Management 2014, 4(3A): 21-26
DOI: 10.5923/s.mm.201401.03

Increasing the Discoverability and Sales Potential of SOA Based Services through QR Codes

Deniz Herand[1,*], Berk Küçükaltan[2]

[1]Industrial Engineering Department, Turkish-German University, Istanbul, 34820, Turkey
[2]Brunel Business School, Brunel University, London, UB8 3PH, UK

Abstract Service oriented architecture (SOA) is generally used to facilitate the adaptation of the software to rapidly changing business processes. With the increasing use of service oriented architecture, the number of SOA-based services is growing rapidly. These services are usually provided from many other service-oriented architectures or similar software architecture paradigms such as SAAS (Software as a service), cloud computing and others with integration objective. At this point, the offer and discovery methods of SOA-based services play a very important role. Although specific standards to the subject matter have been developed, it still lacks a fixed and definite structure. This deficiency hinders the effective perception of services and this situation complicates the discovery of required services from software architects. The developed services will trigger interest and they can be readily sold in the software market in case the above-mentioned problems are eliminated. At this stage, the service description could be implemented using Quick Response (QR) - codes. As already known, they can be used in the sense of their features such as fast readability and large storage capacity. These QR-code features could be one of the best solutions for SOA-based services to take preferred position in the software market. As a result, software architects can obtain the required information about the services they are interested in very fast and this in turn will speed up their decision making process for purchasing.

Keywords SOA, Service Discoverability, QR-code

1. Introduction

In this study, the discoverability problem of the SOA-based services has been highlighted and a single solution has been proposed in several steps. The information about the existing services plays an important role in the discoverability of services in the SOA approach. In addition, this type of information should be rapidly and clearly obtainable. All these topics can be grouped under the discoverability title [1]. In this context, it is highly recommended to use QR-code technology and to develop these codes together with the development of associated services. The newly developed codes will be used for obtaining information about these services.

This study consists of six sections including the introduction. The SOA pattern is emphasized in the next phase in a way that it sheds light upon the trouble spots. The third section deals with the functioning of the QR-code technology. The fourth chapter brings into light the reasons for the need of QR codes in the SOA approach upon the Service Discoverability. What could be the solution to this problem is discussed in the next section. The final part discusses the result of the work carried out and possible future works which can be carried out on the basis of this study.

* Corresponding author:
herand@tau.edu.tr (Deniz Herand)
Published online at http://journal.sapub.org/mm

2. Service Oriented Architecture (SOA)

Today, software problems have many aspects. The fact that most of the developed software is not flexible, that software architecture documentation cannot be made in a simple and understandable way, that software cannot be effectively used and can become obsolete with time (not to mention cases where despite pending codes the same code are re-developed) can be shown as examples of such aspects. At present, all of these situations hit businesses in terms of high costs and wasted time [2].

Much progress has been achieved recently from architectural and technological point of view to solve these problems. From a technological perspective, the structure of programming languages moved from the assembly programming to the procedural programming and finally to the object-oriented programming structure where the latter became common for use. This phase transition was very helpful in reaching the solution of the problems mentioned at the beginning of this section. Since in the assembly programming there was a separate programming language

for each processor type, the software developer was not able to program for other processors after learning a processor-specific programming language. Such a structure brings the software developer many difficulties and limits the usage of the software [3], [4]. With the development of procedural programming languages in which the developed codes can be translated into the machine code, we came one step closer to the solution point. The development concerned has the dependency on processor-type and programming language of software developer eliminated. However, there are still a couple of important missing features in procedural programming languages, although the structure has allowed the applicability to different processors types and programming became easier. The most important missing features were the incomprehensibility and no documentation ability of the software. All these issues were on the next stage of the software development world, namely object-oriented programming, where a bit more was overcome. In object-oriented programming, the software developers are not stuck to a particular processor type and the codes are comprehensible and documentable. In the object-oriented programming, firstly abstract classes and the necessary objects of the classes mentioned (still abstract) are developed and then the objects are placed by programming for use [5]. Overall, to provide the documentation of object-oriented languages some notations have also been created. The structure of object-oriented languages allows the development of such notations and in fact, in this problematic area, these named developments have been very helpful. The open questions today are if last condition in software development can be improved and if the dependency by a single object-oriented programming language can be resolved because even though *object-oriented* the programs are developed with a single programming language and under normal conditions the combined use of programming languages is impossible [6]. This case can be presented as if one would have spoken to a person who speaks only one language with many different languages. One of the best solutions for this mentioned case, a single common language could be created to which all spoken languages must be translated simultaneously. In the software world this can be obtained through the software architecture on which the common language is XML (eXtensible Markup Language).

Along with the above mentioned developments in software development and the solution of the problems designated at the entry of this section, a lot of innovation has been implemented in the software architecture area. Although at the start of the software development process the developer did not pay a great attention to the software architecture as long as the processes have been realized, this area has reached today the point where it is payed the most attention [7]. In fact, the software architecture today is very necessary in the software world just as architecture in the building industry as software industry has now reached a very large size. The documentation of the software should be guaranteed by people who need the software and this documentation should be understood by the software developer. To create documentation of software is very important in many ways, but one could emphasize this importance with a small example. Software is created and treated by a software team and a company then uses this software to automate the business processes, and the company may be forced later to change the processes according to the needs of the market. Therefore, it would need a change in the structure of the software. In this case, if one assumes that the documentation of the software was not created, the change can only be done by the development team. Nowadays the process change requirements in this area have become common and after a certain time the same software developers are very unlikely to be in the same workplace. However, the quality of architectural planning is very important. Finding the right software architecture approach and its implementation according to the needs would also increase its efficiency [8].

In the previous paragraph the requirements for software architecture have been described. However, in this section the varieties of software architecture approaches and their advantages over each other are emphasized. One of the most widely practiced approaches to software architecture is the event-driven architecture in today's market. The event-driven architecture targets to get the inputs of the software functions from an event of the external world and the outputs then are calculated and returned according to the programmed functions [9]. In this case, the software will be developed in time responding to the needs and this brings nowadays certain big advantages. Service-oriented architecture is another software architecture approach that is often performed nowadays. In this software architecture style, the functions are grouped under service referred structures. These services are required to provide a conceptual integrity. Therefore, it is easier to manage the software [10]. The combination of the two types of architecture as event-driven service-oriented architecture approach that combines the advantages of these approaches has started to be used as a new architectural style. In such architecture, the main processes are set as event driven, the sub-processes, however, as service oriented [11]. This case allows software developer a structure which is flexible and which immediately responds to events and can also be controlled. However, although service-oriented architecture has many advantages over time, it requires at the beginning of the application a long-term planning. This means that the ratio of the efficiency of the service generated is in direct proportion to the quality of the planning time. Whether it is combined with an event-driven architecture approach or pure service-oriented architecture is applied, it is necessary that the plans are carried out in light of specific methods so that SOA can give out its best efficiency. Only the creation of Web services and their use may often cause inability to see the *big picture* and make full use of services. It may be that in many cases SOA can be a burden instead of benefit for developers and businesses. For these reasons, many SOA projects can be incomplete and/or abandoned [12].

In this phase, the overall structure of a service-oriented architecture is shown and its principles are emphasized. The strategic goals of a service-oriented architecture and the specific impact on the pursuit of these goals can be summarized as follows [13]:

- Strategic objectives of the SOA;
  - Increased Intrinsic Interoperability
  - Increased Federation
  - Increased Vendor Diversification Options
  - Increased Business and Technology Domain Alignment.
- The specific impact on the pursuit of the strategic goals;
  - Increased ROI
  - Increased Organizational Agility
  - Reduced IT Burden.

In this stage, the structure of a SOA that can show these effects is taken under consideration. SOA achieves all these benefits through its divided structure which can be easily assembled and disassembled as needed. The divided parts of a SOA are called services. However, these services get the desired structure by the design of the well-known web services in a different way. The SOA-based services are developed according to the requirements of eight basic principles.

These principles are as follows [1]:

- *"Standardized Service Contract – Services within the same service inventory are in compliance with the same contract design standards.*
- *Service Loose Coupling – Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment.*
- *Service Abstraction – Service contracts only contain essential information and information about services is limited to what is published in service contracts.*
- *Service Reusability – Services contain and express agnostic logic and can be positioned as reusable enterprise resources.*
- *Service Autonomy – Services exercise a high level of control over their underlying runtime execution environment.*
- *Service Statelessness – Services minimize resource consumption by deferring the management of state information when necessary.*
- *Service Discoverability – Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.*
- *Service Composability – Services are effective composition participants, regardless of the size and complexity of the composition". Service composability can be imagined as shown in Figure 1.*

Due to these principles, developed services are capable to solve most of the problems that were mentioned at the beginning of this section and may be met in the software world. To achieve all of these principles in a combined manner by a service unfortunately is not as simple as written in this publication. For this reason, it makes more sense to follow a specific and relevant methodology. As already mentioned, when a SOA is not properly planned and introduced at the beginning with a more accurate expression (for the case that it does not realize the above named principles essentially) it can cause more harm than advantage to a system. Unfortunately, when not acted in a proper way, SOA projects in many cases may be cancelled or postponed [14].

The most important principle for the purposes of this study is the principle of discoverability. In fact, performing a very effective information management is required in order to achieve a proportional qualitative SOA. The included functions in services and the input/output data types of these functions can be more clear and accessible through QR-code. This can make the discoverability principle more efficient and in this way the quality of the implemented SOA can be increased. One can obtain in the problem definition section more information about this topic.

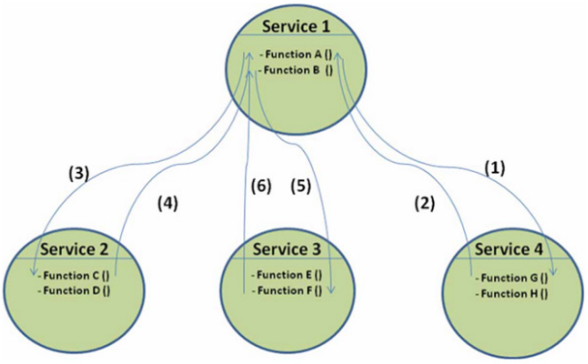

Figure 1. Imagination of service composability [2]

3. Quick Response (QR) - Codes

To use the previous link forms, the reader must remember the URL or have at hand the appropriate newspaper page. This can be by passed with a reference for the smartphone. In some Asian countries, such as Japan and South Korea a great extent of sales are generated already via mobile commerce. Mobile tag codes act as hyperlinks between the real and the digital world. The two-dimensional QR codes - pixel squares of black and white dots - located at shop windows and on billboards serve as a data carrier and can be scanned by passers-by with their smartphones, so they get into an online store [15].

The mobile phone must have a camera and code reader software. If one opens this application and keeps the smartphone over a QR code, the software scans the pixels square and decrypts the data. Then one will be forwarded directly to the corresponding website as shown in Figure 2 [16].

Figure 2. Imagination of Mobile Tagging [17]

The small pixel squares were developed in 1994 by the Japanese company Denso Wave. However, the codes are not subject to license fees as the company does not make use of its patent law. Additionally the commercial use of the squares is free [18].

This progression and fast mobile Internet access facilitated the rapid spread in Japan. In 2005, a survey of 7,000 Japanese Internet users came to the conclusion that over 73% of respondents use the codes, 96% were aware of their function. In November 2007, the first QR code in a German newspaper was printed. The WORLD COMPACT linked to the initiative of Deputy Chief editor Frank Schmiechen on the anniversary of the Berlin Wall with the pixel square on an online video of that press conference [19].

As indicated at the end of the previous section, this effective property of QR codes such as better and faster information obtainability about the products and affairs could also be used in applications to SOA. This issue will be discussed within the next problem definition phase in detail. With the expansion of the service market and in particular SOA-based services (which are used partially at present) the field of application of QR codes and mobile tagging may increase accordingly. After the service market is established, the QR codes that have been developed for each individual service could be placed in catalogues, so that the software developers can obtain information about them easily. Therefore, the main goal of SOA-rapid adaptation of software systems can in fact be met more easily which in turn can form a base for improvement of the quality of applied SOA.

4. Discoverability Problem of Services

As already noted, the actual effect of the benefits would be much lower than the real potential when SOA projects are not well planned. This would lead to *missing the big picture* of SOA and its main aspects which in turn will bring no benefits.

One of the most important issues in SOA projects is discoverability of services. Undiscovered services lead to extra costs as well as loss of valuable time. The discoverability of a service must be done in a simple way. Continuous and sudden changes in market requirements make it imperative to acknowledge this principle. The fact that changes can be carried out easily when needed, increases profitability significantly. It is important to point out that as long as companies can build their system structure on this basis they can gain a further competitive advantage.

According to the projected structure for discoverability of services (for which no specific standard has been developed so far) which is given in Figure 3, the projected structure consists basically of a provider, requestor and registry.

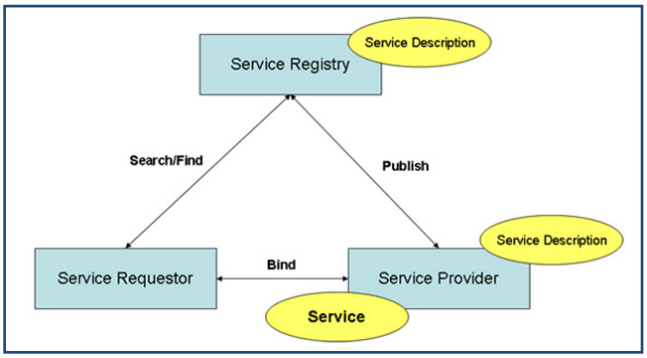

Figure 3. Projected structure for discoverability of services by SOA [20]

The task of the actors in this structure can be summarized as follows. Service providers can develop services themselves, implement or reach over the network and combine them to another new and more comprehensive service. Offered services are published by service providers in service registries. They are used by service users. A service user (eg. a software component) who needs a service has to search for a desired service in the registry. The service user receives from the registry the reference (address) of the relevant service provider from the registry under which s/he can access the service. Interaction with the service begins with querying the interface description of the service. This may include requirements for use of the service such as a form of authentication. It is common to transfer specific guidelines. If an agreement is guaranteed, then the user can connect to the service [21].

Such a structure provides the following benefits. In a company where SOA approach is adopted, the software project team which will probably develop the software project has to design primarily the software architecture. The project team will determine the functions and services required from it conceptually in the developed project. As a result of this determination, the services already obtained should be firstly analysed instead of going to direct development. These services can be examined by a service registry located in the company. Only after this step it should be decided if the services should be developed or bought from the market (in case the required ones do not exist among the analysed services). This can help to reduce the financial burden of a company by saving valuable time [14]. However, the number of services needed by companies today can be counted with thousands. Furthermore, the number of services which could be offered in the international market can reach millions. These can certainly pose problems such as not discoverability of services, not easy obtainable information about these services, difficulty to get such information in a quick and clear manner. One can

assume that these problems will increase exponentially in the future.

5. Increasing of Discoverability and Sales Potential of Services

In the light of the previous sections it is necessary to develop a solution that can ensure that relevant needed services can be found in time according to SOA approach developed services and information about them is obtained fast and clear.

In this regard, one can take advantages of today's trend themes QR codes and mobile tagging. Through QR codes, developers are able to get detailed information about the services which must be changed by required process changes. One can use the QR codes of services available on service catalogue. By using these QR codes one can jump right to the web page of service developers and get the necessary information. In fact, removing the URL processes to deal with Services could provide a higher speed.

At this stage we focus on the possible implementation process of the system. Especially, the software professionals will need to make a detailed plan for the development phase of the SOA-based system. In this planning process, it is necessary first of all to create the design of the system that will meet the market conditions. Since the system is designed SOA based, the necessary conceptual services are determined firstly by their functions. This phase will be completed with the determination of the input and output parameters of the conceptual services.

As a next step we should examine whether services are available with the features of the conceptual designed services in service-inventory. As already mentioned the use of QR codes could be ensured by examining the properties of these services. The task of the project groups may become much easier with the introduction of the standard in which the QR codes are created directly by the development phase of services. The software development groups can get information about the existing services using their mobile devices (which are located in a catalogue that can be used at the same time as a service inventory) after the first phase on which services required are determined.

All these operations can be applied in practice according to the following order. Firstly, the SOA planning of the system will be prepared. Then the required services will be determined. In the next step, the input and output parameters of the service functions will be determined. Services that are conceptually similar to the required services will be pulled out from the catalogs. After the elimination, information about the selected services will be obtained through QR codes as fast as possible. It can be decided than to use either in the catalog (if exists) whether externally provided services. Finally, the QR codes should be developed for the newly developed services to ensure their discoverability.

By taking the steps of the Figure 4, it seems possible to achieve a structure that can ensure the discovery of the required services on time and taking quick and clear information about them through mobile tagging.

Figure 4. A solution for the discoverability problem of services

6. Conclusions

This study highlights a designed system structure which can speed up the work of software development groups in accordance to the SOA approach developed systems. It was identified that the most important setback in such a system is discoverability of the needs to create effective services (after determining the missing ones). It should also be noted that one can meet a large number of services in the future that makes her/him feel like in a service pool.

As a solution, the creation of QR codes for already developed or developing services based on the SOA, could be of big benefit. In this way, the created QR codes of the services could be considered as ID codes. Therefore, it is believed that the services will not only be offered in the micro-based small businesses level but also in major international markets. It could be considered that the information about the services can be shared in a much faster and more efficient way through QR codes. This could be a development which may increase the quality of the evolving SOA projects.

As a result, the discoverability of services can be increased with the use of QR codes and also information about such services can be obtained in a fast and clear manner. In this way it can be ensured that the relevant SOA-based services are not missed substantially. The latter can help to avoid loss of valuable time and unnecessary business expenditure. This could enable companies to gain a strong competitive advantage.

One of the possible researches (which could be a continuation of this study) may be the practical application of this method and creation of an integration process for it. A further research would be calculation of the advantage degree of integration within a unit of time.

REFERENCES

[1] T. Erl, SOA: Principles of Service Design, Prentice Hall, Boston, 2007.

[2] D. Herand, F. Gürder, H. Taşkin, E. N. Yuksel, "A healthcare management system for Turkey based on a service-oriented architecture", Informatics for Health and Social Care, London, 38(3), 2012, 1-19.

[3] A. S. Berger, "Programming in Assembly Language", Hardware and Computer Organization, Elsevier, Burlington, 2005, 193-228.

[4] G. Jain, "Programming Languages", Real World Multicore Embedded Systems, Elsevier, Oxford, 2013, 289-312.

[5] E. G. Pinho, F. H. de C. Junior, "An object-oriented parallel programming language for distributed-memory parallel computing platforms", Science of Computer Programming, 80(A), 2013, 65–90.

[6] H. Jordan, G. Botterweck, J. Noll, A. Butterfield, R. Collier, "A feature model of actor, agent, functional, object, and procedural programming languages", Science of Computer Programming, DOI: 10.1016/j.scico.2014.02.009, 2014.

[7] H. Unphon, Y. Dittrich, "Software architecture awareness in long-term software product evolution", The Journal of Systems and Software, 83(11), 2010, 2211 – 2226.

[8] R. F. Schmidt," Software Architecture Definition", Software Engineering, Elsevier, Waltham, 2013, 305 – 322.

[9] J. Dunkel, A. Fernández, R. Ortiz, S. Ossowski, "Event-driven architecture for decision support in traffic management systems", Expert Systems with Applications, 38 (6), 2010, 6530-6539.

[10] M. Mohammadi, M. Mukhtar, "A Review of SOA Modeling Approaches for Enterprise Information Systems", The 4th International Conference on Electrical Engineering and Informatics, Universiti Kebangsaan Malaysia, 24 -25 June 2013, 794 - 800.

[11] M. B. Juric, "WSDL and BPEL extensions for Event Driven Architecture", Information and Software Technology, 52(10), 2010, 1023 – 1043.

[12] N. Joachim, D. Beimborn, T. Weitzel, "The influence of SOA governance mechanisms on IT flexibility and service reuse", Journal of Strategic Information Systems, 22(1), 2012, 86–101.

[13] T. Erl, SOA Design Patterns, Prentice Hall, Boston, 2008.

[14] D. Herand, SOA'nın İş Süreçlerine Uygulanabilmesi için Bir Metodolojinin Geliştirilmesi, Doktora Tezi, Sakarya Üniversitesi, 2013.

[15] D. Fleischhauer, "Phänomen Pixelquadrat", Lebensmittel Praxis, http://www.lebensmittelpraxis.de/handel/management /3398-phaenomen-pixelquadrat.html, accessed: 25.01.2013.

[16] S. Bär, "Crossmediale Verknüpfung von Kommunikationsk anälen–Wirkungen differenzierter Printverweise", Marketing Review St. Gallen, 5, 2012, 46 - 53.

[17] http://www.sfit.de/wissenswertes/146-qr-code-wie.html, Wie funktioniert ein QR Code?, accessed: 05.04.2014.

[18] K. Koch, "Fließender Übergang von Print zu Web. Per QR-Code schnell informiert", Publisher, 68 - 69, 2009.

[19] T. Vitzthum (2007), WELT KOMPAKT führt den 2D-Code ein, http://www.welt.de, accessed: 03.07.2010.

[20] http://jaxenter.de/artikel/Enterprise-Service-Bus-1, Enterprise Service Bus, accessed: 02.04.2014.

[21] W. Dostal, M. Jeckle, I. Melzer, Zengler, Service-orientierte Architekturen mit Web Services – Konzepte, Standards, Praxis, Spektrum Akademischer Verlag, München, 2005.