Finally, Figure 7 illustrates the **distribution of papers by year of publication and digital databases**. It shows that 2017 and 2019 had the highest number of articles, with the majority of publications found in IEEExplore (36), followed by the ACM digital library (10), Science Direct (4), Web of Science (3), and other databases (14).

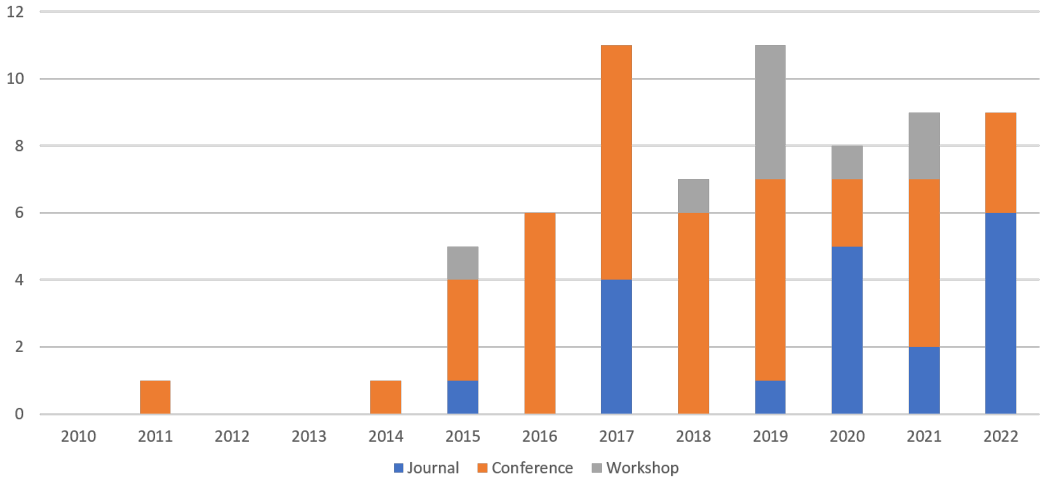

Figure 3. Distribution of papers by year and publication type.

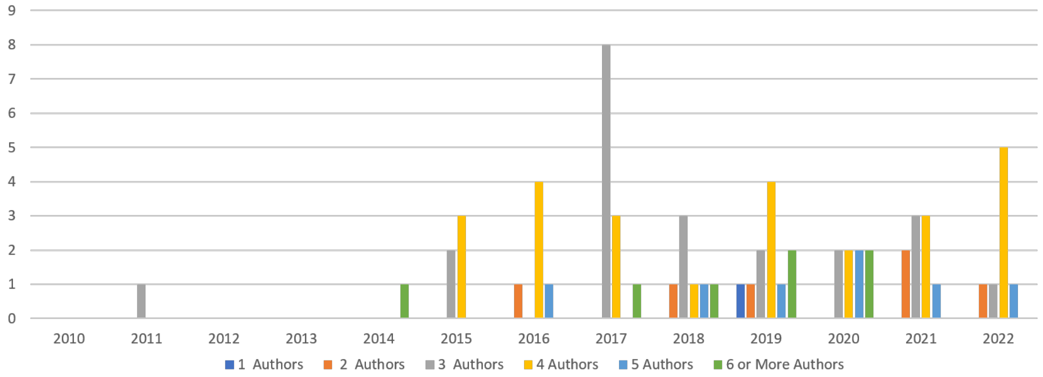

Figure 4. Distribution of papers by year and the number of authors.

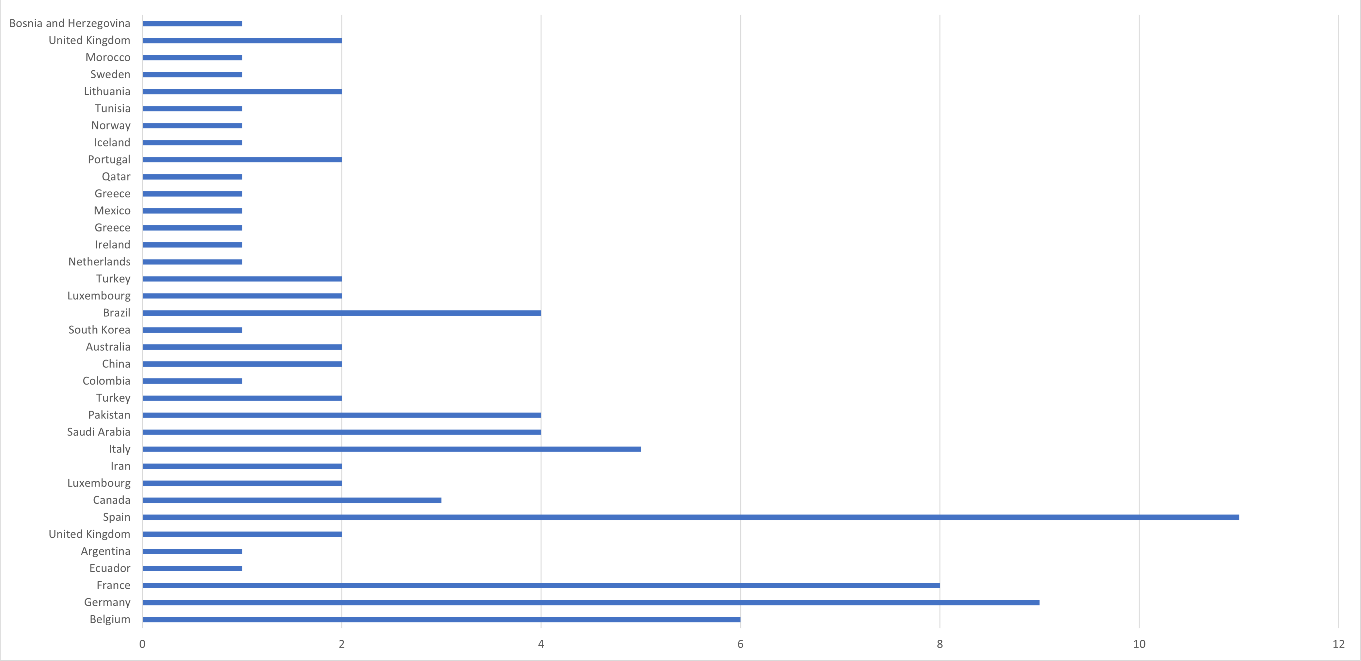

Figure 5. Distribution of papers by country.

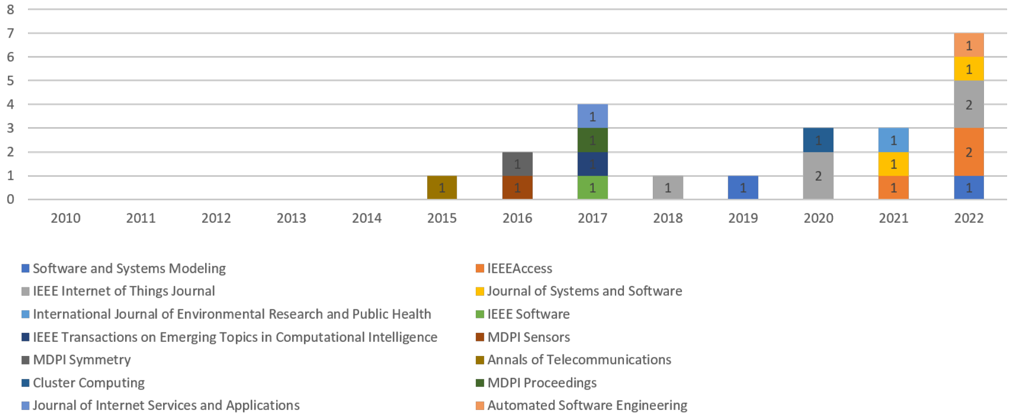

Figure 6. Distribution of papers by year and Journal.

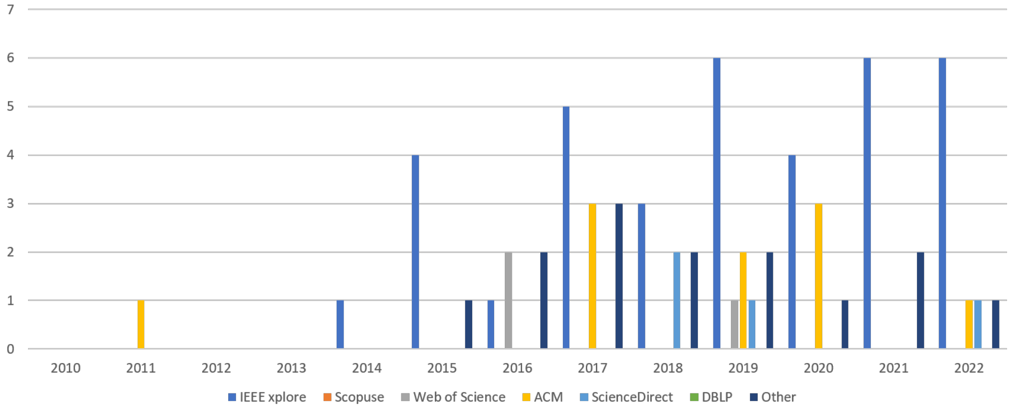

Figure 7. Distribution of papers by year and digital databases.

4.5. Summarize and Report the Results (Stage-5)

The results are summarized and reported in the fifth stage of the scoping review process, and these results are presented and discussed in Section 5.

5. Results Analysis

This section presents the results of the analysis of the responses to the research questions defined in Section 4.1.

5.1. What Is the Current State of MDE Languages and Tools for Developing IoT Applications? (RQ1)

The focus of this question is to analyze the state of the art on the MDE languages, tools, and model-based transformations used in developing IoT applications. However, this question is divided into the following sub-questions.

What modeling languages are used? (RQ1.1)

According to the MDE, the use of modeling languages helps define models with different perspectives (e.g., structure and behavior) and different levels of abstraction (e.g., platform-independent and platform-specific). These models support the design and development of complex software applications.

From the complete set of 68 reviewed studies, 29 studies reported the creation or the use of different DSLs (P1, P3, P5, P6, P7, P9, P10, P13, P14, P18, P19, P20, P24, P29, P30, P32, P35, P36, P39, P46, P41, P48, P50, P56, P59, P65, P66, P68); 11 studies use UML (P61, P55, P53, P51, P42, P38, P37, P25, P16, P11, P1); 2 studies use BPMN (P38, P41); and 5 studies reported the creation or use of UML Profiles (P29, P31, P44, P49, P52). For example, SoaML is an OMG graphical modeling language for designing systems according to service-oriented architecture (SOA), and SOAML4IOT is an extension of specific concepts for the IoT domain. Both languages are defined as UML profiles as discussed in study P29. Study P43 uses SysML to explore the application of model-based design. Study P60 describes SysML4IoT, a SysML profile based on the IoT-A reference model, and discusses model-based transformations from SysML into the NuSMV language. Study P10 describes the CHESSIoT framework that uses the CHESSML language, a UML profile also based on SysML and MARTE languages and implemented with the Papyrus workbench.

Study P38 applies the SEMIoTICS framework that directs using various modeling languages (for instance, Situation Modeling Language) and IoT platform technologies to enhance interoperability inside and between early warning systems. From a different perspective, Study P54 describes the OMG IFML (Interaction Flow Modeling Language) to support the design of software applications' user interfaces.

Table 2 summarizes the modeling languages used in the reviewed papers.

What tools are used? (RQ1.2)

A modeling language has multiple facets, such as its abstract syntax (i.e., its concepts, relationships, and constraints) and concrete syntax (i.e., its textual or graphical concrete representation). These facets are implemented and supported by specific tools commonly known as language workbenches or language frameworks [13,14,39]. Popular language workbenches that provide graphical editors are DiaGen, Eugenia, GMF, Graphiti, MetaEdit+, Obeo Designer, and Sirius [40], while those that support textual languages are Xtext and MetaEdit. Six (6) studies (P1, P4, P21, P32, P46, P56) refer to the use of textual languages implemented with the Xtext framework, while, regarding visual modeling languages (i.e., with graphical concrete syntax), 7 studies reported the use of Sirius (P3, P10, P14, P12, P22, P24, P35), 4 studies the use of Eugenia (P5, P7, P13, P50), 6 studies the use of GMF (P2, P7, P13, P65, P67, P50), and other studies report the use of language workbenches like Obeo Designer (P12) and MetaEdit+ (P39).

Table 3 presents the studies that refer to the implementation or use of textual and graphical modeling languages.

Table 2. Distribution of papers by modeling language.

PID	Framework	DSLs	UML	BPMN	SysMl	UML Profile	Other
P1	ML-Quadrat (ML2)	✓	✓				
P3	Monitor-IoT	✓					
P5	SimulateIoT-FIWARE	✓					
P6	CyprIoT-DSL	✓					
P7		✓					
P9	HealMA	✓					
P10	CHESSIoT						✓
P11	FTG+PM	✓	✓				
P13	SimulateIoT	✓					
P14		✓					
P16	FaultFlow		✓				
P18		✓					
P19	DSML4contiki	✓					
P20		✓					
P22		✓					
P24		✓					
P25			✓				
P29	SoaML4IoT	✓				✓	✓
P30		✓					✓
P31						✓	
P32	Cypriot	✓					
P35		✓					
P36		✓					
P37			✓				
P38	Semiotics		✓	✓			✓
P39	SiMoNa	✓					
P41				✓			
P42			✓				
P43					✓		
P44	UMLOA					✓	
P46		✓					
P47	SmartHomeML	✓					
P48		✓					
P49	UML4IoT					✓	
P50		✓					
P51	ThingML		✓				
P52	IoTA-MD					✓	
P53			✓				
P54							✓
P55			✓				
P56		✓					
P59		✓					
P60	SysML4IoT				✓		
P61	MDE4IoT		✓				
P65	FRASAD	✓					
P66		✓					
P68		✓					
Total		29	11	2	2	5	5

Table 3. Distribution of papers by textual and graphical modeling language tools.

		Textual Modeling Language Tools		**Graphical Modeling Language Tools**				
PID	**Framework**	**Xtext**	**MontiCore**	**Sirius**	**Obeo Designer**	**GMF**	**Metaedit+**	**Eugenia**
P1	ML-Quadrat (ML2)	✓						
P2						✓		
P3	Monitor-IoT			✓				
P4		✓						
p5	SimulateIoT-FIWARE							✓
P6	CyprIoT-DSL							
P7						✓		✓
P9	HealMA			✓				
P10	CHESSIoT							
P12				✓	✓			
P13	SimulateIoT					✓		✓
P14				✓				
P21	IoTSuite	✓						
P22				✓				
P24				✓				
P30			✓					
P32	Cypiot	✓						
P35				✓				
P39	SiMoNa						✓	
P46	EL4IoT	✓						
P50						✓		✓
P56		✓						
P65	FRASAD					✓		
P67						✓		
Total		6	1	7	1	6	1	4

What model transformations and the respective supported languages and tools are employed? (RQ1.3)

Model transformations are used in MDE to convert models from one representation to another. There are various languages and tools that have been used to perform these transformations. For instance, Meta3, Xpand, and Xtend covered both M2M and M2T transformation, while Acceleo and JET covered only M2T transformation. However, despite their popularity, some of these frameworks, like the JET (Java Emitter Templates) are not maintained anymore.

Twenty-five (25) studies refer to using some transformations. Eighteen (18) studies refer using M2T (P1, P5, P6, P7, P10, P11, P13, P14, P22, P23, P24, P25, P26, P27, P36, P47, P62, P68) and 14 studies report using M2M transformations (P5,P6, P11, P15, P19, P23, P26, P27, P28, P41, P52, P58, P62, P68). Only 8 studies refer to using both M2T and M2M transformations. Eleven (11) studies refer the implementation of M2T transformations with Accelo (P5, P6, P10, P13, P14, P19, P22, P24, P36, P62, P67) and 7 studies refer M2M transformation with the ATL framework (P5, P6, P26, P41, P42, P52, P62).

Table 4 shows the distribution of papers by types of model-based transformations and the supported tools.

What are the outputs of M2M and M2T transformations? (RQ1.4)

M2M and M2T transformations are techniques used in MDE to convert models from one representation to another. The output of an M2M transformation is a new model in a different format or representation. This new model is typically semantically equivalent to the original but may have a different structure or syntax. On the other hand, the output of an M2T transformation is text, typically in the form of source code, but it also can be in the form of documentation, configuration files, or other textual artifacts. The text generated is typically based on the structure and/or the information contained in the original model.

In general, M2M transformations are used to transform models from one meta-model to another, for example, from UML to SysML, or from one notation to another, for example, from BPMN to EPC. In comparison, M2T transformations generate source code, documentation, or other textual artifacts from a model. Both M2M and M2T transformations can be performed using a variety of languages and tools, such as QVT, ATL, Acceleo, and Xtend.

For example, Study P10 includes an advanced multi-platform code generation framework that supports multiple target programming languages, such as C, C++, Java, Arduino, and JavaScript. Study P11 uses the ATL framework and discusses M2M and M2T transformations of platform-specific models and software code for systems like RIOT, Arduino, Contiki, and TinyOS, with code written for multiple programming languages C, nesC, Java, JSON, XML. Study P14 discusses a semi-automatic process for generating Ballerina code for RESTful and orchestrator services, Arduino code for deploying IoT nodes, NCL-Lua code for the DTV interface, and Android code for smartphones. Study P24 reports using Acceleo to generate code for programming languages like Flutter, React JS, and VHDL. Studies P8, P56, P59, and P67 discuss M2T transformations with the generation of C++ or Java code. Study P62 discusses M2T transformations that generate Java code and SQL DDL scripts from JavaSE and database models.

5.2. What Application Domains Are These MDE Approaches Applied to? (RQ2)

MDE approaches and tools are applied to a wide range of application domains in IoT, including for Smart Home Automation, Industrial Internet of Things, Smart Grid, Healthcare, Agriculture, Predictive maintenance [41].

However, from the selected papers the most popular domains are the followings: smart buildings and living, smart healthcare, smart environment, smart cities, smart energy, smart transport, and mobility. Table 5 shows the distribution of papers by application domains.

Eight (8) studies show or discuss MDE4IoT techniques applied into the Smart Manufacturing domain (P4, P7, P10, P23, P24, P51, P53, P30); 10 studies into Smart healthcare domain (P4, P7, P10, P23, P24, P51, P53, P30); 3 studies into Smart Cities (P48, P61, P16); 5 studies into Smart Environment (P19, P35, P45, P62, P50); 2 studies into Smart Energy (P1, P39); 16 studies into Smart buildings and living (P1, P5, P7, P8, P13, P14, P18, P20, P21, P22, P27, P28, P32, P36, P47, P53); 6 studies into Smart Transportation (P6, P26, P37, P38, P44, P64); and 3 studies into Smart Agricultural (P11, P43, P25).

Smart Manufacturing aims to increase factories' efficiency by enhancing manufacturing, process control, and monitoring systems. *Smart healthcare* aims to enhance medical systems, making them more efficient, convenient, and personalized, by supporting remote patient monitoring, telemedicine, and medical device management systems [42]. *Smart cities* is a new domain that not only improves the quality of life in a city but also lowers the operating costs of public services like transportation, parking, and waste management systems [43]. *Smart Environment* involves rare event detection, is in high demand because it can warn against natural disasters; may also monitor various environmental factors such as air quality, noise level, and weather conditions, to improve the overall quality of life in cities and other areas. *Smart Energy* refers to the improvements in the distribution and consumption of utilities such as electricity, gas, and water, and may involve control and monitoring lighting in buildings, streets, and other public areas, to increase energy efficiency and reduce costs [41]. *Smart buildings and living* aims to make buildings more energy-efficient and enhance the quality of life of their users [41]. *Smart transport* is a method of incorporating modern technologies into transportation systems. Cloud computing, wireless communication, location-based services, computer vision, and other mobility-enhancing tools are all part of this [44]. *Smart Agricultural* refers to systems that may monitor and control various aspects of agriculture, such as irrigation, fertilization, and crop management, to increase efficiency, reduce environmental impact and increase crop yield [41].

Table 4. Distribution of papers by types of model-based transformations and tools.

		Transformation		Tools for Transformation			
PID	**Framework**	**M2T**	**M2M**	**Accelo**	**ATL**	**M2T Output**	**M2M Output**
P1	ML-Quadrat	✓				Java, Python	
P2						Intermediary Macro-Code Generation, Code Generation for Node-RED Target, Java code application	
P3							
P4		✓				C for Arduino, Python	
P5	SimulateIoT-FIWARE	✓	✓	✓	✓		
P6	CyprIoT-DSL	✓	✓	✓	✓	C, Java, Arduino, AC rules and documentation	
P7		✓				Java, configuration code	
P8	MontiThings					C++	
P9	HealMA	✓		✓		Java, XML	
P11	FTG+PM	✓	✓			C, nesC, Java, JSON, XML, GenerateContiki Code, GenerateTinyOSCode, GenerateJavaCode, GeneratePetrinetConfiguration, GenerateArduinoCode, GenerateRIOTCode	RIOTMode, SystemModel, GenerateContikiModel, GenerateTinyOSModel, GenrateGatewayModel, GeneratePetrinetModel, GenerateNodeRedModel, GenerateArduinoModel, GenerateRIOTModel, GenerateRIOTModel
P13	SimulateIoT	✓		✓		Java, configuration code	
P14		✓		✓		Java, Xml, Ardunio code, json RESTful APIs Associated	
P15	MoSIoT		✓				
P19	DSMl4Contiki		✓	✓			Petri-net Models
P22		✓		✓		Arduino code files (.ino), code Node-Red(.json), Ballerina code (.bal)	
P23	AutoIoT	✓	✓				
P24		✓		✓		Flutter, React JS, And VHDL.	
P25		✓					
P26		✓	✓		✓		
P27		✓	✓				
P28	BRAIN-IoT		✓			Java, Osgi Artifact, C	
P33	GreyCat					Java, TypeScript.	
P35						Code for Mote (SourceNode), SinkNode, Raspberry Pi (Java Code), ESP8266 (Arduino code), configuration for IoT Log Manager	Petri-net Models
P36		✓		✓		Ardunio code	
P41			✓		✓		
P42					✓		
P47	SmartHomeML	✓					
P51	ThingML					C/C++, Java, and JavaScript and several libraries and open platforms (Arduino, Raspberry Pi, Intel Edison, Linux, and so on).	
P52			✓		✓		
P56						Java	
P58			✓				
P59						Java	
P61	MDE4IoT					C	
P62			✓	✓	✓	java code and SQL DDL	
P67				✓		Java	
P68		✓	✓				
Total		18	14	11	7		

Table 5. Distribution of papers by application domains.

PID	Framework	Healthcare	Agricultural	City	Energy	Manufacturing	Building	Environment	Transport
P1	ML-Quadrat (ML2)				✓		✓		
P4						✓			
P5	SimulateIoT-FIWARE						✓		
P6	CyprIoT-DSL						✓		✓
P7						✓	✓		
P8							✓		
P9	HealMA	✓							
P10	CHESSIoT					✓			
P11	FTG+PM		✓						
P13	SimulateIoT		✓				✓		
P14							✓		
P15	MoSIoT	✓							
P16	FaultFlow			✓					
P19	DSMl4Contiki							✓	
P20							✓		
P21							✓		
P22							✓		
P23	AutoIoT					✓			
P24						✓			
P25			✓						
P26									✓
P27							✓		
P28	BRAIN-IoT						✓		
P30						✓			
P31		✓							
P32	Cypriot						✓		
P35								✓	
P36							✓		
P37									✓
P38	Semiotics								✓
P39	SiMoNa				✓				
P40						✓			
P43		✓							
P44	UMLOA								✓
P45								✓	
P47	SmartHomeML						✓		
P48				✓					
P50								✓	
P51	ThingML	✓				✓			
P52	IoTA-MD	✓							
P53							✓		
P54		✓							
P55		✓							
P57		✓							
P58		✓							
P61	MDE4IoT			✓					
P62								✓	
P64									✓
Total		10	3	3	2	8	16	5	6

5.3. What DAML Techniques Are Supported by These MDE Approaches? (RQ3)

DAML methods and techniques are crucial for developing smart IoT services and CPS applications [45]. After analyzing these studies, we identified only four studies that discussed the application of DAML techniques in the scope of MDE4IoT: P1, P4, P33, and P15.

Study P1 introduced the ML-Quadrat (ML2) framework that supports supervised ML with different techniques such as Logistic Regression for linear classification, Linear Regression, Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, Bernoulli Naive Bayes, Categorical Naive Bayes, Decision tree Regressor and Decision Tree Classifier, the random forest regressor and Random Forest Classifier ensemble methods, The Multi-Layer Perceptron (MLP) Artificial Neural Networks (ANN). ML2 also supports unsupervised ML-included techniques, like K-Means, Mini-Batch K-Means, DB-SCAN, spectral clustering, and the Gaussian Mixture Model, and semi-supervised ML support techniques like self-training, label propagation, and label spreading (See Table 6).

Table 6. Data Analytics and Machine Learning in MDE4IoT.

		Types of Machine Learning Algorithms			
PID	**Framework**	**Supervised**	**Unsupervised**	**Semi Supervised**	**Time Series Models**
P1	ML-Quadrat(ML2)	Logistic Regression Linear Regression, Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, Bernoulli Naive Bayes, Categorical Naive Bayes, Decision Tree Regressor Decision Tree Classifier, Decision Tree Regressor Decision Tree Classifier, Random Forest Classifier, Multi-Layer Perceptron (MLP) Artificial Neural Networks (ANN)	K-Means, Mini-Batch K-Means, DB-SCAN, Spectral Clustering, Gaussian Mixture Model,	Self-Training, Label Propagation, Scikit-Learn Label Spreading	NO
P4		Logistic Regression Linear Regression, Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, Bernoulli Naive Bayes, Categorical Naive Bayes, Decision Tree Regressor Decision Tree Classifier, Decision Tree Regressor Decision Tree Classifier, Random Forest Classifier, Multi-Layer Perceptron (MLP) Artificial Neural Networks (ANN)	K-Means, Mini-Batch K-Means, DB-SCAN, Spectral Clustering, Gaussian Mixture Model,	Self-Training, Label Propagation, Scikit-Learn Label Spreading	NO
P15	MoSIoT		Live linear regression, Live decision trees Naive Bayesian models Gaussian Bayesian models baz	KNN, StreamKM++ Gaussian Mixture Models	NO
P33	GreyCat		Live linear regression, Live decision trees Naive Bayesian models Gaussian Bayesian models baz	KNN, StreamKM++ Gaussian Mixture Models	NO

Study P4 proposes a novel approach to software and AI engineering for smart IoT services that allow data analytics and machine learning components to run in part or entirely on IoT edge devices, which may be severely limited in terms of power and computational resources. The proposed approach for software development is based on the

DSM methodology of the MDSE paradigm. In particular, it builds on the previous work of ML-Quadrat [P1].

Study P15 proposed the MoSIoT framework, which integrates a prediction module into the MoSIoT framework that applies an integrated set of learning attributes to the scenario model related to ML algorithms. This framework applied an approach based on GreyCat, which seamlessly integrated machine-learning algorithms into a domain model.

Study P33 integrated ML algorithms with software models. This work implemented the following algorithms: (1) Regression: Live linear regression; (2) Classification: Live decision trees, Naive Bayesian models, Gaussian Bayesian models; and (3) Profiling: Gaussian Mixture Models (Simple Multinomial).

Due to the significance of time series forecasting for IoT systems, we examined these studies to determine whether they discussed or proposed time series modeling techniques. However, our analysis shows that none of the studies has yet considered time series modeling techniques.

6. Discussion

In addition to the thorough evaluation of the outcomes of the scoping review provided in Section 5, this section includes a general discussion of the acquired findings and their consequences.

6.1. Main Findings

Regarding modeling languages addressed by RQ1, results from RQ1.1 showed that the top 3 most used modeling languages for IoT applications are: DSLls, UML, and UMLProfile. The most popular type of modeling language is DSLs which is reported in 29 studies. Then, UML with 11 studies, and UML Profile with 5 studies reported.

Concerning question RQ1.2, our analysis revealed that 6 studies used Xtext and MontiCore with 1 study reported for textual modeling language tools. Regarding graphical modeling language tools, 7 studies reported using Sirius, 6 using GMF, 4 using Eugenia, 1 using Metaedit+, and 1 using Obeo-Designer. These findings showed that graphical modeling language makes it easy to understand because most people benefit from a visual aid.

The analysis of the results from RQ1.3 showed that the M2T transformation, with 18 studies, gains more attention in terms of the existing or developed tools and languages compared to the M2M transformation with only 14 studies. In addition, it is observed that tools like Accelo and ATL are the most popular for model-based transformations, being Accelo the most tool for M2T transformation, as reported in 10 studies.

For the application domain, RQ2 results revealed that most studies reported IoT applications in the smart buildings and living application domains, which were mentioned in 16 studies out of 68. The other application domains were smart healthcare (with 10 studies), smart environment (with 5 studies), smart cities (with 3 studies), smart manufacturing (with 8 studies), smart transport and mobility (with 6 studies), smart agricultural and smart energy (with 2 studies).

Regarding question RQ3, concerning the use of DAML techniques in MDE4IoT, We find 4 studies among all of the analyzed studies. Few researchers have developed supervised machine learning techniques (reported in only 2 studies), unsupervised machine learning (4 studies), and semi-supervised machine learning (4 studies) for this purpose. We also verify that none of the studies supports time series forecasting. The results show that the total number of studies that reported using DAML is quite low (only 4 studies out of 68) for a wide and complex domain like the IoT.

6.2. Study Limitations

The potential of missing primary research frequently threatens any literature review. The databases we checked might not have included all the pertinent studies. However, to increase external validity, we employed several strategies. First, we looked at the author profiles of the most prolific authors to see if we missed any relevant studies. Second, because we were all familiar with the subject, we talked about whether we were aware of any studies that might have been relevant but were suspiciously missing from the sample. Third, we looked at the first 200 Google Scholar results from the same search.

However, despite these safeguards, it is possible that some pertinent studies were still overlooked. The results are biased toward publishing because we decided not to include unpublished work. Still, based on this work, we can observe some research issues, which are mentioned in the next section as future work.

7. Conclusions

MDE approaches may positively affect the quality of IoT systems easily by managing the increasing complexity of such IoT systems. MDE helps easily integrate various technologies (or migrate to new technologies) and make them compatible with the required changes, as well as the maintenance and documentation of the facilities.

The scoping review presented in this paper aimed to analyze and discuss the current state of the art of the MDE applied to the IoT domain. To do this, we selected 68 papers in the field, published between 2010 and 2022, and focused on languages, tools, application domains, and data analytics and machine learning techniques in the MDE field, specifically related to the IoT domain.

Our findings indicate that DSLs were favored above other modeling languages. Xtext and Sirius are the most popular tools for both textual and graphical modeling languages, respectively. Regarding the application domains, smart buildings and living are the most commonly used domains for IoT applications, namely, in what concerns the studies analyzed. Additionally, our review found that few studies have proposed MDE approaches for DAML techniques, and none have considered techniques for analyzing time series data.

Based on our review, it is clear that there is a need for further research in the area of MDE for IoT. Specifically, studies need to investigate the use of different types of modeling languages and tools for IoT systems and explore the various application domains where MDE can be applied.

Additionally, research is needed to explore integrating DAML techniques with MDE for IoT, specifically concerning time series data. This could include developing new methods and techniques for effectively representing and managing the complexity of time series data in the MDE process and for integrating DAML techniques into the MDE process to allow for efficient and effective analysis and processing of time series data. Time series data can be highly dynamic and involve many variables, making creating accurate and comprehensive models difficult. Another challenge is the need to integrate DAML techniques into the MDE process in a way that allows for the efficient and effective analysis and processing of time series data. This may require developing specialized methods and approaches for working with DAML in MDE for IoT services. A further challenge is the need to ensure the scalability and performance of MDE for IoT services when working with large amounts of time series data. This may involve specialized techniques and tools for optimizing the MDE process in this context.

Author Contributions: Conceptualization, Z.M.K., A.M., A.R.d.S., and J.C.F.; methodology, Z.M.K., A.M., and A.R.d.S.; validation, Z.M.K., A.M., and A.R.d.S.; investigation, Z.M.K., A.M., and A.R.d.S.; resources, Z.M.K., A.M., and A.R.d.S.; writing—original draft preparation, Z.M.K.; writing—review and editing, Z.M.K., A.M., and A.R.d.S.; project administration, Z.M.K., A.M., and A.R.d.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: Research partially funded by the following programs: FCT UIDB/50021/2020, European Union´s Horizon Europe research and innovation programme under grant agreement no. 101056765, and EEA Grants Blue Growth Programme (Call #5), Project PT-INNOVATION-0069-Fish2Fork. The views and opinions expressed in this document are those of the authors only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and

Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Included studies in the scoping review.

PID	Study	Title	Year	Type	Country	Name of Journal/Conference	Number of Authors
P1	[46]	A model-driven approach to machine learning and software modeling for the IoT	2022	Journal	Germany, Belgium	SoSyM	4
P2	[47]	IoT efficient data exploitation process using Model Driven Engineering	2022	Conference	France	ICC	5
P3	[48]	A Domain-Specific Language for Modeling IoT System Architectures That Support Monitoring	2022	Journal	Argentina, Ecuador	lEEE Access	4
P4	[49]	Supporting AI Engineering on the IoT Edge through Model-Driven TinyML	2022	Conference	Germany, Belgium, United Kingdom	COMPSAC	4
P5	[50]	SimulateIoT-FIWARE: Domain Specific Language to Design, Code Generation and Execute IoT Simulation Environments on FIWARE	2022	Journal	Spain	IEEE Access	4
P6	[51]	A Model-Driven Methodology to Accelerate Software Engineering in the Internet of Things	2022	Journal	France, Canada, Luxembourg	IoT-J	4
P7	[52]	Designing and simulating IoT environments by using a model-driven approach	2022	Conference	Spain	CISTI	2
P8	[53]	MontiThings:Model-Driven Development and Deployment of Reliable IoT Applications	2022	Journal	Germany	JSS	4
P9	[54]	HealMA: a model-driven framework for automatic generation of IoT-based Android health monitoring applications	2022	Journal	Iran, Canada	ASE	3
P10	[55]	Towards a modeling and analysis environment for industrial IoT systems	2021	Workshop	Italy	CEUR	4
P11	[56]	FTG+PM for the Model-Driven Development of Wireless Sensor Network based IoT Systems	2021	Conference	Belgium, Canada	MODELS	3
P12	[57]	A Model-Driven Framework for Early Analysis of Kill Chain Attacks	2021	Conference	Pakistan, Saudi-Arabia	NCCC	5
P13	[58]	SimulateIoT: Domain Specific Language to Design, Code Generation and Execute IoT Simulation Environments	2021	Journal	Spain	IEEE Access	4
P14	[59]	An approach to integrate IoT systems with no-web interfaces	2021	Conference	Spain	ICITS	3
P15	[60]	MoSIoT: Modeling and Simulating IoT Healthcare-Monitoring Systems for People with Disabilities	2021	Journal	Spain	IJERPH	4
P16	[61]	FaultFlow: A tool supporting an MDE approach for Timed Failure Logic Analysis	2021	Conference	Italy	EDCC	3
P17	[62]	The Need for Model-driven Engineering in the Development of IoT Software for Public Transportation Systems	2021	Conference	Turkey	UYMS	2
P18	[63]	Model-driven Development for ESP-based IoT Systems	2021	Workshop	Belgium	SERP4IoT	2
P19	[64]	Analyzing WSN-based IoT Systems Analyzing WSN-based IoT Systems using MDE Techniques and Petri-net Models	2020	Conference	Belgium	MODELS	4
P20	[65]	A Model-Driven Approach to Unravel the Interoperability Problem of the Internet of Things	2020	Conference	France, Canada	AINA	5
P21	[66]	IoTSuite: A ToolSuite for Prototyping Internet of Things Applications.	2020	Workshop	Spain	AINA	4

Table A1. *Cont.*

PID	Study	Title	Year	Type	Country	Name of Journal/Conference	Number of Authors
P22	[10]	A model-driven engineering approach for the service integration of IoT systems	2020	Journal	Spain	Cluster computing	5
P23	[8]	AutoIoT: a Framework based on User-driven MDE for Generating IoT Applications	2020	Conference	Germany	SAC	3
P24	[67]	A Model-Driven Approach for IoT-Based Monitoring Systems in Industry 4.0	2020	Conference	Iran	SCIOT	3
P25	[68]	A Conceptual Data Model and Its Automatic Implementation for IoT-Based Business Intelligence Applications	2020	Journal	France, China, Colombia	IoT-J	9
P26	[69]	An Application Development Framework for Internet-of-Things Service Orchestration	2020	Journal	China, Saudi-Arabia, South-Korea, Australia	IoT-J	6
P27	[70]	A Model-Driven Framework for Ensuring Role Based Access Control in IoT Devices.	2020	Conference	Pakistan	ICCAI	4
P28	[71]	BRAIN-IoT: Model-Based Framework for Dependable Sensing and Actuation in Intelligent Decentralized IoT Systems	2019	Conference	Italy, Spain	IoT-J	6
P29	[72]	Modeling SOA-Based IoT Applications with SoaML4IoT	2019	Conference	Brazil	WF-IoT	3
P30	[73]	Towards privacy-preserving IoT systems using model-driven engineering	2019	Workshop	Germany	MODELS	4
P31	[69]	A Model-Driven Approach for Load-Balanced MQTT Protocol in the Internet of Things (IoT)	2019	Conference	Pakistan, Saudi Arabia.	CISIS	4
P32	[74]	Cypriot: Framework for modelling and controlling network-based IoT applications	2019	Conference	Canada, France	SAC	5
P33	[75]	The next evolution of MDE: a seamless integration of machine learning into domain modeling	2019	Journal	Luxembourg	SoSyM	4
P34	[76]	Model-driven evidence-based privacy risk control in trustworthy smart IoT systems	2019	Workshop	Spain, France	CEUR-WS.org	4
P35	[77]	Applying model driven engineering techniques to the development of Contiki-based IoT systems	2019	Workshop	Turkey	SERP4IoT	6
P36	[78]	A model-driven approach for the integration of hardware nodes in the IoT	2019	Conference	Spain	WorldCIST	3
P37	[79]	Enabling Model-Driven Software Development Tools for the Internet of Things	2019	Workshop	Canada	MiSE	2
P38	[80]	Semiotics: Semantic model-driven development for IoT interoperability of emergency services	2019	Conference	Netherland	ISCRAM	1
P39	[81]	SiMoNa:A Proof-of-concept Domain-Specific Modeling Language for IoT Infographics	2018	Conference	Brazil, Ireland	ISCRAM	4
P40	[82]	Towards an IoT-based Framework for Evolvable Assembly Systems	2018	Conference	Greece	IFAC	3
P41	[83]	A Methodology Based on Model-Driven Engineering for IoT Application Development	2018	Conference	Mexico	ICDS	3
P42	[84]	A Model-Driven Workflow for Energy-Aware Scheduling Analysis of IoT-Enabled Use Cases	2018	Journal	Germany	IoT-J	2
P43	[85]	A model-based approach for managing criticality requirements in e-health IoT systems	2018	Conference	Qatar, Greece	SoSE	6
P44	[86]	A Model-Driven Approach for Access Control in the Internet of Things (IoT) Applications UMLOA	2018	Conference	Pakistan	ICIST	5
P45	[87]	A Reactive and Model-Based Approach for Developing Internet-of-Things Systems	2018	Conference	Portugal	QUATIC	3
P46	[88]	A modeling domain-specific language for IoT-enabled operating systems	2017	Conference	Portugal	InIECON	7
P47	[89]	SmartHomeML: Towards a domain-specific modeling language for creating smart home applications	2017	Conference	Iceland, Canada	ICIOT	4
P48	[90]	A Domain Specific Language for Smart Cities	2017	Journal	Spain	MDPI Proceedings	3
P49	[91]	A framework for MDE of IoT-Based Manufacturing Cyber-Physical Systems	2017	Conference	Greece	IoT	3

Table A1. *Cont.*

PID	Study	Title	Year	Type	Country	Name of Journal/Conference	Number of Authors
P50	[92]	Modelling Contiki-Based IoT Systems	2017	Conference	Turkey	SLATE	4
P51	[93]	Model-Based Software Engineering to Tame the IoT Jungle	2017	Journal	Norway	IEEE Software	3
P52	[94]	IoTA-MD: A model-driven approach for applying QoS attributes in the development of the IoT systems	2017	Conference	Brazil	SAC	3
P53	[91]	A framework for MDE of IoT-Based Manufacturing Cyber-Physical Systems	2017	Conference	Greece	IoT	3
P54	[16]	Model-driven development of user interfaces for IoT systems via domain-specific components and patterns	2017	Journal	Italy	JISA	3
P55	[95]	A Model-Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare	2017	Journal	France	TETCI	3
P56	[96]	An IoT domain meta-model and an approach to software development of IoT solutions	2017	Conference	Tunisia	IINTEC	3
P57	[97]	Model-driven approach for body area network application development	2016	Journal	Lithuania	MDPI Sensors	4
P58	[97]	A model-driven framework to develop personalized health monitoring	2016	Journal	Lithuania	MDPI Symmetry	4
P59	[98]	A model-driven engineering approach for the well-being of ageing people	2016	Conference	Belgium	ER	4
P60	[99]	Design and Analysis of IoT Applications: A Model-Driven Approach	2016	Conference	Brazil, Australia	DASC/ PiCom/ DataCom/ CyberSciTech	5
P61	[100]	Supporting the Internet of Things with Model-Driven Engineering	2016	Conference	Sweden	IDC	2
P62	[101]	A model-driven architecture-based data quality management framework for the Internet of Things	2016	Conference	Morocco, France	CloudTech	4
P63	[102]	Model-driven interoperability: engineering heterogeneous IoT systems	2015	Journal	UK	Annals of Telecommunications	3
P64	[103]	A model-based approach across the IoT lifecycle for scalable and distributed smart applications	2015	Conference	Italy	ITSC	3
P65	[104]	A Framework for Model-driven IoT Application Development	2015	Conference	Germany	WF-IoT	4
P66	[105]	Design of a domain-specific language and IDE for Internet of things applications	2015	Conference	Bosnia-Herzegovina	MIPRO	4
P67	[106]	IoTLink: An Internet of Things Prototyping Toolkit. 2014	2014	Conference	Germany, Brazil	UIC-ATC	7
P68	[107]	Model-driven development for data-centric sensor network applications	2011	Conference	Germany	MoMM	3

References

1. Munirathinam, S. Industry 4.0: Industrial internet of things (IIOT). In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2020; Volume 117, pp. 129–164.
2. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A survey on IoT security: Application areas, security threats, and solution architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
3. Lee, I.; Lee, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Bus. Horiz.* **2015**, *58*, 431–440. [CrossRef]
4. Akkaş, M.A.; Sokullu, R.; Çetin, H.E. Healthcare and patient monitoring using IoT. *Internet Things* **2020**, *11*, 100173. [CrossRef]
5. Katare, G.; Padihar, G.; Qureshi, Z. Challenges in the integration of artificial intelligence and internet of things. *Int. J. Syst. Softw. Eng.* **2018**, *6*, 10–15.
6. Verma, S.; Kawamoto, Y.; Fadlullah, Z.M.; Nishiyama, H.; Kato, N. A survey on network methodologies for real-time analytics of massive IoT data and open research issues. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1457–1477. [CrossRef]
7. Tsai, C.W.; Lai, C.F.; Chiang, M.C.; Yang, L.T. Data mining for internet of things: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 77–97. [CrossRef]
8. Nepomuceno, T.; Carneiro, T.; Maia, P.H.; Adnan, M.; Nepomuceno, T.; Martin, A. AutoIoT: A framework based on user-driven MDE for generating IoT applications. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Virtual, 30 March–3 April 2020; pp. 719–728.

9. Ciccozzi, F.; Crnkovic, I.; Di Ruscio, D.; Malavolta, I.; Pelliccione, P.; Spalazzese, R. Model-driven engineering for mission-critical iot systems. *IEEE Softw.* **2017**, *34*, 46–53. [CrossRef]
10. Alulema, D.; Criado, J.; Iribarne, L.; Fernández-García, A.J.; Ayala, R. A model-driven engineering approach for the service integration of IoT systems. *Clust. Comput.* **2020**, *23*, 1937–1954. [CrossRef]
11. Frank, U. Domain-specific modeling languages: Requirements analysis and design guidelines. In *Domain Engineering*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 133–157.
12. Jácome-Guerrero, S.P.; Ferreira, M.; Corral, A. Software development tools in model-driven engineering. In Proceedings of the 2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT), Merida, Mexico, 25–27 October 2017; pp. 140–148.
13. Iung, A.; Carbonell, J.; Marchezan, L.; Rodrigues, E.; Bernardino, M.; Basso, F.P.; Medeiros, B. Systematic mapping study on domain-specific language development tools. *Empir. Softw. Eng.* **2020**, *25*, 4205–4249. [CrossRef]
14. Da Silva, A.R. Model-driven engineering: A survey supported by the unified conceptual model. *Comput. Lang. Syst. Struct.* **2015**, *43*, 139–155.
15. Salman, A.J.; Al-Jawad, M.; Al Tameemi, W. Domain-specific languages for IoT: Challenges and opportunities. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Kerbala, Iraq, 5–6 December 2020; IOP Publishing: Bristol, UK, 2021; Volume 1067, p. 012133.
16. Brambilla, M.; Cabot, J.; Wimmer, M. Model-driven software engineering in practice. *Synth. Lect. Softw. Eng.* **2017**, *3*, 1–207.
17. Stahl, T.; Völter, M.; Czarnecki, K. *Model-Driven Software Development: Technology, Engineering, Management*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006.
18. Brown, A.W. Model driven architecture: Principles and practice. *Softw. Syst. Model.* **2004**, *3*, 314–327. [CrossRef]
19. Schmidt, D.C. Model-driven engineering. *Comput.-IEEE Comput. Soc.* **2006**, *39*, 25. [CrossRef]
20. Dias Neto, A.; Subramanyan, R.; Vieira, M.; Travassos, G. A survey on model-based testing approaches: A systematic review. *Empir. Softw. Eng.* **2020**, *25*, 4205–4249.
21. Li, S.; Xu, L.D.; Zhao, S. The internet of things: A survey. *Inf. Syst. Front.* **2015**, *17*, 243–259. [CrossRef]
22. Teixeira, T.; Hachem, S.; Issarny, V.; Georgantas, N. Service oriented middleware for the internet of things: A perspective. In Proceedings of the European Conference on a Service-Based Internet, Poznan, Poland, 26–28 October 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 220–229.
23. Mehta, R.; Sahni, J.; Khanna, K. Internet of things: Vision, applications and challenges. *Procedia Comput. Sci.* **2018**, *132*, 1263–1269. [CrossRef]
24. Sharma, S.K.; Wang, X. Live data analytics with collaborative edge and cloud processing in wireless IoT networks. *IEEE Access* **2017**, *5*, 4621–4635. [CrossRef]
25. Ayodele, T.O. Types of machine learning algorithms. *New Adv. Mach. Learn.* **2010**, *3*, 19–48.
26. Liu, Z.; Zhu, Z.; Gao, J.; Xu, C. Forecast methods for time series data: A survey. *IEEE Access* **2021**, *9*, 91896–91912. [CrossRef]
27. Wolny, S.; Mazak, A.; Wally, B. An Initial Mapping Study on MDE4IoT. In Proceedings of the 2nd International Workshop on Model-Driven Engineering for the Internet-of-Things (MDE4IoT), 2018; pp. 524–529.
28. Ihirwe, F.; Di Ruscio, D.; Mazzini, S.; Pierini, P.; Pierantonio, A. Low-code Engineering for Internet of things: A state of research. In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Virtual Event, 18–23 October 2020; pp. 1–8.
29. Muzaffar, A.W.; Mir, S.R.; Anwar, M.W.; Ashraf, A. Application of model driven engineering in cloud computing: A systematic literature review. In Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing, Cambridge, UK, 22–23 March 2017; pp. 1–6.
30. Abshir Mohamed, M.; Kardas, G.; Challenger, M. Model-Driven Engineering Tools and Languages for Cyber-Physical Systems–A Systematic Literature Review. *IEEE Access* **2021**, *9*, 48605–48630. [CrossRef]
31. Casalaro, G.L.; Cattivera, G.; Ciccozzi, F.; Malavolta, I.; Wortmann, A.; Pelliccione, P. Model-driven engineering for mobile robotic systems: A systematic mapping study. *Softw. Syst. Model.* **2021**, *21*, 1–31. [CrossRef]
32. Mashkoor, A.; Egyed, A.; Wille, R. Model-driven engineering of safety and security systems: A systematic mapping study. *arXiv* **2020**, arXiv:2004.08471.
33. de Araújo Silva, E.; Valentin, E.; Carvalho, J.R.H.; da Silva Barreto, R. A survey of Model Driven Engineering in robotics. *J. Comput. Lang.* **2021**, *62*, 101021. [CrossRef]
34. Ralph, P.; Baltes, S. Paving the Way for Mature Secondary Research: The Seven Types of Literature Review. *arXiv* **2022**, arXiv:2205.01163.
35. Peters, M.D.; Marnie, C.; Tricco, A.C.; Pollock, D.; Munn, Z.; Alexander, L.; McInerney, P.; Godfrey, C.M.; Khalil, H. Updated methodological guidance for the conduct of scoping reviews. *JBI Evid. Synth.* **2020**, *18*, 2119–2126. [CrossRef] [PubMed]
36. Colquhoun, H.L.; Levac, D.; O'Brien, K.K.; Straus, S.; Tricco, A.C.; Perrier, L.; Kastner, M.; Moher, D. Scoping reviews: Time for clarity in definition, methods, and reporting. *J. Clin. Epidemiol.* **2014**, *67*, 1291–1294. [CrossRef]
37. Arksey, H.; O'Malley, L. Scoping studies: Towards a methodological framework. *Int. J. Soc. Res. Methodol.* **2005**, *8*, 19–32. [CrossRef]
38. Shamseer, L.; Moher, D.; Clarke, M.; Ghersi, D.; Liberati, A.; Petticrew, M.; Shekelle, P.; Stewart, L.A. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015: Elaboration and explanation. *BMJ* **2015**, *349*, g7647. [CrossRef]

39. Ribeiro, A.; de Sousa, L.; da Silva, A.R. Comparative analysis of workbenches to support DSMLs: Discussion with non-trivial Model-Driven Development needs. In Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), Rome, Italy, 19–21 February 2016; pp. 323–330.
40. Granada, D.; Vara, J.M.; Blanco, F.P.; Marcos, E. Model-based Tool Support for the Development of Visual Editors—A Systematic Mapping Study. In Proceedings of the 12th International Conference on Software Technologies, Madrid, Spain, 24-26 July 2017; pp. 330–337. [CrossRef]
41. Pekar, A.; Mocnej, J.; Seah, W.K.; Zolotova, I. Application domain-based overview of IoT network traffic characteristics. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–33. [CrossRef]
42. Tian, S.; Yang, W.; Le Grange, J.M.; Wang, P.; Huang, W.; Ye, Z. Smart healthcare: Making medical care more intelligent. *Glob. Health J.* **2019**, *3*, 62–65. [CrossRef]
43. Anthopoulos, L.G. Understanding the smart city domain: A literature review. *Transform. City Gov. Success. Smart Cities* **2015**, *8*, 9–21.
44. Debnath, A.K.; Chin, H.C.; Haque, M.M.; Yuen, B. A methodological framework for benchmarking smart transport cities. *Cities* **2014**, *37*, 47–56. [CrossRef]
45. Moin, A. Data Analytics and Machine Learning Methods, Techniques and Tool for Model-Driven Engineering of Smart IoT Services. In Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Madrid, Spain, 25–28 May 2021; pp. 287–292.
46. Moin, A.; Challenger, M.; Badii, A.; Günnemann, S. A model-driven approach to machine learning and software modeling for the IoT. *Softw. Syst. Model.* **2022**, *21*, 987–1014. [CrossRef]
47. Terracher, A.; Kocik, R.; Hamouche, R.; Rachedi, A.; Bourouina, T. IoT efficient data exploitation process using Model Driven Engineering. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Korea, 16–20 May 2022; pp. 5268–5273.
48. Erazo-Garzón, L.; Cedillo, P.; Rossi, G.H.; Moyano, J. A Domain-Specific Language for Modeling IoT System Architectures That Support Monitoring. *IEEE Access* **2022**, *10*, 61639–61665. [CrossRef]
49. Moin, A.; Challenger, M.; Badii, A.; Günnemann, S. Supporting AI Engineering on the IoT Edge through Model-Driven TinyML. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA, 27 June–1 July 2022; pp. 884–893.
50. Barriga, J.A.; Clemente, P.J.; Hernández, J.; Pérez-Toledano, M.A. SimulateIoT-FIWARE: Domain Specific Language to Design, Code Generation and Execute IoT Simulation Environments on FIWARE. *IEEE Access* **2022**, *10*, 7800–7822. [CrossRef]
51. Berrouyne, I.; Adda, M.; Mottu, J.M.; Tisi, M. A Model-Driven Methodology to Accelerate Software Engineering in the Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 19757–19772. [CrossRef]
52. Barriga, J.A.; Clemente, P.J. Designing and simulating IoT environments by using a model-driven approach. In Proceedings of the 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, 22–25 June 2022; pp. 1–6.
53. Kirchhof, J.C.; Rumpe, B.; Schmalzing, D.; Wortmann, A. Montithings: Model-driven development and deployment of reliable iot applications. *J. Syst. Softw.* **2022**, *183*, 111087. [CrossRef]
54. Mehrabi, M.; Zamani, B.; Hamou-Lhadj, A. HealMA: A model-driven framework for automatic generation of IoT-based Android health monitoring applications. *Autom. Softw. Eng.* **2022**, *29*, 1–41. [CrossRef] [PubMed]
55. Ihirwe, F.; Di Ruscio, D.; Mazzini, S.; Pierantonio, A. Towards a modeling and analysis environment for industrial IoT systems. *arXiv* **2021**, arXiv:2105.14136.
56. Karaduman, B.; Mustafiz, S.; Challenger, M. FTG+ PM for the Model-Driven Development of Wireless Sensor Network based IoT Systems. In Proceedings of the 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, 10–15 October 2021; pp. 306–316.
57. Shirazi, S.S.B.; Rashid, M.; Azam, F.; Rasheed, Y.; Anwar, M.W. A Model-Driven Framework for Early Analysis of Kill Chain Attacks. In Proceedings of the 2021 National Computing Colleges Conference (NCCC), Taif, Saudi Arabia, 27–28 March 2021; pp. 1–5.
58. Barriga, J.A.; Clemente, P.J.; Sosa-Sánchez, E.; Prieto, Á.E. SimulateIoT: Domain Specific Language to design, code generation and execute IoT simulation environments. *IEEE Access* **2021**, *9*, 92531–92552. [CrossRef]
59. Alulema, D.; Criado, J.; Iribarne, L. An Approach to Integrate IoT Systems with No-Web Interfaces. In *International Conference on Information Technology & Systems*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 417–427.
60. Meliá, S.; Nasabeh, S.; Luján-Mora, S.; Cachero, C. MoSIoT: Modeling and Simulating IoT Healthcare-Monitoring Systems for People with Disabilities. *Int. J. Environ. Res. Public Health* **2021**, *18*, 6357. [CrossRef]
61. Parri, J.; Sampietro, S.; Vicario, E. FaultFlow: A tool supporting an MDE approach for Timed Failure Logic Analysis. In Proceedings of the 2021 17th European Dependable Computing Conference (EDCC), Munich, Germany, 13–16 September 2021; pp. 25–32.
62. Arslan, S.; Kardaş, G. The Need for Model-driven Engineering in the Development of IoT Software for Public Transportation Systems. In Proceedings of the 2021 15th Turkish National Software Engineering Symposium (UYMS), Izmir, Turkey, 17–19 November 2021; pp. 1–6.

63. Karaduman, B.; Challenger, M. Model-driven development for ESP-based IoT systems. In Proceedings of the 2021 IEEE/ACM 3rd International Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT), Madrid, Spain, 3 June 2021; pp. 9–12.
64. Karaduman, B.; Challenger, M.; Eslampanah, R.; Denil, J.; Vangheluwe, H. Analyzing WSN-based IoT Systems using MDE Techniques and Petri-net Models. In *STAF Workshops*; 2020; pp. 35–46. Available online: http://www.es.mdh.se/MDE4IoT/mde4iot_p1.pdf (accessed on 13 December 2022).
65. Berrouyne, I.; Adda, M.; Mottu, J.M.; Royer, J.C.; Tisi, M. A Model-Driven Approach to Unravel the Interoperability Problem of the Internet of Things. In *International Conference on Advanced Information Networking and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1162–1175.
66. Soukaras, D.; Patel, P.; Song, H.; Chaudhary, S. IoTSuite: A ToolSuite for prototyping internet of things applications. In *The 4th International Workshop on Computing and Networking for Internet of Things (ComNet-IoT), Co-Located with 16th International Conference on Distributed Computing and Networking (ICDCN)*; 2015. Available online: https://www.researchgate.net/publication/273003282_IoTSuite_A_ToolSuite_for_Prototyping_Internet_of_Things_Applications (accessed on 13 December 2022).
67. Ziaei, M.; Zamani, B.; Bohlooli, A. A Model-Driven Approach for IoT-Based Monitoring Systems in Industry 4.0. In Proceedings of the 2020 4th International Conference on Smart City, Internet of Things and Applications (SCIOT), Mashhad, Iran, 16–17 September 2020; pp. 99–105.
68. Plazas, J.E.; Bimonte, S.; de Vaulx, C.; Schneider, M.; Nguyen, Q.D.; Chanet, J.P.; Shi, H.; Hou, K.M.; Corrales, J.C. A conceptual data model and its automatic implementation for IoT-based business intelligence applications. *IEEE Internet Things J.* **2020**, *7*, 10719–10732. [CrossRef]
69. Anwer, H.; Azam, F.; Anwar, M.W.; Rashid, M. A model-driven approach for load-balanced MQTT protocol in internet of things (IoT). In *Complex, Intelligent, and Software Intensive Systems: Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019)*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 368–378.
70. Bisma, M.; Azam, F.; Rasheed, Y.; Anwar, M.W. A Model-Driven Framework for Ensuring Role Based Access Control in IoT Devices. In Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence, Honolulu, HI, USA, 18–20 September 2020; pp. 455–460.
71. Conzon, D.; Rashid, M.R.A.; Tao, X.; Soriano, A.; Nicholson, R.; Ferrera, E. BRAIN-IoT: Model-based framework for dependable sensing and actuation in intelligent decentralized IoT systems. In Proceedings of the 2019 4th International Conference on Computing, Communications and Security (ICCCS), Rome, Italy, 10–12 October 2019; pp. 1–8.
72. Costa, B.; Pires, P.F.; Delicato, F.C. Modeling SOA-based IoT applications with SoaML4IoT. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 496–501.
73. Michael, J.; Netz, L.; Rumpe, B.; Varga, S. Towards Privacy-Preserving IoT Systems Using Model Driven Engineering. In *EMISA Forum*; De Gruyter: Berlin, Germany, 2020.
74. Berrouyne, I.; Adda, M.; Mottu, J.M.; Royer, J.C.; Tisi, M. CyprIoT: Framework for modelling and controlling network-based IoT applications. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 832–841.
75. Hartmann, T.; Moawad, A.; Fouquet, F.; Le Traon, Y. The next evolution of MDE: A seamless integration of machine learning into domain modeling. *Softw. Syst. Model. (SoSyM)* **2019**, *18*, 1285–1304. [CrossRef]
76. Muntés-Mulero, V.; Dominiak, J.; González, E.; Sanchez-Charles, D. Model-driven Evidence-based Privacy Risk Control in Trustworthy Smart IoT Systems. In CEUR Workshop Proceedings; 2019; pp. 23–30. Available online: https://ceur-ws.org/Vol-2442/paper4.pdf (accessed on 13 December 2022).
77. Asici, T.Z.; Karaduman, B.; Eslampanah, R.; Challenger, M.; Denil, J.; Vangheluwe, H. Applying model driven engineering techniques to the development of contiki-based IoT systems. In Proceedings of the 2019 IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT), Montreal, QC, Canada, 27 May 2019; pp. 25–32.
78. Alulema, D.; Criado, J.; Iribarne, L. A model-driven approach for the integration of hardware nodes in the IoT. In *World Conference on Information Systems and Technologies*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 801–811.
79. Jahed, K.; Dingel, J. Enabling model-driven software development tools for the internet of things. In Proceedings of the 2019 IEEE/ACM 11th International Workshop on Modelling in Software Engineering (MiSE), Montreal, QC, Canada, 26–27 May 2019; pp. 93–99.
80. Moreira, J.L. SEMIoTICS: Semantic Model-driven Development for IoT Interoperability of Emergency Services: Improving the Semantic Interoperability of IoT Early Warning Systems. Ph.D. Thesis, University of Twente, Enschede, The Netherlands, 2019.
81. De Morais, C.M.; Kelner, J.; Sadok, D.; Lynn, T. SiMoNa: A proof-of-concept domain specific modeling language for IoT infographics. In Proceedings of the 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Lisbon, Portugal, 1–4 October 2018; pp. 199–203.
82. Thramboulidis, K.; Kontou, I.; Vachtsevanou, D.C. Towards an IoT-based framework for evolvable assembly systems. *IFAC-PapersOnLine* **2018**, *51*, 182–187. [CrossRef]
83. Sosa-Reyna, C.M.; Tello-Leal, E.; Lara-Alabazares, D.; Mata-Torres, J.A.; Lopez-Garza, E. A Methodology Based on Model-Driven Engineering for IoT Application Development. In Proceedings of the 12th International Conference on Digital Society and eGovernments (ICDS 2018), Rome, Italy, 25–29 March 2018.

84. Iyenghar, P.; Pulvermueller, E. A model-driven workflow for energy-aware scheduling analysis of IoT-enabled use cases. *IEEE Internet Things J.* **2018**, *5*, 4914–4925. [CrossRef]
85. Kotronis, C.; Nikolaidou, M.; Dimitrakopoulos, G.; Anagnostopoulos, D.; Amira, A.; Bensaali, F. A model-based approach for managing criticality requirements in e-health iot systems. In Proceedings of the 2018 13th annual conference on system of systems engineering (SoSE), Paris, France, 19–22 June 2018; pp. 60–67.
86. Khan, M.; Anwar, M.W.; Azam, F.; Samea, F.; Shinwari, M.F. A Model-Driven Approach for Access Control in Internet of Things (IoT) Applications—An Introduction to UMLOA. In Proceedings of the International Conference on Information and Software Technologies, Vilnius, Lithuania, 4–6 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 198–209.
87. Dias, J.P.; Faria, J.P.; Ferreira, H.S. A reactive and model-based approach for developing internet-of-things systems. In Proceedings of the 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC), Coimbra, Portugal, 4–7 September 2018; pp. 276–281.
88. Gomes, T.; Lopes, P.; Alves, J.; Mestre, P.; Cabral, J.; Monteiro, J.L.; Tavares, A. A modeling domain-specific language for IoT-enabled operating systems. In Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October 2017; pp. 3945–3950.
89. Einarsson, A.F.; Patreksson, P.; Hamdaqa, M.; Hamou-Lhadj, A. SmarthomeML: Towards a domain-specific modeling language for creating smart home applications. In Proceedings of the 2017 IEEE International Congress on Internet of Things (ICIOT), Honolulu, HI, USA, 25–30 June 2017; pp. 82–88.
90. Rosique, F.; Losilla, F.; Pastor, J.Á. A domain specific language for smart cities. *Multidiscip. Digit. Publ. Inst. Proc.* **2017**, *2*, 148.
91. Thramboulidis, K.; Bochalis, P.; Bouloumpasis, J. A framework for MDE of IoT-based manufacturing cyber-physical systems. In Proceedings of the Seventh International Conference on the Internet of Things, Linz, Austria, 22–25 October 2017; pp. 1–8.
92. Durmaz, C.; Challenger, M.; Dagdeviren, O.; Kardas, G. Modelling contiki-based IoT systems. In Proceedings of the 6th symposium on languages, applications and technologies (SLATE 2017), Vila do Conde, Portugal, 26–27 June 2017.
93. Morin, B.; Fleurey, F.; Husa, K.E.; Barais, O. A Generative Middleware for Heterogeneous and Distributed Services. In Proceedings of the 2016 19th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE), Venice, Italy, 5–8 April 2016; pp. 107–116. [CrossRef]
94. Alves, M.P.; Delicato, F.C.; Pires, P.F. Iota-md: A model-driven approach for applying qos attributes in the development of the iot systems. In Proceedings of the Symposium on Applied Computing, Marrakech, Morocco, 3–7 April 2017; pp. 1773–1780.
95. Mezghani, E.; Exposito, E.; Drira, K. A model-driven methodology for the design of autonomic and cognitive IoT-based systems: Application to healthcare. *IEEE Trans. Emerg. Top. Comput. Intell.* **2017**, *1*, 224–234. [CrossRef]
96. Hassine, T.B.; Khayati, O.; Ghezala, H.B. An IoT domain meta-model and an approach to software development of IoT solutions. In Proceedings of the 2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), Gafsa, Tunisia, 20–22 October 2017; pp. 32–37.
97. Venčkauskas, A.; Štuikys, V.; Toldinas, J.; Jusas, N. A model-driven framework to develop personalized health monitoring. *Symmetry* **2016**, *8*, 65. [CrossRef]
98. Koshima, A.A.; Englebert, V.; Amani, M.; Debieche, A.; Wakjira, A. A Model-Driven Engineering Approach for the Well-Being of Ageing People. In Proceedings of the International Conference on Conceptual Modeling, Gifu, Japan, 14–17 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–29.
99. Costa, B.; Pires, P.F.; Delicato, F.C.; Li, W.; Zomaya, A.Y. Design and analysis of IoT applications: A model-driven approach. In Proceedings of the 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Auckland, New Zealand, 8–12 August 2016; pp. 392–399.
100. Ciccozzi, F.; Spalazzese, R. Mde4iot: Supporting the internet of things with model-driven engineering. In *Intelligent Distributed Computing X: Proceedings of the 10th International Symposium on Intelligent Distributed Computing–IDC 2016, Paris, France, 10–12 October 2016*; Springer: Berlin/Heidelberg, Germany, 2016, pp. 67–76.
101. Karkouch, A.; Mousannif, H.; Al Moatassime, H.; Noel, T. A model-driven architecture-based data quality management framework for the internet of Things. In Proceedings of the 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), Marrakesh, Morocco, 24–26 May 2016; pp. 252–259.
102. Grace, P.; Pickering, B.; Surridge, M. Model-driven interoperability: Engineering heterogeneous IoT systems. *Ann. Telecommun.* **2016**, *71*, 141–150. [CrossRef]
103. Mazzini, S.; Favaro, J.; Baracchi, L. A model-based approach across the IoT lifecycle for scalable and distributed smart applications. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 149–154.
104. Nguyen, X.T.; Tran, H.T.; Baraki, H.; Geihs, K. FRASAD: A framework for model-driven IoT Application Development. In Proceedings of the 2015 IEEE 2nd world forum on internet of things (WF-IoT), Milan, Italy, 14–16 December 2015; pp. 387–392.
105. Salihbegovic, A.; Eterovic, T.; Kaljic, E.; Ribic, S. Design of a domain specific language and IDE for Internet of things applications. In Proceedings of the 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 996–1001.

106. Pramudianto, F.; Kamienski, C.A.; Souto, E.; Borelli, F.; Gomes, L.L.; Sadok, D.; Jarke, M. Iot link: An internet of things prototyping toolkit. In Proceedings of the 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, Bali, Indonesia, 9–12 December 2014; pp. 1–9.

107. Thang, N.X.; Zapf, M.; Geihs, K. Model driven development for data-centric sensor network applications. In Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia, Ho Chi Minh City, Vietnam, 5–7 December 2011; pp. 194–197.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.