PAPER • OPEN ACCESS

Domain-Specific Languages for IoT: Challenges and Opportunities

To cite this article: Aymen J Salman *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1067** 012133

Domain-Specific Languages for IoT: Challenges and Opportunities

Aymen J Salman [1] Mohammed M. AL-Jawad [2] Wisam Al Tameemi [3]

1 College of Engineering, Al-Nahrain University, Baghdad, Iraq; aymen.j.salman@ced.nahrainuniv.edu.iq
2 College of Science, University of Kerbala, Karbala, Iraq; mohammedm@uokerbala.edu.iq
3 College of Engineering, University of Kerbala, Karbala, Iraq; wisam.i@uokerbala.edu.iq

Abstract. Internet of Things (IoT) is a multidisciplinary tool used by field experts to fabricate technology into their discipline; these fields experts thus need access to a simple, understandable and user-friendly programming language for this purpose. Domain-Specific Languages (DSL) have been developed to pave the way for such domain experts to integrate IoT capabilities into their everyday tasks. IoT is thus one of the main technologies targeting human day-to-day activities, and the concept of IoT has caused a general rethink and reimagining of the shape of life in the future. Many researchers are thus working on fields related to IoT such as networking, power consumption, data aggregation and fusion, finding ways to offer this new technology for end user access in a smooth and easy manner. Introducing new application development methods and using software engineering standards and tools that enable end-users to develop their own IoT applications is another aspect that has received attention in the last decade; however, this aspect requires more work from the research community. The main contribution of this paper is to highlight current efforts related to the IoT field and to perform quantitative analysis to highlight the current state of the art in both positive and negative terms, including the current limitations and obstacles faced by both end-users and field experts seeking to build their own IoT applications.

Keywords: Internet of Things, Application Development, Domain-Specific Language, Model Driven Development

1. Introduction

The Internet of Things (IoT) represents a great opportunity to add smart features to human life and to automate many activities to reduce the need for human intervention. There are many hardware platforms available in the market that can be used or integrated with IoT systems, yet to fully unleash IoT's potential capabilities, it is important to facilitate application development from an end-user perspective, which is a more challenging task [1] for many reasons, not least hardware heterogeneity and the multiple technical low-level details that developers must deal with. Developing such applications requires two main skills: knowledge of the technical hardware's low-level details and knowledge and expertise in low-level programming [2]. Accordingly, many researchers classify IoT developers into three main categories [3]: (1) Technology experts and scientists; (2) General application developers; and (3) Domain Experts or end-user.

The survey and review papers focused on this field are small in number and few clarify and highlight the research community efforts clearly [4], [5]. The main goal of the current paper is thus to focus on research community efforts from this perspective to facilitate quantitative analysis of such work using a systematic literature review in order to highlight the opportunities and challenges currently present in term of development, evaluation, architecture, and deployment of IoT Domain Specific Language (DSL).

To make an IoT application requires the developer to have prior knowledge of backend-based services as well as other areas such as user interfaces, server networking, and machine levels [6]. There is also a great deal of potential growth available for IoT among end-user industries and wearable devices [7] which means that additional sensor types and more development efforts are required. Finding tools to resolve these issues is a crucial task, and DSL is seen as an important tool for the development of IoT applications, based on the fact that most IoT users are not developers and that developers have focused on the growth of tailored IoT apps. Several different studies have referred to the challenges and opportunities DS offers to IoT without defining these or offering a holistic picture: [8] mentioned that DSL allows developers to define a new architecture without extensive knowledge of the hardware and software underlying IoT design of systems, while [9] noted that the heterogeneity of IoT devices is an obstacle for developers in terms of developing IoT applications and sharing information, and [10] presented a list of challenges and opportunities in a generic format with regard to developing IoT applications. Interoperability is one of the main challenges, due to communication protocols diversity [11], though using web-based solutions to provide cross platform capabilities [12] gives DSL the flexibility to facilitate multiple IoT devices.

2. Systematic Literature Review Protocol

The systematic literature review utilised in this paper sought to examine the most relevant research related to DSL in IoT. The work in [13] was thus used as a guideline for the review, and the next four sections illustrate the research questions applied, the search process, the selection and data extraction process, and the resulting analysis.

2.1. Research questions

The research questions were defined as follows:

RQ1: What are the most common HW and SW platforms in each domain?
RQ2: Where have DSLs been deployed?
RQ3: How are DSLs evaluated?
RQ4: Which real scenarios have applied successful DSLs?

2.2. Search string

The search keywords shown in Table 1 were used to address the research questions. Google Scholar was the primary source, with secondary databases utilised as in Table 1. Overall, 1,421 results were returned, as shown in Table 2.

Table 1. Keywords and strings

Digital Library	Search keyword
Google Scholar	("DSL" OR "Domain-Specific Language") AND ("IoT" OR "Internet of Things")
IEEE Xplore	
Elsevier (Scopus)	
Springer	
ACM Digital Library	

Table 2. Number of articles per database

Database	Number of Articles
Others	971
IEEE Xplore	200
Elsevier (Scopus)	118
Springer	79
ACM Digital Library	53

2.3. Study selection and assessment and data extraction

The research questions were used to formulate the search keywords, which produced 1,421 papers overall. An exclusion process was applied to survey and review papers, non-English papers, those not subject to peer-review procedures, and grey papers such as those without bibliographic information such as publication date/type or volume and issue numbers. Duplicate papers were also removed, with only the most complete, recent and improved version included. The remaining items were briefly examined to ensure they applied to the field of IoT application development, and thus had the potential to address at least one research question, and then a thorough review was undertaken of the potential options. The articles were thus processed as shown in Table 3, with only 23 articles out of 1,421 remaining.

Table 3. Articles excluding

Search Results	Papers	After Removing Redundant Versions	After Reviewing Titles and Abstracts	After Skimming Content	After Scanning Content
Elsevier (Scopus)	118	70	32	8	3
Taylor & Francis	6	4	3	2	0
Wiley	5	3	3	1	0
ACM Digital Library	53	33	17	10	5
Springer	79	50	25	11	3
IEEE Xplore	200	120	77	33	11
Others	960	709	302	99	1
Total	1421	989	459	164	23

2.4. Analysis and classification

After applying research methodology noted above, only 23 articles were considered. These articles were then categorised quantitively, as shown in Figures 1, 2, and 3 which display the distribution of the selected articles chronologically, by publishing type, and by publisher, respectively.

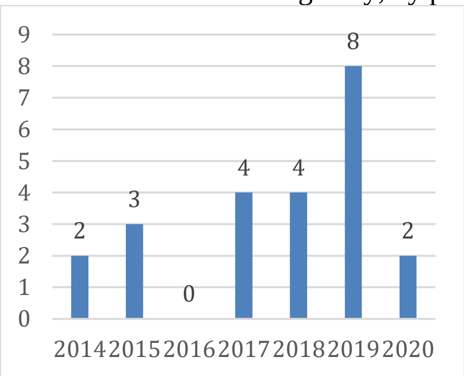

Figure 1. Distribution of Articles by Year

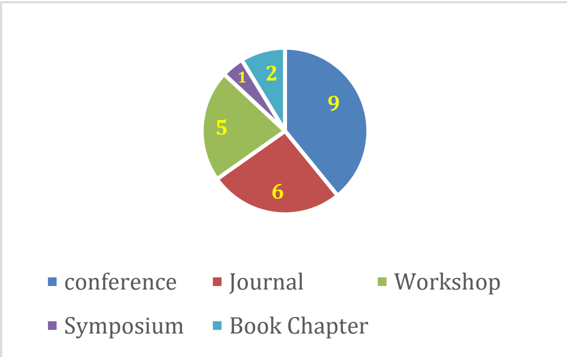

Figure 2. Article Type

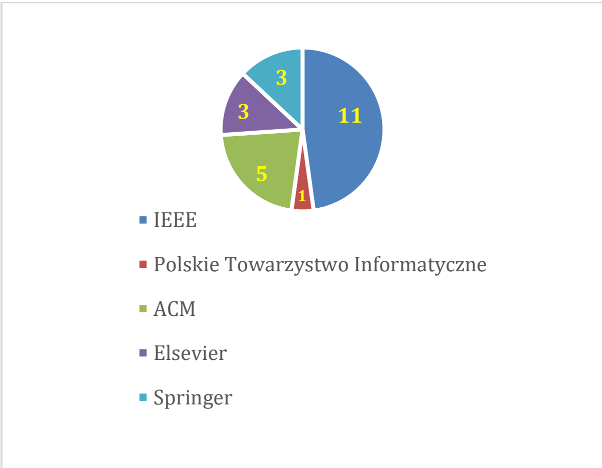

Figure 3. Article Publisher Type

To conduct a deeper analysis, a set of feature criteria related to Domain-Specific language properties was developed based on the DSL development and evaluation criteria (**DDEC**). All of the selected articles were thus examined based on these five criteria:

1. Development Method (**DM**): What strategy is used to define the solution? Generally, two methods can be used:
   - Customised Software (**CS**), including customised middleware or embedded operating systems, or
   - Model-Driven Development (**MDD**).
2. Language Type (**LT**): What type of the programming language is produced (Internal or External). According to [14], Internal DSL (**INT**) is any language hosted in and using a syntax that is part of a general-purpose programming language, while External DSL (**EXT**) is any language that uses a different or customised syntax from the programming language it works with.
3. Syntax Form (**SF**): many of the developed DSLs use graphical or visual interfaces to enable the end-user to model the required functionality without writing it in a textual form. This give three possible values: graphical modelling (**VISUAL**), textual (**TEXT**), or both (**MIX**).
4. Tools and Languages Used (**TLU**): What are the tools and languages used to develop the DSL?
5. DSL Output (**DO**): What is the output of the specified language? Generally, this is a specific type of generated files, such as Functional Files (**FF**), which can act as source code; executable files, which can be used to program the IoT nodes; or Analysis Files (**AF**), which can be used to for network/node modelling using any external or internal simulator.
6. Evaluation Method (**EM**): What evaluation methods are used to evaluate the DSL?

Table 4 illustrates the selected papers' classification based on the six noted. This highlights the variety of evaluation methods arising from a lack of real implementation and environment (domain).

Table 4. DSL Development and Evaluation Criteria (DDEC)*

Article	DM	LT	SF	TLU	DO	EM
EL4IoT [15]	MDD	INT	TEXT	Xtext and Xtend	FF	Proof of concept
VDSML [16]	N/A	INT	VISUAL	UML	N/A	Usability Test
Bilrost [17]	MDD	INT	TEXT	N/A	FF	N/A
ComPOS [18]	CS	INT	MIXED	PalCom middleware	FF	Simple Scenario
DSL-4-IoT [19]	CS	INT	VISUAL	OpenHAB & (JavaScript and Google Blockly)	FF	N/A
DoS-IL [20]	CS	INT	TEXT	Client/Server	FF	N/A
IRON [21]	CS	EXT	TEXT	Lua	N/A	N/A
Midgar [22]	MDD	INT	VISUAL	Java	FF	Experiment and Questionnaire
[12]	CS	INT	N/A	micro-services & Java	FF	N/A
SiMoNa [23]	MDD	INT	VISUAL	MetaEdit+ & GOPPRR	FF	Use Case
IOTCollab [24]	MDD	EXT	TEXT	Xtext and Java	FF	N/A
FRP [25]	CS	INT	TEXT	C	FF	Case Study
[26]	CS	INT	VISUAL	JavaScript	FF	Case Study Scenario
CISL [9]	CS	N/A	TEXT	N/A	FF	Experiment
MoMIT [27]	CS	INT	TEXT	JavaScript	FF	Experiment
[8]	MDD	EXT	VISUAL	N/A	FF	A case study scenario

SoaML4IoT [28]	MDD	INT	VISUAL	OCL	FF	N/A
[29]	MDD	EXT	N/A	Xtext and Xtend	AF	case study
TRILATERAL [30]	MDD	INT	VISUAL	Eclipse EMF	FF	case study
MUCSL [31]	MDD	INT	TEXT	Java	FF	Experiment and Questionnaire
TOP [32]	CS	EXT	TEXT	Lambda Expression	FF	N/A
D'Artagnan	CS	INT	TEXT	Haskell	FF	Use Case
[33]	CS	N/A	TEXT	JavaScript	N/A	N/A

* Some articles names use the proposed abbreviation of the paper or the DSL

To allow more extensive analysis the final articles were also scrutinised based on the IoT Architecture and Deployment Criteria (IADC), related to IoT Technology. These are

1. Modelling Domain (**MD**): generally, an IoT system consists of multiple nodes forming a network. Each node can be customised and programmed for a specific and customised function that differs from that of other nodes. Each IoT application must thus be developed by applying one of the following strategies:
   - Node Modelling Domain (**N**): the developer programs each node in the IoT network individually; or
   - Network Modelling Domain (**Net**): the developer can program the general network behaviour, and the DSL generates the required code files for each node automatically.
2. Target Platform (**TP**): what are the HW/SW platforms targeted to develop the DSL, and on which platform can the generated code be used?
3. Application Domain (**AD**): IoT applications can be used in many application types, such as health, agriculture, smart homes, or industrial applications; this should thus be specified.
4. Developer Types (**DT**): For which developer type was the DSL language designed? DSL users can be categorised into two main user types, End-User and IoT expert. An end-user is any person who will reap the benefits of the IoT application, and this thus includes domain experts such as mechanical and civil engineers, physicians, and biologists.

Table 5 shows the included papers highlighted based on the noted criteria. DSLs must be tailored to a specific domain; however, most of these proposals have targeted a general domain. Moreover, a tendency to support IoT developer types over others has limited the number of DSLs dedicated to users.

Table 5. IoT Architecture and Deployment Criteria (IADC)*

Article	MD	TP	AD	DT
EL4IoT [15]	N	Contiki & generate C Files	General	General
VDSML [16]	N	N/A	Smart Home	END-USER
Bilrost [17]	N	Raspberry Pi or Arduino	General	END-USER
ComPOS [18]	N	Camera & Motion Sensor	N/A	END-USER
DSL-4-IoT [19]	Net	Java & Libelium Waspmote	Smart Home & E-Health	N/A
DoS-IL [20]	N	Arduino	General	IoT EXPERT
IRON [21]	N	N/A	General	IoT EXPERT
Midgar [22]	N	Arduino	General	END-USER
[12]	N	Arduino	General	N/A
SiMoNa [23]	Net	C++ (Qt Framework)	Smart Home	END-USER

IOTCollab [24]	N/A	Arduino	General	IoT EXPERT
FRP [25]	Net	C language	General	IoT EXPERT
[26]	N	JavaScript	Elderly and smart home	N/A
CISL [9]	N	N/A	General	IoT EXPERT
MoMIT [27]	N	Java & C	General	IoT EXPERT
[8]	N	Arduino and Raspberry Pi	General	END-USER
SoaML4IoT [28]	N	N/A	General	IoT EXPERT
[29]	N	N/A	General	IoT EXPERT
TRILATERAL [30]	N	N/A	General	IoT EXPERT
MUCSL [31]	N	Arduino and Raspberry Pi	General	END-USER
TOP [32]	N	Arduino C++ dialect	General	IoT EXPERT
D'Artagnan	Net	Contiki	General	IoT EXPERT
[33]	N	N/A	General	IoT EXPERT

* Some article names have used the proposed abbreviation of the paper or the DSL

The criteria for the development and evaluation criteria are shown in Fig. 4 and Table 4, and these illustrate the fact that customised software (CS) is the most common development model, being preferred to the MDD as a way to speed up DSL development process by using customised software that tightly couples with the specific hardware for a specific purpose; re-using DSL in another domain and changing the hardware type thus makes the development process more complex. Most of the produced DSL use internal languages, which makes the code unreadable for non-developers. The most commonly used syntax used is text syntax, despite using a visual form being more user friendly. Java, JavaScript, and Xtext are the most commonly used programming languages used to develop IoT DSLs, and about 95% of these DSLs are developed to produce functional files that include source code that can be directly deployed to IoT nodes in hardware. This is generally regarded as an advantage of DSL, making it easy to deploy. Case studies and special experiments were generally used to evaluate the target DSLs due to a lack of real use cases.

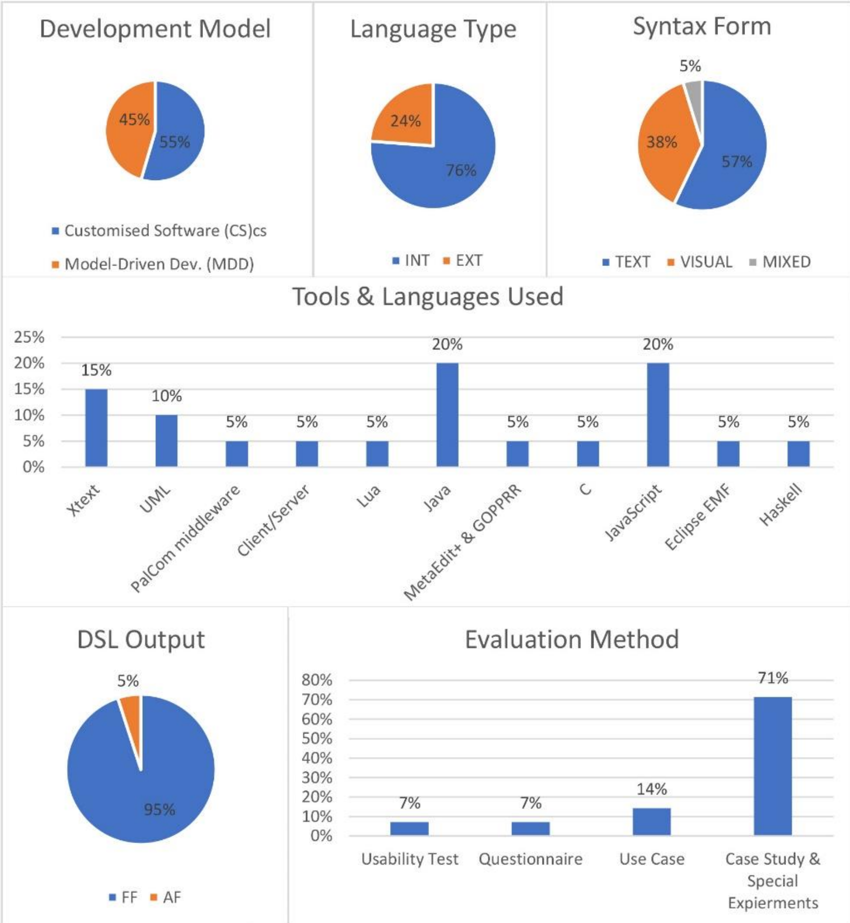

Figure 4. IoT Development and Evaluation Criteria Results

Based on the IoT architecture and deployment criteria, Figure 5 shows that most developed DSLs are used to model a single node in IoT, generally targeting Arduino microcontroller hardware, as this is one of the most common non-industrial hardware platforms. A minority of researchers used whole network modelling to examine user efforts in software development, though these were generally designed for general purposes, not targeting a specific domain, which makes their use of DSL more generic, despite it being intended to be tailored for a specific domain. Finally, the proposed DSLs are generally developed for use by IoT experts rather than domain (field) experts, making DSL use quite difficult for non-IT developers.

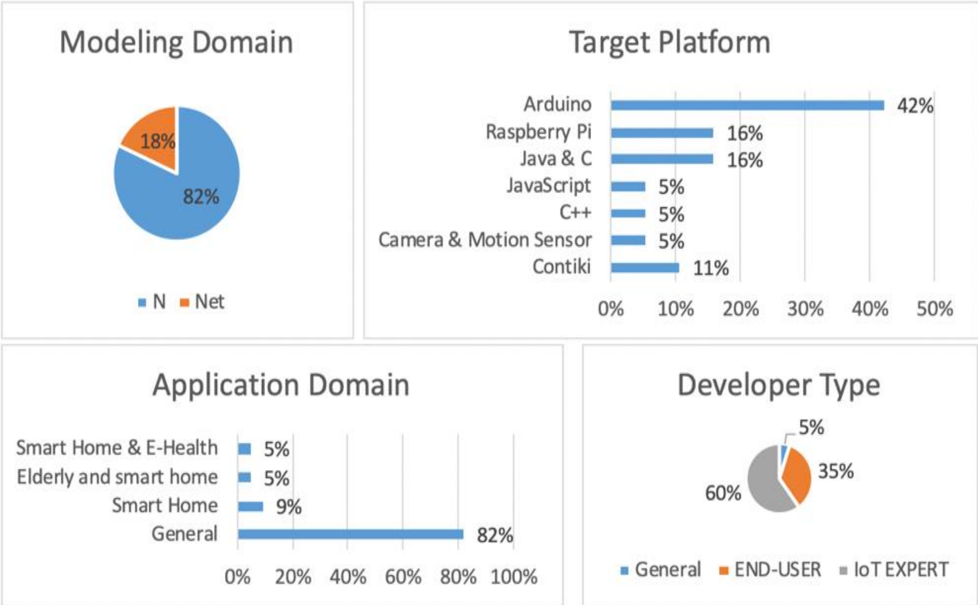

Figure 5. IoT Architecture and Deployment Criteria

3. Discussion

A systematic review was utilised to shed light on proposed DSLs within the research community. The most recent efforts towards developing Domain-Specific Languages dedicated to the Internet of Things (IoT) can be divided into two groups. The first refers to the architecture and deployment criteria, which are major factors in terms of bringing IoT technology to end-users and enabling them to develop more sophisticated applications without having to consider the low-level technical details. The second considers the development and evaluation criteria that aid the end user during the early stages of the development process. Both approaches have positive and negative aspects, as highlighted in It is .2.4 recommended that more efforts are made to produce domain specific languages using visual syntax to allow end-users to interact with DSL more easily in terms of targeting a specific application domain, thus offering more robust and customised functionality. DSLs should also be targeted at specific domains rather than generic approaches, with Network Modelling Domains used to make node extension easier for the end-user.

4. Conclusion

Based on reviewing the state of the art of the development and evaluation of IoT DSLs alongside their architecture and deployment shows that current proposed DSLs have many issues. These include a lack of visual interfaces, overemphasis on MDD at the expense of Customised Software, the excessive use of internal DSL, and reliance on experimental results rather than practical use cases, all of which affect DSL deployment and adaptation negatively. Using unreliable hardware platforms, targeting IoT experts rather than field users, using Node modelling and generalising DSLs also affect the usage and deployment of the proposed DSLs in a negative way. To overcome these issues, the use of Net modelling and tailored DSLs for specific domains are recommended, along with a drive for more reliable hardware and ways to offer solutions to non-IoT experts.

Developing opportunities to promote proposed DSLs depends on making them more user-friendly, particularly with regard to targeting end-users rather than IT experts; this would offer users an opportunity to be involved in the development stage and further enable plug and play concepts in semiautomated functions. However, the challenges of developing standardised software (operating system) and hardware remain significant.

References

[1] P. Nguyen *et al.*, “Advances in Deployment and Orchestration Approaches for IoT - A Systematic Review,” pp. 53–60, 2019, doi: 10.1109/iciot.2019.00021.
[2] A. J. Salman and A. Al-Yasiri, “SenNet: A Programming Toolkit to Develop Wireless Sensor Network Applications,” in *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Nov. 2016, pp. 1–7, doi: 10.1109/NTMS.2016.7792476.
[3] A. J. Salman and A. Al-Yasiri, “Developing Domain-Specific Language for Wireless Sensor Network application development,” in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, Dec. 2016, pp. 301–308, doi: 10.1109/ICITST.2016.7856718.
[4] D. Namiot and M. Sneps-Sneppe, “On internet of things programming models,” *Commun. Comput. Inf. Sci.*, vol. 678, pp. 13–24, 2016, doi: 10.1007/978-3-319-51917-3_2.
[5] A. Rayes and S. Salam, “The Role of Open Source in IoT,” *Internet Things From Hype to Real. Road to Digit.*, pp. 1–328, 2017, doi: 10.1007/978-3-319-44860-2.
[6] Z. Yin and B. Wei, “Apparatus, system and method to provide IoT cloud backend service.” Google Patents, 2019.
[7] H. F. Atlam, R. J. Walters, and G. B. Wills, “Internet of nano things: Security issues and applications,” in *Proceedings of the 2018 2nd International Conference on Cloud and Big Data Computing*, 2018, pp. 71–77.
[8] D. Alulema, J. Criado, and L. Iribarne, “A Model-Driven Approach for the Integration of Hardware Nodes in the IoT,” in *Advances in Intelligent Systems and Computing*, vol. 930, Springer, 2019, pp. 801–811.
[9] S. S. Cho, M. Song, and C. Julien, “Intelligent Information Sharing among IoT Devices: Models, Strategies, and Language,” in *IEEE International Conference on Electro Information Technology*, 2018, vol. 2018-May, no. August, pp. 107–112, doi: 10.1109/EIT.2018.8500128.
[10] I. S. Udoh and G. Kotonya, “Developing IoT applications: challenges and frameworks,” *IET Cyber-Physical Syst. Theory Appl.*, vol. 3, no. 2, pp. 65–72, 2018.
[11] B. Negash, T. Westerlund, A. M. Rahmani, P. Liljeberg, and H. Tenhunen, “DoS-IL: A Domain Specific Internet of Things Language for Resource Constrained Devices.,” in *ANT/SEIT*, 2017, pp. 416–423.
[12] M. Sneps-Sneppe and D. Namiot, “On web-based domain-specific language for Internet of Things,” in *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct. 2015, vol. 2016-Janua, pp. 287–292, doi: 10.1109/ICUMT.2015.7382444.
[13] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Inf. Softw. Technol.*, vol. 64, pp. 1–18, 2015.
[14] M. Fowler, *Domain-Specific Languages*, 1 Edition. Addison-Wesley Professional, 2010.
[15] T. Gomes *et al.*, “A modeling domain-specific language for IoT-enabled operating systems,” in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2017, vol. 2017-Janua, pp. 3945–3950, doi: 10.1109/IECON.2017.8216675.
[16] T. Eterovic, E. Kaljic, D. Donko, A. Salihbegovic, and S. Ribic, “An Internet of Things visual domain specific modeling language based on UML,” in *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*, Oct. 2015, pp. 1–5, doi: 10.1109/ICAT.2015.7340537.
[17] D. Meana-Llorián, C. G. García, J. P. Espada, M. Khari, and V. B. Semwal, “Bilrost: Connecting the Internet of Things through human Social Networks with a Domain-Specific Language,” in *Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering*, Jun. 2017, vol. 10, no. June, pp. 57–61, doi: 10.15439/2017R110.
[18] A. Akesson, G. Hedin, M. Nordahl, and B. Magnusson, “ComPOS: Composing Oblivious Services,” *2019 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2019*, pp. 132–138, 2019, doi: 10.1109/PERCOMW.2019.8730786.

[19] A. Salihbegovic, T. Eterovic, E. Kaljic, and S. Ribic, “Design of a domain specific language and IDE for Internet of things applications,” *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc.*, no. October, pp. 996–1001, 2015, doi: 10.1109/MIPRO.2015.7160420.

[20] B. Negash, T. Westerlund, P. Liljeberg, and H. Tenhunen, “Rethinking ‘Things’ - Fog Layer Interplay in IoT: A Mobile Code Approach,” in *Lecture Notes in Business Information Processing*, vol. 310, 2018, pp. 159–167.

[21] D. R. Cacciagrano and R. Culmone, “IRON: Reliable domain specific language for programming IoT devices,” *Internet of Things*, vol. 9, p. 100020, Mar. 2020, doi: 10.1016/j.iot.2018.09.006.

[22] C. González García, B. C. Pelayo G-Bustelo, J. Pascual Espada, and G. Cueva-Fernandez, “Midgar: Generation of heterogeneous objects interconnecting applications. A Domain Specific Language proposal for Internet of Things scenarios,” *Comput. Networks*, vol. 64, pp. 143–158, May 2014, doi: 10.1016/j.comnet.2014.02.010.

[23] C. Matos de Morais, D. Sadok, J. Kelner, and T. Lynn, “SiMoNa: A Proof-of-concept Domain-Specific Modeling Language for IoT Infographics,” in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019, no. April, doi: 10.24251/HICSS.2019.853.

[24] M. Adda and R. Saad, “A Data Sharing Strategy and a DSL for Service Discovery, Selection and Consumption for the IoT,” *Procedia Comput. Sci.*, vol. 37, pp. 92–100, 2014, doi: 10.1016/j.procs.2014.08.017.

[25] B. Calus, B. Reynders, D. Devriese, J. Noorman, and F. Piessens, “FRP IoT modules as a Scala DSL,” in *Proceedings of the 4th ACM SIGPLAN International Workshop on Reactive and Event-Based Languages and Systems - REBLS 2017*, 2017, pp. 15–20, doi: 10.1145/3141858.3141861.

[26] Y. Valsamakis and A. Savidis, “Personal applications in the internet of things through visual end-user programming,” *Digit. Mark. Unleashed*, pp. 809–821, 2017, doi: 10.1007/978-3-662-49275-8_71.

[27] R. Morales, R. Saborido, and Y.-G. Gueheneuc, “MoMIT: Porting a JavaScript Interpreter on a Quarter Coin,” *IEEE Trans. Softw. Eng.*, vol. X, no. X, pp. 1–1, 2020, doi: 10.1109/TSE.2020.2968061.

[28] B. Costa, P. F. Pires, and F. C. Delicato, “Modeling SOA-Based IoT Applications with SoaML4IoT,” in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Apr. 2019, pp. 496–501, doi: 10.1109/WF-IoT.2019.8767218.

[29] J. Verriet *et al.*, “Virtual Prototyping of Large-scale IoT Control Systems using Domain-specific Languages,” in *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development*, 2019, pp. 231–241, doi: 10.5220/0007250402310241.

[30] A. Iglesias, M. Iglesias-Urkia, B. López-Davalillo, S. Charramendieta, and A. Urbieta, “TRILATERAL: Software Product Line based Multidomain IoT Artifact Generation for Industrial CPS,” in *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development*, 2019, pp. 64–73, doi: 10.5220/0007343500640073.

[31] C. Gonzalez Garcia, L. Zhao, and V. Garcia-Diaz, “A User-Oriented Language for Specifying Interconnections Between Heterogeneous Objects in the Internet of Things,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3806–3819, Apr. 2019, doi: 10.1109/JIOT.2019.2891545.

[32] P. Koopman, M. Lubbers, and R. Plasmeijer, “A Task-Based DSL for Microcomputers,” in *Proceedings of the Real World Domain Specific Languages Workshop 2018 on - RWDSL2018*, 2018, pp. 1–11, doi: 10.1145/3183895.3183902.

[33] B. Fabian, A. Maurice, and B. Christian, “A Mapping Language for IoT Device Descriptions,” in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Jul. 2019, vol. 2, pp. 115–120, doi: 10.1109/COMPSAC.2019.10193.