Subject: Response to request for manuscript improvements.

Dear Editor,

We revised the manuscript, entitled "An Overview of the Declarative Programming Languages for the IoT Domain", which was recently reviewed by JOT staff. We would like to express our sincere appreciation for the JOT staff's careful review and valuable feedback, which greatly contributed to enhancing the quality of our work.

We have made significant enhancements to the manuscript based on the concerns raised, and we are confident that we have successfully addressed them. We have adhered to style guides, employed precise word choices, and ensured grammatical correctness. We are pleased to share with you the improved version of the manuscript.

The following issues were raised…

The paper addresses a relevant topic. However, despite the authors' efforts to enhance the original submission, the manuscript's readiness for acceptance remains incomplete due to the following primary reasons:

- The authors conflate the objectives of Systematic Literature Reviews (SLRs) and Systematic Mapping Studies (SMSs). As per [1], "Systematic mapping studies structure a research area, while systematic reviews focus on collecting and synthesizing evidence." Please refer to Section 2.1, where the authors state:
    - "Research questions in mapping studies are general, aiming to discover research trends (e.g., publication trends over time, topics covered in the literature). Conversely, systematic reviews aggregate evidence, necessitating precisely formulated specific goals (e.g., the practical utility of an intervention for the industry)."

After careful consideration, we acknowledge that our initial submission conflated the objectives of Systematic Literature Reviews (SLRs) and Systematic Mapping Studies (SMSs). To clarify the issue and enhance reader comprehension, we have revised the manuscript accordingly. We added a new paragraph in Section 2, paragraph 2, outlining the principal distinctions between Systematic Literature Reviews (SLRs) and Systematic Mapping Studies (SMSs), to foster more distinct identification of the two methodologies:

As described by Petersen et al. (2015), both methodologies intend to perform a methodical analysis of preexisting literature. Nonetheless, they differ in several respects. Firstly, SMSs typically have broad research questions that aim to uncover research trends, while SLRs require a specific research goal for evidence aggregation. Secondly, the research process in SLRs is guided by a specific topic area, whereas SMSs are driven by specific research questions. Thirdly, SMSs have less strict search requirements compared to SLRs, which demand the inclusion of all relevant studies. Fourthly, 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.

The protocol description in the initial version of the paper has been enhanced. However, the concern raised in the previous review about "Insufficient Problem Definition and Discussion" remains partially unresolved. Considering the defined research questions, the authors aim to conduct a systematic literature review on distinct facets of the IoT domain, mainly focusing on domain-specific languages proposed within this domain. Further clarification of the intended purposes of the languages of interest is essential. For instance, are languages being examined solely for specifying workflows (akin to Node-RED), or are approaches considered that encompass broader aspects, including design and deployment, as supported by ThingML?

Building upon my prior comment, I recommend refining the research questions to encompass the complete IoT system life cycle from a software and system engineering perspective. This broader scope would facilitate identifying language characteristics beyond generic attributes like declarativity or extensibility.

a) We understand the suggestion to broaden the scope to encompass the entire IoT system life cycle, from a software and system engineering perspective. While this is undoubtedly a valuable research avenue, it is important to clarify our specific focus. In this paper, we intentionally narrow our scope to the intrinsic characteristics of programming languages themselves, with the aim of aiding developers and engineers in selecting the most suitable declarative programming language for their IoT projects.
The considerations mentioned are important for developers to choose a language, but the language's itself is more crucial, including its ease of learning and scope. This is especially relevant since the same language can often be interpreted or compiled, since developers have more than one editor or even IDEs to work with same language. Same about debuggers, optimization tools, etc.

b) We acknowledge our awareness of various software and products within the IoT domain that might be considered highly relevant. However, it's crucial to note that while these software and products play a pivotal role in IoT development, they may not be the central focus of our research. Our emphasis lies in evaluating the suitability of declarative programming languages for IoT application development, rather than discussing specific software solutions or platforms.
The results obtained through our research are intended to be applicable to a wide range of IoT software and products, providing insights into how well declarative programming languages can facilitate the fundamental aspects of IoT application development. This knowledge can guide decision-makers, developers, and engineers in making informed choices when it comes to selecting the appropriate programming language for their IoT projects, irrespective of the specific software or product they may choose to implement.

c) Regarding about to include or not imperative languages: before we started this research, we spent several days debating whether we should include both declarative and imperative languages. We even conducted some proofs of concept (some searches and result tables). It turned out that it is very confusing to draw common conclusions between them because the only thing both types of languages have in common is the name: "programming languages". Imperative languages are close to a finite state machine (Turin Machine), while Declarative languages are close to the Alonzo Church’s lambda functions. Declarative languages lack loops, jumps, and (some of them) even variables, among many other not so evident details; making them hard and confusing when mixing with

imperatives. So, We for sake of clarity, decide to focus solely on the declarative languages, an area that is (on the other hand) comparatively less researched.

To clarify the scope an limitations of our research, we added following to the existing text in Section 1 “Introduction”, subsection 1.2. “Research goal”

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 delimited 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.

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

In addition, we have divided Section “6. Conclusions and Future works” into a brief introduction and 2 subsections: “6.1.Conclusions” and “6.2.Future works”.
Under 6.2 we created 2 fields for future works:

- *One titled “Towards a Declarative DSL for the IoT”. This one is about the convenience of creating an DSL for the IoT (it existed previously).*
- *A new one titled “A Comparative Analysis of Imperative vs Declarative Languages for IoT”.*

This is the text for the second new field:

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.

To augment the paper's strength, I propose providing an overview of the identified languages, enabling readers interested in the domain to grasp key elements of existing approaches through illustrative examples. For instance, presenting comparable applications modeled using different approaches would facilitate comparisons based on predefined features outlined in this paper.

We have taken the following 3 actions:

1. We wrote a separate 12 pages paper, more precisely a technical review, covering the 5 final selected DSLs.
2. We made this document public so that it could be referenced from the target paper of this review. The reference is in 3.7 as footnote.
3. We created an IoT "hello world" source code for 4 of the 5 DSLs (it is impossible to do this for one of them and we explain it). This new content has been added to the new section 3.6. “Data analysis and exemplification process” with following contents:

[text]
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: \cite{gabbrielli2018language}.

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

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

[Above text is followed by the “hello world” examples]
It is attached in the email

The manuscript contains multiple instances of English errors that necessitate correction.

The document has been carefully revised and enhanced by an individual with an American background, who possesses a literature degree from the University of Houston.

Throughout this process, small pieces of information were removed, small pieces of information was added and some paragraphs were rewritten along the entire document.

In addition to your comments:

- We have made some minor restructuring throughout the document.
- We have identified an improvement in Section 5, so we have added a new subsection entitled "The Language Risk", as follows:

The language risk

In the course of our research, we deliberately confined our search parameters to English-written articles, a decision that merits further discussion. 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 stems from the recognition that English functions as the default language of science and technology. English's preeminence 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, is written in English.

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.

After careful consideration, we respectfully request that you review this revised text. It is our belief that this version satisfies the rigorous standards established by JOT and represents a valuable contribution to the academic field.

We cannot say goodbye without thanking you for your patience and the extension you have granted us. Since the date of your review coincided with the start of the academic year for both our students at the University of Seville and our children at their schools, it would have been impossible for us to implement these changes without such an extension.

Thank you for your attention to this matter. We eagerly anticipate your feedback and stand available to provide any further information or clarification necessary.

Yours sincerely,
Francisco José Morero-Peyrona (on behalf of the authors)