Agile Architecture for Digital Enterprises

Marina Anshina [0000-0003-1590-0792]

FOSTAS Foundation, Moscow, Russia
manshina@fostas.ru

Abstract. The need for a systematic architectural approach for a digital enterprise is very important. On the other hand rigid centralized architectures for a digital enterprise is inefficient. There are several stages of development of architectural styles from description and analysis to forecasting and designing the future. At present time the most popular models of Enterprise architecture related to service approach. In different sources various models of service architecture are detailed, including Service Oriented Architecture (SOA), agreement-driven service architecture (ADSA), and microservices. Possibly it's related to the values of role-based architectural models, such as the Web services model and the cloud computing reference architecture model. Several architecture models related only to Application later of Enterprise Architecture. But there is relationship between the organization structure and application architectural models. And so the application architecture models quickly propagate to the Enterprise Architecture. The structure of SLA ADSA, which is based on the propagation application service models such as Web services and microservice architecture to the Enterprise architecture, is suitable for building reference architectures of new technologies. We could predict growing popularity of ADSA and appearance of new Architecture frameworks and standards for practically using and creating of Enterprise common information environment.

Keywords: Service architecture model, digital enterprise, Agile Architecture, microservices.

1 Introduction

Digital transformation (further referred as DT), which forms digital enterprises, is understood in different ways. Many people do not see the difference between DT and automation. But the concept DT consists of two words, so the noun "transformation" can not be discounted. If we use this definition of a digital enterprise: "Significant changes in the business or activity of an organization associated with the transition to new business models using digital technologies", then the creation of a digital enterprise implies a revolutionary change in its architecture. And like any revolution it can result in significant shocks up to the closure of the enterprise.

However history has shown that there are also "bloodless" revolutions. Therefore, with the right approach, it is quite possible, following the path of DT, to lead the enterprise to digital prosperity. To do this, first of all, it is necessary to understand that in such a complex system as an enterprise, there are no small details and that an un-

Proceedings of the of the XXIII International Conference "Enterprise Engineering and Knowledge Management" (EEKM 2020), Moscow, Russia, December 8-9, 2020.

justified underestimation of one of the elements or a separate connection can lead to problems. Therefore, a systematic approach takes on such significance.
Previously, the author described reference architectural models of new technologies [1], the use of the ADSA model in the digital economy [2], and the use of the ADSA model for reference models of various technologies [3]. This article describes the development of architecture in the direction of Agile and the role of ADSA in this.

2 Role of the system approach

A systematic approach not only in the field of enterprise automation, but also in the field of management and implementation of any activity, both within the organization and with customers and partners, is becoming increasingly important. This is primarily due to the complexity of the system that is a modern enterprise.
In addition, it is important to note that the boundaries of a modern enterprise are blurred. More and more functions are being outsourced or transferred to the cloud, more and more functions integrate with both suppliers and partners on the one hand, and with customers and consumers on the other, more and more digital elements are being included in the Internet of things and Industry 4.0. Therefore, the enterprise should now be considered not just as an expanded enterprise, but as an ultra-expanded enterprise, which adds complexity to any tasks related to it, both in the field of project and process activities, and increases the importance of a systematic approach.
A systematic approach becomes absolutely necessary if an enterprise has started or is going to follow the path of digitalization, which implies global changes in the structure of its activities and maximum flexibility in response to changes in the external and internal environment. It is impossible to do this without a clear understanding of what the enterprise is.
The system approach is inextricably linked to the concept of Enterprise Architecture, which is perhaps the only known discipline that allows you to look at the enterprise from different points of view and describe its various aspects. The Enterprise architecture (further referred to as EA) includes its external relations related to the concept of an Expanded Enterprise, but does not lose a single holistic view of the Enterprise. Exactly a holistic view allows you to identify and take into account the synergy formed when combining all the elements of the enterprise.
Enterprise architecture as a field of knowledge has been actively developing for more than 40 years and is supported by a large number of standards, best practices, and frameworks [4]. Perhaps the last decade has seen a certain loss of interest in this discipline, due to the fact that enterprises using the architectural approach have not received the benefits that it seemed to promise. Despite the fact that the costs of projects related to architectural modeling are usually very significant, sometimes it is difficult to calculate the return on investment in such projects and, most often, almost impossible. In addition, many EA standards, frameworks, and methodologies have proven to be very difficult to apply in practice. And applied incorrectly EA models brought organizations only unnecessary costs and frustration.

One of the main challenges facing a digital enterprise is to ensure a balance between the flexibility needed to respond to changes occurring inside and outside of it in a timely manner, and the stability that ensures its efficient, safe and reliable operation. Centralized architectural models are an archaic, rigid and highly risky structure in modern conditions, which hinders the development of enterprises and cannot be applied in conditions that require rapid response to a changing operating environment. The situation with the pandemic is still particularly clear. Centralized architectural models do not meet the requirements for flexibility and scalability, because when you need to make changes to elements or their relationships and/or when the number of interactions increases significantly, the load on the Central node increases quickly and to ensure performance, you have to radically change the architecture and rebuild many elements and their relationships. Methods of scaling software systems of the past due to the growth of the power of the technological architectural layer, in particular, due to the increase in server resources, do not solve the problem, since the requirements for the speed of changes and scaling limits have increased significantly. The high risks of such centralized systems are due to dependence on the center: if it fails, the entire system fails.

Only loosely coupled architectures can provide the necessary level of flexibility, reliability, and security. For such architectures, the stability and reliability of the system with great flexibility of components is provided by standardized links between them.

The architectural approach, like all technologies, develops along the Gartner curve [5]. And Gartner notes that the EA curve, after about a decade of decline, is already reaching a plateau characterized by mature technologies and effective, competent application. Gartner, as well as many other experts, notes the trend in the field of EA, associated with the transition from global, more theoretical models, such as the Zachman and Spivak models to simpler and more accessible notations and frameworks such as Archimate [6].

Table 1 shows the classification of EA theories.

Table 1. Categorization of digital transformation theories (based on [5])

Type	Main characteristic
I. Analysis	Focus on analysis and description of the company. It usually doesn't include cause-and-effect relationships. Explore phenomena / events / objects. (Performance Reference Model)
II. Explanation	Answers the questions what, how, why, when and where. The main goal is to explain and understand the company's activities: it does not include planning and forecasting issues. The theory can't be tested. (The Zachman Model [9])
III. Prediction	Answers questions about what is and what will be. Provides forecasts and is verifiable based on their performance. Usually does not include verification cause-and-effect relationships. (TOGAF)
IV. Explanation and prediction	Answers the questions what, why, why, when, where is and what will be. Provides predictions, can there-

Type	Main characteristic
	fore be verified, and includes cause-and-effect relationships.
V. Designing the future	Answers the question of how to achieve something. Provides transparent forecasts (i.e. methods, techniques, principles, and / or functions) for constructing artifacts or complex objects.

In addition, based on the analysis of AP theories, it is possible to identify their General characteristics, shown in table 2.

Table 2. Structure of AP theories (based on [5])

Component	Scope
I. Description tools	Text description, tables, diagrams, notations, and languages.
II. Main objects of the description	The main elements of the enterprise, such as strategy, technology, processes, organizational structure, roles, assets, including information, etc. These objects can be described at different levels of abstraction.
III. Interactions, including cause-and-effect relationships	Interactions between objects, including interactions at different levels of abstraction and transformation of objects over time. Interactions can be of various types: dependencies, associations, usage, assignment, initialization, and so on. They can be either bidirectional or unidirectional, and may or may not support branching.
IV. Content	Defines the degree of detail of the description of the enterprise, assumptions, and assumptions.
V. Hypotheses (for EA models types III, IV and V from table 1)	Assumptions about changes to elements and relationships.
VI. Methods for implementing changes (for EA models type V from table 1)	Defines how the changes described in the hypotheses can be achieved.

Table 2 for the Zachman Model is given in [5].
The architectural approach for a digital enterprise must make a qualitative leap and be enriched with a variety of styles, techniques and tools in order to maintain the flexibility, scalability and richness of the choice of options for forming and linking elements of a digital enterprise. Therefore, we can predict that type IV and V approaches will be actively developed.

3 Service architectural style

In particular, it is necessary to provide a more rigorous description of various architectural styles and expand their options. The variety of styles in enterprise architecture is close to the variety of styles in building architecture: from monolithic architecture to distributed architecture and from resource architecture to service architecture.

Obviously the most popular architectural style for a digital enterprise at present is Service Architecture. The service architecture allows you to achieve flexibility due to the variability of enterprise elements, placing strict requirements on their relationships, built on the basis of well-defined and controlled services. The service architecture is a logical extension of the open interface architecture that was laid down by OMG (Object Management Group) in CORBA (Common Object Request Broker Architecture).
As the service architecture evolves, it is also moving closer to practical application technologies. You can note the most popular service models such as:

- Service Oriented Architecture, in which you can distinguish such technologies as Enterprise Service Bus and Web services;
- Technologies based on SLA (Service Level Agreement), such as agreement-driven service architecture (ADSA);
- Microservices.

The service architecture style allows you to get loosely coupled architectural models that encapsulate the complexity of the architectural elements that make up the enterprise, just as object-oriented programming encapsulates the complexity of software system components. In the case of, for example, ADSA ([1], [2], [3]) interaction is carried out on the basis of well-formalized SLAs, which allows you to achieve high-quality architecture, in particular, its characteristics such as security, reliability, availability, performance and scalability.
Mostly architectural service frameworks are based on the role model. In particular, Figure 1 shows the basic architectural model of Web services based on the formalization of the interaction of three main roles: the service consumer, the service provider, and the intermediary (broker).

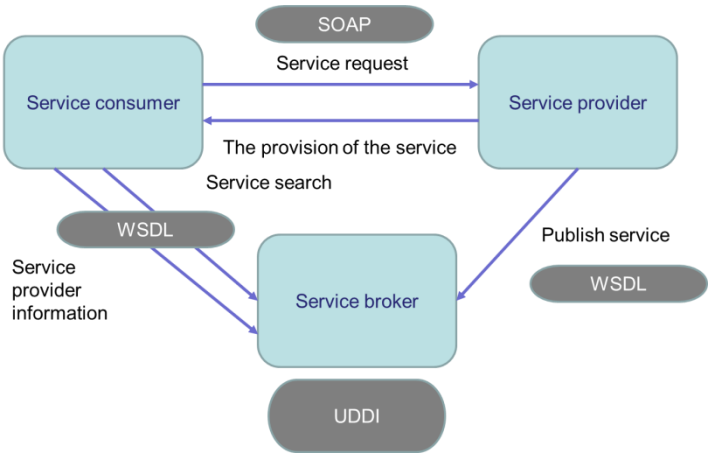

Fig. 1. SOA role model

The figure shows that the main Web standards, such as

- SOAP – Simple Object Access Protocol
- WSDL – Web Service Definition Language
- UDDI – Universal Description Discovery & Integration

relate to the formalization of interaction between the main roles.
Especially important to highlight the role of ADSA in cloud computing models. Having defined the reference architectural role model of cloud computing in the ISO/IEC 17788 [7] and 17789 [8] standards, the international expert community is developing the SLA system of standards (ISO/IEC 19086 [9]) in order to formalize and consolidate the interaction of the main roles of cloud computing, shown in figure 2.

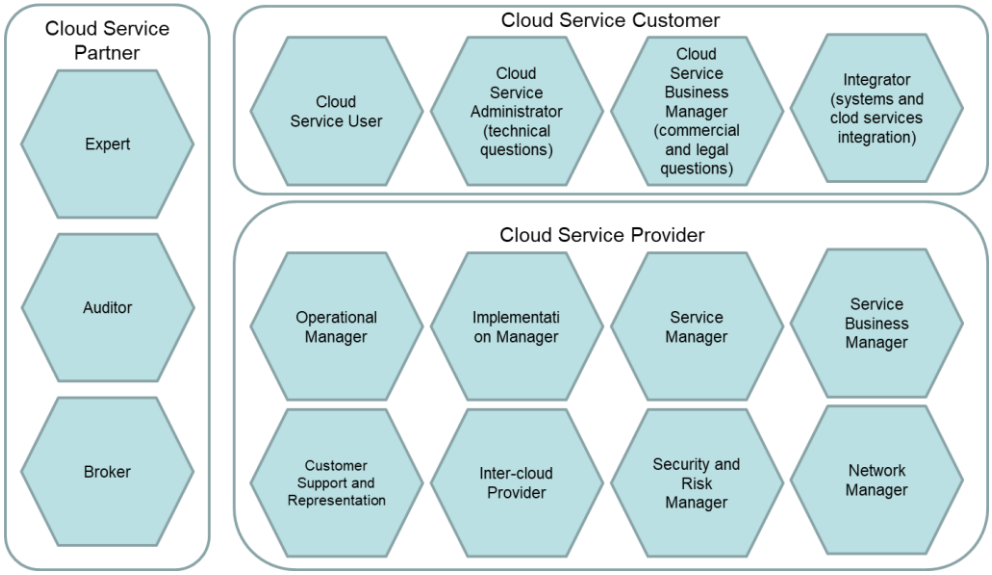

Fig. 2. Reference architecture of cloud computing

Accordingly, the SLA areas defined in ISO/IEC 19086 [9] and the corresponding SQUARE quality models [10] shown in figure 3 are developments of CORBA system services.

Fig. 3. SLA structure for the cloud computing role model

Further development of the service model is associated with the transfer of services to the practical field of design and development of software systems by developing the

microservice model, which has now become widely popular for automating various areas of activity. Microservice architecture is a variant of SOA aimed at interaction, as far as possible, of small, loosely connected and easily modified modules-microservices, which became widespread about 10 years ago. This architectural style is perfect for agile development practices and their development for the entire software lifecycle, i.e. for the processes of its operation is named DevOps.
As early as 1967, Melvin Conway observed that "organizations that develop systems ... create architectures that copy the interaction structures within these organizations" [11]. The digital enterprise must support a flexible, decentralized, democratic role model in order to provide the flexibility needed to respond to the changing environment. Thus, in accordance with Conway's law, the digital enterprise creates service, and at the software level, rather microservice architectures that have the flexibility that modern enterprises need. It should be noted that proper use of microservice architecture allows not to worsen, but even to improve the quality of the enterprise functioning.
The problem that the microservice architecture in its classical form covers only a layer of software applications can be solved with the help of ADSA.

4 Agile and Enterprise Architecture

In conditions of great uncertainty Agile methodologies are most suitable for software engineering, which allow you to develop a software product without a clear understanding of what it will look like as a result of development. One of the main problems of Agile is the choice of the solution architecture, because architectural solutions are often impossible to change after they have been adopted [12].

One of the ways to solve this problem is to classify architectural changes into small ones that do not affect the main architectural solutions and can be implemented using point changes, medium ones, the feasibility of which is quickly determined by the chosen architectural style, and large ones, which cause the reconstruction of the basic structure of the EA, cannot be implemented by point transformations or adding components and therefore require serious costs and efforts [13].

The speed of implementing changes at the business layer by changing elements of the software application layer should be supported by the close interaction between granular services developed and implemented using Agile technologies almost on the fly and the maturity and richness of components, in particular, libraries that these services can use.

Therefore, when designing a basic EA, it is necessary to ensure a reasonable and effective balance between microservices, on the one hand, and libraries and stable components of another type, on the other. This balance will allow you to maintain a stable life cycle of the underlying AP and achieve a return on investment. EA, which is often a drag on Agile methodologies, must evolve from a restrictive function to a driving one.

As the popularity of Agile grows worldwide, there is a growing interest in Agile architecture, which ensures the quality of development processes and the result of a software product. At the Table 3 you can see the comparison Agile Architecture and Classical Architecture.

Table 3. Agile and Classical Architecture Comparison (based on [5])

Agile Architecture	Characteristic	Classical Enterprise Architecture
Dynamic environment Full Agile Agile EA	Environment and planning	Stable Environment Full planning Classical EA
Strategic Planning		
Low	Cost	High
Only key areas	Content	All company
Short	Planning horizon	Long
Image	Goal	Specific state
Implementation of innovations		
Iterative	Logical sequence	Sequential
Many	Number of options	Few
Conceptual	Scope of design	Detail
Ad hoc	Creating a project portfolio	According to plan
Budgeting		
Real Options Valuation – ROV	Budget planning	Annual planning
Architecture management		
Good communication	Management procedures	Regulatory procedures
Weak	Commitment to the plans	Strict
Architectural function		
High	Architectural competencies	Low
	Other aspects	
Simple	Tools	Complex

The microservice architectural model, an example of which is shown in figure 4, as a decentralized, flexible, but formalized model, is well suited for Agile methods of software engineering.

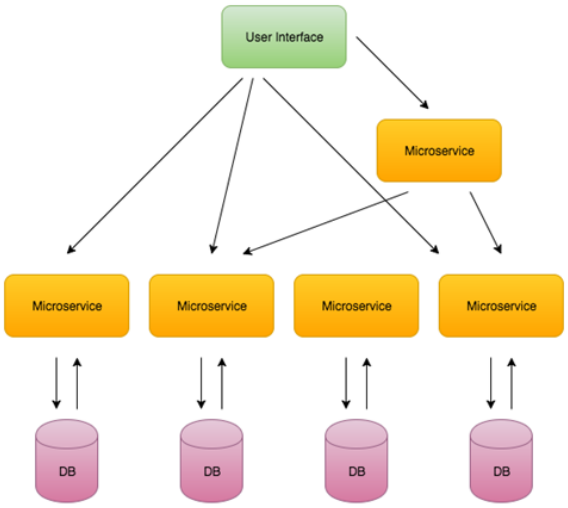

Fig. 4. Microservice Architecture

The microservice architecture is suitable for any type of application and is currently widely used for a wide variety of digital enterprise tasks. The following properties of the microservice architecture can be distinguished:

- modules can be easily replaced at any time: focus on simplicity, independence of deployment and updating of each of the microservices;
- modules are organized around functions: if possible, the microservice performs only one elementary function;
- modules can be implemented using different programming languages, frameworks, middleware, run in different containerization and virtualization environments, and function under different operating systems on different hardware platforms: priority is given to ensuring maximum efficiency for each specific function, rather than unifying development and execution tools;
- the architecture is democratic, not hierarchical: dependencies between microservices are peer-to-peer.

One of the reasons for using microservices is that a digital enterprise needs to be able to change quickly in order to respond to changes in the business requirements of customers, ensure interaction with partners and suppliers, and stay ahead of competitors. For microservices, more attention is paid to the connections rather than the specific implementation of components, so that the system remains stable when changing components.
In many ways, the philosophy of microservices repeats the philosophy of Unix, in which each program must do one thing, and do it well and interact with other programs in simple ways: microservices are minimal and designed to perform an atomic function.

Microservice architecture is directly related to the organizational culture of the organization, which should include a flexible, democratic model of software engineering, which involves not only developers, but also all employees of the organization, highly developed development and testing automation tools, as well as a design culture that requires the implementation of best architectural practices, accounting for previous errors, and excluding the use of inherited code in microservices themselves (since microservices are often easier to replace entirely), and transfer of the unadjusted code to libraries-components of other components of the microservices architecture.

A limitation of the classic microservice architecture is that it only affects the software application layer and only indirectly affects the business layer. One of the ways to overcome this limitation is to formalize the business model, for example, by integrating the microservice architectural model with the ADSA model.

5 Main trends in digital enterprise architecture

Taking into account the mass transition to Agile technologies in the field of software engineering and, in addition, for the implementation of projects in a variety of other areas of activity, we can predict an increase in interest in Agile architects. In this area, a large number of standards, frameworks, and methodologies will need to be developed in order to ensure the flexible and efficient development of digital enterprises. Agile architecture models will be developed according to types III, IV and V of table 1, and their structures will correspond to the structures shown in table 2. The Agile architecture model will include areas related to personnel, their motivation, and development. One of the most important characteristics of EA quality will be flexibility and its compliance with the goals, strategy and objectives of the enterprise.
The direction of decentralization of architectural models based on equal interaction of independent components will continue to develop.
It is possible to predict further enrichment of the service architectural style, standardization and formalization of SAUS, its consolidation with microservices architecture. In particular, the use of blockchain in the form of smart contracts for monitoring and managing SLA, as well as the organization of commercial settlements in cryptocurrencies on this basis, can be highlighted as a promising direction. A conceptual model of the smart contract architecture is shown in figure 6.

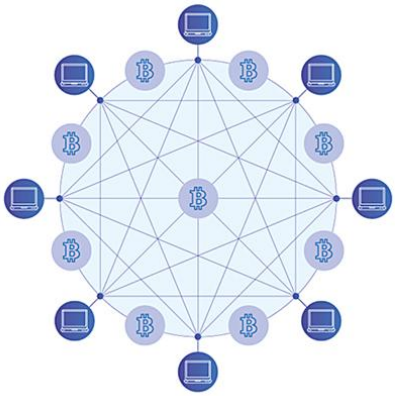

Fig. 5. The use of smart contracts for SLA control

You can also predict the dramatic development of data layer standards and frameworks, because this is necessary for building an Agile architecture and, in particular, for forming, controlling, and changing SLAs. The role of metadata will increase. Work will continue on the standardization of languages and their description notations. The role of data unification and unified international, state, industry, and corporate dictionaries will increase.
You can also predict the development and standardization of the lower Technological layer, and the further formalization of services that the upper architectural layers receive from this layer. You can also predict the appearance of a new type of OS that will include system services, in particular those defined in CORBA.
In order for architectural models to extend to an extended and ultra-extended enterprise, it is necessary to develop common standards for the interaction of architectural components, and providers need to learn how to adhere to them. The idea once started by OMG CORBA must evolve. Unified interfaces or services will allow you to build a truly flexible environment that will allow you to achieve the quality of architecture that the enterprise needs.

All this will require a clearer formalization of the corporate architect profession and the development of educational systems in the field of AP. For our country, we can predict the development of professional standards of AP, the formation of educational standards based on them, and the beginning of an independent professional assessment of specialists in the field of EA.

Conclusion

The article shows the inevitability of the development of the AE discipline in the direction of Agile architecture. In particular, it describes the microservice architecture, which is becoming the de facto standard in the field of software engineering. It is

noted that for effective use, it must be extended to the business layer, for which it is possible to use the ADSA and SLA as the basis for the interaction of layers, their individual elements and relationships.
Agile architecture should:

- provide the flexibility of a digital enterprise, while being resilient to errors and failures;
- make it easier to understand the needs of the digital enterprise, simplify the change of its elements;
- conform to a non-hierarchical organizational structure and help with teamwork.

References

1. Anshina M.L. Reference architectures of technologies used in digital transformation. In: Enterprise Engineering and Knowledge Management (EE&KM-2018). Collection of proceedings XXI Russian scientific conference, April 26–28, 2018. vol. 1, Pp. 10-14.
2. Anshina M.L. Extensions of the concept of service-oriented architecture to the architectural areas of the digital econom. In: Enterprise Engineering and Knowledge Management (EE&KM-2018). Collection of proceedings XXI Russian scientific conference, April 26–28, 2018. vol. 1. Pp. 335-340.
3. Anshina M.L. Structure and interaction of SLA ADSA in reference models of digital transformation. In: Enterprise Engineering and Knowledge Management (EE&KM-2019). Collection of proceedings XXII Russian scientific conference April 25-26, 2019, vol. 1. Pp. 83-91.
4. The Zachman Framework for Enterprise Architecture: An Explanatory IS Theory Aurona Gerber. Pierre le Roux, Carike Kearney, Alta van der Merwe. In: Conference on e-Business, e-Services and e-Society, April, 2020.
5. Hype Cycle for Enterprise Architecture, Santos, J., Allega, P., 2018. Gartner (2018)
6. What Is Agile Enterprise Architecture? Svyatoslav Kotusev (http://kotusev.com) Published in June 2020 by the British Computer Society (BCS) https://www.bcs.org/content-hub/what-is-agile-enterprise-architecture.
7. ISO/IEC 17788-2014 "Information technology - Cloud Computing - Overview and Vocabulary".
8. ISO/IEC 17789-2014 "Information technology - Cloud Computing – Reference Architecture".
9. ISO/IEC 19086-1:2016 "Information technology — Cloud computing — Service level agreement (SLA) framework — Part 1: Overview and concepts".
10. ISO/IEC 25010-2015 "Information technology. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE)".
11. How do Committees Invent?, Conway Melvin E., April 1968, Datamation T. 14 (5): 28–31.
12. Madison J. Agile Architecture Interactions. January 27, 2012. URL: http://www.infoq.com/articles/agile-architecture
13. Gregor S. The nature of theory in information systems. 2006. MIS Q. 30, 611-642. URL: https://doi.org/10.2307/25148742