JOT
JOURNAL OF
OBJECT TECHNOLOGY

Journal of Object Technology | RESEARCH ARTICLE

Reusability and discovery models in software systems: a systematic literature review

Javier Criado and Luis Iribarne
Applied Computing Group, University of Almería, Spain

ABSTRACT Software Engineering principles and practices promote the reuse of existing elements to maximize the performance of development and maintenance processes. The core concepts of reusability are aligned with the FAIR (Findability, Accessibility, Interoperability, and Reusability) principles. From our perspective, FAIR principles are not only related to data in a software system, because there are other resources and parts of a system that are related to data indirectly or require data schemas and models for their description and management. Examples of these resources are services, business logic processes, contracts, security policies, architectures, components, deployment resources, execution rules, or interaction alternatives, among other possible examples. Actually, any resource in a system may need to be discovered either internally by a subprocess, or externally by any client that might be interested in finding, accessing, interoperating, or reusing it. This paper addresses a systematic literature review to analyze the current principles, paradigms, and technologies that can be applied to enable discovering and reusing mechanisms in modern software systems, and how traditional and outdated approaches have been adapted to support present issues. From this analysis, we outlined the challenges and open research lines that can be addressed to achieve a suitable reuse strategy.

KEYWORDS Discovery, Reusability, FAIR principles, SLR, Systematic Literature Review

1. Introduction

Software Engineering (SE) discipline aims at maximizing the reuse of software artifacts that have been previously developed (Sametinger 1997; Szyperski et al. 2002; Frakes & Kang 2005). From more informal techniques such as code copying and pasting, the construction of code templates, or the functionality encapsulation in libraries, reusing entire pieces of software was formalized in practices related to Component-Based Software Engineering (CBSE) (Vale et al. 2016).

CBSE represented a qualitative change in the traditional paradigm of software development, favoring bottom-up approaches through the integration of software pieces, i.e., components. Producers can publish their developed components in different types of directories or marketplaces, and consumers are able to discover, obtain and reuse them (Clark et al. 2004).

Component reuse requires particular management considerations in Service-Oriented software approaches. Services are used for exposing the component functionality, and their descriptions contain information related to the invocation interfaces, communication protocols, use policies, and contracts (Singh & Huhns 2005). Modern service-based systems are normally based on web services, and it is an additional layer of specific information that must be managed for enabling software reuse and discovery (Papazoglou et al. 2007).

Another modern domain related to software reuse but specially bountied by component and service discovery is Cyber-Physical Systems (CPS) (Jirkovský et al. 2018). In this domain, coexisting virtual and physical devices are managed by software components, and both hardware and software elements can be discovered to be used in different contexts, configurations or scenarios. Additionally, services providing their operations can benefit from the registration in a directory to allow their access both internally and externally.

JOT reference format:
Javier Criado and Luis Iribarne. *Reusability and discovery models in software systems: a systematic literature review.* Journal of Object Technology. Vol. 21, No. 4, 2022.
http://dx.doi.org/10.5381/jot.2022.21.4.a3

The first discovery approaches developed in the literature was based on Universal Description Discovery and Integration (UDDI) and Open Distributed Processing (ODP) models (Curbera et al. 2002; Iribarne et al. 2004), which have served as a reference for developing new models and being their functionality more straightforward and more limited. Subsequent proposals, such as Consul and Eureka (which are based on microservices (Jamshidi et al. 2018)), upgrade the functionality of the traditional discovery models, providing their operations through a RESTful API and expanding their capabilities, for example, through state control of the services offered.

The current evolution of the discovery model focuses on solving general problems, which affect multiple application domains, such as the efficient search for services that adapt to users' needs. Faced with such a problem in the field of modern software systems, the need arises for new types of discovery models. For example, CPS domain is focused on the search for elements belonging to the Internet of Things (IoT) and the Web of Things (WoT) (Pourghebleh et al. 2020; Sciullo et al. 2020), and related solutions try to facilitate their functionality through applications or interested entities (Cirani et al. 2014), and to reduce the number of requests made directly on the devices. Those with low energy capacity can increase their time of use (Tanganelli et al. 2018).

Discovery models in the field of IoT/WoT should also consider other essential aspects in CPS, such as interoperability, security, or heterogeneity of services, among other possible examples (Aziez et al. 2019). The 27050-1 standard (ISO/IEC 2019) describes electronic discovery, the process of discovering electronically stored information or data. The P1451.99 standard (IEEE SA 2020) defines a method for data sharing, interoperability, and security of messages over a network, where sensors, actuators and other devices can interoperate, regardless of underlying communication technology. The 30118-1 standard (ISO/IEC 2021), also known as OCF Core Specification, is used for homogenizing functional interactions in the IoT domain, such as messaging, discovery, monitoring, and maintenance. More recently, the 23093-2 standard (ISO/IEC 2022) specifies the abstract class of a media thing, a component to construct the Internet of media things, and it includes the APIs to discover, connect and support transactions of media things.

Regardless of the application domain, a service discovery model offers, as a response, a set of services that satisfy the specifications established in some type of query. This set of services can be ordered, prioritised, and/or filtered to select the best alternative. Such selection may be set on criteria that consider user preferences, usage history, or other information of interest to the application domain. With this target, an existing option is to investigate the application of new paradigms and technologies to suggest the best alternatives during the discovery of services, such as recommender systems (Chan et al. 2012).

In another context, a specific type of reuse in software systems is data, information or knowledge reuse, and there are specific studies related to this area (Markus 2001; Ristoski & Paulheim 2016). Nevertheless, there are not many research papers applying data reuse techniques to service discovery, and the existing ones are focused on using the semantic information for improving description, search and composition of services (Xu & Yu-Shi 2016; Eshuis et al. 2016).

From our point of view, data management principles can play a key role in discovery models since any artifact to be discovered must first be described with a metadata or a data record. As a consequence, data management guidelines can be incorporated in service discovery for improving the description of the services, their storage, the search process and the reuse capabilities, among other possible examples. In this sense, FAIR (Findability, Accessibility, Interoperability, and Reusability) principles (Jacobsen et al. 2020; Shanahan et al. 2021), originally intended to the management and administration of scientific data, could improve the capabilities of a discovery model.

In this paper, we present a systematic literature review on the discovery models and approaches used in recent research work. The selected period covers the last 10 years (from January 2013 to May 2022) and the selected works are focused on discovery solutions in modern and web systems related to CPS. We also have centered the review in aspects related to FAIR principles or that, from our point of view, can support the findability, accessibility, interoperability, or reusability. The scope of the analysis is based on the following research questions:

(RQ1) Are traditional approaches of discovery models being used for current and real web environments or solutions? Have new discovery models been proposed in recent years?

(RQ2) What advantages and disadvantages do discovery models provide in web environments?

(RQ3) What new opportunities on discovery models have emerged from current domains, paradigms and technologies?

(RQ4) Are the FAIR principles a new opportunity applicable in software engineering solutions for discovery web resources that are not directly related to data management?

With the aim at addressing these research questions, the rest of the paper is organized as follows. Section 2 describes the research method performed for the systematic literature review. Section 3 presents the most relevant solutions extracted from the study and related to discovering and reusing in modern software systems. A discussion is provided in Section 4 including an analysis of the results, addressing the threats to validity and proposing the potential future research topics. Finally, Section 5 summarizes the conclusions of this research work, giving the answers to the identified research questions.

2. Method

The systematic literature review was addressed by considering some of the existing methodologies in software engineering (Petersen et al. 2008), (Petersen et al. 2015), (Kuhrmann et al. 2017). The systematic review is a well-organised step-by-step procedure consisting of studying the current literature in a given area to structure and analyse the state of research, identifying its limitations and gaps leading to new lines of work. The literature review procedure focused on determining a search plan for the most relevant scientific articles in the Web of Science database. This plan considered the following actions:

(a) Keywords: The set of keywords to undertake the queries was established, taking into account a prior classification and taxonomy of the field of study: perspectives of Discovery in software engineering.

(b) Query: The collection of queries to be performed on the repository was defined, taking into account the previously identified categories.

(c) Collection: The queries were executed considering some selection and exclusion criteria, filtering the candidate reference repositories at three levels: Web of Science Core Collection, Computer Science and Software Engineering.

(d) Extraction: From the repository of selected works, extraction of specific information of interest to be analysed in this study was carried out.

2.1. Keywords, categories and taxonomy

The relevant literature review considered the starting research questions and the scope of interest, analysing the different Discovery approaches in the software engineering literature for the development of software systems and mapping them to one of the FAIR principles, traditionally linked to the discipline of engineering and data science.

Our research interest in software engineering focuses on Cyber-Physical Systems in the area of the Internet of Things and the Web of Things. For this reason, two of the starting keywords were both “IoT” and “WoT”, and their full meanings are “Internet of Things” and “Web of Things”. The word "Discovery" was the focus of attention for the study, particularly discovery models and services.

In our view, modern approaches in software system development based on discovery perspectives must consider new trends in their design and implementation. Specifically, for this study, we have established four areas of interest or categories:

(a) Integration: This is one of the elementary principles of a discovery model. It should include important issues such as delegation, query splitting and merging, features associated with federated systems and clustering, and interoperability aspects for a discovery design.

(b) Intelligence: Novel discovery models should include intelligence-based aspects such as recommender systems, transformers, machine learning, or semantic elements such as semantic search and ontologies.

(c) Quality: Quality of service (QoS) is another fundamental factor in developing software systems. Since a discovery service has an associated data repository on which queries are made, aspects of data quality and quality of the knowledge or information handled by the service must also be considered.

(d) Accessibility: Discovery services should be designed to be easily located and accessible in a network. Therefore, it is essential to consider some aspects of service availability, *e.g.* in highly available systems, or reliability, fault tolerance, and security.

Starting from these four areas of interest for the study, the following keywords were considered to prepare the queries: “WoT”, “Web of Things”, “IoT”, “Internet of Things”, “Integration”, “Cluster*”, “Interoperability”, “Federat*”, “Intelligence”, “Machine Learning”, “Recommender Systems”, “Semantic*”, “Ontology”, “Ontologies”, “QoS”, “Quality of Service”, “Quality of Data”, “Quality of Information”, “Accessibility” and “Availability”.

For some of these words, possible options were considered based on the lexical root of the term, e.g. for the word "Cluster*" results were found for “Clustered” or “Clustering”, or for “Federat*”, “Federate”, “Federation”, “Federated” or “Federating”. Figure 1 shows a UML class diagram used to conceptually represent the taxonomy of the discovery service perspective according to the four preset categories and subcategories in the form of keywords. As shown in the figure, the Discovery categories realise each of the FAIR principles.

2.2. Querying

The queries were performed on the Web of Science database in the Computer Science Collection. For the preparation of the queries, the term “discovery” was considered a common part. In addition, for the interest of our study, the term “model” and “service” were also supposed to cover works in the literature on the Discovery model or Discovery service. The template used for the query was the Advanced Search. For this common part, the sentence was prepared to query the topics “discovery”, “model”, and “service” in the following form:

```
TS=(discovery) AND (TS=(model) OR TS=(service))
```

This part of the sentence was intended to cover other options in the literature closer to the target topics, such as “discovery model” or “discovery service”, but without using them directly in the sentence itself, as these expressions are more robust. In this way, the previous expression is more relaxed, allowing us to find options such as “discovery meta-model”, “discovery pull model”, and “discovery management service”, among many other options. Although the number of results obtained was much higher, as it is a more relaxed expression, many interesting works that would have been discarded have been covered.

Based on the keywords discussed in the previous section, the following queries shown in Table 1 were prepared, resulting in 13 bags of references.

In addition to the four main categories of study: Integration (Q3, Q4, Q5, Q6), Intelligence (Q7, Q8, Q9, Q10), Quality (Q11), and Accessibility (Q12, Q13); queries were made for papers related to the fields of IoT and WoT, areas of interest of our current research.

Although a single query could be prepared for each of the four categories, it was preferred to conduct the study in different subcategories for a more detailed analysis of the literature later.

2.3. Collection: filtering & selection

For filtering the references, the three-step methodology shown in Figure 2 was followed. It was carried out with a UML activity diagram using structured activities.

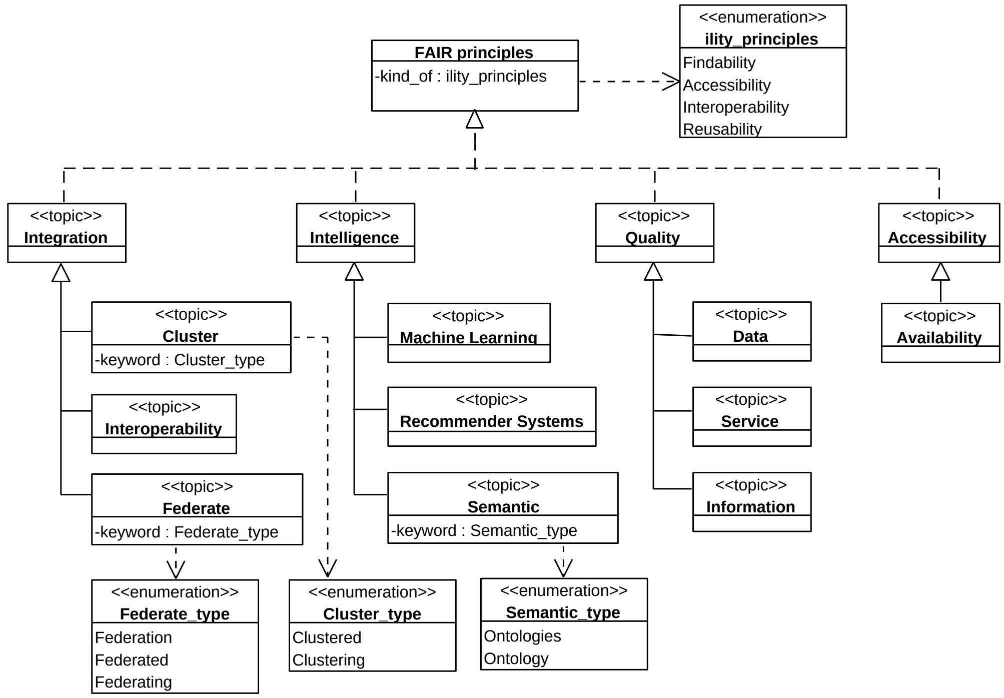

Figure 1 FAIR taxonomy for a discovery perspective.

Q	Query sentence
#1	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND (TS=(WoT) OR TS=(Web of Things))
#2	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND (TS=(IoT) OR TS=(Internet of Things))
#3	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Integration)
#4	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Cluster*)
#5	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Interoperability)
#6	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Federat*)
#7	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Intelligence)
#8	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Machine Learning)
#9	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Recommender Systems)
#10	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND (TS=(Semantic*) OR TS=(Ontologies) OR TS=(Ontology))
#11	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND (TS=(QoS) OR TS=(Quality of Service) OR TS=(Quality of Data) OR TD=(Quality of Information))
#12	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Accessibility)
#13	(TS=(discovery) AND (TS=(model) OR TS=(service))) AND TS=(Availability)

Table 1 List of query sentences considered.

As mentioned above, the queries were applied to the Web of Science Core Collection database in the first step. The first collection of references from the literature was made regardless of the year the work was conducted, obtaining 47076 publications among the four categories, represented as an output pin element of the activity diagram (small grey box). To simplify the picture, only the four datasets of the study categories have been shown. The results were then filtered to obtain those works from the last ten years (27708) and then the last five years. This was planned to analyze the evolution of the literature work over the time.

In the second and third stages, the same procedures were carried out, but now filtering by the Web of Science category in Computer Science, obtaining this time a total of 30436 papers, 16804 (10-Y) and 9958 (5-Y); and then by the topic Software Engineering, obtaining for this one a total of 1099 papers, 501 (10-Y) and 205 (5-Y).

The sets obtained are of the bag type, as there were papers that appeared in more than one category when querying the categories separately. To continue with the study, the dataset of the last ten years in software engineering was selected. In this case, mapping the collection as a set (non-repeating elements), the total number of papers obtained after filtering was 299.

As the last step, two manual filtering operations were carried out, considering, for the first one, should the target topics appear in the title or the abstract of the work, leaving a total of 113 results. On these, their content was analysed with a full reading to finally select 46 works for the study. The list of the selected papers can be accessed in Zenodo[1].

Table 2 shows some search results for each of the three stages (Web of Science Core Collection, Computer Science and Software Engineering) by year (All-Y, 5-Y, 10-Y) and each of the 13 subcategories.

[1] List of selected papers – https://doi.org/10.5281/zenodo.6939596

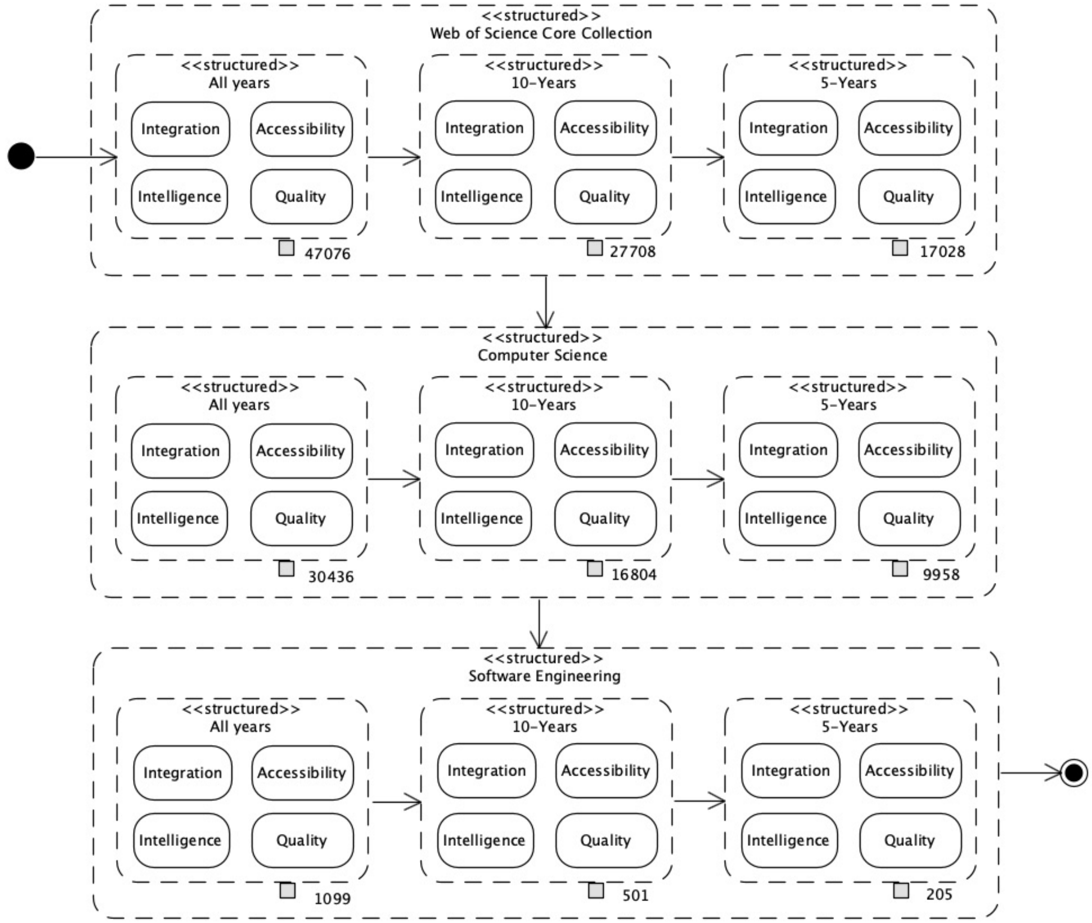

Figure 2 The systematic literature review methodology used.

	Web of Science Core Collection								
				Computer Science					
							Software Engineering		
Topics	All	10Y	5Y	All	10Y	5Y	All	10Y	5Y
WoT, Web of Things	64	55	27	62	53	26	56	18	9
IoT, Internet of Things	1101	1049	649	978	929	565	67	53	20
Integration	6432	3530	1811	3367	1588	757	202	80	27
Cluster, Clustering, Clustered	8955	4800	2503	4533	2350	1164	68	40	15
Interoperability	1043	426	182	882	339	142	88	32	9
Federate, Federated, Federating, Federation	543	251	109	285	136	63	8	3	0
Intelligence	8696	4750	3226	7739	3957	2564	222	97	48
Machine learning	7996	6615	5289	5205	3992	3047	61	50	34
Recommender systems	446	343	159	433	331	150	18	9	2
Ontologies, Ontology	4321	2109	1035	3031	1183	494	177	63	22
QoS, Quality of Service, Quality of Data, Quality of Information	1711	669	276	1469	573	233	72	30	9
Accessibility	893	605	357	321	214	121	7	2	1
Availability	4875	2606	1405	2131	1159	632	53	24	9
Total	47076	27808	17028	30436	16804	9958	1099	501	205

Table 2 Datasets of the literature review considering some Discovery perspectives.

2.4. Data extraction and preliminary analysis

For a first analysis of the results obtained after the searches, some data have been extracted from the final collection of Software Engineering for the last ten years. Table 3 shows a distribution of the papers published in conferences and JCR impact journals for each of the 13 subcategories. In the case of journals, there are, in total, 24 publishers supporting papers on the topic of study in Discovery. As shown in Table 3, the top three publishers in the ranking are Elsevier, IEEE and Springer. In terms of conferences, nine publishers were found publishing papers in the form of proceedings, being the best-ranked publishers in the ranking: IEEE, Springer and ACM. In terms of subject matter, analysis highlights that in the last ten years, there has been a more significant number of publications, standing out, in this order, the areas of Intelligence (97), Integration (80) and semantic aspects and Ontologies (63). On the other hand, less research in Discovery has been carried out in areas such as Recommender Systems (9), Federation (3) or Accessibility (2).

Table 4 shows a correlation matrix through working areas (interactions in grey colour). Notice that the Integration line (I) overlaps with all the research lines of study, while the Federation line (F) is the one with fewer relations with the rest of the lines (2). The matrix also helps to detect trending areas of Discovery research in recent years, highlighting the interaction between Intelligence with semantics and ontologies as the one with a significant number of papers, 41 papers.

Finally, Figures 3 and 4 show a cumulative comparison of existing works in the literature for the four main categories over the last ten years. Note that IoT and WoT were also included as categories in this comparison since they are the main topics of interest in our current research in Discovery systems. Proportionally, of the four study categories, the one with the highest number of published papers is in the area of Intelligence (43%), followed by Integration (31%), Quality (6%) and finally Accessibility (5%). Figure 4 shows the distribution of production by year. The study has been included up to May 2022, with no research work in any of the target categories and subcategories at the time of the survey.

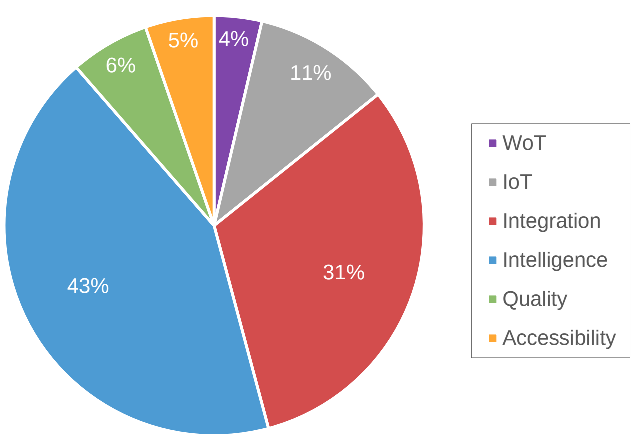

Figure 3 Distribution per categories.

3. Discovering and reusing solutions

This section presents the main contributions related to discovery and reusing solutions from the selected papers of this study. The final set of the 46 papers reviewed has been structured in the four main categories identified in Section 2 as the most relevant topics that may support the realization of the FAIR principles (i.e., Integration, Intelligence, Quality and Accessibility), in addition to IoT and WoT.

3.1. Integration

Integration concept involves physical interconnection of devices, interoperability between application processes and databases, and business logic coordination and management. In a discovery model, integration can be related to interoperability, clustering and federation characteristics.

The authors in (Garriga et al. 2018) present a hybrid service selection approach that gathers and evaluates syntactic and semantic information from the WSDL description of services interfaces. The results of the proposal have been compared to other approaches using syntactic and semantic information for service selection, in particular, for ranking candidate web services for their integration into a client application. The experiments for the comparison were performed using the EasySOC service discovery registry (Crasso et al. 2014) and the proposed hybrid service selection method outperformed the other three compared approaches, i.e., (i) using only syntactic information, (ii) using only semantic information, and (iii) using the service-similarity assessment methods (Stroulia & Wang 2005).

The integration of service matchers into a service market has been also addressed in the literature. In (Platenius et al. 2014), the authors define the requirements and identify the strategies in order to enable a decision-making process with regard to the architectural alternatives. The results can be used to integrate service matchers into existing or emerging service markets and thereby supporting both requesters and providers by achieving their goals.

There are solutions that address aspect modeling for developing the services related to the business processes. For example, the solution presented in (Jose et al. 2020) describes an architecture for service discovery capable of selecting web services that implement crosscutting characteristics that support the objectives established in the modeling phase, and execute them according to a prioritization. These objectives are described using the Web Service Modeling Framework (WSMF) (Fensel et al. 2011), including the desired characteristics of the service to be discovered. The results of this approach show that the domain and business process experts can focus on the aspect modeling without analyzing the implementation of services since they will be discovered and invoked dynamically during the execution of the processes. The aspect goal definition based on the concepts present in the service repository allows that integration feature.

In (Sellami et al. 2015), the authors propose a service-based solution to eliminate structural heterogeneities in a composite data providing (DP or DaaS) service data flow by the discovery of a special type of mediation web service named data mapping (DM) service. Data mapping services provide the data provided

	Categories													
Editorial	Total	W	T	I	C	R	F	L	M	S	G	Q	A	V
Journals														
ACM	8	1	1	1	1	0	0	1	1	0	2	0	0	0
Airiti Library	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Elsevier	66	4	4	5	4	6	1	15	10	0	9	3	0	5
Emerald	5	0	0	1	1	0	0	0	1	0	1	0	0	1
EricData	3	0	0	0	1	0	0	0	0	0	1	1	0	0
Frontiers Media	2	0	0	0	0	0	0	1	1	0	0	0	0	0
Hindawi	5	0	0	1	2	0	0	0	1	0	1	0	0	0
IEEE	48	1	9	3	4	2	0	13	5	0	5	3	0	3
IET	4	0	0	1	1	0	0	1	0	0	1	0	0	0
IGI-Global	6	0	0	1	0	0	0	1	0	0	1	2	0	1
InderScience	7	1	2	0	0	0	0	1	0	0	1	2	0	0
Informing Science Institute	3	0	0	1	0	0	0	1	0	1	0	0	0	0
IOS Press	4	1	1	0	0	0	0	1	0	0	1	0	0	0
MDPI	22	0	1	4	1	3	0	5	2	0	4	1	1	0
National Library of Serbia	3	0	0	1	0	0	0	1	1	0	0	0	0	0
Oxford Academic	1	0	0	1	0	0	0	0	0	0	0	0	0	0
SAGE	3	1	0	1	0	0	0	0	0	0	1	0	0	0
SAI	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Springer	43	0	7	7	4	2	1	7	4	1	5	4	0	1
Taylor & Francis	4	0	0	1	0	1	1	0	0	0	1	0	0	0
Thieme	2	0	0	1	0	1	0	0	0	0	0	0	0	0
Walter de Gruyter GmbH	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Wiley	9	0	0	0	0	2	0	4	1	0	2	0	0	0
World Scientific	4	0	0	0	1	0	0	2	1	0	0	0	0	0
Total Journals	255	9	25	33	20	17	3	54	28	2	36	16	1	11
Conferences														
ACM	15	0	0	2	0	0	0	4	6	0	0	2	0	1
AIP	3	0	0	1	0	0	0	1	0	0	1	0	0	0
EDP Sciences	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Elsevier	6	0	1	2	0	1	0	1	0	0	1	0	0	0
IEEE	177	7	25	27	16	11	0	30	13	6	20	10	1	11
IOPScience	4	0	0	1	0	1	0	0	0	1	0	0	0	1
Scientific Net	3	0	0	1	0	0	0	1	0	0	1	0	0	0
SPIE	3	0	0	2	0	1	0	0	0	0	0	0	0	0
Springer	34	2	2	10	4	1	0	6	3	0	4	2	0	0
Total Conferences	246	9	28	47	20	15	0	43	22	7	27	14	1	13
Total J&C	501	18	53	80	40	32	3	97	50	9	63	30	2	24

Table 3 Editorial distribution: Journal and Conferences (W: WoT, T: IoT, I: Integration, C: Cluster, R: Interoperability, F: Federate, L: Intelligence, M: Machine Learning, S: Recommender Systems, G: Ontologies, Q: Quality, A: Accessibility, V: Availability)

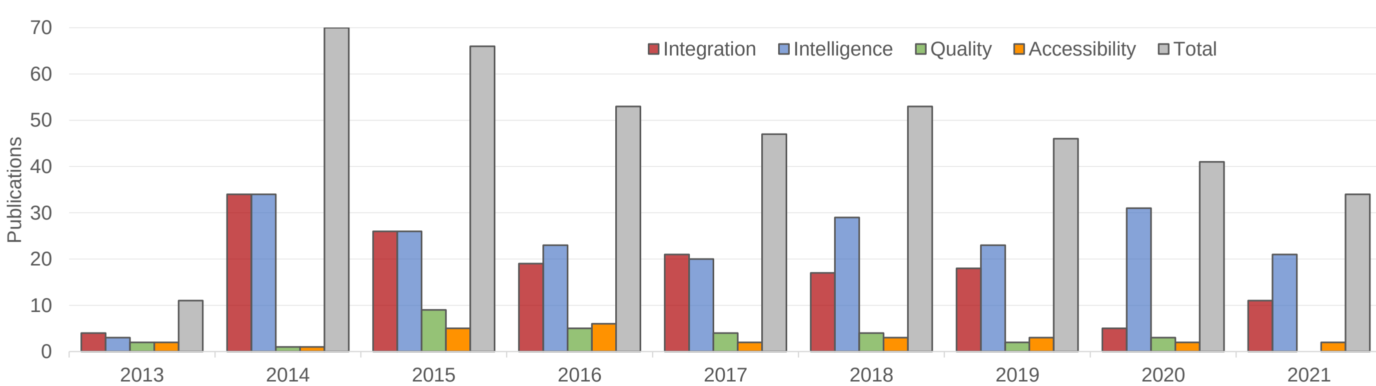

Figure 4 Distribution of the selected publications in the last ten-years (year 2022 does not include selected publications).

	W	T	I	C	R	F	L	M	S	G	Q	A	V
W	18	3	1	0	1	0	2	0	0	2	0	0	0
T	3	53	7	3	10	0	11	1	2	6	5	0	3
I	1	7	80	9	14	1	17	9	2	18	4	1	5
C	0	3	9	40	1	0	7	9	4	8	2	0	3
R	1	10	14	1	32	2	7	1	1	7	0	1	3
F	0	0	1	0	2	3	0	0	0	0	0	0	0
L	1	11	17	7	7	0	97	33	3	41	2	1	4
M	0	1	9	9	1	0	33	50	1	7	0	0	3
S	0	2	2	4	1	0	3	1	9	0	0	0	1
G	1	6	18	8	7	0	41	7	0	63	2	1	1
Q	0	5	4	2	0	0	2	0	0	2	30	0	3
A	0	0	1	0	1	0	1	0	0	1	0	2	0
V	0	3	5	3	3	0	4	3	1	1	3	0	24

Table 4 Category correlation matrix (W: WoT, T: IoT, I: Integration, C: Cluster, R: Interoperability, F: Federate, L: Intelligence, M: Machine Learning, S: Recommender Systems, G: Ontologies, Q: Quality, A: Accessibility, V: Availability)

by a service to meet the target representation expected by another service. The discovery of composite DMs is addressed as a pathfinding problem based on a decentralized solution to publish and discover these services, where the goal is to calculate the required set of services to resolve structural heterogeneities.

The approach in (Verma & Srivastava 2018) describes a service registry framework that makes use of an Extensible Messaging and Presence Protocol (XMPP) based model to maintain a service registry for mobile environments. The research work developed try to demonstrate that UDDI and other traditional approaches for implementing service registries cannot be directly adopted in mobile environments. This registry addresses issues specific to mobile environments and enables the discovery in a peer-to-peer configuration. The results of this study show its efficacy by evaluating the performance, correctness of updates, battery consumption, effect of the random behavior of users, conflict with native mobile applications, and reliability.

Some approaches are focused on mashup service discovery for improving mashup-based software development. The proposal described in (Cao et al. 2017) applies a clustering approach based on K-Means and Agnes algorithms to perform the clustering, and it applies the Latent Dirichlet Allocation (LDA) topic model to infer the topic probability distribution of mashup services for computation of similarity degrees among mashup service documents. This proposal, compared with other existing clustering approaches, achieves a significant improvement in terms of precision, recall, F-Measure, purity, entropy and clustering time, which should facilitate mashup service discovery and recommendation.

There are also clustering solutions that include federation mechanisms. In (Dasgupta et al. 2014), the authors propose a multiagent-based distributed SOA middleware for efficient semantic service discovery. This middleware is distributed into a system of federated registries managed by middleware agents. It includes a distributed algorithm called SmartDiscover, which can be divided into three main capabilities. The first one is responsible for the efficient clustering of service descriptions into a semantic taxonomic cluster (STC) space; the second and third functionalities are in charge of resolving a user query by exploiting the structural and semantic properties of the STC space to discover the relevant service agents. This approach resulted in positive results in terms of average query response time and the number of message exchanges required to maintain the distributed registry.

The work presented in (Field et al. 2014) proposes an architecture for a federated service registry that supports hierarchical and peering topologies enabling autonomous domains to collaborate. First, the registry topology is deployed to manage service records. Then, new domains or federations can be added to the infrastructure when required, thus supporting dynamic service provisioning. The evaluation results of this architecture demonstrate that it is able to handle the scale required for a production infrastructure under real-world conditions in terms of services and discovery requests.

The research performed in (Zhao et al. 2018) outlines the limitations of UDDI related to data replication and sharing. In this sense, smart contracts based on blockchain technology can be used to solve this problem in UDDI registries. Authors also provide a practical solution for protecting the data security and reliability using the blockchain to store information about service providers and the linked provided services to ensure the consistency and security of data. The results show the feasibility and safety of the approach based on the assumption that publishing web services is not a regular action and the number of published web services is limited. As a consequence, using smart contract technology to store the relevant information in the blockchain may form few transactions.

As mentioned, discovery processes are also addressed from a physical perspective by taken into account device features, communication technologies and network configurations. The approach in (Madiwalar et al. 2019) uses Software-defined networking (SDN) flow programmability and the device discovery features of Open Platform Communications Unified Architecture (OPC UA) to define a multi-level system which supports detecting and configuration of new devices, thus improving the integration of devices.

With regard to the IoT domain, there are existing approaches focused on the discovery of devices and related services. The approach presented in (Georgakopoulos et al. 2015) proposes a discovery-driven IoT architecture where devices, cloud resources and application components are described as services, thus allowing dynamic discovery, composition and integration depending on the requirements.

In the domain of Smart Cities, there are solutions or perspectives for enabling discovery mechanisms. For example, the approach presented in (Ben-Sassi et al. 2018) proposes an open and extensible platform for the creation and holistic interconnection of geolocated smart city services and devices from different sectors and stakeholders. The main goal of this platform is to overcome the fragmented and isolated IoT solutions toward a more integrated smart city ecosystem, by using service discovery and composition.

The approach in (Eisele et al. 2017) describes a component-based decentralized software platform called Resilient Information Architecture Platform for Smart Systems (RIAPS), which provides a discovery service infrastructure for smart environments. The discovery service is based on OpenDHT, a lightweight implementation of Distributed Hash Table (DHT), to store, query, and propagate the service details through the connected nodes of the network.

3.2. Intelligence

The intelligence category is used in this research work to gather aspects based on artificial intelligence such as recommender systems, transformers, machine learning, or semantic elements. With regard to the semantics, we have included in this section the most relevant solutions that apply some AI techniques for processing this information. The approaches that incorporate semantics for description and selection features have been reviewed in Section 3.1.

The service discovery normally implies trade-offs among different characteristics. In this context, AI can assist developers in dealing with service-oriented design with a positive impact on the management of services and descriptions. The approach presented in (G. Rodriguez et al. 2016) provides a detailed analysis of significant research works that presented AI-based approaches aimed at discovering, composing, or developing services in a loosely coupled way. As a result, this study demonstrates that using AI techniques improves the exploitation of semantic resources and enables the development of flexible systems requiring service development, composition and discovery. In a later solution described in (Venkatesh et al. 2017), the authors propose a personalized assistance framework using AdaRank, a learning-to-rank machine learning algorithm, to address the analysis of context-aware interactions with users in service selection, and the learning from the history of this process. The proposed algorithm improves the precision in comparison with nine alternative algorithms used to find personalized services.

With regard to learning approaches, the research performed in (Olson et al. 2015) analyzes the use of profiling algorithms in the UniFrame Resource Discovery System (URDS). It proposes to use a learning-based technique as a better mechanism to divide the search space and reduce the complexity of matching. The results show that using this type of algorithms in a discovery service can obtain services with better characteristics with a reduction in discovery time.

The work presented in (Chen et al. 2018) describes a goal-oriented service composition approach that includes a split-choice mechanism into the composition process. It proposes a decentralized heuristic planning algorithm based on backward-chaining to support flexible service discovery and is focused on handling systems with real-time environment changes with high mobility. The results of this approach show a more successful behavior in pervasive computing environments where providers are mobile, and it is uncertain what services are available.

The application of machine learning techniques is also addressed in the literature. The work presented in (Bao et al. 2017) study how the construction of models for fine-grained level attributes may contribute to context-aware API linkage prediction. This way, service discovery and recommendation is supported by LDA technique from the information retrieval field to summarize topic distribution from API description documents.

The EasySOC approach (Crasso et al. 2014) uses text mining techniques for automatically pulling out relevant information about the desired service from the client-side source code. It uses machine learning techniques to allow the manual selection of web services from a list of candidates. The evaluation of this approach shows the effectiveness of its discovery mechanism by comparing four different heuristics for automatic query generation from source code on a large dataset of web services.

In (Zhang et al. 2018), authors present an approach to mining knowledge on service functionalities from a service registry. This work consists of a hybrid service discovery solution based on expanded queries calculated from the recommendations of service goals assigned to similar service goal clusters for the given queries. Semantic similarities are measured between service goals by employing word similarities and distinguishing the importance of different types of words in a specific domain. The results show that the approach can effectively recommend semantically similar service goals for queries and the proposed service discovery approach outperforms several popular approaches. Furthermore, the retrieved services can be displayed along with their service goals relevant to the query, which can help the requester in choosing the most suitable services.

Semantic information can be managed to improve discovery mechanisms. In this sense, ontologies ease the representation of semantic information and enhance mapping operations between related concepts. For example, the approach presented in (Arul & Prakash 2019) uses the semantic description of web services and ontological classes to automatically find the composite services matching the user request. It performs a transformation process from the syntactic description of web service (WSDL) into a semantic description using the OWL-S and the web service modeling ontology (WSMO). This process annotates WSDL files with additional ontologies stored in an ontology-based service repository (OSR).

In (Boukhadra et al. 2016), the authors describe a decentralized and interoperable approach for discovering and composing web services containing semantic information in large-sized P2P networks. It applies a scalable epidemic algorithm based on matching techniques of the OWL-S process model in order to ensure high recall and reduce the search space. Consequently, the number of messages exchanged is decreased and the execution time for discovery is reduced.

The authors in (Garcia Coria et al. 2014) present a method of software component reuse that facilitates the semi-automatic reuse of web services in a cloud computing environment. The engine of web service discovery is based on a multi-agent system (MAS) that represents ontology concepts linked to a UDDI registry. The description of web services (WSDL) includes the ontological concepts that it represents.

Ontologies can be used for the explicit representation of concepts and the relationships that are held among entities, so context ontology can solve the problem of ambiguity in service discovery. In (Zeshan et al. 2017), authors propose a framework that uses a context ontology for modeling the user demand and a service discovery algorithm. It is able to extend and adapt the vocabulary used to describe services. The proposed service matching considers the associated priorities and the requirements of the requester, during the service discovery process.

Modeling techniques can be applied to normalize service descriptions for improving matching and querying operations in service discovery. The approach described in (Schwichtenberg et al. 2014) proposes a matching algorithm based on a Query View Transformation (QVT) to normalize behavioral models for enabling a behavior-aware matching between service requests and service offers. The matching process is a similarity-based matching approach, which leverages background ontologies to identify synonyms, homonyms, as well as correspondences between classes with a similar ontological semantic.

There are alternatives to the use of ontologies for describing and processing semantic information. The approach in (Cheng et al. 2021) describes a distributed web service discovery framework that retrieves the underlying semantic concepts from existing service metadata. This proposal assists service consumers in finding not only similar service operations but also that can be composed ones using the given requests. The discovery framework uses an index library for applying semantics mining to overcome the shortcomings related to low searching accuracy in other compared frameworks.

3.3. Quality

Quality factors must be considered in selection and discovery processes of software artifacts. Quality of service (QoS), quality of experience (QoE), quality of data (QoD), quality of information (QoI), or other attributes from quality models, play a key role both in the result of the elements generated and in response, as well as in the discovery process itself.

Generally, service discovery solutions based on Universal Description, Discovery and Integration (UDDI) provide discovery mechanisms using only the functional aspects of services without considering the non-functional characteristics. Nevertheless, modern software systems related to web services also require the incorporation of quality factors in discovery processes. For example, the approach proposed in (Kim et al. 2015) presents a QoS broker-based architecture that supports the discovery and selection of web services based on QoS properties and historical data.

In (Zisman et al. 2013), the authors present a framework for dynamic service discovery in which candidates are identified in parallel to the execution of the applications. This means that service discovery is executed at runtime to enable service replacement if new components are incorporated to the system. The framework supports service discovery in pull and push modes, and the identification of services relies on distance-based matching of structural, behavioral, quality, and contextual characteristics of services and applications. The results, focused on the execution time of the retrieval process, show a linear increase of time with respect to service registry size and significant gains from the use of push mode query execution.

Regarding flexibility and customization, service selection approaches are constrained by a limited number of attributes or parameters to be used as query inputs. As a consequence, the addition of a new parameter usually requires a reimplementation of the process. Furthermore, users must be provided with mechanisms to establish different selection policies or to modify the configuration.

In (Laga et al. 2013), a friendly service selection mechanism where users define the criteria to be applied for selecting the actual service is proposed. When a virtual service is invoked, a graphical user interface (GUI) is displayed for allowing the end-user to define the policy to be used in the selection process. It also displays a form that allows the user to provide the inputs required. Then, the marketplace selects the services that best satisfy the policy, the selected service is executed, and the results are displayed within the GUI.

The approach in (Houmani et al. 2020) presents a data-driven framework that enables the service discovery depending on the matching between data products requirements and services. The proposed architecture is based on microservices supporting scalability, and it provides the possibility of integrating geographical features. It includes components to overcome the service mesh limitations that prevent the appropriate measurements to maintain a certain level of QoS in terms of response time and percentage of accepted requests when receiving incoming rates that exceed system capacity.

From the point of view related to available resources and, more specifically, energy consumption, the fragmentation of services in modern software systems can lead to considerable energy consumption of servers, which should be considered and analyzed in large-scale computing environments. In this sense, when a service requirement can be answered by different web services, the energy consumption for each service to reply to the service request may be different, and therefore, discovery and selection processes including this information could maximize the energy efficiency of systems. The experimental results of (Liu et al. 2015) show that there is a negative correlation between the nominal power consumption of a specific web service and the performance counter CPU utilization. However, different calls of the same web service may cause different results regarding the calculation and may consume different values of power consumption.

In the IoT domain, discovery services are required to consider many selection criteria, e.g., device capabilities, location, context data type, contextual situations, and quality. Quality of context (QoC) aims to consider additional knowledge such as precision, accuracy, and up-to-date information, among other possible examples, to improve the characterization of devices and the related acquired data. In this sense, the approach presented in (Gomes et al. 2019) describes QoDisco, a QoC-aware discovery service for the IoT. This information can be used to improve the discovery process enabling the filter of non-relevant data o data with insufficient quality. The results show that the strategy of distributing repositories and providing an asynchronous search process can be promising for resource discovery in the IoT.

3.4. Accessibility

Solutions related to decentralized service discovery processes may lead with accessibility issues causing, for example, that the software agent in charge of resolving a discovery request only forwards service queries to its neighbors. As a consequence, it can only search services located in this component or in the connected components, thus reducing the success rate and preventing the adaptability of the system. There must exist a global connectivity mechanism to guarantee the global accessibility of services. For example, the approach in (Li & Jiang 2016) describes a self-organization based on intermediary utilities of agents, which are formed by the service distribution information of its contextual agents, *i.e.*, the neighbors and neighbors' neighbors, in a specific level. It enables a proper performance in responses executed in static and dynamic discovering scenarios.

Availability is normally treated as a QoS property in service discovery and this aspect could be understood as a quality feature to be included in Section 3.3. However, considering our research questions and the particular perspective of accessibility in FAIR principles, availability has a direct impact on the possible access to a service (even if we only refer to its description) and on its subsequent use. With regard to this issue, the proposal presented in (Ma et al. 2021) describes an approach to RESTful service discovery named as Test-Oriented API Search with Semantic Interface Compatibility (TASSIC). It provides candidate services with rankings based on semantic similarity according to user-defined criteria. The process invokes each candidate service to verify its availability and functionality by comparing the actual response with the test cases listed in a reference document.

In addition to these approaches, the research papers reviewed in Section 3.1 addressing federation mechanisms are indirectly related to accessibility because they enable and easy the recovery of services from identifiers and through standardized communication protocols.

3.5. IoT and WoT

Our research study is focused on discovery solutions related to modern web systems. In particular, we are interested in reuse opportunities of cyber-physical systems. This subsection includes the review of the most relevant publications with regard to IoT and WoT discovery mechanisms that are not focused on the four main categories used for the systematic review, i.e., integration, intelligence, quality, or accessibility.

IoT services can be considered a dynamic content, including data sources and middleware infrastructures, and a possible solution to manage them are Content Delivery Networks (CDNs). The approach described in (Forestiero 2017) provides a swarm-based algorithm to build an IoT discovery service based on CDN. This approach offers an alternative to clustering by placing in the same region those metadata representing similar contents.

In (Cabrera & Clarke 2022), authors propose a self-adaptive service model for smart cities to support service discovery. The model is based on a Deep Q-Network (DQN) algorithm to proactively adapt the service information according to three types of city events: unforeseen, scheduled, and periodic. The results show that the proposed model improves the discovery efficiency for unforeseen and scheduled events but has limitations when handling periodic events.

With regard to social IoT solutions, the approach presented in (Ruta et al. 2017) enables autonomic machine-to-machine interactions and a semantic-enhanced service discovery for home and building automation applications. The evaluation of the proposed framework makes use of several types of off-the-shelf devices, in order to verify both feasibility and effectiveness in case of different hardware specifications.

Context-aware discovery services can be used to minimize the overload of control operations in IoT applications. The work in (Barreto et al. 2017) presents a Constrained Application Protoco (CoAP) extension to support the context-aware discovery of smart objects. This extension matches interest and context information with the smart objects state data to retrieve the most relevant artifacts. Those objects that cease to have interest go to an idle state, thus optimizing the network and battery usage. The evaluation of this approach results in a decrease in the number of messages (up to 80%) whereas the overhead of the discovery latency shows a maximum time increase of 1.5 seconds.

Search engines are the key to ensure reusing and discovery solutions in the WoT domain. In this sense, WoT search engines (Tran et al. 2017) include web crawlers, resource registration and local discovery, spatial indexes, clustering mechanisms or prediction models, among other possible examples. All these mechanisms still require addressing many issues, including scalability, diversity or efficiency.

In (Sciullo et al. 2020), the authors propose WoT Store, a platform for managing and deploying WoT resources that allows the dynamic discovery and managing of the active things available on a public or private deployment. It offers a web dashboard for enabling users to search and list things, monitor their properties and events, and execute their commands. The approach presented in (Kamilaris et al. 2016) describes WOTS2E, a WoT Semantic Search Engine for discovering linked data endpoints corresponding to WoT devices and their services. This proposal uses web crawlers acting as agents for the discovery of semantically annotated WoT resources.

The existence of different service discovery protocols related to the WoT could introduce a new heterogeneity issue. To solve this, solutions such as (Son et al. 2015) try to overcome the overhead introduced if enablers and adapters are used for each discovery protocol. This approach presents a light-weight universal protocol that enables the discovery of connected smart objects with different service discovery protocols.

The abstraction layer provided by the WoT reduces the heterogeneity issues related to definitions, accessing and communication protocols. Nevertheless, exactly as in the IoT, the number of possible ways in which things can be described is almost unlimited. For this reason, approaches such as (Antoniazzi & Viola 2019) propose a high-level abstraction, i.e., an ontology, of devices that form part of a smart application. Authors also propose a discovery mechanism based on the ontology that is flexible and customizable by further extending the semantic descriptions with other ontologies.

An ontology-based approach for discovery in WoT ecosystem is proposed in (Serena et al. 2018). It is based on the semantic description, in terms of a WoT ontology, of web things and their access interfaces. Moreover, the approach includes a mapping ontology model to describe how thing values must be processed, i.e., the mechanism for accessing the values provided by web things.

In (B. H. Rodriguez & Moissinac 2015), the authors propose an approach related to the Multimodal Architecture and Interfaces recommendation that enables the discovery and registration of components used in multimodal systems for the WoT. It extends the event model with two new events specified for discovery and registration needs.

Recommender systems can be applied to address discovery and selection processes in the WoT domain. However, recommender systems could have some limitations in handling sparse recommendation spaces, and this type of distributed space is present in most WoT applications. The approach presented in (Cai et al. 2014) exploits the principle of "object typicality" (concept related to cognitive psychology) to develop a recommendation method to address the data sparsity, computational efficiency and error prediction in WoT recommendation. The evaluation results show a significant improvement in comparison with other recommender systems under the condition of sparse training data. It also outperforms the results in terms of mean absolute error and root mean square error.

4. Discussion

The study has resulted in a manageable subset of articles that we have considered as relevant by taking into account a specific application domain of modern software systems and a set of considerations. As part of the discussion of this result, this section firstly provides an analysis of the observations extracted from the review performed in this study. Secondly, the threats to validity have been identified. And finally, we summarize the future research lines that are still open with regard to the existing approaches in the literature.

4.1. Analysis

This study is focused on discovery and reuse solutions that are in some aspects related to modern software systems, paying special attention to the domains of web applications and cyber-physical systems. The literature review shows that traditional approaches continue to be used in modern systems, sometimes requiring the modification of their characteristics or the extension of their functionalities. Furthermore, new discovery models, approaches or mechanisms have been proposed in the last ten years. A selection of the most important observations that have been extracted from the study is described below.

From a high-level perspective of solutions related to discovery in software systems, we have identified three types of research works. The first type corresponds to articles whose all contributions are related to software discovery; the second type includes those publications that are partially dedicated to discovery processes, providing some contribution in this area; finally, in papers of the third type, there is no contribution to the discovery, but the proposed solution indirectly helps search and reuse operations. The review in Section 3 includes a more significant number of articles of the second type, with a few research papers of the first type. Regarding the third type, only those works whose contributions have been considered to have a high impact on software discovery solutions have been included.

Traditional approaches based on WSDL service descriptions and UDDI registries are adapted to support discovery mechanisms in modern applications. Most of the modifications are focused on expanding the descriptions by making use of semantic information, thus enabling the application of new inference and artificial intelligence techniques to improve search processes.

Semantic descriptions are grouped into vocabularies, taxonomies and ontologies, among other possible examples. These descriptors try to expand the information to characterize the software artifacts to be discovered or reused. Nevertheless, there is no unique or common ontology used as reference neither components, services, devices or things. Although the objective is the homogenization of the information representation, a new fragmentation has also been generated in the data and in the metadata, and mechanisms are required to be able to solve all the related concepts. In some cases, new approaches propose new representations or recommendations instead of reusing or expanding the existing vocabularies.

As far as the terminology used in discovery solutions is concerned, there are different approaches to the difference between discovering and selecting a service for reuse. Although in many proposals and models they are combined in a single process, in others, these two operations are considered to be executed sequentially or that have the autonomy to be executed independently. In this second perspective, discovery operations are more tied to queries, while select operations are linked to the execution of a particular service.

Considering particular search features of the solutions in the literature, there are many approaches of service discovery that also address the service composition during the query process. This additional functionality is intended to ease the automatic discovery of operations that could not be provided by only one service; however, the composition or orchestration of the required service could be managed by the client or service requester. With regard to particular selection features, graphical user interfaces can be included in discovery solutions to support semi-automatic or manual (supervised) selection by providing useful information.

Focusing on the accessibility, although solutions around discovery try to maximize the visibility of managed elements in registries and directories, it is still difficult to find marketplaces containing suitable services and, in addition, intermediaries such as middleware and brokers are still not connected to facilitate the delegation of queries. Currently, new solutions such as using OpenAPI for the description of RESTful web services try to reduce fragmentation in this regard.

The resulting subset or articles from Section 2 and the review performed in Section 3 help in answering the research questions raised in Section 1. With regard to $RQ1$, traditional approaches of discovery models are being utilized in modern software systems. In some cases, they are adapted especially focusing on the improvement of semantics, composition and performance issues. Furthermore, new discovery models related to web environments (such as (Crasso et al. 2014) or (Verma & Srivastava 2018)), or to the IoT/WoT and CPS (e.g., (Georgakopoulos et al. 2015) or (Ben-Sassi et al. 2018)) have been proposed.

The $RQ2$ is focused on the advantages and disadvantages of discovery models in web environments. In this regard, web technologies and related paradigms provide the most adequate context for distributing and scaling, but also keep a suitable set of technologies for obtaining high-level performance results related to behavior execution and communication operations. New technologies related to web services aim to reduce communication overload and intend to maximize parallel and asynchronous executions by applying integration patterns, among others.

Regarding the $RQ3$, one of the clearest opportunities for discovery and reusing mechanisms is the modern and trending domain of cyber-physical systems and other related solutions with devices that operate over the IoT. The number of devices, their heterogeneity, distributed deployments together with virtual components providing a great number of reusable services make this domain a perfect target scenario to develop discovery solutions. Apart from the CPS domain, blockchain (Zhao et al. 2018), modeling approaches (Jose et al. 2020), and data management techniques (Sellami et al. 2015), among other examples, have emerged as new opportunities for discovery models.

With regard to data management and analyzing the $RQ4$, FAIR principles could be applicable in discovery solutions since findability and reusability are the core of a discovery model. Moreover, accessibility and interoperability (from the point of view of FAIR concepts) are indirectly related to common features of this kind of solutions. In this sense, the availability of software elements, together with their access capabilities, in addition to the representation of data and metadata have an impact on discovery approaches. However, there are no relevant approaches in the literature applying FAIR principles for discovering or reusing services in modern software systems.

4.2. Threats to validity

Our study, as any other literature review, involves a series of threats to validity (Ampatzoglou et al. 2019). Regarding the selection validity, the main limitation is the inclusion of the most relevant papers related to the topics of this study. The hardest filter has been the period, from 2013 to 2022, i.e., the last ten years. This assumption is made to focus on recent solutions, although consequently many older relevant works in the field have not been included. Another possible threat is the choice of the search terms, taking into account the strings (and the different possible combinations of the strings) used in the query process. In this sense, we tried to maximize the suitability of the search terms by performing search tests to build the final set of strings used for obtaining the articles.

From the final set of obtained articles, we performed two manual filtering processes. The first filter considered the title, abstract and keywords. In particular, the abstract descriptions led us to discard those publications that did not focus on discovery and reuse solutions. Regarding the quality of the publications, the second filter consists of a quick review of the content and the contributions, and those publications with a not noteworthy main contribution related to our application domain were filtered. This approach was chosen for handling the large set of selected papers, although there exists a chance of excluding papers that are related to discovery and reuse solutions, but whose contributions have not been considered as relevant in the manual filtering process. More details about the selection criteria have been provided in Section 2.

4.3. Future Research Lines

The resulting category correlation matrix in Table 4, the challenges identified from the study in Section 3, and the analysis performed in Section 4.1 aim to define the possible future research lines. Next, we propose approaches that could be applied in discovery and reuse solutions in cyber-physical systems.

A discovery perspective offers, as a response, a set of services (functionalities) that satisfy the established specifications. This set of services can be ordered, prioritized, and/or filtered to select the best alternative. Such selection may be set on criteria that consider user preferences, usage history, or other information of interest to the application domain. With this target, an existing open research line is to investigate the application of **recommender systems** to suggest the best alternatives during the discovery of services. Some hybrid models have emerged from the intensive use of recommender systems. These combine some methods such as content-based models, or collaborative filtering, among others, to make predictions and recommendations, mitigating the problems each model has individually.

Another aspect to be considered as an open challenge is the existence of a **federated discovery** model, connecting different registries and directories. A discovery model could include query federation features to delegate requests that cannot be resolved in specific service discovery. Extension mechanisms of the queries can also be included, for example, analyzing the possibility of applying mutation techniques. This additional behavior could add a processing overhead, therefore it is necessary to evaluate the system's performance and carry out optimization processes to reduce execution times. Finally, it is important to balance the performance with the amount of information provided in response to user requests or any other agent involved.

In conjunction, all the information that serves to describe a cyber-physical component, as well as the values generated by the devices themselves, in addition to the interaction of users with the different parts of the system (i.e., service discovery, recommendation systems, each one of the devices, etc.) can be made available to any citizen through **open data** platforms. Furthermore, access to such data can be used as information to adapt the recommendation systems to users' interests (end-users or developers). This aspect could be addressed by applying the FAIR principles, both for descriptions of devices and for the data generated by these devices.

5. Conclusions

Discovery and reuse are two fundamental principles in developing modern software systems. Many of these systems have a functional topology based on an architecture of autonomous software entities (such as components, agents or services) that form a fully operational system. Traditionally, this development approach has a software industry associated with specific methodologies, techniques and tools for producing software artifacts stored in repositories that can be easily located and accessed by third parties. This development perspective is not new since it has already been widely studied and used in Component-Based Software Engineering (CBSE). However, since the rise of the Internet of Things discipline, many traditional CBSE practices are re-emerging for application in modern software systems that provide solutions to new Smart City scenarios, such as mobility, smart buildings, healthcare, and robotics, among others. Cyber-Physical Systems (CPS) offer solutions in these areas through a network system of electronic devices that operate and communicate using the Internet of Things (IoT) technologies.

Due to the popularity of IoT devices, there are many protocols and standards related to IoT. However, they usually focus on the device hardware dimension, its connectivity at the network level, or the software controlling the device at a low level. An important initiative focused on developing standards linked to software development, close to the application layer for cyber-physical components, is the W3C Web of Things (WoT). The WoT paradigm is an approach to IoT that uses web technology to facilitate the construction and use of such embedded systems. Although the WoT is still an emerging technology, it can be considered as a new field to be explored in software engineering since there is currently a lack of software system development practices in this domain. Many principles traditionally studied in CBSE can be applied and extended. One of these principles (the focus of our research interest for years) is the Discovery perspective (of "*things*" in this IoT/WoT domain).

The importance of data and metadata records is also present in discovery models because services, components, or any other software artifact managed in these approaches are described as data registries. Furthermore, data generated by these software artifacts must be formally managed to maximize the use of this information. For this reason, data management techniques and guidelines must be considered to be included in discovery solutions for improving the description of the services, their storage, the search process, or the reuse capabilities. In this sense, FAIR (Findability, Accessibility, Interoperability, and Reusability) principles, mainly linked to the discipline of engineering and data science, could be applied to improve the capabilities of new discovery models in modern software systems.

In our point of view, modern discovery approaches must consider new trending design perspectives in their development. The study performed in this paper has resulted in a delimited survey about discovery models and reuse techniques in developing of software systems, specifically, in solutions related to web applications and CPSs. For this study, four main categories has been considered when building new discovery models: Integration, Intelligence, Quality, and Accessibility. As result, a total of 501 papers were analysed, of which 46 were selected in the field of software engineering. The analysis of the results shows the usefulness of traditional approaches, although it also identifies the existence of new solutions related to new domains, paradigms and technologies. The discussion of the existing approaches in the literature highlights the possible applicability of FAIR principles. For future work, we intend to address some of the open research lines identified in this review.

Acknowledgments

This work has been funded by the EU ERDF and the Research and Knowledge Transfer Department, Andalusian Government (Spain) under the R&D project UrbanITA (ref. PY20_00809) and the Spanish Ministry of Science and Innovation under the R&D project HERMES (ref. PID2021-124124OB-I00).

References

Ampatzoglou, A., Bibi, S., Avgeriou, P., Verbeek, M., & Chatzigeorgiou, A. (2019). Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Information and Software Technology*, *106*, 201-230. doi: https://doi.org/10.1016/j.infsof.2018.10.006

Antoniazzi, F., & Viola, F. (2019). Building the Semantic Web of Things Through a Dynamic Ontology. *IEEE Internet of Things Journal*, *6*(6), 10560–10579. doi: 10.1109/JIOT.2019.2939882

Arul, U., & Prakash, S. (2019). A unified algorithm to automatic semantic composition using multilevel workflow orchestration. *Cluster Computing-The Journal of Networks Software Tools and Applications*, *22*, 15387–15408. doi: 10.1007/s10586-018-2604-2

Aziez, M., Benharzallah, S., & Bennoui, H. (2019). A full comparison study of service discovery approaches for internet of things. *Int. Journal of Pervasive Computing and Communications*, *15*, 30–56. doi: 10.1108/IJPCC-04-2019-0038

Bao, Q., Zhang, J., Duan, X., Ramachandran, R., Lee, T. J., Zhang, Y., … Maskey, M. (2017). A Fine-Grained API Link Prediction Approach Supporting Mashup Recommendation. In I. Altintas & S. Chen (Eds.), *24th IEEE International Conference on Web Services (ICWS)* (pp. 220–228). doi: 10.1109/ICWS.2017.36

Barreto, F. M., Duarte, P. A. d. S., Maia, M. E. F., Andrade, R. M. d. C., & Viana, W. (2017). CoAP-CTX: A Context-Aware CoAP Extension for Smart Objects Discovery in Internet of Things. In S. Reisman et al. (Eds.), *41st IEEE Annual Computer Software and Applications Conference (COMPSAC)* (pp. 575–584). doi: 10.1109/COMPSAC.2017.87

Ben-Sassi, N., Dang, X.-T., Fähndrich, J., Görür, O.-C., Kuster, C., & Sivrikaya, F. (2018). Service Discovery and Composition in Smart Cities. In J. Mendling & H. Mouratidis (Eds.), *Information Systems in the Big Data Era* (pp. 39–48). Springer. doi: 10.1007/978-3-319-92901-9_4

Boukhadra, A., Benatchba, K., & Balla, A. (2016). Efficient distributed discovery and composition of OWL-S process model in P2P systems. *Journal of Ambient Intelligence and Humanized Computing*, *7*(2), 187–203. doi: 10.1007/s12652-015-0313-8

Cabrera, C., & Clarke, S. (2022). A Self-Adaptive Service Discovery Model for Smart Cities. *IEEE Transactions on Services Computing*, *15*(1), 386–399. doi: 10.1109/TSC.2019.2944356

Cai, Y., Lau, R. Y. K., Liao, S. S. Y., Li, C., Leung, H.-F., & Ma, L. C. K. (2014). Object typicality for effective Web of Things recommendations. *Decision Support Systems*, *63*, 52–63. doi: 10.1016/j.dss.2013.09.008

Cao, B., Liu, X. F., Liu, J., & Tang, M. (2017). Domain-aware Mashup service clustering based on LDA topic model from multiple data sources. *Information and Software Technology*, *90*, 40–54. doi: 10.1016/j.infsof.2017.05.001

Chan, N. N., Gaaloul, W., & Tata, S. (2012). A recommender system based on historical usage data for web service discovery. *Service Oriented Computing and Applications*, *6*(1), 51–63. doi: 10.1007/s11761-011-0099-2

Chen, N., Cardozo, N., & Clarke, S. (2018). Goal-Driven Service Composition in Mobile and Pervasive Computing. *IEEE Transactions on Services Computing*, *11*(1), 49–62. doi: 10.1109/TSC.2016.2533348

Cheng, B., Li, C., & Chen, J. (2021). Semantics Mining & Indexing-Based Rapid Web Services Discovery Framework. *IEEE Transactions on Services Computing*, *14*(3), 864–875. doi: 10.1109/TSC.2018.2831678

Cirani, S., Davoli, L., Ferrari, G., Léone, R., Medagliani, P., Picone, M., & Veltri, L. (2014). A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things. *IEEE Internet of Things Journal*, *1*(5), 508–521. doi: 10.1109/JIOT.2014.2358296

Clark, J., Clarke, C., De Panfilis, S., Granatella, G., Predonzani, P., Sillitti, A., … Vernazza, T. (2004). Selecting components in large COTS repositories. *Journal of Systems and Software*, *73*(2), 323–331. doi: 10.1016/j.jss.2003.09.019

Crasso, M., Mateos, C., Zunino, A., & Campo, M. (2014). EasySOC: Making web service outsourcing easier. *Information Sciences*, *259*, 452–473. doi: 10.1016/j.ins.2010.01.013

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, *6*(2), 86–93. doi: 10.1109/4236.991449

Dasgupta, S., Aroor, A., Shen, F., & Lee, Y. (2014). SMARTSPACE: Multiagent Based Distributed Platform for Semantic Service Discovery. *IEEE Transactions on Systems Man Cybernetics-Systems*, *44*(7), 805–821. doi: 10.1109/TSMC.2013.2281582

Eisele, S., Madari, I., Dubey, A., Karsai, G., & IEEE. (2017). RIAPS: Resilient Information Architecture Platform for Decentralized Smart Systems. In *20th IEEE International Symposium on Real-Time Distributed Computing (ISORC)* (pp. 125–132). doi: 10.1109/ISORC.2017.22

Eshuis, R., Lécué, F., & Mehandjiev, N. (2016). Flexible Construction of Executable Service Compositions from Reusable Semantic Knowledge. *ACM Transactions on the Web*, *10*(1), 1–27. doi: 10.1145/2842628

Fensel, D., Facca, F. M., Simperl, E., & Toma, I. (2011). Web Service Modeling Ontology. In *Semantic Web Services* (pp. 107–129). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: http://doi.org/10.1007/978-3-642-19193-0_7

Field, L., Memon, S., Marton, I., & Szigeti, G. (2014). The EMI Registry: Discovering Services in a Federated World. *Journal of Grid Computing*, *12*(1), 29–40. doi: 10.1007/s10723-013-9284-1

Forestiero, A. (2017). A Smart Discovery Service in Internet of Things Using Swarm Intelligence. In C. Martin-Vide, R. Neruda, & M. A. Vega-Rodriguez (Eds.), *6th Int. Conf. on the Theory and Practice of Natural Computing (TPNC)* (Vol. 10687, pp. 75–86). doi: 10.1007/978-3-319-71069-3_6

Frakes, W., & Kang, K. (2005). Software reuse research: status and future. *IEEE Transactions on Software Engineering*, *31*(7), 529–536. doi: 10.1109/TSE.2005.85

Garcia Coria, J. A., Castellanos-Garzon, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. *Expert Systems with Applications*, *41*(4), 1189–1205. doi: 10.1016/j.eswa.2013.08.003

Garriga, M., De Renzis, A., Lizarralde, I., Flores, A., Mateos, C., Cechich, A., & Zunino, A. (2018). A structural-semantic web service selection approach to improve retrievability of web services. *Information Systems Frontiers*, *20*(6), 1319–1344. doi: 10.1007/s10796-016-9731-1

Georgakopoulos, D., Jayaraman, P. P., Zhang, M., Ranjan, R., & IEEE. (2015). Discovery-Driven Service Oriented IoT Architecture. In *IEEE Conference Collaboration Internet Computing* (pp. 142–149). doi: 10.1109/CIC.2015.34

Gomes, P., Cavalcante, E., Batista, T., Taconet, C., Conan, D., Chabridon, S., … Pires, P. F. (2019). A semantic-based discovery service for the Internet of Things. *Journal of Internet Services and Applications*, *10*, 14. doi: 10.1186/s13174-019-0109-8

Houmani, Z., Balouek-Thomert, D., Caron, E., & Parashar, M. (2020). Enhancing microservices architectures using data-driven service discovery and QoS guarantees. In *20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid)* (pp. 290–299). doi: 10.1109/CCGrid49817.2020.00-64

IEEE SA. (2020). *P1451.99 – Standard for Harmonization of Internet of Things (IoT) Devices and Systems.* Retrieved from https://standards.ieee.org/ieee/1451.99/10355/

Iribarne, L., Troya, J. M., & Vallecillo, A. (2004, 01). A Trading Service for COTS Components. *The Computer Journal*, *47*(3), 342–357. doi: 10.1093/comjnl/47.3.342

ISO/IEC. (2019). *27050-1 – Information Technology – Electronic discovery – Part 1: Overview and concepts.* Retrieved from https://www.iso.org/standard/78647.html

ISO/IEC. (2021). *30118-1 – Information Technology – Open Connectivity Foundation (OCF) Specification – Part 1: Core specification.* Retrieved from https://www.iso.org/standard/82127.html

ISO/IEC. (2022). *23093-2 – Information Technology – Internet of media things – Part 2: Discovery and communication API.* Retrieved from https://www.iso.org/standard/81587.html

Jacobsen, A., de Miranda Azevedo, R., Juty, N., Batista, D., Coles, S., Cornet, R., … Schultes, E. (2020). FAIR Principles: Interpretations and Implementation Considerations. *Data Intelligence*, *2*(1-2), 10–29. doi: 10.1162/dint_r_00024

Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The Journey So Far and Challenges Ahead. *IEEE Software*, *35*(3), 24–35. doi: 10.1109/MS.2018.2141039

Jirkovský, V., Obitko, M., Kadera, P., & Mařík, V. (2018). Toward Plug amp;Play Cyber-Physical System Components. *IEEE Transactions on Industrial Informatics*, *14*(6), 2803–2811. doi: 10.1109/TII.2018.2794982

Jose, H. S. A. d. S., Cappelli, C., Santoro, F. M., & Azevedo, L. G. (2020). Implementation of Aspect-oriented Business Process Models with Web Services. *Business & Information Systems Engineering*, *62*(6), 561–584. doi: 10.1007/s12599-020-00643-2

Kamilaris, A., Yumusak, S., Ali, M. I., & IEEE. (2016). WOTS2E: A Search Engine for a Semantic Web of Things. In *IEEE 3rd World Forum on Internet of Things (WF-IoT)* (pp. 436–441). doi: 10.1109/WF-IoT.2016.7845448

Kim, Y., Shin, Y., & Doh, K.-G. (2015). Quantitative trust management with QoS-aware service selection. *International Journal of Web and Grid Services*, *11*(3), 247–264. doi: 10.1504/IJWGS.2015.070962

Kuhrmann, M., Fernández, D. M., & Daneva, M. (2017). On the pragmatic design of literature studies in software engineering: an experience-based guideline. *Empirical Software Engineering*, *22*, 2852–2891. doi: 10.1007/s10664-016-9492-y

Laga, N., Bertin, E., Crespi, N., Bedini, I., Molina, B., & Zhao, Z. (2013). A flexible service selection for executing virtual services. *World Wide Web-Internet and Web Information Systems*, *16*(3), 219–245. doi: 10.1007/s11280-012-0184-2

Li, Y., & Jiang, Y. (2016). Self-Organization Based Service Discovery Approach Considering Intermediary Utility. In S. ReiffMarganiec (Ed.), *IEEE 23rd International Conference on Web Services (ICWS)* (pp. 308–315). doi: 10.1109/ICWS.2016.47

Liu, J., Jiang, J., Cui, X., Yang, W., & Liu, X. (2015). Power consumption prediction of web services for energy-efficient service selection. *Personal and Ubiquitous Computing*, *19*(7), 1063–1073. doi: 10.1007/s00779-015-0887-3

Ma, S.-P., Chen, Y.-J., Syu, Y., Lin, H.-J., & Fanjiang, Y.-Y. (2021). Test-Oriented RESTful Service Discovery with Semantic Interface Compatibility. *IEEE Transactions on Services Computing*, *14*(5), 1571–1584. doi: 10.1109/TSC.2018.2871133

Madiwalar, B., Schneider, B., & Profanter, S. (2019). Plug and Produce for Industry 4.0 using Software-defined Networking and OPC UA. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 126–133). doi: 10.1109/ETFA.2019.8869525

Markus, L. M. (2001). Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success. *Journal of Management Information Systems*, *18*(1), 57–93. doi: 10.1080/07421222.2001.11045671

Olson, A. M., Raje, R. R., Devaraju, B., & Gallege, L. S. (2015). Learning improves service discovery. *Concurrency and Computation-Practice & Experience*, *27*(7), 1679–1694. doi: 10.1002/cpe.3323

Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2007). Service-Oriented Computing: State of the Art and Research Challenges. *Computer*, *40*(11), 38–45. doi: 10.1109/MC.2007.400

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)* (pp. 68–77). BCS Learning & Development. doi: 10.14236/EWIC/EASE2008.8

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

Platenius, M. C., Becker, S., & Schaefer, W. (2014). Integrating Service Matchers into a Service Market Architecture. In P. Avgeriou & U. Zdun (Eds.), *8th European Conference on Software Architecture (ECSA)* (Vol. 8627, pp. 210–217). doi: 10.1007/978-3-319-09970-5_19

Pourghebleh, B., Hayyolalam, V., & Anvigh, A. A. (2020). Service discovery in the Internet of Things: review of current trends and research challenges. *Wireless Networksg*, *26*, 5371–5391. doi: 10.1007/s11276-020-02405-0

Ristoski, P., & Paulheim, H. (2016). Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Journal of Web Semantics*, *36*, 1–22. doi: 10.1016/j.websem.2016.01.001

Rodriguez, B. H., & Moissinac, J.-C. (2015). Discovery and registration of components in multimodal systems distributed on the IoT. *Journal of Visual Languages and Computing*, *31*, 150–159. doi: 10.1016/j.jvlc.2015.10.007

Rodriguez, G., Soria, A., & Campo, M. (2016). Artificial intelligence in service-oriented software design. *Engineering Applications of Artificial Intelligence*, *53*, 86–104. doi: 10.1016/j.engappai.2016.03.009

Ruta, M., Scioscia, F., Loseto, G., & Di Sciascio, E. (2017). A Semantic-Enabled Social Network of Devices for Building Automation. *IEEE Transactions on Industrial Informatics*, *13*(6), 3379–3388. doi: 10.1109/TII.2017.2697907

Sametinger, J. (1997). *Software engineering with reusable components*. Springer Science & Business Media.

Schwichtenberg, S., Gerth, C., Huma, Z., & Engels, G. (2014). Normalizing Heterogeneous Service Description Models with Generated QVT Transformations. In J. Cabot & J. Rubin (Eds.), *10th European Conference on Modelling Foundations and Applications (ECMFA)* (Vol. 8569, pp. 180–195). doi: 10.1007/978-3-319-09195-2_12

Sciullo, L., Gigli, L., Trotta, A., & Di Felice, M. (2020). WoT Store: Managing resources and applications on the Web of Things. *Internet of Things*, *9*, 100164. doi: 10.1016/j.iot.2020.100164

Sellami, M., Gaaloul, W., & Defude, B. (2015). A decentralized and service-based solution for data mediation: the case for data providing service compositions. *Concurrency and Computation-Practice & Experience*, *27*, 1427–1444. doi: 10.1002/cpe.3048

Serena, F., Poveda-Villalon, M., & Garcia-Castro, R. (2018). Semantic Discovery in the Web of Things. In I. Garrigos & M. Wimmer (Eds.), *17th International Conference on Web Engineering (ICWE)* (Vol. 10544, pp. 19–31). doi: 10.1007/978-3-319-74433-9_2

Shanahan, H., Hoebelheinrich, N., & Whyte, A. (2021). Progress toward a comprehensive teaching approach to the FAIR data principles. *Patterns*, *2*(10), 100324. doi: 10.1016/j.patter.2021.100324

Singh, M. P., & Huhns, M. N. (2005). *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons.

Son, H., Kim, B., Kim, T., Lee, D., & Hyun, S.-J. (2015). Sherlock-SD: A Light-Weight Universal Service Discovery for Web of Things (WoT) Services. In *12th IEEE Consumer Communications and Networking Conference (CCNC)* (pp. 276–282). doi: 10.1109/CCNC.2015.7157989

Stroulia, E., & Wang, Y. (2005). Structural and semantic matching for assessing web-service similarity. *International Journal of Cooperative Information Systems*, *14*(04), 407–437. doi: 10.1142/S0218843005001213

Szyperski, C., Gruntz, D., & Murer, S. (2002). *Component Software: Beyond Object-Oriented Programming*. Pearson Education.

Tanganelli, G., Vallati, C., & Mingozzi, E. (2018). Edge-Centric Distributed Discovery and Access in the Internet of Things. *IEEE Internet of Things Journal*, *5*(1), 425–438. doi: 10.1109/JIOT.2017.2767381

Tran, N. K., Sheng, Q. Z., Babar, M. A., & Yao, L. (2017). Searching the Web of Things: State of the Art, Challenges, and Solutions. *ACM Computing Surveys*, *50*(4), 1–34. doi: 10.1145/3092695

Vale, T., Crnkovic, I., de Almeida, E. S., da Mota Silveira Neto, P. A., Cavalcanti, Y. C., & de Lemos Meira, S. R. (2016). Twenty-eight years of component-based software engineering. *Journal of Systems and Software*, *111*, 128–148. doi: 10.1016/j.jss.2015.09.019

Venkatesh, P. K., Wang, S., Zou, Y., & Ng, J. W. (2017). A Personalized Assistant Framework for Service Recommendation. In X. Q. Liu & U. Bellur (Eds.), *IEEE International Conference on Services Computing (SCC)* (pp. 92–99). doi: 10.1109/SCC.2017.20

Verma, R., & Srivastava, A. (2018). A dynamic web service registry framework for mobile environments. *Peer-to-Peer Networking and Applications*, *11*(3), 409–430. doi: 10.1007/s12083-016-0540-6

Xu, C., & Yu-Shi, Z. (2016). Research on Semantic Discovery Technology of Parts Based on Knowledge Reuse. In *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 01, pp. 33–36). doi: 10.1109/IHMSC.2016.14

Zeshan, F., Mohamad, R., Ahmad, M. N., Hussain, S. A., Ahmad, A., Raza, I., … Babar, I. (2017). Ontology-based service discovery framework for dynamic environments. *IET Software*, *11*(2), 64–74. doi: 10.1049/iet-sen.2016.0048

Zhang, N., Wang, J., Ma, Y., He, K., Li, Z., & Liu, X. F. (2018). Web service discovery based on goal-oriented query expansion. *Journal of Systems and Software*, *142*, 73–91. doi: 10.1016/j.jss.2018.04.046

Zhao, Y., Tan, W., & Zhao, L. (2018). Blockchain-Based UDDI Data Replication and Sharing. In W. Shen, J. Luo, J. P. Barthes, F. Dong, J. Zhang, & H. Zhu (Eds.), *22nd IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 384–389). doi: 10.1109/CSCWD.2018.8465396

Zisman, A., Spanoudakis, G., Dooley, J., & Siveroni, I. (2013). Proactive and Reactive Runtime Service Discovery: A Framework and Its Evaluation. *IEEE Transactions on Software Engineering*, *39*(7), 954–974. doi: 10.1109/TSE.2012.84

About the authors

Javier Criado is an Associate Professor at the Department of Informatics and the Applied Computing Group (ACG), University of Almería (Spain). His main research areas are model-driven engineering, component-based software engineering, user interfaces, interoperability, service-oriented architectures, microservices and web of things. You can contact the author at javi.criado@ual.es.

Luis Iribarne is a Full Professor at the Department of Informatics and the Applied Computing Group (ACG), University of Almería (Spain). His main research areas are simulation, software components, model-driven, service and software engineering, interoperability, smart cities and web of things. You can contact the author at luis.iribarne@ual.es.