

# IoT-based Systems Actuation Conflicts Management Towards DevOps A Systematic Mapping Study

Stéphane Lavirotte<sup>1</sup> <sup>a</sup>, Gérald Rocher<sup>1</sup> <sup>b</sup>, Jean-Yves Tigli<sup>1</sup> and Thibaut Gonnin<sup>1</sup>

<sup>1</sup>Université Côte d'Azur, CNRS, Laboratoire I3S, 06903 Sophia-Antipolis, France

{stephane.lavirotte, jean-yves.tigli}@univ-cotedazur.fr; {gerald.rocher, thibaut.gonnin}@etu.univ-cotedazur.fr

**Keywords:** Internet of Things; Cyber-Physical Systems; Conflict Identification; Conflict Resolution; DevOps

**Abstract:** The Internet of Things (IoT) has long been understood as an infrastructure layer allowing to gather environmental data through sensors. However, it also provides means to physically interact with our living environments through actuators. To the extent that actuation effects are not without risks on safety and trustworthiness, providing the IoT infrastructure layer with merely sensors access control mechanisms is no longer sufficient. It is also required to prevent conflicting (and possibly unsafe) actuation effects to occur in the physical environment and deploy means to resolve them. In this paper, we consider actuation conflicts management as part of the DevOps approach, which aims to harmonize tools and objectives of actors involved in IoT-based systems life cycle from their design to their deployment. In this context, a systematic mapping study (SMS) is conducted to better understand the actuation conflicts management approaches and to what extent they could be integrated into the DevOps life cycle.


## 1 INTRODUCTION


Internet of Things (IoT) based systems generally follow a layered architecture: (1) a shared IoT infrastructure layer consisting of a set of connected and distributed resources (e.g., processing units, memory, sensors, actuators, etc.) likely to be embedded in everyday objects (chair, lamp, etc.) and/or things (room, building, etc.), (2) a top layer where applications are deployed and, (3) one or more intermediate layers managing communications and ensuring the overall coherency between applications and the shared infrastructure. Thereby, the literature refers to three-layer architecture, middleware architecture, service-based architecture, five-layer architecture etc. (Kumar and Mallick, 2018). The notion of coherence, here, has a strong connection to that of *conflict*, “[...] ...a context change that leads to a state of the environment which is considered inadmissible by the application or user” (Tuttlies et al., 2007).

Following these architectural schemes, IoT-based systems have long been limited to collecting field information from sensors; in this context, the problem the intermediate layers must solve is merely technological; it aims to provide the infrastructure layer

with a sensors access control mechanism (Cecchinel et al., 2014). The notion of coherence then refers to the problem of managing *direct conflicts* (i.e., concurrent accesses). Managing actuation conflicts is a much more challenging task. It is not only a question of managing direct conflicts but also *indirect conflicts* that arise from the *concurrent interactions* of actuators with a common *physical system* and that can lead to an undesirable evolution of some of its properties (Teixeira et al., 2011) (e.g., simultaneously heating and cooling a room). In this context, the problem the intermediate layers must solve, beyond being technological, lies in the semantic interpretation of the effects produced in the environment and their consequences for humans in terms of *safety* and *trustworthiness*. From that viewpoint, by making explicit the interactions with the physical environment through the IoT shared infrastructure layer, IoT-based Cyber-Physical Systems (CPS/IoT) (Shih et al., 2016; Damjanovic-Behrendt et al., 2018), an “*integration of computation with physical processes, intersection of the physical and the cyber*” (Lee and Seshia, 2016), are most likely to face with this problem.

Actuation conflicts management is a first class concern in the realm of trustworthy and safe IoT-based systems, justifying the efforts put by the Eu-

<sup>a</sup>  <https://orcid.org/0000-0002-3341-6577>

<sup>b</sup>  <https://orcid.org/0000-0002-3874-6276>

ropean Union on this topic (e.g. ENACT project<sup>1</sup>, Brain-IoT project<sup>2</sup> and SecureIoT<sup>3</sup>). However, to date, only few surveys, systematic mappings and literature reviews have been conducted on the actuation conflicts management problem in the context of IoT-based systems. In (Resendes et al., 2014), the authors conducted a Systematic Literature Review (SLR) on the conflict detection and resolution problem in Home and Building Automation Systems (HBAS). They propose a taxonomy that classifies conflicts according to four different dimensions: (1) *source*, (2) *inter-venients*, (3) *time of detection* and (4) *solvability*. Although it provides an overview of the research on conflicts detection and resolution, the study is restricted to HBAS and does not provide recent information on how far these topics are covered in research.

Actuation conflict management is crucial and calls for a methodological break in the development process of IoT-based systems. In this context, the DevOps approach, an agile and incremental development approach that aims to harmonize the practices of actors involved in all stages of a system life cycle (i.e. from development to deployment and maintenance), is promising.

The goal of this paper is to clarify the interest and scope of recent research on the management of actuation conflicts in the area of IoT-based systems. In particular, we are interested in analyzing actuation conflict management in the perspective of the DevOps approach. To this end, we rely on the systematic mapping approach, well established in evidence based medicine and dedicated to provide researchers with the ability to build a classification scheme and structure a field of interest from which specific research questions can be answered (Heinz, 2014; Snyder, 2019).

## 2 BACKGROUND

In this section, we provide an overview of Internet of Things, Cyber Physical Systems and DevOps approach underlying the scope of this study.

### 2.1 Internet of Things (IoT)

Thanks to constant innovations in electronics and communication technologies, the use of our surrounding physical entities (chair, lamp, houses, cities, etc.) is transcended. Being connected to the Internet, new

forms of interaction are emerging from these physical entities thanks to sensors and actuators, embodied in remotely accessible software services and resulting in the fusion of the cyber and physical dimensions of our environments. In this context, IoT is “*the infrastructure enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies*” (International Telecommunication Union, ITU).

From an architectural point of view, IoT-based systems are generally multi-layered. The Fig. 1 includes some examples from (Kumar and Mallick, 2018). Here, it is interesting to note the purpose given to the shared IoT infrastructure layer:

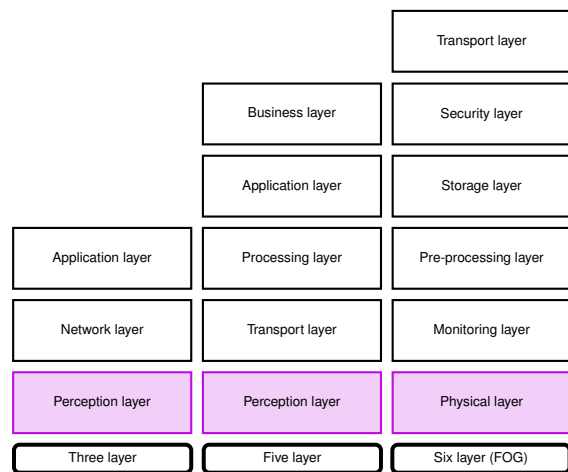


Figure 1: IoT architectures layers (Kumar and Mallick, 2018).

**Perception layer** – provides the ability to detect, collect and gather information about the physical environment and the connected objects within it.

**Environmental layer** – provides the ability to detect objects or places that are under observation. This includes the ability to observe moving physical entities, such as humans, cars, etc. and environmental properties such as temperature or humidity.

This semantics is somehow representative of the prevailing idea that consists in associating IoT to environmental sensing/monitoring capabilities. Although this vision has led to tremendous improvements in human well-being and assistance, optimization of resources, etc., it is important not to forget the action capabilities offered by actuators. Only recently has this capacity, and the associated risks to both humans and their environments, been seriously considered. This is witnessed by some on-going European projects. For instance, ENACT project acknowledges that “[...] IoT

<sup>1</sup><https://www.enact-project.eu>

<sup>2</sup><http://www.brain-iot.eu/>

<sup>3</sup><https://secureiot.eu>

system innovations have until now mainly been concerned with sensors, device management and connectivity, with the mission to gather data for processing and analysis in the cloud” and consider actuation as a first class concern in IoT “[...] The next generation IoT systems need to [...] **manage the closed loop from sensing to actuation**” (Ferry et al., 2018). BRAIN-IoT project specifically “*aims to support the integration into an IoT environment of devices and subsystems with actuation features that could possibly give rise to mixed-critically situations*” (Conzon et al., 2019).

## 2.2 Cyber-Physical Systems (CPS)

Cyber-physical systems (CPS) are a generalization of the concept of embedded systems to that of connected things with the objective of making them collaborate for the control of physical processes (Rajkumar et al., 2010). CPS find applications in the optimization of resources, their means of supply, etc. at the heart of the Industry4.0 and Industrial IoT (IIoT) revolutions. However, by controlling physical processes, these systems are not without risks for humans and their environment, as evidenced by the SecureIoT European project. This project aims to develop secure services targeting the areas of digital automation in manufacturing (Industry 4.0), social assistance robots for coaching and health and connected cars and autonomous driving. Such mechanisms are “*highly demanded by the industry in order to secure a whole new range of IoT applications that transcend the boundaries of multiple IoT platforms, while involving autonomous interactions between intelligent CPS systems and networks of smart objects*”.

## 2.3 DevOps approach

The DevOps (Sharma and Coyne, 2017) approach aims to harmonize the practices of software development (development, integration and testing) and systems administration (deployment, operation and maintenance) stakeholders (Fig. 2). This harmonization is justified by the conflicting objectives of these actors; on the one hand, software developers are constrained by cost and time, with the negative impacts that this can have on the quality of the software delivered. On the other hand, IT administration actors seek to achieve stability and quality objectives, at the expense of costs and deadlines.

This approach is based on agile and lean management methods and results in the collaboration of business managers, developers, operations and quality stakeholders in order to continuously deploy differ-

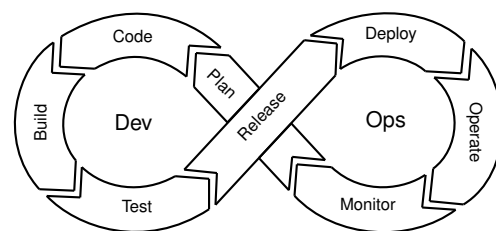


Figure 2: DevOps life cycle.

ent software versions. In this context, this approach aims to pool the tools for implementing software applications, from their design to their deployment. The underlying interest of this harmonization lies in the repeatability of the processes implemented throughout the DevOps loop (Fig. 2) and paves the way for their automation (Lwakatare et al., 2015). The latter is important in order to accelerate and maintain the convergence towards a system that meets the requirements of all actors involved. Automation is of particular importance in IoT-based systems where connected devices are constantly providing feedback. Automation then brings reactivity in updating these systems throughout the DevOps loop as soon as they need to be.

In this paper, we seek to understand to what extent the tools for identifying and resolving actuation conflicts can be integrated into IoT-based systems life cycle in the context of DevOps. To this end, we rely on the systematic mapping approach (Snyder, 2019) consisting in establishing research questions and organizing the answers obtained from the study of scientific publications. The methodology and results are exposed in the sequel.

## 3 RESEARCH METHOD

This systematic mapping study was developed following the guidelines proposed in (Petersen et al., 2015). On the basis of the context and the motivations presented in §1 and 2, we define the research questions (RQ) in §3.1. In order to define the scope of the study and reduce possible biases in the selection process, we explain the inclusion and exclusion criteria in §3.2.

### 3.1 Research questions

This study aims to answer the set of research questions described in Table 1. The questions relating to conflict management (RQ2) remain rather general, the objective being to understand the limitations of

the current actuation conflicts management methods towards their implementation within the DevOps approach.

<b>RQ1</b>	<b>What are the primary studies statistics?</b>
RQ1.1	What is the publication rate over years?
RQ1.2	In which types of venue (workshop, conference, journal) were the studies published?
RQ1.3	How studies are distributed in terms of academic and industrial affiliation and location?
RQ1.4	What application domains are concerned?
<b>RQ2</b>	<b>How actuation conflicts are managed?</b>
RQ2.1	What actuation conflicts are considered (direct/indirect)?
RQ2.2	At what stage of the IoT-based systems life cycle are they implemented?
RQ2.3	What is their automation level?
RQ2.4	What is their maturity level?

Table 1: Research questions

### 3.2 Search strategy

The study was conducted using three different databases: ACM-DL, IEEE Xplore and Scopus. During the selection process, the inclusion and exclusion criteria defined in tables 2 and 3 were applied.

<b>Inclusion criteria</b>
Primary peer-reviewed paper
The scope of the paper is fully related to the research questions
Paper written in English language
Publication year $\geq$ 2008

Table 2: Inclusion criteria

<b>Exclusion criteria</b>
White paper, technical report, thesis, book chapter, patent and presentation
The content of the paper is not appropriate to answer the research questions
Duplicate

Table 3: Exclusion criteria

The queries associated to each database are given in table 4. The scope of the search is restricted to papers related to IoT and CPS domains published in 2008 onward. 2008 was a pivotal year in the field of IoT. This is the year from which the number of IoT-related publications has started to increase significantly (Fig. 3). It was also the year in which the first international conference on IoT was held (Floerkemeier et al., 2008).

The second conjunctive part of the queries restricts the scope of the search to papers dealing with actuation, source of the direct and indirect conflicts.

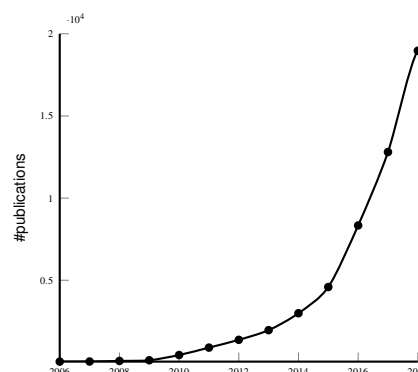


Figure 3: Evolution of publications related to the Internet of Things (IoT).

While the notion of actuation conflict has a strong meaning in the field of robotics (the physical side), in the software engineering domain (the cyber side), it is more about the one of *Feature Interaction* (FI) (Bruns, 2005), a feature “being a unit of functionality that can be developed and evolved independently” (Bocovich and Atlee, 2016). Therefore, it is worth taking into account these different semantic interpretations in the search, IoT-based systems developers having a strong software engineering culture. Finally, the third conjunctive part of the queries expresses the notion of conflict through a set of synonyms.

On the basis of the papers found using the search string, a backward snowballing technique was used (Wohlin, 2014) in order to identify additional relevant papers. By following this approach, three more papers were added. Taking the guidelines and applying the exclusion criteria, an extensive review of the selected papers was made by three researchers analyzing the title, the abstract and the content of each extracted paper. Consensus on keeping or rejecting papers was found during meetings conducted throughout the selection process. Finally, a total of 26 papers (Table 10) were selected as a result of this classification process while 2842 papers were excluded, as shown in Table 5.

## 4 Results highlights

The following sections are devoted to providing an analysis of the selected publications according to the research questions identified in §3.1.

### 4.1 Overview

This section provides answers to the research question RQ1 and associated sub-questions RQ1.1, RQ1.2, RQ1.3 and RQ1.4.

Library	Advanced /Command search query string	Added filters
ACM-DL	("internet of things" OR iot OR "cyber-physical system" OR cps) AND (actuation OR actuator OR feature) AND (conflict OR interaction OR interference OR shared OR concurrency)	Published 2008 onward
IEEE Xplore	("internet of things" OR iot OR "cyber-physical system" OR cps) AND (actuation OR actuator OR feature) AND (conflict OR interaction OR interference OR shared OR concurrency)	Year range: from 2008 to 2019 Conferences and Journals
Scopus	TITLE-ABS-KEY(("internet of things" OR iot OR "cyber-physical systems" OR cps) AND (actuation OR actuator OR feature) AND (conflict OR interaction OR interference OR shared OR concurrency))	Year: limit to "2008 to 2019" Subject area: limit to Computer Science Document type: limit to Conferences and Journals

Table 4: Search strings and filters for each database

Source	SearchResults	Removing duplicates	Reviewing titles	Reviewing abstract	Scanning content
ACM-DL	866	834	112	29	12
IEEE Xplore	970	943	92	28	9
Scopus	1032	505	40	16	5
<b>Total</b>	<b>2868</b>	<b>2282</b>	<b>244</b>	<b>73</b>	<b>26</b>

Table 5: The search and selection stages for primary studies

**Answering RQ1.1:** Until 2011, there were only few papers dealing with actuation conflict management in IoT-based systems as depicted in Fig. 4.

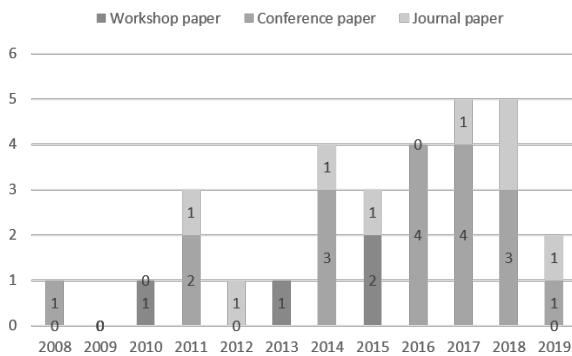


Figure 4: Primary studies per year

Based on the Fig.4, although limited, one can denote an increasing publication trend on the subject from 2011 onward, **demonstrating the growing interest of the research community to the IoT-based systems actuation conflict management problem.** We conducted our search process in November 2019 which can explain the low amount of papers published for this year.

**Answering RQ1.2:** Fig. 5 depicts the distribution of primary studies published over years and per venue type. There is a constant number of publications during the period 2014-2018. 24% are journal papers, 62% are conference papers and the remaining 14% are workshop papers.

**Answering RQ1.3:** by focusing on authors' affiliation, one can denote in Fig. 6 that most of the authors

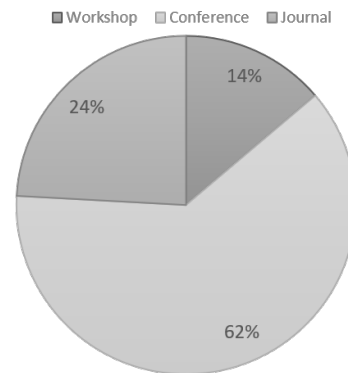


Figure 5: Primary studies per venue type

of the primary studies are researchers (83%).

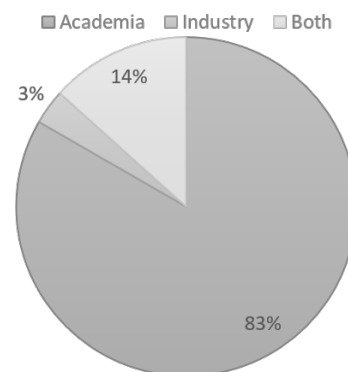


Figure 6: Primary studies per affiliation type

The involvement of industry in this research area is still limited (3%). There are also joint research between academic and industry (14%). This underline

the raising importance of actuation conflict management concern in industrial use cases, still at an academic research level. **These results tend to confirm that IoT-based systems actuation conflict management problem is still in its infancy.** The top most countries are United States of America (8), Austria (4), following by Canada, France and Japan (3). Researchers from other countries are also involved in this research for a total of 19 countries.

**Answering RQ1.4:** in terms of application areas, a short predominance can be observed for Smart Home relative use-cases (and, in the broad sense, smart-\* systems), ahead of the automotive use-cases. The applications specific to CPS as defined in §2.2 will be found under the terms Automotive, Smart Factory and Robotics. **Bottom-line, IoT-based systems actuation conflict management problem affects all socio-economical layers ranging from humans (Smart Health), houses (Smart Home) to cities (Smart City) and industry (Smart Factory).**

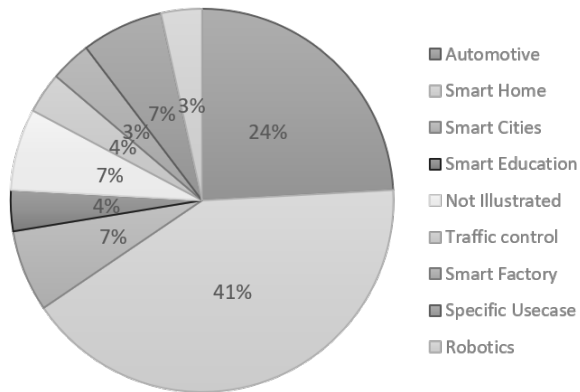


Figure 7: Primary studies per domains

## 4.2 Analysis towards the DevOps perspective

**Answering RQ2.1:** Direct and indirect conflicts are taken into account and are well balanced (Table 6). Papers on IoT deal with direct conflicts more than indirect conflicts. This trend is reversed for papers on CPS that are more focused on indirect conflicts. By making interactions with the physical environment explicit, it is expected to observe such a trend for CPS.

	IoT	CPS	Total
Direct	34.48%	13.79%	48.27%
Indirect	20.69%	31.03%	51.72%
<b>Total</b>	<b>55.17%</b>	<b>44.82%</b>	<b>100%</b>

Table 6: Actuation conflicts types considered in studies

**Answering RQ2.2 & RQ2.3:** The table 7 shows the Dev/Ops dichotomy in the identification of direct and indirect conflicts. Whether at design time (Devs) and operation (Ops), the analysis shows a well-balanced handling to the identification of direct and indirect conflicts.

	@design	@runtime	Total
Direct	27.58%	20.69%	48.27%
Indirect	24.14%	27.58%	51.72%
<b>Total</b>	<b>51.72%</b>	<b>48.27%</b>	<b>100%</b>

Table 7: Actuation conflicts identification

Concerning the resolution, the picture is not as good (Table 8). Conflict resolution is far from being addressed in all the papers, 20.69% of them discuss actuation conflicts identification without proposing a solution to resolve them. On the other hand, 51.71% of the papers assume manual actuation conflict resolution. In addition to the ad-hoc nature that this type of management implies, this raises also the question of the scalability of the proposed approaches. Indeed, the number of the possible combination of interactions and their effects increase exponentially with the number of interactions considered. Scalability is considered a first-class concern for only three papers (S15, S19, S22).

Besides scalability, it is also worth noting that automation is at the heart of the DevOps approach by allowing reactive and timely update of these systems throughout their life cycle, all the more important as IoT devices never stop providing feedback. Given the large number of distributed actuators (and sensors) likely to be involved, a non automated solution is likely to fail in insuring these timely updates. In this context, only 27% of the papers rely on a parameterized approach or synthesize a conflict manager to resolve them, the latter type of approach representing 10% of the papers. These latter approaches therefore seem to be the most relevant for addressing the scalability and automation issues and deserve further work.

	None	Manual	Parameterized	Synthesis
Direct	6.89%	31.03%	3.45%	6.89%
Indirect	13.80%	20.68%	13.80%	3.44%
<b>Total</b>	<b>20.69%</b>	<b>51.71%</b>	<b>17.25%</b>	<b>10.33%</b>

Table 8: Actuation conflicts resolution strategies

### Answering RQ2.4:

We can see here the maturity problem already mentioned in the research/industry dichotomy (Fig. 6). In particular, none of the proposed approaches

	Theoretical	Use-case	In silico	In vitro	In vivo
<b>Direct</b>	3.45%	6.89%	17.24%	20.70%	0.00%
<b>Indirect</b>	0.00%	13.80%	31.03%	6.89%	0.00%
<b>Total</b>	<b>3.45%</b>	<b>20.69%</b>	<b>48.27%</b>	<b>27.59%</b>	<b>0.00%</b>

Table 9: Actuation conflicts management maturity

for identifying and resolving actuation conflicts have been validated *in vivo*. This observation is aggravated for indirect conflicts whose validation does not go beyond the *in silico* stage. This confirms the difficulty of implementing the management of indirect actuation conflicts, and therefore the importance of evaluating them to reach the maturity required for a transfer.

## 5 Conclusion

This SMS study highlighted recent work on the actuation conflict management in the field of IoT-based systems. More specifically, we have been interested in studying their applicability into the DevOps approach. While direct/indirect conflicts identification methods applicability is well balanced from development to execution stages, most of the current resolution methodologies lacks important properties to pretend to their exploitation within the DevOps loop.

Indeed, most of them are not automated failing to scale and ensure timely and reactive systems updates throughout their life cycle (which is one of the main reason for promoting the DevOps approach). Furthermore, the maturity of the identification and resolution of actuation conflict does not go beyond *in vitro* and *in silico* stages respectively. No experimentation has been conducted *in-vivo*.

This study will need to be complemented by a Systematic Literature Review (SLR) to analyse the different methods used for conflict identification and resolution.

## 6 Acknowledgments

The research has received funding from the European Commission's H2020 Program under grant agreement numbers 780351 (ENACT).

## REFERENCES

- Bocovich, C. and Atlee, J. M. (2016). Feature-oriented modelling in bip: A case study. In *ModComp@ MoD-ELS*, pages 6–11.
- Bruns, G. (2005). Foundations for features. In *FIW*, pages 3–11. Citeseer.
- Cecchinell, C., Mosser, S., and Collet, P. (2014). Software Development Support for Shared Sensing Infrastructures: A Generative and Dynamic Approach. In Schaefer, I. and Stamelos, I., editors, *Software Reuse for Dynamic Systems in the Cloud and Beyond*, Lecture Notes in Computer Science, pages 221–236, Cham. Springer International Publishing.
- Conzon, D., Rashid, M. R. A., Tao, X., Soriano, A., Nicholson, R., and Ferrera, E. (2019). Brain-iot: Model-based framework for dependable sensing and actuation in intelligent decentralized iot systems. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, pages 1–8. IEEE.
- Damjanovic-Behrendt, V., Mühlberger, M., de Luca, C., Christos, T., and Arnautovic, E. (2018). Iot4cps—trustworthy iot for cps.
- Ferry, N., Solberg, A., Song, H., Lavirotte, S., Tigli, J.-Y., Winter, T., Muntés-Mulero, V., Metzger, A., Velasco, E. R., and Aguirre, A. C. (2018). Enact: Development, operation, and quality assurance of trustworthy smart iot systems. In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pages 112–127. Springer.
- Floerkemeier, C., Langheinrich, M., Fleisch, E., Mattern, F., and Sarma, S. E. (2008). *The Internet of Things: First International Conference, IOT 2008, Zurich, Switzerland, March 26-28, 2008, Proceedings*, volume 4952. springer.
- Heinz, M. (2014). Systematic mapping studies. *Mainz: Universität Koblenz-Landau*.
- Kumar, N. M. and Mallick, P. K. (2018). The internet of things: Insights into the building blocks, component interactions, and architecture layers. *Procedia computer science*, 132:109–117.
- Lee, E. A. and Seshia, S. A. (2016). *Introduction to embedded systems: A cyber-physical systems approach*. Mit Press.
- Lwakatere, L. E., Kuvaja, P., and Oivo, M. (2015). Dimensions of devops. In *International conference on agile software development*, pages 212–217. Springer.
- 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.
- Rajkumar, R., Lee, I., Sha, L., and Stankovic, J. (2010). Cyber-physical systems: the next computing revolution. In *Design Automation Conference*, pages 731–736. IEEE.
- Resendes, S., Carreira, P., and Santos, A. C. (2014). Conflict detection and resolution in home and building automation systems: a literature review. *Journal*

#	Title	Year	v	#	Title	Year	v
1	Formal Verification of Cyber-physical Feature Coordination with Minimalist Qualitative Models	2019	J	16	Towards a model-based verification methodology for Complex Swarm Systems	2016	C
2	IoT2: A formal method approach for detecting conflicts in large scale IoT systems	2019	C	17	An Application Conflict Detection and Resolution System for Smart Homes	2015	W
3	A spatially aware policy conflict resolution for information services	2018	J	18	Dependable control systems with Internet of Things	2015	J
4	Automata-Based Generic Model for Interoperating Context-Aware Ad-Hoc Devices in Internet of Things	2018	J	19	Safe Composition in Middleware for the Internet of Things	2015	W
5	Context Aware Virtual Assistant with Case-Based Conflict Resolution in Multi-User Smart Home Environment	2018	C	20	Coordination of ECA rules by verification and control	2014	C
6	IoTSan: Fortifying the safety of IoT systems	2018	C	21	DepSys: Dependency aware integration of cyber-physical systems for smart homes	2014	C
7	Taming and optimizing feature interaction in software-intensive automotive systems	2018	C	22	Distributed programming framework for fast iterative optimization in networked cyber-physical systems	2014	J
8	Continuous variable-specific resolutions of feature interactions	2017	C	23	ECA rules for IoT environment: A case study in safe design	2014	C
9	Event management for simultaneous actions in the Internet of Things	2017	C	24	Harnessing evolutionary computation to enable dynamically adaptive systems to manage uncertainty	2013	W
10	Modeling architectures of cyber-physical systems	2017	C	25	Component-oriented Interoperation of Real-time DEVS Engines	2011	C
11	Synchronization abstractions and separation of concerns as key aspects to the interoperability in IoT	2017	C	26	Programming support for distributed optimization and control in cyber-physical systems	2011	C
12	Towards Model Checking of Network Applications for IoT System Development	2017	J	27	Semantic Web-based policy interaction detection method with rules in smart home for detecting interactions among user policies	2011	J
13	Detection of Runtime Conflicts among Services in Smart Cities	2016	C	28	Toward a programming model for safer pervasive spaces	2010	W
14	Minimalist qualitative models for model checking cyber-physical feature coordination	2016	C	29	Feature interaction detection in the automotive	2008	C
15	SPIRE: Scalable and Unified Platform for Real World IoT Services with Feature Interaction	2016	C				

Table 10: List of the papers selected for the study (J: Journal, C: Conference, W: Workshop)

- of Ambient Intelligence and Humanized Computing, 5(5):699–715.
- Sharma, S. and Coyne, B. (2017). Devops for dummies. 3rd limited ibm edn.
- Shih, C.-S., Chou, J.-J., Reijers, N., and Kuo, T.-W. (2016). Designing cps/iot applications for smart buildings and cities. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):3–12.
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339.
- Teixeira, T., Hachem, S., Issarny, V., and Georgantas, N. (2011). Service Oriented Middleware for the Internet of Things: A Perspective. In Abramowicz, W., Llorente, I. M., Surridge, M., Zisman, A., and Vayssière, J., editors, *Towards a Service-Based Internet*, Lecture Notes in Computer Science, pages 220–229, Berlin, Heidelberg. Springer.
- Tuttliès, V., Schiele, G., and Becker, C. (2007). Comity-conflict avoidance in pervasive computing environments. In *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pages 763–772. Springer.
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, page 38. Citeseer.