Semantic Interoperability Methods for Smart Service Systems: A Survey*

Fabian Burzlaff, Nils Wilken, Christian Bartelt and Heiner Stuckenschmidt

Abstract—Functional and non-functional characteristics of software systems are defined by their architecture. Therefore, research streams such as Internet-of-Things or component-based software engineering provide researchers and practitioners with construction guidelines for selected architectural characteristics. Current systems can be categorized in delivering services to the user and being engineered in a smart way. For example, services being provided by IoT-systems must fulfill users' goals in a highly dynamic and ad-hoc way. Consequently, this survey aims at answering various research questions regarding the methodical composition of system components and services. Furthermore, new research opportunities are sketched that should be tackled to make the scientific progress available to practitioners. Based on a systematic literature review from a software architecture point of view, we have identified 75 primary studies for domain-specific IoT component composition approaches and architectures. Initial results show that current integration approaches mainly focus on performance evaluation of their integration solutions, which may be too narrow for fulfilling user goals by utilizing of IoT architectures.

***Index Terms*—Semantic Interoperability, Software Architecture, AI Systems Engineering, Knowledge-based Methods and Approaches, Systematic Literature Review**

I. Introduction

Many technological innovations such as collaborative robots or self-driving cars are currently making their way into everyday life. These things usually do not get their "smartness" out of a closed system but are connected to multiple other systems to sense their environment and reason about their current context. Hence, these "smart" systems can be defined in the application domain Internet of Things as "co-engineered interacting networks of physical and computational components" [1]. Such networked systems typically fulfill domain-specific goals which are realized based on selected system qualities (e.g. reliability, scalability or self-X properties). In contrast to traditional IT-Systems (e.g. an accounting system), open IoT-Systems consist almost always of a larger "swarm" of devices. To cope with the dynamic interplay of platforms and devices in an automated and reliable way, application programming interfaces (API) must be standardized. This makes software development and device interaction in general efficient. Although it will be possible to abstract from device- and vendor specific peculiarities and create cross-industry APIs in the upcoming years [1], we believe that it is not possible to standardize the semantics of (composite) domain-specific functionalities inside and across distinct application fields. Current top-down standardization and engineering approaches cannot deal effectively with the unpredictable availability of services being offered by IoT Systems. Despite the undeniable scientific progress in the research areas of component-based software engineering (CBSE) [2] and web service engineering (WSE) [3], the composition process of spatiotemporal developed software components into a system architecture is once again challenged by a new system class – IoT. To fulfill ad-hoc defined and domain-specific user goals, such systems must be engineered in both, top-down and bottom-up ways which is a challenge for most traditional composition processes. Furthermore, these systems must fulfill system qualities such as robustness to device unavailability or self-adaptivity [1]. Therefore, we derive service engineering and software architecture challenges with a strong focus on integration methods and the management of architectural knowledge [4] for achieving reliable service delivery to end users. Therefore, this survey provides initial results based on 75 rigorously selected primary studies all dealing with the overall research question:

"How can IoT-Systems engineering processes be supported by integration methods to achieve semantic interoperability between applications, services and software platforms?". Answering this question could help to realize the service computing vision [5].

II. Context and Research Method

This chapter outlines the research context and justifies the selected research questions. Next, the overall research process is described, and the data extraction template is introduced in detail.

First, we will outline relevant questions regarding architectures for distributed software systems and component composition

The authenticated final version is available online at F. Burzlaff, N. Wilken, C. Bartelt and H. Stuckenschmidt, "Semantic Interoperability Methods for Smart Service Systems: A Survey," in IEEE Transactions on Engineering Management, doi: 10.1109/TEM.2019.2922103.

F. Burzlaff is with the Institute for Enterprise Systems, University of Mannheim, Germany (burzlaff@es.uni-mannhem.de) and acts as the primary contact for this publication.

N. Wilken is with the Institute for Enterprise Systems, University of Mannheim, Germany (wilken@es.uni-mannheim.de).

C. Bartelt is directory of the Institute for Enterprise Systems, University of Mannheim, Germany (bartelt@es.uni-mannheim.de).

H. Stuckenschmidt is a full professor for Artificial Intelligence at the Data and Web Science Group, University of Mannheim, Germany (heiner@informatik.uni-mannheim.de).

methods from the area of CBSE and WSE. Overall, questions are related to system design- and runtime properties.

A. *IoT-Architectures for distributed software systems*

Smart service systems that fulfill end-user goals by utilizing a set of devices can be affected by the following four influencing factors:

- Engineering Approach
- Technologies
- Class of System
- Field of Application

Based on the class of system (e.g. embedded or cyber-physical system (CPS)) that primarily defines the computational and physical boundaries at the network edge, services offered by a device are primarily constrained by the technology being used. On the one hand, technology selection is highly driven by domain-specific and non-functional requirements. For example, a smart home automation system does not normally need to operate in real-time, but a mission-critical collaborative robot should. On the other hand, the software engineering methodology defines non-functional properties concerning the qualities of IoT-Systems such as reusability of components or self-X properties (e.g. self-adaptation or self-configuration). In order to make software engineering processes and the platform itself more flexible, architectural knowledge management techniques [4] and middleware containing a formal knowledge-base can be used [6].

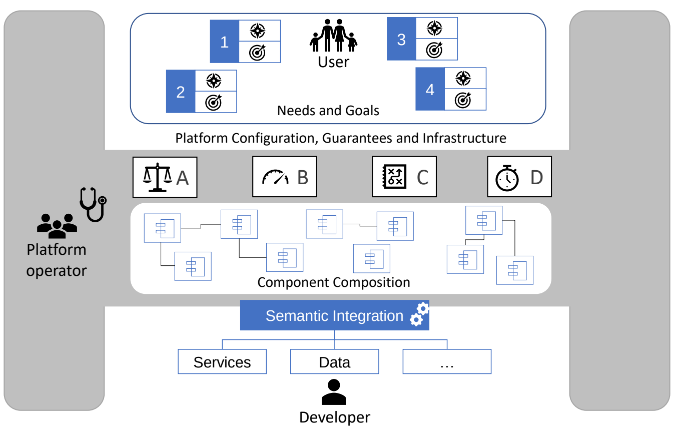


Figure A: Influencing Roles for IoT-System Architectures

Which roles are important when designing IoT-Architectures and what are their concerns?

IoT-Systems architecture roles can be split up into service developers, platform operators and users (see Figure A). Platform operators ensure the correct collaboration between available devices and enforces non-functional properties such as platform availability, service interoperability or scalability. Service developers are concerned with the integration of data, services and business processes based on different sources. Users or a context-sensing device pose needs and goals towards the platform. During runtime, the platform must decide based on service availability and intended usage on whether a request can be fulfilled. As the integration of services offered by heterogenous systems requires manual work, the platform should support the software developer in an effective way.

Which architectural patterns exist when designing IoT systems?

There is no silver bullet to create seamless fitting software architectures in the IoT domain. Architectural mismatch caused by heterogenous device structures is the main reason why services realized by software components cannot be reused without any additional effort towards the engineering process itself. Hence, connectors and adapters are needed in order to reuse available services [7], [8]. Although such integration issues are already solved within predefined boundaries such as Pipes and- Filters, Enterprise Service Buses or Software Product Lines, the evolution of system assemblage consisting of old and new systems is challenging. Although supporting standards and frameworks exist, manual integration effort will be required per use-case.

How do IoT-Devices offer services and how can they be integrated based on current standards?

The main interoperability problems being faced are, among others, heterogeneity of communication protocols, platforms and technical standards as well as syntactical and semantic heterogeneity of data and functions [9]. A lot of research has been conducted for solving semantic interoperability problems based on using formal standards, ontologies or semantic mediators. However, these solutions are perceived as “heavy-weight” by practitioners. Hence, informal standards such as OPC UA[1] are favored over SAWSDL[2] descriptions which would expose formal service semantics.

B. *Composition approaches for software components*

The minimum requirements of a software component model can be categorized as: 1) component description 2) rules for component interoperability 3) precise interface description regarding the component functions 4) interoperability mechanism for using the interface and 5) component runtime behavior truly exposes the aforementioned properties [10].

In contrast to closed IT-Systems, the amount of possible states at runtime for dynamic IoT-Ecosystems cannot be anticipated at design time. Consequently, the combination of nonfunctional service properties (e.g. temporal semantics) of, for example, multiple connected CPSs is a hard verification challenge. This means for IoT-Platforms operators that they must ensure that there exist a machine-readable structural and behavioral component description.

How can domain-specific component interfaces be described?

The interfaces of a software component are mainly described by interface types, the distinction between required and provided functionality, its features, the interface language and the interface levels [10]. For example, cyber-physical systems descriptions can be formalized using classical interface description languages or a programming languages [11], [12]. Using formal descriptions has the advantage that cross-cutting aspects can be defined on a global platform level. A disadvantage of such descriptions is the additional specification effort. Although such components expose a formal component

[1] https://opcfoundation.org/about/opc-technologies/opc-ua/

[2] https://www.w3.org/TR/sawsdl/

description to realize automated component composition during design and runtime, they are rarely used in practice.

Reuse of components and service composition aims at lowering the adaptation time of existing systems based on changing

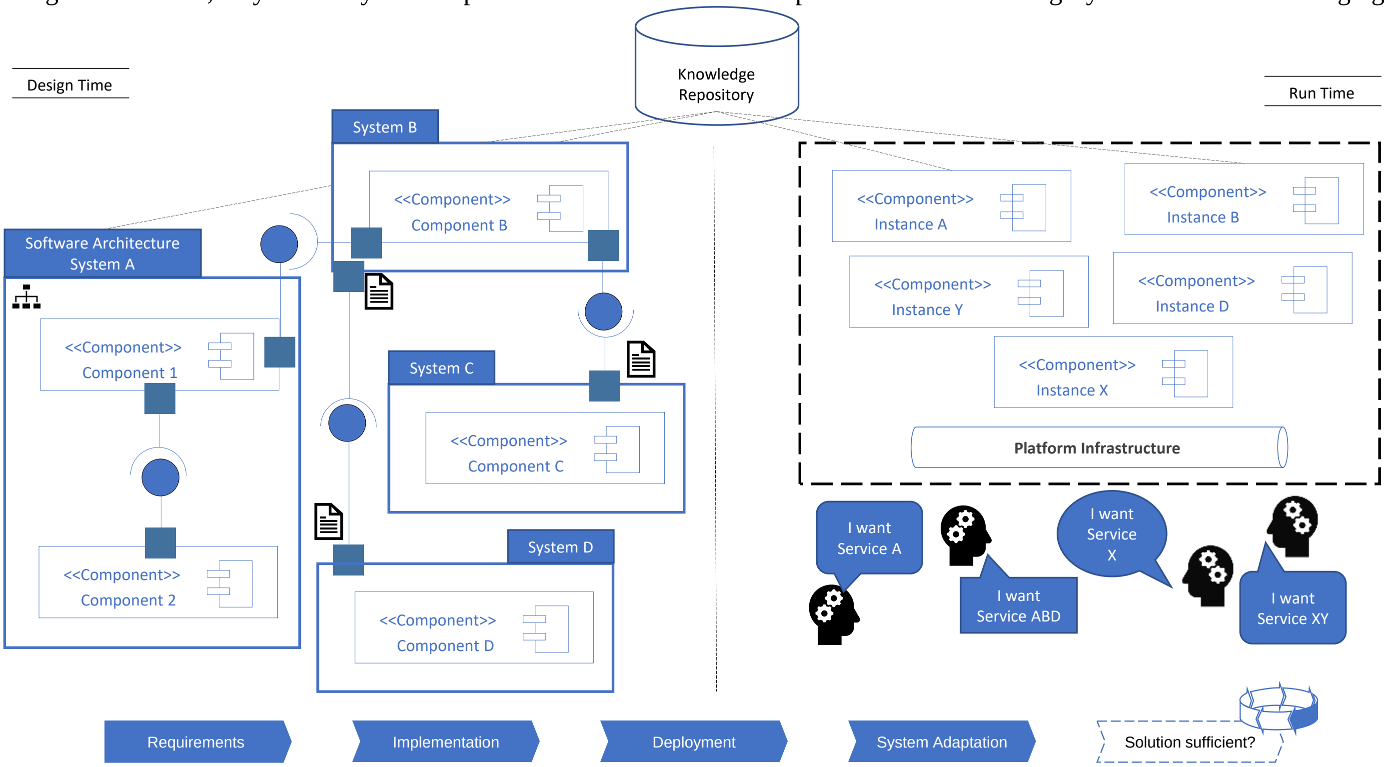


Figure B: Composition challenges for smart service systems

How and when do component development processes compose software components?

Software component compositions in traditional CBSE approaches involve creative work [10]. Depending on the available components, new functionality is realized by producing newly assembled component artefacts (see Fig. B). Within this graphical model, various composition (c.f. integration) challenges can be conceptually located. Regarding data/information provided by IoT devices, corresponding component states can be determined. On the next layer, a service typically requires an input set of arguments and produces an output. By doing so a computational part is involved. Last, an application can be created by composing multiple services and information gathered from devices and context.

As the integration technique is determined by the type of component model used, the component technology and the component selection itself, integration issues should be ideally addressed already in the early phases of the component development process. Component integration can be either performed horizontally or vertically (e.g. calling a function on the same level of abstraction or delegating a function call from an outer component to an inner component). Concerning the example of CPSs, most approaches focus on component deployment at compilation time and not at runtime as during compilation time a better predictability for the set of known required and provided components exists.

How can software components and service compositions be reused?

requirements. In the context of web services, reuse artefacts can be distinct components, data transformation rules, process fragments or examples [3]. Manual reuse techniques range from copy & paste and keyword-based component search engines to complex component recommendation systems. In order to automate service composition, semantic service descriptions (e.g. based on OWL and/or SOAP) are needed. However, many IoT-devices do expose their services using REST and are typically not equipped with a machine-readable functionality description. Hence, system integrators must interpret available component interface artefacts and eventually codify their interpretation as well as their architectural knowledge in a software domain-specific adapter.

C. *Need for conducting this review*

IoT-Systems expose new challenges to the software architecture community. Services realized by IoT-Systems must deal with mobile service providers, heterogenous software interfaces and (self-) adaptability mechanisms to deal with uncertainty. Dynamic integration mechanisms at both design and runtime stages must be present to cope with unpredictable user needs. For example, software and systems architectures that are integrated at design time may not be able to fulfill user goals at runtime as some goals were not taken into consideration (see required Application XY in Fig. B). Hence, IoT-Systems must be engineered with a focus on flexibility. Flexibility is needed when minimizing the knowledge gap between unknown and known user goals which are again realized by provided and required software components at various lifecycle stages.

In this context this survey especially focuses on the interplay of semantic interoperability (i.e. knowledge-related activities) and IoT-Systems. The illustrated technical example problems are not new to the computer science community. Based on the search string (1) multiple adjacent systematic literature reviews as well as systematic mapping studies could be identified.

(1) ("Systematic Mapping Study" OR SMS OR "Systematic Literature Review" OR SLR OR "Literature Review" OR LR) AND (IoT OR "Internet of Things" OR "Internet-of-Things" OR IoE OR "Internet of Everything" OR "Internet-of-Everything" OR WoT OR "Web of Things" OR "Web-of-Things") AND ("Software Architecture" OR "System Architecture" OR Architecture) AND ("Interoperability" OR "Integration")

A well-known review dealing with the "Internet-of-Things" in general has been published by Aztori et al. [13]. Aztori et al. categorize the IoT concept into three overlapping visions named Things-oriented, Internet-oriented and Semantic-oriented Vision. Another highly cited review was conducted by Al-Fuqaha et al. [14] where the authors looked at different vertical markets and their integration from a technical viewpoint (e.g. Agriculture, Manufacturing and Health Care). From a practical point of view standards [15]–[17], market perspectives [18], [19] and IoT platforms [20] are of particular interest. The integration of heterogenous things [21], the deployment of smart spaces [22] and context aware computing [23] have been already surveyed within the IoT domain. Relevant to the proposed research question (see Introduction) are studies dealing with knowledge-based approaches to software systems [24]–[26], semantic interoperability within the Internet-of-Things [27], [28], IoT-Architectures and services [29]–[32] as well as ontologies [33].

Isolated concepts such as knowledge-management activities (e.g. reuse) as defined in [25] and hierarchies describing Interoperability at several levels such as syntactic, device, platform and semantic level [28], [34] are important to this survey. However, none of the found studies investigates how semantic integration knowledge can be created, stored, shared and reused driven by system engineering approaches. Nevertheless, there are recent studies such as [34], [35] that tackle integration approaches from a pure technical perspective meaning that these studies do not have an explicit focus on engineering processes

D. *Research Questions*

Our motivation for conducting this review is codified within the overall research question:

"How can IoT-Systems engineering processes be supported by integration methods to achieve semantic interoperability between applications, services and software platforms?"

From a scientific point of view, this question is relevant as currently proposed state-of-the-art solutions such as formal Interface Definition Languages (c.f. OWL-S[3]) and formal semantic processing engines (c.f. AUTOSAR[4]) rely on the assumption that standardization is a premise for automated component coupling approaches. The Internet-of-Things challenges this assumption due to heterogenous device manufacturers that are expected to cooperate in cross-domain settings.

From a practical point of view, the emerging formalization effort for using formal semantic standards is a non-neglectable circumstance regarding fast innovation lifecycles for software, skilled personal and implementation effort. Current interface description efforts (e.g OpenAPI Specification[5]) and semantic messaging formats (e.g. JSON for Linked Data[6]) can be interpreted as a means to deal with the well-known interoperability gap within IoT-Systems [13]. However, current scientific solution proposals fall short when semantic aspects of IoT data and services are influenced by end user applications during system lifetime (e.g. writing automation rules in smart home scenarios like openHAB[7]). Novel engineering approaches should consider these semantic interoperability circumstances also from an engineering management viewpoint in order to enable "plug-and-play" scenarios for end user goals and needs. To answer the overall research question in a structured way, four sub-questions (RQ1-RQ4) have been selected to specify the review scope.

RQ1: Which semantic interoperability approaches are currently studied in the context of IoT?

Motivation: This question deals with semantic interoperability activities. According to the well-known DIKW hierarchy [36] knowledge can be interpreted as "the ability to reason about information". This interpretation is well aligned with the definition of semantic interoperability regarding semantic aspects of expression from one language and transformation processes between (modelling) languages. Regarding the semantic aspect of an expression, Euzenat et al. [37] defines semantic interoperability as "being able to construct the propositional meaning of a syntactic representation". Regarding the transformation process of different languages, every syntactic expression must map to at least one semantic domain element [38]. In the IoT context this means that multiple languages map a (sub-)set of their syntactic expressions to an identical syntactic (sub-)set within a semantic domain. Here within lies the semantic interoperability problem that researchers try to tackle in the IoT context by making implicit integration knowledge of various roles explicit (e.g. using programmatic examples, manuals or standards. We are explicitly not including pragmatic language aspects or context-aware systems to reach a concrete action decision.

RQ2: Which knowledge-based activities do integration approaches tackle and when are they reified?

Motivation: Knowledge and semantic interoperability are conceptually related. Both concepts are a prerequisite in order to make IoT systems "smart". Here, artificial intelligence techniques come into play as they are a means to extract

[3] https://www.w3.org/Submission/OWL-S/
[4] https://www.autosar.org/
[5] https://github.com/OAI/OpenAPI-Specification
[6] https://json-ld.org/
[7] https://www.openhab.org/

patterns from heterogenous information and service sinks. To structure the related knowledge-based activities in a meaningful way, we reuse activities from the knowledge management society. According to Alavi et al. [39] there are creation, storage/retrieval, transfer, application activities. To transfer these activities to the software architecting process, Li et al. [25] defines the following mappings which are themselves based on various surveys [40], [41]:

Table A: Knowledge-based approaches for software architecture (adapted from [25])

Knowledge-based approach [25]	Knowledge activities based on the KM framework [39]
Knowledge Capture and Representation	Knowledge storage
Knowledge Reuse	Knowledge application
Knowledge Sharing	Knowledge transfer
Knowledge Recovery	Knowledge creation
Knowledge Reasoning	Knowledge creation

Li et al. [25] note, that they see knowledge retrieval as a supporting activity for the others. The reason for this statement may be found in their viewpoint on software systems. Their primary role "software architect" that uses knowledge-based approaches is naturally equipped with relevant skills to perform CRUD (Create-Read-Update-Delete) activities on technical knowledgebases (e.g. using SPARQL). However, within the IoT context also other roles with different skillsets interact with the system. This can be also seen in arising service mashup tools such as Node-RED[8] that enable users to program Internet-of-Things devices in a flow-based manner. Hence, we also include relevant technologies and especially their languages to retrieve knowledge as the usability of such retrieval techniques may influence the applicability of composition methods. In addition, data and service semantics may be determined by the end user during system installation.

RQ3: Which semantic interoperability challenges are currently being tackled in the IoT domain?
Motivation: Answering this question will provide evidence about which challenges for semantic interoperability for IoT systems are currently being tackled. Found challenges can be categorized and compared to known interoperability challenges [34]. Among others, these are 1) use-case centric IoT solutions 2) formalization and use of domain knowledge 3) integration of explicit and implicit conceptual schemata and 4) reuse of existing standards. Based on this, open challenges for future IoT systems can be articulated and reasoned about. This may be interesting for practitioners to judge whether current solutions suffice their application context.

RQ4: How are semantic interoperable IoT methods and architectures evaluated?
Motivation: This question sheds light on how semantic interoperability solutions are evaluated scientifically. As a consequence, solution characteristics based on quantifiable results can be identified. In addition, the type of evaluation provided (e.g. toy example, use case or formal experiment) can help to identify the maturity level of the solution proposals found. Especially the required skillset for executing the proposed approaches and technologies needed as well as the human involvement within the evaluation will provide insights. Lastly assumptions made for conducting the experiment will be categorized and highlighted.

E. Inclusion and Exclusion Criteria

As the term "Interoperability" and "Semantic" are used in various research communities, they have been themselves assigned different meanings. To cope with this circumstance, we have defined the following inclusion and exclusion criteria:

Inclusion criteria: All publications must be written in English. All publications must be available in full-text and subject to a peer review process. All publications must be dealing in some way with semantic interoperability and explicitly leverage approaches, methods or techniques (i.e. focus on engineering activities). All publications must address either data or service integration issues. All publications must at least contain six pages in double-column format.

Exclusion criteria: Studies, that do not explicitly describe an evaluation method or do not contain some technical evaluation at all. Studies that themselves are secondary or tertiary studies (e.g. Systematic Literature Reviews or Systematic Mapping Studies). Studies that focus on interoperability aspects other than software. Studies that focus on analytical aspects of software systems (i.e. explicitly do not address system engineering problems). All publications addressing other integration issues (e.g. Process Matching).

F. Search process

The search process has been split up into the following five search steps.

Step 1 was split up into an automatic and a manual search. On the one hand, an automatic search string was applied. On the other hand, a broader search string was issued to selected journals and conferences. Regarding inclusion criteria the title was scanned and the formal aspects (e.g. length of publications and peer-review process of publisher) were investigated.
Step 2 consisted of reading the abstract to prune out publications that could be clearly classified by at least one exclusion criteria.
In **Step 3** all evaluation sections were scanned. Publications must contain a technical evaluation that is at least a toy example grounded on a detectable set of technologies. Evaluation sections that did not contain an evaluation but rather a pure conceptual application or a qualitative authors judgment about advantages and disadvantages were sorted out.
Step 4 contains a snowballing process based on the publications that passed all previous rounds. We performed a forward- as well as a backward-snowballing step, where forward means evaluation publications that cite a publication and backward means referenced publications of a publication. Finally, search process steps 1 to 4 have been applied to the newly found

[8] https://nodered.org/

publications. Lastly, all publications have been read in full to extract all relevant data into a table-based data extraction form. Regarding the overall search process, the following assumptions hold.

1) Scope and Time

We did not specify a publication time span for primary studies. Nevertheless, there exist an implicit time boundary due to the circumstance that the label "Internet of Things" was coined by Kevin Ashton in 1999[9] and is included in our search string.
The scope of our search was limited to composition methods that tackle networked software interfaces and nothing else. For example, composition methods that relied on a hardware-dependent were excluded.

2) Electronic Databases

Table B lists all electronic databases that have been queried during automated search. Although there may exist other electronic data bases, we believe that the selected ones are more than enough to uncover all relevant and influential publications.

Table B: Electronic Databases included in the automated Search

ID	Electronic Database
DB1	IEEE Xplore
DB2	ACM Digital Library
DB3	Science Direct
DB4	SpringerLink
DB5	Wiley
DB6	ISI Web of Science

3) Journals, Conferences and Workshops

In Table C all Journals are display that have been manually queried with an adapted search string. Similarly, in Table D all conferences are displayed. As the claim of this publication is to assume a software architecture viewpoint on composition methods in the IoT domain, well-known journals and conferences that deal with software architecture-related aspects have been queried manually.

Table C: Journals included in the manual Search

ID	Journals
J1	Empirical Software Engineering
J2	Information and Software Technology
J3	Journal of Systems and Software
J4	Software and System Modeling
J5	Transactions on Software Engineering

Table D: Conferences included in the manual Search

ID	Conferences
C1	International Symposium on Component Based Software Engineering (CBSE)
C2	European Conference on Software Architecture (ECSA)
C3	International Conference on Software Architecture (ICSA)
C4	International Conference on Software Engineering (ICSE)

[9] https://www.rfidjournal.com/articles/view?4986

4) Search Terms

The PICO criteria [42] have been used to structure search strings. PICO translates from an acronym to the English language as follows: **P**opulation deals with the class or group that is central to the research questions. **I**ntervention defines an action, tactic or modification that is applied to the population in some **C**ontext. Lastly, **O**utcome can be defined as the result after applying an intervention to a population in some context. This result can expose qualitative and/or quantitative characteristics.
Our search string used for the automated search instantiates as follows:

#1 Automated Search String:
Population: (IoT OR "Internet of Things" OR "Internet-of-things" OR IoE OR "Internet of Everything" OR "Internet- of-everything")
Intervention: (Method OR Technique OR Approach)
Outcomes: ("Semantic Integration" OR "Semantic Interoperability" OR "Semantic Computing")
To confirm the meaning of each search term within the relevant research fields, a pre-study has been conducted to identify the most used words within each community that aligns with the intended meaning.
All search terms in parentheses have been combined with the logical operator AND (e.g. ..."Internet of Everything" OR "Internet- of-Everything") AND (Method OR Technique …). Terms displayed in quotation marks mean that this exact search term must be present whereas no quotation marks generally means that the word stem must be present. All search engines have been configured so that one term from each sub-group must be present at least in either the title and abstract or in the text body.
During search, some search engine peculiarities occurred.

- ScienceDirect (DB3) allowed at maximum eight different search terms. Therefore, the Population category has been shortened to (IoT OR "Internet of Things" OR "Internet-of-Things")
- SpringerLink (DB4) was only searched within the "computer science" category
- ISI Web of Science (DB6) needed some syntactic sugar to correctly parse the search string. Here the keyword ALL was added so that each PICO category contained an additional parentheses construct (e.g. ALL=(Population) AND ALL=(Intervention) and …)

Regarding the manual search, the search term was adapted so that it covers a broader scope. This is necessary as there exists semantic integration approaches prior to the time when "Internet-of-Things" was a widely used term within research and practice. For example, the component-based software engineering community dealt with semantic interface ambiguities long before the term "IoT" was coined [2]. Depending on the degree of automation, multiple approaches also exist that rely on integration knowledge which in turn can be either formalized or not. Hence, the search string was adapted in the following way:

#2 Manual Search String:
(IoT OR "Internet of Things" OR "Internet-of-Things" OR IoE OR "Internet of Everything" OR "Internet-of-Everything") AND (Semantic OR Interoperability OR Integration)

#3 Manual Search String:
(Method OR Technique OR Approach) AND (Semantic OR Interoperability OR Integration)

The search process for selected conferences and journals included not only the automated search string #1, but also both manual search strings #2 and #3.
We consciously did not include any workshops.

5) Search Strategy

The reason for choosing two search methods, automated and manual is rooted in the application spectrum of "semantic interoperability. From a software engineering perspective, interoperability issues can arise when software is developed by multiple parties that do not share common concepts of a domain (e.g. taxonomies or vocabularies). Hence, interoperability is currently driven by standardization initiatives. However, such efforts require a critical mass of individuals to agree on a subset of terms and grammars in a uniform way. Here, manual search steps are necessary for staying within the frame of software architectures and especially the management of composition approaches for IoT-Systems. Because of the manual search, we believe that almost all relevant publications are found.

G. Data Extraction and Synthesis

A data extraction template was created in order to extract data from primary studies in a structured way. This data is visualized in Table E and can be accessed on this website[10]. For answering our research questions, the extracted data was synthesized accordingly. Aggregated data such as time-series or amount of publications per conference (i.e. "most" relevant conferences) will be discussed in more detail in Section Survey Results.

Table E: Data Extraction Template and assigned Research Questions

Item ID	Item Name	Description	RQ
I1	Publication Year	Year of publication	RQ1
I2	Publication Venue	Conference/Journal/Book publisher of the item	RQ1
I3	Item Type	States whether it is classified as a Conference/Journal/Book contribution	RQ1
I4	Type of Contribution	Indicates the authors contribution (e.g. proposing a new method or model for a certain activity)	RQ1
I5	Research Challenges	Describes the abstract and/or concrete research challenge(s)/question(s)	RQ3
I6	Evaluation Strategy	States evaluation strategy as defined by the authors. Includes the fields type, independent and dependent variable, evaluation goal and human evaluation involvement	RQ4
I7	Applied Interoperability Model	Indicates how semantic interoperability is achieved. Includes the fields predefined, reused standards, fully automated, languages&tools and semantic emphasis	RQ1
I8	Interoperability Viewpoint	Describes whether the approach is applied to Data/Information/Sensors/Things or Services/Interfaces/API	RQ1
I9	Knowledge Management Activities	Includes the activities reuse, share, reasoning, capturing,	RQ2
I10	System Lifecycle	States the lifecycle phases of the overall system. Phases are requirements, design, implementation, deployment, testing and runtime	RQ2

Regarding definitions of the extracted items, it was decided not to align them with one specific definition. The reason for this decision lies within the assumption that not all authors themselves explicitly provide one. Hence, selected extraction items are allowed to have different meanings. However, this circumstance does not affect the data extraction process due to well-defined inclusion and exclusion criteria as well as extraction operationalization efforts.
Overall, the data extraction process was kept as simple as possible to reduce the amount of interpretation effort. This means that whenever possible, generic functions such as searching for terms based on their word stem and synonyms was used. For example, when deciding whether a publication deals with aspects of knowledge capturing, words like representation and descriptions have also been included in the query issued in a publication. Furthermore, items such as type of contribution (I4) are commonly found within the abstract, at the end of the introduction or in the conclusion section. Again, this minimizes the authors bias of interpreting terms from a subjective viewpoint.
Regarding the instance values allowed for each item, either *boolean* or *strings* are used within the template. Depending on the occurrence context (e.g. sentence or paragraph), it was decided whether the occurrence of *boolean* values such as knowledge management activities (I7) are part of the overall contribution or being used to define the contribution scope. For example, the definition of an ontology published by Gruber [43] "An ontology is a formal, explicit specification of a shared conceptualization" was found in various publications. However, this search hit does not directly contribute to the knowledge-management activity of "sharing" as the former is framed as a structural property and the latter as a behavioral activity. If only this result was returned from the search, the publication was not ticked as relevant.

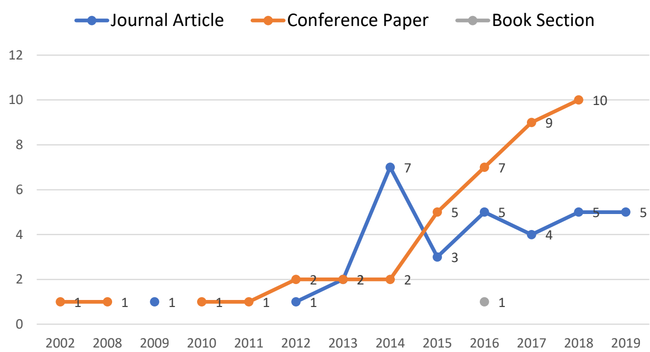
[10] http://www.institute-for-enterprise-systems.de/fileadmin/20190522_DataExtractionForm.xlsx

For string input fields, all relevant occurrences were copied from the publication into the template. As a next step, all instances of an item were rearranged as a list and sorted alphanumerically. Finally, all instance occurrences were checked for different spellings with the same meaning. For example, the publication venue (I3) sometimes contained the event year.

III. Survey Results

In this section, we provide answers to the research questions defined in Section Research Questions. First, we provide a descriptive overview about our findings. Second, we answer our research questions by using textual and visual descriptions. Third and last, we critically discuss and interpret our findings in the context of open research challenges.

A. *Overview of Results*



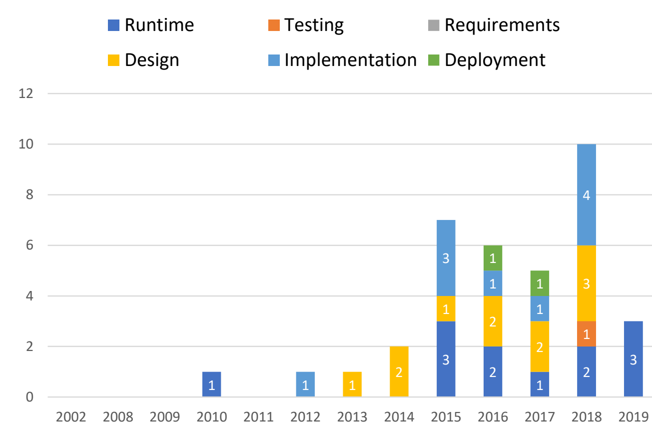
Graphic A: Time-Series Overview for Primary publications

When issuing the automated search string to Google Scholar, the result set contained approximately 5300 publications[11]. After conducting both, automatic and manual search, 207 publications were marked as relevant according to their title. Search step 2 reduced the amount of publications to 107 and search step 3 again reduced the number to 83. These publications were read in detail. After this, the amount of publications was again reduced to 64 primary studies that made it into the synthesis phase. Last, the forward- and backward-snowballing process increased the total amount of primary studies to 75 publications. These 75 publications contain 41 conference publications, 33 journal articles and 1 book section and were published from 2002 until 2019 (see Graphic A). We conducted our search process from December 2018 until the 5th of April 2019. It can be observed that the publication trend for conference and journal publications is increasing from 2013 until 2019. Furthermore, the amount of relevant conference publications in 2019 are already identical to the year 2018 at the time of writing this survey (April 2019).

B. *RQ1: Which semantic interoperability approaches are currently studied in the context of IoT?*

Foundation: For answering the first research question, the meta-data about a publication including publication year, venue and type (I1-4) are relevant. The publication content is screened regarding interoperability viewpoint (I7) and the applied interoperability model (I8).

From 75 primary studies, 17 studies claimed that they support fully automated approaches and 58 did not. From an interoperability viewpoint, 44 publications dealt with interoperability questions from a service viewpoint and 32 publications adopted the viewpoint of data. Here, the term "service" is conceptually aligned with Application Programming Interface (API) or Interface. In contrast, the term "data" conceptually refers to things, information and sensors. The difference between both concepts are grounded within the system architecture. A function invocation (using a network layer) involves handling thing behavior in a domain-specific manner (i.e. stateful) whereas performing create-read-update-delete operations are domain-independent (i.e. stateless).
Regarding the publication venue, there exist 56 different venues. The journal IEEE Transactions on Industrial Informatics published with 5 articles the most publications. On the second place, the European Conference on Software Architecture (ECSA) and the International Conference on Ambient Systems, Networks and Technologies each offer 3 publications. The remaining 53 publication venues either have 2 publications (10 venues) or 1 publication (43 venues).



Graphic B: Amount of Contributions per Lifecycle Phase

In Graphic B, the distribution of semantic interoperability contributions are displayed per year. Overall, 48 publications did not explicitly state any lifecycle phase. One explanation for this can be the type of contribution. Due to the focus on software architecture interoperability, many authors propose a reference architecture or showcase a non-functional property of a prototypical architecture. Hence, authors do not explicitly view semantic interoperability from a process-centric lifecycle viewpoint but rather from a structural viewpoint that exposes a desired system characteristic. Thus, 27 publications remain which are displayed in Graphic B. It can be observed that most semantic interoperability approaches are applied during runtime (12 publications), during design time (11 publications) or during implementation (10 publications). Approaches for deployment (2 publications), testing (1 publication) and during the requirements phase (0 publications) are rare.

[11] Query executed on 05.04.2019

Concerning the application interoperability model, Table F shows the technologies used for achieving or supporting semantic interoperability.

Table F: Overview of Languages and Tools per Purpose

Purpose	Languages&Tools
Semantics	OWL, OWL-S, RDF, DAML-S
Query	SPARQL
Rules	SWRL
Interface	WSDL, SAWSDL, RAML, RESTdesc
Messaging Protocol	CoAP, MQTT, HTTP, AMQP
Messaging Format	JSON-LD
Thing Description	Thing Description Model, DEECo Component Model
Architectural Style	REST
Reasoner and Solver	Euler Yet another proof Engine, FaCT+++, Stanford Research Institute Problem Solver (STRIPS), OLTSA (Model Checker)

Regarding theoretical frameworks and proprietary technologies, authors used Enhanced Labeled Transition Systems (P7), Stochastic Activity Networks (P7), Model Checkers (P8), Semantic line networks (P12), Labeled Transition Systems (P19), Context-aware symbolic Transition Systems (P21), Ordered binary decision diagrams (P21) and Hierarchical Task Planning (P30). The set of proprietary technology include the STARLINK framework (P8), EclEmma (P24), JASON (43), JACAMO (P43), WS4DJ (P45), Alloy REST (P62), SQenIoT JSON (P63) and MetaMap Spark (P71). The most reused standards included the Semantic Sensor Network Ontology (P2, P5, P34, P52, P57, P67), W3C IoT Thing description (P11, P15) and GeoSPARQL (P46, P56, P57).
The arguably most relevant part for achieving semantic interoperability based on a structured method is the semantic emphasis of each proposed approach. Foreclosing the detailed results, it can be observed that the semantic emphasis of each approach is highly diverse. Overall, various activities are needed to integrate Things, Architecture and Applications into an overall service system. Depending on the underlying architecture, different steps are needed. For example, integrating an IoT-Device into a BUS architecture is conceptually different compared to integrating an IoT-Devices into a Client-Server architecture. Nevertheless, all architectural concepts may be adapted in a way so that they expose composition activities (please note, that the adaptation effort for each architecture is neglected within this statement on purpose). The activities found from a service viewpoint are:

- Annotation: P29, P17, P32
- Description: P66, P55, P52, P40, P41, P45, P30
- Discovery: P59, P54, P75
- Reasoning: P26, P19, P29
- Composition & Coupling: P53, P70
- Interoperability & Compatibility: P8, P61
- Matching: P11, P21, P51
- Mapping: P47
- Request resolution: P1

It is worth noticing that, despite the automated search string aiming at semantic interoperability, there are 15 publications that do not specify a semantic emphasis. Among other reasons, this is due to the manual search. For example, a human must decide about the semantics at the time of component integration.

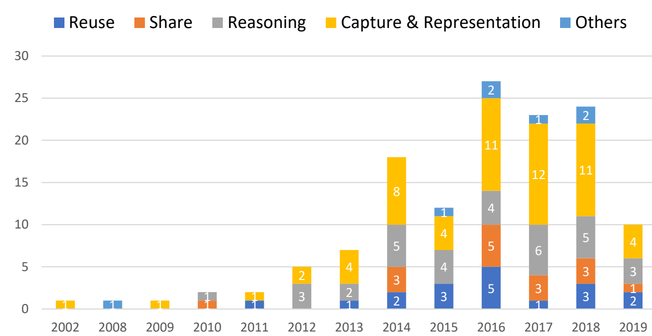
From a thing viewpoint, the following activities could be found:

- Annotation: P48, P16, P67, P2, P71, P56, P25, P37
- Definition of Model: P12, P23, P34, P39
- Description: P13, P31, P35, P18
- Representation: P50, P20
- Query: P5
- Ontology Matching: P49
- Integration: P58, P36

Again, there are 3 publications that do not specify any semantic emphasis at all. The aforementioned reasons for services can be applied as well.

C. *RQ2: Which knowledge-based activities do integration approaches tackle and when are they reified?*

Foundation: For answering the second research question, the extraction items knowledge-management activities (I9) and system lifecycles (I10) are relevant.

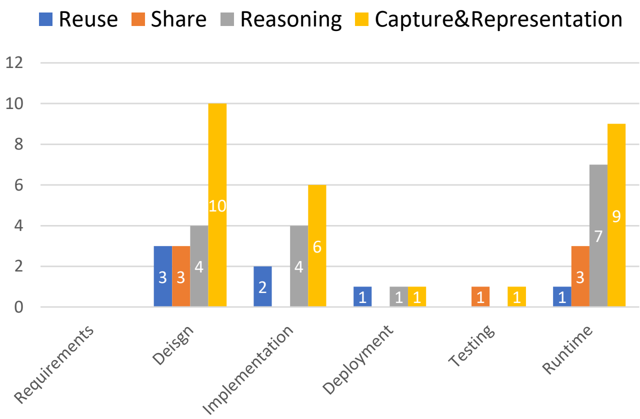


Graphic C: Amount of Knowledge-Management Activities per Year

The knowledge-related activity *Capturing&Representation* is the most performed activity (i.e. 59 publications). Next *Reasoning* (33 publications), *Reuse* (18 publications) and *Share* (16 publications) follow. In 7 publications, other activities are performed that did not match the proposed knowledge-related activities. From 2002 until 2019, the increasing amount of publications can also be identified within the proposed semantic interoperability approaches per activity (see Graphic C). For example, the amount of *Capturing&Representation* activities is increasing until 2016 and then stays at a constant level. Regarding this synthesis process, it should be noted that one approach can support multiple knowledge-related activities.

In Graphic D, all knowledge-related activities are displays in relation to the application phase of the underlying system architecture. Here, each publication can only be assigned to one lifecycle phase.
At first, it should be noted that no semantic interoperability approach was assigned to the requirements phase. Next, knowledge capturing, and representation is in general either performed during design time or during runtime of IoT-Systems. In addition, reusing activities are mostly performed during design time. The least knowledge-related activities are

performed within the deployment and testing phase.



Graphic D: Amount of Knowledge-Management Activities per Lifecycle Phase

D. RQ3: Which semantic interoperability challenges are currently being tackled in the IoT domain?

Foundation: For answering the third research question, the textual descriptions of research challenges (I5) are tackled by the authors.

Among others, these are 1) use-case centric IoT solutions 2) formalization and use of domain knowledge 3) integration of explicit and implicit conceptual schemata and 4) reuse of existing standards. In the following, we group challenges mentioned by the study authors in the primary publications grouped per year (see Table G).

Table G: Overview Challenges and Questions

Year (Amount of Papers)	Challenges/Questions
2002 (1)	Full automation of dynamic composition process
2008 (1)	Dynamic and adaptive aspects of services
2009 (1)	Intuitive user application requests
2010 (1)	Interoperability in Information exchange
2011 (1)	Semantic representation
2012 (2)	Principled approach for automated synthesis of service mediators; Automatically composing Web-APIs
2013 (2)	Generate custom adapters based on convenient specification;
2014 (7)	Gap between semantic Representations; IoT Interoperability, Common way to abstract device heterogeneity; Automated service discovery and matching; Composition concerns; Creation of services by non-technical users; Interoperability problems experienced by user
2015 (7)	Interoperability in Software Engineering; Broker flexibility in matching process; Scalability; Thing discovery; System provisioning generating interoperable IoT-Applications; Amount of service dependencies in service mashups; Lack of separation of concerns and abstractions
2016 (12)	Heterogeneity of information; Descriptions in evolving systems; Interoperability of heterogenous devices; Semantic Query Integration;
2016 (12)	Standardization; Complexity and processing time of semantic technologies; Data Integration; Support of functional- and non-functional interoperability; Flexibility vs. Responsiveness; Sharing of ontology matching results; Semantic relation computing;
2017 (10)	Data annotation performance; Relationships between standards; Minimizing human intervention; Service discovery; Data integration; Integration of platforms; Device Interoperability; Semantic Orchestration; Manual client adjustment in middleware;
2018 (13)	Plug-and-Play principle; Integrate IoT devices without halting the system; Representation of features; Service Heterogeneity, Cloud as a single point of failure; Interoperability between entities; System behavior adjustment based on user goals; Lack of formalization; Semantic extensions; Thing integration; Construction complexity of ontologies; Conventional AI developed for specific application
2019 (4)	Agent-based limitations such as heterogenous vocabulary, use of specific ontologies, use of non-standardized artefacts, inaccurate discovery results; User-friendly semantic language; User-device Interoperability

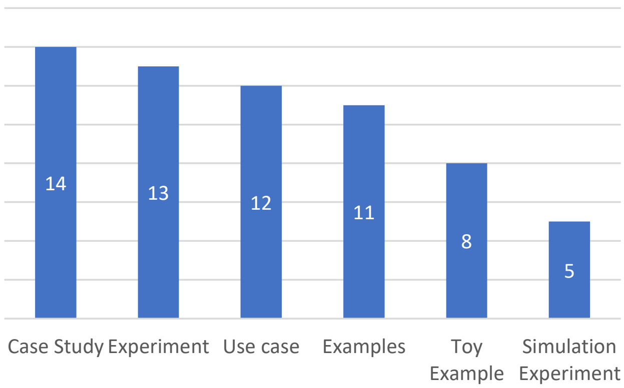
It is worth noticing, that capturing service and data semantics is regarded as a challenge. This observation is in alignment to the results from research question 1 where capturing and representation of knowledge (i.e. annotation activities) is the most performed activity to achieve semantic interoperability. Furthermore, over the years of 2002 until 2019 the common interoperability challenges such as heterogeneity due to decentralized development processes, data and query integration, (lack of) standardization, adapter generation and representation of semantic knowledge prevail.
In addition, during the last years beginning from 2017, a few publications deal with the user as an active part of IoT-Systems. Here (non-technical) users are assigned knowledge-related architecture activities that have been prior executed by domain experts.

E. RQ4: How are semantic interoperable IoT methods and architectures evaluated?

Foundation: For answering the last research question, the selected evaluation strategies (I6) are categorized and compared.

In Graphic E, an overview of the six most denoted experiment types by the authors are visualized. In total, all found primary studies did specify some type of evaluation. This can be interpreted as a correct result for the applied inclusion and exclusion criteria.
However, the denomination among the evaluation types differs significantly between publications. The most stated evaluation type was “Case Study” with 15 publications. Next, authors utilized “Experiments”, “Use cases”, “Examples”, “Toy Examples” and “Simulation Experiments”. All other evaluation types such as Hackathons, Interviews or Comparisons were only used once. It should be noted that there was no similarity comparison between evaluation types. Consequently, toy

example is listed as a distinct type next to example although toy example can be viewed as a hierarchical subclass of example.



Graphic E: Evaluation Settings Overview

During data extraction, the evaluation goal, the independent as well as the dependent variable were extracted. Here, the independent variables are seen as a lever to influence other variables. We denote variables that are influenced as the dependent variable. Such variables are explicitly measured and supervised to reflect a (un-) desired effect of an intervention. Whenever quantitative measurements were provided within the publication, the independent and the dependent variables were manually extracted. This information is displayed in Table H. Overall, Proof-of-Concept was the most stated goal (33 publications) followed by System Performance (11 publications). Both evaluation types mostly include a technical implementation.

Table H: Evaluation Goal and measured variables

Evaluation Goal	Independent Variable	Dependent Variable
Proof-of-Concept	Mapping-Levels	Execution time
	Measurement Runs	Time per automated activity
	Message size	CPU utilization, Energy consumption
	Query Execution	Query Response Time
	Amount of data, amount of service descriptions	Round trip latency, average latency
	Transferred data items	Latency and idle time
	Number of sensors	Processing time, Query processing time
System Performance	Event detection algorithm, Number of broker nodes, Event arrival rate	Event detection accuracy, search delay, response time
	Algorithms	Precision, Recall, Term entropy, document entropy
	Rules	Rule execution time
	Number of ontology instances, number of ontology classes	Query processing time
	Protocol adapter	Protocol translation time, time for adapter synthesis, development effort
	Number of dependencies	Time for parsing and reasoning
	Protocols	Queue time
	Number of users, request payload, number of requests	Response time, Technical Error rate
Other	Heterogeneity types	Precision, Recall
	Subscriptions	Time to insert new triples
	Engineering Methods	Engineering time, subjective evaluation of ease of use, time and satisfaction

Other evaluation types that were used were qualitative assessments, comparisons & benchmarks and query engine evaluations.

Another point of view that might be interesting are empirical evaluations or evaluations that include at least some human interactions. From 75 publications, 60 publications did not include any human participation within their evaluation strategy. Only one publication (P36) did include an empirical evaluation and only one other publication (P3) did provide explicitly insights within the engineering process of IoT-Systems with focus on semantic interoperability. Activities performed by humans within evaluations are:

- Using the system under discussion (P67, P25, P57, P52)
- Programming / Developing applications (P38)
- Query Formulation (P11, P1)
- Solve Task with help of system (P3, P17)
- Searching for interfaces (P59)
- Using composition languages (P21)

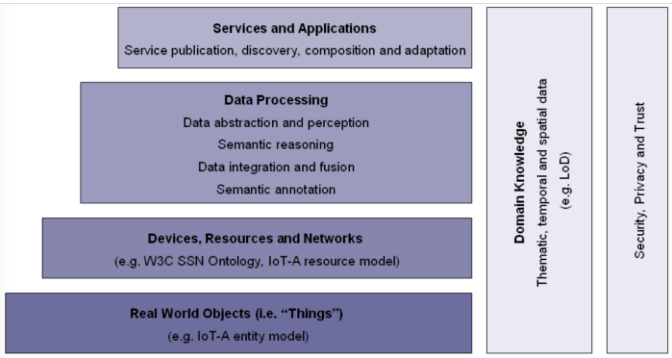
The remaining publications used qualitative evaluation strategies. The presented solutions were showcased using illustrative concepts (e.g. toy examples based on a technical foundation), examples using a proposed language or other evidence.

F. Result Interpretation and Open Research Challenges

In order to derive implications from our results and discuss the state-of-open-research-challenges, our findings can be conceptually integrated within well-known adjacent surveys.
In 2012, Barnaghi et al. [44] published their survey about “Semantics in the Internet of Things”. The authors present the progress so far and pose the following research challenges for IoT-Systems:

- Dynamicity and complexity
- Scalability
- Semantic service computing for IoT
- Distributed data storage and query
- Quality, trust and reliability of data
- Security and privacy
- Interpretation and perception of data

Especially the research challenges dynamicity and complexity, semantic service computing for IoT and Interpretation and perception of data can be seen relevant for semantic interoperability methods and approaches as discussed in this survey.
Furthermore, Barnaghi et al. [44] introduced "semantics" at different levels in IoT (see Graphic F).



Graphic F: Semantics at different IoT levels
(adapted from Barnaghi et al. [44])

Again, activities identified at different levels of abstractions can be syntactically matched to activities that are also discussed in this survey. Especially while answering research questions 1 and 2. For example, recent approaches to achieve semantic interoperability also use semantic data annotations, the Semantic Sensor Network Ontology (SSN) or service discovery and composition principles.
From a methodical viewpoint, Hamzei and Navimipour [45] surveyed in "Efficient Service Composition Techniques in the Internet of Things" (published in 2018) different methods that enable service composition. They categorize their findings into framework-based, heuristic-based, model-based and service-oriented architectures and RESTful-based approaches. Within their result set they found methods dealing with scalability, execution time and reliability (Top 3). These properties can also be spotted in the answers regarding research question 4.
Overall, Barnaghi et al. [44] explicitly focuses on identifying semantics on different IoT levels and Hamzei and Navimipour [45] concentrate on service composition techniques without a semantic focus.
As a consequence, research challenges arising from achieving semantic interoperability between applications, system architecture and IoT devices from a methodical viewpoint are still missing. Therefore, multiple findings per research question will be stated. Based on these findings, themes for open research challenges will be rendered.

RQ1: Which semantic interoperability approaches are currently studied in the context of IoT?

- Semantic Interoperability is either tackled from a service or data point of view. This is in accordance to previous works.
- From a service viewpoint, description, discovery and annotation of services are performed the most. From a data viewpoint, the activities annotation, semantic model definition and data description are leading. Hence, these activities are (still) an active area of research for achieving semantic interoperability within the IoT domain.
- Reusing standards and established semantic technologies/tools is performed heavily. However, they are utilized in a highly diverse manner as multiple standards at different levels of abstractions exist for identical scenarios.

RQ2: Which knowledge-based activities do integration approaches tackle and when are they reified?

- Capturing and Representation is the most performed knowledge-related activity. This can be related to standardization efforts on different levels of abstractions. On the one hand, most approaches found assume necessary semantic models to be available. On the other hand, most approaches also adapt these standards to fulfill their needs.
- Most semantic interoperability approaches are applied either during design time or runtime of an IoT-System. As a consequence, certain semantic properties are either built-in into a system or are expected to be fulfilled by IoT Things.

RQ3: Which semantic interoperability challenges are currently being tackled in the IoT domain?

- Most approaches that want to achieve semantic interoperability rely on model-based (e.g. based on first-order logic) mechanisms. Heuristic approaches such as matching approaches or framework-based application development are rare. For example, deep learning techniques (heuristic-based) are not found for integration issues. In contrast, web interfaces such as (RESTful or HTTP/JSON) are studied more intensively but do not incorporate semantics in their native form. To extend such interfaces, service or interface descriptions languages (e.g. RESTdesc, RAML or WSDL) are used.
- Semantics defined by the system user and not by third parties became a recent research focus. However, due to formal language complexity and required skill-set by users, such approaches are still in their infancy regarding semantic interoperability approaches.

RQ4: How are semantic interoperable IoT methods and architectures evaluated?

- Empirical research results on how to incorporate semantics with their accompanying characteristics such as evolutionary or recent domain-specific adaptations are scarce. Although search strings placed a specific focus on the methodical aspects of semantic interoperability, research contributions are (still) being evaluated mostly regarding their system performance and tend to be rather fully automated and not incremental. This stands in contrast to IoT-System evolutions during runtime as such systems tend to be "always on, always available and always connected" and are conceptually affected by fast innovation lifecycles.
- One reason for performing more system-related performance evaluation may be the academic setting in which semantic interoperability approaches are tested. However, adaptability between applications and available things within their distinct installation context by humans is strongly needed (e.g. within a smart home).

Based on our findings, the following research challenges for achieving semantic interoperability from a methodical viewpoint are proposed:

- Standardization-driven, top-down engineering processes that incorporate semantic interoperability are suited for well-defined and mostly static domains. However, fast innovation cycles for technologies require also bottom-up mechanisms to incorporate semantics in an incremental way.
- Human involvement in research experiments seem to be a complex task. Such experiments are strongly needed for defining semantics at runtime as IoT-Device manufacturer and application developer may not know which use-cases exist for each user at design time.
- There exists a trade-off between formalization effort and the degree of automating system engineering activities. Here, new approaches are needed that are flexible enough to incorporate changes in domain-specific semantics easily and being automated as much as possible.
- In comparison to heuristic-based artificial intelligence approaches (e.g. deep neural nets), model-based reasoning techniques are currently not regarded as a hot topic in computer science. Nevertheless, they are necessary for achieving reliable semantic interoperability as probabilistic matching results are not acceptable for IoT actuators. Making such models usable to the non-technical users will be challenging.

IV. Discussion

In this section, a critical discussion regarding the survey scope, study quality assessment and threats to validity is outlined.

A. Scope of Systematic Review

According to the PICO criteria, we scoped this systematic review towards the term “Internet-of-Things”, “Methods” and “Semantic Interoperability”. Although semantic interoperability has been handled rather as a property instead of a first-class citizen in the past (e.g. web service composition approaches or searching for software libraries), several methods for these properties exist. We explicitly allowed all applications of basic research results on the system class of IoT. We acknowledge that the term “Internet of Things” is vague and ambiguous. However, this term has indeed many facets and needs to be looked at from different angles. For gaining focus from an architectural viewpoint, we explicitly required the commonly used term combinations Semantic “Integration”, “Semantic Interoperability” and “Semantic Computing”. As we are not interested in reference architecture, we chose the words “Method”, “Approach” or “Technique” to further tie the review scope down. The rational for doing so lies within the semantic term concepts. As we wanted to explicitly exclude purely standard-based integration mechanisms where each party acts accordingly to one globally defined standard, several integration steps are naturally necessary for integration at different levels of abstractions (see Graphic F). However, the usage of “approach” and “technique” is ambiguous in the different publications found.

Overall, the amount of publications found when issuing the search strings indicates, that the scope is not too broad (75 final primary studies found)

B. Study Quality Assessment

By using inclusion and exclusion criteria, non-scientific publications such as market studies by companies could be excluded successfully. When choosing whether a publication makes it to the list of primary studies, the following questions served as a guideline:

- What is the semantic emphasis of the proposed approach?
- Is there are clear statement how the knowledge-related approach is evaluated?
- Does the evaluation contain a technical grounding?
- Is there a clear statement which knowledge-related contribution is being made?
- Which semantic interoperability challenge is being tackled on which level?

When more than two of these questions could not be clearly answered by the data extractor, the respective publication was dismissed.

Finally, the content of text fields is arguably prone for interpretation.

C. Validity Threats

Although interpretative activities can be minimized as much as possible during the data extraction process, there exist validity threats to every systematic literature review. In the following, we provide comments to conclusion validity, internal and external validity and construct validity

Conclusion validity: Conclusion validity or study reliability focuses on whether a third person would produce similar results when the study is reproduced [46]. Regarding the data extraction process, two PhD students independently extracted the data based on a structured template. This template was incrementally designed and pre-tested within a pilot study by a master student. Here, the inclusion/exclusion criteria were tuned in order to arrive at unambiguous criteria. In addition, one experienced researcher from the domain of software engineering supervised all implications drawn from the presented results. All relevant processual steps have been formalized using a review protocol. All data extraction mechanisms can be viewed within the data extraction template.

Internal validity: Internal validity focuses on the study design and whether findings really follow from the data [46]. For improving the internal validity, we performed random sampling on a subset of primary publications during data extraction and discussed potential differences in data extraction results and process. After the pilot study, we also dismissed two columns that where highly subjective (i.e. Adaptability of the interoperability model by the user and Need for custom Adaptation of the interoperability model). The overall threat to internal validity is further minimized by using many descriptive results and generic search processes.

External validity: External validity states whether the extracted data set can be used for generalization [46]. We applied various validity mechanisms such as automatic and manual search steps in relevant publication venues, forward-snowballing and reference checking, extraction of research type, method and venue, PICO criteria and definition rules for study identification. All other validity mechanisms can be reviewed within the data extraction protocol.

Construct validity: Construct validity focuses on whether the theoretical concepts used within this study are interpreted and measured correctly [46]. The presented knowledge-related activities are grounded on well-defined theoretical concepts [39]. Furthermore, the chosen lifecycle phases are regarded as applicable as they are used in different well-known software engineering approaches (e.g. Rational Unified Process). To ensure an identical interpretation of knowledge as well as system lifecycle phases, all involved researchers have been introduced to the overall aim of this study and differences interpretations were clarified before proceeding with study-related activities. All terms have been further checked by an experienced researcher from the field of software engineering. However, as the term "Internet of Things" is used in many research communities, we might have missed publications that were not published within a computer science venue (e.g. Management Information Systems Quarterly MIS). For example Böhmann et al. discuss in their work "Service System Engineering" [47] a possible research agenda from an Information Systems Research frame of reference. From our point-of-view, this does not affect our system architecture focus directly but might be interesting for empirical studies regarding semantic interoperability solution evaluation.

D. *Scientific Survey Impact*

The offered research challenges as well as the findings provided in sections Result Interpretation and Open Research Challenges disclosed various underexplored research areas.

- Little work has been done regarding empirical evaluations of semantic interoperability approaches. Although there exist user studies working with semantic data integration such as [48], there are few publications that actively deal with user involvement in IoT service systems. Especially in user owned IoT systems the semantics of data, services and applications is mainly driven by the user's interpretation and not by a predefined standard.
- Based on the provided results, practitioners can easily identify semantic interoperability approaches based on performance criteria. This might help to further improve the presented approach by guiding the scientific community with feedback from practical application scenarios.
- There exists a significant body of knowledge of how to apply model-based techniques for semantic interoperability within the software engineering and architecture community. However, authors mostly use logic-based languages as a means to an end (i.e. achieving semantic interoperability by using formalized ontologies). Here, the artificial intelligence community, especially logic-based approaches, seem to be a promising direction for research collaboration. In such a setting, extended reasoning techniques could be applied to further automate semantic component composition activities without losing description flexibility.
- Although knowledge stores such as semantic databases are applied often when conducting semantic interoperability research, little is known on how to compute semantic relations and/or semantic similarity measures between use-cases. Currently, semantic similarity measures are either based on syntactic similarity (e.g. heuristic approaches) and/or using system-wide defined standards or a combination of both. However, the semantic similarity between use cases used in various IoT applications is currently underexplored. For example, service subsets are used within heterogenous smart home applications in different ways. For example, a solar panel on a roof can be used for both, triggering an automation rule that turns on the washing machine or triggering an automation rule that sells electricity to an electricity company. Such semantic relations between use-case application and IoT device are currently not being utilized. Hence, more research is needed that focus on a methodical application of technology instead of applying technology somehow.

V. Conclusion

In this systematic literature review, we investigated the research question how IoT-Systems engineering processes can be supported to achieve semantic interoperability between applications, services and software platforms. In contrast to other surveys about semantic interoperability research, we focus on semantic interoperability achieved by applying well-defined methods. A key aspect of achieving semantic interoperability within IoT-Systems is the usage of knowledge-related activities for composing data, services and applications. Therefore, we extracted 75 primary studies from the field of IoT and analyzed them according to their contribution, evaluation strategy, knowledge activity and application phase. In this review, we find an increasing number of publications that deal with semantic interoperability from a methodical viewpoint. Most contributions do rely on predefined semantic standards and further adapt them to their specific needs. However, most studies also evaluate semantic interoperable systems with performance-related metrics. Although efficiency aspects of smart IoT service systems are necessary, their effectiveness regarding their every-day additional value cannot be neglected. We encourage the IoT research community to perform more (empirical) research for achieving smart service systems that go beyond dashboards for IoT sensors and actuators which currently provide only isolated states and act like boolean switches.

References

[1] X. Larrucea, A. Combelles, J. Favaro, and K. Taneja, "Software Engineering for the Internet of Things," *IEEE Softw.*, vol. 34, no. 1, pp. 24–28, Jan. 2017.

[2] T. Vale, I. Crnkovic, E. S. de Almeida, P. A. da M. Silveira Neto, Y. C. Cavalcanti, and S. R. de L. Meira, "Twenty-eight years of component-based software engineering," *J. Syst. Softw.*, vol. 111, pp. 128–148, Jan. 2016.

[3] A. L. Lemos, F. Daniel, and B. Benatallah, "Web Service Composition: A Survey of Techniques and Tools," *ACM Comput Surv*, vol. 48, no. 3, pp. 33:1–33:41, Dec. 2015.

[4] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, and M. A. Babar, "10 years of software architecture knowledge management: Practice and future," *J. Syst. Softw.*, vol. 116, pp. 191–205, Jun. 2016.

[5] A. Bouguettaya *et al.*, "A Service Computing Manifesto: The Next 10 Years," *Commun ACM*, vol. 60, no. 4, pp. 64–72, Mar. 2017.

[6] A. Rausch, C. Bartelt, S. Herold, H. Klus, and D. Niebuhr, "From Software Systems to Complex Software Ecosystems: Model- and Constraint-Based Engineering of Ecosystems," in *Perspectives on the Future of Software Engineering*, Springer, Berlin, Heidelberg, 2013, pp. 61–80.

[7] D. Garlan, R. Allen, and J. Ockerbloom, "Architectural mismatch: why reuse is so hard," *IEEE Softw.*, vol. 12, no. 6, pp. 17–26, Nov. 1995.

[8] D. Garlan, R. Allen, and J. Ockerbloom, "Architectural Mismatch: Why Reuse Is Still So Hard," *IEEE Softw.*, vol. 26, no. 4, pp. 66–69, Jul. 2009.

[9] H. Abukwaik, D. Taibi, and D. Rombach, "Interoperability-Related Architectural Problems and Solutions in Information Systems: A Scoping Study," in *Software Architecture*, 2014, pp. 308–323.

[10] I. Crnkovic, S. Sentilles, A. Vulgarakis, and M. R. V. Chaudron, "A Classification Framework for Software Component Models," *IEEE Trans. Softw. Eng.*, vol. 37, no. 5, pp. 593–615, Sep. 2011.

[11] I. Crnkovic, I. Malavolta, H. Muccini, and M. Sharaf, "On the Use of Component-Based Principles and Practices for Architecting Cyber-Physical Systems," in *2016 19th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE)*, 2016, pp. 23–32.

[12] H. Muccini and M. Sharaf, "CAPS: Architecture Description of Situational Aware Cyber Physical Systems," in *2017 IEEE International Conference on Software Architecture (ICSA)*, 2017, pp. 211–220.

[13] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[14] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutor.*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[15] V. Gazis, "A Survey of Standards for Machine-to-Machine and the Internet of Things," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 1, pp. 482–511, Firstquarter 2017.

[16] C.-M. Chituc, "Interoperability Standards for Seamless Communication: An Analysis of Domain-Specific Initiatives," in *On the Move to Meaningful Internet Systems: OTM 2016 Workshops*, 2017, pp. 36–46.

[17] N. Schmidt, A. Lüder, R. Rosendahl, D. Ryashentseva, M. Foehr, and J. Vollmar, "Surveying integration approaches for relevance in Cyber Physical Production Systems," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, 2015, pp. 1–8.

[18] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things From Industrial Market Perspective," *IEEE Access*, vol. 2, pp. 1660–1679, 2014.

[19] C. Perera, C. H. Liu, and S. Jayawardena, "The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey," *IEEE Trans. Emerg. Top. Comput.*, vol. 3, no. 4, pp. 585–598, Dec. 2015.

[20] K. J. Singh and D. S. Kapoor, "Create Your Own Internet of Things: A survey of IoT platforms.," *IEEE Consum. Electron. Mag.*, vol. 6, no. 2, pp. 57–68, Apr. 2017.

[21] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How Can Heterogeneous Internet of Things Build Our Future: A Survey," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 3, pp. 2011–2027, thirdquarter 2018.

[22] D. G. Korzun, S. I. Balandin, and A. V. Gurtov, "Deployment of Smart Spaces in Internet of Things: Overview of the Design Challenges," in *Internet of Things, Smart Spaces, and Next Generation Networking*, 2013, pp. 48–59.

[23] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Commun. Surv. Tutor.*, vol. 16, no. 1, pp. 414–454, First 2014.

[24] W. Ding, P. Liang, A. Tang, and H. van Vliet, "Knowledge-based approaches in software documentation: A systematic literature review," *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 545–567, Jun. 2014.

[25] Z. Li, P. Liang, and P. Avgeriou, "Application of knowledge-based approaches in software architecture: A systematic mapping study," *Inf. Softw. Technol.*, vol. 55, no. 5, pp. 777–794, May 2013.

[26] B. Ramis *et al.*, "Knowledge-based web service integration for industrial automation," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 733–739.

[27] G. Bajaj, R. Agarwal, P. Singh, N. Georgantas, and V. Issarny, "4W1H in IoT Semantics," *IEEE Access*, vol. 6, pp. 65488–65506, 2018.

[28] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things: Taxonomies and Open Challenges," *Mob. Netw. Appl.*, Jul. 2018.

[29] H. Muccini and M. T. Moghaddam, "IoT Architectural Styles," in *Software Architecture*, 2018, pp. 68–85.

[30] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A Survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[31] A. Y. Teka, N. Condori-Fernandez, and B. Sapkota, "A Systematic Literature Review on Service Description Methods," in *Requirements Engineering: Foundation for Software Quality*, 2012, pp. 239–255.

[32] S. Evdokimov, B. Fabian, S. Kunz, and N. Schoenemann, "Comparison of Discovery Service Architectures for the Internet of Things," in *2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2010, pp. 237–244.

[33] I. Szilagyi and P. Wira, "Ontologies and Semantic Web for the Internet of Things - a survey," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6949–6954.

[34] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective," *J. Netw. Comput. Appl.*, vol. 81, pp. 111–124, Mar. 2017.

[35] I. Szilagyi and P. Wira, "Ontologies and Semantic Web for the Internet of Things - a survey," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6949–6954.

[36] J Rowley, "The wisdom hierarchy: representations of the DIKW hierarchy," *J. Inf. Sci.*, vol. 33, no. 2, p. 163.

[37] J. Euzenat, "Towards a principled approach to semantic interoperability," presented at the Proc. IJCAI 2001 workshop on ontology and information sharing, 2001, pp. 19–25.

[38] D. Harel and B. Rumpe, "Meaningful modeling: what's the semantics of 'semantics'?," *Computer*, vol. 37, no. 10, pp. 64–72, Oct. 2004.

[39] M. Alavi and D. E. Leidner, "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Q.*, vol. 25, no. 1, pp. 107–136, 2001.

[40] A. Tang, P. Avgeriou, A. Jansen, R. Capilla, and M. Ali Babar, "A comparative study of architecture knowledge management tools," *J. Syst. Softw.*, vol. 83, no. 3, pp. 352–370, Mar. 2010.

[41] P. Liang and P. Avgeriou, "Tools and Technologies for Architecture Knowledge Management," in *Software Architecture Knowledge Management: Theory and Practice*, M. Ali Babar, T. Dingsøyr, P. Lago, and H. van Vliet, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 91–111.

[42] P. W. Stone, "Popping the (PICO) question in research and evidence-based practice," *Appl. Nurs. Res. ANR*, vol. 15, no. 3, pp. 197–198, Aug. 2002.

[43] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?," *Int. J. Hum.-Comput. Stud.*, vol. 43, no. 5, pp. 907–928, Nov. 1995.

[44] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things: Early Progress and Back to the Future," *Int. J. Semantic Web Inf. Syst. IJSWIS*, vol. 8, no. 1, pp. 1–21, Jan. 2012.

[45] M. Hamzei and N. J. Navimipour, "Toward Efficient Service Composition Techniques in the Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3774–3787, Oct. 2018.

[46] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting Empirical Methods for Software Engineering Research," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. London: Springer London, 2008, pp. 285–311.

[47] T. Böhmann, J. M. Leimeister, and K. Möslein, "Service Systems Engineering," *Bus. Inf. Syst. Eng.*, vol. 6, no. 2, pp. 73–79, Apr. 2014.

[48] H. Stuckenschmidt, J. N. sner, and F. Fallahi, "A Study in User-centric Data Integration," in *ICEIS*, 2012.

Study ID	Title	Authors	Publication year
P1	A comprehensive semantic model for smart object description and	Yachir, Ali et al.	2016
P2	A Lightweight Semantic Web-based Approach for Data Annotation on IoT Gateways	Al-Osta, Mahmud; Ahmed, Bali; Abdelouahed, Gherbi	2017
P3	A novel I4.0-enabled engineering method and its evaluation	Prinz, Frederick et al.	2019
P4	A Semantic Mechanism for Internet-of-Things (IoT) to Implement Intelligent Interactions	You, S.; Li, X.; Chen, W.	2018
P5	A Semantic Processing Framework for IoT-Enabled Communication Systems	Ali, Muhammad Intizar et al.	2015
P6	A Semantic-aware Framework for Service Definition and Discovery in the Internet of Things Using CoAP	Khodadadi, Farzad; Sinnott, Richard O.	2017
P7	Achieving functional and non functional interoperability through synthesized connectors	Nostro, Nicola et al.	2016
P8	Achieving Interoperability through Semantics-Based Technologies: The Instant Messaging Case	Bennaceur, Amel et al.	2012
P9	Adaptable Interfaces, Interactions, and Processing for Lined Data Platform Components	Keppmann, Felix Leif; Maleshkova, Maria; Harth, Andreas	2017
P10	Adapting Heterogeneous Devices into an IoT Context-Aware Infrastructure	Pötter, H. B.; Sztajnberg, A.	2016
P11	An Architecture for Interoperable IoT Ecosystems	Schmid, Stefan et al.	2017
P12	An extensible and active semantic model of information organizing for the Internet of Things	Sun, Yunchuan; Jara, Antonio J.	2014
P13	Annotating Real-world Objects using semantic Entities	Hasemann, Henning et al.	2013
P14	Approach for Dynamically Composing Decentralised Service Architectures with Cross-Cutting Constraints	Myllärniemi, Varvana et al.	2008
P15	Approach for Semantic Interoperability Testing in Internet of Things	Datta, S. K. et al.	2018
P16	Artificial Intelligence-Based Semantic Internet of Things in a User-Centric Smart City	Guo, Kun; Lu, Yueming; Gao, Hui; Cao, Ruohan	2018
P17	Assisting IoT Projects and Developers in Designing Interoperable Semantic Web of Things Applications	Gyrard, A.; Bonnet, C.; Boudaoud, K.; Serrano, M.	2015
P18	Automated Sensor Registration, Binding and Sensor Data Provisioning	Hirmer, Pascal et al.	2016
P19	Automated Synthesis of Mediators to Supprt Component Interoperability	Bennaceur, A.; Issarny, V.	2015
P20	Connecting IoT Sensors to knowledge-based Systems by transforming SenML to RDF	Su, Xiang et al.	2014
P21	DAMASCo: A Framework for the Automatic Composition of Component-Based and Service Oriented Architectures	Cubo, Javier; Pimentel, Ernesto	2011
P22	Design of Ensemble-based Component Systems by Invariant Refinement	Keznikl, Jaroslav et al.	2013
P23	Domain-specific diagrammatic modelling: a source of machine-readable semantics for the Internet of Things	Buchmann, Robert Andrei; Karagiannis, Dimitris	2017
P24	Enabling high-level application development for the Internet of Things	Patel, Pankesh; Cassou, Damien	2015
P25	Enabling Semantics in Enterprises	Pomp, André et al.	2018
P26	Functional Composition of Sensor Web apis	Verborgh, Ruben et al.	2012
P27	Generic Driver Injection for Automated IoT Application Deployments	Saatkamp, Karoline et al.	2017
P28	GrOWTH: Goal-Oriented End User Development for Web of Things Devices	Noura, Mahda; Heil, Sebastian; Gaedke, Martin	2018
P29	Integrating Service Matchers into a Service Market Architecture	Platenius, Marie Christin; Becker, Steffen; Schäfer, Wilhelm	2014
P30	Interoperability in semantic Web of Things: Design issues and solutions	Silva, André Luis Meneses et al.	2019
P31	IoT Semantic Interoperability with Device Description Shapes	Thuluva, Aparna Saisree; Anicic, Darko; Rudolph, Sebastian	2018
P32	IoT-Based User-Driven Service Modeling Environment for a Smart Space Management System	Choi, Hoan-Suk; Rhee, Woo-Seop	2014
P33	IoT-DDL–Device Description Language for the "T" in IoT	Khaled, A. E.; Helal, A.; Lindquist, W.; Lee, C.	2018
P34	IoT-Lite: A Lightweight Semantic Model for the Internet of Things	Bermudez-Edo, M.; Elsaleh, T.; Barnaghi, P.; Taylor, K.	2016
P35	IoT-O, a Core-Domain IoT Ontology to Represent Connected Devices Networks	Seydoux, Nicolas et al.	2016
P36	Knowledge Graphs for Semantically Integrating Cyber-Physical Systems	Grangel-González, Irlán et al.	2018
P37	Linking Health Web Services as Resource Graph by Semantic REST Resource Tagging	Peng, Cong; Goswami, Prashant; Bai, Guohua	2018
P38	MatchBox: A Framework for Dynamic Configuration of Service Matching Processes	Platenius, Marie C.; Schäfer, Wilhelm; Arifulina, Svetlana	2015
P39	Ontology-Based Semantic Modeling and Evaluation for Internet of Things Applications	Ma, M.; Wang, P.; Chu, C.	2014
P40	Practical semantics for the Internet of Things: Physical states, device mashups, and open questions	Kovatsch, M.; Hassan, Y. N.; Mayer, S.	2015

P41	PROFICIENT: Productivity Tool for Semantic Interoperability in an Open IoT Ecosystem	Kolbe, Niklas; Robert, Jérémy; Kubler, Sylvain; Traon, Yves Le	2017
P42	RedSib: A smart-M3 semantic information broker implementation	Morandi, F.; Roffia, L.; D'Elia, A.; Vergari, F.; Cinotti, T. S.	2012
P43	Responsive Decentralized Composition of Service Mashups for the Internet of Things	Ciortea, Andrei et al.	2016
P44	Self-Commissioning Industrial IoT-Systems in Process Automation: A Reference Architecture	Koziolek, H.; Burger, A.; Doppelhamer, J.	2018
P45	Semantic Context-Aware Service Composition for Building Automation System	Han, S. N.; Lee, G. M.; Crespi, N.	2014
P46	Semantic Interoperability Architecture for Pervasive Computing and Internet of Things	Kiljander, J. et al.	2014
P47	Semantic Interoperability as Key to IoT Platform Federation	Jacoby, Michael et al.	2017
P48	Semantic Interoperability in Heterogeneous IoT Infrastructure for Healthcare	Jabbar, Sohail et al.	2017
P49	Semantic Matching of Engineering Data Structures	Kovalenko, Olga; Euzenat, Jérôme	2016
P50	Semantic relation computing theory and its application	Sun, Yunchuan; Lu, Cheng; Bie, Rongfang; Zhang, Junsheng	2016
P51	Semantic-based automatic service composition with functional and non-functional requirements in design time: A genetic algorithm approach	FanJiang, Yong-Yi; Syu, Yang	2014
P52	Semantics Based Service Orchestration in IoT	Chindenga, Edmore; Scott, Mfundo. S.; Gurajena, Caroline	2017
P53	Semantics-Based Composition of Factory Automation Processes Encapsulated by Web Services	Puttonen, J.; Lobov, A.; Lastra, J. L. Martinez	2013
P54	Semantics-based Context-aware Dynamic Service Composition	Fujii, Keita; Suda, Tatsuya	2009
P55	Semi-automatic Composition of Web Services using Semantic Descriptions	Sirin, Evren; Hendler, James; Parsia, Bijan	2002
P56	SEMIoTICS: Semantically Enhanced IoT-Enabled Intelligent Control Systems	Milis, G. M.; Panayiotou, C. G.; Polycarpou, M. M.	2019
P57	Sequential Behavioral Modeling for Scalable IoT Devices and Systems	Korkan, E.; Kaebisch, S.; Kovatsch, M.; Steinhorst, S.	2018
P58	SIGHTED: A Framework for Semantic Integration of Heterogeneous Sensor Data on the Internet of Things	Nagib, Ahmad M.; Hamza, Haitham S.	2016
P59	Simurgh: A framework for effective discovery, programming, and integration of services exposed in IoT	Khodadadi, F.; Dastjerdi, A. V.; Buyya, R.	2015
P60	Situation-Aware IoT Service Coordination Using the Event-Driven SOA Paradigm	Cheng, B.; Zhu, D.; Zhao, S.; Chen, J.	2016
P61	Smart-M3 and OSGi: The interoperability platform	Manzaroli, D. et al.	2010
P62	Specifying Semantic Interoperability between Heterogeneous Cloud Resources with the FCLOUDS Formal Language	Challita, S.; Zalila, F.; Merle, P.	2018
P63	SRE: Semantic Rules Engine for the Industrial Internet-Of-Things Gateways	Kaed, C. E. et al.	2018
P64	System integration by developing adapters using a database abstraction	Mooij, Arjan J.	2013
P65	Tabdoc approach: an information fusion method to implement semantic interoperability between IoT devices and users	Yang, S.; Wei, R.	2019
P66	The industry 4.0 standards landscape from a semantic integration perspective	Grangel-González, I. et al.	2017
P67	The MASSIF platform: a modular and semantic platform for the development of flexible IoT services	Bonte, Pieter et al.	2017
P68	Toward better horizontal integration among IoT services	Al-Fuqaha, A. et al.	2015
P69	Toward Plug Play Cyber-Physical System Components	Jirkovský, V.; Obitko, M.; Kadera, P.; Mařík, V.	2018
P70	Towards a Semantic Administrative Shell for Industry 4.0 Components	Grangel-González, I. et al.	2016
P71	Towards a Semantic Medical Internet of Things	Dridi, A.; Sassi, S.; Faiz, S.	2017
P72	Towards an Internet of Agents model based on Linked Open Data approach	Pico-Valencia, Pablo; Holgado-Terriza, Juan A.; Senso, J. A.	2019
P73	Towards Cooperative Semantic Computing: A Distributed Reasoning Approach for Fog-Enabled SWoT	Seydoux, Nicolas et al.	2018
P74	User Interoperability With Heterogeneous IoT Devices Through Transformation	Xiao, G.; Guo, J.; Xu, L. D.; Gong, Z.	2014
P75	Using Semantic Queries to Enable Dynamic Service Invocation for Processes in the Internet of Things	Huber, S.; Seiger, R.; Kühnert, A.; Schlegel, T.	2016