

Digital Transformation with Service Oriented Architecture and Performance Considerations

Ravi Kumar Gedela
Jawaharlal Nehru
Technological University
Hyderabad, Telangana-500085, India

Kamakshi Prasad Valurouthu
Jawaharlal Nehru
Technological University
Hyderabad, Telangana-500085, India

ABSTRACT

Digital transformation is disrupting the banking industry, and Service Oriented Architecture (SOA) is recognized as one of the key enablers. Interestingly, however, performance considerations are quoted by some on the pro side, but others, the con. This paper examines from the what, why and then the how, and specifically, how to accelerating digital transformation with SOA and predictive performance engineering. One of the top agenda items of digital transformation is customer experience. There are three challenges fundamental to legacy transformation and modernization initiatives – a) how to effectively leverage the existing assets in terms of logic and data, b) how we sustain the changes and c) how to ensure the improving customer experience by building high performance applications. The first challenge focuses on integration; second focusses on modernization and the third focusses on predicting and building high performance applications SOA is taking hold in the industry as the preferred new way of integration and modernization. For some banks, the big question remains: Will SOA die out as new technological innovations come out? This is alluding to financial derivatives that led to the financial market crash. One of such innovation is Micro Service proposition. Our research reveals that even sound architecture like SOA can be of hype for some banks unless a roadmap based on engineering principles is established and implemented. This paper reviews some of the failure and success factors across the industry and proposes a template to develop such a roadmap. What differentiates this paper from others is, however, a set of pressures or pain points, technical and business drivers, proven solutions and enablers (SDEP) that the authors summarized from their in-the-field and on-the-project experiences. Also, the authors have outlined the performance modelling exercise using Queuing Petri Nets.

General Terms

Service Oriented Architecture (SOA), Performance Engineering, Petri Nets

Keywords

Software Architecture, Service Oriented Architecture, Data Models, Canonical Data, transformation, Queuing Petri Nets and QoS

1. INTRODUCTION

Over the years, banks and insurance companies have built different systems to address different business needs, which resulted in duplicated data in silos. Master Data Management (MDM) solutions, or one-view of data initiatives, have been attempted to construct a single source of truth. Service based transformation is the process of integrating and modernizing legacy applications by following principles of Service

Oriented Architecture (SOA), to which MDM is positioned as the enabler. This paper provides a roadmap template for large scale enterprise application systems transformation to maximize business value from the silos.

It is observed in working with fortune 500 customers, the mother of all the business problems is ‘inflexibility’. Fundamentally, flexibility is the key to every organization’s profitability, longevity and success [1]. As the ancient Greek philosopher Heraclitus once noted, “Change alone is unchanging.” The inertia of doing things the way “they’ve always been done” makes companies ineffective, inefficient and inflexible. Even today, change is the one constant in all pressures of business. According to Gartner 2011 predictions, Banks and Investment firms’ core applications and core infrastructure are at risk [2]. Therefore planning for change is preparing for future and Service Orientation helps in building systems that can adapt the change.

It is also observed that the most of the financial institutions and insurance companies have attempted to implement Service oriented architecture (SOA) in order to have seamless flow of information between different departments and functions and for better control over the business processes. SOA promotes greater re-usability with in and across enterprises and flexibility to construct the business processes. The approach for designing services based on the standards is widely adopted in the industry because of the advantages. In spite of all these benefits, several analysts reported SOA adoption and modernization failures. Therefore right transformation roadmap is essential for successful implementations.

This paper is organized as follows. We begin with a brief literature review to emphasize the research context, and then we discuss our decision analysis framework for modernization / transformation and then discuss the proposed framework SDEP for legacy transformation. We then done the performance modelling of Order Management process to predict the performance, and finally end with conclusions and future study.

2. RESEARCH MOTIVATION

Legacy applications are the back bone for the financial enterprises as the core business is running on those applications and transformation of these applications are necessary to continue to be in business. When the legacy transformation initiatives are incepted in organization, there is likely to exist strong resistance from the existing employees due misconception of the definition of Legacy, as there are under impression that legacy is meant to replace the systems, there by losing job opportunities. But the definition is an illusion. The Legacy transformation is about maintaining and extending the value of this legacy investment through

transformation. These legacy systems in the existing landscape are not isolated, rather connected to several other systems with multi integration products and solutions. The “legacy” term is over used in IT, but the definition considered as the application developed with older technologies running on mainframes, and difficult to modify or replace them. Also some of the applications developed with different technologies which are no longer supported by the vendors. Now the biggest question for the C-level executives is how the business value can be extracted and extended from such systems? What are best ways to re-use the legacy investment? How to present the business case to the executive management for legacy investment present the business value and ROI to substantiate the numbers?

Successful Financial Institutions and Insurance companies want to implement innovative business models and design their business processes in such a way that they can differentiate themselves from their competitors. In order to be build/modernize the systems, it is important to consider the pain points, business drivers, technical drivers and best practices before making the transformation decision. It is revealed based on our survey that transformation with legacy investment re-use is the only choice left to the many of the enterprises with best-in-class technologies keeping in mind the change. In short, the legacy transformation process can be a cost-effective and accurate way to preserve legacy investments and thereby avoiding costs in developing new software with SOA. SOA is an architectural approach that aims to enhance the efficiency, agility and productivity of an enterprise [11].

The goal of legacy transformation is to retain the value of the legacy asset on the new or modernized platform. In practice this transformation can take several forms –rewrite / replace / re-use / transform, but this paper is focused on the transformation as the industry proven considered option is making transformation when compared to rewriting or replacing the legacy application. Enterprises have hard time to remain competitive and profitable with the new requirements in compliance and regulation in addition to meeting the Quality of Service (QoS) for the increase in volume of transactions and users. In order to support these agile requirements, most of the Enterprises intend to enhance use of Information Technology products and solutions to improve the efficiency, reduce the cost, standardize IT Infrastructure, processing, applications, databases, platforms and optimize through use of shared resources.

According to Gartner Industry Research 2011, one of the key issues for banking and Investment services is transformation of core applications and architecture [9], specifically architecture, data management and sourcing to address targeted revenues as their strategic growth. The primary intent for use of SOA is to move from large-scale, tightly integrated applications designed to meet specific functional needs to more modular, interoperable, and reusable functions[4][5][6][7]. According to Forrester Research study in December 2010, 80% of the firms use SOA in a production environment today and nearly 20% are exploring SOA for test purposes or within initial pilots [3]. The use of SOA as an IT design principle presents several opportunities for operational efficiency—specifically, SOA can lead to a reuse of IT assets (the encapsulated services), in turn lowering IT development costs, decreasing development project time, reducing development risk, and leveraging existing IT investments. Loosely coupled services reduce vendor lock-in and create a flexible infrastructure to distribute

processing, resulting in server cost efficiencies and greater reliability [8]. It is believed that, web services technology are appropriate in enabling flexible business processes, which can be integrated with those of organization’s partners on the fly. As well as, web services are considered as the most appropriate way to adopt Service Oriented Architecture (SOA) [10].

Over the past few years, the benefits and promises of SOA have been quite extensively discussed in literature [12], [13], [14], [15]. Vendors are busily promoting the hardware, software, tools and services that support SOA implementation [16], [17]. Some researchers have warned businesses not to blindly follow the vendor hype [17], [18]. In fact, an extensive literature survey was not able to extract much empirical research that proves, or disproves, all the hypothesized advantages of SOA [19].

Based on the consulting experience, we believe that SOA initiative can be used as platform to break down the barriers across the organization and beyond, bring the distributed capabilities together for the enterprise(s). The service composition provides a way to meet the business needs and addressing the agile requirements. The financial institutions can define, design, develop and deploy services and these services can be re-used from the repository to construct the applications. Bank customers can create situational applications by using the exposed services of the bank and the other services/external services (mashups). In short, SOA provides the customizable power to Financial Institutions customer and enable active participation.

3. CURRENT LANDSCAPE AND INTEGRATION TECHNIQUES AT FINANCIAL INSTITUTIONS

Making all the applications in the Enterprise work in an integrated manner to provide unified and consistent data and functionality is a difficult task. It involves integrating various kinds of applications, such as legacy mainframe application (IMS, CICIS), custom-built applications (C++/ Java,.NET), packaged applications (such as SAP, CRM). Furthermore, these applications are dispersed geographically and run on varied platforms. The following diagram depicts current state of IT in most enterprises.

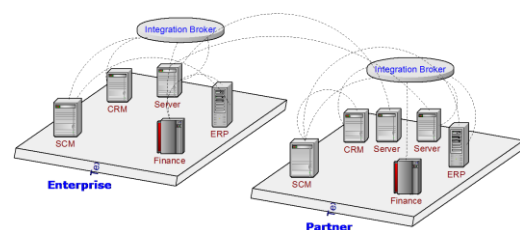


Figure 1: Current state of IT in a typical enterprise

Service oriented approach to integration is a side effect of building composite, loosely coupled service oriented processes. We’re composing processes out of services and then exposing those processes as services so other processes can consume them. The whole idea of service oriented integration is to compose services regardless of their underlying technology. It’s important to note, however, that if you take a service oriented approach to integration, it is not sufficient simply to wrap existing systems with standards-based interfaces and call them services. Loosely coupled composition enables Enterprises to support frequent changes in

the underlying systems, as well as changes to business processes without the need to make interface changes that break the loose coupling of the services. The real win with service oriented integration is in the dramatic reduction of cost at the maintenance and change phases of integration.

One of the North American Bank that we have consulted is considered to articulate the work. The Bank IT landscape consists of major sets of applications viz., mainframe (IMS/CICS), Java/J2EE applications, .NET application, AS/400 Systems. It is generally a challenge to integrate this sort of landscape. A Service Manager from iWay software can support different types of end points like file transfer protocol (FTP), Asynchronous messaging (JMS), etc. But for an agile enterprise it is required that the service provider and service consumer are loosely coupled. This requires the use of web services in the architecture. With the use of Web Services based SOA it is not a difficult task to support seamless integration across the IT landscape in the Bank.

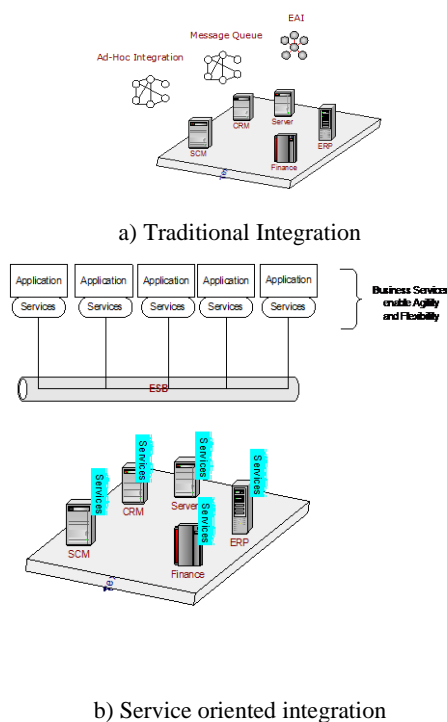


Figure 2: Integration Techniques

4. DECISION ANALYSIS FRAMEWORK

In the recovering market trends, making a business case with a suitable justification for financial investments and commitments is a biggest challenge for CIO/CTO and other senior leaders in legacy transformation as the legacy transformation is a critical activity coupled with rewards and risks. Therefore transparent picture of the transformation initiative would help senior leadership to make best-in-class decisions.

Based on our in-filed and on-IT Strategy consulting engagement experience, we propose 7-steps high level decision analysis work flow for transformation to derive the 3600 view. The steps are

1. Conduct Business Value Assessment
2. Conduct IT Systems, Quality Assessment

3. Review the outcome of 1 and 2
4. Consider the Industry/domain pain points
5. Consider the Drivers for Modernization
6. Provide inputs of Step 3, 4, 5 to the Business Case construction.
7. Once the Business Case is approved, use the proposed SDEP framework.

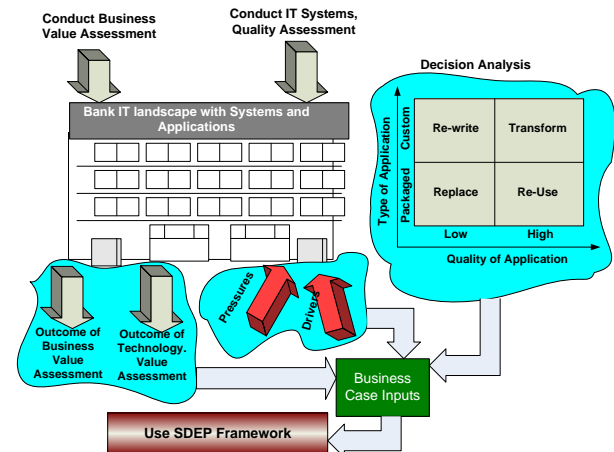


Figure 3: Decision Analysis framework for transformation

5. SERVICE BASED TRANSFORMATION

A transformation roadmap demonstrates the series of explicit activities required to achieve a particular stated objective, mapped to the series of tacit activities required to achieve the complimentary organizational cultural change. By harmonizing these two types of activities into a single transformation roadmap the risk of failure due to discord between them is identified.

Transformation requires as much planning and business involvement as any other IT project as it has a charter of specific objectives. Unlike the new project, the challenge is to use legacy assets foundation considering the pain points, technical, business drivers and use best-in-class practices to build effective and efficient solutions to promote agility. As a part of the transformation, some of the key activities are

- a) Develop transformation roadmap
- b) Use of transformation template, which is built on best engineering practices and project implementation experience
- c) Perform transformation enablement.

In order to support the above three activities some of the following actions are critical.

- i. Procurement of enabling technologies for services development
- ii. Data consolidation, integration and migration
- iii. Re-use promotion
- iv. Conduct a proof of concept (PoC) with the enablement technologies for risk mitigation and people, process, application and technology impact.

- v. Based on the experience in PoC, derive a linear implementation/transformation strategy.
- vi. Develop a reference architecture that validates the SOA design guidelines as a practical reference to working assets of the organization
- vii. Segment the current applications as being service providers, service consumers or both.
- viii. Come up with long term, mid-term and short term gains and risks.
- ix. Plan for governance

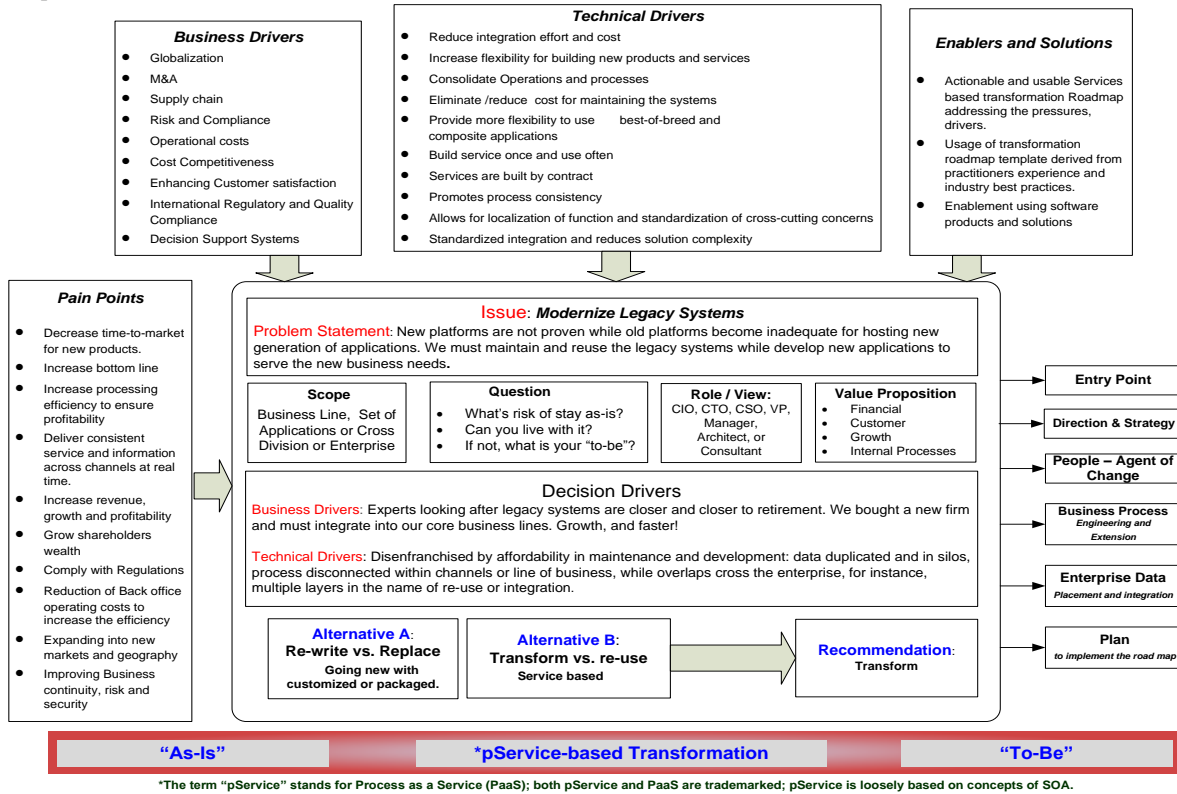


Fig 4: Service based legacy system transformation roadmap

5.1 Service Data Model

Forrester Consulting conducted a phone survey of 407 senior IT decision makers in the United States and eight Western European countries. The study was focused on companies with a minimum of \$250 million in annual revenues, and it found that manual efforts remain the dominant approach today for basic integration of data silos. The results revealed that data integration efforts are increasing across most major industries. 55% of respondents reported that they had undertaken four or more integration efforts over the past two years, and the majority of respondents do not see this number decreasing in the near future. 48% of those surveyed predicted that the number of Integration efforts would increase during the next two years, while 41% said it would stay the same. Since the applications are developed over a period of time, the

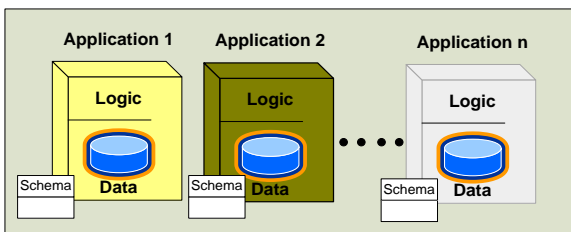


Figure 6: 'As-Is' in Bank landscape

data of the application is tightly coupled with the respective applications/logic as shown in figure 6. As a part of modernization, one of the key steps is data integration across applications and come up commonality of entities and initiates data integration as shown in figure 7. The most expensive and complex challenges of integrating business applications is ensuring the validity of data exchange between systems, and taking steps to ensure the validity of data through semantic integration is essential for SOA successful deployments.

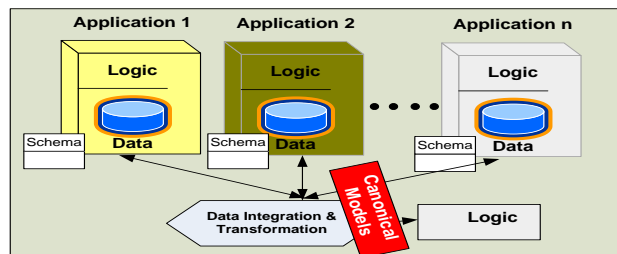


Figure 7: Data Integration between applications & landscape

In large scale enterprise systems, no solution can succeed without a sound data model. The service data model logically consists of four parts and conceptually captures four types of data: transactional data, metadata, business rules and analytical data. Analogous to MDM but without local storage, services access transactional data via a system of reference—

an aggregate pointing to physical sources—aggregating from and distributing to silos through integration infrastructure, for instance, currently existing connectors or messaging middleware, or newly installed service bus or cloud platform. Leveraged from MDM, or build anew, are utility services or batch jobs that manage the data, for instance, CRUD operations for customer and product information. Part and parcel of data management are data mapping. Data mapping, coupled with proprietary transforms, elevates service data model to the jurisdiction of service based transformation, to which the enabler is a trinity (legacy representation, object orientation and XML document) of the single truth (data) supported by schemas and constrained by protocols. This is the first part.

Second part is the metadata that models complex services, services that constitute workflow steps or compose business processes, which in turn are reusable services: a repository of pService. The concept of pService defines service based transformation and differentiates it from pure SOA integration, defining by integrating data and logic into services at a higher level, differentiating by better aligning with business processes. Building a repository of such services, however, requires a mature project delivery framework and an enforceable service governance process. In essence, this type of metadata captures the definition and granularity of services, the policy and security aspects of services, how services are used, their composition, annotation and versioning.

Third part captures business rules and forth part is for performance management. Business rules in this context capture the mechanics of service composition at design time and service execution at runtime, and govern how business processes flow through silos of data or islands of logic. Presented typically as snowflakes of statistics or via a dashboard, performance data measure delivered services to the roadmap in real time (not only quality, but also predictability): where we are now, how healthy the solution is, what the actual ROI is, and what should be our best next step.

Think of four types of data as the four quadrants of the service data model—once established, and converged into a holistic foundation, the envisioned enterprise landscape will emerge—like the roadmap of treasure (value) hunting, where all the hidden treasures (data) are ear-marked and all the routes (processes) are color-coded.

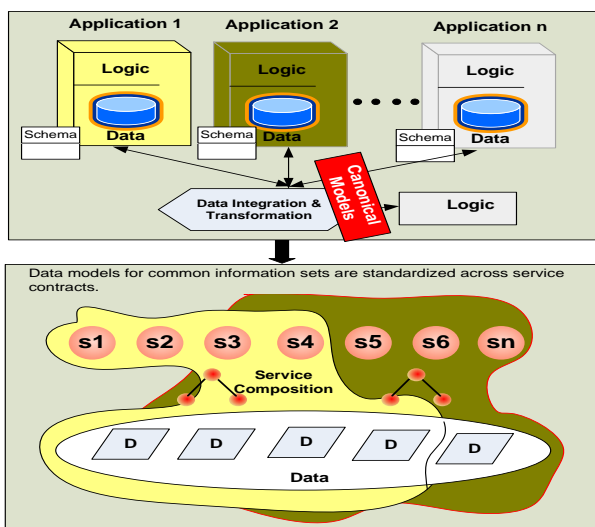


Figure 8: Data beyond boundaries with SOA adoption

The next step is to transform the siloed applications into set of application services (s1, s2, s3...sn) and data services based on standard SOA patterns and practices as shown in below figure 8. By making the data exposed as a service allowing the application owners can share the data and promoting re-use. ‘Re-use’ word is been overly re-used by IT community, but it is important to re-use the assets, both in terms of hardware and software to drive the cost lower. This is been one of the premise for Service based / SOA based software development of IT projects.

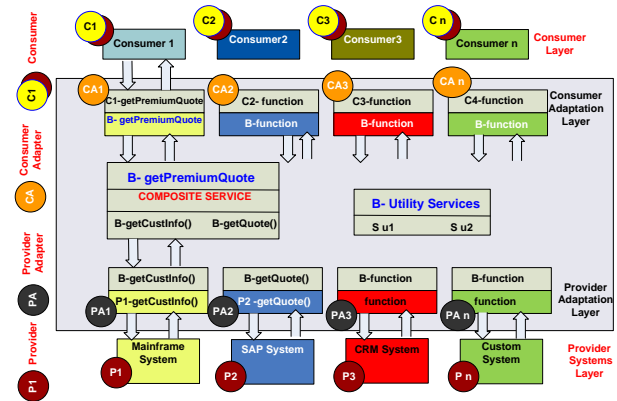


Figure 9: Generic Canonical Data Model

The diagrams above depict composite and atomic scenarios of obtaining “Premium Quotes”. The same approach can be easily applied to the other business functions. Because composites deliver a form of integration, a wrong assumption has emerged that SOA and Web services are the answer to application integration problems. Whereas SOA is helpful in some styles of integration (composite transactions); it should not be considered as a universal replacement. In the several situations involving composite SOA transactions, integration technology is may be required. In the context of canonical models, data re-use play a vital role. Data standardization is very critical in order re-use the data effectively, especially with services. Therefore if the canonical model is established right, then the P2P integration can be made loosely coupled with services.

The roadmap template should be used as a framework for transformation by all the stakeholders including technical and business. Also the service lifecycle is initiated by business people on the basis of a service contract template; the first step is a high-level description of the desired service characteristics. As described in figure 9, there are 2 set of layers, one set is consumer – provider and the other set is Provider adaptation and consumer adaptation. The adaptation layer would bring the standardization into the organization in terms of request- response transformation.

6. PERFORMANCE MODELLING AND CASE STUDY

The following section outlines 3 items – a) Order Management Business Process and Analysis b) Performance Engineering Methodology and c) performance modelling with Queueing Petri Nets.

6.1 Order Management Business Process and Analysis

Order management process is long and complex process involving a large number of activities, system-to-system interaction, and system-to-human interaction. The flow is as follows:

- Customer logs-in to the portal
- If authenticated the customer goes to next step
- The customer can search for products
- Customer can selected products to Wish List
- Customer has an options- exit after adding products to Wish Lists or place order of the selected products
- Invocation of place order will result in the system to fetch customer master
- The system will then validate the order; makes checks like credit limit, availability and so on
- System can then go to payment process

6.2 Performance Modelling Methodology

Performance modelling methodology suggested by Samuel Kounev [20], was adopted, to conduct the performance

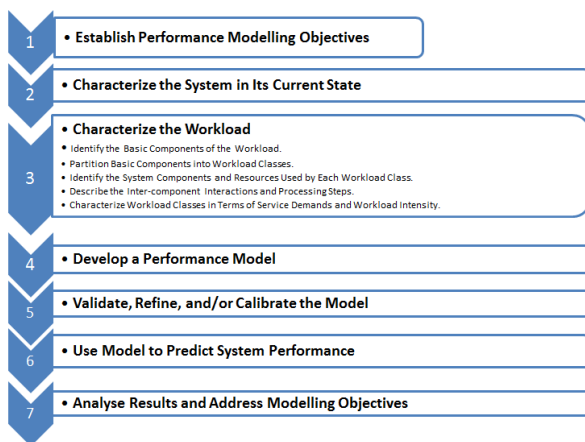


Figure 10: Performance Modelling Methodology

modelling of order management business process, a complex process which has several integration points with other third party systems.

6.3 Queuing Petri Nets (QPN) Model for Order Management Business Process

The motivation for this modelling exercise is the Order management business process that we have discussed earlier. The approach that we have adopted in designing the QPN model is based on the directions provided by the developer of QPME tool and illustrated in [20]. The first step to modelling is to understand the workload. The following workload classes were identified:

- Browse
- Search (for Products)
- Order (for Products)
- Payment

Figure 11 illustrates the tokens or colors selected for each type of transactions and sub transaction – “B” for “Browse”, “S” for “Search”, “O” for “Order”, and “P” for “Payment”. There could be a set of sub-transaction for each transaction. In order to make the performance model more compact, it is assumed that each server used during processing of a sub-transaction is visited only once and that the sub-transaction receives all of its service demands at the server’s resources during that single visit. This simplification is typical for queuing models and

has been widely employed. Similarly, during the service of a sub-transaction at a server, for each server resource used (e.g., CPUs, disk drives), it is assumed that the latter is visited only one time, receiving the whole service demand of the sub-transaction at once. These simplifications make it easier to model the flow of control during processing of sub-transactions. The assumption is, while characterizing the workload service demands the total service demand of a transaction at a given system resource is spread evenly over its sub-transactions. This allows us to consider the sub-transactions of a given workload class as equivalent in terms of processing behavior and resource consumption. Thus, we can model sub-transactions using a single token type (color) per workload class as follows: “b” for “Browse”, “s” for search, “o” for “order”, “p” for “payment” sub- transaction at once.

Name	Real Color	Description
B	Brown	Browse
S	Green	SearchTransaction
O	Blue	PlaceOrderTransaction
P	Red	ReceivePaymentTransaction
b	Magenta	BrowseSubTransaction
s	Black	SearchSubTransaction
o	Purple	PlaceOrderSubTransaction
p	Yellow	ReceivePaymentSubTransaction

Figure 11 – A screen shot of Colors (tokens) for each type of transaction

These simplifications make it easier business process management built on top of web services has become an important research subject. In this work we transform the basic BPEL structure to QPN model. In this work a QPN model is designed to analyze the performance of web services. Figure 12 illustrates the QPN model of order management process.

Figure 12 illustrates QPN model of order management process and its implementation. The client C is the input place and the number of concurrent users or tokens is defined as one of its properties. Starting with C client requests or sub-transaction token (b,s,o or p) is first sent to place represented by the Load Balancer. The load balancers are clustered here to take care of heavy load. The client request is serviced by the CPU (resources) of the load balancers. After that, it is moved to Web servers which are also clustered and, from there, it is routed to the process server which actually

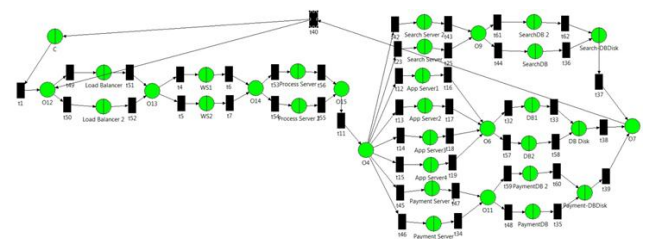


Figure 12- QPN Model of Order Management Process steps

Implements the BPEL Process and based on the process activity the request is routed to the appropriate application. For example, if the request is a sub-transaction of search then the transition t11 will fire only search transactions and they will be routed to Search Servers only. In this way it is possible to measure the performance of each and every application in the architecture.

The process server will route the browse and order sub-transaction to one of the N application server CPUs represented by places App Server1 to App Server 4. As

browse and order transactions have many workload classes it is required to provide higher computing capacities for these transactions. Transitions t12; t13; t14 and t15 have equal firing probabilities (weights) so that sub-transactions are probabilistically load-balanced across these application servers. This approximates the round-robin mechanism used by the load balancer to distribute incoming requests among the servers. Having completed its service at the application server CPU, the sub-transaction token is moved to place from where it is sent to database server CPUs and database disks with equal probability. Similarly, when a payment sub-transaction is sent to the process server it routes the request to payment application servers. All token service times at the queues of the model are assumed to be exponentially distributed. The following table provides description of the places in the QPN model used for modelling order management process and all the transitions are considered as immediate.

Place	Tokens	Queue Type	Description
C	B,S, O,P	G/M/∞/IS	Queueing place used to model Customer/client's concurrent requests.
Load Balancers	b,s,o ,p	G/M/1/PS	Queueing place used to model load balancers
Web Servers	b,s,o ,p	G/M/1/PS	Queueing place used to model web servers
Process Server	b,s,o ,p	G/M/1/PS	Queueing place used to model Process Servers
App Server (1,...,4)	b, o	G/M/1/PS	Queueing place used to model App Servers
Payment Servers	p	G/M/1/PS	Queueing place used to model Payment Servers
Search Server	s	G/M/1/PS	Queueing place used to model Search Servers
Search DB	s	G/M/1/FC FS	Queueing place used to model Search database server
Search DB2	s	G/M/1/FC FS	Queueing place used to model Search database server
Payment DB	p	G/M/1/FC FS	Queueing place used to model payment database server
Payment DB2	p	G/M/1/FC FS	Queueing place used to model payment database server
DB1	b, o	G/M/1/PS	Queueing place used to model database server
DB2	b, o	G/M/1/PS	Queueing place used to model database server
DB Disk	b, o	G/M/1/FC FS	Queueing place used to model the disk subsystem of the database server
Payment	p	G/M/1/FC	Queueing place used to

DBDisk		FS	model the disk subsystem of the database server of the payment application
Payment DBDisk2	p	G/M/1/FC FS	Queueing place used to model the disk subsystem of the database server of the payment application
Search DBDisk	s	G/M/1/FC FS	Queueing place used to model the disk subsystem of the database server of the search application
O12, O13, O14, O15, O4,	b,s,o ,p	----- ---	Ordinary places where sub-transaction tokens arrive. These places are used to design the clusters. The sub-transaction tokens that arrive in these places will be evenly distributed between the CPUs of the queueing place
O9	s	----- ----	Ordinary place where sub-transaction tokens of Search functionality arrive. The tokens are evenly distributed across search servers.
O6	b, o	----- ----	Ordinary place where sub-transaction tokens of browse & order functionality arrive. The tokens are evenly distributed across the cluster of 4 App servers.
O11	p	----- ---	Ordinary place where sub-transaction tokens of payments functionality arrive. The tokens are evenly distributed across the cluster of payment servers.

The same process was implemented using IBM process server with IBM suggested hardware. Once modeled, we have conducted the experiment and recorded the results for when we compared the simulated results with actual measurement of the scenario: B = 40; S = 20; O = 25; P = 20 concurrent users for each of these transactions. The results from the model are below:

Queue	Queue Utilization	Total Arrival Throughput	Total Departure Throughput	Mean Total Token Population	Mean Token Residence Time
Q0 (queue)		0.048	0.048		
Q1 (queue)	0.59	0.043	0.043	1.333	31.039
Q10 (queue)	0.511	0.006	0.006	0.979	177.286
Q11 (queue)	0.024	0.012	0.012	0.024	2.044
Q12 (queue)	0.974	0.026	0.026	18.495	723.726
Q13 (queue)	0.089	0.006	0.006	0.097	17.54
Q14 (queue)	0.491	0.051	0.051	1.001	19.585
Q15 (queue)	0.066	0.011	0.011	0.071	6.421
Q16 (queue)	0.19	0.012	0.012	0.232	19.489
Q17 (queue)	0.142	0.024	0.024	0.164	6.922
Q18 (queue)	0.024	0.012	0.012	0.024	2.045
Q19 (queue)	0.505	0.006	0.006	0.95	172.35
Q2 (queue)	0.588	0.043	0.043	1.319	30.669
Q20 (queue)	0.593	0.043	0.043	1.326	30.863
Q21 (queue)	0.586	0.043	0.043	1.314	30.631
Q22 (queue)	0.972	0.026	0.026	18.441	720.922
Q23 (queue)	0.088	0.005	0.005	0.096	17.505
Q24 (queue)	0.19	0.012	0.012	0.232	19.546
Q3 (queue)	0.59	0.043	0.043	1.332	31.054
Q4 (queue)	0.588	0.043	0.043	1.31	30.472
Q6 (queue)	0.673	0.013	0.013	2.027	158.346
Q7 (queue)	0.665	0.013	0.013	1.992	156.24
Q8 (queue)	0.668	0.013	0.013	2.024	157.728
Q9 (queue)	0.663	0.013	0.013	1.963	153.943

Results for the Simulation for the load - $B = 40$; $S = 20$; $O = 25$; $P = 20$

METRIC	MODEL	MEASURED	ERROR
	(%)	(%)	(%)
T_P	0.6	0.585	2.6
T_S	1.2	1.18	1.7
U_P	51.1	50.4	1.4
U_S	2.4	2.35	2.1

Comparison of Simulation results with Actual Measurements

(T_P – Throughput of Payment Server; T_S – Throughput of Search Server; U_P – Utilization of payment Server; U_S – Utilization of Search Server)

From the above table it is evident that the minor differences between simulated values from the model and the actual measurements. The error value between simulated values and measured values is less than approximately 2%. By varying the parameters in the modelling environment it is possible to refine the model so as to ensure accuracy of model results

7. CONCLUSIONS

In this paper, we have proposed a roadmap template for large scale enterprise applications transformation in banks and insurance companies. We used the template to examine our business and technical drivers in the context of numerous pressure points roaming around across divisions of a large bank. We strategically started by building up a data model, then a repository of pServices, and finally set the stage to transform legacy applications. Following the tradition of the IT industry, we abstracted our methodology as SDEP, denoting we must take steps to integrate our data and logic, and gradually modernize our systems. Currently, we are replicating our success to more applications in a large North American Organization, and would like to share our insights with fellow practitioners and decision makers in the industry. In addition to the roadmap, a detailed performance modelling exercise is also conducted and shared the results.

Different from technology-driven solutions, for example, SOA or MDM initiatives, SDEP aligns enterprise application transformation more closely with business processes towards a service based platform on which business services of mobile banking will be launched. However, we adhere to the principles of service orientation and master data technology lessons are learned and best practices are integrated to the steps of establishing the roadmap, planning the transformation, executing the prioritized tasks on demand of the business, and rollout the landscape step by step. We position a sound data model as the enabler of large scale transformation solutions, and we advocate services as the key

to strategically and innovatively transform the enterprise for forthcoming business expansion on mobile platforms.

8. ACKNOWLEDGMENTS

Our thanks to the tool QPME and Samuel Kounev for his extensive research insights.

9. REFERENCES

- [1] Jason Bloomberg and Ronald Schmelzer, “Service Orient or Be Doomed”, John Wiley & Sons Inc. 2006.
- [2] Gartner Report, gartner.com/predicts2011, Gartner’s Top Predictions for IT Organizations and Users, 2011 and Beyond: IT’s Growing Transparency, 2011.
- [3] Jost Hoppermann, http://www.forrester.com/rb/Research/soa_is_anything_but_dead_in_financial/q/id/58087/t/2, SOA Is Anything But Dead In Financial Services, Findings From Our 2010 Financial Services Architecture Online Survey, December 2010.
- [4] M. Ren and K.J. Lyytinen, “Building Enterprise Architecture Agility and Sustenance through SOA,” Comm. AIS, vol. 22, article 4, 2008; <http://aisel.aisnet.org/cais/vol22/iss1/4>.
- [5] R. Varadan et al., “Increasing Business Flexibility and SOA Adoption through Effective SOA Governance,” IBM Systems J., vol. 47, no. 3, 2008, pp. 473-490.
- [6] A. Arsanjani et al., “SOMA: A Method for Developing Service-Oriented Solutions,” IBM Systems J., vol. 47, no. 3, 2008, pp. 377-396.
- [7] S. Khoshafian, Service Oriented Enterprises, Auerbach, 2006.
- [8] Richard Welke et al, “Service-Oriented Architecture Maturity”, IEEE, 2011
- [9] Kristin R. Moyer et al, “Key Issues for Banking and Investment Services, 2011, Gartner Research, Publication Id: G00210761.
- [10] Deependra Moitra, & Jai Ganesh, (2005). Web services and flexible business processes: towards the adaptive enterprise. Journal of Information & Management, 42, 921-933.
- [11] Gonc-alo Ca’ndido, Jose’ Barata, Armando Walter Colombo, & Franc-ois Jammes, (2009). SOA in reconfigurable supply chains: A research roadmap. Engineering
- [12] 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.
- [13] Z. Mahmood, “The promise and limitations of service oriented architecture,” INTERNATIONAL JOURNAL OF COMPUTERS, vol. 1, no. 3, pp. 74-78, 2007.
- [14] 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.
- [15] 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.
- [16] 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/>.
- [17] 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.
- [18] 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.
- [19] U. Zdun, C. Hentrich, and S. Dustdar, “Modeling processdriven and service-oriented architectures using patterns and pattern primitives,” ACM Trans. Web, vol. 1, no. 3, p. 14, 2007.
- [20] Samuel Kounev, Performance Modeling and Evaluation of Distributed Component-Based Systems Using Queueing Petri Nets, IEEE Trans. On Software Engineering., VOL. 32, NO. 7, JULY 2006