Context-Aware Service Oriented Architecture for Secure Data Transmission

[1]**Shubham Joshi**
Assistant Professor, Computer Engineering, SVKM's NMIMS MPSTME Shirpur
[2]**Radha Krishna Rambola**
Associate Professor, Computer Engineering, SVKM's NMIMS MPSTME Shirpur

Abstract— Context-aware systems open up completely new opportunities for application developers and End users collect and transfer contextual data and adjust system behavior accordingly. Especially the combination of mobile devices that increase the high price of these mechanisms and usability has made a huge leap. In this paper, we present Context-aware service-oriented architecture for secure data transmission. Design based on the principles we have developed a number of existing contextual knowledge systems, which focuses on the context of knowledge systems, and frames, and we have developed a framework for the implementation of context-aware middleware. We deal with the context-aware end-to-end data delivery services adaptive composition of middlebox services when dynamically service provider than virtual functions. We can achieve the deployment of network service chain adapted by it and enable programmability of the data delivery path through a dynamically established sequence of virtual middlebox functions. It also includes specifications for processing functions that must be passed by service data, while taking advantage of contextual awareness and self-adaptation capabilities. We discuss various approaches and analyze important aspects of context-aware computing, based on the systems presented. The effectiveness of the proposed approach to distinguish service data processing is to be investigated using an experimental test when using optimization transfer tables in switches.

Index Terms— Context-aware system, Service-oriented system, Data transmission, sensors, Context-aware services, Context management, Context awareness with services.

1 Introduction

The increasing use of mobile computing devices today users around the world these devices are already part of the everyay life of carriers. Contrary to the limitations of its predecessors, many current mobile devices have decent computing power, a large amount of vast memory, and can be used locally and globally. The amount of communication required with mobile apps can be reduced in some situations by taking advantage of contextual information about users and devices. Sensor hardware, which can be used to collect contextual information, is highly accessible and increasingly available on mobile standard devices are introduced by T. Erl, et al. [1] and W. John, et al. [3]. We can use low-level sensing data, such as physical location and temperature, to infer high-level contextual information, such as situations (e.g. "sunbathing on the beach"). The user experience of mobile apps dependent on contextual information can be greatly enhanced by providing these apps with contextual information derived automatically from sensor data.

Service-oriented architecture (SOA) is a loosely coupled, distributed software component accessible through platform-independent protocols and interfaces, T. Erl, et al. [1] and Y. Qin, et al. [4]. The methods in the service are invoked in various applications. The main goal of such architecture is to service

the provision of functions to the industry, mainly considering business processes. Deployed only once for use by multiple applications. Concepts, context-conscious computing, and the service-oriented architectural paradigm, can take mutual advantage of each other. Context-aware system summarizes the raw sensor data and uses context data to provide higher level background information for other functional parts of the system.

Context recognition and adaptation are closely related, and these two terms are often used as synonyms. Adaptation is the ability to provide a different version of the service or presentation of different products. A document to meet the needs of the user, environment, equipment, and etc. contextual awareness is the ability to perceive the user's situation in many of its aspects and, as a result, adapt the behavior of the system. A context-oriented application should manage the context as one of its input data, processing each user's request according to different context cases. However, it is easier to think of context as input for a single application, since it is difficult to classify all combinations of user situations, devices, and other context-sensitive parameters in advance. Therefore, the context should be managed separately, and its influence on the application behavior should be described orthogonally in relation to the application data, Z. Laliwala, et al. [5].

The service-oriented paradigm helps in creating intermediate software solutions that make certain pieces of contextual information available to the components they relate to, regardless of the physical distribution of the sensors from which this information comes. Access to contextual information at various levels of abstraction can be published as a service, and these services can be compiled to provide higher-level contextual information again for publication as a service. On the other hand, service-oriented architectures can be improved by contextual awareness, Y. Qin, et al. [4] and Riaz. M, et al. [9].

In this paper systems that combine context-awareness and service-orientation for data transmission. Each of these systems provides a certain type of context Management, which we check in more detail. Some of them use context management to build context-conscious solutions used in service-oriented architectures. The proposed system is validated through the experimental operation performed by tested. The results show that the proposed system is feasible.

2 Related Work

Currently, context awareness has become essential for software applications and services due to the high demand of users, especially for mobile applications. This need to provide context awareness requires the software infrastructure not only to obtain context information but also to use it, so it provides favorable services that can be customized according to the user's needs. The author V. A. Immanuel, et al. [10] is to apply context-sensitive computing using a service-oriented architecture to obtain, analyze, and assist physicians and nurses with the necessary information. B. A. Kumar [11] proposed approach selects a web service based on user context from experimental results, and the proposed method provides a reliable solution in the context of known services. This research can be utilized for the development of large-scale dynamic adaptive applications in the future. Y. Yengı, et al. [12] introduces the service-oriented context recognition architecture for the public and the provision of services in the Internet of things. A. Garcia De Prado, et al. [13] describes event-based services supported by the enterprise service bus that facilitate the inclusion of IOT data and the provision of context-sensitive services in real time.

G. Ortiz, et al. [14] the authors define and implement a micro services-based architecture that provides context-sensitive real-time actions based on predicted streaming data processing. The purpose of paper written by author, Á. R. Lopes, et al. [15] is to develop a context-aware ubiquitous learning environment that is characterized by the ability to acquire and use information from the user's context. This makes it possible to learn the context based on the adaptive behavior of the software, the service, or the resource. G. Guerrero-Contreras, et al. [16] proposes context-conscious software architecture to support the availability of services used in mobile and dynamic network environments. M. Ayoubi, et al. [17], proposes a highly scalable architecture that consists of three functional elements. H. Taktak, et al. [18], evaluates how to manage service functions and user interface specifications at Design time and at runtime. They also discuss how an adaptive user interface and a model-driven approach to creating popular services can facilitate the work of designers and developers, limit incompatibility issues, and support the dynamic generation of systems adapted to different usage situations. B. Cheng, et al. [19] proposes a situation event detection algorithm based on an automaton, a situation event-driven service cooperative behavior model. K. Wan, et al. [20] proposes a resource-Aware service-oriented service model to design a system that can efficiently provide high-quality medical services.

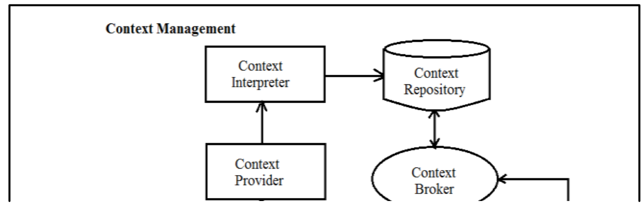
K. Cemus, et al. [21], proposes a service-oriented architecture. Such construction is adaptive in the context of service saving and easy maintenance while dealing with the reuse of crosscuts of all services. K. Lin, et al. [22], service-oriented object (SOT) paradigm that can recognize user activities and construct, map, and deploy context recognition services in an object domain. Adopt a service-oriented framework to develop context-aware interactive services for cloud robots, J. Huang, et al. [23]. N. Ibrahim, et al. [24], offers a service-oriented context-sensitive architecture for publishing, discovering, and providing services in the Internet of things. The L. Kovacs, et al. [25] designs and creates a service-oriented framework to help develop and use client-service-oriented applications that support location-and context. The name of the proposed service-oriented architecture is omnipresent, Damiao R, et al. [26]. Khouja M, et al. [27] shows the various semantic interactions that can occur in coxel. It also describes the role coxell plays in the surroundings. Lu. S [28], presents a problem of delayed response due to an exceptional event from a distributed Cognitive Service Group. They propose a new architecture to guarantee the overall consistency and real-time response of distributed Cognitive Services. The existing IoT service is proposed in the architectural context service discovery for general purposes, Sasirekha. S, et al [29].

The chang. C, et al. [30] suggests a service-oriented mobile indie-fog server architecture that allows implementing a dynamic algorithm, as well as supports distributed CaaS processing between mobile devices. Zoppi T, et al. [31] presents a context-aware anomaly detection framework that obtains information about running services and calibrates anomaly detection. Faieq S, et al. [32] proposes architecture for configuring context-aware services in a cloud environment. Kashevnik, et al. [33] proposed to build a recommendation system based on loosely connected modules, which implements both personalized and non-personalized recommendation methods, as well as a synthesis module that adapts the modular system to the specific conditions of various types of source information. The author Suchanek M, et al. [34] presents and developing machine-driven reports that can be sent between services. It's not just a way of encoding reports and composing together; it allows you to combine semantics using technologies semantic web and ontology design, mainly JSON-LD Schema.org. Corral-Plaza D, et al. [35], proposed architecture of utilizing a new technology and a cloud platform will process a huge amount of heterogeneous data and provide a way for specific applications in that context.

Li. X, et al. [36] draw attention to the activity-oriented context-sensitive application of (AOCA), where environmental requirements depending on the user actions and provide AOCA application of the developed. Guo. J, et al. [37] proposes a method for assessing the global trust in composite services. This can be used to evaluate non-performing composite services. The recommendation is based on the system for service configuration intended for a smart environment. The proposed system by Faieq. S, et al. [38] can capture the situation of the user by the analysis of the context information, and the system can choose the appropriate service model to catch the demand and satisfy the needs. Kyriazis D. [39] presents an approach that suggests using security controls as plug-ins that can be used in service-oriented environments. The latter allows users to adapt the appropriate security and privacy levels by using security measures that have been selected and implemented by themselves, which eases their security and privacy concerns.

3 Methodology

Context-aware systems can be implemented in several ways. The approach depends on specific requirements and conditions, such as location sensors (local or remote), the number of potential users (one user or many), the available resources used for the devices (high-end Pc or small mobile devices) or the extension of the facility's system. Also, the method of collecting contextual data is very important in the design of context-dependent systems, because it, at least to some extent, determines the architectural style of the system. fig. 1, shows that context-aware service-oriented architecture for secure data transmission. Where, the component for context management is separated from the operating core. Each component corresponds to the context management step described below.



1. Capturing context concerns physical sensors and the raw data they generate. This part is very device-dependent, and the overall model is difficult to build. In our architecture, we define a context provider, which is a context capture system. While the low-level views that we initially captured may not make sense for the application, high-level views are easier to interpret, use, and interpret in the context of our architecture and context interpreter module.
2. Context storage stores context values. In connection with the model parameters, we can use the storage and exchange of XML documents in connection with values. For each context address (user, network, device, metadata), we define a set of elements that contain parameter values relevant to that dataset. Defining context parameters is a function of the application designer, which also defines their syntactic and semantic structure.
3. A context-conscious statement should consume part of the context. In architecture it must subscribe to the context broker that transports the appropriate information for each service application. The subscription service tells the broker that is part of the context. The broker can then offer a context view of each service. This view can develop dynamically during execution, which requires some intelligence from the mediator. Services can collect context values whenever they are needed, or the context broker can provide a connection to subscribers whenever it is updated.
4. Context consumers must adapt to the context. They're registered first to the context broker. Customizing the context can apply to three levels: data flow (content adaptation), visualization (the user interface adaptation), and application behavior (service adaptation). All of these adaptations can be static (before run time) or dynamic (during run time). We need to use both to ensure that the context is as good as possible. Data or content and user interface adaptation into the architecture.

This means that the adaptation process consists of different combinations of services selected independently in different dimensions of the application (user interface, content, business services) shows in fig.2. The service is used to ensure that the dimensions of things are adjusted.

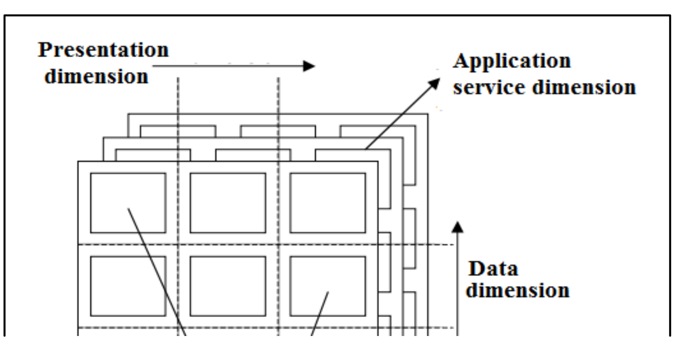


Fig 2, point 3D space defines an adaptation plan, i.e. customizing the data route, plus customizing the adaptation service route, plus customizing the presentation route. The three routes are designed to be shared, but can be handled independently. The results are compiled to adjust the context. Each service has a set of common services available, which can also be context-dependent. Common services are common adaptation services.

The user interface is a very important part of the system. It sends or presents data to the user according to the context situation in different modalities. The model is a set of business services that interact with the view and is a controller connection between the service and the view interaction component. It detects the interaction events of the view and calls the corresponding services of the model. User interface customization depends mainly on device capabilities and context user aspects. We have generated the architecture based on adaptations on run-time and pre-run adaptation rules to adapt user interfaces to these two aspects. This allows performing user interface with any service to send data. When a service is called, its inputs and outputs are also adapted to the context.

The content adaptation is customized by changing some of the features of the data provided to the user.

1. Format adaptation: this variant consists of completely changing or changing the data format. For example, or reduce the image colors or synthesize sound from the text.
2. Language translation: text can be translated into another language to adapt to the user's preferences. Translation services can offer these types of adaptations.
3. Data Compression: There are two main types of data compression: raw data compression (for example, zip) to reduce storage size and transfer time, and multimedia compression (for example, jpeg), which is directly processed on the user terminal. For text data, semantic compression can be used to calculate the document summary.
4. Data decomposition and aggregation: Data decomposition is more preferable for the user, or to extract some of the more important media. If the terminal is compatible, the rest can be displayed to the user. Users can request the aggregation of different media objects: for example, tourists can find a list of nearby restaurants.

4 Experimental Results

To verify the proposed approach, we have developed an experimental setup consisting of three Open Flow enabled emulated switches shown in fig 3. SW2 is a transit when the Switch SW1 and SW3 are also connected to the user's network and service feature, respectively. In addition, SW1 and SW3 are both connected to the emulated Cloud Platform, which contains two middlebox cases; each of them identifies a unique VLAN ID. To imitate the cloud platform we used a hub that gets the port 3 Switch and sends back to the same package to port 4 the same switch. Switches are replicated using open source software Open vSwitch version 2.61 and OpenFlow version 1.0 [40]. That communication-only uses a network controller.

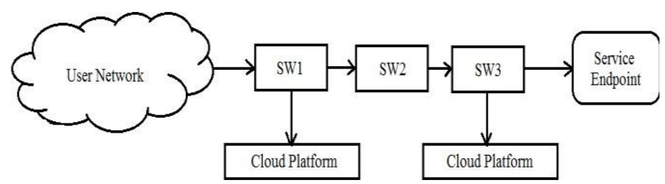


Fig. 3. Tested Setup

Based on the user's location, i.e. user classes, we linked the services as shown in Table 1, which reproduces the contents of the service chain database. The service chain database includes a service configuration specification that defines cascading logic as a workflow for middebox functionality. This information is used to select a middebox instance for each type of feature defined in the service configuration plan. The choice can be made by considering information based on context.

TABLE 1
CONTEXT-ORIENTED SPECIFICATION FOR THE COMPOSITION OF A MIDDLEBOX INSTANCE

Class	Subnet Mask	Middlebox instance chain
A	22	12
B	24	10-12
C	26	10-12-13

First, we estimated the setup time, i.e. the time from receiving the request to configure the delivery path to sending the Service response back to the application object. We repeated the test 10 times and plotted the average setup time for different service contexts and baselines (i.e., traffic passes through all available middlebox instances). From fig. 4, we can see that the installation time is on average about 370 ms. although this does not depend on the number of middlebox instances involved, the installation time increases slightly.

Second, we plotted the output Round Trip Time (RTT) experienced by the packages to compare the performance of our system with the baseline. Although such latency depends on the network we are installing, we can observe that latency decreases, and thus network usage can be optimized by differentiating packet processing while improving overall performance.

Fig. 5. RTT for a different type of classes

Fig 6 shows the average availability of the service. In which the mobile nodes act like static nodes, the availability of service increments with each addition of a new static server. Although, the maximum availability provided a static installations, 79.32 % with 4 servers nodes. fig 7, shows the efficiency of the proposed system. From this we can understand that architecture specification system for the use of the smallest number of copies and provides maximum availability according to the current state of the network.

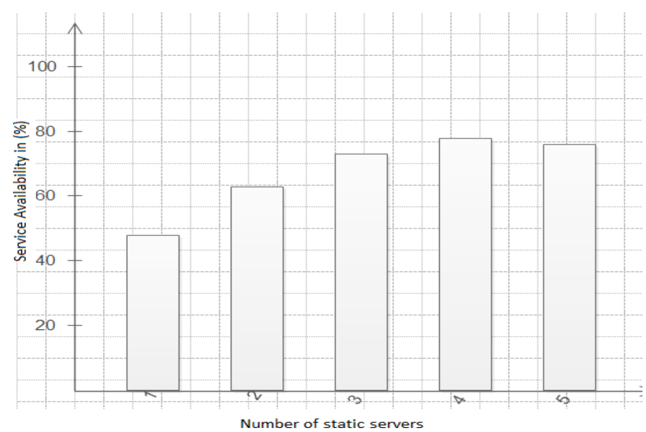


Fig. 6. Service availability of system

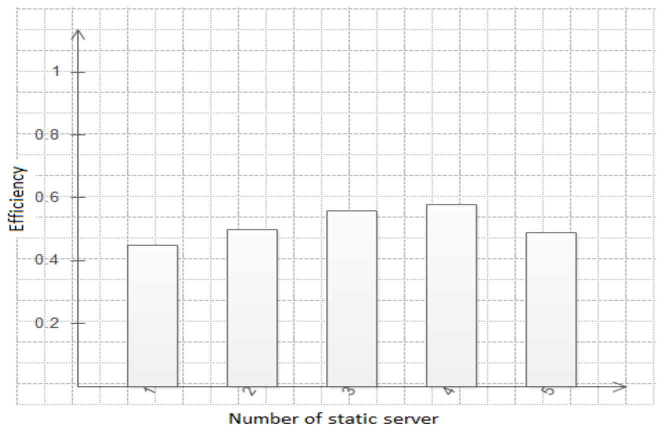


Fig. 7. Efficiency of system

5 Conclusion

In this paper, context-aware service-oriented architecture for secure data transmission is presented. The access to contextual information at various levels of abstraction can be published as a service, and these services can be compiled to provide higher-level contextual information again for publication as a service. On the other hand, service-oriented architectures can be improved by contextual awareness. We also address the problem of a dynamic service chain within service overlay networks for contextual delivery of application service data and creating new offers and, consequently, new ways for operators to monetize their networks. The results show that network usage could be optimized for a specialized number of passed middlebox cases. In the future, large-scale assessments are planned by creating an emulated network environment.

References

[1] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 2005.

[2] M. B. Juric, I. Rozman, B. Brumen, M. Colnaric and M. Hericko, “Comparison of Performance of Web Services, WS-Security, RMI, and RMI–SSL,” The Journal of Systems and Software, vol. 79, no. 5, pp. 689-700, 2006.

[3] W. John, et al., “Research Directions in Network Service Chaining”. In 2013 IEEE SDN4FNS, pp.1-7, 11-13 Nov. 2013, Trento, Italy

[4] Y. Qin, et al, "Service-oriented multi-granular optical network testbed" OFC 2009, 22-26 March 2009

[5] Zakir Laliwala, Sanjay Chaudhary, “Event-Driven Service Oriented architecture”, IEEE, Conference, Software Engineering 2008

[6] W. R. Cook and J. Barfield, “Web Services versus Distributed Objects: A Case Study of Performance and Interface Design,” Proc. of the IEEE International Conference on Web Services (ICWS), Sept. 18-22, 2006.

[7] F. Hohl, L. Mehrmann, and A. Hamdan, “A Context System for a Mobile Service Platform,” Trends in Network and Pervasive Computing, LNCS, Springer Verlag, vol. 2299, pp. 21–33, March 2002.

[8] C. Mascolo, L. Capra, W. Emmerich, “Principles of Mobile Computing Middleware,” Middleware for Communications, Wiley, 2004.

[9] Riaz, M., Kiani, S.L., Lee, S., Han, S.M., Lee, Y.K.: Service delivery in context aware environments: Lookup and access control issues. In: RTCSA ’05: Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA’05), Washington, DC, USA, IEEE Computer Society (2005)

[10] V.A. Immanuel and P. Raj, "Enabling context-awareness:A service oriented architecture implementation for a hospital use case," 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Davangere, 2015, pp. 224-228, doi: 10.1109/ICATCCT.2015.7456886.

[11] B. A. Kumar, "Context aware service selection and its application to medical emergency," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015, pp. 708-712, doi: 10.1109/NGCT.2015.7375213.

[12] Y. Yengı and K. Küçük, "Context aware internet of things for large scale data analytics," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 702-706, doi: 10.1109/UBMK.2017.8093506.

[13] A. García De Prado, G. Ortiz and J. Boubeta-Puig, "CARED-SOA: A Context-Aware Event-Driven Service-Oriented Architecture," in IEEE Access, vol. 5, pp. 4646-4663, 2017, doi: 10.1109/ACCESS.2017.2679338.

[14] G. Ortiz, J. A. Caravaca, A. García-de-Prado, F. Chavez de la O and J. Boubeta-Puig, "Real-Time Context-Aware Microservice Architecture for Predictive Analytics and Smart Decision-Making," in IEEE Access, vol. 7, pp. 183177-183194, 2019, doi: 10.1109/ACCESS.2019.2960516.

[15] Á. R. Lopes, E. F. Barbosa and R. T. V. Braga, "Service Oriented Reference Architecture for the Development of Context-aware Learning Environments," 2019 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA, 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028596.

[16] G. Guerrero-Contreras, J. L. Garrido, S. Balderas-Díaz and C. Rodríguez-Domínguez, "A Context-Aware Architecture Supporting Service Availability in Mobile Cloud Computing," in IEEE Transactions on Services Computing, vol. 10, no. 6, pp. 956-968, 1 Nov.-Dec. 2017, doi: 10.1109/TSC.2016.2540629.

[17] M.Ayoubi and S. Ravichandra, "SEDA-SOA:AScalable Event-Driven Context-Aware Service Oriented Architecture," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018, pp. 1-7, doi: 10.1109/ICCCNT.2018.8493724.

[18] H. Taktak and F. Moussa, "Context-Aware ServUI: A Conceptual Method for User Interfaces and Associated Services Generation," 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, 2017, pp. 856-862, doi: 10.1109/AICCSA.2017.172.

[19] B. Cheng, D. Zhu, S. Zhao and J. Chen, "Situation-Aware IoT Service Coordination Using the Event-Driven SOA Paradigm," in IEEE Transactions on Network and Service Management, vol. 13, no. 2, pp. 349-361, June 2016, doi: 10.1109/TNSM.2016.2541171.

[20] K. Wan and V. Alagar, "Resource-aware Service-oriented Approach for Elderly Healthcare," 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, 2018, pp. 1199-1205, doi: 10.1109/ICSAI.2018.8599300.

[21] K. Cemus, F. Klimes and T. Cerny, "Aspect-driven context-aware services," 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, 2017, pp. 1307-1314, doi: 10.15439/2017F397.

[22] K. Lin, S. Zhou, S. Yan and T. Ito, "Service-Oriented Things: Delivering Smart Services Using Internet of Things," 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, 2017, pp. 217-226, doi: 10.1109/CBI.2017.66.

[23] J. Huang, W. Lee and T. Lin, "Developing Context-Aware Dialoguing Services for a Cloud-Based Robotic System," in IEEE Access, vol. 7, pp. 44293-44306, 2019, doi: 10.1109/ACCESS.2019.2905616.

[24] N. Ibrahim and B. Bench, "Service-Oriented Architecture for the Internet of Tings," 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2017, pp. 1004-1009, doi: 10.1109/CSCI.2017.173.

[25] L. Kov´acs & P. M´at´etelki & B. Pataki, “Service-oriented Context-aware Framework”, M.H. ter Beek (Ed.): Young Researchers Workshopon Service-Oriented Computing 2009 (YR-SOC’09).EPTCS 2, 2009.

[26] Damião R. Almeida, Cláudio S. Baptista, Elvis R. da Silva, “Using Service-Oriented Architecture in Context-Aware Applications”, Springer 2015.

[27] Khouja M., Juiz C., Puigjaner R., Kamoun F, “Semantic Interactions for Context-Aware and Service-Oriented Architecture”. In: Meersman R., Tari Z., Herrero P. (eds) On the

Move to Meaningful Internet Systems: OTM 2008 Workshops. OTM 2008. vol 5333. Springer, Berlin, Heidelberg

[28] Lu S. (2019) A Goal-Driven Context-Aware Architecture for Distributing Cognitive Service Group. In: Liu X. et al. (eds) Service-Oriented Computing – ICSOC 2018 Workshops. ICSOC 2018. Lecture Notes in Computer Science, vol 11434. Springer.

[29] Sasirekha S., Swamynathan S., Keerthana S. (2018)A Generic Context-Aware Service Discovery Architecture for IoT Services, (eds) Smart Secure Systems – IoT and Analytics Perspective. ICIIT 2017. Communications in Computer and Information Science, vol 808. Springer, Singapore

[30] Chang C., Srirama S.N. "Providing Context as a Service Using Service-Oriented Mobile Indie Fog and Opportunistic Computing", ECSA 2018. Lecture Notes in Computer Science, vol11048. Springer.

[31] Zoppi T., Ceccarelli A., Bondavalli A. "Context-Awareness to Improve Anomaly Detection in Dynamic Service Oriented Architectures", In: SAFECOMP 2016. Lecture Notes in Computer Science, vol 9922. Springer, Cham

[32] Faieq S., Saidi R., Elghazi H., Rahmani M.D. (2017) A Conceptual Architecture for a Cloud-Based Context-Aware Service Composition. In: Advances in Ubiquitous Networking 2. UNet 2016. Lecture Notes in Electrical Engineering, vol 397. Springer, Singapore

[33] Kashevnik, A.M., Ponomarev, A.V. & Smimov, A.V. A multimodel context-aware tourism recommendation service: Approach and architecture. J. Comput. Syst. Sci. Int. 56,245–258 (2017). https://doi.org/10.1134/S1064230717020125

[34] Suchánek M., Slifka J. "Evolvable and Machine-Actionable Modular Reports for Service-Oriented Architecture. In: Enterprise and Organizational Modeling and Simulation. EOMAS 2019. Lecture Notes in Business Information Processing, vol 366. Springer

[35] Corral-Plaza D., Ortiz G., Boubeta-Puig J. (2018) Paving the Way for a Real-Time Context-Aware PredictiveArchitecture. In: Service-Oriented Computing – ICSOC 2017 Workshops. ICSOC 2017. Lecture Notes in Computer Science, vol 10797. Springer

[36] Li, X., Tao, X. & Lu, J. Towards a programming framework for activity-oriented context-aware applications. Front. Comput. Sci. 11, 987–1006 (2017).

[37] Guo, J., Ma, J., Guo, X. et al. Trust-based service composition and selection in service oriented architecture. Peer-to-Peer Netw. Appl. 11, 862–880 (2018).

[38] Faieq, S., Front, A., Saidi, R. et al. A context-aware recommendation-based system for service composition in smart environments. SOCA13, 341–355 (2019).

[39] Kyriazis D. (2019) Enabling Custom Security Controls as Plugins in Service Oriented Environments. In: Transactions on Computational Collective Intelligence XXXIII. Lecture Notes in Computer Science, vol 11610. Springer

[40] OpenFlow, http://www.openflow.org