Journal of Internet Services and Applications, 2022, 12:1,10.5753/jisa.2022.2372

The state of the art of macroprogramming in IoT: An update

Iwens G Sene Jr [**Universidade Federal de Goiás**| ***iwens@ufg.br***]
Thalia S. Santana [**Universidade Federal de Goiás**| ***thaliassantana15@gmail.com***]
Renato F. Bulcão-Neto [**Universidade Federal de Goiás**| ***rbulcao@ufg.br***]
Barry F. Porter [**Lancaster University** | ***b.f.porter@lancaster.ac.uk***]
Received 30 November 2021 • **Accepted** 7 July 2022 • **Published** 18 November 2022

Abstract
Macroprogramming's primary goal is to increase developers' productivity by providing high-level specifications of applications' behaviour at the system level. Macroprogramming may be a viable solution for developing complex IoT applications, such as those manipulating high data volume and heterogeneity. This paper updates a recent work identifying and analysing primary research on macroprogramming in IoT through a systematic literature mapping (SLM). We extended the search strategy scope by conducting an automatic search over five new databases and also performed the snowballing technique. As a result, besides the 38 studies group found in previous SLM, nine new papers were classified as relevant and rigorously analysed, totalising forty-seven studies. In comparison to previous work, results still point out the recurrence of abstractions in the network infrastructure, highlighting the use of frameworks in one-third of the applications and contributing with an overview of macroprogramming by researchers in different knowledge areas.

Keywords: Internet of Things, Wireless Sensor Network, Systematic Mapping, Adaptation, Programming abstraction

1 Introduction

The literature has proposed a variety of macroprogramming approaches to mitigate the challenges of making programming more efficient for IoT and WSN applications (Madden *et al*., 2005; Newton *et al*., 2007). In this model, high-level programs are written to represent the overall control logic of an IoT deployment, and parts of this high-level program may then be compiled into low-level code and pushed directly into IoT devices.

Macroprogramming allows developers to specify a single program which defines the high-level collaboration behaviour for WSNs and IoT applications at the system level, treating the entire network as if it were a "single abstract machine" (Sugihara and Gupta, 2008). Low-level network and device details, such as state maintenance or message transmission, are intentionally hidden from the programmer through an automated translation and deployment of the macroprogram into per-node logic. As a result, macroprogramming, which has already been heavily used in works related to WSN (Newton and Welsh, 2004a; Gummadi *et al*., 2005; Mottola and Picco, 2011) is emerging as a viable technique for the development of complex and distributed IoT applications, as demonstrated by a number recent efforts (Noor *et al*., 2019; Hammoudeh *et al*., 2021a).

Despite a large range of proposals for macroprogramming paradigms, there remains a lack of convergence on the right abstractions that users most benefit from across different application domains. In situations such as when a new device is added into the target deployment, for example, an IoT application should therefore be capable of autonomously integrating the new device and assign to it a task that contributes to the overall goal of the application as defined in the macroprogram (Kephart and Chess, 2003; Salehie and Tahvildari, 2009; Alajlan and Elleithy, 2014). Self-adaptation is the ability of a system to reconfigure itself automatically and dynamically in response to changes, by installing, updating and integrating existing software elements with alternative ones at run-time.

Recently, researchers have proposed various macroprogramming approaches to mitigate the challenges of making programming more efficient for IoT and WSN applications (Mizzi *et al*., 2019; Qiao *et al*., 2018). In this model, high-level programs represent the overall control logic of an IoT deployment, and excerpts of this high-level program may then be compiled into low-level code and pushed directly into IoT devices. There are three significant benefits of this approach. Firstly, it allows the programmer to work at a higher level of abstraction and encode the business logic of an entire deployment in a top-down way, which is often significantly more straightforward than writing code for individual nodes. Secondly, as selected control logic from this high-level model can be deployed into the IoT, data processing can still be performed on IoT devices saving significantly in long-haul communication costs to the cloud. This strategy can also provide lower-latency decision and actuation control where IoT-resident control logic is positioned closer to the data sources on which decisions are being made. Finally, as high-level macro logic will often not specify the fine details of exactly where specific data should come from or where control logic should execute, this offers attractive degrees of freedom in the often dynamic deployment environments of IoT systems – so that the macroprogram translator can perform real-time tuning of which data sources are being used and work around failures and node mobility in the placement of control logic.

To establish baselines for comparison with ongoing, recent research results or even identify suitable areas for future research, a systematic literature mapping (SLM) may be of great usefulness (Petersen *et al*., 2015). An SLM identifies, selects, evaluates, interprets, and summarises relevant studies about a topic, e.g., macroprogramming in IoT/WSN.

This paper extends a previous SLM we conducted on how the macroprogramming paradigm has been investigated in IoT and WSN (Santana *et al.*, 2021). In such previous work, the search strategy included automatic search in three sources, namely ACM DL, IEEE Xplore, and Scopus. In this updated SLM, we searched on five new sources of studies and performed the snowballing technique to find more relevant studies. The snowballing technique (Wohlin *et al.*, 2012) allows identifying relevant studies through the scanning of the list of bibliographic references or citations of a paper.

Considering this broader search scope, this SLM classified nine new studies as relevant in a total of 47. The contribution is the mapping of the macroprogramming's state of the art in IoT, identifying the level of adaptations performed, with trends in abstractions applied to the group of nodes. Besides, we show how it has been used in WSN and its increasing shift to IoT research in recent years.

This paper is organised as follows: Section 2 overviews the SLM, Section 3 discusses the SLM results, and Section 4 brings our concluding remarks and future work.

2 Materials and Methods

The SLM presented in this paper is depicted in **Figure 1** and includes three phases: planning, conducting, and publishing. First, a protocol is planned so that one can reproduce it later. It includes research questions, search strategy, search string, sources of studies, and study selection criteria.

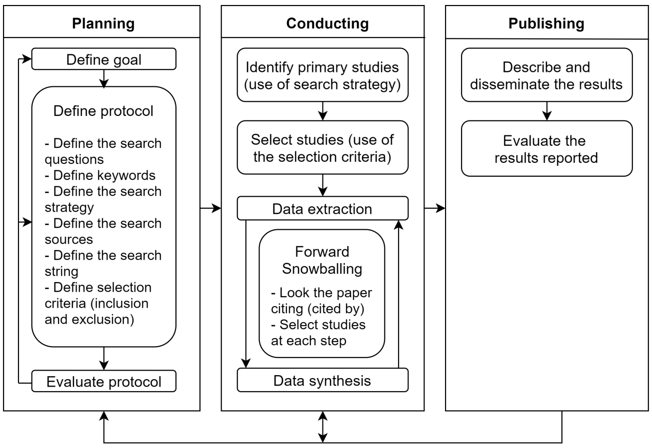

Figure 1. Phases and activities of this SLM's update.

In the conduction phase, studies gathered from search sources are initially selected through studies' metadata reading and applying inclusion and exclusion criteria previously planned. After, helpful information is extracted from these selected studies that, in turn, can still be excluded using the same selection criteria. As a novelty in this SLM's update, snowballing is performed by checking the citation list of the resulting papers of the data extraction step. This process, called forward snowballing, finishes when no new study is included. Following the SLM goal, the studies remaining of this whole process constitute the set of relevant papers from which answers for the research questions of the protocol are analysed and synthesised.

Finally, the entire protocol and the results of each previous stage are documented as scientific papers or technical reports in the publishing phase.

2.1 Research questions and search terms

This SLM's main goal is to identify primary research investigating macroprogramming with abstractions for WSN and IoT, which must also perform adaptations at the infrastructure level. The following are the research questions (RQ) we elaborated to be answered in this SLM:

- **RQ1:** What are the application domains found in primary studies?
- **RQ2:** When and where are primary studies published?
- **RQ3:** At what levels does adaptation occur, and what are abstraction types in the infrastructure?
- **RQ4:** How are adaptations carried out in primary research on WSN and IoT?
- **RQ5:** What are the adaptability-related issues found?

The next step was to select the proper search terms to identify the most relevant primary studies to answer these research questions. Helped by experts in macroprogramming and IoT, we chose the following set of candidate search terms for the definition of the search string: *macroprogramming, macro-programming, declarative approach, imperative approach, programming abstraction, high level, internet of things, cyber physical, cyber-physical, sensor networks, and wireless sensor networks*.

2.2 Automatic search

After evaluating the trade-off between coverage and relevance of the search results in a pilot search, we opted for the following combination of keywords as the final search string:

(macroprogramming OR "macro-programming" OR "declarative approach" OR "imperative approach" OR "programming abstraction") AND ("high level") AND ("internet of things" OR "sensor networks")

Specialists in macroprogramming, IoT, and systematic literature research contributed to the search string definition process.

In our previous work, we adapted the final search string to the *ACM DL, IEEE Xplore*, and *Scopus's* search engines (Santana *et al.*, 2021). In this SLM's update, we also performed searches on studies metadata at the *Engineering Village, ScienceDirect, Springer Link, Web of Science*, and *Wiley* websites. Finally, it is worth mentioning that we chose the *ACM Guide to Computing Literature*[1] option because it indexes both the full-text collection of ACM publications and other digital databases on Computing. This search option turns the ACM DL into the most comprehensive bibliographic database on Computing.

[1]More information can be found at `https://libraries.acm.org/digital-library/acm-guide-to-computing-literature`.

Table 1 details the number of studies retrieved in each source of study. There is a differentiation between the original research[2] and this revisited work[3]: forty-four studies were identified in this extended version (including duplicate documents) after adding five new sources and updating the search results in the three original sources.

Table 1. Number of studies returned per source.

Source	Original	Extension	Difference
ACM Digital Library	16	16	0
IEEE Xplore	15	15	0
Scopus	80	85	5
Engineering Village	-	23	23
Science Direct	-	1	1
Springer Link	-	5	5
Web of Science	-	10	10
Wiley	-	0	0
Total	111	155	44

2.3 Study selection and data extraction

We applied the same original selection criteria to the 155 papers returned by the automatic search process (Santana *et al.*, 2021). The exclusion criteria (EC) are:

EC1: The paper does not describe primary research.
EC2: The document retrieved is not a paper (e.g., preface or summary of journals or conference proceedings).
EC3: The full study text is not in English.
EC4: The full study text is not accessible.
EC5: The paper was not published before 2004.
EC6: The paper does not address the IoT or WSN domains.
EC7: The paper does not propose, report, or evaluate the usage of adaptation in the context of macroprogramming for programming abstractions.
EC8: The paper is a preliminary or short version of another study.

A paper is removed from this SLM whenever it meets at least one of the exclusion criteria (EC) presented. Otherwise, the study is categorised based on the only inclusion criteria (IC): “the study reports on the adoption of abstraction in programming and adaptation in infrastructure in IoT and WSN application domains.”

As previously presented in **Figure 1**, study selection occurs on two occasions: after performing the search strategy (with papers’ metadata reading) and during data extraction (with papers’ full-text reading). This strategy significantly reduces the number of non-relevant papers to the SLM.

After the automatic search process, we identified and removed 66 duplicate papers (from the 155 studies group) with the support of the *Parsif.al* tool (available at `http://parsif.al`). Next, we read the title, summary, and keywords of each of the 89, upon which we applied EC and IC and eliminated 20 papers (see **Table 3**). As a result, we selected 69 ”probably relevant” studies since this selection only relies on the reading and interpretation of papers’ metadata.

[2]Search carried out on July 14, 2020.
[3]Search update carried out on July 31, 2021.

Next, the data extraction activity requires a form whose fields must be mapped to the research questions in the planning phase. These fields are filled in during the full-text reading of each paper. **Table 2** presents the mapping between form fields and research questions.

Table 2. Mapping between research questions and data extraction form fields.

Research question	Data extraction form field
RQ1	Knowledge area
	Application domain
	Case study
RQ2	Publication vehicle
	Publication year
RQ3	Adaptation level
	Abstraction type
RQ4	Proposal
	Experimental validation
RQ5	Limitations
	Future work

The data extraction activity eliminated 29 papers more, totalising 49 excluded papers. As described in **Table 3**, the EC7 criterion excluded most. It means that only full-text reading allowed us to eliminate papers not focusing on adaptation and macroprogramming in IoT or WSN.

As a result of the data extraction activity, 40 papers are relevant considering the SLM goal. Thus, the automatic search found only two new studies in comparison with our previous work, which identified thirty eight.

2.4 Snowballing

Besides automatic search, our search strategy includes forward snowballing (FSB) (Wohlin *et al.*, 2012) as an attempt to obtain other relevant studies using the forty studies group as input. In this SLM, the citation list of each paper was retrieved from the *Scopus* search engine.

In the first round of FSB[4], we identified 535 papers, from which 43 were duplicates considering the one-hundred-fifty-five initial studies group. The set of EC rejected 485 studies after metadata and full-text reading.

As seven papers remained, a second-round of FSB was performed[5]. Forty-five studies cited these seven papers. However, nine of them were duplicates, and EC rejected the remaining. As no new paper was identified, the snowballing procedure ended up identifying 580 studies, but only seven (from the first-round) relevant to this SLM[6].

Table 3. The number of studies excluded by exclusion criteria.

Activity	EC 1	2	3	4	5	6	7	8	Total
Automatic search	4	8	0	0	0	5	3	0	20
Data extraction	0	6	0	2	0	0	20	1	29
Snowballing	54	28	1	2	0	7	425	4	521
Total	58	42	1	4	0	12	448	5	570

[4]First-round carried out on August 5, 2021.
[5]Second-round carried out on September 6, 2021.
[6]Online documentation of the forward snowballing procedure is available at `https://bit.ly/328QCE6`.

Therefore, besides the 38 studies found in the original version of this SLM, this updated version retrieved *nine new relevant studies*: two by the automatic search and seven by the FSB procedure. The full list containing the forty-seven relevant papers is in **Table 4**. From now on, we identify them as S1 to S47 (S for study).

By analyzing the source of these relevant studies, we concluded that 98% of them came from *Scopus*. In other words, *Scopus* indexes most of the publication venues whose papers investigate abstraction, macroprogramming, and adaptation in infrastructures in IoT or WSN. Further information about these is also available elsewhere[7].

Finally, **Figure 2** depicts the entire selection process with the respective number of primary studies chosen and removed in each activity of the conduction phase. Besides, data extracted from each relevant study is also available[8].

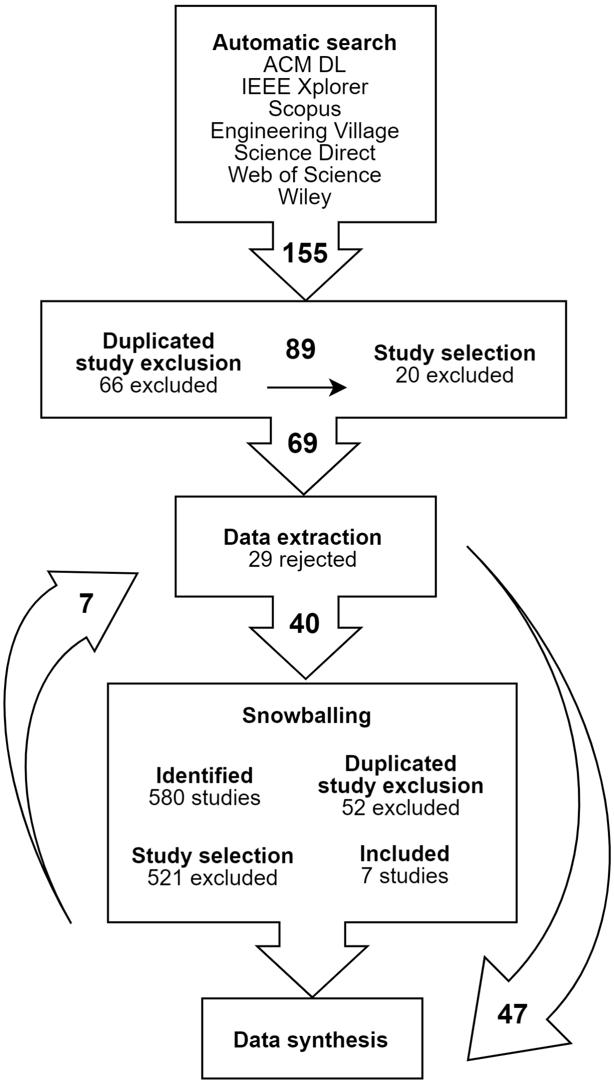

Figure 2. A detailed view of the conduction phase: automatic search, study selection, data extraction, snowballing, and data synthesis.

[7]Online documentation of the forty-seven relevant papers is available at `https://bit.ly/3kKDF9N`.

[8]Online documentation with data extracted of the accepted papers is available at `https://bit.ly/3nqwcOX`.

3 Results and Discussion

This section presents the analysis and synthesis of data extracted from the 47 studies to answer the SLM's research questions.

3.1 About research question 1

To answer RQ1, *"What are the application domains found in primary studies?"*, we found out that 72% of the papers (34 out of 47) focused exclusively on WSN, being the area with the highest number of publications. The remaining papers' subjects are IoT (7) or both IoT and WSN (6).

Figure 3 presents the distribution of papers per publication year. In 2015, the first IoT-oriented papers came out, and the number of such papers has increased since then. This IoT research's growth is confirmed by the literature (Greer *et al.*, 2019). As WSN is one of the IoT enabling technologies, it may explain the decreasing number of macroprogramming research in WSN favoring IoT.

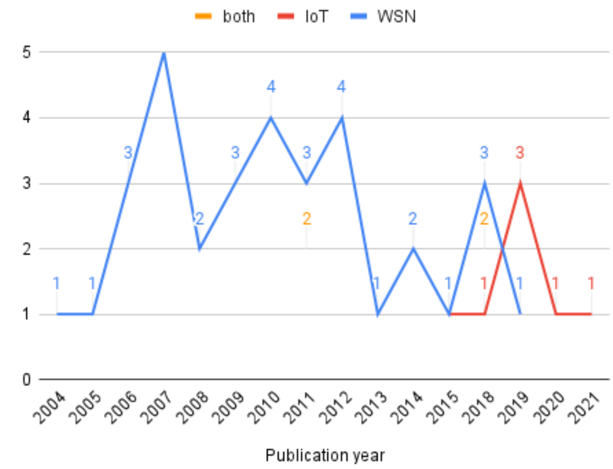

Figure 3. Knowledge area per publication year.

As depicted in **Figure 4**, almost 80% of the studies (37 of 47) investigated macroprogramming concepts and practices in real-world case studies. In total, those studies cover sixteen application domains, such as intelligent environments and monitoring. However, no case study was reported in papers published in 2004 and 2015. Besides, we represented in the *Others* category those papers whose application domain was not explicit.

The smart application domain seems to be a trend since 2018, including smart homes, buildings, grids, and transportation. Moreover, all these scenarios may converge to smart cities, representing a more complex picture for adopting macroprogramming abstractions.

To summarise, the answer to RQ1 is roughly the same as this SLM's previous version (Santana *et al.*, 2021): most of the macroprogramming studies in WSN with an increasing focus shift to IoT since 2015, and a diversity of application domains with an apparent inclination to smart environments in the last years.

Table 4. The forty-seven papers analysed on this SLM.

ID	Paper	Reference
S1	A component-based approach for service distribution in Sensor Networks	(Taherkordi *et al*., 2010)
S2	A constraint programming approach for managing end-to-end requirements in sensor network macroprogramming	(Hassani Bijarbooneh *et al*., 2014)
S3	A library for developing real-time and embedded applications in C	(Basanta-Val and García-Valls, 2015)
S4	A service-oriented approach to facilitate WSAN application development	(Cañete *et al*., 2011)
S5	A service-oriented middleware for wireless sensor and actor networks	(Cañete *et al*., 2009)
S6	A state-based programming model and system for wireless sensor networks	(Bischoff and Kortuem, 2007)
S7	Adaptive dynamic checkpointing for safe efficient intermittent computing	(Maeng and Lucia, 2018)
S8	Adaptive teams of autonomous aerial and ground Robots for situational awareness	(Hsieh *et al*., 2007)
S9	Adaptive Wireless Networks as an Example of Declarative Fractionated Systems	(Choi *et al*., 2014)
S10	An easy-to-use 3D visualization system for planning context-aware applications in smart buildings	(Su and Huang, 2014)
S11	An overview of the VigilNet architecture	(He *et al*., 2005)
S12	D'Artagnan: An Embedded DSL Framework for Distributed Embedded Systems	(Mizzi *et al*., 2018)
S13	Deductive Approach to Processing High-Level Video Activity Queries in UAV Networks	(Gupta, 2018)
S14	Defining Services and Service Orches-trators Acting on Shared Sensors and Actuators	(Bouali Baghli *et al*., 2018)
S15	Design and compilation of an object-oriented macroprogramming language for wireless sensor network	(Oppermann *et al*., 2014)
S16	Developing wireless sensor network applications based on a function block programming abstraction	(Kerasiotis *et al*., 2012)
S17	EcoCast: Interactive, object-oriented macroprogramming for networks of ultra-compact wireless sensor nodes	(Tu *et al*., 2011)
S18	Efficient configuration and control of SANETs using FACTS	(Terfloth and Schiller, 2008)
S19	Efficient routing from multiple sources to multiple sinks in wireless sensor networks	(Ciciriello *et al*., 2007)
S20	Energy-efficient task mapping for data-driven sensor network macroprogramming	(Pathak and Prasanna, 2010)
S21	Logical neighborhoods: A programming abstraction for wireless sensor networks	(Mottola and Picco, 2006)
S22	Macro programming a spatial computer with bayesian networks	(Mamei, 2011)
S23	MBMF: A framework for macroprogramming data-centric sensor network applications using the Bird-Meertens formalism	(Loke and Nadarajah, 2009)
S24	Nano-CF: A coordination framework for macro-programming in Wireless Sensor Networks	(Gupta *et al*., 2011)
S25	PICO-MP: Decentralised macro-programming for wireless sensor and actuator networks	(Dulay *et al*., 2018)
S26	A platform independent communications middleware for heterogeneous devices in smart grids	(Chen *et al*., 2019)
S27	ProFuN TG: Atool for programming and managing performance-aware sensor network application	(Elsts *et al*., 2015)
S28	Programming iMote networks made easy	(Bauderon *et al*., 2010)
S29	Intelligent IoT Systems with a Python-based Declarative Tool	(D'Urso *et al*., 2019)
S30	Programming the smart home	(Bischoff *et al*., 2007)
S31	PS-QUASAR: A publish/subscribe QoS aware middleware for Wireless Sensor and Actor Networks	(Chen *et al*., 2013)
S32	Region streams: Functional macroprogramming for sensor networks	(Newton and Welsh, 2004b)
S33	The omni macroprogramming environment for sensor networks	(Awan *et al*., 2006)
S34	TinyReef: A register-based virtual machine for wireless sensor networks	(Marques *et al*., 2009)
S35	Transactuations: Where transactions meet the physical world	(Sengupta *et al*., 2019)
S36	UBIQUEST, For Rapid Prototyping of Networking Applications	(Ahmad-Kassem *et al*., 2012)
S37	USEME: A service-oriented framework for wireless sensor and actor networks	(Cañete *et al*., 2008)
S38	µsETL: A set based programming abstraction for wireless sensor networks	(Hossain *et al*., 2011)
S39	A Service-Oriented Approach for Sensing in the Internet of Things: Intelligent Transportation Systems and Privacy Use Cases	(Hammoudeh *et al*., 2021b)
S40	ACAIOT: A Framework for Adaptable Context-Aware IoT applications	(ElKady *et al*., 2020a)
S41	A modular and extensible macroprogramming compiler	(Hnat *et al*., 2010)
S42	A Resource-Oriented Programming Framework Supporting Runtime Propagation of RESTful Resources	(Qiu *et al*., 2014)
S43	Enabling Scope-Based Interactions in Sensor Network Macroprogramming	(Mottola *et al*., 2007)
S44	Hybrid Macroprogramming Wireless Networks of Embedded Systems with Declarative Naming	(Intanagonwiwat, 2012)
S45	makeSense: Simplifying the Integration of Wireless Sensor Networks into Business	(Mottola *et al*., 2019)
S46	Role-based automatic programming framework for interworking a drone and wireless sensor networks	(Min *et al*., 2018)
S47	snBench: Programming and virtualization framework for distributed multitasking sensor networks	(Ocean *et al*., 2006)

3.2 About research question 2

To answer RQ2, *"When and where are primary studies published?"*, we observed that 25% of the studies (12 of 47) about macroprogramming in IoT/WSN were published from 2018. Following our protocol, there was no paper about that subject in 2016 and 2017.

Concerning publication venues, conferences and workshops cover 72% (34 of 47) of the accepted papers (see **Figure 5**). Besides, from 45 distinct publication venues, only two published two papers each: the *ACM/IEEE International Conference on Information Processing in Sensor Networks* and the *IEEE Conference on Local Computer Networks*.

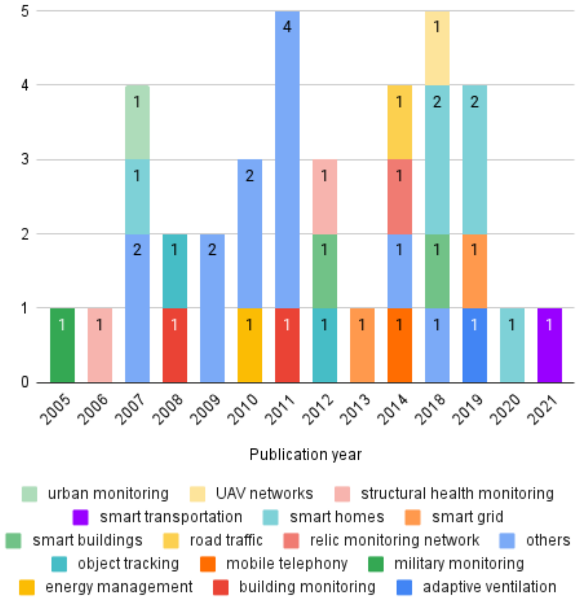

Figure 4. Application domain per publication year.

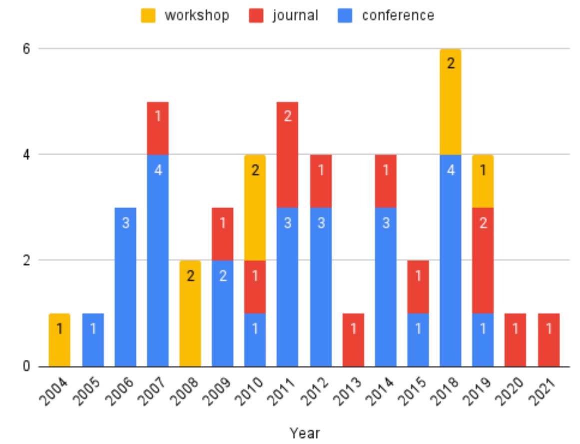

Figure 5. Publication venue per publication year.

In brief, the answer to RQ2 is similar to the one described in Santana *et al*. (2021): a significant number of studies (25%) about macroprogramming in IoT/WSN during the last four years and a heterogeneous collection of publications venues. These results suggest an increasing interest in research on macroprogramming for IoT/WSN in recent years. Besides, the community has a great list of options to publish their research on that subject.

3.3 About research question 3

The RQ3 investigates *"At what levels does adaptation occur, and what are abstraction types in the infrastructure?"*. Concerning adaptation level, we followed the Krupitzer's taxonomy that describes five levels of adaptation, as depicted in **Figure 6**: application (individual or a set of applications), software systems (middleware or operating system), communication (network infrastructure or communication patterns), context, and technical resource (Krupitzer *et al*., 2015).

The most investigated adaptation levels are, in this sequence, communication in the network infrastructure (25.5%), context (21.3%), and (ensemble of) applications (19.1%), represented by 12, 11, and 9 studies of the 47 accepted papers (see **Figure 7**). The most investigated adaptation levels are, in this sequence, communication in the network infrastructure (25.5%), context (21.3%), and (ensemble of) applications (19.1%), represented by 12, 11, and 9 studies of the 47 accepted papers.

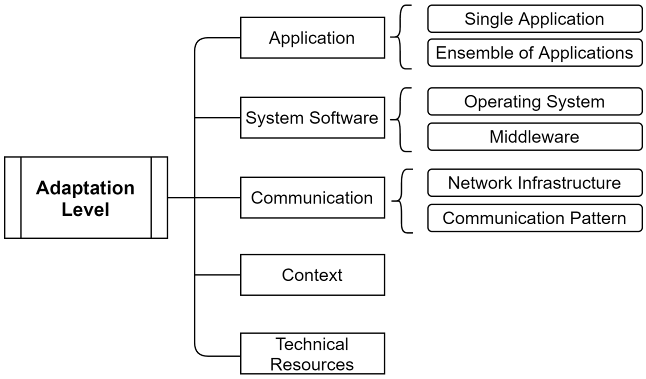

Figure 6. A taxonomy for adaptation level (Krupitzer *et al*., 2015).

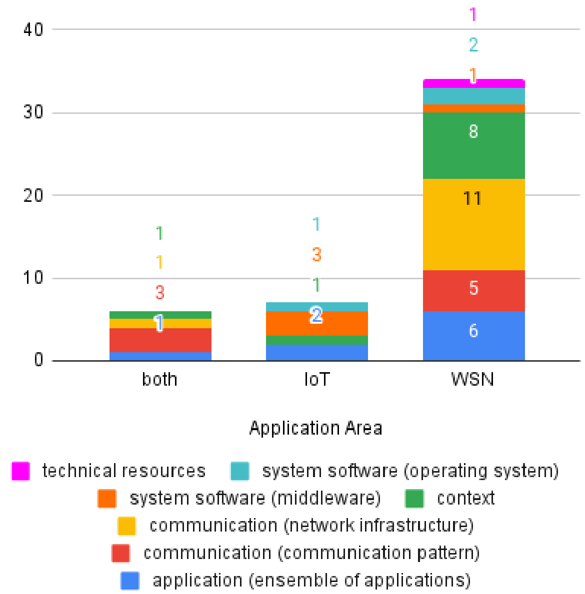

Figure 7. Adaptation levels and knowledge area.

Considering the adaptation level and the knowledge area of each study in **Figure 7**, a deeper analysis reveals that communication in the network infrastructure is studied most (11), followed by context (8) and communication pattern (6). Besides, adaptation is more frequent at the middleware (3) and the application (2) levels in IoT-oriented papers. It may be explained because middleware is helpful in situations with often resource-constrained IoT devices. Besides, there is no study examining adaptation at a single application level.

Regarding abstraction type, we used Motolla's work that classifies it as nodes, groups, and systems (Mottola, 2008). At the node level, macroprogramming abstractions alter individual nodes' states. At the group level, such modifications occur in a group of nodes. Finally, macroprogramming instructions spread over the network at the system level.

As shown in **Figure 8**, the group adaptation type is present in almost half of the studies (23 of 47) — we suppose the flexibility of subdividing a sensor network into smaller groups with common characteristics may explain this high percentage. Next, we crossed adaptation levels and abstraction types from the 47 accepted papers. Results reveal that context and communication in the network infrastructure are most present at the system and group levels, respectively. On the other hand, there is a more balanced distribution between adaptation levels at the node abstraction level.

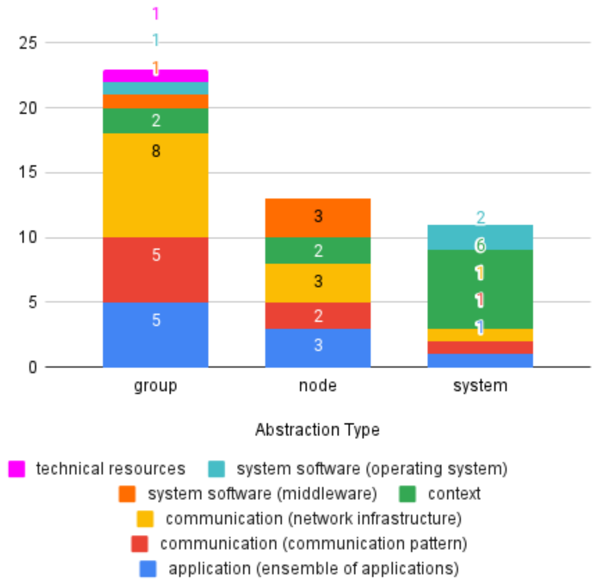

Figure 8. Adaptation level in relation to abstraction classification.

The answer to RQ3 somewhat differs from the one presented in Santana *et al.* (2021). Communication in the network infrastructure remains the most investigated adaptation level; the same applies to the studies examining groups of nodes as abstraction type. However, this SLM's update shows that the number of studies about nodes as abstraction succeeds the number of studies about the system abstraction.

3.4 About research question 4

To identify *"How are adaptations carried out in primary research on WSN and IoT?"*, we found out twelve different ways of implementing adaptation using macroprogramming in IoT/WSN, as depicted in **Figure 9**.

Software frameworks are present in more than one-third of the studies (17 of 47). Frameworks hide low-level details of designers' and programmers' tasks, automate part of these tasks, and ease software development. We believe these assumptions explain the high number of studies implementing adaptation demands in a software framework. Other implementations of the adaptation requirement include programming languages, middleware, and systems (six studies each).

We also classified the studies under the validation point of view: implementation, prototype, simulation, and testbed. Approximately two-thirds of the studies (30 of 47) validated their research proposals through implementation. On the other hand, simulations were performed in ten studies, and the testbed was the less frequent validation type (only 2 of 47).

As shown in **Figure 9**, implementation was also the most used validation type among the four most employed adaptation proposals (i.e., framework, programming languages, middleware, and system).

Thus, a software framework is the most frequent adaptation implementation, as also described in Santana *et al.* (2021). However, this SLM's update describes more studies employing programming language, middleware, and the system as adaptation implementations.

3.5 About research question 5

To answer *"What are the adaptability-related issues found?"* we identified problems, limitations, and future work proposals in each accepted paper. This SLM revealed 25 distinct issues: communication, network topology, network traffic, context-awareness, coordination, among others. Communication was the highest cited issue (4 of 47), even in IoT-oriented studies.

Observing the knowledge area (**Figure 10**), in IoT, the communication limitations were present in two papers. That represented the majority. In WSN, however, concentrated middleware and studies in development (two papers each).

One of the key results of our study is that very few research papers examine the opportunities for adaptation of a deployment guided by a high-level macroprogram. This is a key opportunity that we seek to exploit in our future research – building on the challenges in RQ5, we aim to develop formal approaches to continually guide a deployed system towards a more optimal form according to its current deployment environment conditions, while using a high-level macroprogram to ensure that the deployed system remains within an envelope of behaviour expected by the system designer.

Of these, 37 have different directions, and the others converge on the following themes, which show the target problems that researchers aim to solve using macroprogramming. This provides insight into challenges researchers view as being particularly suited to a macroprogramming-based solution. Overall, the dominant target problems across the study period are energy efficiency, aiming to extend the lifetime of deployed infrastructures, and scalability, given the large sizes typical of most deployments. We also note that scalability became the dominant target problem in the three latest years of the period comprised by our study. Besides energy efficiency and scalability, other target problems that have received significant interest include device location, collaboration, fault resilience, and time synchronization.

Finally, the answer to QP5 showed a large number of different types of limitations, as well as trends for future work, and it was not possible to identify any particular kind of trend. Similar to previous work (Santana *et al.*, 2021), communication had the most significant number of limitations, with 4 studies, most of them in IoT.

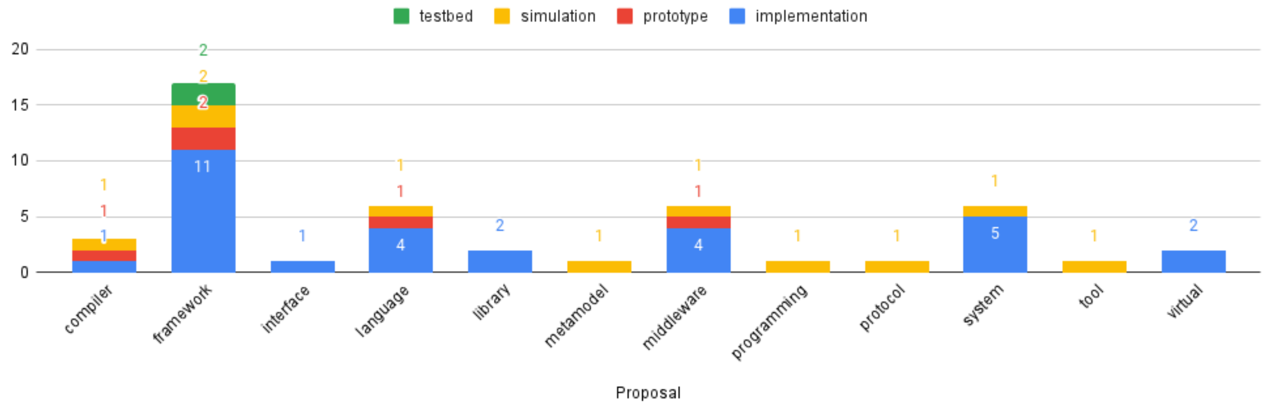

Figure 9. Adaptation implementations and validation types.

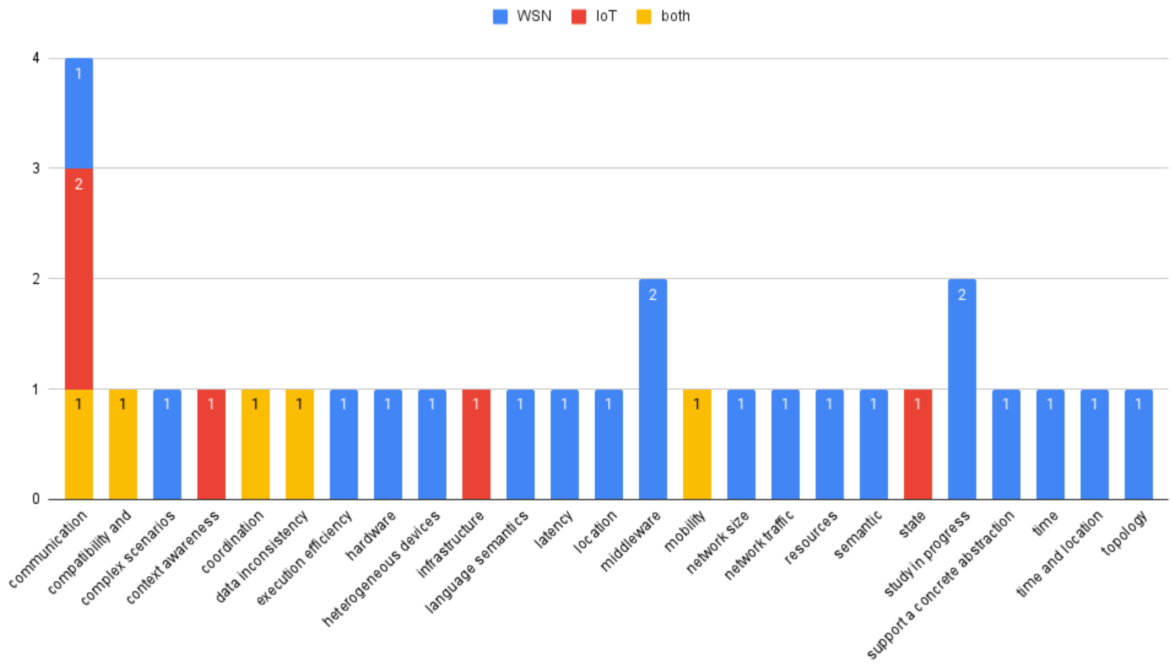

Figure 10. Limitations pointed out in relation to the application area.

3.6 Results synthesis

Figure 11 illustrates a bubble graph synthesizing the most relevant information we extracted and analysed from the accepted papers in this SLM. Three axes of information compose that bubble chart: adaptation level, abstraction type, and application area. The bubble size represents the number of studies that investigate the intersection of each two axes.

Notice the high concentration of macroprogramming research involving the communication adaptation level in WSN-oriented work. Besides, observe the number of studies in which modifications caused by macroprogramming abstractions disseminate in a group of nodes. Finally, there was no study in which adaptations take place in a single application. This finding confirms that macroprogramming should not be tackled at IoT/WSN isolated components.

4 Conclusions and future work

Overall, we posit that macroprogramming remains a topic of significant interest and a natural approach for IoT systems. Because these systems are often composed of a large number of devices controlled by a single organization, and because these devices are typically heterogeneous and relatively difficult to program in themselves, it is highly desirable to gain high-level abstractions to program the entire system.

We draw on the main results of our study to present a discussion of the challenges and opportunities for future research on macroprogramming for WSNs and IoT:

Converging on the right paradigms: our study revealed various macroprogramming paradigms for different applications and problems. For example, many existing programming abstractions for WSN and IoT provide a specification of actions performed by individual de-

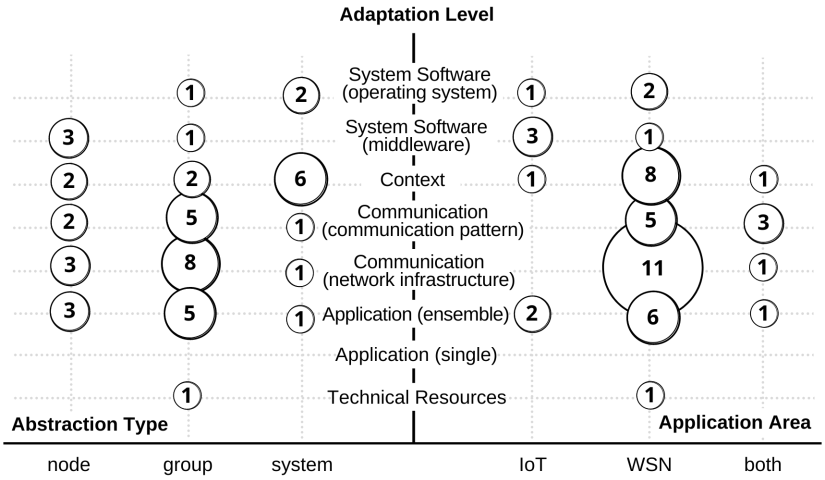

Figure 11. A bubble chart describing the mapping among adaptation level, abstraction type, and application area.

vices or instead allow one to program the network and customize the underlying run-time, which is often dynamic. However, we did not observe any notable convergence on accepted macroprogramming paradigms in general or for specific applications/challenges. Among the notable exceptions observed in the papers, ACAIOT (Adaptive Context-Aware IoT applications) was the proposed framework by comparing its architecture with recent research studies (ElKady *et al*., 2020b). Also, we use ACAIOT to implement smart home application services by using a real dataset.

Embracing the dynamic nature of the environment: the devices and services of an IoT deployment can change frequently and vary their availability at any given time. This can make it challenging for developers to define applications that seamlessly persist across this volatility to offer a continuous level of service. Macroprogramming appears to provide a straightforward solution to this problem, in that the overall business or scientific logic of an application can be defined separately from specific devices, with the deployment of a macroprogram able to adjust autonomously to the currently available resources.

Variable distribution of logic: as IoT deployments envision each device becoming a uniquely addressable Internet endpoint, and with the prevalence of cheap cloud computing, there is an inclination to use IoT nodes as non-intelligent data endpoints or actuation endpoints, with all business logic placed on cloud services which collect data from all nodes and make decisions based on that data. However, this architecture requires significant network capacity to get all data into the cloud and denies opportunities to perform at least some processing within the network. Macroprogramming offers a potential chance to automate the distribution of logic both within cloud services and within the IoT network itself, with automated macroprogram deployment tool chains able to decide which logic is best suited for which location based on available processing, network, and energy capabilities. One of the challenges relates to the degrees of freedom offered by macroprograms: as the system description is inherently high-level, the operationalisation of macroprograms has significant freedom in how they are deployed over time – including the placement of logic and the adaptation to fluctuations in the environment and resources. Let's take this opportunity to its extreme. We could envision a macroprogram acting as a specification of what the system is designed to do in an ideal scenario and an envelope of acceptable ways to implement that functionality. A smart deployment manager could then take that idealized specification and intelligently map it onto the available resources continuously, reporting how close the actual deployment is to the idealized specification of the microprogram.

This work contributes to the IoT and WSN fields, with the results of a systematic literature mapping (SLM). This SLM brings important work and reporting aspects as an alternative to propose the programming of devices at a high level, mainly with the growth of networks in the volume of data (high number of sensors) and device heterogeneity.

Finally, SLM aims to categorize the main findings of primary research about a topic and to benefit researchers in establishing baselines for other research activities. SLM is an open form of what the literature calls a systematic literature review (SLR), i.e., a deeper analysis and comparison of a collection of studies. As such, future work may consist of the conduction of an SLR on macroprogramming in IoT, focusing on those papers exploring multiple adaptation levels in groups of network nodes (see **Figure 11**). It is common practice to perform an SLR on pieces of evidence found in an SLM. Results of an SLR can be used to understand the efficacy and efficiency of a method or technology or the strengths and weaknesses of methods and technologies under certain circumstances. As study quality assessment is a widely deployed technique in SLMs and SLRs, we can also elaborate on a set of quality criteria to evaluate the different contributions of each of the 47 papers selected.

Acknowledgements

This work was partly funded by the Royal Society – Newton Mobility Grant NMG-R2-170105. This study was financed in part by the CAPES - Brazil. This research is also part of the INCT of the Future Internet for Smart Cities funded by CNPq proc.465446/2014-0, CAPES proc.88887.136422/2017-00, and FAPESP procs.14/50937-1 and 15/24485-9.

References

Ahmad-Kassem, A., Bobineau, C., Collet, C., Dublé, E., Grumbach, S., Ma, F., Martínez, L., and Ubéda, S. (2012). Ubiquest, for rapid prototyping of networking applications. In *Proceedings of the 16th International Database Engineering & Applications Sysmposium*, pages 187–192. DOI: 10.1145/2351476.2351498 .

Alajlan, A. M. and Elleithy, K. M. (2014). High-level abstractions in wireless sensor networks: Status, taxonomy, challenges, and future directions. In *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*. IEEE. DOI: 10.1109/asee-zone1.2014.6820645.

Awan, A., Sameh, A., and Grama, A. (2006). The omni macroprogramming environment for sensor networks. In *Proceedings of the 6th International Conference on Computational Science - Volume Part III*, ICCS'06, page 465–472, Berlin, Heidelberg. Springer-Verlag. DOI: 10.1007/11758532$_6$2.

Basanta-Val, P. and García-Valls, M. (2015). A library for developing real-time and embedded applications in c. *Journal of Systems Architecture*, 61(5):239–255. DOI: 10.1016/j.sysarc.2015.03.003.

Bauderon, M., Grumbach, S., Gu, D., Qi, X., Qu, W., Suo, K., and Zhang, Y. (2010). Programming imote networks made easy. In *2010 Fourth International Conference on Sensor Technologies and Applications*, pages 539–544. DOI: 10.1109/SENSORCOMM.2010.87.

Bischoff, U. and Kortuem, G. (2007). A state-based programming model and system for wireless sensor networks. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, pages 261–266. DOI: 10.1109/PERCOMW.2007.14.

Bischoff, U., Sundramoorthy, V., and Kortuem, G. (2007). Programming the smart home. In *2007 3rd IET International Conference on Intelligent Environments*. DOI: 10.1049/cp$_2$0070424.

Bouali Baghli, R., Najm, E., and Traverson, B. (2018). Defining services and service orchestrators acting on shared sensors and actuators. In *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development - MODELSWARD*, MODELSWARD 2018, page 237–246, Setubal, PRT. SCITEPRESS - Science and Technology Publications, Lda. DOI: 10.5220/0006609402370246.

Cañete, E., Chen, J., Díaz, M., Llopis, L., and Rubio, B. (2008). Useme: A service-oriented framework for wireless sensor and actor networks. In *2008 Eighth International Workshop on Applications and Services in Wireless Networks (aswn 2008)*, pages 47–53. DOI: 10.1109/ASWN.2008.5.

Cañete, E., Chen, J., Díaz, M., Llopis, L., and Rubio, B. (2009). A service-oriented middleware for wireless sensor and actor networks. In *2009 Sixth International Conference on Information Technology: New Generations*, pages 575–580. DOI: 10.1109/ITNG.2009.39.

Cañete, E., Chen, J., Díaz, M., Llopis, L., and Rubio, B. (2011). A service-oriented approach to facilitate wsan application development. *Ad Hoc Networks*, 9(3):430–452. DOI: 10.1016/j.adhoc.2010.08.022.

Chen, J., Cañete, E., Garrido, D., Díaz, M., and Piotrowski, K. (2019). Pico: A platform independent communications middleware for heterogeneous devices in smart grids. *Computer Standards & Interfaces*, 65:1–14. DOI: 10.1016/j.csi.2019.01.005.

Chen, J., Diaz, M., Rubio, B., and Troya, J. M. (2013). Ps-quasar: A publish/subscribe qos aware middleware for wireless sensor and actor networks. *Journal of Systems and Software*, 86(6):1650–1662.

Choi, J.-S., McCarthy, T., Kim, M., and Stehr, M.-O. (2014). Adaptive wireless networks as an example of declarative fractionated systems. In Stojmenovic, I., Cheng, Z., and Guo, S., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 549–563, Cham. Springer International Publishing. DOI: 10.1007/978-3-319-11569-6$_4$3.

Ciciriello, P., Mottola, L., and Picco, G. (2007). Efficient routing from multiple sources to multiple sinks in wireless sensor networks. In *European Conference on Wireless Sensor Networks*, volume 4373, pages 34–50. DOI: 10.1007/978-3-540-69830-2$_3$.

Dulay, N., Micheletti, M., Mostarda, L., and Piermarteri, A. (2018). Pico-mp: De-centralised macro-programming for wireless sensor and actuator networks. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 289–296. DOI: 10.1109/AINA.2018.00052.

D'Urso, F., Longo, C. F., and Santoro, C. (2019). Programming intelligent iot systems with a python-based declarative tool. Available at: `http://ceur-ws.org/Vol-2502/paper5.pdf`.

ElKady, M., Elkorany, A., and Allam, A. (2020a). Acaiot: A framework for adaptable context-aware iot applications. *International Journal of Intelligent Engineering and Systems*, 13:271–283. DOI: 10.22266/ijies2020.0831.24.

ElKady, M., Elkorany, A., and Allam, A. (2020b). Acaiot: A framework for adaptable context-aware iot applications. *International Journal of Intelligent Engineering and Systems*, 13:271–283. DOI: 10.22266/ijies2020.0831.24.

Elsts, A., Hassani Bijarbooneh, F., Jacobsson, M., and Sagonas, K. (2015). Profun tg: A tool for programming and managing performance-aware sensor network applications. In *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, pages 751–759. DOI: 10.1109/LCNW.2015.7365924.

Greer, C., Burns, M., Wollman, D., and Griffor, E. (2019). Cyber-physical systems and internet of things. Technical report, National Institute of Standards and Technology.

Gummadi, R., Gnawali, O., and Govindan, R. (2005). Macroprogramming wireless sensor networks using kairos. In *Distributed Computing in Sensor Systems*, pages 126–140. Springer Berlin Heidelberg. DOI: $10.1007/11502593_12$.

Gupta, H. (2018). Deductive approach to processing high-level video activity queries in uav networks. In *2018 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, pages 1–5. DOI: 10.1109/SECONW.2018.8396347.

Gupta, V., Kim, J., Pandya, A., Lakshmanan, K., Rajkumar, R., and Tovar, E. (2011). Nano-cf: A coordination framework for macro-programming in wireless sensor networks. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 467–475. DOI: 10.1109/SAHCN.2011.5984931.

Hammoudeh, M., Epiphaniou, G., Belguith, S., Unal, D., Adebisi, B., Baker, T., Kayes, A. S. M., and Watters, P. (2021a). A service-oriented approach for sensing in the internet of things: Intelligent transportation systems and privacy use cases. *IEEE Sensors Journal*, 21(14):15753–15761. DOI: 10.1109/jsen.2020.2981558.

Hammoudeh, M., Epiphaniou, G., Belguith, S., Unal, D., Adebisi, B., Baker, T., Kayes, A. S. M., and Watters, P. (2021b). A service-oriented approach for sensing in the internet of things: Intelligent transportation systems and privacy use cases. *IEEE Sensors Journal*, 21(14):15753–15761. DOI: 10.1109/JSEN.2020.2981558.

Hassani Bijarbooneh, F., Pathak, A., Pearson, J., Issarny, V., and Jonsson, B. (2014). A constraint programming approach for managing end-to-end requirements in sensor network macroprogramming. Available at: `https://www.scitepress.org/Papers/2014/47152/47152.pdf`.

He, T., Luo, L., Yan, T., Gu, L., Cao, Q., Zhou, G., Stoleru, R., Vicaire, P., Cao, Q., Stankovic, J., Son, S., and Abdelzaher, T. (2005). An overview of the vigilnet architecture. In *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)*, pages 109–114. DOI: 10.1109/RTCSA.2005.27.

Hnat, T. W., Sookoor, T. I., Hooimeijer, P., Weimer, W., and Whitehouse, K. (2010). A modular and extensible macroprogramming compiler. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications*, SESENA '10, page 49–54, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/1809111.1809126.

Hossain, M. S., Islam, A. B. M. A. A., Kulkarni, M., and Raghunathan, V. (2011). µsetl: A set based programming abstraction for wireless sensor networks. Available at: `https://ieeexplore.ieee.org/abstract/document/5779051?casa_token=z4nmskHo8A0AAAAA:6Q850VtoyDKczBDxJir8Sv9AeH_t2Xxbth3yYCD5ZZ5Hhi_Fbpmq1E30dZWSunoB3jhjll_egsU`.

Hsieh, M. A., Cowley, A., Keller, J. F., Chaimowicz, L., Grocholsky, B., Kumar, V., Taylor, C. J., Endo, Y., Arkin, R. C., Jung, B., Wolf, D. F., Sukhatme, G. S., and MacKenzie, D. C. (2007). Adaptive teams of autonomous aerial and ground robots for situational awareness: Field reports. *J. Field Robot.*, 24(11–12):991–1014. DOI: 10.1002/rob.20222 .

Intanagonwiwat, C. (2012). Hybrid macroprogramming wireless networks of embedded systems with declarative naming. *Int. J. Distributed Sens. Networks*, 8. DOI: 10.1155/2012/490826.

Kephart, J. and Chess, D. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50. DOI: 10.1109/mc.2003.1160055.

Kerasiotis, F., Koulamas, C., and Papadopoulos, G. (2012). Developing wireless sensor network applications based on a function block programming abstraction. In *2012 IEEE International Conference on Industrial Technology*, pages 372–377. DOI: 10.1109/ICIT.2012.6209966.

Krupitzer, C., Roth, F. M., VanSyckel, S., Schiele, G., and Becker, C. (2015). A survey on engineering approaches for self-adaptive systems. *Pervasive and Mobile Computing*, 17:184–206. DOI: 10.1016/j.pmcj.2014.09.009.

Loke, S. W. and Nadarajah, S. (2009). Mbmf: A framework for macroprogramming data-centric sensor network applications using the bird-meertens formalism. In *2009 Joint Conferences on Pervasive Computing (JCPC)*, pages 359–364. DOI: 10.1109/JCPC.2009.5420159.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2005). TinyDB: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1):122–173. DOI: 10.1145/1061318.1061322.

Maeng, K. and Lucia, B. (2018). Adaptive dynamic checkpointing for safe efficient intermittent computing. Available at: `https://www.usenix.org/system/files/osdi18-maeng.pdf`.

Mamei, M. (2011). Macro programming a spatial computer with bayesian networks. *ACM Trans. Auton. Adapt. Syst.*, 6(2). DOI: 10.1145/1968513.1968519.

Marques, I. L., Ronan, J., and Rosa, N. S. (2009). Tinyreef: a register-based virtual machine for wireless sensor networks. In *SENSORS, 2009 IEEE*, pages 1423–1426. DOI: 10.1109/ICSENS.2009.5398437.

Min, H., Jung, J., Kim, S., Kim, B., and Heo, J. (2018). Role-based automatic programming framework for interworking a drone and wireless sensor networks. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 1853–1856. DOI: 10.1145/3167132.3167432. .

Mizzi, A., Ellul, J., and Pace, G. (2018). D'artagnan: An embedded dsl framework for distributed embedded systems. In *Proceedings of the Real World Domain Specific Languages Workshop 2018*, RWDSL2018, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3183895.3183899.

Mizzi, A., Ellul, J., and Pace, G. J. (2019). Porthos: Macroprogramming blockchain systems. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE. DOI: 10.1109/ntms.2019.8763784.

Mottola, L. (2008). Programming wireless sensor networks: from physical to logical neighborhoods. Available at: `https://mottola.faculty.polimi.it/theses/mottola08programming.pdf`.

Mottola, L., Pathak, A., Bakshi, A., Prasanna, V. K., and Picco, G. P. (2007). Enabling scope-based interactions in sensor network macroprogramming. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–9. DOI: 10.1109/MOBHOC.2007.4428655.

Mottola, L. and Picco, G. P. (2006). Logical neighborhoods: A programming abstraction for wireless sensor networks. In Gibbons, P. B., Abdelzaher, T., Aspnes, J., and Rao, R., editors, *Distributed Computing in Sensor Systems*, pages 150–168, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI: $10.1007/11776178_10$.

Mottola, L. and Picco, G. P. (2011). Programming wireless sensor networks. *ACM Computing Surveys*, 43(3):1–51. DOI: 10.1145/1922649.1922656.

Mottola, L., Picco, G. P., Oppermann, F. J., Eriksson, J., Finne, N., Fuchs, H., Gaglione, A., Karnouskos, S., Montero, P. M., Oertel, N., Römer, K., Spiess, P., Tranquillini, S., and Voigt, T. (2019). makesense: Simplifying the integration of wireless sensor networks into business processes. *IEEE Transactions on Software Engineering*, 45:576–596. DOI: 10.1109/TSE.2017.2787585.

Newton, R., Morrisett, G., and Welsh, M. (2007). The regiment macroprogramming system. In *2007 6th International Symposium on Information Processing in Sensor Networks*. IEEE. DOI: 10.1109/ipsn.2007.4379709.

Newton, R. and Welsh, M. (2004a). Region streams. In *Proceeedings of the 1st international workshop on Data management for sensor networks in conjunction with VLDB 2004 - DMSN '04*. ACM Press. DOI: 10.1145/1052199.1052213.

Newton, R. and Welsh, M. (2004b). Region streams: Functional macroprogramming for sensor networks. In *Proceeedings of the 1st international workshop on Data management for sensor networks: in conjunction with VLDB 2004*, pages 78–87. DOI: 10.1145/1052199.1052213.

Noor, J., Sandha, S. S., Garcia, L., and Srivastava, M. (2019). DDF scpLOW/scp visualized declarative programming for heterogeneous IoT networks on heliot testbed platform. In *Proceedings of the International Conference on Internet of Things Design and Implementation*. ACM. DOI: 10.1145/3302505.3312598.

Ocean, M. J., Bestavros, A., and Kfoury, A. J. (2006). Snbench: Programming and virtualization framework for distributed multitasking sensor networks. In *VEE '06: Proceedings of the 2nd international conference on Virtual execution environments*, VEE '06, page 89–99, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/1134760.1134774.

Oppermann, F. J., Römer, K., Mottola, L., Picco, G. P., and Gaglione, A. (2014). Design and compilation of an object-oriented macroprogramming language for wireless sensor networks. In *39th Annual IEEE Conference on Local Computer Networks Workshops*, pages 574–582. DOI: 10.1109/LCNW.2014.6927705.

Pathak, A. and Prasanna, V. K. (2010). Energy-efficient task mapping for data-driven sensor network macroprogramming. *IEEE Transactions on Computers*, 59(7):955–968. DOI: 10.1109/TC.2009.168.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18. DOI: 10.1016/j.infsof.2015.03.007.

Qiao, Y., Nolani, R., Gill, S., Fang, G., and Lee, B. (2018). ThingNet: A micro-service based IoT macroprogramming platform over edges and cloud. In *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE. DOI: 10.1109/icin.2018.8401626.

Qiu, J., Li, D., Shi, H., Hou, C., and Cui, L. (2014). Easismp: A resource-oriented programming framework supporting runtime propagation of restful resources. *Journal of Computer Science and Technology*, 29:194–204. DOI: 10.1007/s11390-014-1422-0. .

Salehie, M. and Tahvildari, L. (2009). Self-adaptive software. *ACM Transactions on Autonomous and Adaptive Systems*, 4(2):1–42. DOI: 10.1145/1516533.1516538.

Santana, T. S., Sene Júnior, I. G., and Bulcão-Neto, R. F. (2021). Macroprogramming in the internet of things: A systematic mapping study. In *Anais do XIII Simposio Brasileiro de Computacao Ubiqua e Pervasiva (SBCUP 2021)*. Sociedade Brasileira de Computação. DOI: 10.5753/sbcup.2021.15999.

Sengupta, A., Leesatapornwongsa, T., Ardekani, M. S., and Stuardo, C. A. (2019). Transactuations: Where transactions meet the physical world. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 91–106, Renton, WA. USENIX Association. DOI: 10.1145/3380907.

Su, J.-M. and Huang, C.-F. (2014). An easy-to-use 3d visualization system for planning context-aware applications in smart buildings. *Comput. Stand. Interfaces*, 36(2):312–326. DOI: 10.1016/j.csi.2012.07.004.

Sugihara, R. and Gupta, R. K. (2008). Programming models for sensor networks. *ACM Transactions on Sensor Networks*, 4(2):1–29. DOI: 10.1145/1340771.1340774.

Taherkordi, A., Rouvoy, R., and Eliassen, F. (2010). A component-based approach for service distribution in sensor networks. In *MidSens '10: Proceedings of the 5th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks*, MidSens '10, page 22–28, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/1890784.1890789.

Terfloth, K. and Schiller, J. H. (2008). Efficient configuration and control of sanets using facts. In *HeterSanet '08*. DOI: 10.1145/1374699.1374702.

Tu, Y.-H., Li, Y.-C., Chien, T.-C., and Chou, P. H. (2011). Ecocast: Interactive, object-oriented macroprogramming for networks of ultra-compact wireless sensor nodes. Available at: `https://ieeexplore.ieee.org/abstract/document/5779052`.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-29044-2.