FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Decentralized Orchestration of IoT in End-user Programming Environments

Ana Margarida Oliveira Pinheiro da Silva

WORKING VERSION

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Hugo Sereno Ferreira

Second Supervisor: André Restivo

January 10, 2020

Decentralized Orchestration of IoT in End-user Programming Environments

Ana Margarida Oliveira Pinheiro da Silva

Mestrado Integrado em Engenharia Informática e Computação

January 10, 2020

Abstract

The Internet-of-Things (IoT) is an ever growing network of devices connected to the Internet. Such devices are heterogeneous in their protocols and computation capabilities. With the rising computation and connectivity capabilities of these devices, the possibilities of their use in IoT systems increases. Concepts like smart cities are the pinnacle of the use of these systems, which involves a big amount of different devices in different conditions.

There are several tools for building IoT systems; some of these tools have different levels of expertise required and employ different architectures. One of the most popular is Node-RED [13]. It allows users to build systems using a visual data flow architecture, making it easy for a non-developer to use it.

However, most of these mainstream tools employ centralized methods of computation, where a main component — usually hosted in the cloud — executes most of the computation on data provided by edge devices, *e.g.* sensors and gateways. There are multiple consequences to this approach: (a) edge computation capabilities are being neglected, (b) it introduces a single point of failure, and (c) local data is being transferred across boundaries (private, technological, political...) either without need, or even in violation of legal constraints. Particularly, the principle of Local-First — *i.e.*, data and logic should reside locally, independent of third-party services faults and errors — is blatantly ignored.

Previous work attempt to mitigate some of these consequences, usually through tools that extend existing visual programming frameworks, such as Node-RED. They go as far as to propose a solution to decentralize flows and its execution in fog/edge devices. So far, achieving such decentralization requires that the decomposition and partitioning effort be manually specified by the developer when building the system.

Our goal is to extend Node-RED to allow automatic decomposition and partitioning of the system towards higher decentralization, by inferring computational boundaries. Furthermore, through automatic detection of abnormal run-time conditions, we also intend to provide dynamic self-adaptation. The prototype developed will be first validated with real devices and later with simulations.

As a result, we expect to achieve a more robust and efficient execution of IoT systems, by leveraging edge and fog computational capabilities present in the network, and improving overall reliability.

Keywords: Internet of Things, Visual Programming, Edge Computing

Resumo

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vehicula lorem commodo dui. Fusce mollis feugiat elit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu quam. Aenean consectetuer odio quis nisi. Fusce molestie metus sed neque. Praesent nulla. Donec quis urna. Pellentesque hendrerit vulputate nunc. Donec id eros et leo ullamcorper placerat. Curabitur aliquam tellus et diam.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum. Nulla lobortis purus pharetra turpis. Nulla laoreet, arcu nec hendrerit vulputate, tortor elit eleifend turpis, et aliquam leo metus in dolor. Praesent sed nulla. Mauris ac augue. Cras ac orci. Etiam sed urna eget nulla sodales venenatis. Donec faucibus ante eget dui. Nam magna. Suspendisse sollicitudin est et mi.

Fusce sed ipsum vel velit imperdiet dictum. Sed nisi purus, dapibus ut, iaculis ac, placerat id, purus. Integer aliquet elementum libero. Phasellus facilisis leo eget elit. Nullam nisi magna, ornare at, aliquet et, porta id, odio. Sed volutpat tellus consectetuer ligula. Phasellus turpis augue, malesuada et, placerat fringilla, ornare nec, eros. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus ornare quam nec sem mattis vulputate. Nullam porta, diam nec porta mollis, orci leo condimentum sapien, quis venenatis mi dolor a metus. Nullam mollis. Aenean metus massa, pellentesque sit amet, sagittis eget, tincidunt in, arcu. Vestibulum porta laoreet tortor. Nullam mollis elit nec justo. In nulla ligula, pellentesque sit amet, consequat sed, faucibus id, velit. Fusce purus. Quisque sagittis urna at quam. Ut eu lacus. Maecenas tortor nibh, ultricies nec, vestibulum varius, egestas id, sapien.

Phasellus ullamcorper justo id risus. Nunc in leo. Mauris auctor lectus vitae est lacinia egestas. Nulla faucibus erat sit amet lectus varius semper. Praesent ultrices vehicula orci. Nam at metus. Aenean eget lorem nec purus feugiat molestie. Phasellus fringilla nulla ac risus. Aliquam elementum aliquam velit. Aenean nunc odio, lobortis id, dictum et, rutrum ac, ipsum.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum. Nulla lobortis purus pharetra turpis. Nulla laoreet, arcu nec hendrerit vulputate, tortor elit eleifend turpis, et aliquam leo metus in dolor. Praesent sed nulla. Mauris ac augue. Cras ac orci. Etiam sed urna eget nulla sodales venenatis. Donec faucibus ante eget dui. Nam magna. Suspendisse sollicitudin est et mi.

Phasellus ullamcorper justo id risus. Nunc in leo. Mauris auctor lectus vitae est lacinia egestas. Nulla faucibus erat sit amet lectus varius semper. Praesent ultrices vehicula orci.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum.

Keywords: keyword1, Keyword2, keyword3

Acknowledgements

Aliquam id dui. Nulla facilisi. Nullam ligula nunc, viverra a, iaculis at, faucibus quis, sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur magna ligula, ornare luctus, aliquam non, aliquet at, tortor. Donec iaculis nulla sed eros. Sed felis. Nam lobortis libero. Pellentesque odio. Suspendisse potenti. Morbi imperdiet rhoncus magna. Morbi vestibulum interdum turpis. Pellentesque varius. Morbi nulla urna, euismod in, molestie ac, placerat in, orci.

Ut convallis. Suspendisse luctus pharetra sem. Sed sit amet mi in diam luctus suscipit. Nulla facilisi. Integer commodo, turpis et semper auctor, nisl ligula vestibulum erat, sed tempor lacus nibh at turpis. Quisque vestibulum pulvinar justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam sed tellus vel tortor hendrerit pulvinar. Phasellus eleifend, augue at mattis tincidunt, lorem lorem sodales arcu, id volutpat risus est id neque. Phasellus egestas ante. Nam porttitor justo sit amet urna. Suspendisse ligula nunc, mollis ac, elementum non, venenatis ut, mauris. Mauris augue risus, tempus scelerisque, rutrum quis, hendrerit at, nunc. Nulla posuere porta orci. Nulla dui.

Fusce gravida placerat sem. Aenean ipsum diam, pharetra vitae, ornare et, semper sit amet, nibh. Nam id tellus. Etiam ultrices. Praesent gravida. Aliquam nec sapien. Morbi sagittis vulputate dolor. Donec sapien lorem, laoreet egestas, pellentesque euismod, porta at, sapien. Integer vitae lacus id dui convallis blandit. Mauris non sem. Integer in velit eget lorem scelerisque vehicula. Etiam tincidunt turpis ac nunc. Pellentesque a justo. Mauris faucibus quam id eros. Cras pharetra. Fusce rutrum vulputate lorem. Cras pretium magna in nisl. Integer ornare dui non pede.

Author

"Until I began to learn to draw,
I was never much interested in looking at art."

Richard P. Feynman

Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
IoT	Internet of Things
VPL	Visual Programming Language
WWW	*World Wide Web*

Chapter 1

Introduction

This chapter introduces the motivation and scope of this project, as well as the problems it aims to solve. Section 1.1 details the context of this project in the area it is based on. Section 1.2 explains the reason why this work and the area it belongs to is important. Then, section 1.3 defines the problem we aim to solve and the goals of this dissertation are described in section 1.4. Finally, the section 1.5 describes the structure of this document and what content it contains.

1.1 Context

The Internet of Things paradigm states that all devices, independently of their capabilities, are connected to the Internet and allow for the transfer, integration and analytic of data generated by them [7]. This paradigm has several characteristics, such as the heterogeneity and high distribution of devices as well as their increasing connectivity and computational capabilities [4]. All this factors allow for a great level of applicability, enabling the realization of systems for management of cities, health services and industries [9].

The interest in Internet of Things has been growing massively, following the rising of connected devices along these past years. According to Siemens, in 2020 there will be around 26 billion physical devices connected to the Internet and in 2025 the predictions are pointing at 75 billion [3]. Although this allows for more opportunities, it is important to note that these devices are very different in their hardware and capabilities, which causes several problems in terms of development the systems, as well as their scalability, maintainability and security.

Visual Programming Languages (VPLs) allow the user to communicate with the system by using and arranging visual elements that can be translated into code [8]. It provides the user with an intuitive and straightforward interface for coding at the possible cost of loosing functionality. There are several programming languages with different focuses, such as education, video game development, 3D building, system design and even Internet of Things [**?**]. Node-RED[1] is one of

[1] https://nodered.org/

the most famous open source visual programming tool, originally developed by IBM's Emerging Technology Services team and now a part of the JS Foundation, which provides an environment for users to develop their own Internet of Things systems.

1.2 Motivation

Still needs something...more references and stuff

Internet of Things is a rapid growing concept that is being applied to several areas, such as home automation, industry, health, city management and many others. Given the number of existing systems with different protocols and architectures, it becomes difficult for a user to build a system that is in accordance to standards [2].

With the appearance of visual programming languages focused in IoT, more specifically Node-RED, users can build their own systems in an easier and streamlined way, removing the overhead of learning advanced programming concepts and protocols.

1.3 Problem Definition

Most mainstream visual programming tools focused on Internet of Things, Node-RED included, have a centralized approach, where a main component executes most of the computation on data provided by edge devices, e.g. sensors and gateways. There are several consequences to this approach: (a) computation capabilities of the edge devices are being ignored, (b) it introduces a single point of failure, and (c) local data is being transferred across boundaries (private, technological, political...) either without need, or even in violation of legal constraints. The principle of Local-First - i.e, data and logic should reside locally, independent of third-party services faults and errors - is being ignored.

Besides being a single point of failure, centralized systems can be less efficient than decentralized ones and in this context it might be the case, since there are computation capabilities that aren't being taken advantage of.

Chapter 4 expands on the problem definition, explaining it in bigger detail, defining its scope, use cases and research questions. ****CHECK THIS****

1.4 Goals

The main goal of this dissertation is to leverage the computation capabilities of the devices in the network, increasing efficiency, fault-tolerance, resiliency and scalability in an Internet of Things system.

To achieve this goal, a prototype will be developed, extending or rewriting Node-RED, that enables IoT devices to communicate their "computational capabilities" back to the orchestrator. In its turn, the orchestrator is able to partition the computation and send "tasks" to the nodes, which are the devices in the network, leveraging their computation power and independence.

As a secondary goal, several other challenges will be tackled, viz: (i) inferring computational capabilities of the devices in the network, (ii) detecting non-availability and using alternative computation resources, and (iii) exploring different alternatives of leveraging current IoT devices, including using firmwares that allow the execution of programs written in Lua, Javascript, Python, etc., amongst others.

1.5 Document Structure

Chapter 2 introduces the background information and explanation about concepts necessary for the full understanding of this dissertation with the use of a Systematic Literature Review on the state of the art of visual programming applied to the Internet of Things paradigm. Chapter 3 describes the state of the art regarding the ecosystem of this project's scope. Chapter 4 presents the problem this dissertation aims to solve, as well as the approach taken to solve it. Chapter 5 details how the solution was implemented and all the decisions and efforts taken to answer the problem statement mentioned before. Chapter 6 analyzes the evaluation process and explains how the solution was validated. Finally, Chapter 7 concludes the dissertation with a reflection on the success of the project by presenting the a summary of the contributions made and detailing the difficulties and future work.

Chapter 2

Background

This chapter describes the necessary foundations regarding visual programming tools for the Internet of Things context. Secion 2.1 describes the background of the Internet of Things paradigm and important concepts in that area. Section 2.2 mentions visual programming languages, their uses as well as their benefits and drawbacks. **TODO**

2.1 Internet of Things

Internet of Things paradigm was defined by the committee of the International Organization for Standardization and the International Electrotechnical Commission [1] as:

> *"An infrastructure of interconnected objects, people, systems and information resources together with intelligent services to allow them to process information of the physical and the virtual world and react."*

This paradigm is built upon the network of heterogenous devices interconnected between themselves, people and the environment. According to Buuya [10], the applications of IoT systems can be divided into four categories: (i) Home at the scale of an individual or home, (ii) Enterprise at the scale of a community, (iii) Utilities at a national or regional scale and (iv) Mobile, which is spread across domains due to its large scale in connectivity ans scale.

However, one might think that IoT only relates to machines and interactions between them. Most of the devices we use in our day-to-day - mobile phones, security cameras, watches, coffee machines - are now computation capable of making moderately complex tasks and are constantly generating and sending information, some of it to their users. This relates to the *human-in-the-loop* concept, where humans and machines have a symbiotic relationship [14].

2.1.1 IoT architectures

Internet of Things systems deal with big amounts of data from different sources and has to process it in efficient and fast ways. Typical IoT systems use a Cloud Computing architecture, where it takes advantage of centralized computing and storage. This approach has several benefits, such as increased computational capabilities and storage, as well as an easier maintenance. However, it comes with several problems such as (a) high latency and (b) high use of bandwidth, due to the need to send data from sensors to the centralized unit [11]. These problems originated solutions like Edge and Fog Computing.

Fog Computing brings computing closer to the perception layer, which contains the sensors. Due to this proximity and their high distribution, Fog Computing allows for several benefits like location-awareness, which can be useful for mobility and security requirements and low latency. These features are critical for Fog Computing and it is what differentiates it from Cloud Computing. One example of its use is in a system with several gateways that allow access to a back-end service. These gateways can be visualized as a network that besides passing information between users and the back-end, can also process and store data if necessary [16].

Edge Computing is a distributed architecture that uses the devices computational power to process the data they collect or generate. Its goal is to minimize the bandwidth and time response of IoT systems while leveraging the computational power of the devices in them. This paradigm is useful in overcoming common cloud computing problems, reducing latency and lessening the bandwidth bottleneck caused by the transfer of huge amounts of data [12]. Despite the vantages of Fog and Edge Computing, these approaches don't replace Cloud Computing. However, they can complement the high availability and great processing capacity and storage of the Cloud.

2.2 Visual Programming Languages

Visual Programming, as defined by Shu, consists of using meaningful graphical representations in the process of programming [18]. With this definition, we can consider Visual Programming Languages (VPLs) as a way of handling visual information and interaction with it, allowing the use of visual expressions for programming. According to Burnet and Baker [6], visual programming languages are constructed in order *"improve the programmer's ability to express program logic and to understand how the program works"*.

There are several applications of visual programming languages in different areas, such as education, video game development, automation, multimedia, data warehousing, system management and simulation, with this last area being the area with most use cases [17].

Visual programming languages have several characteristics, such as a concrete process and depiction of the program, immediate visual feedback and requires the knowledge of fewer programming concepts (e.g. pointers, memory allocation, etc) [6].

VPLs were categorized by Downes [5] based on their visual paradigms and architecture:

- **Purely Visual Languages**, where the creation is made using only graphical elements and the subsequently debugging and execution is made in the same environment.
- **Hybrid text and visual systems**, where the programs are created using graphical elements but their executions is translated into text language.
- **Programming-by-example systems**, where a user uses graphical elements to teach the system.
- **Constraint-oriented systems**, where the user translates physical entities into virtual objects and applies constraints to them, in order to simulate their behavior in reality.
- **Form-based systems**, which were based in the architecture and behavior of spreadsheets.

The categories mentioned can be present in a single system, making them not mutually exclusive.

2.3 Summary

This chapter introduces two areas that are fundamental for the understanding of this dissertation. Internet of Things is defined, as well as its use cases and categories. Fog and Edge computing paradigms are explained, which will be mentioned throughout this document. Finally, a definition and categorization of visual programming languages is introduced and explained.

Chapter 3

State of the Art

TEMP

This chapter describes the state of the art in visual programming tools in Internet of Things context, as well as decentralized methods of work distribution in flow-based architectures. Section 3.1 presents a systematic literature review on the topic of visual programming tools applied to the Internet of Things paradigm, which aims to answer the research questions defined in section 3.1.1.1. Section 3.1.2 ...

3.1 Systematic Literature Review

A Systematic Literature Review was made to gather information on the state of the art of visual programming applied to the Internet of Things paradigm. The goal of a systematic literature review is to synthesize evidence with emphasis on the quality of the it [15].

3.1.1 Methodology

During this Systematic Literature Review, a specific methodology was followed to reduce bias and produce the best results [15]. We started by defining the research questions to be answered as well as choosing data sources to search for publications.

3.1.1.1 Research Questions

REVIEW

In this Systematic Literature Review we intent to answer the following questions:

RQ1: How did Visual Programming Languages and Internet of Things evolve over time? Internet of Things is a paradigm with several years, but in the last few years it has been increasing in its applications, specifically with its integration with visual programming tools and environments. It is important to analyze the evolution of these concepts and their integration, to be able to compare with the state of the art.

RQ2: Methodologies implemented in Internet of Things with Visual Programming Languages? With the integration of visual programming tools with Internet of Things, several methodologies were implemented for it to be possible and provide users with a better experience.

RQ3: What is the maintenance and resilience of a Visual Programming Language integrated with an IoT system? Visual programming tools provide users a easy way of programming, with the use of visual elements and relations between them. However, this approach has downsides, such as difficulty in constructing and maintaining complex systems and high level programming that undermines efficiency and resilience.

3.1.1.2 Databases

The publications retrieved during this research were retrieved from the following databases, which are considered good and reliable sources:

- IEEE
- ACM
- Scopus

3.1.1.3 Search Process

To obtain results from the databases chosen, a research question was written with the union of the keywords "visual programming", "node-red", "dataflow" and intersection with the keyword "Internet of Things".

```
((vpl OR visual programming OR visual-programming) OR (node-red OR node red OR
    nodered) OR (data-flow OR dataflow)) AND (IoT OR internet of things OR
    internet-of-things)
```

The search was performed in October of 2019 and the results produced are the ones present in the table 3.1.

Table 3.1: Systematic Literature Review search results per database

Database	Total Results	Extracted Results
IEEE	410	379
ACM	171,768	2021
Scopus	540	500

3.1.1.4 Inclusion Criteria

To be included in the results, all publications should respect the inclusion criteria. If one of the criteria were not checked, the publication would not be included in the results. The inclusion criteria are the following:

1. On the topic of visual programming in internet of things;
2. Includes sufficient explanation of the research findings;
3. Publication year in the range between 2008 and 2019.

3.1.1.5 Exclusion Criteria

In addition to the inclusion criteria, all publications were analyzed in their compliance to the exclusion criteria. If any publication failed to comply with at least one of the exclusion criteria, it would not be included in the results. The exclusion criteria are the following:

1. Has less than two (non-self) citations when more than five years old;
2. Presents just ideas, magazine publications, interviews or discussion papers;
3. Not in English.

3.1.1.6 Quality Assessment

In order to classify if a publication is relevant to the research field, 4 assessments were made in order to better facility the process. The quality assessments are the following:

Table 3.2: Parameters for measuring the quality of a publication

Quality Assessment Query	**Quality Indicator (0-2)**
Is the publication relevant to us?	BARELY-PARTIALLY-SATISFACTORILY
Does the publication include and define research objectives adequately?	NO-PARTIALLY-YES
Are limitations and challenges well defined?	NO-PARTIALLY-YES
Is the proposed contribution well described?	NO-PARTIALLY-YES

Each assessment was posed in the form of a questions, and to each question there were three possible answers, with a numeric value each. If a publication didn't address the assessment the value with be 0, if the assessments was partially addressed the value would be 1. If the assessment was successfully satisfied, the value would be 2. In the end, the sum of all the assessments would represent the quality of the publication.

3.1.1.7 Evaluation Process

The evaluation process of the publications followed six steps with specific purposes:

1. **Range:** Publications are evaluated on date range, between 2008 and 2019;
2. **Relevance:** Title and abstract are scanned for relevance regarding the defined research field;
3. **Inclusion:** Publications are assessed against inclusion and exclusion criteria. Any publications not meeting the full inclusion criteria are discarded as well as all publications failing to comply to any exclusion criteria;
4. **Specificity:** Reading the publication to verify if it relates closely enough to the defined research field;
5. **Data:** Selected publications are analyzed for data related to the research questions and contribution details;
6. **Publication quality:** Publications are assessed using quality criteria defined in Table 3.2.

3.1.2 Results

TODO
basicamente chapar aqui os resumos do twist, organizados por artigo e explicar o que cada um traz de novo

Talvez especificar mais nos artigos focados em decentralização e assim?

3.1.3 Analysis and Discussion

TODO

3.1.3.1 Result Analysis

TODO Organizar os artigos por categorias?

3.1.3.2 Research Questions

Responder às research questions com os resultados

3.1.4 Conclusions

TODO

3.2 Decentralized Architectures in Visual Programming Tools applied to the Internet of Things paradigm

Colocar aqui os resumos sobre os artigos que li sobre decentralização, mais específicos ao meu tema Ainda nao sei como organizar esta parte

3.3 Summary

TODO

Chapter 4

Problem Statement

This chapter describes the problem, as it can be seen in Section 4.1. In Section 4.2 it is presented the wanted features for the proposed solution and in Section 4.3 the scope of the project is defined. Section 4.4 details the use cases, followed by Section 3.1.1.1, which contains the research questions to be answered by this dissertation. The validation methodology is outlined in Section 4.6. Finally, this chapter is summarized by Section 4.7 with an overview of the topics mentioned before.

4.1 Current Issues

Chapter 3 contains several solutions that provide decentralized architecture in visual programming tools applied to the internet of things paradigm. However, some of this tools are specific to a certain paradigm, like Smart Cities or industry. Check this after SOTA We can define the problem in these issues:

1. **Discovery of computation capabilities**: the current work lacks the automatic discovery of the computational capabilities of the devices in the network. This information is normally manually introduced by the developer.
2. **Leveraging devices in the network**:
3. **Inferring computational capabilities**:
4. **Detecting non-availability**:

4.2 Desiderata

explain what a desiderata is

D1: Infer computational capabilities of devices connected

D2: Decomposition and partition of the computation

D3: Convert computational tasks into runnable code

D4: Provide self-adaptation of the system

4.3 Scope

4.4 Use Cases

4.5 Research Questions

4.6 Validation

4.7 Summary

Chapter 5

Solution

Nullam eleifend condimentum nibh. Integer leo nibh, consequat eget, mollis et, sagittis ac, felis. Duis viverra pede in pede. Phasellus molestie placerat leo. Praesent at tellus a augue congue molestie. Integer eu ante pellentesque, viverra orci vitae, facilisis risus. Nunc eget pulvinar orci.

Chapter 6

Evaluation

Nullam eleifend condimentum nibh. Integer leo nibh, consequat eget, mollis et, sagittis ac, felis. Duis viverra pede in pede. Phasellus molestie placerat leo. Praesent at tellus a augue congue molestie. Integer eu ante pellentesque, viverra orci vitae, facilisis risus. Nunc eget pulvinar orci.

6.1 Assessment

6.2 Research Questions

6.3 Conclusions

Chapter 7

Conclusions

Nullam eleifend condimentum nibh. Integer leo nibh, consequat eget, mollis et, sagittis ac, felis. Duis viverra pede in pede. Phasellus molestie placerat leo. Praesent at tellus a augue congue molestie. Integer eu ante pellentesque, viverra orci vitae, facilisis risus. Nunc eget pulvinar orci.

7.1 Difficulties

7.2 Contributions

7.3 Conclusions

7.4 Future Work

References

[1] ISO/IEC JTC 1. Internet of things (iot) - preliminary report. *ISO,Tech. Rep.*, 2014.

[2] S. A. Al-Qaseemi, H. A. Almulhim, M. F. Almulhim, and S. R. Chaudhry. Iot architecture challenges and issues: Lack of standardization. In *2016 Future Technologies Conference (FTC)*, pages 731–738, Dec 2016.

[3] Tanweer Alam. A reliable communication framework and its use in internet of things (iot). 3, 05 2018.

[4] Fahed Alkhabbas, Romina Spalazzese, and Paul Davidsson. Iot-based systems of systems. *Proceedings of the 2nd edition of Swedish Workshop on the Engineering of Systems of Systems (SWESOS 2016)*, 2016.

[5] Marat Boshernitsan and Michael Downes. Visual programming languages: A survey. 08 1998.

[6] M. M. Burnett, M. J. Baker, C. Bohus, P. Carlson, S. Yang, and P. Van Zee. Scaling up visual programming languages. *Computer*, 28(3):45–54, March 1995.

[7] Rajkumar Buyya and Amir Vahid Dastjerdi. *Internet of Things: Principles and Paradigms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2016.

[8] S K Chang. *Handbook of Software Engineering and Knowledge Engineering*. World Scientific Publishing Company, 2002.

[9] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things Journal*, 1(4):349–359, Aug 2014.

[10] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29, 07 2012.

[11] D. S. Linthicum. Connecting fog and cloud computing. *IEEE Cloud Computing*, 4(2):18–20, March 2017.

[12] C. Martín Fernández, M. Díaz Rodríguez, and B. Rubio Muñoz. An edge computing architecture in the internet of things. In *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*, pages 99–102, May 2018.

[13] Node-red. Flow-based programming for the Internet of Things. https://nodered.org/, 2017. [Online; accessed November 2019].

[14] D. S. Nunes, P. Zhang, and J. Sá Silva. A survey on human-in-the-loop applications towards an internet of all. *IEEE Communications Surveys Tutorials*, 17(2):944–965, Secondquarter 2015.

[15] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1 – 18, 2015.

[16] Amir M. Rahmani, Pasi Liljeberg, Jrgo-Sren Preden, and Axel Jantsch. *Fog Computing in the Internet of Things: Intelligence at the Edge*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[17] Partha Pratim Ray. A survey on visual programming languages in internet of things. *Scientific Programming*, 2017:1–6, 2017.

[18] N. C. Shu. Visual programming: Perspectives and approaches. *IBM Syst. J.*, 38(2–3):199–221, June 1999.