An Overview of the Declarative Programming Languages for the IoT Domain

F. Morero-Peyrona*, **J.G. Enríquez***, **and A. Jiménez-Ramírez***
*Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla

ABSTRACT This paper presents an extensive review of the existing body of work in the realm of programming languages designed to mimic natural language constructs; with a specific focus on their applicability within the Internet of Things (IoT) domain. In essence, it offers a comprehensive examination and comparative analysis of previous research conducted in this domain. To achieve this objective, we employ the widely recognized Systematic Literature Review (SLR) methodology, a proven approach in the field of Software Engineering. We articulate the research questions that guide our inquiry, initiate rigorous searches through prominent academic document repositories and meticulously curate the results through a systematic filtering process. Subsequently, we scrutinize each of the resulting research papers, targeting specific salient features. In the final phase, we present the results of our analysis and engage in a thoughtful discussion on the risks inherent to this research, accompanied by a comprehensive overview of the results obtained. We conclude by highlighting potential avenues for future research within this domain.

KEYWORDS IoT, Internet of the Things, declarative programming language, natural programming language, Systematic Literature Review, SLR

1. Introduction

1.1. The complexity of IoT

Day by day, the number of devices, communication protocols, gateways, and other components in the field of Internet of Things (IoT) is growing. The scope of this issue is so broad that there are studies devoted solely to suggesting approaches to address it: there have also been numerous attempts to apply established and verified strategies and theoretical models to the realm of the Internet of Things, such as the utilization of Model-Driven Engineering and the utilization of meta-models: "...the [IoT] domain is characterized by many different devices that typically need to communicate with each other across different protocols and means urging system designers to realize various interface adapters to let parts communicate" (Lombardi et al. 2021). An example of these efforts is GeneSIS, a model-driven approach for the Generation and Deployment of Smart IoT Systems; which "aims to support the continuous deployment of smart IoT systems over IoT, Edge and Cloud infrastructures" (Ferry et al. 2020).

On the other hand, devices increase not only in number, but also in complexity as in any other area of the IT industry: let's think for a second about a mobile phone in 2022 and a mobile phone only 20 years earlier: the gap in terms of technical complexity between them is enormous.

It is difficult to determine the exact number of connected devices in use around the world today, as this figure is constantly growing. However, it was estimated in August 2022 that by the end of that year there would be 13.1 billion ($13.1x10^9$) devices connected. This number is expected to rise to 75.4 billion ($75.4x10^9$) by 2025[1].

To make matters worse, almost every IoT manufacturer has its own proposal to solve the problem of providing common users (not programmers) a software tool to facilitate the use of their physical products (hardware): these solutions can be as simple as a smartphone app for managing an RGB light[2]; or as

JOT reference format:
F. Morero-Peyrona, J.G. Enríquez, and A. Jiménez-Ramírez. *An Overview of the Declarative Programming Languages for the IoT Domain.* Journal of Object Technology. Vol. vv, No. nn, yyyy.
http://dx.doi.org/10.5381/jot.yyyy.vv.nn.aa

[1] https://techjury.net/blog/how-many-iot-devices-are-there/
[2] https://play.google.com/store/apps/details?id=cloud.blynk

complex as a SCADA[3] to manage say, a bottling machine in a beer factory production line.

The proposals of IT manufacturers cover a wide range in terms of complexity: some of them propose non-code-needed solutions, while others offer programming languages or a mixture of both. As in any other aspect of life, each approach has benefits and drawbacks. For achieving simple tasks, non-code-needed solutions are widely used, being very rare when there is a need to achieve complex goals. In any case, programming languages have their defenders and their audience, as they tend to be more suitable for solving complex problems. When talking about programming languages, it appears as an undeniable idea that a natural language is the best option for the non-programmers audience (Cambranes 2013), while the learning curve for programmers will be close to flat.

Traditionally, programming languages close to natural language had been also command-based languages, this is the case of Structured Query Language (SQL), which "is a standardized language for defining and manipulating data in a relational database." [4] or AppleScript, "a scripting language [that] allows users to directly control scriptable Macintosh applications, [. . .] can create scripts—sets of written instructions—to automate repetitive tasks, combine features from multiple scriptable applications, and create complex workflows."[5]

1.2. Problem Definition

As researchers at the University of Seville, we have often encountered these issues. We faced the requirement of having a lingua franca, or at least a framework that would enable us to collaborate with IoT devices from different producers, as well as to reduce the learning curve as much as possible; if achievable, using a language that is as close as possible to natural language.

We believe that it would be beneficial to provide developers and nontechnical individuals with a straightforward approach to tackle these situations, ideally something that resembles natural language, because "using a natural language-based programming environment [. . .] could be more effective and appealing"-(Keene & Jamil 2022).

1.3. Research goal

As we have said, the purpose of our research is to conduct a Systematic Literature Reviews (SLR) of the available academic publications in the field of declarative (as close-to-natural as possible) scripting programming languages in the area of the Internet of Things (IoT).

We evaluate programming languages ideal for IoT, with a focus on their technical specifications and features; ranging from declarative (close-to-natural) ones while eschewing their imperative counterparts. This strategic choice to concentrate on declarative programming languages in the IoT domain is firmly grounded in their affinity to natural language, the disparity of paradigms, and the necessity for paradigmatic cohesion in the interest of achieving unambiguous and comprehensible software development within the IoT ecosystem.

On the other hand, the discussion will only explore an in-depth evaluation of the programming languages themselves, while avoiding broader aspects such as workflow or deployment: while this is undoubtedly a valuable avenue for investigation, it is imperative to clarify our specific research focus. This paper is intentionally encompassed to a narrower scope, focusing on the inherent attributes of programming languages themselves. Our research prioritizes the intrinsic characteristics of the programming language. This emphasis is particularly pertinent because a single programming language may undergo interpreted or compiled, developers may employ different editors or integrated development environments (IDEs) to work with the same language. Same with debuggers and other optimizations tools, like profilers.

Our aim is to objectively highlight the strengths and weaknesses of each selected programming language and provide an overview of their core specifications. Also, to shed light on how well these declarative languages facilitate the fundamental aspects of IoT application development.

We hope that this review is a valuable resource for developers and engineers seeking the most suitable declarative programming language for their IoT projects, with a focus on the technical aspects of IoT application development. Our review maintains objectivity by providing clear and concise information.

1.4. Paper organization

The rest of the paper is organized as follows: in the next section, we will expound the methodology used (the standard SLR in our case), following we summarize the goal of this research and will present the research questions: those that will unambiguously lead to our goal. In the next section we conduct the search itself, where several iterations of systematic searches and selection criteria are applied. In the next step, to present the comparison among the findings, we opted to summarize the research using a comparative table (not in vain, tables provide quite a lot of information at a glance). This paper ends by revealing the conclusions and finally pointing to a possible way for future work.

2. Methodology

In the academic literature, different methodologies are proposed to allow a systematic analysis of the state-of-the-art of a given topic in a systematic way. Systematic Literature Reviews (SLRs) (Kitchenham & Brereton 2013) and Systematic Mapping Studies (SMSs) (Petersen et al. 2015) are two prominent methodologies employed in academic research to explore and synthesize the existing literature in a systematic and rigorous manner. These approaches play a crucial role in advancing knowledge, informing evidence-based decision making, and identifying research gaps within specific fields of study.

As described by Petersen et al. (2015), both methodologies intend to perform a methodical analysis of the preexisting literature. However, they differ in several respects. First, SMSs

[3] https://new.siemens.com/global/en/products/automation/industry-software/automation-software/scada.html

[4] https://www.ibm.com/docs/en/i/7.2?topic=concepts-structured-query-language

[5] https://developer.apple.com/library/archive/documentation/AppleScript/Conceptual/AppleScriptLangGuide/introduction/ASLR_intro.html

typically have broad research questions that aim to uncover research trends, while SLRs require a specific research goal for evidence aggregation. Second, the research process in SLRs is guided by a specific topic area, whereas SMSs are driven by specific research questions. Third, SMSs have less strict search requirements compared to SLRs, which require the inclusion of all relevant studies. Fourth, quality assessment is imperative in systematic literature reviews (SLRs) to measure the dependability of included studies, whereas systematic mapping studies (SMSs) do not necessitate such evaluations. Lastly, SMSs produce an inventory of papers on a topic area to provide an overview of the field to spot research gaps and trends. In contrast, SLRs focus mainly on synthesizing evidence on a narrowly defined research question.

Due to the nature of this research and considering that the number of published studies related to the subject matter of this article is limited, we will apply the methodology proposed by Petersen et al. (Petersen et al. 2008, 2015) to perform an SMS. This methodology is composed of 5 steps (cf. Figure 1): definition of the research question, conducting the search, screening of papers, keywording using abstracts, and data extraction and mapping process.

- **Definition of the research question:** The first step in an SMS involves clearly defining the research question or the objective of the study. This involves identifying the scope and boundaries of the research topic and determining the specific aspects to be explored or mapped within the literature.
- **Conduct search:** Once the research questions are defined, a systematic and comprehensive search is conducted across relevant databases, journals, conference proceedings, and other sources. The search strategy is developed, including selecting appropriate keywords, controlled vocabulary terms, and search operators. The aim is to identify as many relevant publications as possible that address the research question.
- **Screening of papers:** After the initial search, the retrieved papers are screened to identify potentially relevant studies. Initially, titles and abstracts are screened to determine their potential relevance to the research question. Papers that pass this initial screening are then subjected to a full-text assessment to determine their eligibility for inclusion in the SMS.
- **Keywording using abstracts:** During the screening process, a common practice in SMSs is to extract keywords or phrases from the abstracts of the selected papers. These keywords capture essential concepts, themes, or variables the literature addresses. The extracted keywords can be used to develop a classification framework or taxonomy for organizing and categorizing the included studies.
- **Data extraction and mapping process:** Once the final set of papers is identified, a systematic data extraction process is carried out. This involves systematically extracting relevant data from each included study, such as bibliographic information, study characteristics, methodologies used, key findings, and other pertinent details. The extracted data are organized and synthesized in a structured manner.

3. Execution

3.1. Definition of the research questions

In order to identify the most advanced programming languages for the Internet of Things (IoT) that are accessible to non-programmers, yet still provide the capacity to handle complex projects for experts, we have formulated a set of questions that are straightforward and unambiguous. This paper will answer these questions by conducting a Systematic Literature Review (SLR) of relevant scientific papers up to the date of the research:

- Are there close-to-natural or at least declarative programming languages for the IoT domain?
- Can these languages be extended in any way?
- Can the applications created cooperate in any way (e.g., forming any kind of grid)?

3.2. Conduct search

Our first step was to search where to search, or better to say: which are the most relevant search engines to find academic publications in addition to generic search engines (Google, MS-Bing, etc.).

We conducted a thorough examination of the most popular academic search engines reported. After reading reviews from the scientific community and assessing their value, we tested the ones that were the most frequently mentioned and highly rated. Unsurprisingly, the search engines we chose are the ones that are most commonly used in academic literature:

- Scopus
- IEEE Explore
- Science Direct
- WebOfScience

The aim of this essay is to identify publications related to computer languages that are close to natural language and specific to the Internet of Things (IoT) domain. To do this, we conducted multiple cycles of sending a search criterion and analyzing the results to determine which criteria yielded the most accurate results. To expand the search and maximize the number of hits, we then implemented the criteria listed in the table 1.

Search engines enable users to conduct "intelligent" searches. We initially searched for the term "iot" in the title, metadata, and keywords. We received more hits, but after further examination, we realized that the results were not as pertinent as we had anticipated. Therefore, as mentioned previously, we tried different combinations in a trial-and-error cycle and discovered that the searches listed in Table 1 were the most suitable for our case, returning more precise results.

Having a smaller number of results (while still being accurate) has another major benefit: we can examine each one thoroughly instead of having to come up with a new criterion to limit the amount of data that a person can process.

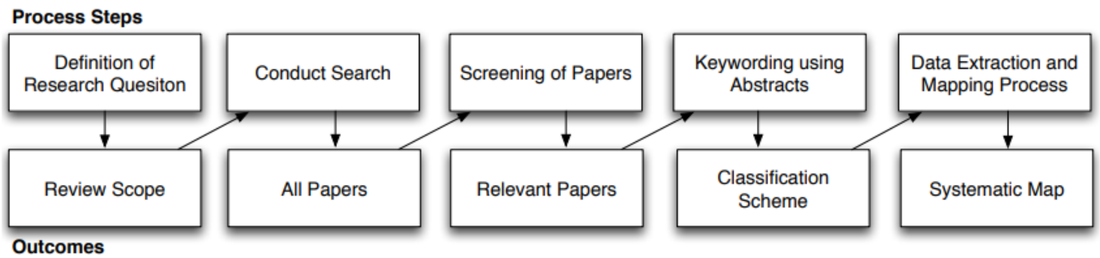

Figure 1 Systematic Mapping Study methodology by Petersen et. al (Petersen et al. 2008)

Engine	Search	Results (2022/06)
Scopus	(TITLE (iot) AND TITLE-ABS-KEY (natural OR declarative) AND TITLE-ABS-KEY (language)) AND (LIMIT-TO (SUBJAREA , "COMP"))	132
IEEE	("Document Title":iot) AND ("All Metadata":natural OR declarative) AND ("All Metadata":language)	79
Science Direct	Title, abstract, keywords: "IOT" AND ("natural" OR "declarative") AND "language"	25
Web of Science	IOT (Title) and NATURAL OR DECLARATIVE (All Fields) and language (All Fields)	110

Table 1 Executed queries

3.3. Screening of papers

Following is the procedure (Exclusion Criteria or EC) used to select the papers that will be selected for the next steps and ultimately analyzed:

1. Paper has to be written in English
2. Read the title and the abstract: if it is promising, go to next step, otherwise, discard it.
3. If the paper is not indexed in JCR[6] nor in SCIE[7], then discard it.
4. Read the entire paper in case of doubt to either select or discard the paper.

Figure 2 diagram shows the entire selection process and the amount of papers remaining after applying every step.

3.4. Keywording using abstracts

Starting with the previously mentioned research questions, we arrived at the following key points (we will look for every question in each one of the selected papers):

– Is the language close to natural language or at least declarative?
– Is a script[8] type language (vs. procedural)?
– Provides the language of any kind of decision-making entity (e.g., rules)?
– Allows interacting in any way with other languages (Java, JS, Python, etc.)?
– Can the language itself and/or its functionality be extended in any way?
– Provides the language a mechanism to incorporate physical devices?
– Provides protocols to communicate/monitor/interact with running instances?
– Allow collaboration of running instances (grids) and spreads logic and functionality?

3.5. Paper selection process

The 15 papers that passed Exclusion Criteria #1 (Paper has to be written in English) and Exclusion Criteria #2 (Promising title and abstract) correspond with the following references, which are presented at the end of this paper under the References section:

1. (Poroor 2018)
2. (Hamdan et al. 2019)
3. (Kumar et al. 2021)
4. (Soic et al. 2020)
5. (Amrani et al. 2017)
6. (D'Urso et al. 2019)
7. (Gyrard et al. 2014)

[6] https://jcr.clarivate.com/
[7] https://scie.lcc.uma.es:8443/gii-grin-scie-rating/ratingSearch.jsf
[8] The action of scripting is essentially writing a series of commands that are interpreted one by one by an application or scripting engine (https://en.wikipedia.org/wiki/Scripting_language)

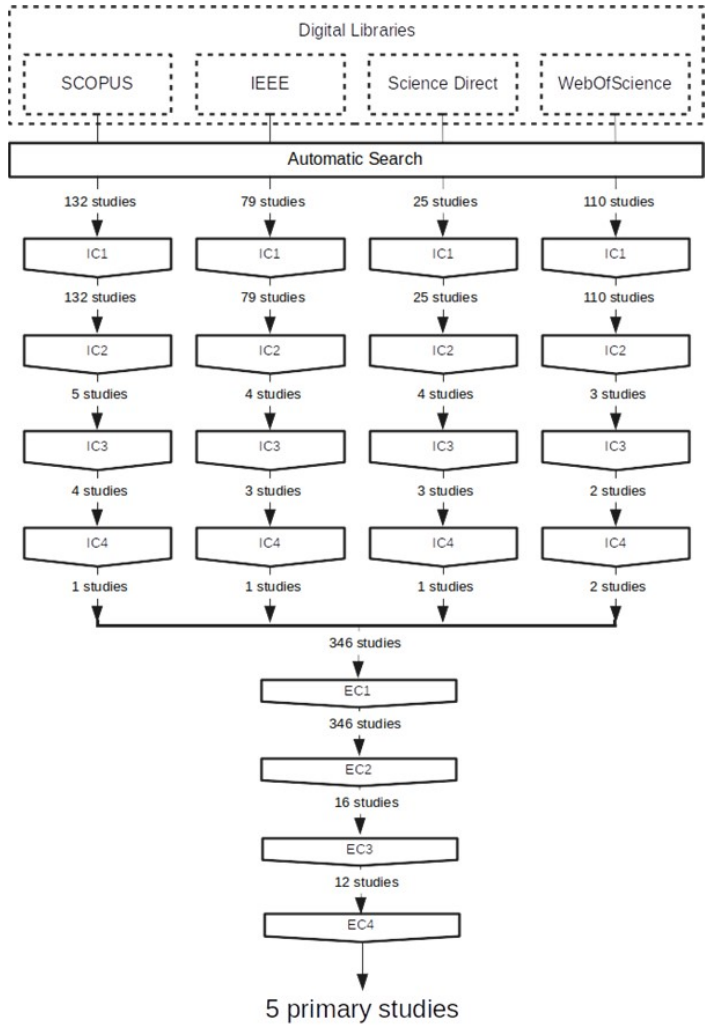

Figure 2 Execution results

8. (Yu et al. 2021)
9. (Longo et al. 2021)
10. (Petnik & Vanus 2018)
11. (Zúñiga et al. 2020)
12. (Lago et al. 2021)
13. (Li et al. 2018)
14. (Dong et al. 2020)
15. (Gabbrielli et al. 2018)

Table 2 shows the papers regarding Exclusion Criteria number 3 (Indexed in JCR6 or SCIE).

The following are the papers that did not pass Exclusion Criteria number 4:

(1) Cannot pass because it is merely a declaration of intentions. They explain in just 2 pages what they think would be a good idea for a language IoT. This rough draft presents a language that presumably would work properly embedded in low-power consumption devices.

(2) Cannot pass because it just describes in less than 2 pages the hardware architecture that authors think would be suitable for IoT environments. No mention of a programming language is even made.

(7) Cannot be passed because the proposed solution is highly technical: only IT professionals with adequate knowledge in the fields of IoT, ontology-web, and semantic web would be able to use it.

(10) Cannot be passed because the paper does not propose any language at all. On top of this, its scope is different: the authors propose a general-purpose NLP that, even if it could be suitable for IoT, this is not its main target but defense. On the other hand, it can only be used by senior engineers.

(11) Cannot be passed because it is just a software architecture (a set of software artifacts) proposal with no guarantee of proper functioning. No language is involved in this proposal.

3.6. Achieved Results

After applying these exclusion criteria, the resulting papers are shown in this concise table 3 as the first row cell. As mentioned above, we identified several key points that were pursued in

Paper #	Source	Type of paper	JCR	SCIE	None
1	IEEE	Congress		YES	
2	IEEE	Congress		YES	
3	IEEE	Congress			X
4	IEEE	Congress		YES	
5	Scopus	Congress		YES	
6	Scopus	Congress			X
7	Scopus	Congress		YES	
8	Scopus	Journal	YES		
9	Science Direct	Journal	YES		
10	Science Direct	Journal	YES		
11	Science Direct	Congress			X
12	Science Direct	Journal	YES		
13	Web of Science	Congress			X
14	Web of Science	Congress		YES	
15	Web of Science	Congress		YES	

Table 2 Paper classification

each paper. The result is summarized in the same table 3.

Table 3 has the following columns:

- NL: Is close to Natural Language or at least declarative (vs imperative)?
- Script: Is a script-type language (vs. procedural)?
- Rules: Can it use rules (or similar constructions) to make decisions?
- Multi: Allows to interact in any way with other languages (Java, JS, Python, etc.)?
- Extend: Allows to extend in any way the language and/or its functionality?
- Devices: Provides a mechanism to incorporate physical devices?
- Comms: Provides communication protocols to access/interact with running instances?
- Girds: Allow collaboration of running instances and spreads logic and functionality?

Table 3 uncertainty explanation: cells with a question mark will now be explained (please refer to the number shown after the question mark and inside parentheses).

- (1) Because the proposed system is a cooperation between modular existing subsystems, and because the authors do not provide much information about the hidden details, it is impossible to know this topic; but as per our experience in this kind of cooperative system, the use of a scripting language would not be allowed.
- (2) The present proposal is comparable to (1); therefore, it is highly unlikely that the formation of cooperative grids will be provided.
- (3) This proposal, which involves an adaptation of C for the Arduino environment, is unlikely to enable the utilization of grids.

3.7. Exemplification process

As mentioned, a preliminary exploration was undertaken to gauge the suitability of five close-to-natural programming languages for the development in the field of the Internet of Things.

Traditionally, programming languages have been introduced with a "Hello world" application. In the case of the Internet of Things, its equivalent is lighting up a light-emitting diode (LED). In this section, we implement this basic program in each selected language, serving as an initial foray into their syntax and basic functionality.

Please note that due to the limited information provided, comprehensive code editing, compilation, and deployment processes were not feasible. Thus, the evaluation presented herein focuses solely on the initial code composition, as the full assessment of code suitability, efficiency, and performance would require a more extensive development and testing environment, which was provided only in one case: (Gabbrielli et al. 2018).

Nevertheless, this initial exercise offers an initial glimpse into the languages' ease of use and foundational constructs. Further in-depth evaluation would be necessary for a comprehensive assessment of their suitability for IoT applications.[9]

[9] Because it is beyond the scope and purpose of this paper to provide a more detailed view of each of these languages, we created a wider review which is available here: https://doi.org/10.5281/zenodo.10007572

	NL	Script	Rules	Multi	Extend	Devices	Comm	Grids
(Soic et al. 2020)	Yes	?(1)	Yes	No	Yes	Yes	Yes	Yes
(Amrani et al. 2017)	Yes	Yes	Yes	No	No	Yes	Yes	No
(Petnik & Vanus 2018)	Yes	No	Yes	No	No	Yes	Yes	?(2)
(Dong et al. 2020)	No	No	No	No	Yes	Yes	Yes	?(3)
(Gabbrielli et al. 2018)	No	No	No	Yes	Yes	Yes	Yes	Yes

Table 3 Papers classification framework

Soic et al. (2020)

```
#conditions
[condition]there is a command like ‘‘{message}’’
    =Event(message==’’{message}’’)

#consequences
[consequence]turn on ‘‘{connector}‘‘
    =Connectors.set(‘‘{connector}’’), ‘‘ON’’);

rule ‘‘Activation’’
    when
        there is a command like ‘‘activate light’’
    then
     turn on ‘‘light’’
end
```

Amrani et al. (2017)

```
configuration MyHome {
    node gw: Central
    node frontdoor: DoorLock
    node light: LightBulb

    from frontdoor to gw via MQTT
    from light to gw via MQTT
}

rule SwitchLightsWhenEnter:
    when (frontdoor.opened()) do {
            light.on()
        }
```

Petnik & Vanus (2018) The authors did not provide a sample, not even a rudimentary one, and it was not feasible to locate the software related to their paper.

Dong et al. (2020)

```
#include ‘‘TL_DeviceID.h’’

#define MAINBOARD ARDUINO_UNO
#define WIFI ESP8266_ESP01
#define LIGHT GROVE_LIGHT

#define WIFI_UART_RX 1
#define WIFI_UART_TX 0
#define LIGHT_ANALOG_OUTPUT 0

void setup() {
    TL_WiFi.init();
    TL_WiFi.join(‘‘SSID’’, ‘‘PASSWORD’’);
    TL.Light.setMeassuringRange(1,30000,’’LUX’’);
}

void loop() {
    TL.Light.write();
    upload();
}

void upload() {
    double light = TL_Light.data();
    double sm = TL_Soil_Moisture.data();
    String url = ‘‘http://hostname/ul.php?’’;
    url += String(‘‘light=’’) + String(light);
    TL_WiFi.get(url);
}
```

Gabrielli et al. (2018)

```
type LightType: void { .id: string } | int { .id: string }

interface LightInterface {
    RequestResponse: getTmp( LightType )( int )
    OneWay: setLight( LightType )
}

outputPort Light {
    Location: "datagram://light:5683"
    Protocol: coap {
        .osc.setLight << {
            .messageType = "CON",
            .messageCode = "POST",
            .alias = "/%!{id}/setLight"
        }
    }
Interfaces: LightInterface
}

main {
    setLight@Device( 1 { .id = "42" } )
}
```

4. Discussion

Let's revisit the initial research questions and provide answers to each of them.

4.1. Are there close-to-natural or at least declarative programming languages for the domain of the IoT?

Undoubtedly, the closer the programming language is to the natural language, the easier it will be for regular people (no programmers) to learn it. As shown in the previous table, there are 3 papers referring to Natural Language solutions: (Soic et al. 2020), (Amrani et al. 2017) and (Petnik & Vanus 2018). Although (Amrani et al. 2017) is the only one that is not hardware-eager, it is -unfortunatelly- the only one of these three that clearly does not support any kind of cooperation (grids). On the other hand, (Soic et al. 2020) and especially (Amrani et al. 2017) are very high-level systems that require considerable amounts of hardware and software, making them not suitable for all scenarios where hardware requirements must be kept to a minimum: quite a lot in IoT.

4.2. Can these languages be extended in any way?

Many programming languages can be extended by allowing the creation of callable subroutines. Such subroutines may be coded using the same language or incorporate a way to invoke subroutines developed in a different language. The inclusion of a mechanism to extend the language is regarded as critical by most developers.

Concerning this, and bringing back the previous table, we can see that meanwhile (Soic et al. 2020; Dong et al. 2020) and (Gabbrielli et al. 2018) allow some kind of extensibility, (Amrani et al. 2017) and (Petnik & Vanus 2018) do not.

4.3. Can these applications cooperate in any way (e.g. forming any kind of grids)?

Breaking down the total power of a system into smaller components has been successful in many situations, hence the phrase "divide and conquer" is part of our language. Regarding this, we were unable to find out if this ability is provided by (Petnik & Vanus 2018) or (Dong et al. 2020). However, (Soic et al. 2020) and (Gabbrielli et al. 2018) included this possibility, but (Amrani et al. 2017) did not.

4.4. Outcomes

In conclusion (Amrani et al. 2017) is the only language among the cited papers that is not hardware-eager, but lacks support for cooperation and has high hardware and software requirements. On the other hand, (Soic et al. 2020) and (Gabbrielli et al. 2018) are complex systems that require considerable resources, yet they offer the possibility of being extended and working together. (Petnik & Vanus 2018) and (Dong et al. 2020) do not explicitly state their support for expansion or collaboration.

5. Risks to the validity of this research

5.1. The coyote and roadrunner problem

This challenge is a pervasive phenomenon not only within the sciences but is particularly pronounced in the dynamic field of Information Technology. It resembles the timeless pursuit of the coyote in catching the elusive roadrunner, despite the unwavering diligence and determination. The coyote's pursuit is a metaphor for the perpetual chase for the latest developments and insights in the ever-evolving landscape of IT research.

Within the realm of Information Technology, the Internet of Things (IoT) is currently a focal point of extensive interest and research. This heightened attention has led to a constant influx of new publications, making the task of providing an entirely up-to-date and comprehensive study an arduous endeavor. The IoT domain, similar to the elusive roadrunner, continues to dart forward, introducing novel concepts, technologies, and applications on a daily basis. Keeping pace with this rapidly changing landscape is a formidable undertaking, as the research community strives to capture and analyze the most recent developments while they remain in perpetual motion.

5.2. The language risk

In the course of our research, we deliberately confined our search parameters to articles written in English, a decision that merits further discussion because this restriction is not without its implications. It is worth noting that the scientific community is truly global, with research endeavors extending across linguistic boundaries. By focusing solely on English-language articles, we acknowledge the existence of potentially relevant content that may be composed in other languages.

The decision to restrict our search to English is based on the recognition that English functions as the default language of science and technology. The prominence of English in academia and research is largely due to historical factors, including the scientific innovations of English-speaking countries and the globalization of higher education. In fact, a significant portion of academic writing, particularly in tech-heavy disciplines such as IoT and programming languages, are written in English. This topic has been the subject of much discussion in the academic literature, two examples of which are (Tardy 2004) and (Drubin & Kellogg 2017).

However, it is important to acknowledge that although English is the leading language, significant contributions to the field are not restricted to it. The decision to limit our search to English-language sources is a practical trade-off that acknowledges the dominant role of English in scientific and technological discourse.

5.3. The methodology risks

All methodologies have inherent limitations and imperfections. None is perfect and none ensures 100% infallible results, neither SLR. Although it is also true -as we mentioned previously- that this methodology is very well proven and extensively used, minimizing in this way incorrect results.

The following are some of the most commonly cited problems when using the SLR methodology (Oivo et al. 2018):

- Limited scope and bias: SLRs depend on the availability and quality of published literature. If relevant studies are limited or biased, they can affect the validity and reliability of the review's findings. Publication bias, where positive results are more likely to be published than negative or inconclusive ones, can distort the overall conclusions of an SLR.
- Heterogeneity and comparability of the included studies: variations in the studies can make it difficult to combine or compare the results. The synthesis of findings may be limited and quantitative meta-analysis may not always be feasible or appropriate.
- Quality assessment and risk of bias: assessing the quality and risk of bias in individual studies is a crucial step in an

SLR, but there is a lack of standardized criteria for quality assessment.

- Incomplete or outdated evidence (similar to the previously mentioned "coyote and roadrunner problem"): the process of conducting an SLR takes time, and by the time it is completed and published, new studies might have been published, rendering the review outdated.

5.4. The search criteria risk

It could be that the search criteria used were not optimal to obtain the most suitable outcomes. As previously said, we mitigated this by implementing a process of refining the keywords by executing several cycles of sending a search criterion and analyzing the results to identify those that yielded the highest level of precision.

5.5. The indexes risk

By excluding papers that do not appear in the JCR or SCIENCE indexes, we could be excluding relevant papers. Even if this is absolutely undeniable, it is also true that these are two of the indexes that include more academic and relevant papers.

5.6. Final thought

In conclusion after all is said, it is the sincere opinion of the authors of this paper that even if there are risks to the validity of the study and the data obtained, these were kept to a minimum or at least constrained to the scope of what is reasonable.

6. Conclusions and future works

In this paper, we carried out a systematic investigation using a well-established and widely used methodology (SLR) to determine what has been academically published at the time of writing this paper (Summer 2022) pertaining to scripting, declarative (akin to natural) languages in the field of IoT.

We started by implementing a rigorous methodology for conducting academic research papers. After identifying the most suitable sources to access these papers, we formulated our research questions and conducted multiple iterations of search criteria and evaluation of results until we identified papers that aligned with our objectives. We then applied a stringent selection criterion and meticulously analyzed the chosen papers, the outcomes of which are now presented. It is now time to present our conclusions.

6.1. Conclusions

Perhaps the initial finding that emerges is the stark contrast in hardware requirements and its correlation with the level of abstraction. Those operating at the highest levels of abstraction(Soic et al. 2020; Longo et al. 2021; Petnik & Vanus 2018) require significantly more hardware (alongside additional software components). On the contrary, those who require a lower-cost metal infrastructure (comprising not much more than an O.S. and a compiler) need to invest more effort and IT proficiency to achieve comparable or even less ambitious objectives (Hamdan et al. 2019; Gyrard et al. 2014; Dong et al. 2020).

There is a crucial fact that should always be kept in mind: although IoT covers appliances of any complexity in terms of hardware and software, many times we are referring to small, inexpensive commodities, of which household appliances or industry sensors are just two examples. But these low-end devices and top-hardware demanding systems are in the opposite corners of the room: it does not make much sense to propose a solution that needs a big computer to solve a problem that has to be shipped inside a 5€ production-cost chip. Clearly, research should not be constrained by anything except imagination, but it is also important to provide the best solutions using what we have today for today's problems.

At the other end of the spectrum, we should not forget that it is also a crucial IT concern to provide the better user experience, even (and specially) to those that are not IT professionals. IT companies struggle in providing best user experience (UX) to their customers and this could be especially tricky when trying to sell inexpensive products.

6.2. Future works

Towards a Declarative DSL for the IoT

Our research highlights the need for a simple, declarative programming language designed specifically for the IoT sector. Such a language would aim to facilitate usage among non-programmers or developers. The objective is to create a language that offers a smooth learning curve, enabling anyone with basic spreadsheet knowledge to create IoT applications after investing only a fraction of the time required by the most commonly used programming languages. Something that anyone proficient with spreadsheet software can utilize for simple applications such as managing household devices, but also equipped to handle issues of any kind and magnitude when employed by an IT specialist.

This solution should be in its simplest form small enough to be embedded in inexpensive appliances, but capable of being extended to handle much more ambitious requirements, although demanding more expensive hardware in these cases.

As mentioned previously and based on others' experience and our own, a programming language that is rule-based, declarative, and closely resembles natural language could be an effective approach. Although it may not be the only solution and may not be optimal in all situations, it strikes a good balance between various factors.

A Comparative Analysis of Imperative vs Declarative Languages for IoT

An important area for future research is conducting a thorough comparative analysis of both imperative and declarative programming languages in the IoT context. Such a study could offer valuable insights into how well these paradigms match with the IoT applications' distinctive requirements and challenges. Let take a brief look into how a "Rationale for the Comparative Analysis" could be:

- *Holistic understanding*

When examining the domain of IoT, comparing imperative and declarative languages can lead to a comprehensive understanding of the trade-offs between the two paradigms. With this understanding, researchers can make informed decisions when selecting the most suitable language for individual IoT projects.

- Scalability and Resource Efficiency

IoT applications often require efficient resource utilization and scalability. Conducting a comparison can provide insight into which paradigm is best suited for optimizing resource usage and overcoming scalability challenges.

- Development Productivity

A comparison could investigate development productivity aspects, like code complexity, maintainability, and ease of learning, to determine which paradigm is more efficient and error-resistant in IoT application development.

- Real-Time Constraints

IoT frequently requires real-time data processing and response. An analysis can evaluate how well each paradigm fits real-time constraints, such as low latency and high-throughput requirements.

References

Amrani, M., Gilson, F., Debieche, A., & Englebert, V. (2017). Towards user-centric dsls to manage iot systems. In *Modelsward* (pp. 569–576).

Cambranes, E. (2013). Supporting novice programmers with natural language in the early stage of programming. In *2013 ieee symposium on visual languages and human centric computing* (pp. 173–174).

Dong, W., Li, B., Guan, G., Cheng, Z., Zhang, J., & Gao, Y. (2020). Tinylink: A holistic system for rapid development of iot applications. *ACM Transactions on Sensor Networks (TOSN)*, *17*(1), 1–29.

Drubin, D., & Kellogg, D. (2017). English as the universal language of science: opportunities and challenges. *Molecular Biology of the Cell*, *23*(8).

D'Urso, F., Longo, C., & Santoro, C. (2019). "programming intelligent iot systems with a python-based declarative tool". In *Ceur workshop proceedings* (Vol. 2502).

Ferry, N., Nguyen, P. H., Song, H., Rios, E., Iturbe, E., Martinez, S., … others (2020). Continuous deployment of trustworthy smart iot systems. *The Journal of Object Technology*.

Gabbrielli, M., Giallorenzo, S., Lanese, I., & Zingaro, S. P. (2018). A language-based approach for interoperability of iot platforms..

Gyrard, A., Bonnet, C., & Boudaoud, K. (2014). Demo paper: Helping iot application developers with sensor-based linked open rules. In *Tc/ssn@ iswc* (pp. 105–108).

Hamdan, O., Shanableh, H., Zaki, I., Al-Ali, A.-R., & Shanableh, T. (2019). Iot-based interactive dual mode smart home automation. In *2019 ieee international conference on consumer electronics (icce)* (pp. 1–2).

Keene, J., & Jamil, H. M. (2022). Natural language programming with trypl. In *2022 international conference on advanced learning technologies (icalt)* (pp. 38–40).

Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and software technology*, *55*(12), 2049–2075.

Kumar, S., Kumar, B., K, S., & R, R. (2021). "iot based secured home automation system using nlp". In *International conference on advancements in electrical, electronics, communication, computing and automation (icaeca)* (pp. 1–5).

Lago, A. S., Dias, J. P., & Ferreira, H. S. (2021). Managing non-trivial internet-of-things systems with conversational assistants: A prototype and a feasibility experiment. *JOURNAL OF COMPUTATIONAL SCIENCE*, *51*.

Li, Z., Peng, X., Chao, L., & Xu, Z. (2018). Everylite: A lightweight scripting language for micro tasks in iot systems. *THIRD IEEE/ACM SYMPOSIUM ON EDGE COMPUTING (SEC).*, 381-386.

Lombardi, T., Cortellessa, V., Pierantonio, A., & Model, I. (2021). Co-evolution of metamodel and generators: Higher-order templating to the rescue. *J. Object Technol.*, *20*(3), 7–1.

Longo, C. F., Longo, F., & Santoro, C. (2021). Caspar: Towards decision making helpers agents for iot, based on natural language and first order logic reasoning. *Engineering Applications of Artificial Intelligence*, *104*, 104269.

Oivo, M., Méndez, D., & Mockus, A. (2018). *Esem '18: Proceedings of the 12th acm/ieee international symposium on empirical software engineering and measurement.* New York, NY, USA: Association for Computing Machinery.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In *12th international conference on evaluation and assessment in software engineering (ease) 12* (pp. 1–10).

Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and software technology*, *64*, 1–18.

Petnik, J., & Vanus, J. (2018). Design of smart home implementation within iot with natural language interface. *IFAC-PapersOnLine*, *51*(6), 174–179.

Poroor, J. (2018). "work-in-progress: Verticalthings - a language-based microkernel for constrained iot devices". In *International conference on embedded software (emsoft)* (pp. 1–3).

Soic, R., Vukovic, M., & Jezic, G. (2020). Speech controlled iot system based on context-driven rule engine. In *2020 international conference on innovations in intelligent systems and applications (inista)* (pp. 1–7).

Tardy, C. (2004). The role of english in scientific communication: lingua franca or tyrannosaurus rex? *Journal of English for Academic Purposes*, *3*(3), 247–269.

Yu, R., Lu, W., Lu, H., Wang, S., Li, F., Zhang, X., & Yu, J. (2021). "sentence pair modeling based on semantic feature map for human interaction with iot devices". In *International journal of machine learning and cybernetics. vol. 12(11)* (pp. 3081–3099).

Zúñiga, A., Sierra, G., Bel-Enguix, G., & Gomez, J. (2020). "siciot: A simple instruction compiler for the internet of things". In *Internet of things. vol. 12* (p. 100304).