*IDEA GROUP PUBLISHING*
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

**ITB12877**

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2* edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

# An Architectural Framework for Web Services Based on SOA Realization in a Bank

Jakakumar Venkataraman & Sriram Anand, Infosys Technologies, Ltd., Bangalore, India

## INTRODUCTION

### Background

SOA represents the conceptual model of an enterprise where most of the collaborating systems produce or consume services that are loosely coupled entities representing coarse grained business functions. SOA implemented using web services leverages open standards to provide a flexible model of integration without a dependency on specific implementation technology. A fundamental element of SOA is the separation of concerns between service description, implementation, binding and declarative policies governing service interactions. In this paper, we discuss the key drivers for a SOA implementation in a typical bank and introduce an architectural framework that may be followed for web service implementation in the Bank. To illustrate application of the framework, we discuss a sample web service implementation with respect to cash management.

### Key Drivers for SOA in a Bank

Some of the key drivers for SOA in a bank are discussed in the following:

a. Banks and Financial services firms have traditionally been one of the earliest adopters of new technology. The typical IT landscape of any bank consists of technologies ranging from Mainframes, Windows Servers, Unix Machines and Java / .NET applications. Celent [1] in their Jan 2005 report the North American Perspective: IT Spending at US and Canadian Banks estimates that the Maintenance expenditures account for 78 – 80 % of IT budget allocation at the US and Canadian banks. While replacement expenditure is mainly in new technologies, the existence of mission critical applications on mainframes implies that they cannot be ignored.

b. Traditionally systems development was a decentralized activity and integration of the various systems across the organization was not given much priority. However with the rise of distributed computing technologies that allowed business functionality to be executed on different systems, integration became a challenge. While banks have been investing in integration solutions over the last decade and a half, many of the solutions were point-to-point connecting a few or a handful of applications. This resulted in islands of inter-connected applications, without any enterprise level integration.

c. Straight Through Processing (STP) / Real-time transaction processing: Banks have been investing in solutions that enable them to offer Straight Through processing / real-time processing of transactions end-to-end. This involves connecting a variety of applications across multiple technologies in the front office, mid-office and back office to ensure that data is exchanged in real-time and the transaction fulfillment is completed.

d. Multi-Channel Access: Banks have always been looking at more ways of connecting with the customer. The rise of the internet, IVR (Interactive Voice Recorder) and mobile technologies have meant that banks could provide customers access to banking products and services through more channels. While the customer uses any of these above channels he may still be accessing the same back-end systems i.e. accessing his account, placing a payment request or updating his personal information. Banks typically built individual interfaces to the back-end systems and have ended up with as many interfaces as there were channels.

e. Unified customer view: With global banks a customer may interact with multiple arms of the bank. For example, the customer may be a key client of the Private Banking Group and may enjoy privileges but he may be a small customer for the Cash Management & Payments Group of the bank. Without a unified view of the customer, the Cash Management Group will not be aware of the customer's privileged status in the Private bank. This may lead to service dissonance and customer dissatisfaction. To achieve unified view, banks are investing in solutions that necessitate integrating various applications that house customer data [8].

f. Strategic value of IT: Increasingly banks are taking a strategic look at their IT investments. However competition for budgets between the various business units and IT departments implies that the CIO / CTO are expected to show higher ROI and shorter pay-back. In addition to the above, CIO/CTO must ensure that the investments are future proof to ensure that the bank can adapt to the changing business needs effectively.

Service Oriented Architecture using Web Services provides the following 3 key benefits [4]

a) Easy Integration: - an architecture that is technology independent and allows any application irrespective of its underlying technology to be plugged in and out easily

b) Re-Usability – reduce costs through re-usability of applications and IT assets so that duplication of effort can be avoided leading to overall cost reduction.

c) Flexibility – through loose coupling of services that enables flexibility in their orchestration and modification to make business process changes...

### Factors in Favor of Web Services

It may be reasoned that for many of the drivers cited above enterprise integration solutions may alleviate some of the problems. In many cases, the problems faced by large banks are related to the proliferation of systems, technologies and processes. One approach to address this issue is to deploy an enterprise integration solution such as an Enterprise Application Integration (EAI) product. Leaders in this space include products from Webmethods, TIBCO, IBM and SeeBeyond (now a part of Sun Microsystems). Other options include the development of custom solutions based on messaging platforms such as IBM MQ Series. However, it needs to be noted that in both these approaches there is a substantial amount of "lock-in" with respect to the technology, approaches and skill sets. The enterprise architecture will be dependent on the nuances, releases and upgrades of these products. Apart from this disadvantage, cost is a key factor as well for these types of solutions. An enterprise integration solution such as this can also constrain banks from

being able to collaborate with inter or intra enterprise applications due to the technology considerations. Web services offer a level of independence from these constraints due to their dependency on open standards and the adoption by leading technology companies. The other compelling reason to adopt web services is related to the ease of converting component based applications into web services. Currently, most development platforms allow the simultaneous development of applications as services and regular components. Therefore, web services as an integration strategy makes sense and can provide banks with the flexibility they need to meet the complexities of today's business.

**Service Oriented Architecture Framework (SOAF) Overview**

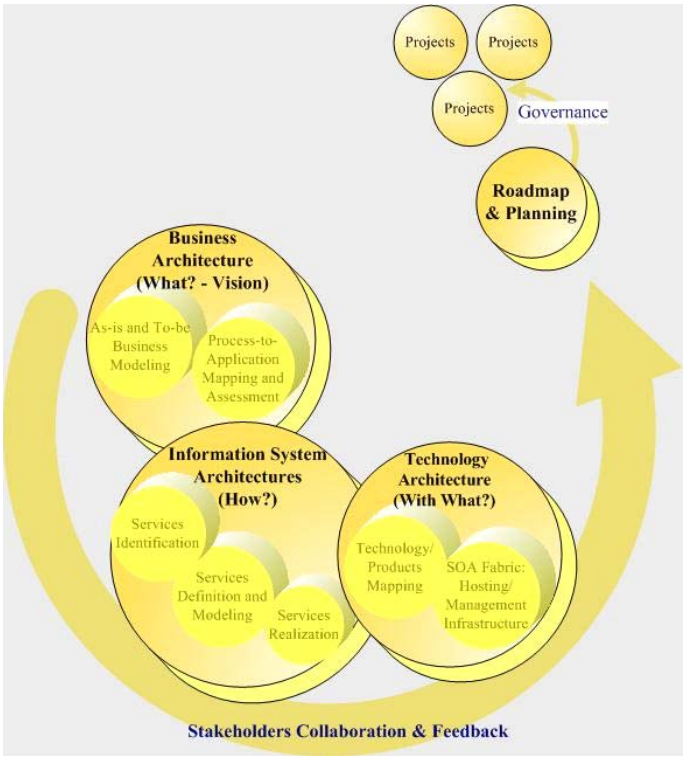
An Architectural framework (AF) provides a systematic approach to guide the design, evaluation and development of an Enterprise Architecture (EA). While there are a number of established AFs for defining an EA ([5],[7],[11]) The Open Group Architectural Framework (TOGAF) is one of the most widely adopted AF and in particular its *Architecture Development Method* (ADM)[6]. ADM defines a generic iterative process for developing an organization-specific architecture at the enterprise level as well as the individual system level. Service Oriented Modeling and Architecture (SOMA) is another methodology proposed by IBM [3]. TOGAF is used by 9% of enterprises and has been used as a guideline for the framework discussed in this paper [11].

SOAF comprises of four phases as depicted in Figure 1.

Business Architecture phase defines the business processes that participate in value creation for the bank and the customer. This focuses on the "As-Is" and "To-be" processes and the mapping of the business processes to applications. Active involvement of the business personnel is a critical requirement for this phase.

Information System Architecture takes over from the Business Architecture phase and focuses on Service Identification, definition, modeling and realization. In this phase, application owners (both the business and technology personnel) must jointly carve out the business services and formalize the realization of these services.

*Figure 1. Phases of SOAF*




Technology Architecture is focused on identifying the infrastructure required to implement the services identified. This may involve new development, modifications to the Legacy systems, developing API wrappers etc.

Roadmap & Planning is the classical program and project management phase. The successful SOA implementation program is critically dependent on the planning of the projects and ensuring the continued buy-in from the various stakeholders through the completion of the implementation.

## IMPLEMENTING THE SOAF

This section presents a detailed execution view of SOAF and discusses key activities and deliverables.

**Phase I - Business Architecture**

Migration to SOA of large industrial scale applications needs to be initiated by a top-down capture of key business process as well as the mapping of the business processes to the existing application portfolio.

*As-is and To-be Business Modeling*

The key outcome of this activity is to establish the scope, the constraints, and the drivers motivating the move to SOA. This is followed by defining the "As-is" and the "To-be" business process models (BPM) so that business services can be properly identified. BPM consists of the decomposition of the business domain into functional areas and business use cases. The "To-be" process models must focus on identifying activities or sub-processes that may be grouped as a shared service.

The level of functional decomposition of business processes depends on the complexity of the processes. For example a business process could be decomposed into sub-processes, which in turn could be further decomposed into high-level business use cases comprising a set of activities. For instance, the registration of a new customer is a business use case in a Sales Order process.

The key deliverables produced by this phase are:

- Revalidation of the business and technology drivers motivating the migration to SOA.
- Current business process model (As-is model)
- Evaluation of the "As-is" business process model to highlight key business issues and technical pain-points
- "To-be" business model that incorporates the required changes to business processes and facilitates easy service identification.
- Candidate business services from the "To-be" business process model
- Identifying, categorizing and prioritizing non functional requirements (NFRs) and Business Level Agreements (BLAs)

*Process-to-Applications Mapping and Assessment*

Process-to-Application Mapping (PAM) is an important activity wherein the existing software assets are examined to identify the application functionality for realizing business services. This results in identifying all the applications that support fulfillment of particular business process. The enterprise IT applications portfolio is then assessed based on the criteria given in the following:

- **Business value contribution** can be determined based on the application criticality in achieving business objectives and its ability to generate business returns in terms of financial benefits and improved customer satisfaction.
- **Functional reusability** indicates whether the business functionality in the application can be reused by other business processes.
- **Technical health** measures the system's reliability in delivering required functionality and its scalability in coping with additional workload.

- **Technical flexibility** indicates the level of system complexity, maintainability, and extensibility.
- **Cost of ownership** covers one-time and recurring costs with respect to hardware, maintenance and upgrades.

The key deliverables of this activity are:

- The PAM matrix linking the business Processes with IT systems.
- Gap analysis matrix capturing iterative gaps between the as-is state and the to-be state.
- List of software assets well-suited to be transformed to provide services.
- List of candidate services that can be mined from the applications portfolio
- List of existing programs and applications that take part in service realization

**Phase 2 - Information Systems Architectures**

*Services Identification and Matching*

The process modeling and the PAM Matrix should provide a) a set of business activities that can be grouped as services and b) the relevant applications in the IT portfolio that support those business activities. The task then is to identify the services from the business perspective and evaluate them for the technical feasibility of realizing those services. This requires a cross-functional team of business and technology owners of the application who are familiar with the problem context to draw up a list of services that can be realized. The top down identification of business services is matched by the bottom up identification of the realizable services.

Bottom-up service identification is an outcome of the PAM activity that examined the functionality resident across various software applications. These are typically fine-grained services such as: updating customer's personal information, or financial information, updating accounting entries for a cash payment transaction or loan transaction etc. These fine grained services must be consolidated in a coherent manner to come up with coarse grained services of appropriate granularity.

These services must then be matched up with the ones identified by the top-down process to come up with the detailed service portfolio for the bank. As a result of this matching process we will get a list of a) matched services, b) services which will require some modifications and c) new services that may need to be developed.

The service identification process requires domain analysis to identify reusable business functions as potential services. Some of the business services can be partially realized from those harvested from existing legacy systems. The harvesting can be facilitated by reverse-engineering techniques and tools to extract data flow graphs and control flow graphs from the legacy system.

*Services Realization*

This defines transformational strategies that will help transition to the future architecture through reuse, build, outsource and subscribe to commodity services. Based on the degree of services reuse, this phase also organizes services into appropriate vertical/horizontal layers using architectural patterns like the façade service pattern.

The realization approaches can be divided into two broad categories:

- Non-invasive service enablement is a tactical approach to align existing systems to business needs through wrapping by using new layers of flexible technologies such as EAI solutions, messaging tools, and recently with standardized interfaces using web services.
- Invasive transformation is a strategic approach that aims to revitalize and streamline the application portfolio to ease maintenance, extensions and interoperability. This can be realized through different strategies such as reengineering and redevelopment, refactoring, restructuring and componentization, or re-hosting. It often involves a deep and detailed analysis of the existing code base, understanding the system functionality and data architecture. Subsequently, it involves the extraction and rationalization of data definitions, data and business rules. This is followed by an iterative process that involves refactoring, consolidation, componentization and redesigning activities to make the code more modular and ease the service realization

The main challenge of the service realization phase is to select the optimal combination of service realization options. To address this problem, we have developed a decision framework, called Solution Options Analysis Process (SOAP). The framework guides architects in the systematic ranking of the most appropriate combination of services realization approaches for a given problem context, while taking into account the consequences on the desired quality attributes

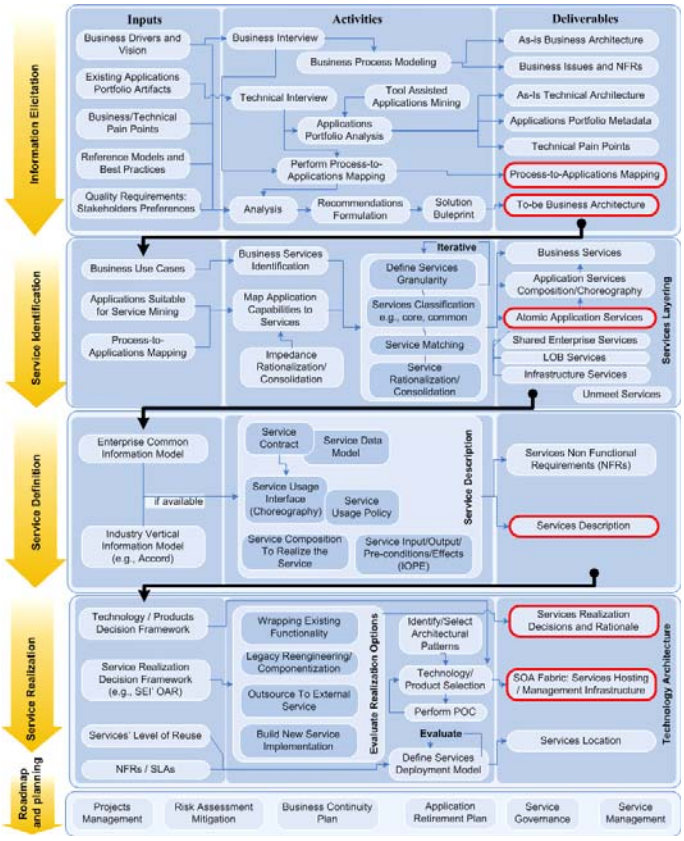
The process of Information elicitation, service identification and realization is presented in detail in Fig 2 - SOAF Execution view

**Technology Architecture**

In this phase, the enterprise architecture will be analyzed in the light of the business and technology drivers that have been identified. The various options for achieving SOA will be identified based on the existing infrastructure at the bank, such as the existence of an EAI platform or messaging middleware. Based on the architectural options, an appropriate option must be short listed in conjunction with the architects of the bank and identified as the model for future development. Thus, as a part of this sub phase, the following artifacts may be produced:

- Future state architectural model
- Technology shortlist
- Enterprise-wide standards to be adopted

*Figure 2. SOAF execution view*



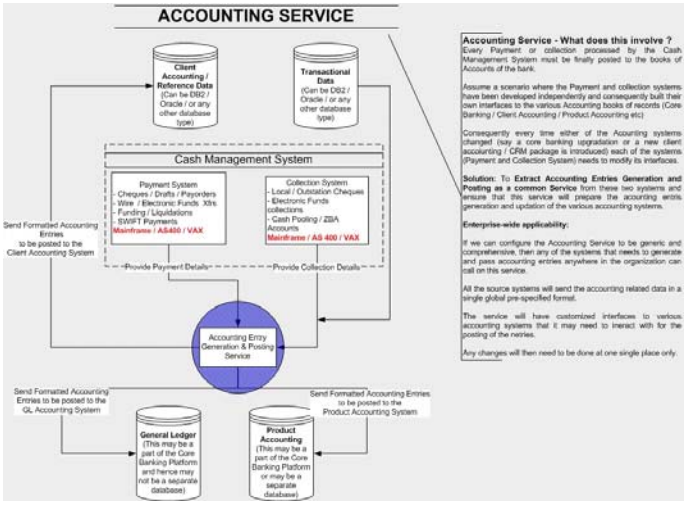
For example, in the cash management application being studied, it was decided to adopt web services as the means to achieve SOA. This was decided on account of the lack of existing middleware along with a need for synchronous response from the services.

**Roadmap & Planning**

The roadmap and planning phase involves detailed planning of projects to implement the transformation initiatives as well as identifying business and technical risks and defining mitigation strategies. The main goal is to reduce disruptions to the business continuity during the SOA rollout. The key artifacts of this phase are:

- Governance model specifying services ownership, change requests policy, auditing procedures, processes to manage and maintain services, etc.
- SOA rollout roadmap with a prioritized list of projects
- Resource requirements and availability estimates per project
- Risk assessment and mitigation plan
- Impact analysis per project with a plan to ensure business continuity during SOA rollout.
- Applications retirement plan

*Figure 4. Accounting service*




## APPLYING SOAF TO CASH MANAGEMENT

This section presents and discusses an application of SOAF to a case study from the banking domain. A typical cash management system with its components and their underlying technologies are shown in Fig 3.

Cash Management was chosen as a sample application for a variety of reasons, chief among them being

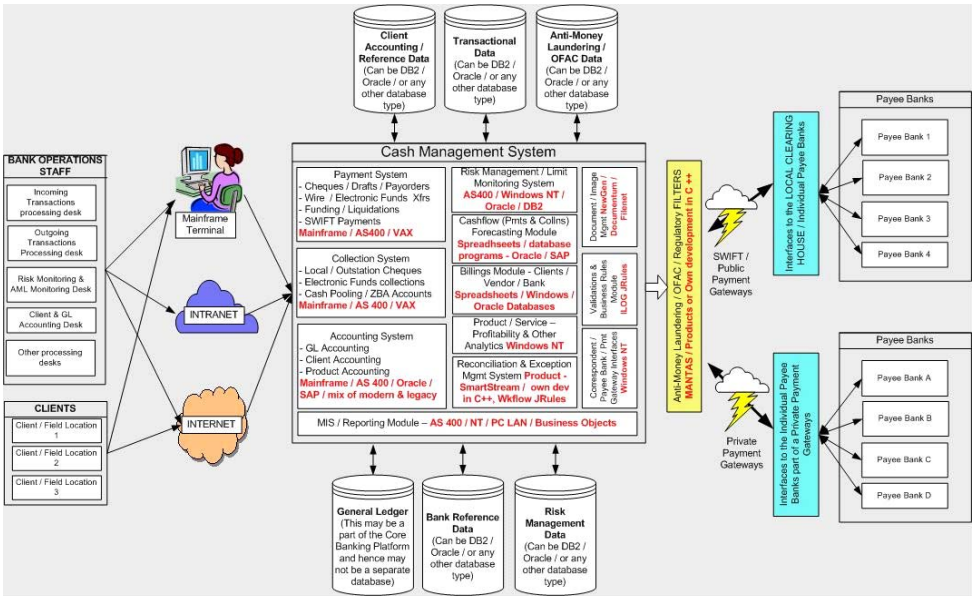
- Cash Management & Payments have traditionally attracted significant IT investments. Consequently they have an IT systems landscape that covers the entire range from the legacy systems to the modern web technologies.
- Cash Management and Payments are rich in business functionality that can find a high degree of re-use across the banking organization. For example Payment functionality is also present in departments such as Loans, Trade Finance, Client Services, Mortgages, Credit Cards etc. that have the need for making and receiving payments [9].
- Cash Management and Payment products are most eagerly sought by the banks' clients and hence any enhancements can help the bank demonstrate value to their clients.

SOAF was applied to this program context by performing a top down business process analysis. Subsequently, the "to be" business process was developed in conjunction with the business and technology stakeholders. The cash management system that was analyzed had the following functionality located on the legacy platform:

- Limit Checking functionality – this is performed by programs associated with the different business processes such as Payment processing, Loan processing trade finance etc.
- GL (General Ledger) Accounting
- Customer Information

The details of this functionality were identified by mining the legacy portfolio using tools such as ASG Recap. Based on this, the consolidated functionality points provided by the legacy platform were identified. To provide a specific example, it was found that the credit limit calculation functionality was spread out across multiple programs. However, based on the top down domain decomposition process, it was determined that the various businesses needed a common, consolidated check credit limit

*Figure 3. Cash management system*



service. Therefore, it was decided to consolidate the check credit limit functionality into one program and expose it as a service for all applications.

Additionally a cross-functional team of business and technology experts performed the PAM linking the business functionalities to the underlying IT applications. Based on these discussions and data mining, we identified a sample list of services:

- Create Payment – This service will facilitate payment creation through a web user interface, allow the client's ERP systems to directly call the service and load the payables data directly to the bank for making the payment. The data requirements for this service were detailed and the use case created.
- Client Validation – This service performs all the necessary Client validations that are performed by the bank. This will involve checking with the Client database for valid account numbers, valid products that can be availed, Anti-Money Laundering Checks etc.
- Check Limits – This service calls the central limit monitoring system with the details of the payment transaction to check whether adequate limits are available to undertake the payment operation.
- GetFXRate – This service will contact the Treasury systems to get the FX (Foreign Exchange) rates for performing the necessary conversions in making the payment.
- Generate Accounting – This service will be called by the Cash Management product systems (Payment System / Collection System) with the details of the accounting entries to be passed to the Core Banking application (GL accounts) and Client Accounting Applications.

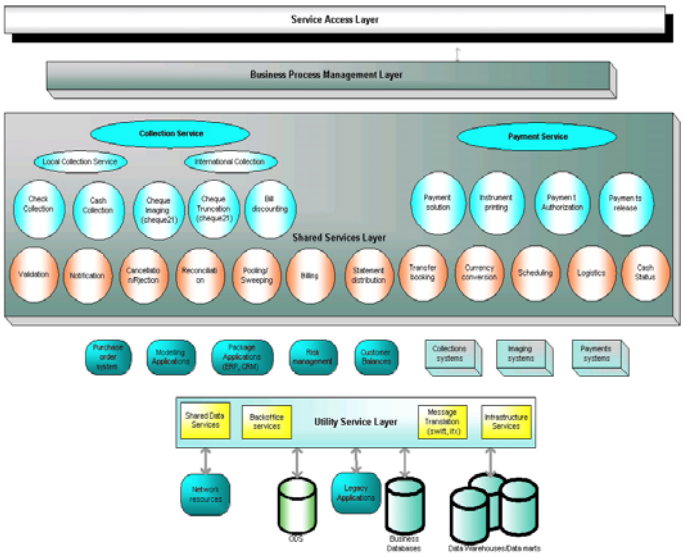
**Business Process based SOA implementation using BPEL**

Business processes associated with different products provided to the bank customers were defined. Coarse and fine-grained services were then designed to satisfy these business processes. These processes were encoded in BPEL using a BPEL engine such as Oracle BPEL Process Manager. The individual services were dynamically used by the business process code.

This is indicated by the Business Process Management Layer indicated below [10]. For each of the service identified, re-use was evaluated for each of the service by analyzing the business processes of other allied LOBs such as Trade Finance, Loans etc. For example, the Accounting entry posting functionality (GenerateAccounting service) was performed by different business processes such as Payment creation process, Loan application processing and trade finance etc. Each of these services were then exploded as a Use case detailing the activities they performed, the inputs they received and the outputs they generated and their interactions with the rest of the CMS components. A sample GenerateAccounting Service is shown in Fig 4.

From this the detailed code was developed for the service which was then registered with the Service registry. The process was repeated for each service identified (both business and infrastructure services) and the existing CMS architecture was transitioned to the desired state as shown in Fig 5.

*Figure 5. Shared services architecture for the cash management domain*




## DISCUSSION AND LESSONS LEARNED

While the SOA approach strongly reinforces well-established, software architecture principles such as encapsulation, modularization, and separation of concerns, it also adds additional dimensions such as service choreography, service repositories, and the service bus middleware. Our case study highlights the following:

- The key considerations in deciding the services to expose are delivering business value and traceability to the business model.
- The enterprise wide common information model (CIM) is important to support the consistent representation of key business entities and to reduce syntactic and semantic mapping between services.
- Need for services metadata management repository to support SOA governance and evolution and to foster service reuse.
- Need for governance model to be followed across the organization. The model needs to be evangelized and audited formally to ensure that the projects architecture adheres to enterprise and Line of Business (LOB) architecture blueprints. The goal is to limit architectural and technological divergence.
- Need to maintain strong sponsorship from all stakeholders and buy-in from decision makers through regular communication. A SOA creates cultural as well as enterprise-wide technological change, which can create barriers; therefore, communication is critical.

## CONCLUSION AND FUTURE WORK

SOA is gaining momentum as an approach to deliver maximum reusability, flexibility and responsiveness to change. However, practical implementation of these architectures requires careful planning, a comprehensive guiding framework and sound engineering principles. The main contribution of this paper is a systematic and repeatable SOA framework, called SOAF, grounded in our past experiences with SOA enablement projects and assignments. This framework has proven to be effective in optimizing the time and effort and raising the quality through well defined processes combining top-down context comprehension with transformations in a bottom-up fashion.

## REFERENCES

1. Celent Communications, "A North American Perspective: IT Spending at US and Canadian Banks" Jan 2005
2. Alonso, G., Casati, F. et al. 2003, Web Services: Concepts, Architectures, and Applications, Springer, Berlin.
3. Arsanjani, A. 2004, Service-oriented modeling and architecture (SOMA). http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/
4. Crawford, C. H., Bate, G. P., Cherbakov, L., Holley, K. and Tsocanos, C. 2005, 'Toward an on demand service-oriented architecture', IBM Systems Journal, vol. 44, no. 1, pp. 81-107.
5. The Institute For Enterprise Architecture Developments (IFEAD) 2004, A Comparative Survey of Enterprise Architecture Frameworks. http://www.enterprise-architecture.info/Images/Documents/Comparative_Survey_of_EA_Frameworks.pps [29 August 2005].
6. The Open Group 2003, TOGAF 8.1 Enterprise Edition. http://www.opengroup.org/architecture/togaf8-doc/arch/

7. Zachman, J. A. 1987, 'A framework for Information Architecture', IBM Systems Journal, vol. 26, no. 3, pp. 276-292
8. Dhulipala Ramesh, Sreekumar S, Venkataraman Jayakumar, "Integrating Information in Cash, Trade & Treasury – How banks can respond to the Challenges", HSBC Guide to Cash & Treasury Management in Asia Pacific 2005
9. Venkataraman Jayakumar, Khurana Meenu, Anand, Dr. Sriram, Padmanabhuni, Dr. Srinivas, "Service Orientation in Cash Management", HSBC Guide to Cash & Treasury Management in Asia Pacific 2006
10. Venkataraman, Jayakumar, Anand, Dr. Sriram, "Future Proof your Bank's IT Architecture", HSBC Guide to Cash & Treasury Management in Asia Pacific 2006
11. Zachman, J. A., "A framework for Information Architecture", IBM Systems Journal, vol. 26, no. 3, pp 276-292