A COLLABORATIVE FRAMEWORK FOR MANAGING RUN-TIME CHANGES IN ENTERPRISE WEB SERVICES

Thirumaran. M[1], Dhavachelvan. P[2] and Naga Venkata Kiran. G[3]

[1,3]Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India
thirumaran@pec.edu, nagavenkatakiran@pec.edu
[2] Department of Computer Science, School of Engineering, Pondicherry Central University, Puducherry, India
dhavachelvan.csc.@pondiuni.edu.in

ABSTRACT

In a business atmosphere changes are ever occurring incidents which are supported by Information Technologies and there should be proper management techniques to handle these changes effectively. Also business processes must be aligned to IT to ensure the proper functionality of their services even after changes are propagated. Services might be moved, or relocated and may undergo changes during their life cycle. Hence there should be an effective framework for managing these changes without affecting the business functionality. This work stresses mainly on handling these dynamic changes to a business entity as it should support at business logic level and a new architecture model is proposed based on finite state machine theory of automata. It aligns IT with the business entity by automatically adapting the corresponding execution environment in case a change in the business process occurs.

KEYWORDS

Change Management, Policy Driven Approach, Finite State Machine, Business Logic, Business Process Automation.

1. INTRODUCTION

A web service is a piece of business logic that enables a paradigm shift in business structures allowing them to outsource required functionality from third party Web-based providers. A consumer of a web service is not tied to the service directly as the service interface can change over time without compromising the client's ability to interact with it. As services are not static, they undergo many changes during their life cycle. Hence there arises the need for an effective framework managing those changes without affecting the business functionality. Change management is the process responsible for controlling and managing the lifecycle of all changes in an IT environment. Web services provide a new approach for accessing systems in a loosely coupled, platform independent and standardized manner. However services might be moved, or relocated and may undergo changes during its life cycle. Hence there should be an effective framework for managing these changes without affecting the business functionality. Change management is the process responsible for controlling and managing the lifecycle of all changes in an IT environment. The goal of change management is to "ensure that standardized methods and procedures are used for efficient and prompt handling of changes, in order to minimize the impact of change-related incidents upon service quality and, consequently, improve the day-to-day operations of the organization. There are different approaches for managing changes in long term composed services at different levels. Changes can be propagated either at service

DOI : 10.5121/ijwest.2012.3306

developer's side or at service provider's site. If Web service is exposed to changes the Web service clients on the customer's side is neither part of the upgraded piece of software nor of the IT environment which is being upgraded. This probably results in erroneous behaviour whose real cause is located somewhere in the system. Changes shouldn't attract attention to themselves in this way. Ideally they shouldn't be noticeable at all. If this is not possible consumers should be notified about upcoming changes in advance. Based on this notification they could request additional details of upcoming changes and prepare their Web service clients accordingly. The necessary measures often depend on the change that has been carried out on the Web service provider side. Changes can be mainly classified into two ways – Internal changes and External changes. Internal changes to a web service are propagated by the service developer. External changes are the changes made by third party service providers and here, service consumers have to manage changes should have the capability to understand service descriptions, discover the services dynamically and invoke them. We propose a framework for managing changes that are propagated during run time using FSM based theory of automata.

2. RELATED WORKS

While different researchers proposed different methodologies for change management, they all included activities that help organizations transition from a current state to desired future state during the implementation of ERP systems to achieve the organizational outcome. Dimitris Apostolou et all presented an ontology-based approach for managing e-Government services that enables systematic response of e-Government systems to changes by applying formal methods for achieving consistency when a change is discovered; enables knowledgeable response of service designers and implementers to changes by utilizing design rationale knowledge [1]. Xumin Liu et all proposed a framework where managing changes in LCSs is modeled as a dual service query optimization process where in the first phase, the selection of Web services is based on using reputation as the key parameter and in the second phase, the non-functional QoWS is used to narrow down the set to those Web services that are both reputable and best meet the QoWS [2]. It is central to recognize that differences don't only exists among those reviewed methodologies; change management tasks are not the same even in one ERP project as different groups and individuals will be affected differently and therefore will need different strategies [3]. Combination of Ordinary Petri nets and Reconfigurable Petri nets were used to model the triggering changes and reactive changes, respectively by Salman Akram and proposed an automatic change management framework that is based on the Petri net models [4]. Christian Gerth and others described an approach that allows the semantic comparison of different business process models using a normal form. Business process models have been transformed into a process model term and presented a rewriting system consisting of several rules that transform process model terms into a normal form and compared the normalized terms to identify equivalent fragments and process models [5]. The dependencies between business processes and services in service oriented environment for managing changes were proposed by Yi Wang, Jian Yang and Weiliang Zhao. The taxonomy for changes of business processes and services has been established based on the service oriented business model. A set of change impact patterns have been identified. Functions for deriving impact scopes of a service change and a process change have been defined. This approach can be used as the foundation to analyze and control the ripple effect of changes of business processes and services [6]. Praveen K. Muthuswamy developed change scheduling algorithms that seek to attain the "change capacity" of the system. This change management process handles critical updates in the system that often use overlapping sets of servers, resulting in scheduling conflicts between the corresponding change classes [7]. Bruno Wassermann proposes the Change 2.0 approach to cross-domain change management based on an inversion of responsibility for impact assessment and the facilitation of cross-domain service process integration. In this approach, the initiator of a change is relieved of having to identify the set of potentially affected clients. Instead, each domain maintains its dependencies and change owners publish change notifications [8]. Xiang Luo, Koushik Kar et all considered the Change Management Process for Enterprise IT services with the goal of improving the efficiency of this process, i.e., minimizing change completion time and maximizing the “change capacity”. Their results indicate that a simple longest queue based scheduling approach works well in a wide range of practical scenarios [9]. Uttam Kumar Tripathi presented a methodology and system for changing SOA-based business process implementation at two layers: the design layer processes are modelled in the ontology-based semantic mark-up language for web services OWL-S, and for execution, the processes are translated into BPEL [10]. Change Management is a vital duty of project management and furthermore project managers can use project management web service as a consultant for change management [11]. Heiner Stuckenschmidt defined a formal representation for modular ontologies based on the notion of Distributed Description Logics and introduced an architecture that supports local reasoning by compiling implied axioms. They addressed the problem of guaranteeing the correctness and completeness of compiled knowledge in the presence of changes in different modules and proposed a heuristic for analyzing changes and their impact on compiled knowledge and guiding the process of updating compiled information that can often reduce the effort of maintaining a modular ontology by avoiding unnecessary recompilation [12]. To keep align IT with business, Knut Hinkelmann methodology heavily relies on the fact that the processes are represented in a declarative, semantically rich modelling language based on ontology –OWL-S [13]. There are several tools for managing changes for work flow management systems (WFMS) [14]. Changes to a single system can have an unexpected impact on other dependent systems. Boris Stumm’s Caro approach tried to cope with this problem by observing each system participating in the infrastructure and analyzing the impact of any change that occurs. The analysis process is driven by declaratively defined rules and works with a generic and extensible graph model to represent the relevant metadata that is subject to changes. This makes Caro applicable to heterogeneous scenarios and customizable to special needs [15]. Alexander Keller and Remi Badonnel have successfully integrated a work flow based change management system with a commercial service provisioning system that allows the execution of automatically generated change plans as well as the monitoring of their execution. Their prototype demonstrates that change plans, generated by the CHAMPS change management system, can be executed by the TIO deployment engine and that the BPEL4WS workflow language can be used effectively to describe change plans [16]. Requirements for ontologies that can be used in representing QoS constraints and in management of Web Services, with special focus on QoS metrics, measurement units, and currency units are discussed in [17].

Antonio Oliveira Filho distinguished between two kinds of concerns related to impact analysis: business-specific concerns, those related to stakeholders interested in checking if other business rules are impacted by the change and also need to be modified; and software-specific concerns, those related to stakeholders interested in the impacted software artifacts that need to be modified [18]. Mi Wang et alll proposed a method to automatically get the global services dependency matrix of the distributed cross-organization process via analyzing the private orchestration and the partial public choreography and introduced the impact analysis model to compute the impact that these changes have on the other services with the intra-service and inter-service change propagation matrix to calculate the service impact effects [19]. Change Management for semantic web services was published in [20]. An impact analysis model based on service dependency is discussed in [21]. Shuying Wang et alll developed the graph-based service dependency matrix and intra- and inter service relation matrix to calculate the service cohesions and impact effects. Also, they analyzed the proposed change operations and suggest service change specifications using the WSDL-S based annotation. Hua Xiao et alll presented a change impact analysis approach that integrates the change impact analysis in business process and the change impact analysis in source code, and proposed a formula to quantify the impact caused by changes made in business process level [22]. Yi Wang et alll’s methodology of change impact is analyzed based on the study of the dependency between services and business processes. Types of changes and change impact patterns are identified on the foundation of a service oriented business process model [23]. Anton Michlmayr et alll have presented an approach for service provenance which is integrated into the VRESCo runtime where several security mechanisms have been implemented to guarantee access control and integrity of service provenance information [24]. Mauro Andreolini et alll proposed a two-phase strategy that first aims to obtain a representative view of the load trend from measured system values and then applied this representation to support runtime decision systems .The two-phase strategy is based on stochastic functions that are characterized by a computational complexity that is compatible with runtime decisions [25].]. Eduardo et alll DynamiCoS, which is a framework for the dynamic and automated composition of services provided support to the whole dynamic service composition process [26].

3. FRAMEWORK FOR MANAGING RUN-TIME CHANGES

A framework for satisfying dynamic run time changes in web service is proposed along with the evaluation of changes using Finite State Machine and to handle the critical and emergency changes that arise during runtime rather than the standard and normal changes [Figure 1]. Also it provides a sophisticated change monitoring system through the change evaluation process. This approach incorporates the run time changes on the business logic more easily and also for evaluating the changes made at service business logic in terms of business workflow with respect to the change requirements, policy enforcement and the service performance. In our work we use finite state machine for run time evaluation and management of the business logic change activity. Hence the automated business logic evaluation model with dynamic change criterion analysis empower the task of service management tremendously by reducing the programming effort and development cost thus significantly enhancing the speed of business logic change adaption in the nascent business market.

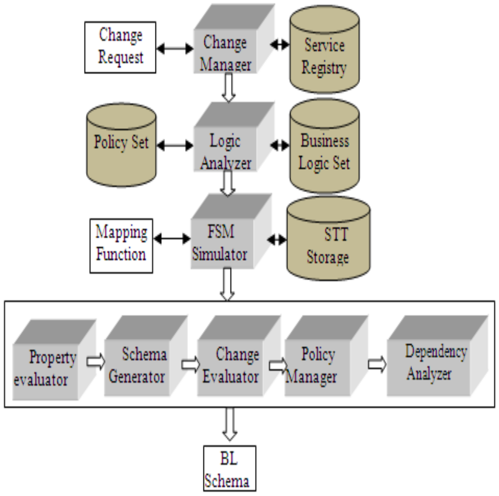


Figure 1. Framework for Policy driven Web Service Change Management

3.1. Working Model

The proposed work manages and implements the changes done to business logic in an effective manner. Here Business Logic (BL) is defined as a set consisting of Business Functions (BF), Business Parameters (BP), and Business Rules (BR).

$$BL= \{BF, BR ,BP, BD\} \qquad (1)$$

Also we identify the dependencies existing between functions, rules and parameters through the dependency set BD. If change request is generated dynamically, our system first measures the whether the change can be implemented or not. This is done as described in following sections.

3.1.1. Procedure/Steps Involved In Dynamic Business Logic Change Management

(i) The change request [CR] is the input to the source manager which consists of command, resource and condition.

CR = <command> <resource> <condition>,
Where,
<Command> →action raised by the user
<Resource> →where the change has to be implemented
<Condition> → the condition which the system has to be satisfied.

(ii) Once the change request is analyzed and measured to be complete and finite, corresponding business logic schema is composed for the change request generated.

(iii) Here, before implementing the change on a particular logic, the corresponding business functions and parameters which are dependent on the business entity to be changed are identified as this establishes the dependency set and overcomes the problem of incorrect functioning of service logic.

(iv) The dependency relations are then mapped into corresponding BL schema file

(v) The BL schema generation deals with generating a XML code that reflects changes to the web service business logic as expected and versioning of the schema file is updated.

(vi) Rule to be modified or changed is written to this file and these results in automatic change in the corresponding source code without the intervention of developer.

B. Advantages

(i) As business logic requirements change, business analysts can update the business logic without the support of IT staff.

(ii) Saves labour costs as everything i.e. the change management is done at the business analyst level.

(iii) Automates the business logic change management in an effective manner

(iv) Time can be saved drastically as there is no need to change the entire source code from the scratch

(v) All the changes to be implemented are dynamic in nature i.e. run time

4. MOTIVATING EXAMPLE

Consider a real time scenario for a banking service application. If a change request "modifywithdrawlimit" has to be validated which is generated dynamically first the functions, rules, parameters and dependency sets to the functions interdependent on this are checked. Here the change request is the rule to be modified. For example, the rule set corresponding to the function deposit () are: check_acc_holder () and check_policy_details (). If these rules are verified and validated to be true then only the change request is accepted by the source manager and the corresponding schema to the change request is generated automatically in dynamic fashion. Hence if change request is generated for one method, it indirectly depends on various rules which are interdependent. This change is made based on the dependency relation existing between the functions (i.e.) order of execution. Thus change in one particular function or rule can result in various modifications on other rules that fall under the same category or set. The methodology is based on verifying completeness and finiteness of changes in business rules propagated dynamically. This technique automatically supports the change and generates the new schema that satisfies the change requirement.

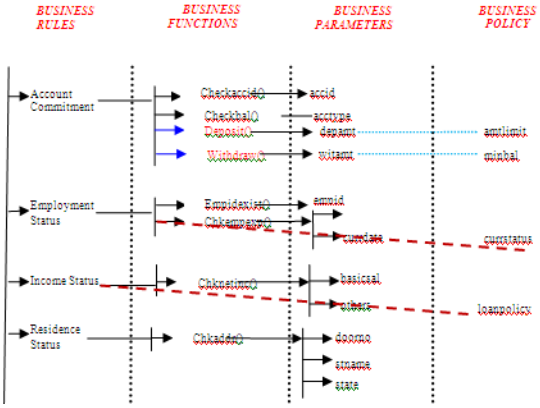
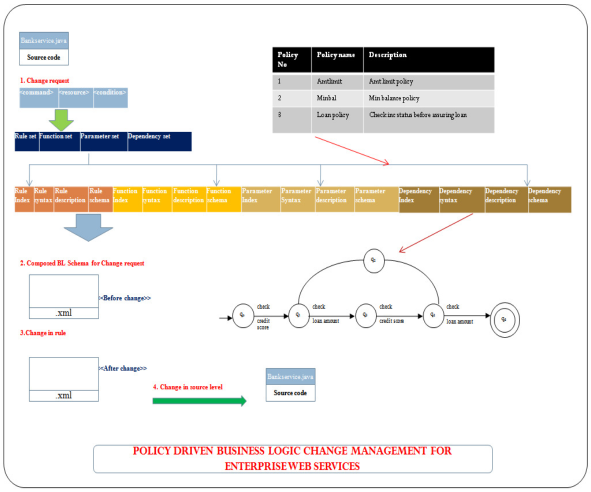


Figure 2. Policy Dependencies

The architecture flow of the proposed system is shown in the below figure.



Policy No	Policy name	Description
1	Amtlimit	Amt limit policy
2	Minbal	Min balance policy
3	Loan policy	Check inc status before assuring loan

Figure 3. Architecture Flow of the Proposed System

5. Architecture Evaluation Using ATAM

The proposed architecture is evaluated by the Software Architecture Trade off Analysis Method (ATAM). All the scenarios are listed and evaluated.

5.1. ATAM Process

Step 1 - Present the ATAM process:
ATAM stands for Architecture Trade off Analysis Method. It is a method that tells how well an architecture satisfies particular goals by providing insight of how quality goals interact and how they trade off.

Step 2 - Present the business drivers:
Due to the increase of business-to-business communication between different organizations over internet resources, the current architecture will provide and suggest the service provider to accept the service consumer requirements to add new features to perform their tasks reliably. The major quality attributes are listed as below.

Table 1. Quality Attributes.

Sl. no	Quality Attribute
1	Reliability
2	Scalability
3	Security
4	Adaptability
5	Modifiability
6	Maintainability
7	Performance

Step 3 - Identify architecture approaches:

Table 2. Architecture Rationale.

Architectural Approach	Rationale
Layered architecture	It organizes the system in hierarchical structure that allows for easy system change management.

Step 4 – Generate quality attribute utility tree

Table 3. Scenarios Classification.

Quality Attribute	Scenario
Scenario1: Performance	Precise output for any type of input
Scenario2: Scalability	Works for wide range of services
Scenario3: Security	Manages authenticated services alone
Scenario4: Adaptability	Supports changes in rules, parameters and functions in the service
Scenario5:Modifiability	Allows addition of new components in the architecture
Scenario6: Maintainability	Capability to accommodate new technology
Scenario7: Performance	Minimum response time
Scenario8: Reliability	Output of the evaluation system is trustworthy
Scenario 9: Scalability	Increased number of users
Scenario 10: Reliability	Handles all types of exceptions
Scenario11: Maintainability	Given service is evaluated perfectly
Scenario 12: Reliability	The property file produced is devoid of technical terms and easily understandable

Step 5- Scenarios Description

Table 4. Scenarios Description.

Quality Attribute	**Scenario**
Scenario1: Performance	**(Source)** User **(Stimulus)** User submits an order. **(Artifact)** Flow decider **(Environment)** Normal operations **(Response)** Precise output for any type of input **(Response Measure)** Not more than one person-day of team effort is required for the implementation
Scenario2: Security	**(Source)** External to system **(Stimulus)** Service provider experiences an overflow of calls through the Web Service Broker endpoint that do not provide legitimate information i.e., does not passes authentication and other security tests **(Artifact)** Flow decider **(Environment)** Normal operation **(Response)** Access to service is denied **(Response Measure)** Manages authenticated services alone
Scenario3: Adaptability	**(Source)** Business Analyst/user **(Stimulus)** Change a business rule **(Artifact)** BL Analyzer **(Environment)** Emergency operations **(Response)** Supports changes in rules, parameters and functions in the service **(Response Measure)** The system responds to the business analyst/user in less than seven seconds.
Scenario4: Modifiability	**(Source)** XML Schema **(Stimulus)** Add a new business process **(Artifact)** Flow decider **(Environment)** Emergency operations **(Response)** New process is added using architecture's web service **(Response Measure)** Not more than one person-day of team effort is required for the implementation
Scenario 5: Maintainability	**(Source)** Business Analyst/user **(Stimulus)** Verify syntax of business logic **(Artifact)** Business logic analyzer **(Environment)** normal operation **(Response)** Capability to accommodate new technology **(Response Measure)**
Scenario6 Performance	**(Source)** Internal to the system **(Stimulus)** Fault occurs at application server **(Artifact)** Service repository **(Environment)** normal operation **(Response)** The source control manager is notified of the fault; the system continues taking order requests; and data remains in consistent state. **(Response Measure)** The fault is detected, and failover action is taken within 25 seconds.
Scenario7: Reliability	**(Source)** System failure in the web/enterprise server **(Stimulus)** Fault occurs at web/enterprise server **(Artifact)** Service repository

	(Environment) Failure mode **(Response)** Output of the evaluation system is trustworthy **(Response Measure)** In 100% of the cases
Scenario 9: Reliability	**(Source)** Enterprise/Application server **(Stimulus)** Exception is raised by invalidated input **(Artifact)** Exception Handler **(Environment)** Emergency operations **(Response)** Handles all types of exceptions **(Response Measure)** The consumer is notified with the fault and the execution proceeds further.
Scenario 10 Maintainability	**(Source)** Business Analyst/Customer **(Stimulus)** Add information for existing service logic. **(Artifact)** Service repository **(Environment)** Emergency operations **(Response)** Necessary information is integrated with the system, and is available for users and given service is evaluated perfectly **(Response Measure)** The system responds to the user in less than seven seconds.
Scenario 11: Reliability	**(Source)** Business Analyst **(Stimulus)** Delete existing elements from the BL Schema **(Artifact)** Business analyst side **(Environment)** Emergency operations **(Response)** The property file produced is devoid of technical terms and easily understandable **(Response Measure)** The system responds to the business analyst/user in less than seven seconds.

Step 6 - Architecture elicitation and analysis

Table 5. Architecture Elicitation for Scenario 1.

Scenario Summary	Add a new service to the existing business logic schema
Business Goal(s)	Permit easy integration with new business services.
Quality Attribute	Modifiability
Architectural Approaches and Reasoning	interoperability issues across platforms are less likely to happen as the proposed architecture is designed for SOA i.e. web services technology
Risks	Not apply to architecture
Tradeoffs	Less likely to happen

Table 6. Architecture Elicitation for Scenario 11.

Scenario Summary	Add information for existing service logic.
Business Goal(s)	Must be easily integrated.
Quality Attribute	Maintainability
Architectural Approaches and Reasoning	Necessary information is integrated with the system, and is available for users and given service is evaluated perfectly
Risks	Not apply to architecture
Tradeoffs	Less likely to happen

5.2. Scenario Interaction Conflicts

Scenario interaction is to reveal changes in components when two or more scenarios affect components in architecture. Table 7 shows the list of scenarios which affects the component description.

Table 7. Scenarios affecting the components.

Components	Scenarios affecting the components
Service Repository	scenario 7, scenario 8, scenario 11
Business Logic analyzer	scenario 4, scenario 6
Flow decider	scenario 1,scenario 3, scenario 5
Dependency analyzer	Scenario 1, scenario 5, scenario 6
Exception handler	scenario 10

The performance measure for the above case study is given in the following graph depicted as Fig. 4

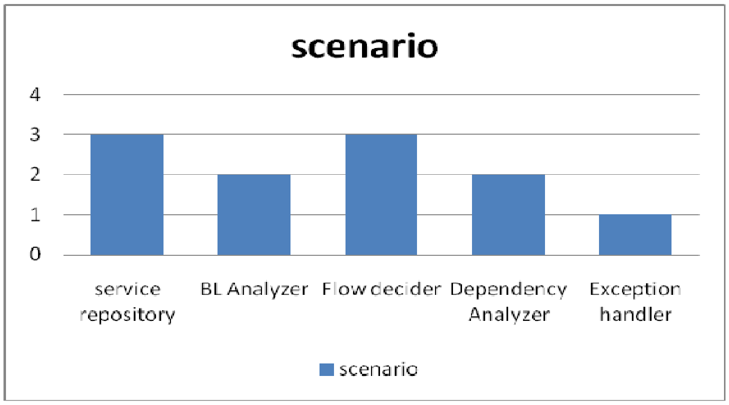


Figure 4. Scenarios affecting the components

6. Case study

We adopted travel agency architecture as case study for implementing the proposed system. The architecture of the system is shown below:

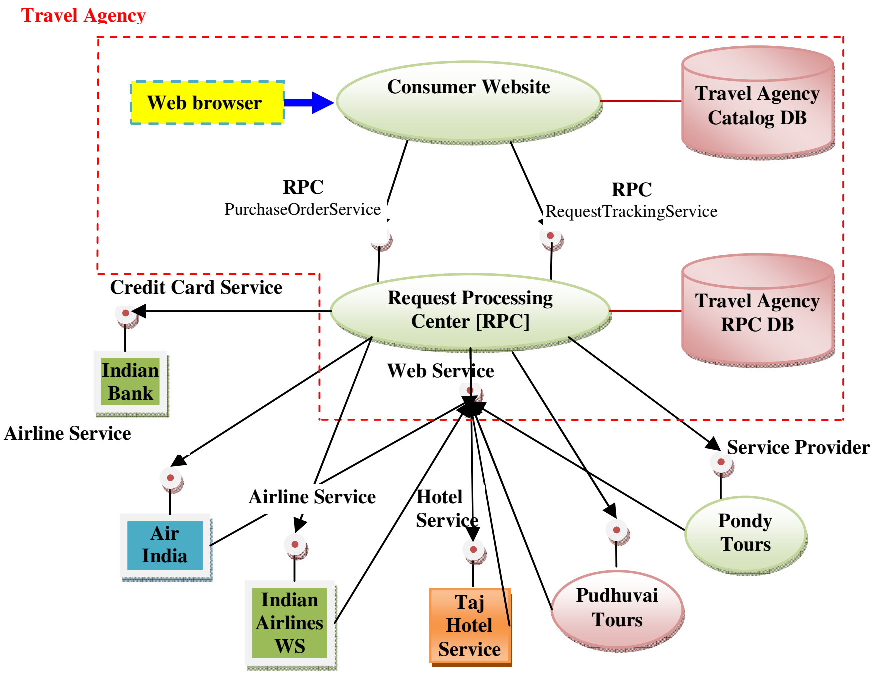
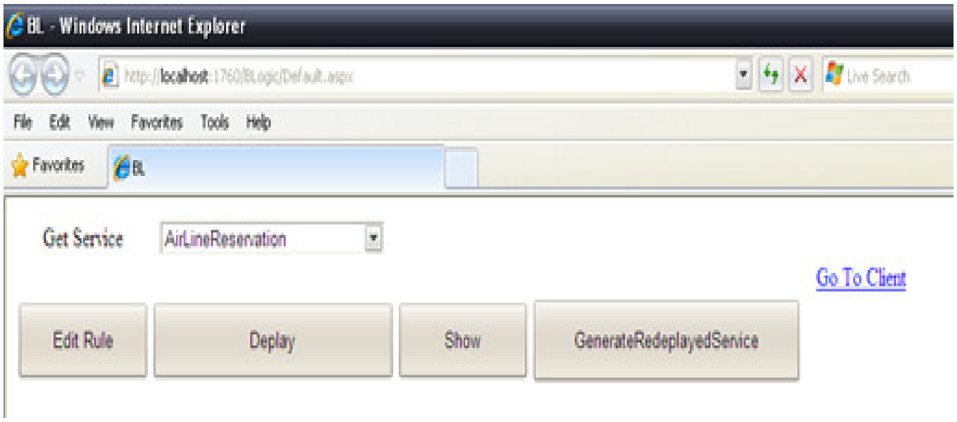


Figure 5. Run-time view of Travel Agency Web Services Based Architecture with Services added dynamically

7. Implementation Results

7.1 User Interface for Business Analyst

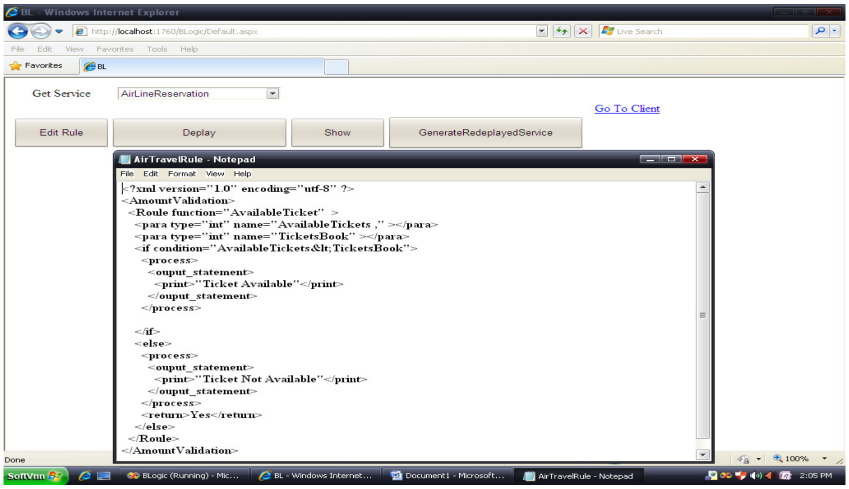
To accommodate the changes to be done to the business logic of the web service, a user interface has been designed for business analysts where they can specify the changes and implement by themselves.


BL - Windows Internet Explorer
http://localhost:1760/BLogic/Default.aspx
Live Search
File Edit View Favorites Tools Help
Favorites
BL
Go To Client
Edit Rule
Deplay
Show
GenerateRedeplayedService

Get Service AirLineReservation

7.2 BL Schema loaded for Airline reservation service editing

After specifying the change request to the system, the business logic schema in form of easily readable and understandable XML statements are generated for the particular / corresponding web service source code where any non-programmer can do the changes to it.

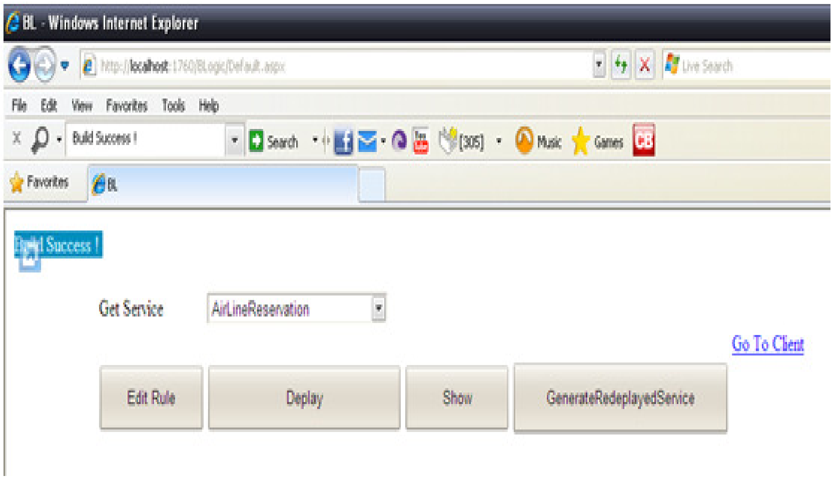

BL - Windows Internet Explorer
http://localhost:1760/BLogic/Default.aspx
Get Service
AirLineReservation
Go To Client
Edit Rule
Deplay
Show
GenerateRedeplayedService
AirTravelRule - Notepad
File Edit Format View Help

```
<?xml version="1.0" encoding="utf-8" ?>
<AmountValidation>
  <Roule function="AvailableTicket"  >
    <para type="int" name="AvailableTickets ," ></para>
    <para type="int" name="TicketsBook" ></para>
    <if condition="AvailableTickets&lt;TicketsBook">
      <process>
        <ouput_statement>
          <print>"Ticket Available"</print>
        </ouput_statement>
      </process>

    </if>
    <else>
      <process>
        <ouput_statement>
          <print>"Ticket Not Available"</print>
        </ouput_statement>
      </process>
      <return>Yes</return>
    </else>
  </Roule>
</AmountValidation>
```

7.3 Status after Changes done

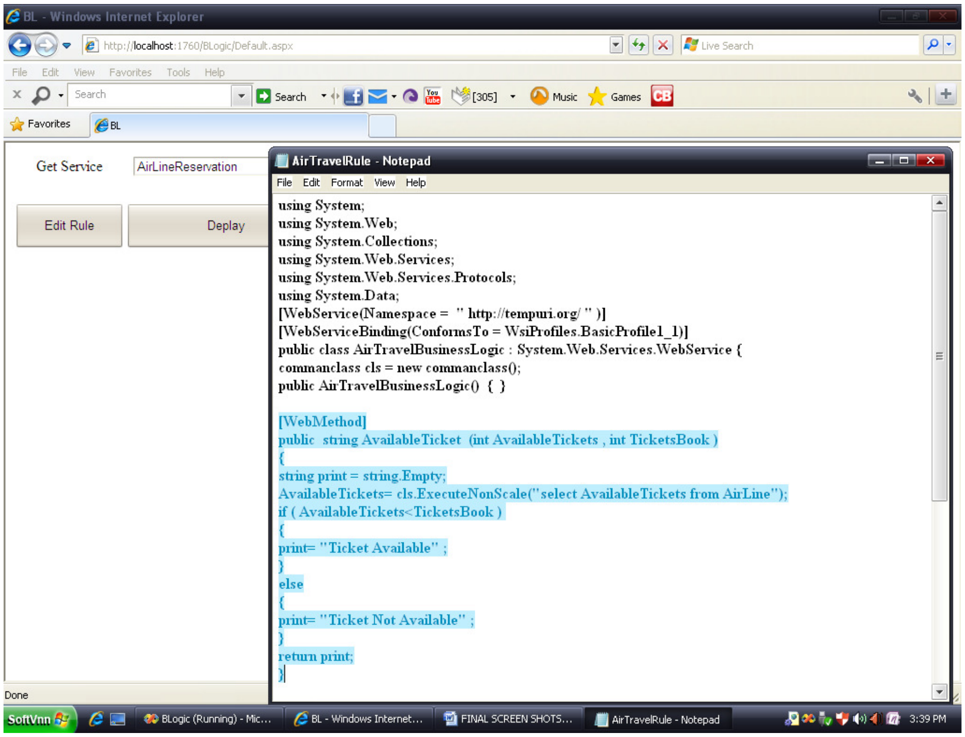
Once changes such as addition of business rules, deletion or modification has been done, the business logic of the web service will be regenerated wherever the new statements or modified statements exist in the BL Schema (Business logic schema). If any logical errors arise, exceptions are raised which gives detailed information about the error caused. If the system does not come across of any such error, “Build Successful” message is displayed to the user.


BL - Windows Internet Explorer
http://localhost:1760/BLogic/Default.aspx
Live Search
File Edit View Favorites Tools Help
Build Success !
Search
Music
Games
Favorites
BL
Get Service
AirLineReservation
Go To Client
Edit Rule
Deplay
Show
GenerateRedeplayedService

Build Success !

7.4 Service Code before changes

The below snippet shows the business logic of the web service before changes have been implemented.

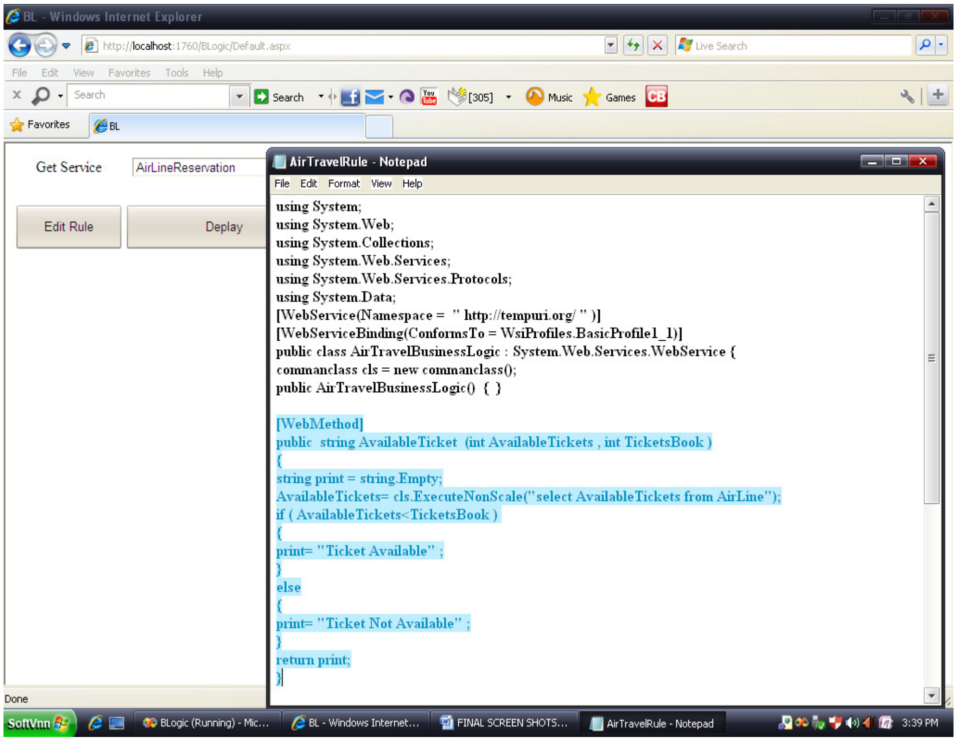

BL - Windows Internet Explorer
http://localhost:1760/BLogic/Default.aspx
Get Service
AirLineReservation
Edit Rule
Deplay
AirTravelRule - Notepad
File Edit Format View Help
Done

```
using System;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Data;
[WebService(Namespace = " http://tempuri.org/ " )]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class AirTravelBusinessLogic : System.Web.Services.WebService {
commanclass cls = new commanclass();
public AirTravelBusinessLogic() { }

[WebMethod]
public  string AvailableTicket  (int AvailableTickets , int TicketsBook )
{
string print = string.Empty;
AvailableTickets= cls.ExecuteNonScale("select AvailableTickets from AirLine");
if ( AvailableTickets<TicketsBook )
{
print= "Ticket Available" ;
}
else
{
print= "Ticket Not Available" ;
}
return print;
}
```

7.5 Service Code Re-generated after Changes done

If any changes are done to the business logic schema of the corresponding business rules of the source code, the service code will be re-generated and the process of compilation takes place.

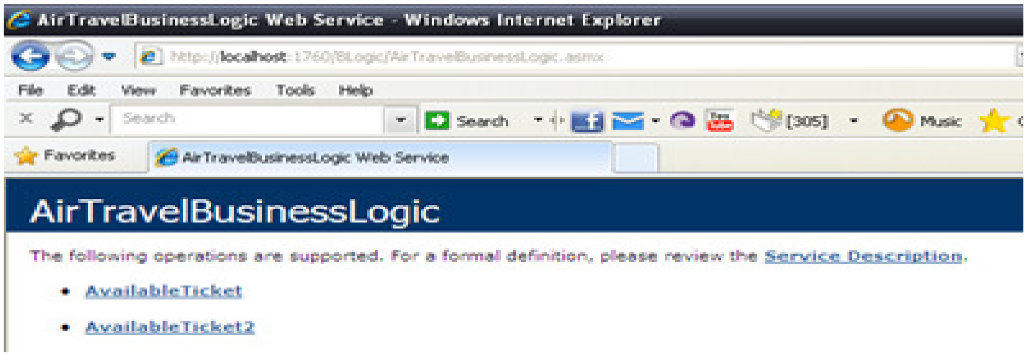

BL - Windows Internet Explorer
http://localhost:1760/BLogic/Default.aspx
Get Service
AirLineReservation
Edit Rule
Deplay
AirTravelRule - Notepad
File Edit Format View Help
Done

```
using System;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Data;
[WebService(Namespace = " http://tempuri.org/ " )]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class AirTravelBusinessLogic : System.Web.Services.WebService {
commanclass cls = new commanclass();
public AirTravelBusinessLogic() { }

[WebMethod]
public  string AvailableTicket  (int AvailableTickets , int TicketsBook )
{
string print = string.Empty;
AvailableTickets= cls.ExecuteNonScale("select AvailableTickets from AirLine");
if ( AvailableTickets<TicketsBook )
{
print= "Ticket Available" ;
}
else
{
print= "Ticket Not Available" ;
}
return print;
}
```

7.6 Service after re-deployment

Once the services are compiled successfully without any errors, then they should be redeployed automatically to make those services available to consumers/clients consuming this enterprise web service


AirTravelBusinessLogic Web Service - Windows Internet Explorer
http://localhost:1760/BLogic/AirTravelBusinessLogic.asmx
File Edit View Favorites Tools Help
Search
Search
[305]
Music
Favorites
AirTravelBusinessLogic Web Service

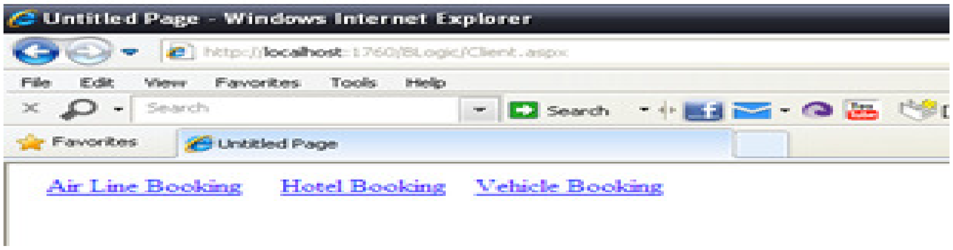
AirTravelBusinessLogic

The following operations are supported. For a formal definition, please review the Service Description.

- AvailableTicket
- AvailableTicket2

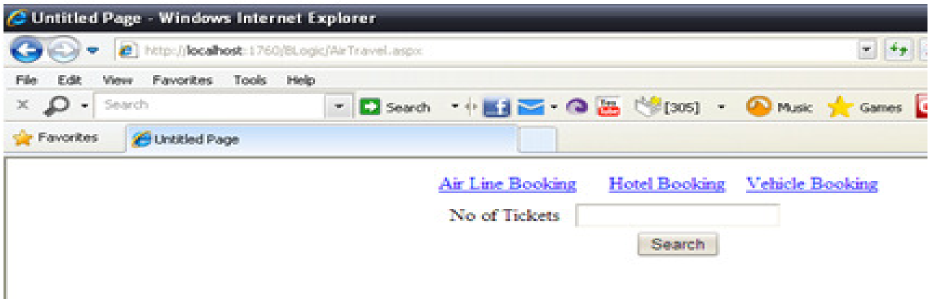
7.7 Client side

The below snippet shows the client side of the travel agency website consuming services from the enterprise web services architecture which we have built.


Untitled Page - Windows Internet Explorer
http://localhost:1760/BLogic/Client.aspx
File Edit View Favorites Tools Help
Search
Search
Favorites
Untitled Page
Air Line Booking
Hotel Booking
Vehicle Booking

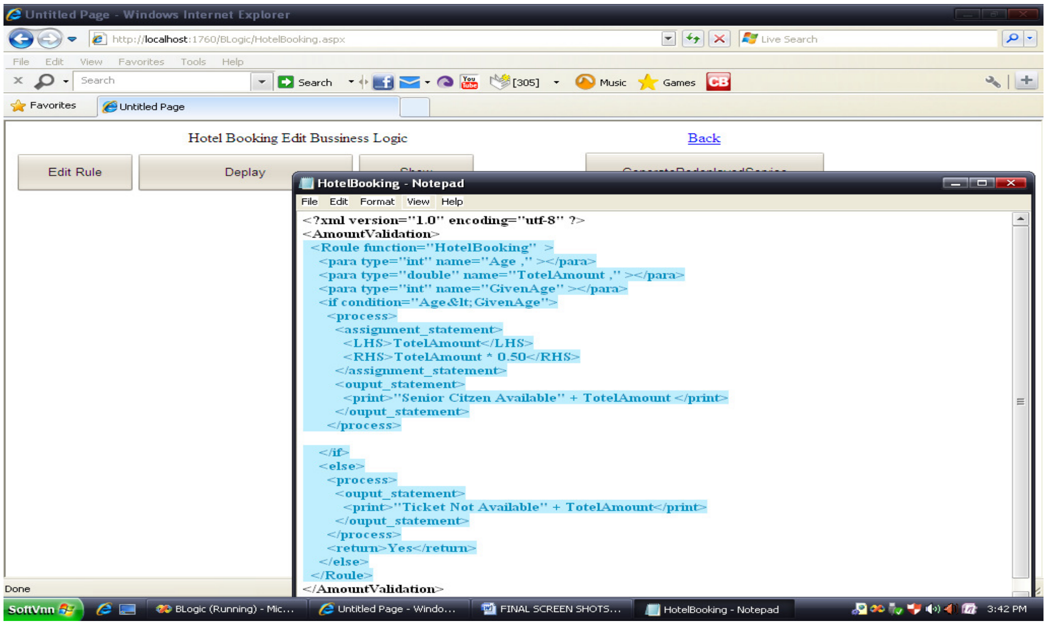
7.8 Airline booking service

The airline reservation service of the travel agency website is shown below. It gets number of tickets to be booked and process from the results from the service.


Untitled Page - Windows Internet Explorer
http://localhost:1760/BLogic/AirTravel.aspx
File Edit View Favorites Tools Help
Search
Search
[305]
Music
Games
Favorites
Untitled Page
Air Line Booking
Hotel Booking
Vehicle Booking
No of Tickets
Search

7.9 BL Schema of Travel Reservation service

The below snippet shows the business logic schema of travel reservation service in form of XML like statements.

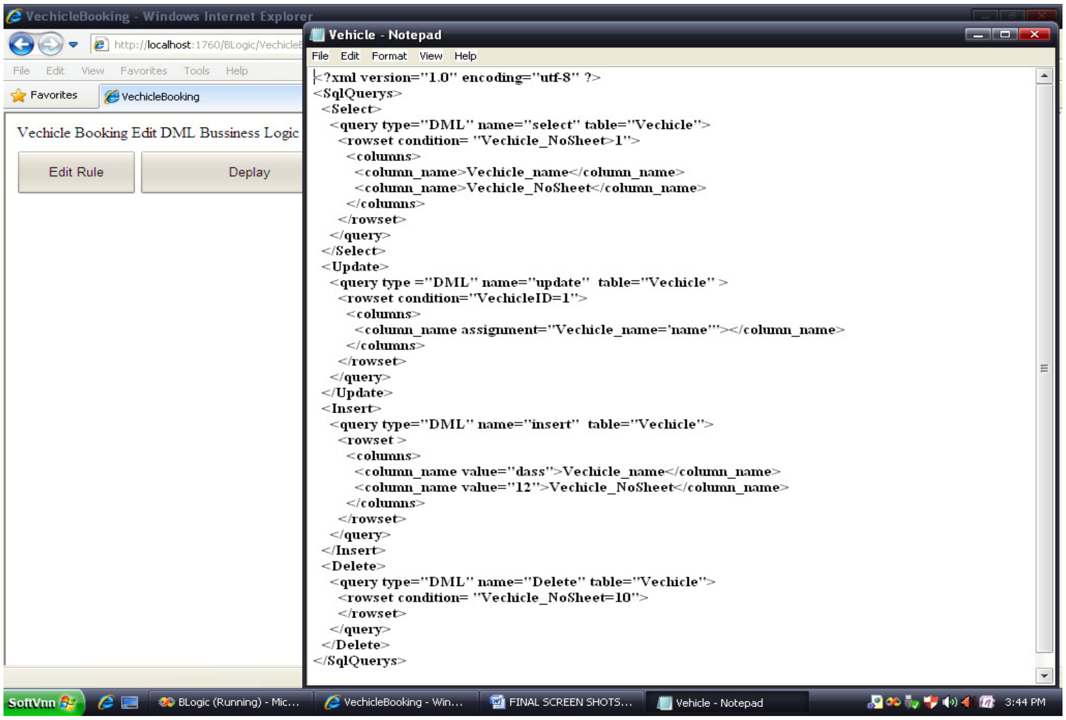

Untitled Page - Windows Internet Explorer
http://localhost:1760/BLogic/HotelBooking.aspx
Hotel Booking Edit Bussiness Logic
Back
Edit Rule
Deplay
HotelBooking - Notepad
File Edit Format View Help

```
<?xml version="1.0" encoding="utf-8" ?>
<AmountValidation>
 <Roule function="HotelBooking"  >
  <para type="int" name="Age ," ></para>
  <para type="double" name="TotelAmount ," ></para>
  <para type="int" name="GivenAge" ></para>
  <if condition="Age&lt;GivenAge">
   <process>
    <assignment_statement>
     <LHS>TotelAmount</LHS>
     <RHS>TotelAmount * 0.50</RHS>
    </assignment_statement>
    <ouput_statement>
     <print>"Senior Citzen Available" + TotelAmount </print>
    </ouput_statement>
   </process>

  </if>
  <else>
   <process>
    <ouput_statement>
     <print>"Ticket Not Available" + TotelAmount</print>
    </ouput_statement>
   </process>
   <return>Yes</return>
  </else>
 </Roule>
</AmountValidation>
```

7.10 BL Schema loaded for editing DML statements [SQL Queries]

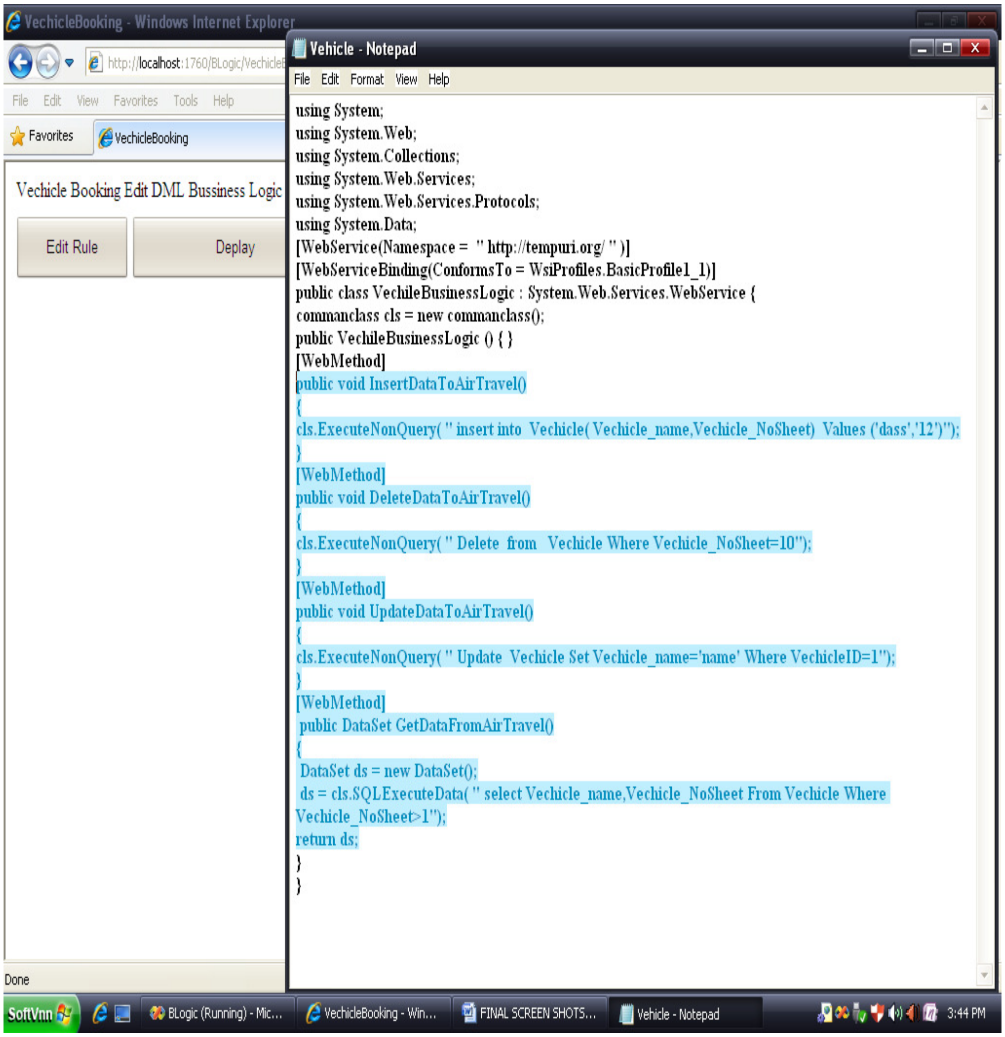
The software prototype developed is also capable of making changes to Structured Query Language statements such as Data Manipulation Language queries such as Select, Insert, Modify, Delete and Update. The below snippet shows the interface where the analyst is capable of making changes to such type of constructs.


VechicleBooking - Windows Internet Explorer
Vechicle Booking Edit DML Bussiness Logic
Edit Rule
Deplay
Vehicle - Notepad
File Edit Format View Help

```
<?xml version="1.0" encoding="utf-8" ?>
<SqlQuerys>
 <Select>
  <query type="DML" name="select" table="Vechicle">
   <rowset condition= "Vechicle_NoSheet>1">
    <columns>
     <column_name>Vechicle_name</column_name>
     <column_name>Vechicle_NoSheet</column_name>
    </columns>
   </rowset>
  </query>
 </Select>
 <Update>
  <query type ="DML" name="update"  table="Vechicle" >
   <rowset condition="VechicleID=1">
    <columns>
     <column_name assignment="Vechicle_name='name'"></column_name>
    </columns>
   </rowset>
  </query>
 </Update>
 <Insert>
  <query type="DML" name="insert"  table="Vechicle">
   <rowset >
    <columns>
     <column_name value="dass">Vechicle_name</column_name>
     <column_name value="12">Vechicle_NoSheet</column_name>
    </columns>
   </rowset>
  </query>
 </Insert>
 <Delete>
  <query type="DML" name="Delete" table="Vechicle">
   <rowset condition= "Vechicle_NoSheet=10">
   </rowset>
  </query>
 </Delete>
</SqlQuerys>
```

7.11 Business logic (source code) for DML statements

The business logic of DML statements are shown in the below snippet.


VechicleBooking - Windows Internet Explorer
http://localhost:1760/BLogic/VechicleB
File Edit View Favorites Tools Help
Favorites
VechicleBooking
Vechicle Booking Edit DML Bussiness Logic
Edit Rule
Deplay
Done
Vehicle - Notepad
File Edit Format View Help
SoftVnn
BLogic (Running) - Mic...
VechicleBooking - Win...
FINAL SCREEN SHOTS...
Vehicle - Notepad
3:44 PM

```
using System;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Data;
[WebService(Namespace =  " http://tempuri.org/ " )]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class VechileBusinessLogic : System.Web.Services.WebService {
commanclass cls = new commanclass();
public VechileBusinessLogic () { }
[WebMethod]
public void InsertDataToAirTravel()
{
cls.ExecuteNonQuery( " insert into  Vechicle( Vechicle_name,Vechicle_NoSheet)  Values ('dass','12')");
}
[WebMethod]
public void DeleteDataToAirTravel()
{
cls.ExecuteNonQuery( " Delete  from   Vechicle Where Vechicle_NoSheet=10");
}
[WebMethod]
public void UpdateDataToAirTravel()
{
cls.ExecuteNonQuery( " Update  Vechicle Set Vechicle_name='name' Where VechicleID=1");
}
[WebMethod]
 public DataSet GetDataFromAirTravel()
{
 DataSet ds = new DataSet();
 ds = cls.SQLExecuteData( " select Vechicle_name,Vechicle_NoSheet From Vechicle Where
Vechicle_NoSheet>1");
return ds;
}
}
```

8. CONCLUSIONS

The proposed model which has been implemented provides good benefits to the business analysts who want to implement changes at the business logic layer of the software process (addition, deletion and modification of business logic) by themselves without the help of technical staff and hence reduces the implementation costs. The change management process can be done very easily and provides much flexibility. First, the analyst gets the services of the enterprise web service which are available and business logic schema file is loaded for the service logic where he wants to do modifications. After doing changes, the corresponding source code is generated for the changes done in the BL Schema. And finally the web service is re-deployed automatically for the changes done in the service code through BL Schema. At the client side, the modified services can be accessed and can be included into the software process. The current system was modelled using FSM (Finite State Machine) for analyzing dependencies where the states are represented as business rules (i.e. functions). However at enterprise architecture level, business analysts look at the system as a whole process and not only at business logic level. So we can extend the present system by modelling it through Cellular automata where the processes of the whole enterprise is constructed as state spaces (cells) which gives more clear view of the system to the analysts for implementing the process of web services change management.

ACKNOWLEDGEMENTS

The authors would like to the panel review members of ICCSEA 2012, New Delhi who have given valuable suggestions and comments to improve the technical content of the paper “A Policy Driven Business Logic Change Management for Enterprise Web Services”. This research article submitted to this journal is the extended version of that paper.

REFERENCES

[1] Dimitris Apostolou, Gregoris Mentzas, Ljiljana Stojanovic, Barbara Thoenssen, Tomás Pariente Lobo, “A collaborative decision framework for managing changes in e-Government services”, Science Direct, Government Information Quarterly ,pp. 101–116, 2010.

[2] Xumin Liu, Athman Bouguettaya, Qi Yu, Zaki Malik, “Efficient change management in long-term composed services”, Original Research Paper, Springer-Verlag, 2010.

[3] Hailah Alballaa, Abdullah S. Al-Mudimigh, “Change Management Strategies for Effective Enterprise Resource Planning Systems: A Case Study of a Saudi Company”, in International Journal of Computer Applications, pp.0975 – 987 Volume 17– No.2, March 2011.

[4] Salman Akram, Athman Bouguettaya, Xumin Liu, Armin Haller, Florian Rosenberg, Xiaobing Wu, “A Change Management Framework for Service Oriented Enterprises”, International Journal of Next-Generation Computing (IJNGC), Vol. 1, No. 1, 09 2010, Pages 1-077.

[5] Christian Gerth, Markus Luckey, Jochen M. Kuster and Gregor Engels, “Detection of Semantically Equivalent Fragments for Business Process Model Change Management”, IEEE International Conference on Services Computing 2010, IEEE Computer Soceity.

[6] Yi Wang, Jian Yang and Weiliang Zhao, “Managing Changes for Service Based Business Processes”, IEEE Asia-Pacific Services Computing Conference 2010,IEEE Computer Society.

[7] Praveen K. Muthuswamy, Koushik Kar, Sambit Sahu , Prashant Pradhan and Saswati Sarkar, “Change Management in Enterprise IT Systems: Process Modeling and Capacity-optimal Scheduling”, IEEE INFOCOM 2010 proceedings, IEEE Communications Society.

[8] Bruno Wassermann, Heiko Ludwig, Jim Laredo, Kamal Bhattacharya, Liliana Pasqual, “Distributed Cross-Domain Change Management”, IEEE International Conference on Web Services 2009, IEEE Computer Soceity.

[9] Xiang Luo, Koushik Kar, Sambit Sahu, Prashant Pradhan, Anees Shaikh, “On Improving Change Management Process for Enterprise IT Services”, IEEE International Conference on Services Computing 2008, IEEE Computer Soceity.

[10] Uttam Kumar Tripathi, Knut Hinkelmann and Daniela Feldkamp, “Life Cycle for Change Management in Business Processes using Semantic Technologies”, Journal Of Computers, Vol. 3, No. 1, January 2008.

[11] Shahriar Mohammadi, Ali Khalili, “A Semantic Web Service-Oriented Model for Project Management”, IEEE 8th International Conference on Computer and Information Technology Workshops, IEEE Computer Soceity.

[12] Heiner Stuckenschmidt, Michel Klein, "Reasoning and change management in modular ontologies", Data & Knowledge Engineering 63 (2007) 200–223, Elsevier B.V.

[13] Uttam Kumar Tripathi, Knut Hinkelmann, "Change Management in Semantic Business Processes Modeling", Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07), IEEE Computer Soceity.

[14] Qi Yu, Xumin Liu, Athman Bouguettaya, Brahim Medjahed, "Deploying and managing Web services: issues, solutions, and directions", The VLDB Journal (2008) 17:537–572, Springer-Verlag 2006.

[15] Boris Stumm, "Change Management in Large-Scale Enterprise Information Systems", EDBT 2006 Workshops, LNCS 4254, pp. 86–96, 2006, Springer-Verlag Berlin Heidelberg 2006.

[16] Alexander Keller and Remi Badonnel, "Automating the Provisioning of Application Services with the BPEL4WS Workflow Language", Proceedings of the 15th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 2004), Davis, CA, USA, November 2004.

[17] Vladimir Tosic, Babak Esfandiari, Bernard Pagurek, and Kruti Patel. "On Requirements for Ontologies in Management of Web Services", WES 2002, LNCS 2512, pp. 237–247, 2002. Springer-Verlag Berlin Heidelberg 2002.

[18] Antonio Oliveira Filho. "Change Impact Analysis from Business Rules", ICSE '10, May 2-8 2010, Cape Town, South Africa, ACM 2010.

[19] Mi Wang and LiZhen Cui. An Impact Analysis Model For Distributed Web Service Process, Proceedings of the 2010 14th International Conference on Computer Supported Cooperative Work in Design, IEEE Press.

[20] Xumin Liu, Salman Akram,Athman Bouguettaya. Change Management for Semantic Web Services, Springerlink.

[21] Shuying Wang, Miriam A. M. Capretz. A Dependency Impact Analysis Model For Web Services Evolution, IEEE International Conference on Web Services, 2009, IEEE Computer Society.

[22] Hua Xiao, Jin Guo and Ying Zou. Supporting Change Impact Analysis for Service Oriented Business Applications, IEEE International Workshop on Systems Development in SOA Environments (SDSOA'07)

[23] Yi Wang, Jian Yang and Weiliang Zhao. Change Impact Analysis for Service Based Business Processes.

[24] Anton Michlmayr, Florian Rosenberg, Philipp Leitner, and Schahram Dustdar. Selective Service Provenance in the VRESCo Runtime, International Journal of Web Services Research , Vol.X, No.X, 2010.

[25] Mauro Andreolini, Sara Casolari, and Michele Colajanni. Models and Framework for Supporting Runtime Decisions in Web-Based Systems, ACM Transactions on the Web, Vol. 2, No. 3, Article 17, Publication date: July 2008.

[26] Eduardo Goncalves da Silva, Lus Ferreira Pires, Marten van Sinderen. Towards Runtime Discovery, Selection and Composition of Semantic Services, Computer Communications March 26, 2010.

[27] Yang Xiao, Urban, S., "Using Rules and Data Dependencies for the Recovery of Concurrent Processes in a Service-Oriented Environment", IEEE Transactions on Services Computing, Issue 1, Volume 5, 2012, pp.59-71.

[28] Yiwei Gong, Marijn Janssen, "From policy implementation to business process management: Principles for creating flexibility and agility", Government Information Quarterly, Volume 29, Supplement 1, Elsevier B.V, January 2012, Pages S61–S71.

[29] Raúl Palmaa, Oscar Corchoa, Asunción Gómez-Péreza, Peter Haase, “A holistic approach to collaborative ontology development based on change management”, Elsevier journal of Web Semantics: Science, Services and Agents on the World Wide Web, Volume 9, Issue 3, September 2011, Pages 299–314.

[30] Xumin Liu, Athman Bouguettaya, Jemma Wu, and Li Zhou, "Ev-LCS: A System for the Evolution of Long-term Composed Services", IEEE TRANSACTIONS ON SERVICES COMPUTING, Issue: 99, ISSN : 1939-1374, 23 June 2011.

[31] Hans Weigand, Willem-Jan van den Heuvel, Marcel Hiel, "Business policy compliance in service-oriented systems", Elsevier journal of Information Systems, Volume 36, Issue 4, June 2011, Pages 791–807.

[32] H. Yahyaoui, L. Wang, A. Mourad, M. Almullah, Q.Z. Sheng,"Towards context-adaptable Web service policies", in Procedia Computer Science, Volume 5, 2011, Pages 610–617.

[33] Pedro Antunesa, Hernâni Mourão, "Resilient Business Process Management: Framework and services", Elsevier journal of Expert Systems with Applications, Volume 38, Issue 2, February 2011, Pages 1241–1254.

[34] Harry Jiannan Wang, J. Leon Zhao, "Constraint-centric workflow change analytics", Elsevier journal of Decision Support Systems, Volume 51, Issue 3, June 2011, Pages 562–575.

[35] Shari S.C. Shang, Eldon Y. Li,Ya-Ling Wu, Oliver C.L. Hou, "Understanding Web 2.0 service models: A knowledge-creating perspective", Elsevier journal of Information & Management, Volume 48, Issues 4–5, May 2011, Pages 178–184.

[36] Minhong Wang, Huaiqing Wang, "From process logic to business logic—A cognitive approach to business process management", Elsevier journal of Information & Management, Volume 43, Issue 2, March 2006, Pages 179–193.

[37] Daniel Żmuda, Marek Psiuk, Krzysztof Zieliński, "Dynamic monitoring framework for the SOA execution environment", in proceedings of ICCS 2010,Procedia Computer Science, Volume 1, Issue 1, May 2010, Pages 125–133.

[38] Ontology Support for Managing Top-down Changes in Composite Services.X. Liu, A. Bouguettaya, The 4th International Conference on Collaborative Computing (CollaborateCom 2008), Orlando, FL, Nov. 13-16. 2008.

[39] Thirumaran. M, Dhavachelvan. P and NagaVenkataKiran. G, “Business Process Property Evaluation Engine for Interactive and Reliable Business Process Modelling” proceedings of second international conference on Recent Trends in Information Technology, April 19-21, 2012, MIT Campus, Anna University, Chennai. Conference proceedings available online at IEEE Xplore and archived in IEEE Xplore Digital library.

[40] Thirumaran. M, Dhavachelvan. P and NagaVenkataKiran. G, “A Policy Driven Business Logic Change Management for Enterprise Web Services”, proceedings of the second International Conference on Computer Science, Engineering and Applications (ICCSEA 2012), May 25-27, New Delhi, India, Volume 2. Conference proceedings available online at springerlink.com and published by Springer in Advances in Intelligent and Soft Computing book Series with ISBN 978-3-642-30110-0 and ISSN 1867-5662.

Authors

Mr. M. Thirumaran is currently working as Assistant Professor, Department of Computer Science and Engineering at Pondicherry Engineering College, India. He pursued his B.Tech in Computer Science and Engineering at Pondicherry Engineering College and M.Tech in Computer Science and Engineering at Pondicherry University, A Central University, India. Currently he is pursuing doctoral research program at Pondicherry University under supervision of Dr. P. Dhavachelvan. He had also cleared the UGC NET (National Eligibility Test) for Lectureship three times in his career. His areas of research interest include Business Logic management Systems, Business Object Model. Currently he is working on developing a model for Business Logic Systems for various E-Commerce systems. Six of his conference publications are available online at IEEE Xplore Digital Library, two research papers have been archived in ACM Digital library and three papers are available online at Springerlink.com.

Dr. P. Dhavachelvan is currently the Head and Professor at the Department of Computer Science, School of Engineering, Pondicherry University- A Central University, India. He has obtained M.E and Ph.D in Computer Science and Engineering at College of Engineering, Anna University, Chennai. He has qualified Graduate Aptitude Test for Engineers (GATE) two times in his career with 89.50 and 81.25 percentile scores in 1997 and 1998 respectively. He is a good academician as he got overall 9th rank in state level in SSLC and University 3rd rank in M.E. Seven of his research publications (journal papers) have been published by Elsevier and are archived in Science Direct. He is a member of IEEE, Fellow member of ACEEE, Life member of ISTE, Editorial member and reviewer of various journals. He was awarded best paper award for his presentation in ISTE organized national conference in 2004 and was ranked among the top 25 research publications in Elsevier journals for two times in 2006 and 2007. He has 16 conference publications archived in IEEE Xplore, 5 publications in ACM digital library through ACM conferences and 11 publications in Springer Link Digital Library.

Mr. G. NagaVenkataKiran has completed his Master of Technology in Computer Science and Engineering with specialization in Distributed Computing Systems at Pondicherry Engineering College, India in the year 2012 and is presently designated as Trainer and Learning developer at SRM Infotech, Pondicherry, a division of SRM Learning Tree Pvt. Ltd., Chennai. He has been awarded first place for performance in Technology Up gradation [TECH UP 2011-2012] in .NET conducted at Cuddalore region on 19th August, 2011 by SRM Infotech Computer Education, an ISO 9001:2008 certified institution and division of SRM Learning Tree Pvt. Ltd, Chennai. Earlier he had worked as Software Programmer at Navayuga Computer Solution, Pondicherry, India for a period of twenty months from December 2008 to July 2010. His areas of interest include Web Services technology, Business Management Systems etc. He has two research publications through conference proceedings. One of his papers is archived in IEEE Xplore digital library and the other article is published by Springer in Advances in Intelligent and Soft Computing book Series through the conference proceedings of the second International Conference on Computer Science, Engineering and Applications (ICCSEA 2012), May 25-27, New Delhi, India and is available online at Springerlink.com.