RESEARCH IN PROGRESS

COMPONENT MODELS FOR IOT SEARCH ENGINE

VAIDAS GIEDRIMAS

Panevėžio kolegija/State Higher Education Institution, Panevežys, Lithuania
vaidas.giedrimas@panko.lt

The more distributed system expands, the higher need for search engines of its elements we have. The Internet of Things (IoT) systems become very complex, and the number of devices is growing exponentially. The demand for the search engine of things now reached the demand for the search engine of web pages as it was in the 1990s. We propose to use component-based architecture for the search engine of things (IoTSE). As IoT systems are heterogeneous, and the interoperability of various component models is problematic, we must focus on selected component models only. This paper surveys existing component models and discuss their feasibility to be used in IoT search engine.

Keywords:
search engine, internet of things, component model, component-based software engineering

University of Maribor Press

DOI https://doi.org/10.18690/um.fov.6.2023.51
ISBN 978-961-286-804-8

1 Introduction

The Internet of Things (IoT) is an example of a complex computational system. It evolved from the idea in the 1990s to the infrastructure of everyday life nowadays. The number of IoT devices is growing exponentially, so the demand for the search engine of "things" is as big as for the search engine of webpages 3 decades ago (Gubbi et al. 2013; Fathy, Y., Barnaghi, P., Tafazolli, 2018; Tran et al. 2017; Tang et al. 2023). They are a pilot products already such as *IoTCrawler* and *Shodan*.

However, the development of IoT search engines is not entirely the same as the development of "ordinary" search engines. It exists some challenges and threats. When webpages are uniform to some extent, the IoT network is highly heterogeneous and dynamic. A typical web search engine's data source is the web pages, metadata, and content. IoT content is far more complex as they provide very dynamic data. Instead of static content they usually provide the data streams. On the other hand, even the type of data can change. For example, a temperature sensor can provide floated point values (temperature), when it is working, Boolean value (false) when it is out of order, or the textual content (via its representative). Because of the nature of IoT, we cannot just reuse the principles of web page search engines.

In software engineering, it is common practice to use reusable software components and/or software services when it is required to address dynamism and agility. Component-based software engineering (CBSE) uses modules of higher granularity of the components. This enables to reduction the time required for the development, testing of component-based software significantly. Components are "hot swap" modules in software similar to the storage disks, power supplies, and other modules in contemporary hardware (mainly servers). As all the communication between software components is organized via Interfaces, each software component can be easily and quickly replaced by another one, which implements the same interface. After this, there is no need to perform testing of the overall system. A component-based system can be considered highly reliable and easily maintainable (Vale et al., 2016).

The goal of this paper is to survey existing component models and discuss their feasibility to be used in IoT search engines (IoTSE). The rest of the paper is organized as follows: Section 2 presents the main architectural parts of search engines and the state-of-art taxonomy of IoT search engines. Section 3 surveys the most popular software component models (CORBA, EJB, .NET, OSGi) and discusses their place in the overall architecture of dynamic Internet of things search engine. Section 4 shortly introduces related work in this area. Finally, the conclusions are made, and open questions are discussed.

2 Search engine concept in the IoT context

As presented by Brin and Page (1998), and by Bruce et al. (2010), main parts of the search engine are as follows:

1. A *Crawler*, or a bot, is a program that traverses the web to discover and download web pages. This is a critical component of a search engine because it enables the system to discover new web pages and update its index of existing ones.
2. *Indexer* : Once the crawler has downloaded a web page, the next step is to extract its content and store it in an index. An indexer is a program that processes the content of web pages and creates an index of words and their locations. This allows users to find relevant pages quickly by searching for specific words.
3. The *Query processor* is responsible for interpreting user queries and finding the most relevant web pages. This component usually involves a combination of natural language processing, machine learning, and other algorithms to match user queries with relevant pages in the index.
4. A *Ranking algorithm* is used to determine the order in which web pages are presented to users. Different search engines use different algorithms, but the basic idea is to assign a score to each page based on factors such as relevance, authority, and popularity.
5. The *User interface* is the part of the search engine that users interact with. This includes the search box, search results page, and other elements that make it easy for users to find what they're looking for.

Each IoT *search engine* must be able to perform at least two activities: discovery activity and search activity. During the *discovery activity* engine retrieves the list of IoT devices by scanning the local/global environment, IoT device-related websites, or cached databases. According to (Tran et al. 2019), more than 90% of the engines include discovery activities. After this list is made, the second activity – searching – is running. Its goal is to filter the former list by rejecting all the devices whose properties do not conform with the initial query. IoT search engines can be classified according to their meta-path. Meta-path reveals the source of the data for discovery and search and the target of these activities. Nguyen et al. (2019) distinguish 8 classes of IoTSE:

1. $R \rightarrow R$ class IoTSE
2. $D + R \rightarrow T \rightarrow R$ class
3. $D \rightarrow D$
4. $R \rightarrow T \rightarrow R + D$
5. $D \rightarrow T \rightarrow R$
6. $F \rightarrow F$
7. $R \rightarrow T \rightarrow F$
8. $S \rightarrow S$

where R is Representative of IoT device (e.g. dedicated web page for its monitoring and control), D - Dynamic IoT content (e.g. not usual data stream), T - IoT device (aka Thing).

1st class (R-R) of IoT search engines is highly influenced by web search engines. The information about devices is taken from representatives only. No direct data gathering from IoT devices at the moment of search is performed.

The second class (D+R-TR) of search engines gets data not only from representatives of things but (what is more specifically) from the IoT devices directly as dynamic streams. The result of 2nd class IoTSE is the representatives. In other words, R-R class can be considered as a subset of D+R-TR, whit zero D component. 3rd class (D-D) goes beyond of D+R-TR class and gets the data from the very end sensors of Things, not just from the public observable states of things. 4th class (R-T-R+D), according to (Tran et al., 2019), is an extension of the R-R class. What is interesting, is that search engines from this class return not only links to IoT

representatives but the data stream from them as well. 5th class (D-T-R) is a little bit similar to 2nd and 3rd classes. Typically they work using SPARQL queries over contextual information. As (Tran et al., 2019) observed in contrast with 2nd class IoT search engines, 5th class engines select things only by the data streams, not considering any other features (e.g. representatives). 6th class (F-F) is very interesting as the IoT devices are analyzed not by their representatives or data stream, but by their functionality. The functionality of things is exposed in a shared ontology. 7th class (RTF) of engines selects devices by its representatives. Its results include the functionality of selected things. As pointed out by Tran et al. (2019), this functionality is presented as RESTful Web services and is capable of self-describing with specifications written in the Web Application Description Language (WADL). In our context (CBSE) it is a very promising fact. And 8th class of IoTSE (SS) query on cached static information from IoT devices. In contrast to 3rd class, these IoTSE, do not acquire the newest data from IoT devices nor perceive dynamic data streams. However, then can be easily applied in the fields where the device state does not change often. The examples of IoT search engines are listed in Table 1.

Table 1: IoT search engines

Engine	Class	Source	URL
Shodan	N/A		https://www.shodan.io
ThingFul	DRTR?		https://www.thingful.net
IoTCrawler	DD?		https://iotcrawler.eu
ForwarDS-IoT	RR	(Gomes et al., 2015)	
DiscoWoT	RR	(Mayer, Guinard, 2011)	
ThingSeek	N/A	(Shemshadi, Sheng, Qin, 2016)	
IoT-SVK	DRTR	(Jin et al., 2011)	
CASSARAM	DD	(Perera, 2013)	
Snoogle	RT-R+D	(Wang, Tan, Li, 2010)	
ASAWoO	FF	(Mrissa, Médini, Jamont, 2014)	https://liris.cnrs.fr/asawoo
N/A	RTF	(Kamilaris, Papakonstantinou, Pitsillides, 2014)	
Microsearch	SS	(Tan et al., 2009)	

Table 2: The coverage of IoTSE functions

Function	CORBA	EJB	OSGi	.NET	WS
Crawler					
Indexer		(Arellanes, Lau, 2020)		(Fathy, Barnaghi, Tafazolli, 2018)	
QP					
RA	(Chang, Yuan, Lo, 2000)				
UI	(Chang, Yuan, Lo, 2000)				

3 Component-based approach

The component in our context is a reusable software module, having explicit interfaces, capable to discover IoT devices and/or getting their data. It is the subject of the third part of the composition and is strictly dependent on the middleware/framework.

The idea to combine IoT with software components and services is not very new. The group of Zhiming Liu (2010) has been working on software-hardware components fusion for a long time before the rise of IoT. In another reference, Ruppen et al. (2015) present a system based on model-driven architecture (MDA) which helps to bind IoT devices and RESTful web services in a semi-automated way. The component in (Ruppen et al., 2015) work is the pair of IoT devices - RESTFull service. In both cases, the component is a "piece" of the system having its hardware and software parts. In CBSE each component must have at least one interface (or Provided interface) in which it is implemented, and (optionally) can have a Required interface, which declares what interfaces are needed from other components to work properly. Distributed components communicate using a Framework, or Middleware, which encompasses main data structures, and services (e.g. networking, DB) as well.

In this section, we continue the architectural approach described in (Gula, Flakova, 2017; Giedrimas, Backys, 2022; Saari, Nurminen, Rantanen, 2022) and will analyze the most popular component-based technologies: CORBA, EJB, OSGi, and .NET.

3.1 CORBA

CORBA (Common Object Request Broker Architecture) is a middleware technology that provides a way for distributed applications to communicate with each other. CORBA was an industrial standard for the past two decades. They are still several legacy systems using this technology. It can be used in the development of search engines to connect different components of the system and provide a standard way for them to communicate. The use of the CORBA component model for web search engines is described in (Chang, Yuan, Lo, 2000). CORBA is supported by OMG consortium, which gives grates opportunities to embed CORBA components into any existing system. However, CORBA is only one technology that does not separate the processes of component development and the use of the components. CORBA components usually are not a subject of third-party composition. This does not let to use the "hot swap" advantage of CBSE.

Table 3: The coverage of IoTSE functions

Class	CORBA	EJB and OSGi	.NET	WS
RR	(Chang, Yuan, Lo, 2000)			
DRTR	(Chang, Yuan, Lo, 2000)			(Dang, Pham, Duong, 2018)
DD		(Arellanes, Lau, 2020)		(Arellanes, Lau, 2020; Dang, Pham, Duong, 2018)
RTRD	(Chang, Yuan, Lo, 2000)			
DTR		(Arellanes, Lau, 2020)		(Arellanes, Lau, 2020; Dang, Pham, Duong, 2018)
FF				(Dang, Pham, Duong, 2018)
RTF		(Arellanes, Lau, 2020)		(Arellanes, Lau, 2020)
SS				(Tan et al., 2009)

3.2 EJB and OSGi

EJB (Enterprise Java Beans) is a server-side component architecture for Java that can be used for building distributed applications, including search engines. EJB provides a set of services such as transaction management, security, and persistence, which can be useful in building complex distributed applications.

OSGi (Open Services Gateway initiative) is a modular framework for Java that provides a way to build complex applications using a set of reusable components. It can be used in the development of search engines to create a modular architecture that allows components to be developed and tested independently.

3.3 .NET

.NET is a software framework developed by Microsoft that can be used for building web applications, including search engines. It includes a variety of libraries and tools for developing web applications, such as ASP.NET for building web pages and MVC for creating web applications.

3.4 Web services

Web services provide a standardized way for software components to communicate with each other over the internet, using technologies such as XML, SOAP, JSON. This makes it easier to integrate different components into a single system, which is important for building complex IoT search engines. Microservices are a more recent approach to software architecture, where a system is broken down into small, independently deployable components that communicate with each other via lightweight protocols such as HTTP or messaging systems. This approach allows for greater flexibility and scalability in building IoT search engines, as different microservices can be developed and deployed independently and can be easily replaced or scaled up or down based on demand. We refer to both technologies as to WS. In analyzed papers most of pilot projects on IoTSE was made using web services and microservices.

3.6 Frameworks

As mentioned earlier, in component-based software engineering we rely not only on the components but on the frameworks as well. The most often used frameworks are:

1. Lightweight M2M (LwM2M): LwM2M is a protocol that is designed for use in IoT devices. It provides a standardized way for devices to communicate with servers, which can be useful for integrating data from IoT devices into a search engine.
2. CoAP protocol that is designed for use in IoT devices. It is a lightweight protocol that is well-suited for constrained environments and can be used to integrate IoT device data into a search engine.
3. Apache Kafka is a distributed streaming platform that is commonly used in IoT contexts. It can be used to ingest data from IoT devices in real-time and can be integrated with search engines to provide real-time search capabilities.
4. Apache Spark is a popular distributed computing platform that can be used for processing large volumes of data in near-real-time. It can be used to process data from IoT devices and integrate it with search engines.
5. Elasticsearch for IoT : Elasticsearch is a popular search engine that has been adapted for use in IoT contexts. It can be used to store and search data from IoT devices, and provides a variety of features that are optimized for IoT use cases, such as geospatial search and time series data.

4 Related work

The architectures of component-based IoTSE are described only in (Giedrimas, Backys, 2022; Gula, Flakova, 2017). Other papers cover this topic only partially. The idea to use the (reusable) workflow engines in the overall science gateway is architecturally like our approach. Glatard et al. (2017) refer to workflow engines as software components while presenting 6 software architectures: Tight integration, Service invocation, Task encapsulation, Pool model, Nested workflows with service invocation and Conversion, and Workflow conversion with service invocation. We assume that component-based IoTSE will have a backbone (instead of the science

gateways), and smaller parts - software components implementing different IoT search and discovery algorithms.
Other papers have weak (or negative) relations with software components; however, they are highly related to software architectures (Cambazoglu et al., 2007; Ozcan et al., 2012) and/or search engines (Kejriwal, 2021; Cambazoglu et al., 2007; Ozcan et al., 2012).

5 Conclusions and Future work

After the analysis, we came to the following conclusions:

1. Even though the market of IoTSEs is new and growing, enough different search engines is developed already. Fortunately, IoTSEs are not unique and can be classified into a limited number of classes. Each class of the IoTSE can deny an interface and each IoTSE from this class can be considered as a component implementing this interface.
2. The area of research (component models for IoTSE), is almost not covered by the scientific community nor the business applications. We found only a few sources. Surprisingly search engines provide some results about the topic, which led to broken links and non-existing pages. We assume that the papers were withdrawn because of the lack of maturity or experimental data.
3. The survey of the 5 most popular component-based models shows that most of the pilot projects on IoTSE were made using web services and microservices. More hardware-related component models such as OSGi and CORBA are least used for this purpose.
4. We do not find any evidence of component-based technology used for the development of the Crowler and Query processing elements of search engines.

Our future work on this topic includes the selection of one component model and experimental development of the IoTSE system.

References

Arellanes, D., Lau, K.K.: Evaluating iot service composition mechanisms for the scalability of iot systems. Future Generation Computer Systems 108, 827848 (2020). https://doi.org/https://doi.org/10.1016/j.future.2020.02.073,

Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1-7), 107117 (1998)

Cambazoglu, B.B., Karaca, E., Kucukyilmaz, T., Turk, A., Aykanat, C.: Architecture of a grid-enabled web search engine. Information Processing and Management 43(3), 609623 (2007). https://doi.org/https://doi.org/10.1016/j.ipm.2006.10.011, Special Issue on Heterogeneous and Distributed IR

Chang, Y.S., Yuan, S.M., Lo, W.: A new multi-search engine for querying data through an internet search service on corba. Computer Networks 34(3), 467480 (2000). https://doi.org/https://doi.org/10.1016/S1389-1286(00)00131-6,

Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice. Addison-Wesley Professional (2010)

Dang, T., Pham, L.T., Duong, T.Q.: Building an intelligent search engine for iot using microservices architecture. Journal of Computer Networks and Communications 2018 (2018). https://doi.org/10.1155/2018/9259473,

Fathy, Y., Barnaghi, P., Tafazolli, R.: Large-scale indexing, discovery, and rankingfor the internet of things (iot). ACM Comput. Surv. 51(2) (Mar 2018).https://doi.org/10.1145/3154525

Giedrimas, V., Backys, G.: Component-based architecture of IoT search engine. Taikomieji tyrimai studijose ir praktikoje – Applied research in studies and practice 18(1), 100105 (Dec 2022),https://ojs.panko.lt/index.php/ARSP/article/view/183

Glatard, T., Étienne Rousseau, M., Camarasu-Pop, S., Adalat, R., Beck, N., Das, S., da Silva, R.F., Khalili-Mahani, N., Korkhov, V., Quirion, P.O., Rioux, P., Olabarriaga, S.D., Bellec, P., Evans, A.C.: Software architectures to integrate workflow engines in science gateways. Future Generation Computer Systems 75, 239255 (2017). https://doi.org/https://doi.org/10.1016/j.future.2017.01.005

Gomes, P., Cavalcante, E., Rodrigues, T., Batista, T., Delicato, F.C., Pires, P.F.: A federated discovery service for the internet of things. In: Proceedings of the 2nd Workshop on Middleware for Context-Aware Applications in the IoT. p. 2530. M4IoT 2015, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2836127.2836129

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems-The International Journal of Escience 29(7), 16451660 (SEP 2013). https://doi.org/{10.1016/j.future.2013.01.010}

Gula, M., Fláková, K.: Proposal of component based architecture for internet of things: online laboratory case study. IFAC-PapersOnLine 50(1), 337342 (2017). https://doi.org/https://doi.org/10.1016/j.ifacol.2017.08.153, 20th IFAC World Congress

Jin, X., Zhang, D., Zou, Q., Ji, G., Qian, X.: Where searching will go in internet of things? In: 2011 IFIP Wireless Days (WD). pp. 13 (2011).

Kamilaris, A., Papakonstantinou, K., Pitsillides, A.: Exploring the use of dns as a search engine for the web of things. In: 2014 IEEE World Forum on Internet of Things (WF-IoT). pp. 100105 (2014). https://doi.org/10.1109/WFIoT. 2014.6803128

Kejriwal, M.: A meta-engine for building domain-specific search engines. Software Impacts 7, 100052 (2021). https://doi.org/https://doi.org/10.1016/j.simpa.2020.100052

Liu, Z., Morisset, C., Stolz, V.: rcos: Theory and tool for component-based model driven development. In: Arbab, F., Sirjani, M. (eds.) Fundamentals of Software Engineering. pp. 6280. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

Mayer, S., Guinard, D.: An extensible discovery service for smart things. In: Proceedings of the Second International Workshop on Web of Things. WoT '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/1993966.1993976

Mrissa, M., Médini, L., Jamont, J.: Semantic discovery and invocation of functionalities for the web of things. In: 2014 IEEE 23rd International WETICE Conference. pp. 281286 (2014). https://doi.org/10.1109/WETICE.2014.50

Ozcan, R., Sengor Altingovde, I., Barla Cambazoglu, B., Junqueira, F.P., Özgür Ulusoy: A ve-level static cache architecture for web search engines. Information Processing and Management 48(5), 828840 (2012). https://doi.org/10.1016/j.ipm.2010.12.007, large-Scale and Distributed Systems for Information Retrieval

Perera, C., Zaslavsky, A., Christen, P., Compton, M., Georgakopoulos, D.: Contextaware sensor search, selection and ranking model for internet of things middleware. In: 2013 IEEE 14th International Conference on Mobile Data Management. vol. 1,pp. 314322 (2013). https://doi.org/10.1109/MDM.2013.46

Ruppen, A., Pasquier, J., Meyer, S., Rüedlinger, A.: A component based approach for the web of things. In: Proceedings of the 6th International Workshop on the Web of Things. WoT '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2834791.2834792

Saari, M., Nurminen, M., Rantanen, P.: Survey of component-based software engineering within iot development. In: 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO). pp. 824828 (2022). https://doi.org/10.23919/MIPRO55190.2022.9803785

Shemshadi, A., Sheng, Q.Z., Qin, Y.: Thingseek: A crawler and search engine for the internet of things. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 11491152. SIGIR '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2911451.2911471

Tan, C.C., Sheng, B., Wang, H., Li, Q.: Microsearch: When search engines meet small devices. In: Proceedings of the 6th International Conference on Pervasive Computing. p. 93110. Pervasive '08, Springer-Verlag, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-79576-6-6

Tang, J., Lu, X., Xiang, Y., Shi, C., Gu, J.: Blockchain search engine: Its current research status and future prospect in internet of things network. Future Generation Computer Systems 138, 120 141 (2023). https://doi.org/https://doi.org/10.1016/j.future.2022.08.008,

Tran, N., Sheng, Q., Babar, M., Yao, L.: A kernel-based approach to developing adaptable and reusable sensor retrieval systems for the web of things. In: 18th International Conference on Web Information Systems Engineering (WISE 2017) : proceedings. Lecture Notes in Computer Science, vol. 10569, pp. 315329. Springer, Springer Nature, United States (2017). https://doi.org/10.1007/978-3-319-68783-4-22

Tran, N.K., Sheng, Q.Z., Babar, M.A., Yao, L., Zhang, W.E., Dustdar, S.: Internet of things search engine. Commun. ACM 62(7), 6673 (Jun 2019).

Vale, T., Crnkovic, I., de Almeida, E.S., Silveira Neto, P.A.d.M., Cavalcanti, Y.a.C., Meira, S.R.d.L.: Twenty-eight years of component-based software engineering. J. Syst. Softw. 111(C), 128148 (Jan 2016). https://doi.org/10.1016/j.jss.2015.09.019

Wang, H., Tan, C.C., Li, Q.: Snoogle: A search engine for pervasive environments. IEEE Transactions on Parallel and Distributed Systems 21(8), 11881202 (2010). https://doi.org/10.1109/TPDS.2009.145