Global Journal of Computer Science and Technology
Volume 11 Issue 10 Version 1.0 May 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
ISSN: 0975-4172 & Print ISSN: 0975-4350

An approach for exposing legacy applications for integration in modern enterprise business processes

By Manish Verma, Prof. Vijay Shah

Abstract- Organizations have legacy software applications and systems, which are valuable assets and still critical to their current business operations. One of the key challenges for today's enterprises is how to effectively and efficiently integrate these legacy assets with other. internal and external applications and business processes. Modernizing legacy systems using SOA may help an organization to meet the increasing demands of the business processes integration while maintaining its large investment in legacy systems. SOA bridges the gap between business processes and technology resulting in improved reuse of existing applications and interoperability of desperate applications. Sometimes it is required to integrate legacy system with other application in a short span of time. As per the business requirements although SOA may address the solution, but it is very difficult to integrate due to several issues in service identification and mapping in legacy application and business processes.

In this paper an approach is proposed for participation and alignment of legacy applications in the enterprise business processes in a short span of time with the help of business process modeling and SOA, incorporating both Top down and bottom up approach together. As a case study, the paper describes the exposure of a prototype legacy banking application for participation in a service-oriented architecture.

The paper also proposes a novel way of determining the key business logic and resources embedded in a legacy C/C++ application with a reverse engineering approach so that the SOA mapping is automated and the constraints like memory and processing can be analyzed.

Keywords: Legacy Applications, Web services, Integration, Service oriented architecture, business process modeling.

GJCST Classification: H.2.4, H.3.4, C.2.1

An approach for exposing legacy applications for integration in modern enterprise business processes

Manish Verma[α], Prof. Vijay Shah[Ω]

Abstract - Organizations have legacy software applications still critical to their current business operations. One of the key challenges for today's enterprises is how to effectively and efficiently integrate these legacy assets with other Internal and external applications and business processes to address the changing business needs. Modernizing legacy systems using SOA may help an organization to meet the increasing demands of the business processes integration while maintaining its large investment in legacy systems. Sometimes it is required to integrate legacy system with other application in a short span of time. As per the business requirements although SOA may address the solution, but it is very difficult to integrate due to several issues in service identification and mapping in legacy application and business processes.

In this paper an approach is proposed for participation and alignment of legacy applications in the enterprise business processes in a short span of time with the help of business process modeling and SOA, incorporating both Top down and bottom up approach together. As a case study, the paper describes the exposure of a prototype legacy banking application for participation in a service-oriented architecture. The paper also proposes a novel way of determining the key business logic and resources embedded in a legacy C/C++ application with a reverse engineering approach so that the SOA mapping is automated and the constraints like memory and processing can be analyzed.

Keywords— Legacy Applications, Web services, Integration, Service oriented architecture, business process modelling.

I. Introduction

Business organizations have a large investment in software written in legacy programming languages and architecture. Today, most of the organizations run systems that have been developed a long time ago and are still adapted and maintained to meet current needs [1]. Approaches used in legacy systems sometimes differ considerably from today's modern paradigms. Adapting legacy software systems to new requirements also requires their migration and integration with new technologies and applications.

A Legacy Information System can be defined as "any information system that significantly resists modification and evolution" [1]. These Information Systems pose considerable problems including brittleness, inflexibility, isolation, non-extensibility, lack of openness on the enterprise IT infrastructure. They significantly resist further modification and evolution. Legacy systems were designed for internal, enterprise wide usage while today's enterprise demands for inter enterprise communications and support for agile business processes.

There are various methods proposed including supplementing or replacing restrictive legacy applications and technology with newer, open standards-based applications and technologies, but still retaining existing business functions [2]. Legacy renovation is an expensive task. As the requests for modifications, new development and integration accumulate, it is very difficult to keep pace and enable a more agile enterprise. So it is advantageous to integrate legacy application with new applications while retaining the business logic intact.

SOA models the enterprise business workflow as a collection of business services, these services can be easily Reused and integrated in the organization to address the application requirements [3] [4]. Rather than developing from the ground up every time a change is needed, new assets can be created easily from existing services, saving time and money [5]. The most important advantage of SOA is to provide a data bridge between incompatible and desperate technologies [6].

BPM is a set of technologies and standards for the design, execution, and administration and monitoring of complex, automated business processes BPM provides the facility to monitor, analyze, control and improve the execution of these processes in real time [7]. Existing Web Services can be integrated as the counterparts of steps in business processes. Examples are credit verification in an on-line order process [8].

Several approaches for Legacy application integration and migration towards Service oriented architecture have been proposed in the past but they

About[α]- Research Scholar, Samrat Ashok Technological Institute Vidisha, India.
E-mail- Manish.rc.verma@gmail.com
About[Ω]- Reader, Department of Computer Science Samrat Ashok Technological Institute Vidisha, India.
E-mail- vijayvidisha@gmail.com

include difficulties in identifying candidate services from legacy application and their alignment with the organizational business processes requirements. Wrapping functions of legacy systems for exposing as web services may be a fast way for migration towards SOA, However the services provided are technology-oriented and do not fully address business goal or scenario, which might lead to the services not being fully utilized as per the business goals. *The top-down* approaches analyze existing business processes in order to identify adequate services but services are business oriented and may failed to get the needed functionality in legacy applications.

Our proposed method uses BPM and SOA together in order to make legacy participation in the enterprise business processes by means of exposing and orchestrating services. We used both top- down BPM driven and bottom up SOA driven service identification mechanism along with automated discovery of business logic in legacy codes to better align legacy application with business needs.

The reminder of the paper is organized as follows: Section 2 provides background information needed to understand the paper, Section 3 presents the proposed approach for legacy service exposure and participation in enterprise business processes with a case scenario, section 4 describes Discussion of result and historical approaches for legacy system migration and integration. We conclude with summary and future directions in section 5.

II. Background Information

An organization has a collection of interrelated business processes, which consist of a series of activities, jointly represent a business goal. For instance an order process in a B2B environment includes several activities networked with each other to complete the order process business goal. These business processes may have their span in several desperate applications in the enterprise and beyond.

The ''service'' is an application logic or underlying computing resource with an abstraction of the underlying implementation [3], that is exposed through an interface and which can be invoked over a network with standard protocols [9]. The most important application of SOA is to align the services to the business requirements and to facilitate reusability of existing services connecting the various operational systems that automate an enterprise's business processes [10].

Bottom-up approaches examine legacy systems and standard software for the identification of services. The bottom-up approach analyses the existing application for all available functions that can be exposed as a web service. With a bottom-up approach the definition of services originates in the domain of the application [11]. This approach results in a large number of available functions which can be exposed as services but does not tell anything about their usability. In bottom-up approach Services are discovered, rather than defined as requirements, therefore it tends to lead to poor business services as it preserves the current solution instead of addressing business needs.

The top-down approaches analyze existing business processes in order to identify adequate services. This approach looks at the processes in the business domain, decomposes them into sub process/tasks/functions until 'logical units of work' are defined, which we call services. The decomposition of the process in itself is not enough to discover services, because we will have to convert the 'candidate-service' into a real-world executable and discoverable service. This means we have to compare the functionality of the service with the available infrastructure but it does not reach deep into the architecture and implementation domain.

By combining these two approaches together we may get the best of both eliminating the shortcomings of each one and integration with the current application infrastructure and readiness for the business needs we are looking for. Therefore, a hybrid approach combining both methods is applied in order to minimize the problems associated with these approaches [12][13].The hybrid meet-in-the middle approach combines top-down and bottom-up techniques in order to identify services that better correspond to both the business requirements and the existing systems [14]. It provides a compromise between the top-down and bottom-up techniques that eliminates the most significant disadvantages. We proposed a meet in the middle approach with automatic discovery of prospective services in the legacy application for integration in business processes.

III. Proposed Approach

The proposed approach is a combined meet in the middle approach consists of BPM and SOA along with the reverse engineering of legacy application in order to identify the candidate services and their participation in the enterprise business processes. It eliminates the disadvantages of failure in getting the needed functionality in legacy applications and lengthy time of top down approach and non-business orientation of bottom up approach. Basically, we are proposing consolidated approach that provides guidance on the derivation of both business services and supporting software services to achieve a close business and IT alignment. The proposed method is influenced by meet in the middle approach by Inaganti and Behera with some improvements by minimizing the problem in finding of candidate service in legacy application [15].

This approach requires a comprehensive analysis of the legacy application and not just the mapping .In this paper we also presented a reverse engineering approach to extract the information of the legacy application through resource extraction from the application along to generate and publish the service.

The proposed technique can be summarized as a methodology for mapping legacy application resources to modern enterprise environment by BPM and SOA with the help of Reverse Engineering to facilitate the identification and extraction of the key business logic from Legacy application and alignment with business requirements. Figure 1 shows the block diagram of proposed method.

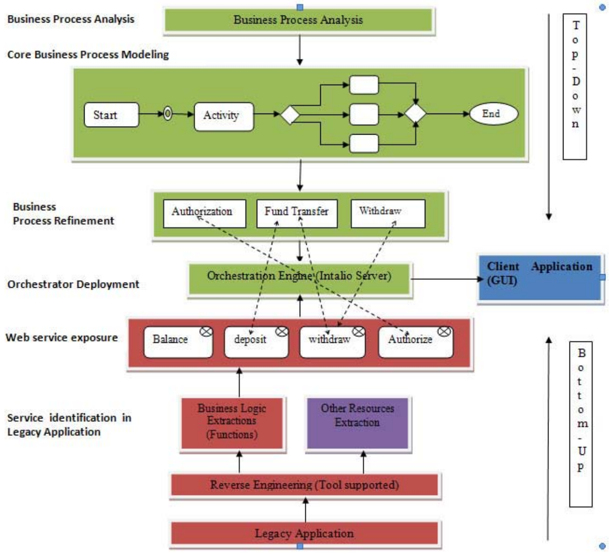

Business Process Analysis
Business Process Analysis
Core Business Process Modeling
Start
Activity
End
Top-Down
Business Process Refinement
Authorization
Fund Transfer
Withdraw
Orchestrator Deployment
Orchestration Engine (Intalio Server)
Client Application (GUI)
Web service exposure
Balance
deposit
withdraw
Authorize
Service identification in Legacy Application
Business Logic Extractions (Functions)
Other Resources Extraction
Reverse Engineering (Tool supported)
Legacy Application
Bottom-UP

Fig. 1 Block diagram of proposed method

a) Problem Scenario

As an example scenario, here we introduced a prototype legacy banking application developed in C++. The Application is a monolithic standalone C++ application with no external interface for communication with other applications. The application stores the data in flat file system and provides various inbuilt functionalities like opening an account, deposit money, withdraw money, get balance, viewing of transactions and various reports. One can interact with it only through application GUI which is inbuilt in the code.

We assume that there are certain situations arises due to business needs, in which above application is required to be integrate with some other modern applications and business processes. As stated earlier in the paper, it is not viable to replace or rewrite the legacy application with new code. As the legacy application is standalone and the logic is embedded deep in the form of functions, it is very difficult to handle this situation. Our approach resolves the problem without rewriting the legacy code and without interrupting the operation or maintenance of the legacy system.

We presented a business process of ATM here. The ATM process resembles the new business requirement and we have to address it by using core business logic of legacy application. The ATM application will be a thin client with only GUI. The ATM GUI will provide only the user interface and the orchestrator takes care of proper sequencing of operations and calling of web services from legacy application. Various new business conditions and constraints, which are not present in legacy application may be addressed with the orchestrator, for example prohibiting withdraw operation in case of insufficient balance or not allowing a certain transaction if the number of transaction exceeds, without rewriting of the core component of the legacy application. The methodology is mentioned in following steps.

b) Top down Business process Analysis and identification of services in the Business domain

i. *Analyze the Core Business process.*

In the first step, it is necessary to analyze functional domain of business process and understand the business requirements. During the initial analysis phase, basic decisions regarding identifying the most important functions of the domain and showing their interrelationships is identified. The business structure and the functions of the business process are specified [16]. A business process reveals how an individual process activity is linked with another in order to achieve a business objective. We have to identify the start and end events of a particular business activity, human interactions with the process, automated activities and links with the other applications and organization's processes. This step results in a core coarse grained business process. This analysis results in the development of the "to-be" process model that an SOA solution is intended to implement [8].

In the context of our scenario we analyzes how a user will interact with ATM, how the business process will progress with the user interaction. This step results in the use case of ATM process, actors, activities and the flow of business.

ii. *Model the core business processes*

In this step we design the core business process found in previous step into a graphical model using a BPM modeler, as in our case Intalio [7]. At first, the business process is modeled at a high level where activities in the Business Process Diagram usually aggregate sub processes, which are graphically evaluated in another sub process BPM diagram. A [+] sign inside an activity denotes a process that can be decomposed into sub-processes [17]. We marked each task which requests an external service as service task. One of the purposes of the process model is to define process execution and to create the BPMN notation for this purpose.

The BPMN defines both the graphical notation and the semantics of a Business Process Diagram, We model the diagram with core elements of a BPD which comprises Flow Activities, Tasks represented by a rounded-corner rectangle which is a generic term for work that has to be performed, *Events* which are diagrammed as a circle, *Gateway Objects* represented by a *diamond* symbol which are used to control the divergence and convergence of a sequential flow. By means of these basic Flow Objects. The core process contains at least one *POOL* which is executable and can be deployed on orchestration Engine.

The core business process acts as an orchestrator which will direct and invokes the various Web services from legacy and other third party applications to form and execute a complete business activity [18].

iii. *Business Process refinement*

In this step we perform decomposition or refinement of sub processes of core BPM Diagram into elementary or atomic business functions. Business process refinement identifies service tasks, message activities, and events and actions in the sub process [11]. When the business process is subdivided into sub-processes or decomposed into granular activities the lowest level tasks will consist of small, cohesive "logical units of work", that are supported by the functionality offered by distinct services. It results in discovery of elementary business functions which can be addressed by web services in the application domain [19]. Delegating functionality into sub business processes prevents business processes from becoming overly large, complex and difficult to maintain. The refinement can be seen in authorization process diagram.

After refinement of sub processes the next step is to find out the Service Tasks which provide some sort of service which could be calling a Web service or an automated application. We will find the task which will call the legacy and other web services process. This service tasks will call atomic (single) web service interfaces of legacy application. In the case of authorization sub process the call authorize task invokes the authorize function of legacy application as shown in yellow color in figure 2.

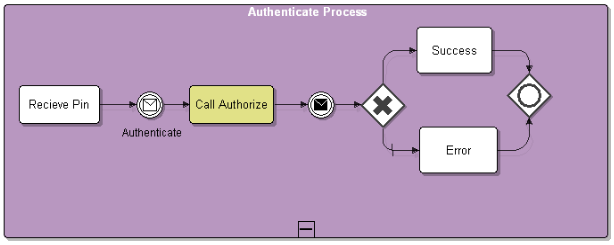


Fig. 2 Authorization sub process

c) *Existing asset analysis and Service identification in Legacy Application*

The purpose of this step is to identify assets in the legacy application i.e. business logic and legacy functionality which are capable of supporting the realization of services that meet business needs. The focus is on existing assets that play a role in business processes 14]. We initially perform a coarse-grained mapping of business processes and process activities to the portfolio of existing applications and application interfaces. Later in the specification, we perform a more fine-grained mapping, including message specification. We also identify various system resources so that the performance can be measured.

i. *Component identification in legacy application*

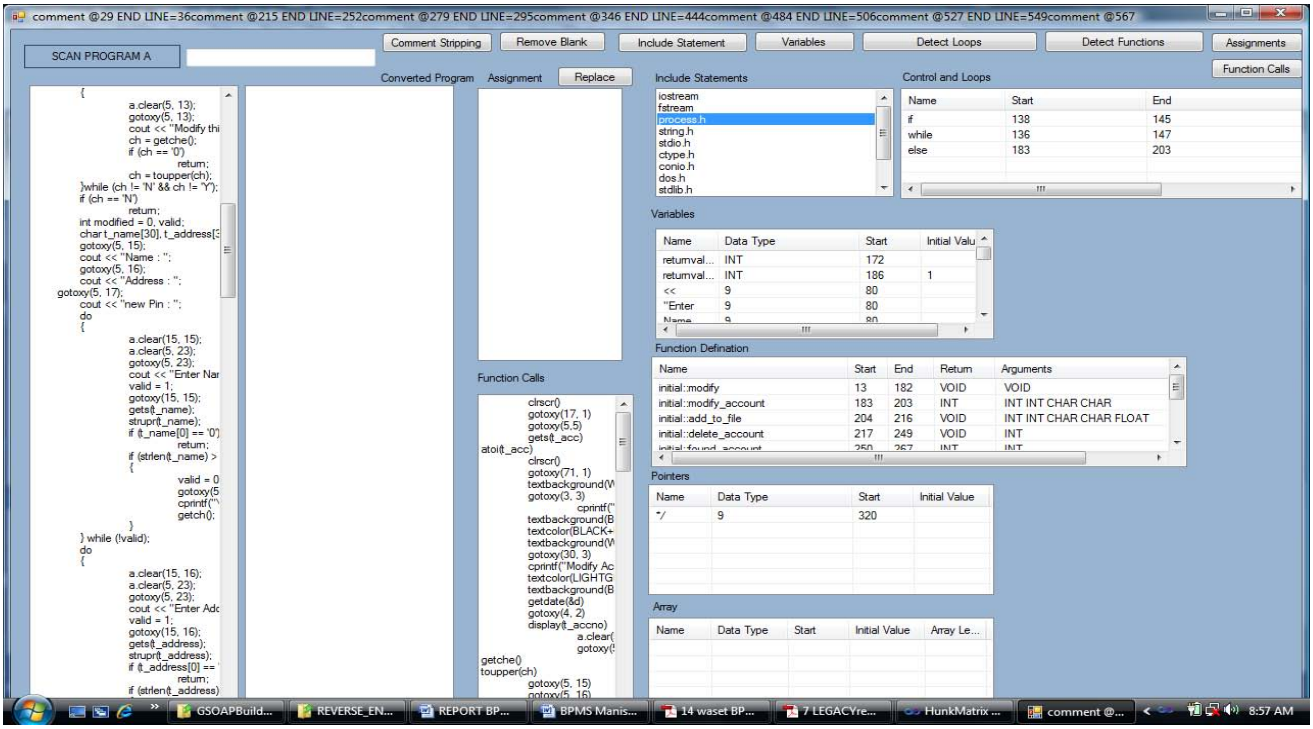
Service identification On the one hand, it is to determine what business functionalities should be provided by the target service in the application domain. On the other hand, the business functionalities embedded in the legacy system should be identified in order to be reused in the target service [20]. In this step the legacy application will be reverse engineered and we will find the various resources and embedded business logic to be exposed as web services by adopting bottom up approach. The analysis reveals the reuse potential of existing legacy components. The results of this step include information on system architecture, system components, infrastructure details and interfaces.

The proposed work identifies that this is a tedious process and the complexity grows with the line of code that reflects the business logic. Therefore a summary of the business logic present in the code is of extreme important. Here it is performed by a automatic reverse engineering tool. The working description of the tool is as given below.

First the tool Scans the program and tokenizes the program. Thereafter it removes the comments and blank sections. It Extracts the function definitions and other resources using following logic:-

1. Check if there is a “(“ token in the line or not If so, extract the previous token (which can be function name or a predefined function or key words like if, for and so on)
2. Check if the token is a key word or not. If not, scan till the end and look for a “;” if it is not there, scan for another token appeared before the current token with space separation.
3. If the previous token is a key word (which reflects the return type) then marks the current line as the function definition. Extract the arguments and return types.

4. Generate a summary of the function definition such that it can be used for header file to generate GSOAP web service interface.
5. Extract the variables like arrays, pointers, loops to understand the resources used by the application. The output is expressed in figure 3 below.



comment @29 END LINE=36comment @215 END LINE=252comment @279 END LINE=295comment @346 END LINE=444comment @484 END LINE=506comment @527 END LINE=549comment @567

```
        {
                a.clear(5, 13);
                gotoxy(5, 13);
                cout << "Modify thi
                ch = getche();
                if (ch == '0')
                        return;
                ch = toupper(ch);
        }while (ch != 'N' && ch != 'Y');
        if (ch == 'N')
                return;
        int modified = 0, valid;
        char t_name[30], t_address[3
        gotoxy(5, 15);
        cout << "Name : ";
        gotoxy(5, 16);
        cout << "Address : ";
    gotoxy(5, 17);
        cout << "new Pin : ";
        do
        {
                a.clear(15, 15);
                a.clear(5, 23);
                gotoxy(5, 23);
                cout << "Enter Nar
                valid = 1;
                gotoxy(15, 15);
                gets(t_name);
                strupr(t_name);
                if (t_name[0] == '0')
                        return;
                if (strlen(t_name) >
                {
                        valid = 0
                        gotoxy(5
                        cprintf("\
                        getch();
                }
        } while (!valid);
        do
        {
                a.clear(15, 16);
                a.clear(5, 23);
                gotoxy(5, 23);
                cout << "Enter Adc
                valid = 1;
                gotoxy(15, 16);
                gets(t_address);
                strupr(t_address);
                if (t_address[0] == '
                        return;
                if (strlen(t_address)
```

Converted Program

Assignment

Function Calls

```
                clrscr()
                gotoxy(17, 1)
                gotoxy(5,5)
                gets(t_acc)
atoi(t_acc)
                clrscr()
                gotoxy(71, 1)
                textbackground(W
                gotoxy(3, 3)
                        cprintf("
                textbackground(B
                textcolor(BLACK+
                textbackground(W
                gotoxy(30, 3)
                cprintf("Modify Ac
                textcolor(LIGHTG
                textbackground(B
                getdate(&d)
                gotoxy(4, 2)
                display(t_accno)
                        a.clear(
                        gotoxy(!
getche()
toupper(ch)
                gotoxy(5, 15)
                gotoxy(5, 16)
```

Include Statements

iostream
fstream
process.h
string.h
stdio.h
ctype.h
conio.h
dos.h
stdlib.h

Control and Loops

Name	Start	End
if	138	145
while	136	147
else	183	203

Variables

Name	Data Type	Start	Initial Valu
returnval...	INT	172	
returnval...	INT	186	1
<<	9	80	
"Enter	9	80	
Name	9	80	

Function Defination

Name	Start	End	Return	Arguments
initial::modify	13	182	VOID	VOID
initial::modify_account	183	203	INT	INT INT CHAR CHAR
initial::add_to_file	204	216	VOID	INT INT CHAR CHAR FLOAT
initial::delete_account	217	249	VOID	INT
initial::found_account	250	267	INT	INT

Pointers

Name	Data Type	Start	Initial Value
*/	9	320	

Array

Name	Data Type	Start	Initial Value	Array Le...

Fig. 3 Reverse Engineering Tool

ii. *Identify the point of functionality in legacy application*

In the previous step we found the information about application resources and various functions present in the Legacy application through reverse engineering. Now we identify the points of functionality of the legacy application that may be useful to implement the needed web services to interact with service task in the Orchestrator [15]. In the output of automated tool we find the independent functions calls so that it can be used to designate or make as a point of functionality, it means, among several functions we found the one that contains the functionality of interest for service tasks found in top down approach. For example *found_account function* may be used to make authorize function wrapper that can be used as point of functionality and can be exposed as web service and mapped with service task.

iii. *Exposure of identified functions as web service.*

The functions identified in previous step as the point of functionality are now exposed as web services [19]. The Service interface defines the types of messages and the message exchange patterns that are involved in interacting with the service, together with any conditions implied by those message.. The definition of functions which has to be called by service task has been written in a single header file *test.h*. The header file guides the GSOAP tool to generate stubs for needed web services. Now we write the web service module by using identified functions of previous step and with the help of GSOAP supporting files. All the exposed web services can be called through the WSDL Interface and the Service end point URL [20]. WSDL describes the operations that a service can support, and the parameters each operation accepts and returns. End point URL serves as a service provider. The application web service will provide the response to service task upon calling with appropriate parameters. Figure 4 describes the wsdl portion for input and output parameter for Authorize operation.

```
<!-- operation request element -->
  <element name="authorize">
   <complexType>
    <sequence>
     <element name="acc-no" type="xsd:int" minOccurs="1" maxOccurs="1"/>
     <element name="acc-pin" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    </sequence>
   </complexType>
  </element>
  <!-- operation response element -->
  <element name="authorizeResponse">
   <complexType>
    <sequence>
     <element name="result" type="xsd:int" minOccurs="0" maxOccurs="1"
nillable="true"/>
    </sequence>
   </complexType>
  </element>
  <!-- operation request element -->
```

Fig. 4 part of a wsdl for legacy application web service interface *d) Service matching and mapping*

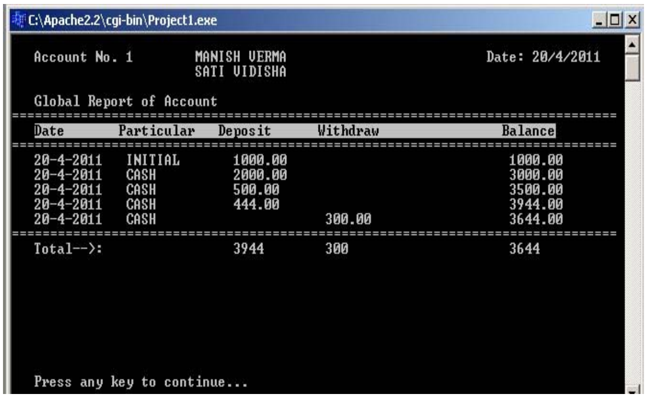
The service matching activity maps the service task to corresponding atomic web service in the application layer. Each atomic service task exactly maps single web service [15]. There may be some situation in which a service task will be divided in more than one service task and it may comprised of more than one point of functionality. We will map the service accordingly, for example the fund transfer service task will be associated to two services i.e. deposit and withdraw. Several service task may be associated to same service, for example a withdraw service may be associated to both withdraw and fund transfer service tasks. These concepts can be seen in Fig 1 and Fig 6.

At the end of this step we complete the matching and mapping of all the service task and corresponding web services with each other.

e) Service realization and orchestration

All the web services exposed from legacy system were atomic and asynchronous, that means there is no such execution or sequence order is set by default. In a business process there are several operating conditions upon which a business activity relies. For example a withdraw service can only be happened after authentication service, it must not happened before. Assembling services in order to accomplish business logic and business processes is called orchestration. [3] Explains orchestration as "the implementation of a business related workflow owned by a single entity through the combination of business relevant services". The orchestrator is deployed as a BPEL program in the Orchestration director engine.

In this step we define orchestration logic as the business logic that sequences, coordinates and manages conversations among Web services. This entire thing is expressed in BPEL. To execute a BPEL process a BPEL engine is needed which parses the BPEL code and executes the contained instructions. The Core Web services that are used by the BPEL program have to be deployed in advance. The whole orchestrator is deployed on Intalio BPMS server which acts as an Orchestration director engine. The orchestrator may be called in presentation mechanism through its WSDL and endpoint URL. In the ATM application a Java GUI reforms the presentation task which uses the orchestrator.



```
Account No. 1          MANISH VERMA                          Date: 20/4/2011
                       SATI VIDISHA

Global Report of Account
==============================================================================
Date         Particular   Deposit       Withdraw                 Balance
==============================================================================
20-4-2011    INITIAL       1000.00                                1000.00
20-4-2011    CASH          2000.00                                3000.00
20-4-2011    CASH          500.00                                 3500.00
20-4-2011    CASH          444.00                                 3944.00
20-4-2011    CASH                       300.00                    3644.00
==============================================================================
Total-->):                 3944         300                       3644




Press any key to continue...
```

Fig. 5 Legacy application account statement

IV. Discussions and Related Work

The implementation and results of the proposed method may be understood by following demonstration of the case scenario. In figure 5 an account summary page is displayed. We can understand that all the logic is embedded deep in the code of legacy application.

In figure 6 the composite fund transfer service task in the orchestrator is elaborated. New business process requires a fund transfer mechanism, which needs to be achieved from the existing code. This is done in the modeler by calling and sequencing two web services from legacy application thus making a composite process.

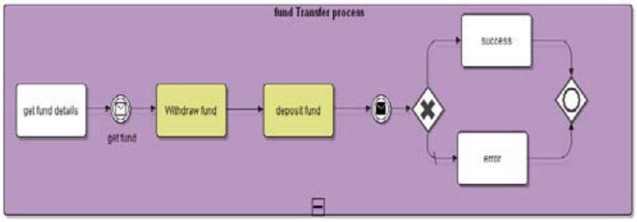


Fig.6 Composite service for fund transfer

In the sample banking application no individual authentication is required as the software interacts only through its GUI and it is accessed by bank authorities. But as per the requirement of ATM business process as the application is exposed to an external world then it is very important to put an authentication mechanism. This is incorporated in the orchestrator as shown in figure 7.

The results demonstrate the process proposed in this work with a sample banking application as elaborated in the case description. The system can map legacy application at par with the requirement of business using exposure of functionality through web services and the usability by orchestration of BPM process. Further the released services can be used by any language or application like Java and C# through the service description. This also proves that by adopting the proposed technique one would be able to integrate new logics with the existing business logic or would be able to seamlessly upgrade the existing services without having to make any changes in the coding in the legacy application. Therefore the architecture minimizes the maintenance cost by maintaining the business logic in the middle layer independent of the implementation.

In the past several methods for legacy system modernization have been proposed and discussed. [21] elaborated the white-box modernization and black-box modernization approaches comparisons.

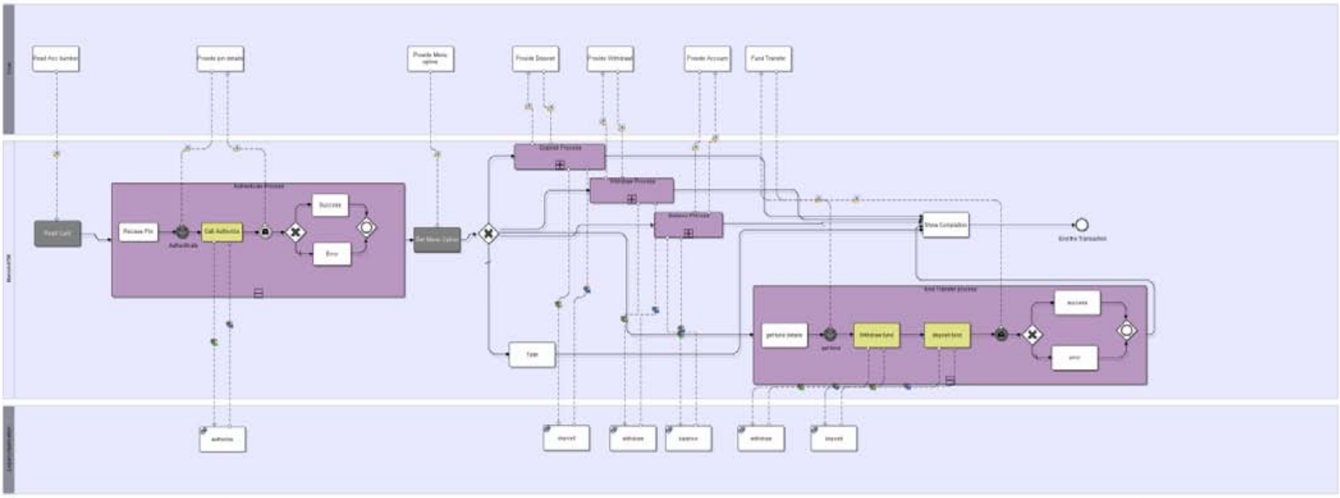


Fig. 7 Intalio ATM orchestrator process with interfaces

[22] Proposed a method to address the issues on migrating legacy systems into a Web enabled environment by using CORBA. [23] Used reverse engineering techniques to encapsulate the logic with JNI wrapper. [24] Proposed a method to make interactive legacy systems accessible by wrapping interactive functionalities. [25] Proposed a tool based approach to wrap the legacy codes. All of the methods were focused on the application modernization only rather than addressing the business domain. In our approach we emphasized on a close business and IT alignment.

V. Conclusion

In this work a legacy application mapping technique by combining top-down approach with bottom-up approach is proposed using SOA and BPM. The method defines how a legacy application should be exposed as services and how these services can be used to fulfill business goals. The solutions principles are demonstrated through the results. Instead of using the manual functionality discovery in the legacy service identification stage, a tool is developed to automatically identifying the services. The interoperability of the entire solution is shown in the result section. The prime criteria and objective of the work has been to prove the proposed design of legacy application mapping and integration to participate in enterprise business processes. As the services are exposed to be used on network, performance calibration and tuning is another aspect and the speed and processing capabilities of the environments used must be taken into consideration. The proposed approach performance and speed issues are some of the work that can be associated with the future direction of research for this work.

References Références Referencias

1. J. Bisbal, D. Lawless, B. Wu, and J. Grimson "Legacy information systems: issues and directions", *IEEE Software*, Vol. 16 No. 5, pp. 103-111, Sept/Oct 1999.
2. "The Modernization Imperative", Oracle Magazine, http://www.oracle.com/technetwork/issue-archive/ma10-advertorial-v1-134269.pdf
3. Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River, NJ, USA.
4. *A S Rao, "*Go for SOA with open source" , pp. 18-21, Linux for you December 2008
5. Mike Oara , "Finding Services in the Mainframe", SOA Magazine Issue VII, May 2007
6. J. *Chatarji*, "Introduction to Service Oriented Architecture. (*SOA*)". http://www.devshed.com/c/a/Web-Services/Introduction-to-Service-Oriented-Architecture-SOA/

7. T. Mitra. “Business-driven development”. IBM developerWorks article, http://www.ibm.com/developerworks/webservices/library/ws-bdd, IBM, 2005.
8. Koehler J., Hauser R., Kuster J., Ryndina K., Vanhatalo J., Wahler M. , “The Role of Visual Modeling and Model Transformations in Business-driven Development”, *Electronic Notes in Theoretical Computer Science*, Vol. 211 (28 April 2008), pp. 5-15
9. Papazoglou, M., Georgakopoulos, D. “*Service-Oriented Computing”*. Communications of the ACM. Vol46, No. 10, 2003.
10. E Eric Newcomer, Greg Lomow, “Understanding SOA with Web Services”. Addison-Wesley, 2005
11. Papazoglou, M.P. and van den Heuvel, W.J., “Service-Oriented Design and Development Methodology.” Int.J. of Web Engineering and Technology (IJWET), 2006.
12. Ali Asranjani, “Service-oriented modeling and architecture, How to identify, specify, and realize services for your SOA”. IBM Developerworks. November 2004. www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/
13. Gandhi S., “Fastening the SOA/SOMA Service identification approaches with project environments” , IBM Developerworks. March 2010
14. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., & Holley, K. (2008). “SOMA: a method for developing service-oriented solutions”. IBM Systems Journal,2008, 47(3), 377-396.
15. Inaganti, S., Behara, G. “*Service Identification: BPM and SOA handshake”* BPTrends, 2007.
16. O. Zimmermann, N. Schlimm, G. Waller, and M. Pestel, "Analysis and Design Techniques for Service-Oriented Development and Integration", in Proc. GI Jahrestagung (2), 2005, pp.606-611.
17. Emig, C., Weisser, J., Abeck, S. “Development of SOA-Based Software Systems – an evolutionary Programming Approach”, In: IEEE Conference on Internet and Web Applications and Services ICIW’06, Guadeloupe / French Caribbean, February 2006.
18. OMG Business Process Modeling Notation, V1.2, 2009, available at http://www.omg.org/spec/BPMN/1.2/.
19. Fareghzadeh, N. (2008). Service Identifi cation Approach to SOA Development, In: Proceedings of World Academy of Science, Engineering and Technology, Vol. 35.
20. Z. Zhang, L. Ruimin, H. Yang, “Service identification and packaging in service-oriented reengineering”, Proc. of the 17th International Conference on Software Engineering and Knowledge Engineering, IEEE Computer Soc., 2005.
21. Comella-Dorda S., Wallnau K., Seacord R.C., Robert, J., (2000). “A survey of black-box modernisation approaches for information systems”. Proceedings of the International Conference on Software Maintenance. IEEE CS Press, pp. 173–183.
22. Ying Zou, Kostas Kontogiannis, “Web-based Legacy System Migration and Integration” the 4th International Conference in Systematic, Cybernetics and Informatics (SCI), Orlando, USA, July 2000, pp. 254-260.
23. L. Jianzhi, Z. Zhang, H.Yang., “A grid oriented approach to reusing legacy code in ICENI framework”, Proc. of the 2005 , IEEE I nternational Conference on Information Reuse and Integration, IEEE Computer Soc., 2005.
24. Canfora, G., Fasolino, A.R., Frattolillo, G., Tramontana, P.:” A wrapping approach for migrating legacy system interactive functionalities to service oriented architectures”. J. Systems and Software 81(4) (2008) 463-480
25. H.M. Sneed, “Integrating legacy software into a service oriented architecture”, Proc. 10th European Conference on SoftwareMaintenance and Reengineering, IEEE Computer Soc., 2006.