A SOFTWARE DESIGN PATTERN FOR BANK SERVICE-ORIENTED ARCHITECTURE

C.K.Gomathy and.Dr.S.Rajalakshmi

SCSVMV University,Enathur,

Kanchipuram,TamilNadu, India.

ABSTRACT

In service-oriented architecture (SOA) relies on a proficient resolution come within to integrate and potentially distributed in the banking industry software design and enterprise. There is still lack of knowledge and service. Service oriented Architectures are exploring of great important role of evaluation, purchase and ongoing use of design pattern practices of the system. This paper focuses how the Self-Service and comprehensive venture design patterns, and the function incorporation techniques, can be used to start executing solutions of banking industry using the service-oriented architecture approach. However, there are many functionalities and deal with for software Architect services such as flexible, Reliability, and etc. SOA brings additional settings of proper governance of web service with interoperability of design pattern becomes a critical issue, however it can be successfully implement with an Architect for Service Oriented Pattern based enterprise can play in transformation terms applying the quality conceptual for framework.

Keywords

Quality Management in SOA, Service oriented bank computing, Patterns of SOA, service Architect banking management

1. INTRODUCTION

The position of architect is to evaluate techniques, problems and build solutions to solve them. Architect begins by gathering input of the problem, outline of the preferred explanation and any individual considerations or requirements that need to be factored into that solution balanced on power of internet. The architect then receives this contribution and designs the solution. This solution can comprise computer applications that address the problems by supplying the necessary utilities. In the real world, web apps are complicated. Service oriented architecture intuitively denotes the high level structures of a system. It is explored as the set of structures needed to reason about the pattern group, which hold the design factors, the relations between them, and the properties of both elements and relations. SOA restructuring to accommodate the new requirements due to the prospecting and technologies, policy and frameworks. Although universal banks are likely to endure to switch the largest share of the marketplace, community banks, industry specialists. Software architecture exhibits the following characteristics: mass of stakeholders, Separation of concerns, Quality-driven, Recurring styles & Conceptual integrity. These design patterns have been used, tested and polished by experts. A software design pattern is a repeatable solution for a commonly-occurring. Software architecture is “the overall structure of the software and the ways in which that structure provides conceptual reliability for a system”. Software architectural design pattern is immensely challenging, prominently domain based, bank, commercial, and extremely will source products to constrain with some exceptions of operational efficiency.

2. RELATED TECHNOLOGY

A handful of researches have been done in the field of software architecture since it has gained more importance with the development in computer technologies. Some of the recent researches are as mentioned below,

It has investigated the possibility of using some of existing principles for partial formal representation of SOA design patterns. In the area of enterprise applications integration effort is not focused on a formal way of work with patterns and pattern interpretations. Consequently, individual approaches represent and work with patterns in different ways. We do not consider different approaches as a drawback but we think that better results could be reached if one unified standard for pattern representation could be used. Approach in establishes formal pattern representation for banking process integration through Meta models in UML notations. These Meta models are used as bases for domain specific language. The language is however specifically fixated on these patterns. Therefore we prefer more general approach to formalizing SOA design patterns.

The existing proposed a research question of transforming dependability requirements into corresponding software architecture constructs, by proposing first that dependability needs could be classified into three types of requirements and second, an architectural pattern that allows requirements engineers and architects to map the three types of dependability requirements into three corresponding types of architectural components. The proposed pattern was general enough to work with existing requirements techniques and existing software architectural styles, including enterprise and product-line architectures.

The previous paper presented a survey of current component-based software technologies and the description of promotion and inhibition factors in CBSE. The features that software components inherit were also discussed. Quality Assurance issues in component based software were also catered to. The feat research on the quality model of component based system starts with the study of what the components are, CBSE, its development life cycle and the pro & cons of CBSE. Various attributes were studied and compared keeping in view the study of various existing models for general systems and CBS. When illustrating the quality of a software component an apt set of quality attributes for the description of the system should be selected.

The prior methods focus on a formal definition of architectural decision models as directed acyclic graphs with several types of nodes and edges. In their model, architectural decision topic groups, issues, alternatives, and outcomes form trees of nodes connected by edges expressing containment and refinement, decomposition, and triggers dependencies, as well as logical relations such as (in)compatibility of alternatives. The formalization could be used to verify integrity constraints and to organize the decision making process; production rules and dependency patterns could be defined.

A key aspect of the design of any software system was its architecture. An architecture description provides a formal model of the architecture in terms of components and connectors and how they were composed together. COSA (Component-Object based Software Structures), was based on object-oriented modeling and component-based modeling.

3. PROBLEM DEFINITION

In present time software architecture is a major issue in patterns and banking software organization, which develop banking software for some particular banking sector or firm. Lots of thing effect on software development life cycle. To design any software designs have to keep some point in mind develops effective software in reasonable time and cost. Here we described following issues, which have to remove at the time of software design phase:

- What is the most essential part for a software development in banking sector do to get the main out of its software architects and provide software architectures of the top essential quality?
- What should be steps to measure the capability?
- In what way the "theory of software architecture design pattern competence" look like?
- What are the possible banking service practices presently at work to enhance capability and efficiency?

4. PRINCIPLES OF SOA

The Knowledge has been recognized as part of the service-orientation design paradigm. Service-orientation represents a different approach to designing solution logic in support of a very specific set of goals.

4.1 Approaches in SOA

SOA is the aggregation of components that satisfy a design needs. It comprises components, services, and processes. Components are binaries that have a defined interface (usually only one), and a service is a grouping of components (executable programs) to get the job done. This higher level of banking application development provides a strategic advantage, facilitating more focus on the business requirement. SOA isn't a new approach to software design; some of the notions behind SOA. When it started drawing boundaries around software and providing access to that banking software only through well-defined interfaces. Software Architecture system leverages an investments in information and systems to give easy-to-use data that improves decision-making throughout a company. A service is generally implemented as a coarse-grained, discoverable banking software entity that exists as a single instance and interacts with applications and other services through a loosely coupled (often asynchronous), message-based communication model. The most important aspect of SOA is that it separates the banking service's implementation from its interface. Service consumers view a service simply as a communication endpoint supporting a particular request format or contract. How service executes service requested by consumers is irrelevant; the only mandatory requirement is that the service sends the response back to the consumer in the agreed format, specified in contract.

4.2 Designing Performance in SOA

Organizations today face significant challenges in their quest for high performance. They must understand the shifting technology landscape; harness technology for business needs; and implement increasingly sophisticated technology as a foundation for providing greater flexibility and improved results demanded by customers and shareholders. According to research, fully half of high-performing IT organizations are committing service-oriented architecture (SOA) to a significant portion of their design patterns.. But achieving high performance with SOA requires a strategic design pattern approach.

4.3 SOA Design

The SOA design helps organizations adopt new service oriented architectures by mapping this new architectural banking approach to business processes, technology initiatives and overall business and IT transformation. The reference architecture simplifies and the service-oriented architecture solutions and bank development to enable association to:
• Design and development cycles with reference patterns.
• Design and development on specific projects with a reference application designed and built by SOA experts.
• Rapidly create a well-designed architecture with proven SOA concepts and experiences.

4.4 SOA in Banks

Reputed banks created SOAs for their own infrastructures to increase re-use and reduce integration of their own applications. But as banks increasingly buy software off the shelf, there needs to be an industry-wide SOA standard so they can integrate third-party software easily. SOA frameworks break bank architectures up into services and define what data and integration these services require. In the past, if software was brought in from a third party there would be a large integration job. “If the software is Banking Industry in Network for SOA framework compliant it will be plug and play,” adding that a lot of time and energy is normally spent on integration. Banks are moving away from in-house development and towards commercial software to save money. There are many potential savings and banks comprise emerging markets for packaged software suppliers and banking industry is moving towards standardization of necessarily SOAs.

The underlying concept is that a bank with a desire to implement solutions based on a Service Oriented Architecture (SOA) must do so within the context of existing systems, business procedures and technology governance. Of necessity, then, adoption of SOA solutions is likely to be incremental and evolutionary. In order to be successful, the SOA solution(s) must and must be flexible enough to adapt to the bank's environment while at the same time moving the systems forward with respect to standardization and value-added functionality.

5. MAKING EFFICIENCY ROLE FOR SOA

The SOA Reference Architecture asset ensures that brought together in best processes from service-oriented architecture engagements worldwide and positions to clients to realize high performance faster and more capably than ever. It includes three parts, which together deliver comprehensive value by helping organizations overcome the most challenging obstacles in service oriented architecture development and deployment:

• SOA Delivery Architecture
• SOA Patterns
• SOA Platform

SOA Delivery Architecture consists of definitions and descriptions of how SOA fits across all application styles, including application frameworks, technical architecture and services that encompass SOA concepts. The SOA Delivery Architecture during the technical architecture phase to help organizations build the right SOA foundation. It will do by using this tools that significantly reduce project risks and make it more likely to achieve desired SOA benefits. In particular,

• Identify relevant architecture styles
• Generate a list of architecture services
• Architecture services, based on project requirements

SOA Patterns comprise proven approach for addressing common scenarios during SOA implementations, scenarios that may require pre-designed decision trees, functional designs and use cases. the patterns during the design phase to:
• Optimize design patterns so the need for redesign is reduced when performance or operational
Challenges occur.
• Harness experience gleaned from previous Accenture projects, empowering designers through better decision-making.
SOA Platform are acute to successful SOA projects and are split into three areas: Development architecture, execution and operations. (SOA projects with greater speed, efficiency and impact)
The diagram describes a situation when service transformation is needed because many different data service models are used and to take the efficacy for the same data by explore generate of data and using test case generation of proper utilized techniques. (Service in Established Schema pattern).

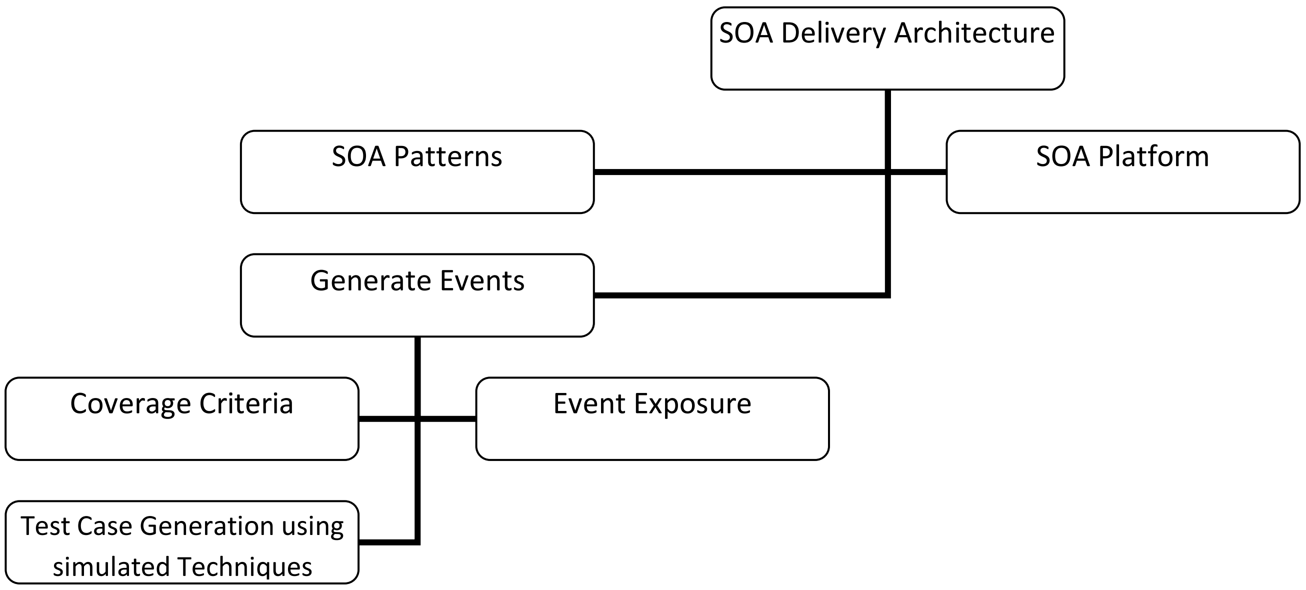
SOA Delivery Architecture
SOA Patterns
SOA Platform
Generate Events
Coverage Criteria
Event Exposure
Test Case Generation using simulated Techniques

Fig 1: Web service interaction in SOA

6. RULES FOR SOA DESIGN PATTERN

Pattern focused in preciseness, flexibility and tool support. Using rules for a partial formal representation of SOA design patterns as partial formal representation of patterns will be in a declarative form and in an adequate format. This will enable us to utilize tools for their elaboration. Thus we will be able to reach desired independence on the concrete implementation in the final product. In order to reach goal - SOA design patterns partial formalization, its defined transformation process from informal text pattern specification to rule based partial formal pattern representation. Results of this process are production rules and processes of pattern identification, which can be used in expert systems. It have realized and consecutively experimented with banking sector in consumer details of SOA design patterns. It has performed in following Manners:

Transformation process from informal text pattern specification to rule based partial formal pattern **Fig 1: Web service interaction in SOA** representation.
It defined transformation process in three steps. The descriptions of individual process steps follow.

***Pattern specification**:* Pattern can be specified in different ways. It defined pattern structure as:

Pattern =< Name, Icon, P, I, F > ------------- (1)
Where

P =< Summary, Draft, Example >-------------- (2)

I =< Requirement, Problem,

Solution, Context, Impacts >-------------------- (3)

F =< Effort, Application,

Specialties, Relationships >--------------------- (4)

Set *P* (Formula (2)) defines presentation specification, set *I* (Formula (3)) defines identification specification and set *F* (Formula (4)) defines application specification of patterns. Subsequently we create production rules based on identified objects, object relationships and pattern specification from step 2. Production rules (assigned to corresponding group) describe patterns in one of their state - problem, solution, context, and impacts. Then it define process for pattern identification for banking domains. This process is necessary and specifies the moment, when the certain group of rules is executed.

6.1 State of Research on SOA and Banking Industry

The banking industry is recognized as a technology and utilization of new information technologies with SOA. The applications and benefits of implementing SOA include the ability to build development enterprise system architectures which are able to support banking flexibility and speed. A adaption and active application of SOA is foundation for transforming banking model.

According the information systems have to couple in order to enable communication within channel with core banking applications.SOA provides an approach of customer information, product information and reduces the complexity.

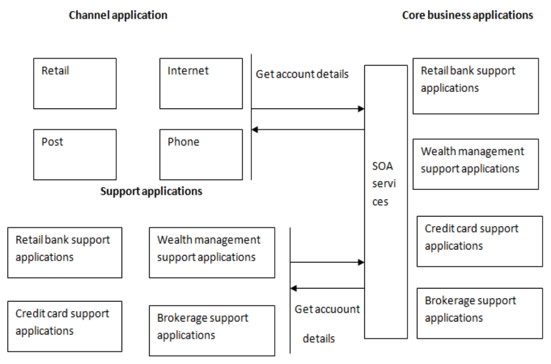
Channel application
Core business applications
Retail
Internet
Post
Phone
Get account details
SOA servi ces
Retail bank support applications
Wealth management support applications
Credit card support applications
Brokerage support applications
Support applications
Retail bank support applications
Wealth management support applications
Credit card support applications
Brokerage support applications
Get accuount details

Fig 2: Communication channels in SOA

6.2 Banking Value Services in SOA

The banking view of its information and the SOA layer makes it easier to integrate applications. The bank can offer its customers products and services. The information is shared in real time control within the application best suited to manage it.

Each time a service or channel is added. it can connect to and access everything else through the SOA layer. This helps improve flexibility, lower time and cost, reduce risk and optimize the value of each customers. It also permits the bank to pursue optimizes pricing based on a customer's entire relationship with the bank.

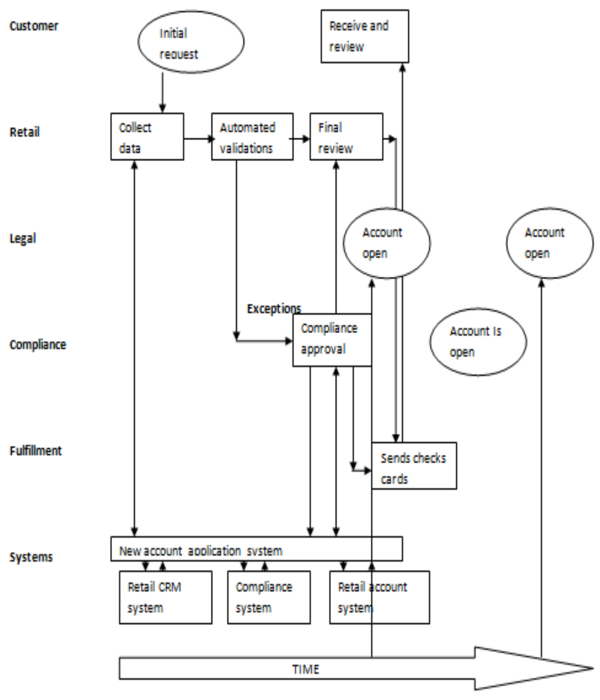

Fig 3: Integrate platforms in SOA

6.2 Quality Services in Bank with SOA

The quality aware software architecture remains as one of the basic needs while designing banking software. The software architecture remains to be a tedious job for the designers with the consideration of the quality metrics. Various quality attributes such as maintainability, reliability, readability, usability etc. have to be considered while designing software. In this work, proposed an efficient software architecture model based on Software design a pattern which is based on Service Oriented Architecture (SOA) with major consideration being the quality metrics. Architectural and design patterns are recurring solutions to software design problems. The SOA design is generally the way of designing a software system to provide service to either end user or other service through published interfaces.

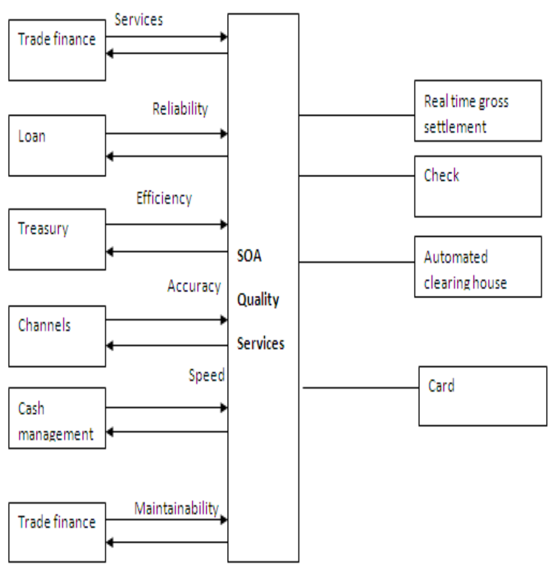

Fig 4: Quality methods in SOA

The usage of service oriented architectural patterns in design provides reusable and extensible technical solutions to common design problems in a standard Architectural format. The architecture design is also incorporated with the quality metrics like usability and portability to perform a better software architectural design. As the system operates based on the service oriented architectural banking design process, it have used application software as the input for which the architectural pattern is designed in support service construction, and banking logic.The SOA based design strategy can be more efficient in designing such application software's rather than other such methods.

7. CONCLUSION

This paper proposed a new quality architect paradigm to enable system quality to connect with software architectural models from which it is possible to extract precisely information. Our scheme has been proven quality in the standard model. A systematic complexity analysis and extensive experiments show that our proposal is also efficient in terms of computation and design. These features quality analysis banking framework scheme a talented solution to group-service oriented communication with access control in various types of design network.

8. REFERENCES

[1] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender Haluk Gumuskaya,"Core Issues Affecting Software Architecture in Enterprise Projects",World Academy of Science, Engineering and Technology, Vol.9, pp.32-37, 2005.

[2] Arun Sharma, Rajesh Kumar, and P. S. Grover, "A Critical Survey of Reusability Aspects for Component-Based Systems",World Academy of Science, Engineering and Technology, Vol.33,pp.35-39, 2007.

[3] Ghulam Rasool and Nadim Asif, "Software Architecture Recovery", World Academy of Science, Engineering and Technology, Vol.34, pp.99-104, 2007.

[4] Lihua Xu, Hadar Ziv, Thomas A. Alspaugh and Debra J. Richardson,"An architectural pattern for non-functional dependability requirements", The Journal of Systems and Software, Vol.79, pp.1370–1378, 2006.

[5] Jehad Al Dallal,"Software Similarity-Based Functional Cohesion Metric", Journal of IET Software, Vol. 3, No. 1,pp.46 - 57,Feb 2009.

[6] Iqbaldeep Kaur, Parvinder S. Sandhu, Hardeep Singh, and Vandana Saini ,"Analytical Study of Component Based Software Engineering",World Academy of Science, Engineering and Technology,Vol.50,pp.437-442, 2009.

[7]Olaf Zimmermann, Jana Koehler, Frank Leymann, Ronny Polley and Nelly Schuster,"Managing architectural decision models with dependency relations, integrity constraints, and production rules", The Journal of Systems and Software, vol.82, pp.1249–1267, 2009.

[8] Adel Smeda, Adel Alti, Mourad Oussalah, and Abdallah Boukerram,'Cosastudio: A Software Architecture Modeling Tool ", World Academy of Science, Engineering and Technology,Vol.49, pp.263-266, 2009.

[9] W.M.Abdelmoez, A.H.Jalali,K.Shaik ,T. Menzies and H.H. Ammar,"Using Software Architecture Risk Assessment For Product Line Architectures",In.Proc.of.International Conference On Communication, Computer And Power (Icccp'09) Muscat, February 15-18, 2009.

[10] Peter Eeles ,"Software Architecture Masterclass",In .proc.of IBM Rational Software Conference,2009.

[11] Outi Raiha, Erkki Makinen and Timo Poranen ,"Using Simulated Annealing for Producing Software Architectures",Thesis,Department Of Computer Sciences ,University Of Tampere, Apr 2009.