SOSE4BD: Service-Oriented Software Engineering Framework for Big Data Applications

Muthu Ramachandran
School of Computing, Creative Technologies, and Engineering, Leeds Beckett University, Headingley Campus, Leeds LS6 3QS, U.K.

Keywords: Service-Oriented Software Engineering for Big Data Applications (SOSE4BD), Software Engineering Framework for Service and Cloud Computing (SEF-SCC), Cloud Software Engineering, Service-Oriented Architecture (SOA), Service Computing, Reference Architecture, Service Reuse, Software Engineering for Service and Cloud Computing (SE-Cloud), Business Process Driven Service Development Lifecycle (BPD-SDL), Business Process Modelling Notation (BPMN), Service-Oriented Architecture Modelling Language (SOAML), Quality of Service (QoS).

Abstract: Service computing has emerged to address the notion of delivering software as a service and Service-Oriented Architecture emerged as a design method supporting well defined design principles of loose coupling, interface design, autonomic computing, seamless integration, and publish/subscribe paradigm. Integrated big data applications with IoT, Fog, and Cloud Computing grow exponentially: businesses as well as the speed of the data and its storage. Therefore, it is time to consider systematic and engineering approach to developing and deploying big data services as the data-driven applications and devices increasing rapidly. This paper proposes a software engineering framework and a reference architecture which is SOA based for big data applications' development. This paper also concludes with a simulation of a complex big data Facebook application with real-time streaming using part of the requirements engineering aspect of the SOSE4BD framework with BPMN as a tool for requirement modelling and simulation to study the characteristics before big data service design, development, and deployment. The simulation results demonstrated the efficiency and effectiveness of developing big data applications using the reference architecture framework for big data.

1 INTRODUCTION

Service computing has emerged to deliver software as a service, based on established design principles of reuse, composability, autonomic computing, stateless, platform and enterprise integration. SOA is a way of architecting and structuring and designing reusable software assets: service components and resources using message passing as the core design principle to maximise reuse (design for reusable services based on the design principles of composition and scalability). In other words, make it available for reuse and scalability. Big data can be defined by the famous 5Vs (Volume, Velocity, Veracity, Variety, and Value) with extensive data sets captured from multiple channels (NIST). Big data applications with IoT, Fog, and Cloud Computing grow exponentially: businesses as well as the speed of the data and its storage. Therefore, it is the correct time to consider the systematic engineering approach to developing and deploying big data services. For example, based on 2017 data, google search handles 3.5 million searches per minute and Facebook handles 1 billion active users and stores more than 300 petabyte per minute.

In this changing era of development, services are to be Robust, Agile, Accessible and Available to its clients. For secured and guaranteed delivery of services, every big organization is shifting their service delivery model to Enterprise Service Bus (ESB) which is the key design paradigm of Service-Oriented Architecture (SOA) and guarantees Reuse, Reliability, Resiliency (3Rs), as well as Availability. In this context, the following research questions are posed:

- What are the design principles for an SOA driven reference architecture?
- What are services comprise reference architecture for big data systems?
- How to classify technologies and products/ servi-

Ramachandran, M.
SOSE4BD: Service-Oriented Software Engineering Framework for Big Data Applications.
DOI: 10.5220/0007708702480254
In *Proceedings of the 4th International Conference on Internet of Things, Big Data and Security (IoTBDS 2019)*, pages 248-254
ISBN: 978-989-758-369-8

ces of big data systems?

This paper presents a systematic software engineering approach to developing big data services and analytics services and applications. This paper also presents a service-oriented software engineering framework for big data (SOSE4BD). Section 1 discusses an introduction and sets research agenda in the area of software engineering in the era of big data, IoT, and cloud computing technologies and software as a service paradigm. Section 2 presents background studies. Section 3 presents SOSE framework.

2 BACKGROUND: SOFTWARE ENGINEERING FOR BIG DATA APPLICATIONS

One of the main characteristics of big data systems is commonly known as 3Vs, 5Vs, and 7Vs and are discussed as velocity, volume, variety, veracity, and value of incoming data in real-time or a captured data over several time periods which is shown in Figure 1.

- Velocity: BD requires real-time processing at varying intervals and may include stream as well as batch processing
- Volume: BD provides a massive historical data over several time periods (years, months, weeks, days, etc.)
- Variety: The BD captured may be in a variety of formats (multiple files and multi-modal data) and may be structured and unstructured.
- Veracity: The BD captured may contain unwanted data which require extraction, transformation, and cleaning
- Value: BD may contain very highly valuable as well as not so useful data and it requires a skilled data scientist to identify what to consider for analytical processing and what to discard.

Figure 1: 5Vs of Big Data.

According to Internet Minute (2018), it captures, 973K logins in 60 seconds globally, 4.3 million videos watched in 60 seconds, etc. This demonstrates the increasing volume and velocity of data being used and generated. In addition, the number of devices used to generate these data are rapidly increasing and the fusion of devices, applications and composition of new applications and analytics is also on a fast pace. Therefore, it is important to adopt a systematic approach to developing, capturing, analysing, measuring, and using big data.

Most of the big data projects fail due to lack of findings on the ways to capture, systematically manage, interpret, and to predict business directions out of big data investments. Gorton (2014) says that lack of knowledge in technologies, systematic approaches, and discipline around big data are new and therefore it is difficult for people to make business judgement based on data visualisation alone. Therefore, this paper emphasises on software engineering approach to big data and its applications. Gorton (2014) also states that big data is a complex software engineering problem than a data science problem. It has been proposed a lightweight evaluation and architecture prototyping for big data (LEAP4BD) which is based on creating a knowledge base to derive quality requirements, evaluation criteria, candidate selection and prototyping. Most of the problems that have been identified are the size of data, speed of data, horizontal and vertical scaling of distribution, different political sources of data, consistency of data, scalability of data, performance of data and availability of data. We all know that over fifty years of software engineering practices revealed that scalable architecture, technologies, processes and platforms have been successful in delivering cost-effective solutions. In this context, SEI has developed a knowledge base for big data architecture and technologies known as QuABaseBD (2018).

Gorton et al., (2016) discuss what is known as Eric Brewer's CAP theorem which means a system must be able to support Consistency, Availability, and Partition (support for message between nodes in the cluster). They also state that this theorem forces to develop scalable architecture. However, the above approach has limitation in providing software engineering approach to big data problem. Therefore, in this paper, we believe, a reference architecture is the best solution to tackle large scale applications of big data with a systematic software engineering approach.

Therefore, in this paper, we propose a software engineering framework for big data (SEF4BD) which provides a systematic process for big data projects

and a reference architecture for big data (REF4BD), providing strict architectural structuring based on reference architecture model.

Karakaya (2017) discusses big data frameworks such as Hadoop, Spark, Storm and Flink which are specifically developed to solve big data applications by providing facilities to collect, process, manage, monitor and to analyse big data. However, this paper also discusses the big data applications and their limitations without software engineering approach.

Madhavji et al., (2015) present a contextual model of big data software engineering which includes scenarios of data capturing, storing and visualising support. Arruda and Madhavji, (2017) and Xu et al., (2018) have proposed requirements engineering artefact model for big data systems in which they classified requirements engineering activities into four categories such as data consumer requirements, data transformation requirements, data source requirements, and data capability requirements. All are part of big data requirements which include traditional requirements engineering aspects such as functional and non-functional. Non-Functional requirements should include key quality attributes for big data systems such as performance, reliability, privacy, and security. However, it is ongoing research and details of RE for BD remains unclear.

Arndt (2018) discusses the importance of the interplay between software engineering and big data and has discussed two distinct areas for further exploration:

1. **Software Engineering for Big Data** which can provide a systematic process for improving the development of big data systems. The process includes requirements gathering for BD, software architecture for BD, testing and debugging BD systems (performance, reliability, and security) where the logs of analysing 5V characteristics should be included, SE process for BD which could include CMMI, and finally Managing BD projects.
2. **Big Data Software Engineering** is an area of research which should focus on utilising BD for the benefit of improving SE practices and to improve software production. The typical activities should include analytics for software engineering, data mining software repositories, visual analytics for software engineering, and self-adaptive systems which utilises data generated and self-learn.

Similarly, Bagriyanik and Karahoca (2016) has discussed extensive systematic literature survey on big data in software engineering and have concluded that there is a need for a holistic approach to developing a big data system. Kacha and Zitouni (2018) presented a data security model based on cloud characteristics and security attributes (confidentiality, integrity, and availability) to be built into the data lifecycle (stored, used, and transitioned data).

The existing studies have started to identify the importance of the 50 years of software engineering practices and to benefit from the emerging big data approaches and technologies to improve businesses. However, the field is at an early stage, and therefore there is a lack of a clear picture of software engineering role.

Our earlier work on a software engineering framework for service and cloud computing (Ramachandran, 2018) and business intelligence architecture for big data systems (Ramachandran, 2017) have established a standardised method and process in the cloud-based services. Hence, this paper provides a framework for big data software engineering which is a service based (SOA), data service component model, SOSE Development lifecycle for BD, and a reference architecture for BD.

3 SOSE4BD: SERVICE-ORIENTED SOFTWARE ENGINEERING FRAMEWORK FOR BIG DATA APPLICATIONS

SOA has emerged supporting business integration by providing service components, architectural framework with unique and unified enterprise service bus, service orchestration, and service composition. Therefore, it is beneficial to design and construct big data applications using well established over 50 years of software engineering best practices. Big data have emerged to improve business best practices by utilizing various data that have been generated in the past as well as at present in a various formats and from a variety of sources (multi-channels). Therefore, it is essential to merge the two disciplines of big data and service-oriented software engineering, Service-Oriented Software Engineering for Big Data (SOSE4BD). The basic principles of this new discipline is shown in Figure 2, the Four Pillars of SOSE4BD Principles.

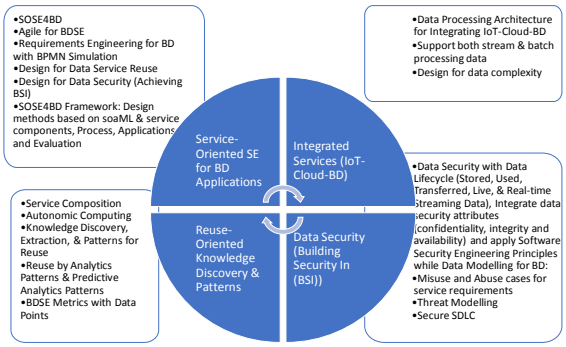

Figure 2: Four Pillars of SOSE4BD Principles.

It consists of four quadrants that provide an integrated approach to BD, SOA, and SE as follows:

1. **Service-Oriented SE for BD Applications.** This is essentially the integration of best practices of three disciplines viz: BD, SE, and SOA, known as SOSE4BD. It mainly consists of a software engineering framework for big data, software engineering framework for service computing and cloud computing (SEF-SCC, Ramachandran, 2018), Agile practices for service computing, requirements engineering for service computing with BPMN modelling and simulation to verify service process and its efficiency, design for service reuse (building-in reusability), design for security (build security in (BSI)).
2. **Integrated Services (IoT-Cloud-BD).** It promotes integrating Data Processing Architecture with cloud and IoT based data streaming services.
3. **Data Security** (Build Security In (BSI)). We believe in the principle of developing a secured system throughout the lifecycle. (Ramachandran, 2012). It also provides Data Security with Data Lifecycle (Stored, Used, Transferred, Live, & Real-time Streaming Data), Integrating data security attributes (confidentiality, integrity and availability) and apply Software Security Engineering Principles while
4. Data Modelling for BD such as misuse and abuse cases for service requirements, threat modelling and secure SDLC
5. **Reuse-Oriented Knowledge Discovery & Patterns.** In this principle, the main aim is to cultivate service level reuse through service composition, autonomic computing, knowledge discovery, extraction, patterns for reuse, reuse by analytics patterns & predictive analytics patterns, and BDSE metrics with data points.

Our approach to big data software engineering which integrates software engineering, best practices with the use of repositories for software engineering data (Menzies and Zimmermann, 2018; Yang et al., 2018), SOA, Service Computing, and Cloud Computing. Therefore, Figure 3 presents a framework known as SOSE4BD which consists of requirements engineering for modelling and simulating service requirements with BPMN as shown in SOSE lifecycle (Figure 6), well proven software design using SoaML and architecture principles, a reference architecture for big data (REF4BD) which is a service-centric based on SOA.

REF4BD is based on well proven design concepts and principles as shown in Figure 4. SOSE4BD also recommends tools for big data software engineering analytics and predictive modelling with SAS, Visual Paradigm, and Azure/ML. SOSE4BD also supports a

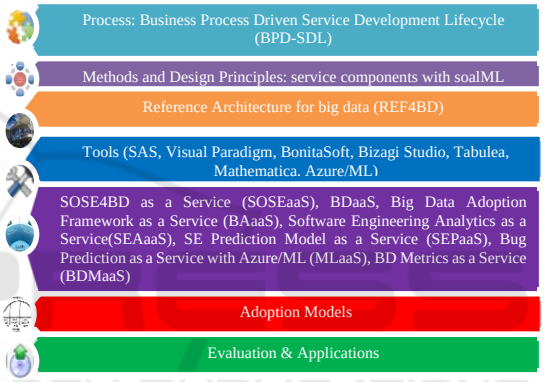

Figure 3: SOSE4BD Framework.

A number of data security-centric services are part of ongoing research such as SOSE4BD as a service, Bug Prediction as a Service with MLaaS (Azure Machine Learning as a service), etc. SOSE4BD also supports an adoption model for employing big data in an organisation, and an evaluation of the framework through simulation and a number of applications such as British gas energy efficiency using our approach (Ramachandran, 2017).

A reference architecture is the key to achieving standard practice of developing software product lines and services based on common architectural style across the product family and family of software services. Hence, SOSE4BD framework has developed a reference architecture for big data as shown in Figure 4 and SOSE4BD framework has also developed a set of service component models re-enforcing to map services into REF4BD. Oracle (2013) also recommends a reference architecture for big data whereas REF4BD's reference architecture is a service based (based on the principles of Service-Oriented Architecture). It consists of three sets of

layers namely BD Source & Storage Layer which focusses services on data stream and data storage, followed by an big data enterprise service bus which integrates multi-channel data sources (mobile, IoT, sensors, actuators, location-based services, etc), followed by Big Data Processing Layer which mainly focus on data processing, data transformation, data visualization, data analytics, and knowledge discovery of identifying data patterns and behaviors' for knowledge extraction, and the top layer known as Big Data Application and Prediction Services which focus on providing other business and improvements monitoring services through service orchestration, prediction modelling based on machine learning as a service such as Microsoft Azure/ML, and provides data security.

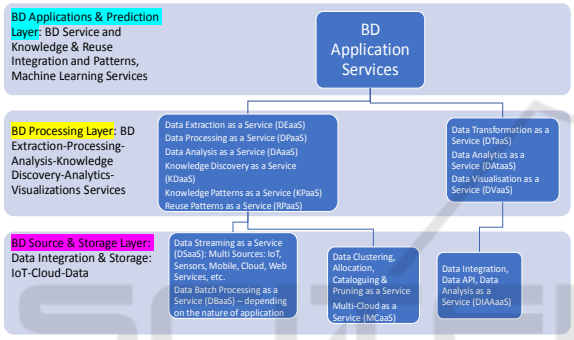

Figure 4: Reference architecture for big data (REF4BD).

Figure 5 shows a SOSE4BD lifecycle which consists of starting with BD requirements stage of identifying data source such as software repositories for big data software engineering projects and other data sources as shown by Menzies and Zimmermann (2013), identifying goal for improving software process, methods, project efficiencies from software project managers, users, and developers, and to identify requirements for analytics and predictive analytics. In addition, we need identify data requirements such as data source, data transformation, data streaming, data storage, data capability, and business intelligence and business continuity. Secondly, the BD design stage should start soon after new data and software practices and process improvement services are validated with a BPMN process modelling and simulation for efficiency and resource constraints. During, the BD design stage, used soaML for service components and REF4DB for mapping service components into REF4DB architectural layers. Thirdly, SOSE4BD lifecycle recommends container based technology for big data service implementation which could include capturing project artefacts autonomically by deploying BD SE services as part of the IDE (Integrated software development environment) or into to a cloud driven software development services.

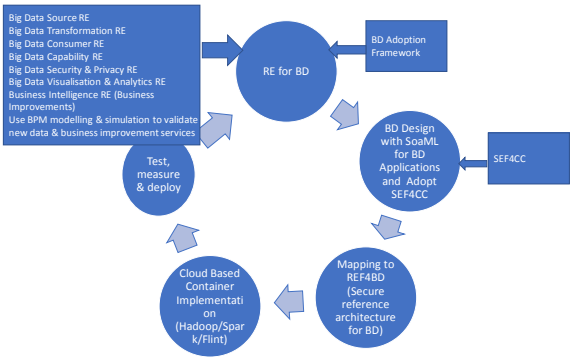

Figure 5: SOSEBD Lifecycle.

SOSE4BD framework recommend a number of BD SE services such as handling real-time data with multiple-channels and cloud service providers such as Microsoft, IBM, Google, Opensource, etc. The soaML SOA design for SOSE4BD based on REF4BD architecture as shown in Figure 6.

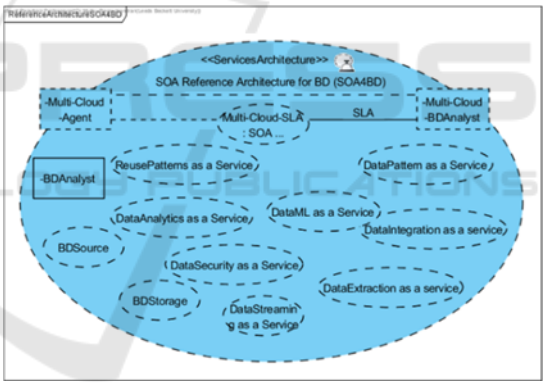

Figure 6: SOA driven SOSE4BD data services.

The services include BD Analyst, Reuse of Data Patterns as a Service, Bug Prediction as a Service with Machine Learning (Subbiagh et al., 2018), Data extraction as a Service, Data Streaming as a Service, Data Modelling as a Service, Data ETL (Data Extraction, Data Transformation, and Data Load as a Service), and Visual Analytics as a service, and finally Predictive modelling and Continuous Improvement as a Service. The next section provides an evaluation with a Facebook real-time data analytics case study.

4 SOSE4BD BIG DATA REQUIREMENTS ENGINEERING EVALUATION WITH BPMN MODELLING AND SIMULATION: BIG DATA FACEBOOK CASE STUDY

Facebook handles trillions of multi-channel data in real-time, batch processing, real-time analytics, and response within seconds. Therefore, it needs a high-performance computing architecture to handle big data processing (Chen et al., 2016). Facebook is mainly concerned in measuring performance, fault-tolerance, correctness, and scalability. Chen et al. (2016) have reported that Facebook uses their own big data processing tools such as Puma, Swift, and Stylus stream processing systems. In this case study, we have used real-time processing business processes mapped onto our SOA Based reference architecture (REF4BD) as shown in Figure 7.

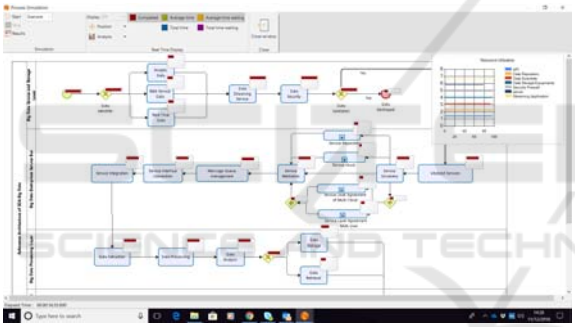

Figure 7: Facebook Big Data Processing with REF4BD.

As shown in the Figure, a snapshot of using Bizagi Studio for BPMN 2.0 modelling and simulation, consists of a number of business processes such as start with 100s of real-time data split by a data identifier (gateway notation in BPMN) into analytics data or web service data or real-time streaming data. This is then processed in our REF4BD data source layer, and then passed onto other layers in the reference architecture. The results show a number of times a particular business service has been accessed and executed to process that data, and time taken. In addition, Bizagi BPMN also shows a number of times each resource has been used such as an API, Data Scientist (Human Tasks in BPMN), Data Repository, Servers, Firewall, Data Storage, etc. The results show by implementing Facebook types of big data processing into REF4BD is more secure and uses resources efficiently than suing non-standard architectures. The efficiency result shows about 95% use of automated processing by API and Data Application (Service Components) services.

In conclusions, compared to Chen et al. (2016) Facebook uses more filters to do real-time streaming events. We argue that the filters can cause extra-overheads and resources required whereas REF4BD is more predictable, and can achieve correctness, fault-tolerance, and scalability since it is standardised across all data process applications and services.

5 CONCLUSIONS

SOA has emerged based on established software design principles of find-request-service paradigm suitable for service-oriented applications such as big data processing and analytics. Therefore, it is time to consider systematic and engineering approach to developing and deploying big data services as the data-driven applications and devices increasing rapidly. In this context, this paper proposed a software engineering framework and a reference architecture which is SOA based for big data applications' development. This paper also concluded with a simulation of a complex big data Facebook application with real-time streaming using BPMN simulation to study the characteristics before big data service design, development, and deployment. The simulation results demonstrated the efficiency and effectiveness of developing big data applications using the reference architecture framework for big data.

REFERENCES

Al-Jaroodi, J., Hollein, B., and Mohamed, N (2017) Applying software engineering processes for big data analytics applications development, 2017 *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, USA.

Arruda D., and Madhavji, N.H. (2017) Towards a Big Data Requirements Engineering Artefact Model in the Context of Big Data Software Development Projects, *2017 IEEE International Conference on Big Data (BIGDATA)*.

Arndt, T (2018) Big Data and software engineering: prospects for mutual enrichment, Iran Journal of Computer Science, 1:3–10, https://doi.org/10.1007/s42044-017-0003-0

BCS (2004) The Challenges of Complex IT Projects, The report of a working group from *The Royal Academy of Engineering and The British Computer Society*.

Bagriyanik, S. & Karahoca, A. (2016). Big data in software engineering: A systematic literature review. *Global Journal of Information Technology*, 6(1), 107-116.

Caldarelli, G and Vespignani, A (eds) (2007) Large Scale Structure and Dynamics of Complex Networks from information technology to finance and natural science, *World Scientific Publishing Co.* Pte. Ltd.

Cao, L.B (2015). Metasynthetic Computing and Engineering of Complex Systems. *Springer-Verlag*, London, U.K.

Cao, L.B (2017) Data Science: Challenges and Directions, *Communications of the ACM*, 60 (8), August

Chen, G. J et al. (2016) Real-time Data Processing at Facebook, *ACM SIGMOD 2016 San Francisco*, CA USA.

Dehmer, M et al. (2016) Big data of complex networks, *Chapman and Hall/CRC.*

Fontana, A., and Wrobel, B (2013) Evolution and development of complex computational systems using the paradigm of metabolic computing in Epigenetic Tracking, *Wivace 2013 - Italian Workshop on Artificial Life and Evolutionary Computation.*

Gorton, I. 2004, Software Architecture for Big Data Systems, Software Architecture: Trends and New Directions SEI/CMU, Technical Presentation, https://resources.sei.cmu.edu/asset_files/Webinar/2014_018_101_298351.pdf.

Gorton, I., Bener, A., and Mockus, A (2016) Software Engineering for Big Data Systems, Special Issue, *IEEE Software*, March/April 2016.

Internet Minute (2018), http://www.visualcapitalist.com/internet-minute-2018/

Jin, X., et al (2015) Significance and Challenges of Big Data Research, *Big Data Research* (2015) 59–64 http://dx.doi.org/10.1016/j.bdr.2015.01.006

Karakaya, Z (2017) Software Engineering Issues in Big Data Application Development, *2nd Int. Conference on Computer Science and Engineering (UBMK'17), IEEE Press.*

Kacha L., Zitouni A. (2018) An Overview on Data Security in Cloud Computing. In: Silhavy R., Silhavy P., Prokopova Z. (eds) Cybernetics Approaches in Intelligent Systems. CoMeSySo 2017. *Advances in Intelligent Systems and Computing*, vol 661. Springer, Cham.

Laigner, N. R et al. (2018) A Systematic Mapping of Software Engineering Approaches to Develop Big Data Systems, 2018. *44th Euromicro Conference on Software Engineering and Advanced Applications.*

Madhavji, N., H., Miranskyy, A., and Kontogiannis, K. (2015) Big Picture of Big Data Software Engineering, *2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering*.

Menzies, T and Zimmermann, T (2013) Software Analytics: So What?, *IEEE Software*, vol. 30, no. 4, 2013.

Navlakha, S and Bar-Joseph, Z (2015) Distributed Information Processing in Biological and Computational Systems, Communications of the ACM | January 2015 | VOL. 58 | NO. 1.

NIST, NIST Big Data Interoperability Framework: Volume 1, Definitions, https://www.nist.gov/publications/nist-big-data-interoperability-framework-volume-1-definitions.

Ng, I et al (Eds) (2011) Complex Engineering Service Systems: Concepts and Research, Springer, London.

Oracle (2013) Big Data & Analytics Reference Architecture, White Paper, Oracle.

QuABaseBD (2018) https://quabase.sei.cmu.edu/mediawiki/index.php/Main_Page.

Ramachandran, M (2008) Software Components: Guidelines and Applications, *Nova Science Publications*.

Ramachandran, M (2012) Software Security Engineering, *Nova Science Publications.*

Ramachandran, M (2017) Service-Oriented Architecture for Big Data and Business Intelligence Analytics in the Cloud, Paper 9, Computational Intelligence Applications in Business Intelligence and Big Data Analytics" Sugumaran, V. Sangagaiah, A and Thangavelu, A (eds), CRC Press, (*Taylor & Francis Group*).

Ramachandran, M (2018) SEF-SCC: Software Engineering Framework for Service and Cloud Computing, Fog Computing: Concepts, Frameworks and Technologies Edited by Z. Mahmood (ed), Springer.

Sommerville, I (2016) Software Engineering, *10th edition, Pearson.*

Subbiah, U and Ramachandran, M., and Mahmood, Z (2018) Software Engineering Approach to Bug Prediction Models Using Machine Learning as a Service (MLaaS), Porto, Portugal, 26-28th July.

Xu, X., et al. (2018) A New Paradigm of Software Service Engineering in the Era of Big Data and Big Service, *Computing, Springer*, April 2018, Volume 100, Issue 4, pp 353–368.

Yang, Y. et al (2018) Actionable Analytics for Software Engineering, Actionable Analytics, Guest editors Introduction to Special Issue on Actionable Analytics for SE, IEEE Software, Jan/Feb 2018.

Zanetti, S.M (2013) A Complex Systems Approach to Software Engineering, *DSc Thesis*, Eth Zurich.