International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 9 – No.1, June 2015 – www.ijais.org

Change Management in Semantic Web Services in Legal Domain using FSM and XXM

Gaurav Kant Shankhdhar
Asst. Professor
SOCA
Babu Banarasi Das University,
Lucknow, UP, India

Narendra Kumar Jha
Asst. Professor
SOCA
Babu Banarasi Das University,
Lucknow, UP, India

Atul Verma
Asst. Professor
SOCA
Babu Banarasi Das University,
Lucknow, UP, India

ABSTRACT
The effort here focuses on Semantic support, Change specification and Change control of web services related to Indian law coordinated by a control web service designed, namely, Web Service Controller Architecture (WSCA) in the proposed legal system, LegalCosmos. The change here mainly constitutes the changes made to the Law Acts, called amendments. The amendments are necessary to ensure that the latest updated Acts (through Indian Penal Code, IPC) be imposed in the whole process of legal system. The LegalCosmos is presently assumed to utilize the electronic static version of Legal Acts. Legal Cosmos involves various web services like WS_PublicInterface, WS_AdvocateAssociation and WS_Judgments. The web services namely WS_UnionOfIndia and WS_StateAct have to be added to introduce the feature to update their existing Acts with the amendments done, in case, to those acts and exclusively export the amendments to the WSCA. The amendments contain the past details of the Act, revised Act (amendment), date of amendment and when it comes into effect. LegalCosmos primarily functions to electronically relate the process of online allotment of advocates to the user in need, the liberty to refer judgments, notify the clients with updates in his/her case and satisfy the user queries related to law.

General Terms
Change management, law, amendments, semantic web

Keywords
Legal change management, semantic web, xxm, fsm, law, amendments

1. INTRODUCTION
In this paper we discuss these elementary challenges of change management in the area of Web services incorporated and managed by Web Service Change Management, WSCM). Currently there are no sound mechanisms and engineering principles for changing Web services and their related entities. Through analysis of a suitable scenario, specifically looking at the details of the Web service provider and consumer roles, one can identify the various problems that exist in this domain. Therefore we will start our approach with the consideration of an application scenario from the business domain of application and change management, the addition of Web services in private legal sector. With the new paradigm of Service Oriented Computing, many enterprises attempt to utilize services as fundamental elements for developing applications/solutions as an additive layer on top of existing components. The Web Service Controller Architecture (WSCA) for service-based, enterprise-scaled business solutions provides exactly this flexibility. The design, exposure and management of services can be accomplished through a Service Oriented Architecture (SOA)[1] that supports the usage, composition and coordination of services in a loosely coupled manner. Web services appear to be particularly suitable for addressing these aspects of a SOA. Furthermore, composition languages such as BPEL add value by composing Web services and facilitating the implementation of business processes. As the SOA paradigm brings this big behavior change relying on loose coupling of services it raises new questions in the area of change management. Change management is a central aspect in any software design, but its complexity for Web services is increased by both composition languages and loose coupling. The resulting advantages like composability, autonomy, message-based communication, and the avoidance of combined compilation and binding prove to be deficiencies in this context.

2. LITERATURE REVIEW
2.1 Service-Oriented Architecture
A service-oriented IT architecture is an architectural style for building software applications that use available services in a network. It is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. An SOA may support a variety of different communication protocols, but common protocols based on open standards (e.g., SOAP and WSDL) are used in current SOA implementations. These open standards are by no means the only technology with which an SOA can be established, and it is likely that most large SOAs will also provide access to services with a mix of technologies that are not necessarily based on the WS standards. Nonetheless, Web services based on WS standards are at the core of the integration products of the major vendors, including IBM, SAP, and Oracle, and according to recent industry surveys, they remain the prevalent underlying technology standard for the major SOA platforms[1]. Perhaps the key aspect that differentiates SOA with Web services from prior attempts of distributed computing (e.g., using CORBA) is the level of standardization and ubiquitous acceptance of these standards by the major vendors and service providers. Thus, we focused our attention in this paper on SOAs realized with Web services. For the purpose of this research, we adopted the definition for Web services put forth by the World Wide Web Consortium (W3C), which states that Web services are software systems designed to support interoperable machine-to-machine interaction over a network with an interface described in a machine-processable format, specifically the

Web Services Description Language (WSDL). Other systems interact with Web services in a manner prescribed by its description using SOAP messages, typically conveyed using Hypertext Transfer Protocol (HTTP) with an Extensible Markup Language (XML) serialization in conjunction with other Web-related standards. A service registry based on the Universal Description Discovery & Integration (UDDI) standard can be employed to publish and discover Web services. Web services enable the SOA concept to be applied in a Web based environment. According to the Gartner 2008 hype cycle for emerging technologies [2] basic Web services are seen to be fairly mature and reaching the "plateau of productivity," while SOA is placed on the "slope of enlightenment" as costs and benefits are now being viewed more realistically. Best practices for SOA, however, have still yet to mature [1]. SOA adoption overall is increasing, albeit slower that initially anticipated [3]. Consequently, the question of how to arrive at an SOA that enables the expected business benefits at an acceptable cost merits further examination and best practices need to be developed.

2.2 XXM:

In this paper we explore the design changes as made to software projects by the use
of a formal model known as Extreme X-Machines (XXM) [4], which are based on the work of Eilenberg and Holcombe [4]. An XXM model describes the functionality of the software without defining exactly how this functionality is achieved. This perspective allows an analysis of functional change whilst excluding specific implementation or requirements issues.

XXM are a state based model, they are intended to be used by developers as a method to design their systems from at the top level but here, XXM are used as effectively to analyze and incorporate changes in, even addition of new web services. Each model typically consists of a set of states which correspond to screens in the final system and functions which link the screens together. The functions are typically labeled with an enabling action such as “click_ok” which corresponds to a user clicking the OK button.

3. AN OVERVIEW OF A WSCA

A WSCA consists of several autonomous outsourced Web services, but acts as a virtually coherent entity. Business entities, in the form of Web services, are often geographically distributed and organizationally independent. While WSCA has a potential to introduce new business opportunities through dynamic alliances, the challenges of fully realizing a WSCA lie in managing changes during its lifecycle through Extreme X Machines (XXM).

Figure depicts the architecture of a WSCA. There are two key components and two supporting components in this architecture. The key components include a WSCA schema and a WSCA instance. The two supporting components include ontology providers and Web service providers.

• **WSCA schema**: A WSCA schema consists of a set of abstract services and the relationships among these services. An abstract service specifies one type of functionality provided by the Web services. They are not bounded to any concrete services. They are defined in terms of service concepts in a Web service ontology.

•**WSCA instance:** A WSCA instance is a composition of a set of concrete services, which instantiates a WSCA schema. It actually delivers the functionality and performance of a WSCA.

• **Ontology providers:** The ontology provider manages and maintains a set of ontologies that describe the semantics of Web services. A WSCA outsources semantics from an ontology provider to build up its schema.

• **Web service providers:** The Web service providers offer a set of Web services, which can be outsourced to form WSCA instances.

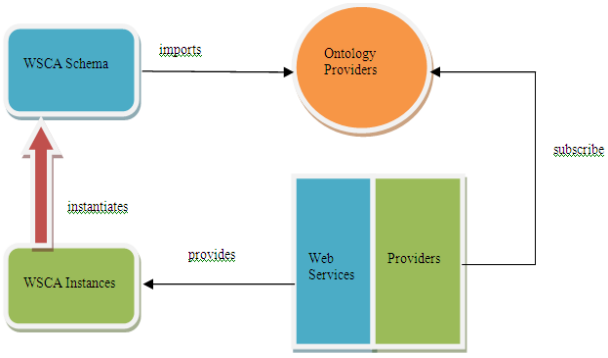

Fig1: An overview of WSCA

The lifecycle of a WSCA is a series of stages through which it passes from its inception to its termination. There are four phases in a WSCA lifecycle: initiation, composition, service-orchestration and termination. The initiation phase is the first stage, where the WSCA is described at a high level. It is initiated when the owner of the WSCA establishes a need for a business objective.

The composition phase deals with integrating the selected Web services. After this, the selected Web services are orchestrated to provide the value-added service. The termination phase occurs when the owner of the WSCA decides that the execution of the WSCA is no longer required.To materialize the concept of WSCA, the WSCA must automatically adapt to its dynamic environment, i.e., to deal with changes during its lifetime. Because of the dynamic nature of Web service infrastructure, changes should be considered as the rule and managed in a structured and systematic way. Changes may be introduced by the occurrence of new market interests, new business regulation, or underlying service availability. Such changes require a corresponding modification of the WSCA structure with respect to the functionality it provides and the performance it delivers.

Once a change occurs, a WSCA must react in a reasonable time and realign itself to deal with the change. This alignment must be performed in an automatic manner considering the frequent occurrence of the changes to a WSCA. By doing this, the WSCA can not only deal with unanticipated changes to the underlying services and infrastructure, but also maximize its market value, optimize functionality outsourcing, and maintain competitiveness.

4. PROBLEM FORMULATION

The problem with the existing system of legal decision making process is that:

- There is no universal coalition body to unite the various modules of law in private sector like:

- A Public Interface to query for a case
- Online allotment of Advocates to users
- Judgments
- Union of India (to export Central Acts and Amendments)
- State Acts (to export State Acts and Amendments)

- The whole system of legal enactment starting from user query, allotment of Advocate, and referring judgments is not under a single roof.
- Due to inadequacy of getting the latest and updated Acts or Amendments, sometimes old acts are cited which later cause problems when the case is taken up in courts. This is a cause of embarrassment to the lawyer.
- In the existing System there is no spontaneous notification to the legal bodies regarding a change in the Acts. They themselves have to search for it on internet or find them in the latest issues. The problem is that they are not notified about the change in Acts automatically.

5. PROPOSED SOLUTION

The need is to have an automated Legal System in India, at least at the private level. And this automation should be done with the most contemporary and best suited group of technologies [5][6]. The problem uncovers the major issue of incapability of notification of changed Acts or Amendments to the related bodies.

The proposed WSCA model namely LegalCosmos functions to provide a comprehensive Legal Solution that outsources the functionality from various service providers, such as WS_IndianUnionAmmendments, WS_StateActAmmendments, WS_PublicInterface, WS_AdvocateAssociation, WS_Judgments (depicted in Figure 2).

The user in need of a legal solution makes use of the web service WS_PublicInterface and submits his, here accused, details and the details of the FIR filed by the complainant and forwards the FIR details like Case Crime No, Name, crime, date etc., to the WSCA which allots a Lawyer for the case with the help of WS_AdvocatesAssociation. An interested user (may be lawyer or a normal user) can find the details of the judgments through the WS_Judgments service. When a judgment is made the WS_Judgment web service provides the user with the copy of the order of the judgment.

When these services are combined together as a WSCA, the WSCA will invoke the services on behalf of the user. There may potentially be some dependency relationship between them. These dependencies determine the composition of the services [7]. In the case of the LegalCosmos WSCA, users do not necessarily need to provide the information for each service. The input of some services can be derived from the dependency relationship. Like in WS_AdvocateAssociation selected advocate details are directly passed to the WS-Judgments.

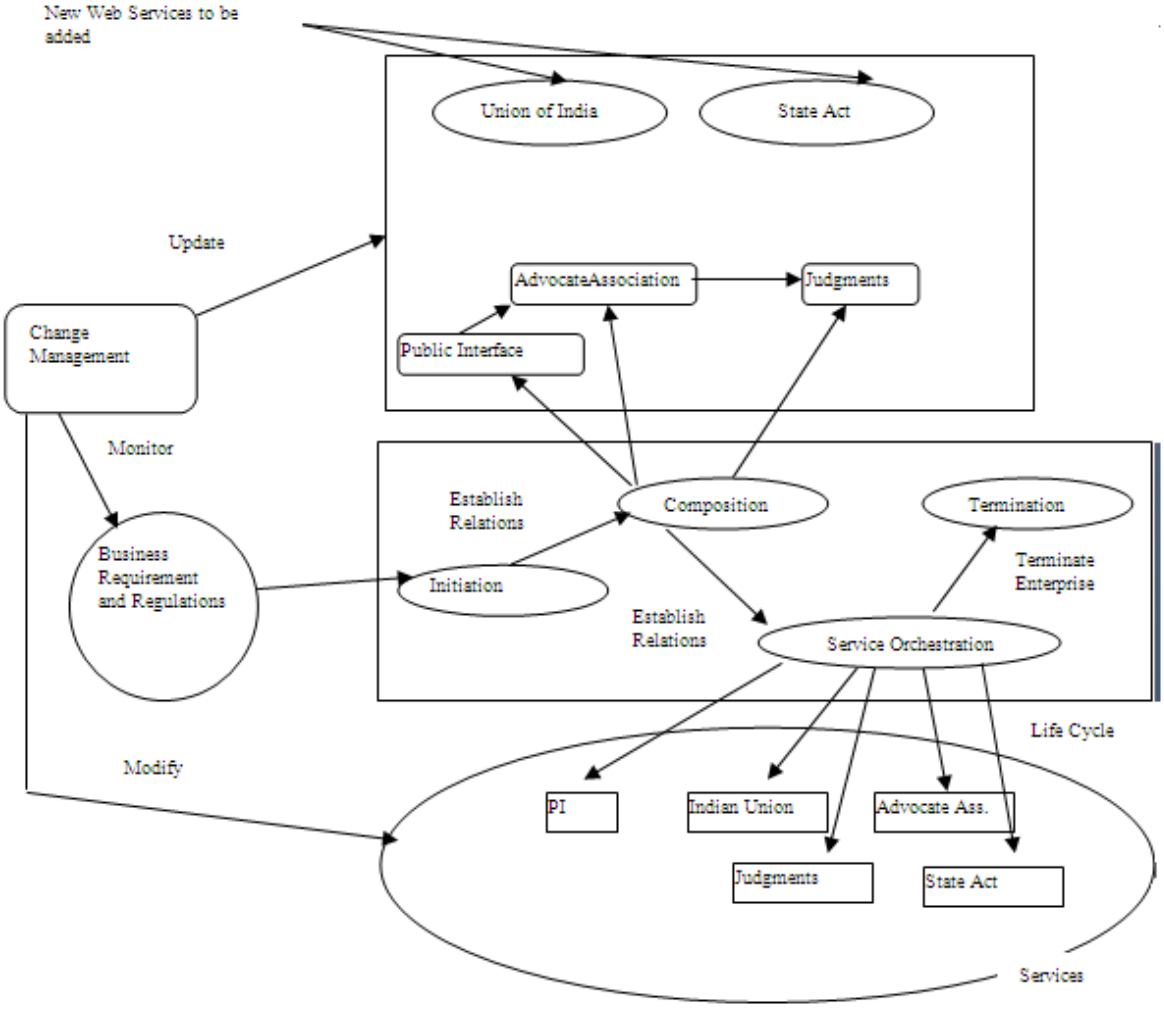

Fig. 2: Service Providers in WSCA

6. WSCA SCHEMA USING FINITE STATE MACHINE

Definition: FSM is a mathematical model of computation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of *states*. The machine is in only one state at a time; the state it is in at any given time is called the *current state*. It can change from one state to another when initiated by a triggering event or condition; this is called a *transition*. A particular FSM is defined by a list of its states, and the triggering condition for each transition.

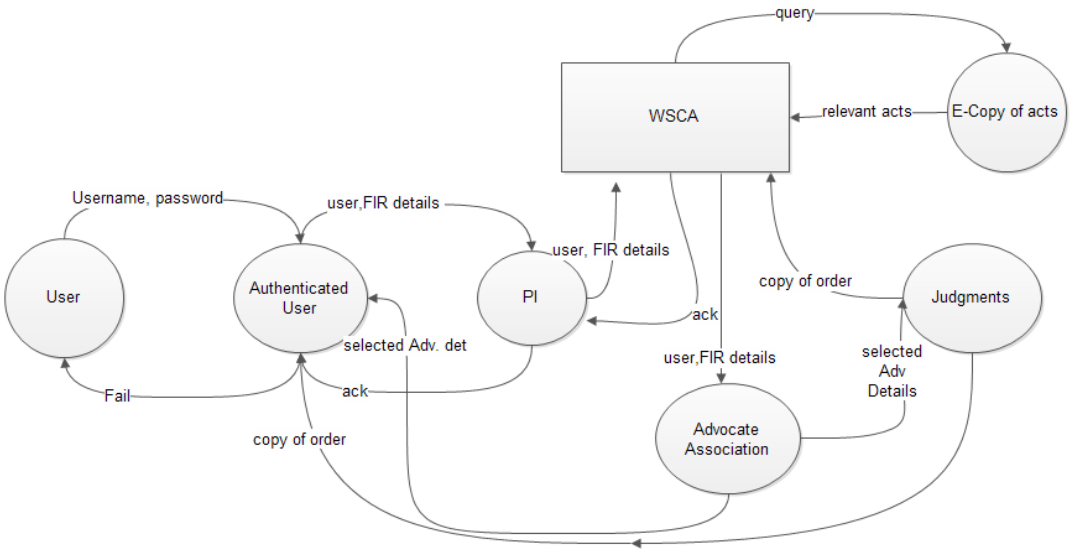

Fig.3 WSCA Schema (A simple XXM)

7. CHANGE MANAGEMENT IN LEGALCOSMOS

The change management in the proposed LegalCosmos WSCA primarily deals with the change in Acts in the form of Amendments. This change origins from the addition of two web services provided by the "Union of India"[8] that produces amendments at the central level and the "State Acts" that produces amendments at the State level. Since these changes are initiated by the outsourced service providers and incorporate an alteration in web service. They form the part of bottom up changes.

The bottom-up approach[2] for managing changes is highly dependent on the services that compose the WSCA. Therefore, it is necessary to first define the changes that occur to Web services, and then map them onto the business level. These changes include **Service changes** that occur at the service level and **WSCA changes**[9] that are executed at the business level. Finally, we provide rules for mapping triggering changes to their respective reactive changes.

In our work, we assume that service changes occur asynchronously. For instance, the WS_PublicInterface service may not change its data types while the service level change of unavailability is being managed. Another assumption we make is that the service is associated with a set of states. We associate each change with a transition between two states: precondition and postcondition. For example, a precondition for PI's unavailability is that it was previously available and the postcondition is that it has become unavailable. Service level changes and their respective preconditions and postconditions will later be used to model changes using FSM and XXM.

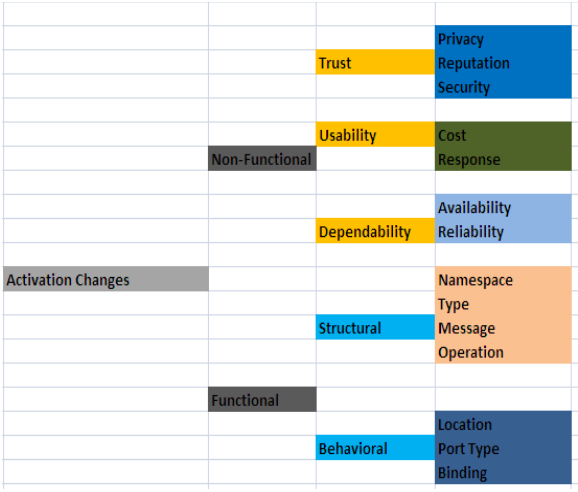

Fig. 4 Service/Activation Changes

7.1 Non-Functional Changes

Non-functional changes represent the dependability, usability, and trust associated with a member service. This information may be stored by a third party, the WSCA, or member services. We assume the information is stored as attributes that are maintained by an independent third party service provider. Changes to the trustworthiness of a Web service relate to the security, reputation, and privacy of a Web service. Changes in usability are dependent on the responsiveness and service cost. Finally, changes to dependability are associated with the availability and reliability of the Web service. Changes to service dependability are binary, because they represent two distinct states. For example, a service may either be available or unavailable. Alternatively, changes to service trust and usability are non-binary. For instance, service cost may assume more than two values during a member service's lifetime.

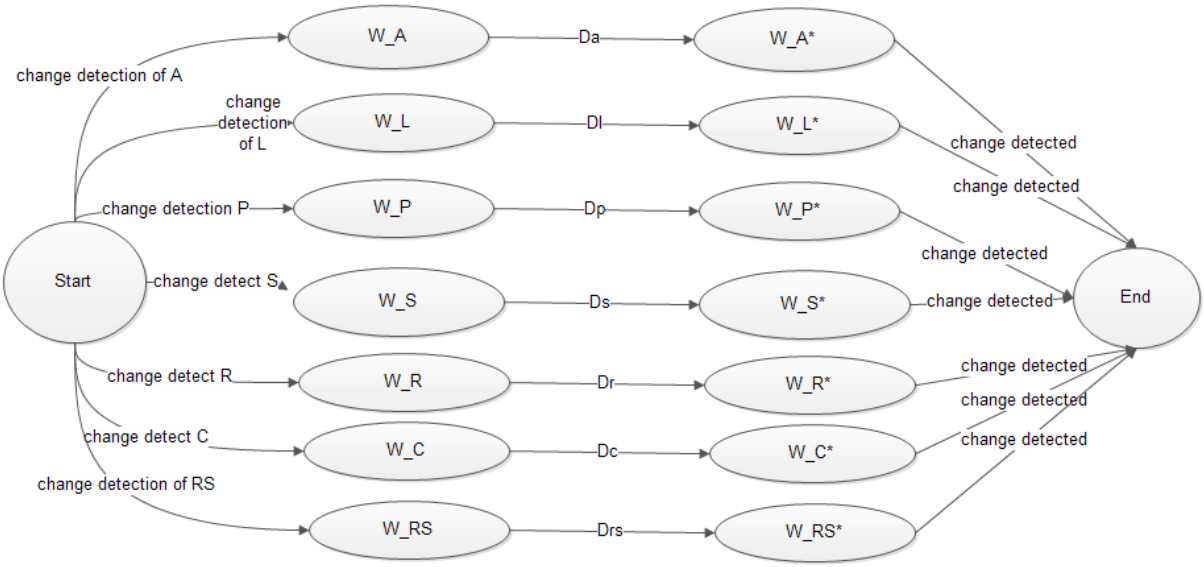

Fig 5 Non-Functional Changes through FSM

Therefore, we assume that a threshold is set and maintained by the WSCA. This threshold declares the minimum and maximum intervals of a value accepted by the WSCA. For example, the WSCA has the threshold of minimum $5 and maximum $10 for any judgment service cost. Every time a change occurs to the cost of a member judgment service, it is compared with the threshold. Only if the change exceeds the threshold, we consider that a triggering change has occurred. Note that the changes we have defined, such as changes to availability, are applicable to member services only. Once a member service is replaced, it is no longer part of the change management mechanism. For example, the e-Acts member service EA1 may become unavailable, and prompt the WSCA LegalCosmos to select an alternate e-Acts service EA2. After some time, EA1 may become available. However, since it is no longer a part of the LegalCosmos, the LegalCosmos WSCA does not manage the change in EA1. Table 1 summarizes the non-functional changes in Web services.

Table 1 Non-Functional Changes

Change	PreCondition of Attribute	Transition	PostCondition of Attribute
ChangeAvailability	W_S	Da	W_S*
ChangeReliability	W_L	Dl	W_L*
ChangePrivacy	W_P	Dp	W_P*
ChangeSecurity	W_S	Ds	W_S*
ChangeReputation	W_R	Dr	W_R*
ChangeCost	W_C	Dc	W_C*
ChangeResponsiveness	W_RS	Drs	W_RS*

7.2 Functional Changes

Unlike non-functional changes, which are based on attributes, functional changes deal with changes to a service's WSDL description [2]. We represent functional changes as a combined execution of a remove followed by an add. We further classify functional changes into structural and behavioral changes (Figure 6). Structural changes refer to the operational aspects of a Web service. For example, a structural change in a PI service can be caused by changing the operations offered to a user. Changes to the behavior of a Web service are indicated by changing its interaction with external entities. Functional changes to a member Web service occur when its WSDL description is modified. We assume these changes are initiated by the service provider.

Table 2 Functional Changes

Change	PreCondition	Transition	PostCondition	Change in LegalCosmos
removeNamaspace	**W_N**	**Dn-**	**W_N-**	**Changes to namespace occur if a version of existing framework is upgraded**
addNamespace	**W_N**	**Dn+**	**W_N+**	**New namespace is added**
removeType	**W_T**	**Dt-**	**W_T-**	

addType	**W_T**	**Dt+**	**W_T+**	**A data type may be removed or added from the list of service parameters**
removeMessage	**W_M**	**Dm-**	**W_M-**	
addMessage	**W_M**	**Dm+**	**W_M+**	
removeOperation	**W_O**	**Do-**	**W_O-**	**The WSCA removes the operation of static reference to unchanged Legal Acts**
addOperation	**W_O**	**Do+**	**W_O+**	**The WSCA adds the operation of dynamic linking of Acts (amendments)**
removePortType	**W_P**	**Dp-**	**W_P-**	**WSCA may add or remove ports**
addPortType	**W_P**	**Dp+**	**W_P+**	
removeBinding	**W_B**	**Db-**	**W_B-**	**The PI service (Public Interface) may decide to offer binding via SMTP in addition to HTTP**
addBinding	**W_B**	**Dp+**	**W_B+**	
removeLocation	**W_L**	**Dl-**	**W_L-**	**WSCA may decide to offer services from different locations**
addLocation	**W_L**	**Dl+**	**W_L+**	

7.3 XXM for Existing LegalCosmos WSCA, when Functional Changes are not introduced (before the removal of E-CopyofActs web service)

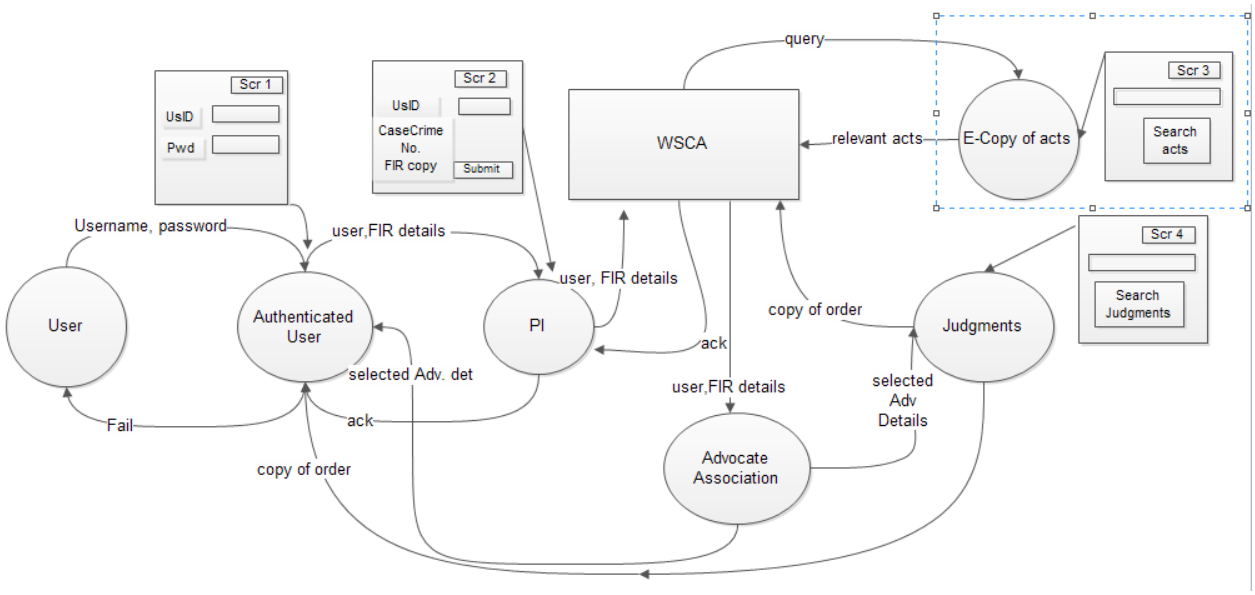

Fig 6 XXM for Existing LegalCosmos WSCA, when Functional Changes are not introduced (before the removal of E-CopyofActs web service)

The key functional changes involved here are:

- Removal of e-Acts web service (E-Copy of acts) shown in dashed rectangle. It has to be removed as it gives only immutable acts and does not involve latest amendments done by the Bare Acts. This causes problems to the WSCA in maintaining latest updates of the amendments and sometimes leads to false application of acts.
- A new web service called WS_UnionOfIndia is added to the WSCA which supplies the latest acts and amendments to WSCA. The Union of India is responsible for making and amending acts on the Central level.
- A new web service called WS_StateActs is added to the WSCA which supplies the latest acts and amendments to WSCA [10]. The State Acts is responsible for making and amending acts on the State level.

7.4 XXM for WSCA when Functional Changes are introduced (after the removal of E-CopyOfActs web service and addition

of two new web services, WS_UnionOfIndia and WS_StateActs)

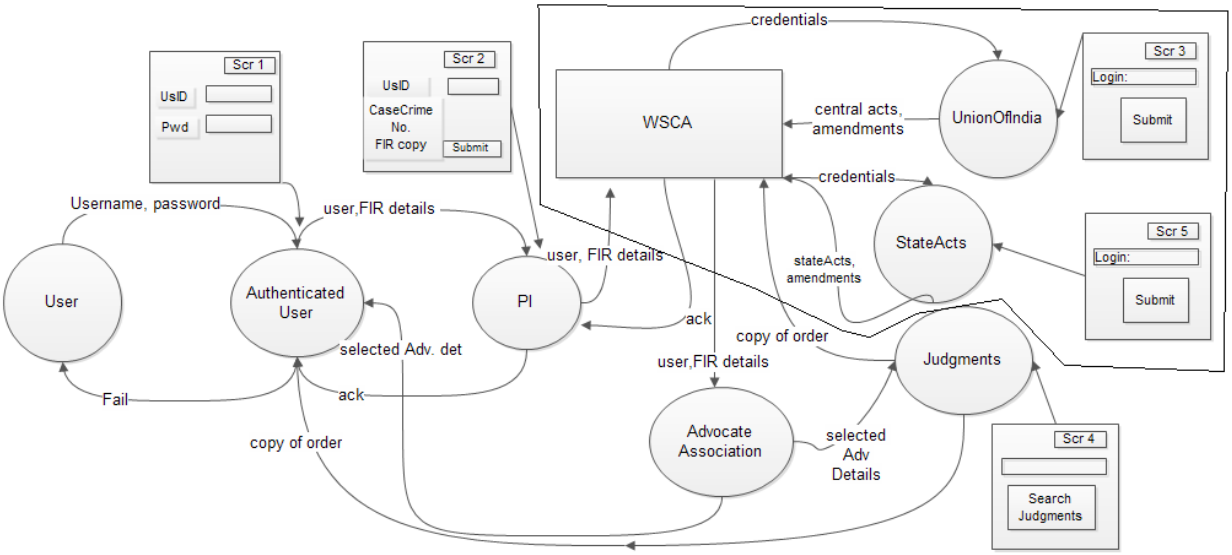

Fig. 7 XXM for WSCA when Functional Changes are introduced (after the removal of E-CopyOfActs web service and addition of two new web services, WS_UnionOfIndia and WS_StateActs

The above figure shows the two newly added web services namely WS_UnionOfIndia and WS_StateActs.

Table 3 WSCA Changes at business level

Change	PreCondition	Transition	PostCondition
removeMember	**WSCA_M**	**Dm-**	**WSCA_M-**
addMember	**WSCA_M**	**Dm+**	**WSCA_M+**
removeParameter	**WSCA_P**	**Dp-**	**WSCA_P-**
addParameter	**WSCA_P**	**Dp+**	**WSCA_P+**
removeInstance	**WSCA_I**	**Di-**	**WSCA_I-**
addInstance	**WSCA_I**	**Di+**	**WSCA_I+**
changeState	**WSCA_S**	**Ds**	**WSCA_S***
changeServiceInstance	**WSCA_SI**	**Dsi**	**WSCA_SI***
changeOrder	**WSCA_O**	**Do**	**WSCA_O***

7.5 WSCA Changes

Each service change will initiate a WSCA change at the business layer. WSCA changes may occur at the composition and service orchestration levels of a WSCA. For instance, a Dt change in the PI service may trigger inconsistencies in the WSCA composition, such as incorrect parameter data types. Alternatively, a Da change may disrupt WSCA service orchestration.

7.6 Mapping of Changes

A Mapping specifies how changes in one layer correspond to changes in another layer. These mappings must remain consistent in the presence of frequent changes. When a change occurs at the service level, the business layer must react to manage the changes. Triggering changes have a reactive impact on the business layer of the WSCA. Our approach of mapping changes is based on mapping rules. These rules are based on the service changes and their corresponding business level changes of WSCA. Some changes may have more than one rule associated with them. The rules are depicted in the matrix shown below. The algorithm for detecting the service changes that occur at the service level is shown Algorithm 1. It compares the old descriptions with the new descriptions and if a non-functional change is encountered it generates the FSM for the same. In case of functional changes it generates XXM for the transformation. Now these service level changes have to be propagated to the top (business) level WSCA. Here the Algorithm 2 maps the service level changes to the WSCA Business Level changes. This is done based on the rules as shown in the figure. It is pertinent to mention over here that the rules can vary organization wise. In the matrix below, the horizontal direction belongs to the non-functional and functional changes and the vertical direction consists of business level changes that are mapped through the functional and non-functional changes. Eg., Dt- Functional change is mapped on Dp- and Dt+ Functional Change is mapped on Dp+ business level change. Similarly Non-Functional changes

Dp,Dr,Ds,Dc,Drs,Ds,Dl are mapped on Ds business level change and so on.

Algorithm 1 Change Detection Algorithm
1: **ChangeDetection** (**Input:** oldDesc, newDesc)
2: **while** ne0wDesc **do**
3: Compare (oldDesc[Functional], newDesc[Functional])
4: **if** oldDesc[Functional]!= newDesc[Functional] **then**
5: GenerateXXM (FunctionalXXM)
6: **end if**
7: Compare (oldDesc[NonFunctional], newDesc[NonFunctional])
8: **if** oldDesc[NonFunctional] != newDesc[NonFunctional]
9: GenerateFSM(NonFunctionalFSM)
10: **end if**
11: **end if**
12: **end while**
13: ChangeReaction (FunctionalXXM, NonFunctionalFSM)

Propagation of changes to the Top Level WSCA
Algorithm 2 WSCA Changes Algorithm (WSCABuisnessLevelChanges)
1: **ChangeWSCA** (**Input:** FunctionalXXM, NonFunctionalFSM)
2: WSCABuisnessLevelChanges = φ
3: **while** FunctionalXXM **do**
4: Map (FunctionalXXM, WSCABuisnessLevelChanges)
5: **end while**
6: **while** (NonFunctionalFSM) **do**
7: Map (NonFunctionalFSM, WSCABuisnessLevelChanges)
8: **end while**

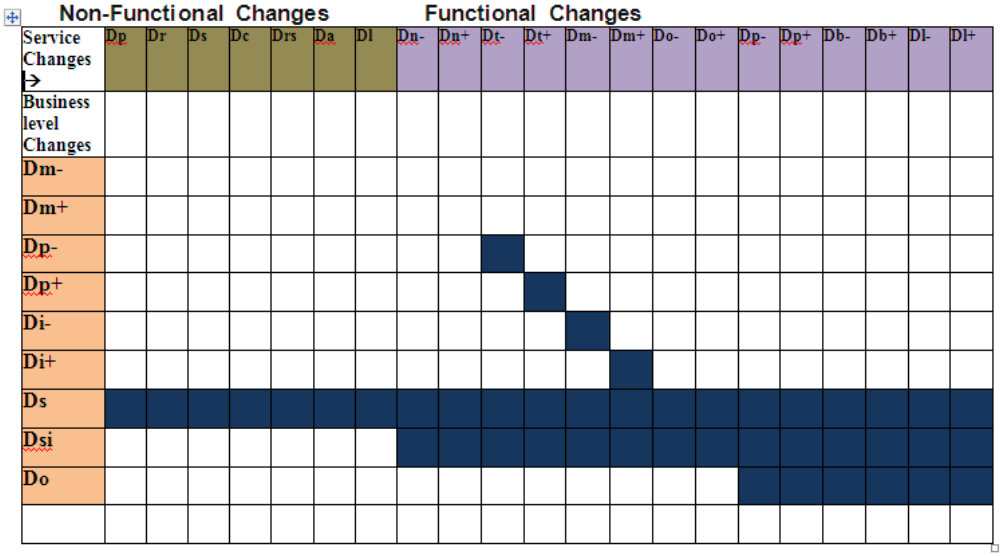

Fig 8 Mapping of Functional and Non-Functional Changes

8. REFERENCES

[1] Antikainen, J. and Pekkola, S. 2009. "Factors influencing the alignment of SOA development with business objectives", in: Proceedings of the 17th European Conference on Information Systems (ECIS). Verona, Italy.

[2] Fenn, J., Drakos, N., Andrews, W., Knox, R. E., Tully, J., and Ball, R. J. G., Hype Cycle for Emerging Technologies 2008, No. G00159496, Gartner, 2008.

[3] Xumin Liu ”Change Management for Semantic Web Services”, 2011, Springer

[4] Christopher Thomson, Mike Holcombe, Department of Computer Science, University of Sheffield, 211 Portobello Street, Sheffield S20 1EE, UK, “A Design Change Metric Derived From Extreme Xmachines”

[5] Erl, T., Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, Upper Saddle River, NJ, 2006.

[6] MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., and Metz, R., Reference Model for Service Oriented Architecture 1.0 (Committee Specification soa-rm-cs), Organization for the Advancement of Structured Information Standards, 2006.

[7] Tang, Q. C., and Cheng, H. K. "Optimal Strategies for a Monopoly Intermediary in the Supply Chain of Complementary Web Services," Journal of Management Information Systems, Volume 23, Number 3, 2007, pp. 275-307.fs

[8] Papazoglou, M. P., Web Services: Principles and Technology, Pearson, Harlow, England, 2008.

[9] Sholler, D., 2008 SOA User Survey: Adoption Trends and Characteristics, No. G00161125, Gartner, 2008.

[10] Gaurav Kant, V.K. Singh, M. Darbari., Legal Semantic Web- A Recommendation System, International Journal of Applied Information Systems (IJAIS) (Volume-7, Number-3)-May 2014.