

New Trends in Service-Oriented Systems Engineering  
-- A Personal Reflection on the IEEE SOSE 2017 Conference

Hong Zhu  
Oxford Brookes University, Oxford, UK  
Email: [hzhu@brookes.ac.uk](mailto:hzhu@brookes.ac.uk)

## 1. INTRODUCTION

Twelve years ago, when service-oriented computing was the buzz word in the computing community of the academic and the industry as well, the vision on services oriented systems engineering (SOSE) starts with the recognition of the fundamental differences in service-oriented software and traditional software (say those written in object oriented programming languages), and thus the need of a new engineering discipline to develop, operate, maintain and evolve service-oriented software systems that suits the new features. Over these years, service-oriented computing advanced continuously and rapidly. In this process, various branches of service-oriented computing formed, which include web services, semantic web services, social networking media, cloud computing, mobile cloud, big data, to the current hot topics of Internet of Things, data science and deep learning, and fog/edge computing, BlockChains, robotics, etc. A great number of computer application systems have been developed based on the principle of service-orientation and delivers services to billions of people around the world 24 hours a day and 7 days a week. It has penetrated to all aspects of our lives and human society. No doubt, that is a great achievement.

It is observed that, for all these hot topics of research (mostly from the academics), the focuses have been on the enabling techniques (in contrast to engineering methodologies). Researcher on service-oriented systems engineering (SOSE) distinguish themselves from those on service-oriented computing by focusing on more generally applicable engineering principles, methods and theories to build ever larger, even more complex, more reliable, more trustable, more efficient, more flexible and more usable service-oriented systems. We observed that in the recent years the IT industry has made great advances in service-oriented systems engineering, which is in a completely different direction that the majority of researchers from the academics have pursued and suggested. The following storming developments have revolutionized the IT industry significantly and profoundly. These developments point out the directions that research in the academics should pursue. Here, we briefly summarize these new trends.

## 2. MicroServices

The first trend is the wide adoption of the so-called microservices software architecture in cloud computing.

Microservices is a software architectural style in which a software system consists of a large number of services of micro scale. Each component of a system in the microservices architecture provides a single function and runs on its own process. They can be deployed to computers of a cluster. They communicate with each other through a lightweight communication mechanism. Moreover, new copies of a micro-service can be created and idle existing copies can be destroyed dynamically at runtime to achieve elastic scalability, load balance and performance optimization. A faulty micro-service can be dynamically replaced by a new one to achieve high reliability through fault tolerance. Practitioners in the development of applications in

microservices architecture also claimed a wide range of benefits of this architectural style, which include the support of team collaboration, continuous testing, integration and delivery (thus evolutionary development), etc.

### 3. DevOps

The second noticeable event folding right now is the movement of the so-called DevOps, which is a new word created by the concatenation of two words of development and operations.

DevOps is a software methodology that includes the whole life cycle of service-oriented systems. In particular, it emphasizes on the integration of software development and operation processes and human resources. Given the ever increasing complexity of deployment, operation, monitoring and fault diagnosing large scale service systems, the philosophical principles of DevOps are, firstly, the process of deployment and operation must be seamlessly integrated with the development process. Secondly, development and operation activities can be more effective and efficient if shifted to the left, i.e. done as earlier as possible. Consequently, the team of developers and the team of deployment and operation teams must be integrated by removing the communication boundary between them. The products of development can then be moved down the DevOps pipeline and shipped from development platform, to a testing platform, to a stage platform and eventually to the operation platform, where each platform is typically a cluster of computers and may contain multiple different operating systems. A great number of so-called DevOps tools have been developed to monitoring the execution of software on a cluster of computers, to analysis the log files for diagnosing the systems behavior, to register and access services deployed to a cluster, to configure and deploy software to a cluster. DevOps inherits many principles of agile methodology, but geared to develop large-scale complicated service oriented software running on large-scale clusters. As a methodology, it has been adopted by many major players in the cloud computing service providers and applied together with microservices architecture and container technology.

### 4. Container Technology

The third storm unfolding right now in the IT industry is the so-called container technology, in particular, the Docker.

Container technology is a lightweight virtualization technology that enables sharing file systems, memory space, processors, communication, etc. It is realized by wrapping a piece of runnable software code together all the resources that are needed to run the code on a given platform into a package (which is called a container): the code, runtime environment, system tools, system libraries, etc. A container can be quickly distributed to a virtual or real machine in a cluster of computers and be started quickly. Multiple containers can be hosted on a machine (virtual or real) on top of a single operating system sharing the same operating system kernel, but separated by the containers. Docker is an open source project that develops the container technology. It is the de facto industry standard for packaging program code into containers and running containers on top of operating systems. It is widely adopted by cloud service providing industry, which includes all main players in the sector of the cloud industry. For example, Amazon's Container Service provides a highly scalable and fast container management service. It enables to run, stop and manage Docker containers on a cluster of Amazon EC2 instances.

Oracle's Solaris Zones provides containers for Oracle Solaris 11 OS. A number of container orchestration engines, such as Google's Kubernetes, have been developing and some already in practical uses. They automate the deployment of applications by distributing a number of containers to a cluster of machines in pre-scripted configuration. Container technology provides strong support to the microservices architecture by enabling services to be deployed and started quickly and flexibly.

## 5. New Paradigm of Programming Languages

Finally, dozens of software languages have emerged aiming at developing web-based and cloud computing software. In addition to new programming languages, such as Go, Scala and CAOPLE (which is my own attempt of developing a programming language for service oriented systems), just to mention a few, other categories of such software languages include script languages, such as Node.js (JavaScript), service formal specification languages, such as Sofia (again it is my attempt), markup and data representation languages, query languages, web application languages, and finally service description languages. New languages are still emerging every year.

Among many efforts of research and development in search of new software languages that are suitable for the development of cloud service applications, those languages that explore distributed and parallel programming models worth mention. It is a subject that a ground-breaking progress is in preparation.

## 6. XbS

An observation that I have made at the 11<sup>th</sup> IEEE SOSE conference held from 7<sup>th</sup> to 10<sup>th</sup> April 2017 is that service-oriented systems engineering approaches are being applied to a number of domains, including Internet of Things, Smart City, BlockChains, Big Data, etc. What is new is that, in addition to X-as-a-Service (XaaS), it seems that every thing is implemented by service orientation. It is just like microservices, in which the services delivered to the users are also implemented by services. The advantages of service-oriented systems engineering approach to the development such applications are widely recognized by researchers world-widely regardless of the application domain. The trend is clear: we will see more and more of IoT implemented by Services (IoTbS), Smart City implemented by Services (SCbS), BlockChain implemented by Services (BCbS), etc. Thus, I would like to propose a term called "XbS" for X implemented by Service Orientation, where X can be anything.

## 7. SUMMARY

Based on the observations above, we can conclude that the research on service-oriented systems engineering is coming to a new era, an era of XbS, i.e. everything is going to be implemented by applying service-oriented systems engineering approach.

The industry has again led the way forward. In particular, microservices addresses the architectural design problem by proposing a new architectural style. DevOps addresses the development methodology problems by proposing a new set of philosophical principles, a new service lifecycle and/or process model, and a set of supporting tools. Container technologies address the problems in the deployment

and distribution of microservices into complex environment by proposing a more flexible and efficient runtime environment and tools. Researchers on software languages are still searching for new languages for developing service-oriented systems on various stages of the life cycle. These are indeed the heart of the engineering problems of service-oriented systems.

(Note: For references to this article, please cite the following.

“Hong Zhu, New Trends in Service-Oriented Systems Engineering, April 11<sup>th</sup>, 2017.  
Available Online at URL:  
<http://cms.brookes.ac.uk/staff/HongZhu/Publications/NewTrendsInSOSE.pdf>.”)