Delft University of Technology
Software Engineering Research Group
Technical Report Series

Adopting and Evaluating Service Oriented Architecture in Industry

Khalid Adam Nasr, Hans-Gerhard Gross, and Arie van Deursen

Report TUD-SERG-2010-001

TUD-SERG-2010-001

Published, produced and distributed by:

Software Engineering Research Group
Department of Software Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

ISSN 1872-5392

Software Engineering Research Group Technical Reports:
http://www.se.ewi.tudelft.nl/techreports/

For more information about the Software Engineering Research Group:
http://www.se.ewi.tudelft.nl/

Note: Accepted for publication in the Proceedings of the European Conference on Software Maintenance and Reengineering (CSMR), 2010, IEEE Computer Society.

Adopting and Evaluating Service Oriented Architecture in Industry

Khalid Adam Nasr
Software Engineering Research Group
Delft University of Technology
Delft, The Netherlands
Email: K.A.Nasr@tudelft.nl
Logica
Amstelveen, The Netherlands
Email: adam.nasr@logica.com

Hans-Gerhard Gross
Software Engineering Research Group
Delft University of Technology
Delft, The Netherlands
Email: h.g.gross@tudelft.nl

Arie van Deursen
Software Engineering Research Group
Delft University of Technology
Delft, The Netherlands
Email: Arie.vanDeursen@tudelft.nl

***Abstract*—In this paper, we present a descriptive case study covering the re-engineering and furhter evolution of adopting service oriented architecture (SOA) in industry. The goal of this case study is to identify the possible benefits and bottlenecks of adopting SOA, as well as to come up with best practices and research directions based on real-life experience gained in IT industry. The case involves an application portfolio of over 700 systems for a company in the transport sector.**

First, the case study involves the engineering of a portal application involving the integration of various services via the Enterprise Service Bus (ESB). Second, the case study is concerned with the setting up of a central coordination point within the organization to deal with SOA-related integration requests and requirements coming from different business units. Finally, the case discusses the actual implementation and integration of a service through the ESB. From this case study, a list of lessons learned and research directions is obtained.

***Keywords*-SOA; re-engineering; case study;**

I. Introduction

Today, many enterprises worldwide are focused on increasing their business flexibility and simplifying their IT infrastructure in order to better meet their business objectives. In a world of rapid change, business evolution is forcing IT to innovate constantly. IT systems need to be adaptable but also sustainable. The conflicting demands of flexibility, increasing business efficiency, reducing complexity of current infrastructure and minimizing software evolution costs bring forth a number of dilemmas that organizations are facing.

The Service Oriented Architecture (SOA) paradigm has been positioned as a promising way forward for organizations to solve business flexibility, efficiency and software evolution costs dilemmas. Through the flexible and proficient design, implementation and operation of SOA systems, organizations are supposedly able to adapt to the demands of rapidly changing environments [1], [2].

However, there is as yet insufficient solid empirical evidence that the positive assumptions made about SOA are justified in all cases [3], [4]. This paper will examine some of the assumed benefits and the potential bottlenecks of implementing SOA based on real-life experience gained in the IT industry with the implementation of a SOA solution. The case study will highlight the challenges that implementing SOA solutions pose for the IT industry and will contribute to developing a research agenda.

This paper emphasizes the importance of using the industry as a laboratory for research. Real industry cases can provide and support practical tests and contribute to the further development of SOA technology and the tools for its implementation. Such empirical research will also point at areas that are still not sufficiently put on the research agenda.

The case discussed in this paper concerns a project carried out by Logica in the transport sector. Logica is a major European IT and business services company, employing 40,000 people across 36 countries. It provides business consulting, systems integration, and IT and business process outsourcing services. The case discussed presents a three-stage process of a) designing a Proof-of-Concept (PoC), b) making organizational decisions for a smooth transition to a SOA, and c) the actual re-engineering and evolution of part of the system towards SOA.

Section 2 of this paper reviews related work. Section 3 presents the case study set up. Section 4 provides the case study report, which discusses the various steps of design and implementation. Section 5 discusses the lessons learned and relevant future research directions. Section 6 provides some concluding remarks.

II. Related Work

Over the past few years, the benefits and promises of SOA have been quite extensively discussed in literature [5], [6], [7], [8]. Vendors are busily promoting the hardware, software, tools and services that support SOA implementation [9], [10]. Some researchers have warned businesses not to blindly follow the vendor hype [11], [10]. In fact, an extensive literature survey was not able to quarry much empirical research that proves, or disproves, all the hypothesized advantages of SOA [12].

Kontogiannis, Lewis and Smith [4] point out that many of the case studies that do provide anecdotal evidence of the business value of SOA are sponsored by vendors. The number of independent industry case studies that document and discuss aspects of SOA implementation in practice, is still rather limited. To the best of our knowledge, these studies include [12], [13], [14], [15], [16], [17], [18], [19], [20]. Therefore, more rigorous empirical research - using the "industry as a research laboratory approach" proposed by Potts [21] - is needed before hard conclusions can be drawn about the reusability, increased agility and cost reduction that SOA promises.

Based on studies of five different companies that have implemented SOA, Schelp and Aier [18] argue that, while SOA appears to contribute to corporate agility, the reuse and cost cutting potential - as claimed by vendors and consultants - is less evident in all cases. In general, investments only pay off in the long run. The authors conclude that the measurement and evaluation of SOA is still 'in its infancy' [18].

Costs in software evolution are primarily dominated by software maintenance issues. Some authors argue that maintenance consumes up to 90% of the total software cost [14]. Canfora et al. [13] also looked into the question of cost-effectiveness, specifically in relation to the wrapping technique that the authors present. Their two case studies stress the need to devise methods for supporting the process tasks of reverse engineering and validation, as effort data showed that these were the most expensive tasks.

With their case study into Verification and Validation as part of a software development project for the US Department of Defense, Gehlot and Pujari [15] show the benefits of a model driven approach for the development of SOA systems. Zdun et al. [12], in their industrial case study within a large telecommunications company, also focus on a modeling approach. They present a model validation tool that the authors developed, which turned out to be valuable in order to support modeling the process-oriented integration of services.

The case study documented by Cuadrado et al. [14] describes the steps involved in the evolution of an existing medical imaging system, based on medium-sized legacy systems, to SOA principles. While it provides guidelines for practitioners on how to use certain tools, its focus on architecture recovery tasks is less relevant for our paper.

Lewis et al. [17] present a case study of migration of legacy components to services carried out for the US Department of Defense. The study suggested that disciplined analysis is needed to navigate the complex process. Therefore the authors further developed the Service-Oriented Migration and Reuse Technique (SMART), an approach for collecting the necessary information and method for identifying the risks of a SOA migration effort [22].

The case study presented by Kulkarni and Dwivedi [16] concerning a leading US financial Institution, openly discusses the risks of 'riding on the SOA hype wave'. Services that are, in theory, well-designed can, nevertheless, cause usage and maintenance challenges, once implemented. Like in [22] and [17], the authors present a framework for SOA adoption, called "InSOAP". They conclude that 'creation of web services in itself does not really help to deliver increased reusability, flexibility and responsiveness to change. This requires a strategic approach towards building a strong service foundation as well as sound engineering principles to realize it' [16].

In their empirical exploratory study, Kokko et al. [23] investigate SOA adoption in nine Finnish organizations by conducting interviews with key staff. SOA adoption challenges pointed out include resistance to change and unexpected extra work compared to traditional approaches.

In summary, the specific challenges for SOA that these case studies have pointed out include:

- developing a method that provides a clear framework, based on rigorous analysis, for the design, migration and implementation of SOA [16], [17],
- developing methods and techniques for expensive tasks of the migration process, in particular re-engineering and validation [13],
- examining the actual cost-effectiveness of SOA as expressed in its payback time [13], [18],
- managing organizational issues, such as resistance from staff, often especially senior members who may lack comprehension for web-based technologies [19].

One of the major challenges that is hardly, if at all, discussed in the above mentioned case studies, is testing. This issue has generally received little attention in research, with some of the notable exceptions being [24], [25], [26].

III. Case study setup

Enterprises and public sector organizations around the world are increasingly eager to embrace SOA, hoping to benefit from its acclaimed advantages, which promise increased business efficiency through more flexibility as well as a higher Return on Investment (ROI). The specific expectations toward SOA include: 1) achieving higher productivity by speeding up the introduction and implementation of new products, 2) simplifying the integration process during mergers and acquisitions, 3) achieving cost reductions.

However, there is not yet a wealth of solid evidence from industry that SOA really offers these assumed advantages in all cases and circumstances. In fact, it is our hypothesis that implementation and operation of SOA may well present more challenges than is currently envisaged. Important questions are:

1) To what extent have the claimed benefits of SOA actually been realized?
2) Which factors complicate implementation of and transition to SOA in practice?

3) Through which best practices can the advantages of SOA be maximized and its risks and costs minimized?
4) How can research help to enhance SOA development while further minimizing its risks and costs?

In order to answer these questions, we observe and reflect on a descriptive case study which consists of three phases [27]. For each phase we include a short description of the actual project: what was the goal, which stakeholders were involved, and what activities were employed. Furthermore, by evaluating each phase, we provide answers for questions 1, 2, and 3, covering benefits, challenges, and best practices identified in the case at hand. Question 4, concerning the research directions, is addressed in Section 5, as this requires a broader perspective.

IV. SOA in the Transport Domain: Case Study Reports

This section describes an industry case study about a SOA solution designed and implemented for one of Logica's customers in the transport sector, an organization which, for reasons of confidentiality, in this paper we will refer to as 'Ndi Moyo'.[1]The first author was involved in implementing the solution as an IT consultant. Ndi Moyo wanted to implement SOA in order to achieve better flexibility and agility, to reduce the complexity of its existing infrastructure and to minimize development and maintenance costs.

Integration of systems is a crucial aspect of SOA. Together with Ndi Moyo, the project team therefore started with investigating their existing IT landscape. This process was meant to gain a comprehensive understanding of its many different systems, to try and define specific functional and non-functional requirements, and moreover to explore the drivers behind the business initiative.

The investigation showed that Ndi Moyo's application landscape was a mix of many systems. These consisted of packaged and custom-made applications running on heterogeneous platforms using diverse data formats and technologies (J2EE, .NET, and various legacy applications). Over 700 applications were identified, mostly implemented as single monolithic structures and integrated in a complex point-to-point integration approach. The purpose of the overall project was to re-engineer this integration, in order to eventually achieve a well manageable, all-round integration infrastructure. The Enterprise Service Bus (ESB) was used as the middleware technology to expose all the applications as services.

Based on the first analysis, and because a full transition can only be made incrementally, the decision was made to start off with a pilot project instead of immediately initiating a fully-fledged design and implementation of SOA. The pilot led to a second project, which tackled the organizational structure necessary to support the transition to SOA. In the last phase of the project, the first integration solutions were implemented.

A. Phase 1: A SOA Proof-of-Concept at Ndi Moyo

The SOA transition project at Ndi Moyo started off with a pilot project for two different reasons. Firstly, as mentioned above, Ndi Moyo's existing application landscape was very complex and diverse and the project team therefore judged it wise to first start with a Proof-of-Concept (PoC). Secondly, the team realized it was important to actively involve the different business units in the process from the very start. This made it possible to explore, together with Ndi Moyo, their different motivations for this project, as well as the possible organizational and personal inhibitors to making it successful. Creating as much ownership as possible is crucial especially when a far-reaching change such as a migration to SOA is at stake [28].

The Logica team suggested to include a portal solution in the pilot project. This was because experience had shown that business managers as well as decision makers and staff can more easily relate to visual information, which is much more tangible, than to technical concepts. The portal solution clearly visualizes the possibilities of exposing business functions as services, spanning the whole organization. A portal provides authentication features and has personalization features to provide customized content to users. This content is presented as a variety of services. These services could be as simple as providing a corporate news item to staff of a global company. A service user sends an office location name and receives a report back on the latest business news for this specific location. Alternatively, portal users can interact with services by clicking links or submitting forms on the portal page, resulting in requesting services from service providers. All these concepts were well represented in the pilot project. See Figure 1.

Scenario: We implemented typical usage scenarios in the portal. One example is the one that visualized the following process: A customer wants to buy a ticket for a journey. He has already consulted Ndi Moyo's online timetable for his preferred date, time and means of travel. To buy his ticket he can choose between various payment methods (credit card, Ideal, company business card, bank card etc.). He can also choose his preferred way of receiving the ticket (E-ticket, mail, collecting from the customer service desk at Ndi Moyo). This scenario requires the interaction of at least the following company systems: Portal; Customer Relations Management (CRM) system; Online timetable; Payment provider; Revenue management system.

We used the Enterprise Service Bus (ESB) as the middleware technology to integrate these various services and to contain the differences between service implementations. As stated by Papazoglou et al. [1, p. 8] the ESB approach allows loosely coupled systems to take part in the integration

[1] Ndi Moyo means 'This is the life' in Chechewa.

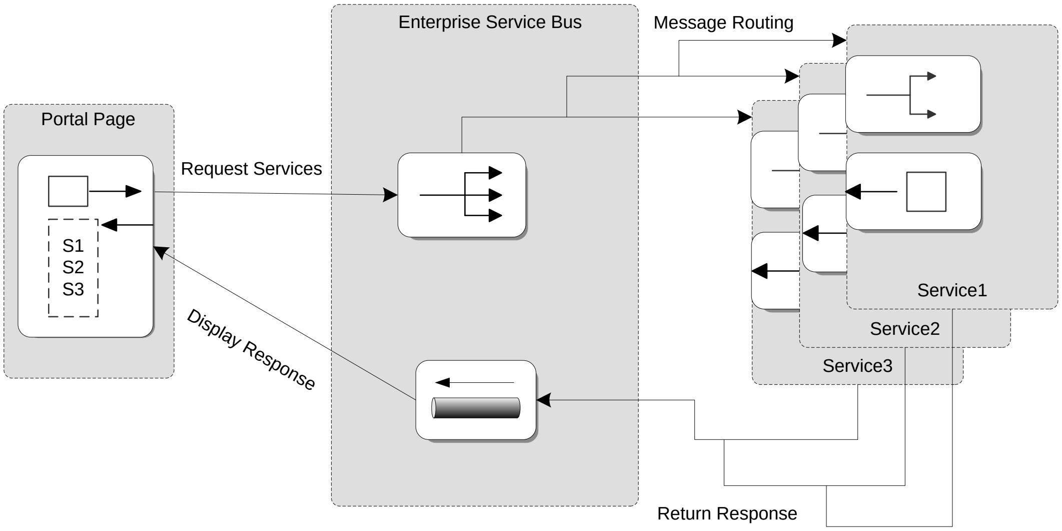

Figure 1. Architecture of the Portal Solution

and 'to break up the integration logic into distinct and easily manageable pieces' .

Benefits, challenges and best practices: From a business perspective, the Proof-of-Concept (PoC) examined the potential benefits and capabilities of SOA principles if applied to Ndi Moyo's requirements. The PoC moreover verified that the specific technology solutions operated successfully within Ndi Moyo's technical architecture landscape.

A common challenge during SOA transitions is the issue of ownership. In this first stage of the project it was evident that this would indeed become a challenge for Ndi Moyo. There was confusion over who would be responsible for the different services and their governance once these would have been integrated. This issue became a bottleneck in phase 2 (see below under B.).

To start small and design the SOA transition according to incremental steps proved a good practice. Moreover, the pilot portal indeed was a useful way of conveying the notion of SOA and its potential benefits for the business of Ndi Moyo. The selected technology worked and this formed the basis for a go or no-go decision. As it happened, the customer was keen on continuing with the SOA journey. The level of investment made by Ndi Moyo up to this stage was approximately one million euro. Ninety percent of this investment went to middleware infrastructure, management and licensing; less than ten percent was spent on professional resources to design and engineer the PoC. Given that the PoC was specifically meant to convince the various business units (BU) of the added value of SOA, it is interesting to note that Ndi Moyo had already invested approximately 900,000 euro before being assured of the cooperation of its internal organization.

B. Phase 2: Setting up the Integration Center of Competence (ICoC)

The second part of the project focused on preparing and organizing Ndi Moyo's internal organization to adapt to the SOA paradigm. Ndi Moyo's different BU have distinctive IT systems that provide various business functions spanning the organization. These are used by a wide variety of internal business units as well as by external partners and customers.

The nearly 700 different IT systems utilize a large number of diverse concepts, technologies and protocols. Every BU can make its independent decisions on how to deliver solutions. These solutions mostly concern separate business units without any reference to a centrally manageable and proficient integration infrastructure.

All this meant that to achieve a successful and controlled integration of these various systems through the Enterprise Service Bus, and to coordinate and organize the different integration requests put forward by the BUs, Ndi Moyo needed one focused central point where all the integration requests were collected. An integration request means that business unit A wants to connect its systems to business unit B in order provide, consume or exchange each other's services.

An Integration Center of Competence (ICoC) was set up to coordinate the integration requests and to keep an inventory of the application landscape and the different systems' readiness to apply SOA principles. Moreover, Ndi Moyo wanted to get better insight into, and control over, the high costs of software evolution and maintenance that it was facing.

Setting up ICoC was no easy task. First of all, the different BUs were apprehensive about the central role that ICoC was going to play regarding IT budgets and decisions. These BUs had been operating relatively independently in a decentralized structure, and did not want to give away the responsibility for controlling and managing their business services to a central third party. They were resistant to the idea that ICoC would take over the re-engineering of their systems.

Work process methodology: A work process methodology was developed to guide the decision making on integration, explicitly involving ICoC as well as the relevant BUs, as is visualized in figure 2. The initial request contains the vision and scope of the business scenario, the business rules and the software architecture of the system consuming/providing the service. In realizing this first step of the integration request the following roles from the BU are involved: business consultant, information analyst and software architect. They can consult ICoC for supporting them in this task.

The request goes to ICoC for validation. For this, ICoC developed a matrix that allowed easy insight into costs, effort and value of possible integrations as well as comparison between different integrations on these parameters. The selection criteria were based on an assessment of feasibility and reusability, routing, and estimates of the time and cost for re-engineering, implementation and maintenance. Moreover, validation includes determining whether the requested service already exists in the organization's service portfolio. If not, the appropriate BU system to provide this service is determined, and this BU will be engaged in the process.

If the request passes the validation, ICoC starts the High Level Design (HLD). This includes an impact analysis, business modeling, initial project plan, costs and resources. This HLD goes back to both BUs for approval.

If approval is given, ICoC prepares the total cost estimate, which again needs to go back to the requesting BU for approval. After this, a detailed project plan is developed, which includes interface definitions, supplementary specifications, resources planning as well as a test plan. In addition to the roles already involved, at this point, the release coordinator and project manager are assigned.

The next step is optional. If the request concerns an unknown integration (no previous experience) a Proof of Concept (PoC) is developed and tested. If it is a mainstream integration that this step can be omitted. If the PoC is accepted, the actual development starts.

Ndi Moyo invested approximately one million euro for this second phase of setting up and operating the ICoC; note that this covered only the external expertise hired.

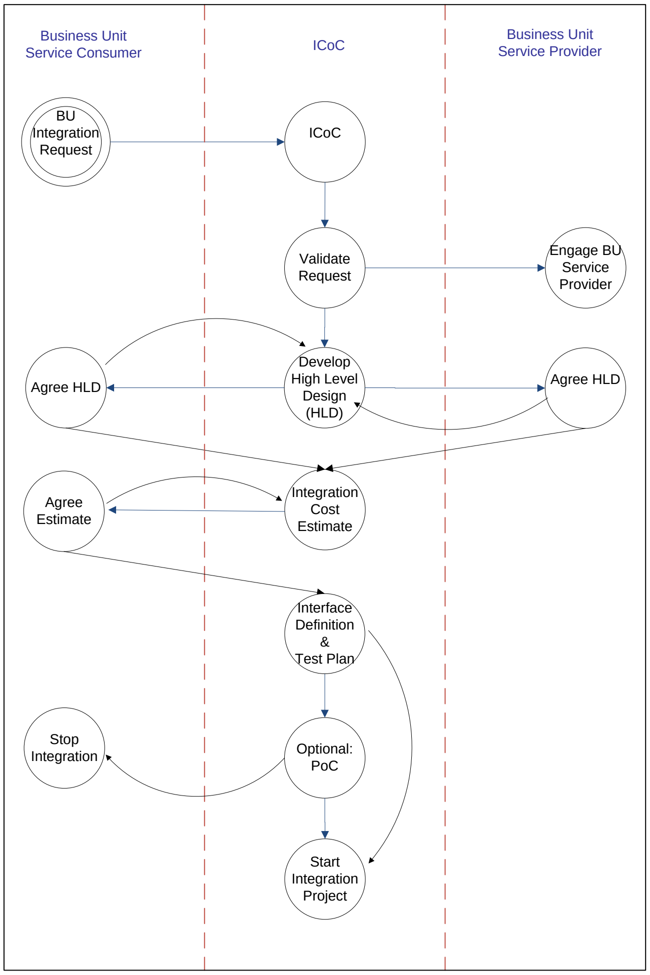

Figure 2. Work Methodology

Benefits, challenges and best practices: One challenge encountered in this phase was that the great majority of Ndi Moyo's systems were not mature enough to expose their business functions as services without being re-engineered according to SOA principles. In many cases, this meant that efort and money had to be invested in the re-engineering of these systems.

Secondly, a SOA transition is not only a technology challenge but also an organizational one. One tricky issue is how to deal with the resistance to change, and in particular changing responsibilities, that business and IT staff often display [23]. Resistance to change is a well-documented phenomenon and many of the methods suggested in the change management literature are applicable to overcome this also in a SOA context. In this project, the method mostly resorted to was creating awareness among the different stakeholders through their active involvement. Training of personnel to make them more confident about the transition also helped to take away some of the resistance.

Eventually, setting up ICoC as an independent, central coordination and assessment point was a successful way to manage the multifaceted process of this SOA transition. It is crucial that a central unit such as ICoC is an independent body, in order to diffuse the possible distrust between different BUs that hamper their willingness to cooperate. The work process methodology that was developed ensured the active engagement of the different BUs.

ICoC facilitated a proper coordination of the integration requests from different BU's. Moreover, it helped estimating the scope of a particular SOA integration as well as defining how to realize the integration in the most efficient and effective way. The structured methodology allowed Ndi Moyo to make informed go or no-go decisions in an early stage of the proposed integrations. Lastly, for the first time did Ndi Moyo have clear, centralized insight into the actual integration and maintenance costs of services of its different BUs.

C. Phase 3: Integrating a new service

Based on the positive assessment carried out by ICoC, it was decided to proceed with the implementation of several of the already incoming integration requests. The starting point for this was a preliminary classification of possible services that could be built from components of the existing system, based on the criteria mentioned above. This section briefly discusses one of the implementations.

Scenario: The example taken concerns a new service, namely the "report visitor" service, that was integrated into Ndi Moyo's existing corporate portal. This service, provided to the portal users, involves two business units. The first BU is the one responsible for the corporate portal. The second one (Facilities BU) is responsible for the visitor registration system. The portal user registers a visitor, or checks for available parking spaces in the visitor registration system. The portal communicates with the back end system of the FBU through the Enterprise Service Bus (ESB). The back end registration system in this example is the service *provider*, providing the business function (register visitor) as a service. This service is provided by an application that is described using the web service definition language (WSDL) and deployed as a Web service. The WSDL specification defines SOAP bindings for the service. The portal system in this case is the service *requester* and consumes the "register visitor" service. The WSDL service definition specifies the provider service interface and implementation that is used to access the target service. See figure 3.

Integrating this new service required the involvement of a software engineer for the portal, a software engineer for the back-end service provider application, an integration engineer for the ESB integration, an IT architect, a tester and a project manager.

Testing the integration: As part of the implementation of this particular integration, an integration test was included. The test descriptions were based on the functional requirements specifications, the acceptance criteria, and the provided scenarios. These tests were performed without back-end systems, but by simulating these systems using stubs. The scope for testing the system was mainly to test the functionality described in the provided interfaces. Functionality which did not result in any messaging traffic was outside the scope of integration testing, and was assumed to be covered by unit-level testing of the underlying components.

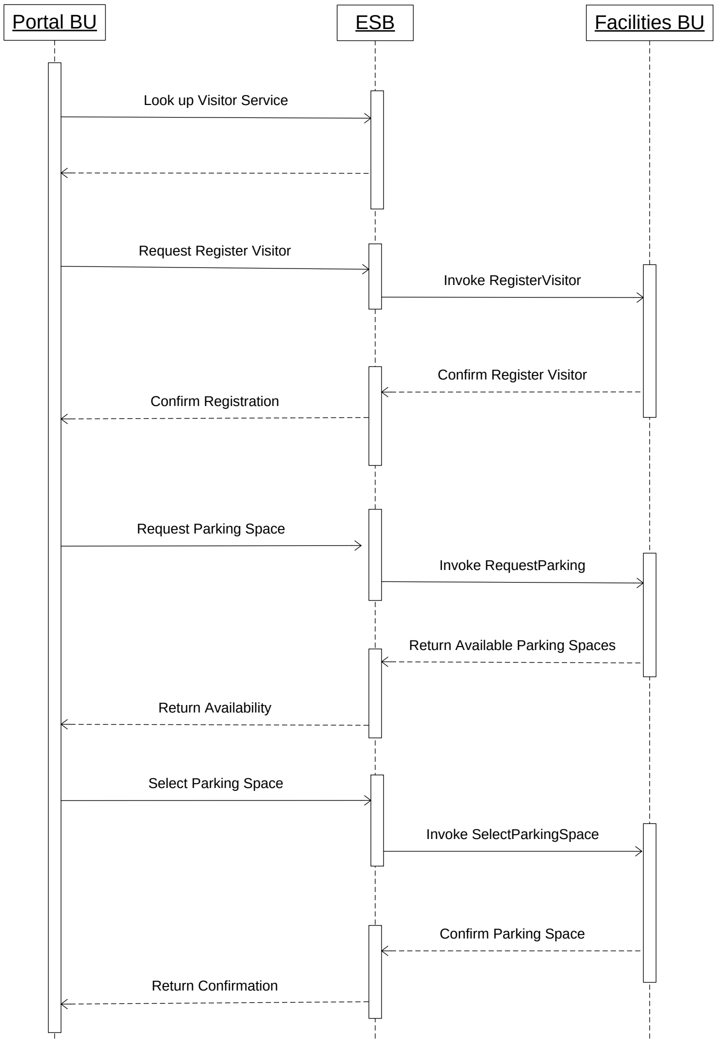

Figure 3. Service Provider-Consumer Sequence

For each test, a separate test data set was available. The test data included the message to be sent, a script to send the message, a script to save the result, results and expected results. Messages were simulated using a web service testing tool.

In this case, the service provider developed and tested the integrated service in its development environment. These tests are mainly on the service interface level. The service requester system is simulated. The test is fulfilled when the received data match the expected data defined through the service interface. How the service requester handles the received data is not dealt with at this stage, i.e., w.r.t Quality of Service (QoS), did the service fulfill its intended purpose?

The second stage is comprised of running system tests in the test environment where live data and systems (replication from production data and other systems) are mostly not present. The question of how services of different business units interact with each other and with the underlying systems is not tested until this stage. This last step of testing the interaction between the systems of various BUs happens in the acceptance environment where a snapshot of the production environment should be present.

Benefits, challenges and best practices: A business service was identified, developed and deployed, which made this business function available and which exposed it for use by other BUs. This successful real-life integration paved the way within the organization to proceed with other service integrations.

By testing this single new integration it was realized that doing the same in a context of many different simultaneous integrations would present serious challenges. There is as yet a lack of methods and tools for testing and managing the effects that one change can have on the overall system of systems. The impact of such a change will only be discovered during runtime.

The phased approach applied in this project enabled a thorough analysis as well as good coordination between the various stakeholders and a proper understanding of the well-trained experts. It led to a smooth real-life integration.

V. Discussion

Based on the lessons learned from the above case study and the existing literature, this section revisits the research questions posed in section 3 of this paper. It also throws up some urgent questions for further research.

A. Revisiting research questions

Without wanting to repeat all the above mentioned challenges, benefits and best practices, we summarize some of the most important lessons learned throughout the implemented case study.

1. To what extent have the claimed benefits of SOA actually been realized?: Proponents of SOA suggest that a migration to SOA will help organizations to cut costs, increase flexibility and achieve higher productivity by speeding up the introduction of new products and services. The case study presented in this paper does not allow us to draw final conclusions on all these points. The reason is that a full migration to SOA had not been completed at Ndi Moyo at the time of writing this paper. In fact, after a few integrations were realized, Ndi Moyo decided - for budgetary reasons - to postpone further implementations.

It is, thus, impossible to ascertain whether Ndi Moyo has benefited from higher flexibility and/or productivity since it started the SOA process. Assessments as to whether it will enjoy a reduction of total costs for re-engineering and integrating its systems, in order to meet rapidly changing business demands, cannot be made yet.

However, as a result of the SOA process, Ndi Moyo has gained better insight into, and control over, the expected costs of software evolution and maintenance. This is attributable to setting up the central coordination body ICoC, which developed a structured methodology to estimate the scope and costs of individual service integrations. This allows Ndi Moyo to make informed go or no-go decisions at an early stage of the proposed integrations.

2. Which factors complicate implementation of and transition to SOA in practice?: An important lesson learned is that implementing SOA is not only a complicated technical endeavour, but also presents issues of organizational structure and capacity, including the question of ownership. People have a natural inclination to resist change. Therefore, it is very important to involve, from the very start, not only all different BUs, but also the technical IT staff who will, eventually, be responsible for the integration. The efforts needed to get people on board and to ensure sufficient training of personnel for the use of tools and development should not be underestimated [6].

Secondly, a practical challenge that organizations may face is that a majority of their systems are not mature enough to expose their business functions as services without first being re-engineered according to SOA principles. If this turns out to be the case, organizations will have to make considerable investments to re-engineering these systems.

Thirdly, the rationale of migrating to SOA is to be able to manage multiple integrations at the same time. However, this presents testing challenges. Because, while testing single new integrations is not very complicated, doing so for multiple integrations at the same time is daunting. Methods and tools for testing the effects that one change can have on the overall system of systems are not yet available. This can jeopardize the benefits of SOA, especially when negative impacts of changes are discovered during runtime.

3. Through which best practices can the advantages of SOA be maximized and its risks and costs minimized?: Firstly, in this case study it was decided to implement the transition towards SOA in three phases. This strategy proved to be very useful, firstly because it gave the various stakeholders within Ndi Moyo the opportunity to get acquainted with the implications of the SOA transition. The first PoC phase, which involved the design of a visually appealing portal, helped to make the potential benefits of SOA tangible for decision makers, most of whom do not have a technical background.

Allowing enough time at the beginning of the project to introduce the SOA concept to all stakeholders involved, will pay off in the long-term. After all, a SOA transition requires the continued shared commitment of the whole organization.

Conducting a PoC also has an important technical function: it verifies whether the specific technology solutions can operate successfully within an organization's architectural landscape.

Moreover, the phased approach enabled Ndi Moyo to spread and control the required investments for the SOA transition. Incidentally, other organizations have also adopted this strategy [14], [28]. Equally important is that a phased approach allows transition to happen incrementally while core business can go on as usual.

Finally, the case study showed that it is crucial that a central coordinating unit such as ICoC is an independent body. If the BUs are not assured of this, it may be hard to overcome their resistance to giving away some of the responsibility for controlling and managing their business services to a third party. The work process methodology that ICoC devised actively involved the BUs in various steps of decision making. This helped to foster trust in, and ownership of, the SOA transition.

B. Research Directions

We fully support the directions for further research as suggested in recent literature [4], [6], [11], [29]. In this paper we emphasize three important areas for future research that follow from the challenges identified and the lessons learned through our case study.

How is impact measured?: Most of the SOA transitions, migrations and implementations that are carried out today claim to be successful, and vendors publicize this. The experience reported in this paper also suggests that the SOA transition performed was successful, because it was technically feasible and the client was "happy" and decided to continue with the further implementation. However, we need to ask ourselves what so-called "atisfactory" or "acceptable" results really mean? We believe that a much clearer distinction should be made between outcome and impact. A SOA transition can be (technically) successful, but is that necessarily an indication of achieved business goals?

Measuring ROI in software engineering is a delicate problem in general, discussed, for example, in the realm of software process improvement by Van Solingen [30]. As for the specific case of SOA, it has by now achieved a certain level of technical maturity, which requires similar investments in analyzing its true benefits for business. Much more rigorous research should be done, and tested in real-life situations, to establish whether organizations that made the transition to SOA are really better off, in the medium and long term, on the various aspects that SOA promised to deliver (increased flexibility, reduced costs, etc). Independent researchers and organizations should collaborate to develop this important research direction.

Automated runtime engineering: The industry report showed that an actual transition towards SOA is not a simple and straightforward exercise. The fact that SOA no longer deals with autonomous systems, but with systems of systems that together display a much higher degree of complexity, means that some of the traditional engineering methods for testing and the composition of services may no longer be suitable. SOA challenges us to execute and perform more of the engineering activities in an automated way and during runtime. New process models and product models need to be developed to enable this.

Tools and methodologies to reduce time and human resources have to be developed. Global costs and risks of SOA migration processes need to be assessed through case studies, in a similar way that this has been done for migrating legacy systems to more modern systems [31].

One area to which this applies in particular, and which is still unexplored, is testing. While SOA realizes the technological foundations for runtime reconfiguration and maintenance, adequate methods, techniques and tools for dealing with runtime integration and testing are still lacking. We are currently involved in an ongoing research project that investigates the primary industry challenges of runtime integration and testing. We believe that that runtime evolution and testing of SOA can only be implemented successfully if, especially, the communication between stakeholders, and the test-isolation and test-awareness of services are improved [32].

Empirical research: Finally, we urge that more case studies are conducted in order to achieve progress in the understanding and improvement of SOA development and implementation. This requires more collaboration between industry and independent research [4], [33]. If researchers are closely involved with industry, this allows them first-hand observation and investigation of the problems that occur during the SOA development and migration. This will help to identify the most urgent practical bottlenecks that need to be addressed in academic research.

C. Threats to Validity

The main threats to the validity of the results of a descriptive case study are concerned with repeatability (reliability validity) and generalizability (external validity) [27]. With respect to reliability, the confidential nature of the case is a clear threat. This is inherent to industrial SOA implementations, and hard to avoid. Moreover, the level of control in this kind of case study - especially because it involves a third party - will always be lower than in an independent experiment [34]. The potential threat to internal validity due to the first author's involvement in this project is countered by the fact that as a third party implementer of the solution, he had no direct business interest in its success or failure.

With respect to external validity, it is clear that more empirical studies will have to be designed and executed to extend the validity of the case study findings presented here. As mentioned in the introduction, several such case studies are already on the way; this paper is in fact the first of a series that examines SOA implementations in industry.

A strong point of an industrial case study like this one is that it avoids scalability problems [34]. It can be safely said that organizations of similar scale as the one studied here, will present similar challenges and results. Moreover, the results are unlikely to be domain related - i.e. they do not exclusively apply to the transport sector.

A possible threat to the external validity is the fact that this case study is based on only one actual new SOA integration. In reality, several such integrations were executed, and these did not throw up new challenges. Nevertheless, it is likely that if all 700 applications were to be integrated, this would lead to unexpected problems, either organizational (because of the number of people involved) or technical. As indicated before, testing in such a complex context may also pose new challenges.

VI. Conclusions

In this paper, we report on our experience in conducting a transition from a non-SOA multi-point integration to a re-engineered SOA ESB infrastructure in the transport and logistics domain. In particular, we describe a case, involving a proof-of-concept portal implementation, setting up an Integration Center of Competence, and the integration of a new service into the existing application portfolio.

We consider the following as our key contributions:

- The experimental design of our case study, permitting us and others to repeat similar case studies;
- The reports of the case conducted;
- The identification within this case of (1) actually achieved benefits; (2) best practices that helped to achieve the success; (3) challenges that require further attention;
- The identification of research directions addressing the challenges encountered.

Several avenues for future research have been sketched in Section V-B. We are in the process of setting up a next case study, focusing this time on monitoring variables that directly affect the return on investment. Another challenge that we will address in the near future is the development of test methods targeting SOA-specific faults, related to, e.g., service publication, discovery, binding, and composition.

References

[1] M. P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, and B. J. Krämer, “05462 service-oriented computing: A research roadmap,” in *Service Oriented Computing (SOC)*, ser. Dagstuhl Seminar Proceedings, F. Cubera, B. J. Krämer, and M. P. Papazoglou, Eds., no. 05462. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2006/524

[2] M. Lormans, H. van Dijk, A. van Deursen, E. Nöcker, and A. de Zeeuw, “Managing evolving requirements in an outsourcing context: an industrial experience report,” in *Software Evolution, 2004. Proceedings. 7th International Workshop on Principles of*, Sept. 2004, pp. 149–158.

[3] S. Cetin, N. Ilker Altintas, H. Oguztuzun, A. Dogru, O. Tufekci, and S. Suloglu, “Legacy migration to service-oriented computing with mashups,” in *Software Engineering Advances, 2007. ICSEA 2007. International Conference on*. IEEE Computer Society, Aug. 2007, pp. 21–21.

[4] K. Kontogiannis, G. A. Lewis, and D. B. Smith, “A research agenda for service-oriented architecture,” in *SDSOA ’08: Proceedings of the 2nd international workshop on Systems development in SOA environments*. New York, NY, USA: ACM, 2008, pp. 1–6.

[5] M. Huhns and M. Singh, “Service-oriented computing: key concepts and principles,” *Internet Computing, IEEE*, vol. 9, no. 1, pp. 75–81, Jan-Feb 2005.

[6] Z. Mahmood, “The promise and limitations of service oriented architecture,” *International Journal of Computers*, vol. 1, no. 3, pp. 74–78, 2007.

[7] N. Bieberstein, S. Bose, M. Fiammante, K. Jones, and R. Shah, *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[8] D. T. Sanders, J. A. Hamilton, Jr., and R. A. MacDonald, “Supporting a service-oriented architecture,” in *SpringSim ’08: Proceedings of the 2008 Spring simulation multiconference*. San Diego, CA, USA: The Society for Computer Simulation, International, 2008, pp. 325–334.

[9] K. Channabasavaiah, K. Holley, and E. M.Tuggle, “Migrating to a service-oriented architecture,” *IBM White paper*, 2004, http://www.ibm.com/developerworks/library/ws-migratesoa/.

[10] G. A. Lewis, E. Morris, S. Simanta, and L. Wrage, “Common misconceptions about service-oriented architecture,” *Commercial-off-the-Shelf (COTS)-Based Software Systems, International Conference on*, vol. 0, pp. 123–130, 2007.

[11] K. Kontogiannis, G. A. Lewis, D. B. Smith, M. Litoiu, H. Muller, S. Schuster, and E. Stroulia, “The landscape of service-oriented systems: A research perspective,” in *SDSOA ’07: Proceedings of the International Workshop on Systems Development in SOA Environments*. Washington, DC, USA: IEEE Computer Society, 2007, p. 1.

[12] U. Zdun, C. Hentrich, and S. Dustdar, “Modeling process-driven and service-oriented architectures using patterns and pattern primitives,” *ACM Trans. Web*, vol. 1, no. 3, p. 14, 2007.

[13] G. Canfora, A. R. Fasolino, G. Frattolillo, and P. Tramontana, “A wrapping approach for migrating legacy system interactive functionalities to service oriented architectures,” *Journal of Systems and Software*, vol. 81, no. 4, pp. 463–480, 2008.

[14] F. Cuadrado, B. Garcia, J. Dueas, and H. Parada, “A case study on software evolution towards service-oriented architecture,” in *22nd International Conference on Advanced Information Networking and Applications (AINAW) Workshop Proceedings*. IEEE Computer Society, March 2008, pp. 1399–1404.

[15] V. Gehlot and G. Pujari, “A case study in defining colored Petri nets based model driven development of enterprise service oriented architectures,” in *Proceedings 42nd Hawaii International Conference on System Sciences (HICSS'09)*. IEEE Computer Society, 2009, pp. 1–9.

[16] N. Kulkarni and V. Dwivedi, “The role of service granularity in a successful soa realization a case study,” in *Services - Part I, 2008. IEEE Congress on*, July 2008, pp. 423–430.

[17] G. Lewis, E. Morris, and D. Smith, “Analyzing the reuse potential of migrating legacy components to a service-oriented architecture,” *Software Maintenance and Reengineering, 2006. CSMR 2006. Proceedings of the 10th European Conference on*, pp. 9–23, 2006.

[18] J. Schelp and S. Aier, “SOA and EA - sustainable contributions for increasing corporate agility,” in *42nd Hawaii International Conference on System Sciences (HICSS'09)*. IEEE Computer Society, 2009, pp. 1–8.

[19] I. Wong-Bushby, R. Egan, and C. Isaacson, “A case study in soa and re-architecture at company abc,” in *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on*. IEEE Computer Society, 2006, pp. 1–8, track 8.

[20] J. Zhang, J.-Y. Chung, and C. K. Chang, “Migration to web services oriented architecture: a case study,” in *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2004, pp. 1624–1628.

[21] C. Potts, “Software engineering research revisited,” *IEEE Software*, vol. 10, no. 5, pp. 19–28, September 1993.

[22] G. Lewis, E. Morris, and D. Smith, “Service-oriented migration and reuse technique (SMART),” in *Software Technology and Engineering Practice (STEP), 2005. 13th IEEE International Workshop on*. IEEE Computer Society, 2005, pp. 222–229.

[23] T. Kokko, J. Antikainen, and T. Systä, “Adopting SOA - experiences from nine finnish organizations,” in *CSMR '09: Proceedings of the 2009 European Conference on Software Maintenance and Reengineering*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 129–138.

[24] G. Canfora and M. Di Penta, “Testing services and service-centric systems: Challenges and opportunities,” *IT Professional*, vol. 8, no. 2, pp. 10–17, 2006.

[25] S. Dustdar and S. Haslinger, “Testing of service-oriented architectures – a practical approach,” in *Object-Oriented and Internet-Based Technologies, Net.ObjectDays*, ser. Lecture Notes in Computer Science, vol. 3263. Springer, 2004, pp. 97–109.

[26] W.-T. Tsai, X. Zhou, Y. Chen, and X. Bai, “On testing and evaluating service-oriented software,” *Computer*, vol. 41, no. 8, pp. 40–46, 2008.

[27] R. K. Yin, *Case Study Research: Design and Methods*, 3rd ed. SAGE Publications Inc, 2003.

[28] M. Kajko-Mattsson, G. A. Lewis, and D. B. Smith, “A framework for roles for development, evolution and maintenance of soa-based systems,” in *SDSOA '07: Proceedings of the International Workshop on Systems Development in SOA Environments*. Washington, DC, USA: IEEE Computer Society, 2007, p. 7.

[29] M. P. Papazoglou and W.-J. van den Heuvel, “Service oriented architectures: approaches, technologies and research issues,” *The VLDB Journal*, vol. 16, pp. 389–415, 2007.

[30] R. van Solingen, “Measuring the ROI of software process improvement,” *IEEE Software*, vol. 21, no. 3, pp. 32–38, 2004.

[31] M. Rahgozar and F. Oroumchian, “An effective strategy for legacy systems evolution,” *Journal of Software Maintenance*, vol. 15, no. 5, pp. 325–344, 2003.

[32] M. Greiler, H.-G. Gross, and K. A. Nasr, “Runtime integration and testing for highly dynamic service oriented ict solutions an industry challenges report,” in *TAIC-PART '09: Proceedings of the Testing: Academic & Industrial Conference on Practice And Research Techniques*. IEEE, 2009.

[33] A. van Deursen, P. Klint, and C. Verhoef, “Research issues in the renovation of legacy systems,” in *Fundamental Approaches to Software Engineering '99*, ser. Lecture Notes in Computer Science, J.-P. Finance, Ed. Springer, 1999, pp. 1–23.

[34] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in software engineering: An introduction*. Kluwer Academic Publishers, 2000.

TUD-SERG-2010-001
ISSN 1872-5392