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 In this paper, we will present our findings about other people's works in the area of programming languages close to natural language for the domain of the IoT (Internet of Things). In other words, we will review and compare what others have done previously in the same area. To accomplish this task, we will use the well-known Systematic Literature Review (SLR) methodology in the area of Software Engineering to conduct our research.
We will set the questions we try to answer, perform several searches through most relevant academic documents searchers, later we will systematically filter the results and we will analyze every resulting papers looking for certain key points. Finally, we will present the results and discuss the risks in this research as well as the results obtained. We will end proposing possible future works.

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 after day, in the area of the Internet of Things (IoT), the variety of devices, communication protocols, gateways, etc. is being increased. The nature of this problem is so vast that there are studies dedicated just to propose approaches that could deal with it: there have also been many efforts trying to adapt known and proven strategies and theoretical models into the area of IoT, like the use of Model-Driven Engineering and use of metamodels: "...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 before: the gap in terms of technical complexity between them is enormous.

In terms of the amount of them currently in use, it is impossible to set a number of the worldwide connected devices today, specially because this number increases literally every second; but in August 2022 it was estimated that by the end of 2022 there will be 13.1 billions ($13.1x10^9$) worldwide, and the estimated amount by 2025 is 75.4 billions ($75.4x10^9$)[1] .

To make matters worse, almost every IoT manufacturer has its own proposal to solve the problem of providing to common-users (not programmers) a software tool to facilitate the use of their physical products (hardware): these can be as simple as an app for a smart phones[2] to manage let's say a RGB light; or as complex as a SCADA system[3], to manage let's say a bottling machine production line in a beer factory.

As in any other IT field, manufacturers' proposals cover a

[1] https://techjury.net/blog/how-many-iot-devices-are-there/
[2] https://play.google.com/store/apps/details?id=cloud.blynk https://play.google.com/store/apps/details?id=com.coolkit https://play.google.com/store/apps/details?id=com.nest.android&gl=US
[3] https://www.onlogic.com/solutions/industrial-computers-with-integrated-software/ https://new.siemens.com/global/en/products/automation/industry-software/automation-software/scada.html

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

wide range in terms of complexity: some of them propose zero-programming solutions, meanwhile others offer programming languages or a mix of both. As in any other aspect of life, each approach has advantages and disadvantages. For achieving simple tasks, zero-programming 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 to solve complex problems. When talking about programming languages, it appears as an undeniable idea that a natural language is the best option for non-programmers audience (Cambranes 2013), meanwhile 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 problems. We needed a lingua franca, or at least a framework that would allow us to work with IoT devices from different manufacturers, and we also needed the learning curve for using this tool to be as smooth as possible; if possible using natural language, or as close to it as possible. Because of all this, we thought it would be good to give developers, but also non-technical people, a simple way to address these scenarios -ideally something close to 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 an Systematic Literature Reviews (SLR) of available academic publications in the field of the declarative (and as close to natural as possible), scripting programming languages in the area of Internet of Things (IoT).

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. The next section is about to 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 works.

2. Methodology

There are different methodologies proposed in the literature that allow the 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 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.

Both methodologies share the common objective of conducting a methodical analysis of extant literature. However, SLRs emphasize a comprehensive examination and integration of carefully selected studies to address precise research inquiries, whereas SMSs offer a comprehensive survey and classification of the literature to identify prevalent trends and discern research gaps within a given field.

Due to the nature of this research, and considering that the number of published studies related to the subject matter of this paper 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, conduct 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 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

[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

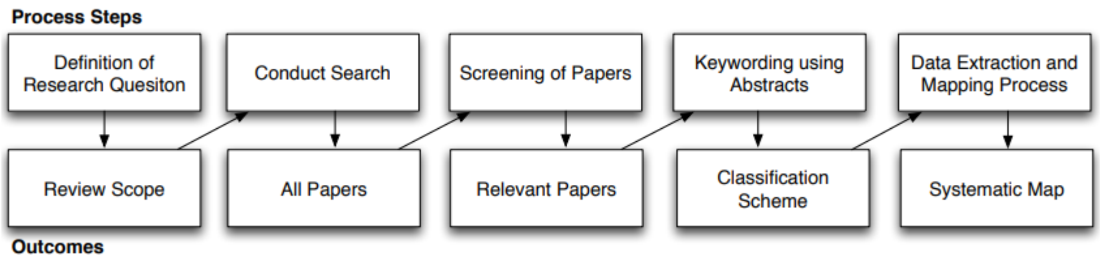

Figure 1 Systematic Mapping Study methodology by Petersen et. al (Petersen et al. 2008)

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 is organized and synthesized in a structured manner.

3. Execution

3.1. Definition of the research questions

To find the state-of-the-art in IoT programming languages that facilitate their use even to non-programmers meanwhile keeping the capability to address complex projects of any size to professionals, we have chosen a set of questions that appear to be clear and unambiguous and which this paper will respond by systematically analyzing scientific papers up to the date of performing a research by systematically applying the SLR methodology:

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

3.2. Conduct search

Our first step was to search where to search, or better to say: which are the most relevant searchers for finding academical publications besides the generic search engines (Google, MS-Bing, DuckDuckGo, etc.).

As a result, we found a bunch of places. After visiting these academica engines and reviewing what others said about them, we tested those most mentioned and better valued by the science community. Not surprising, the selected search-engines are those vastly used in academic publications:

- Scopus
- IEEE Explore
- Science Direct
- WebOfScience

As the title of this essay shows and as was mentioned previously, the goal is to find publications related to computer languages -close to natural- specific for the IoT domain.

We started by executing several cycles of sending a search criterion and analyzing the results to find out those criteron that produce the most accurate results. After this and to broad the search trying to maximize the hits, we performed the ones appearing in Table 1.

As search engines allow us to perform “smart” searches, initially, we searched for the term “iot” not only in the title but also in the metadata as well as in the keys. Obviously, we obtained more hits, but after examining the results more in detail, we realized these results were not as relevant as we expected. So, as mentioned above, we performed several combinations in a try-and-improve cycle, finding out that the searches shown in Table 1 are the ones that best fits our case, returning better (more accurate) results. Having fewer results (while being accurate) has another significant advantage: we can go through all of them in detail.

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 paper is not indexed in JCR[6] neither in SCIE[7], then discard it.
4. Read the full paper in case of doubts to either select or discard the paper.

Figure 2 diagram shows the whole 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 (vs. imperative)?

[6] https://jcr.clarivate.com/
[7] https://scie.lcc.uma.es:8443/gii-grin-scie-rating/ratingSearch.jsf

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

- 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. Data extraction and mapping 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)
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 papers regarding Exclusion Criteria number 3 (Indexed in JCR6 or in SCIE).

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. Not even making a single reference to a programming language.

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

(10) Can not pass 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. And can be used only by senior engineers.

(11) Cannot pass 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.

After these Exclusion Criterias were applied, the resulting papers are shown in this succinct Table 3 as first row cell. As mentioned, we identified several key points that were pursued in each paper. The result is summarized in the same Table 3.
Table 3 has following columns:

- NL: Is close to Natural Language or at least declarative (vs imperative)?
- Script: Is a script type language (vs procedural)?

[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)

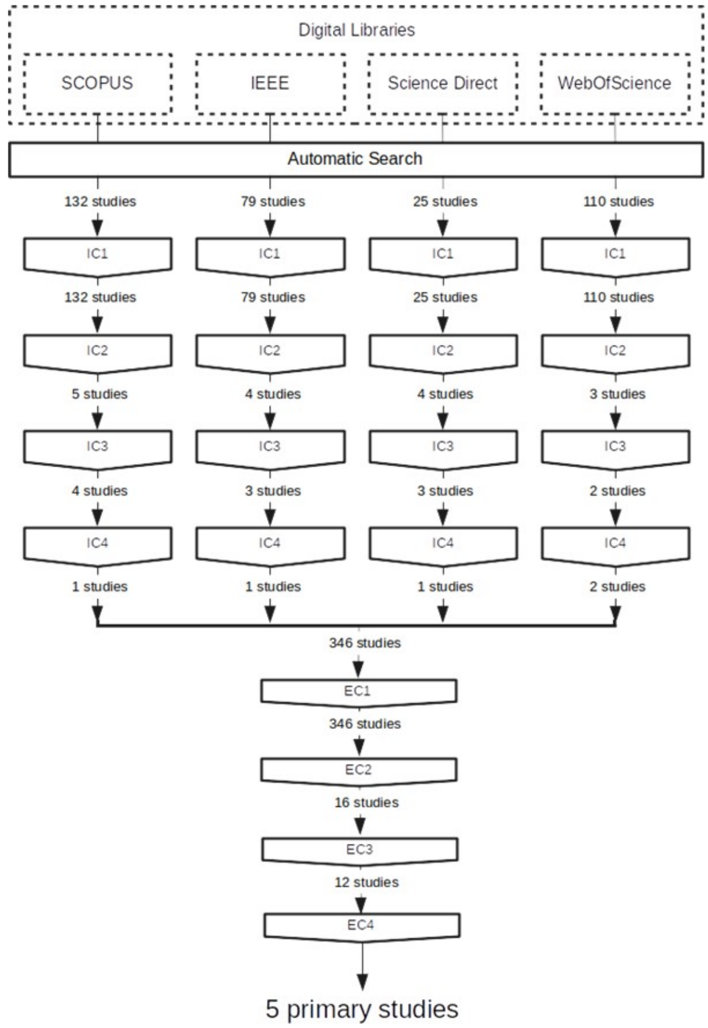

Figure 2 Execution results

- Rules: Can it use rules (or any 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 protocols to access/interact with running instances?
- Girds: Allow collaboration of running instances and spreads logic and functionality?

Table 3 uncertainties explanation: cells with a question mark, will be now explained (please refer to the number in cells after the question mark and inside parenthesis).

- (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 systems, it would not allow the use of a scripting language.
- (2) The case of this proposal is exactly as (1): not in vane both proposals are very similar. So, we should conclude it is very unlikely they facilitate the creation of cooperative grids.
- (3) Because this proposal uses quite low level (the adaptation of C done for the Arduino ecosystem), it is very unlikely it allows the use of grids.

4. Discussion

Let's bring back the initial questions that leaded this research and let's provide an answer to each one of them.

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

Undoubtedly, the closer to the natural language a programming language will be, the easier it will be to be learned by regular people (no programmers). As we've seen 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 only (Amrani et al. 2017) is not a hardware-eager, unfortunately (Amrani et al. 2017) is the only among these three

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

	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

that clearly do not support any kind of cooperation (grids). On the other hand, (Soic et al. 2020) and specially (Amrani et al. 2017) are very high-level systems which demand considerable amounts of hardware and software, making them not suitable for all those scenarios where hardware requirements must be kept to minimum: quite a lot in IoT.

4.2. Can these languages be extended in any way?

Many programming languages can be extended by allowing to create some kind of callable subroutines. These subroutines sometimes can be written using the same language, other times the language incorporates a way to invoke subroutines written in a different language. To be able to extend the language is considered crucial by most part of the developers. Regarding 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 the applications created cooperate in any way (v.g. forming any kind of grids)?

Dividing the total power of a system into individual smaller pieces is an approach that has proven its idoneity in many scenarios, not in vain "divide and conquer" form part of our vocabulary. Regarding this ability, we were not able to find out if this ability is provided by (Petnik & Vanus 2018) and (Dong et al. 2020), but meanwhile (Soic et al. 2020) and (Gabbrielli et al. 2018) included this possibility, (Amrani et al. 2017) did not.

4.4. Outcomes

In summary, the conclusions from what said above suggest that (Amrani et al. 2017) is the only language among the mentioned 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 high-level systems that demand significant resources but provide extensibility and cooperation capabilities. (Petnik & Vanus 2018) and (Dong et al. 2020) do not explicitly mention their support for extensibility or cooperation.

5. Risks to the validity of this research

5.1. The coyote and roadrunner problem

This is a common problem in live sciences and especially in IT: no matter how hard the coyote will struggle in catching the roadrunner: he will never succeed. The IoT area is receiving a lot of attention nowadays and therefore new publications appear daily, making it impossible to provide an up-to-date study. The language risk We only searched among English-written articles; therefore, there could be relevant articles written in other languages. It is also true and it must be said that most articles -specially those in the fields of technologies- are written in English.

5.2. The methodology risks

All methodologies suffer from shortcomings, and none is perfect or 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, it can impact 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 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.

5.3. The search criteria risk

It could be that the search criteria used were not the best to obtain the appropriate results. 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 find out those that produced the most accurate results.

5.4. The indexes risk

By excluding papers that do not appear in JCR or SCIENCE indexes, we could be excluding relevant papers. Even if this is absolutely undeniable, it is also true that these are 2 of the indexes that include more academic and relevant papers. A final word about risks: after all this 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 conducted a systematic research using a well-proven and extensively used methodology (SLR) to find what is academically published as per the time of writing this paper (Summer 2022) in the field of scripting, declarative (close to natural) languages in the area of IoT.

We started by choosing a well-proven methodology in the area of conducting academic papers research, we looked for the best places to search such papers and identified those places, we sat the questions we wanted to answer in this research and performed several “search criteria / evaluation of the results” iterations until we found those papers that better fitted to our purpose, we performed a well-defined selection criterion, carefully analyzed the papers that passed the cut and we exposed the results. Now is the time to present the conclusions.

Perhaps the first fact that clearly emerges is the polarization in terms of hardware needs and the correlation with the abstraction level: those at the highest level of abstraction (Soic et al. 2020; Longo et al. 2021; Petnik & Vanus 2018) also need much more hardware (and more software pieces); on the other side, those demanding cheaper metal infrastructure (no much more than an O.S. and a compiler) demand more effort and IT knowledge to achieve similar or even less ambitious goals (Hamdan et al. 2019; Gyrard et al. 2014; Dong et al. 2020).

But there is a crucial fact that we should always keep in mind: although IoT covers appliances of any complexity in terms of hardware and software, many times we refer to small, inexpensive commodities (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 (sometimes something close to a Data Center) to solve a problem that has to be shipped inside a 5€ production-cost chip. Clearly research should not be constrained by anything except the imagination, but it is also important to provide the best solutions using what we have today for today problems.

At the other end of the spectrum, we should neither 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, not in vane, IT companies struggles in providing best UX to their customers; this could be specially tricky when trying to sell inexpensive products.

As a result of this research, we've found the necessity of having a programming language for the area of the IoT as simple as possible to facilitate its use by no programmers (developers). The goal would be to provide a language with such a smooth learning curve that anyone having basic knowledge of spreadsheets should be able to create applications in the area of IoT after investing in learning it just a fraction of the time needed nowadays when using most used languages. Something that any person that can handle a spreadsheet could use for basic purposes (e.g., manage the devices at his/her own house), but capable to address problems of any kind and size when used by an IT professional.

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

As mentioned before and based on others' experience and our own, a rule-based, declarative, and close to natural-language scripting programming language could be a great approach. Clearly not the only one, perhaps not the best one in all scenarios, but probably the best balanced one.

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.

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).

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).