

Automated Web Service Change Management (AWSCM - A Tool)

Animesh Chaturvedi

animesh[dot]chaturvedi88@gmail[dot]com

Indian Institute of Information Technology, Design and Manufacturing – Jabalpur

1. Introduction

→ *Automated Web Service Change Management (AWSCM)* constructs *Subset WSDL* based on change impact analysis of WSDL and WS code. AWSCM visualize and capture changes in the form of intermediate artifacts during impact analysis, figure 1 (a) (b).

→ We have proposed two WS regression testing approaches, namely, *Operationized Regression Testing of Web Service (ORTWS)* and *Parameterized Regression Testing of Web Service (PRTWS)* and prototyped them using AWSCM.

→ AWSCM contains two modules for regression testing, namely, ORTWS and PRTWS.

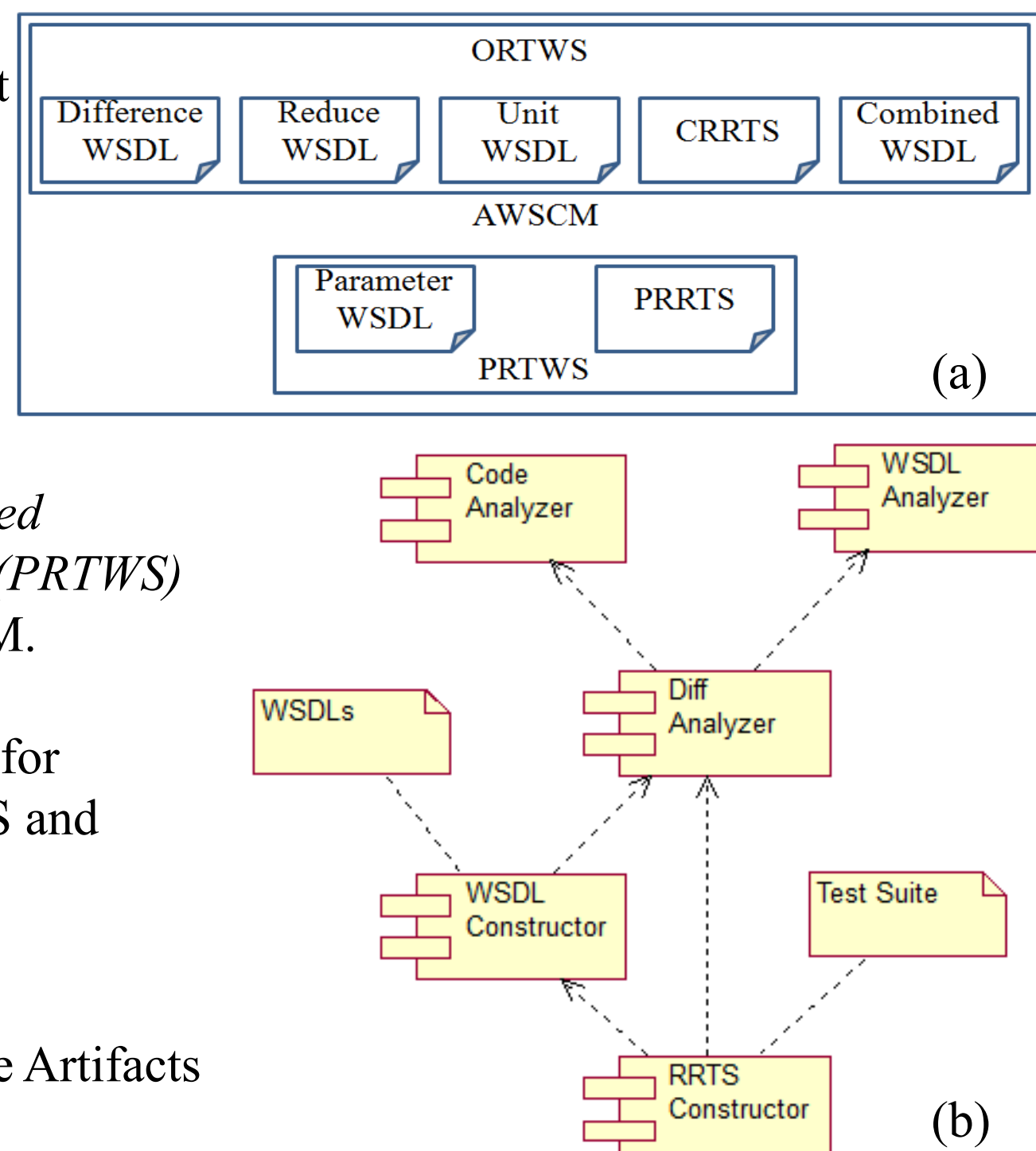


Fig. 1. AWSCM (a) Intermediate Artifacts (b) Components

2. AWSCM – A Tool

→ *ORTWS* module has three intermediate artifacts. Changes at WSDL are captured in *Difference WSDL (DWSDL)* and changes at code are captured in *Unit WSDL (UWSDL)*. Additionally, selective re-testing using *Reduced WSDL (RWSDL)*. These WSDLs combined to form *Combined WSDL (CWSDL)*, which is used for the construction of *Combined RRTS (CRRTS)* as shown in figure 2.

→ *PRTWS* module has the *Parameter WSDL (PWSDL)* as an intermediate artifact. *Parameter WSDL* is an extension of *Unit WSDL* for WS compositions and internal call interaction as shown in fig 3.

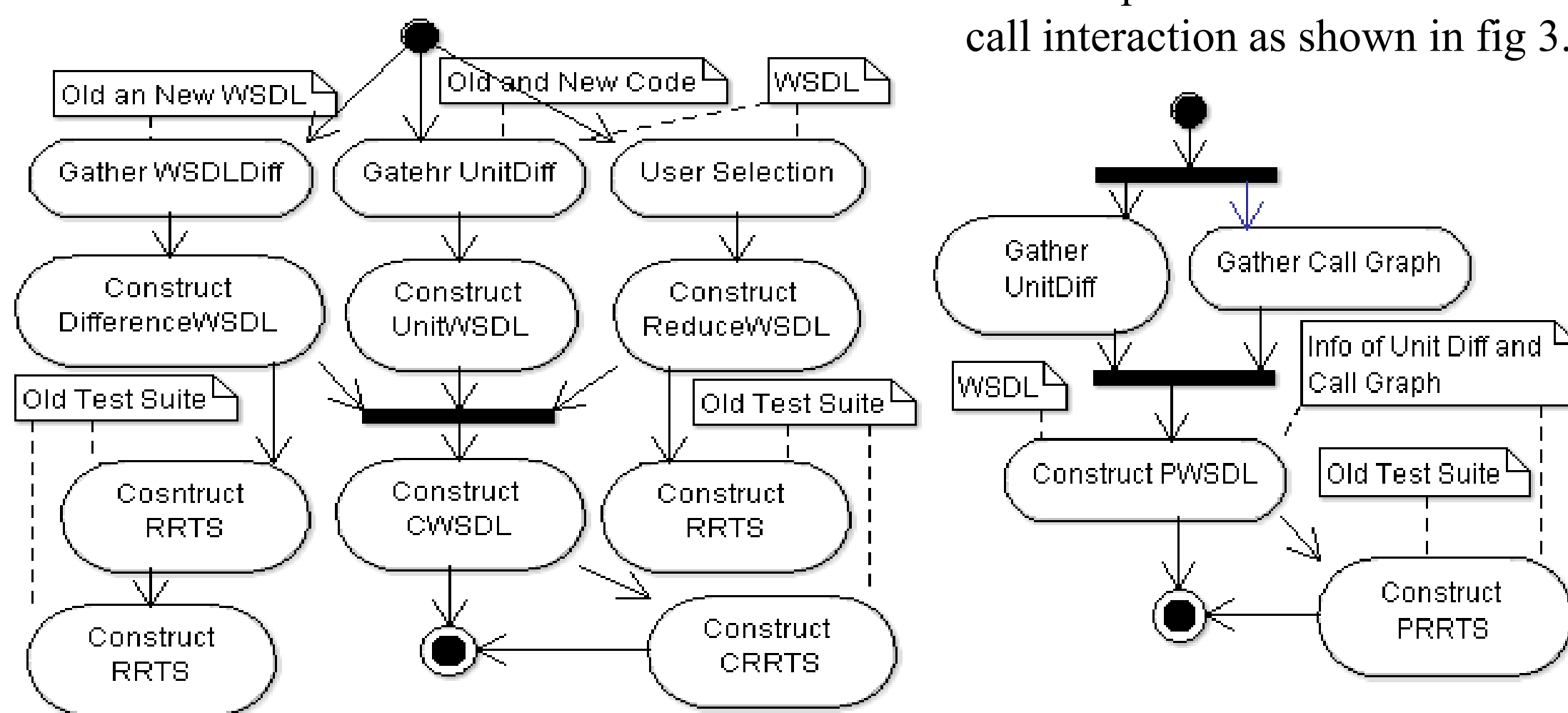


Fig. 3. PRTWS process.

Fig. 2. ORTWS process.

2 A. Algorithm inside ‘WSDL Constructor’

→ (*D/U/R/P*) WSDLs are together termed as **Subset WSDL (SWSDL)**. The *SWSDL* further helps in accessing the **Subset Service**. Subset Service can be accessed in three ways **Difference Service, Reduce or Selective Service and Combined Service**.

→ The *PWSDL* are constructed with the operations that are indirectly affected from changes such that their called methods or operations have undergone changes. Called operations can also be part of other WS i.e. this is also applicable in WS composition. Two inputs are required: ‘*newWSDL*’ and ‘*affectedOperations*’

→ The *DWSDL* algorithm takes two inputs: new and old WSDL. The algorithm gathers operations that have undergone WSDL changes to construct *DWSDL*.

→ The *RWSDL* algorithm takes any WSDL as input to gather its operations and then ask the user to select operations. *RWSDL* gives us access to the *Reduce or Selective Service* i.e. selective operations of a Service.

→ The *UWSDL* algorithm takes three inputs: new, old code of operations and new WSDL. *UWSDL* is constructed with the operations that have gone through changes at code. *PWSDL*, *DWSDL* and *UWSDL* give access to the *Difference Service* i.e. difference or changes between the two Services.

→ The *CWSDL* is constructed with the operations in one or more *Subset WSDLs* such that it contains only unique and non-redundant operations. *CWSDL* give access to the *Combined Service* i.e. combination of unique operations of Services.

2 B. Algorithm inside ‘RRTS Constructor’ Component

→ *Reduce Regression Test Suite (RRTS)* can be constructed from the old test suite only with the operations in a *SWSDLs*. RRTS is returned as output with test cases for only those operations of (*D/U/R/C/P*) WSDLs.

→ Suppose, WS = WS version 1, WS* = Modified WS version 2, T-old= Test Cases for the code of WS, T-new = Test Cases for the code of WS*, T* = Reduced test Cases, and *requiredOperations* = operations undergone changes. The procedure to construct RRTS is as follows

1. $T^* = T\text{-old}$: Reduced test case initially has old test cases.
2. The declaration of operation is inserted, deleted, modified, and unmodified with their test cases.
3. If operation is deleted then delete its corresponding test cases (td).
4. If operation is inserted then add test cases (ti) template.
5. If operation is modified then add test cases (tm) with selective test sequences/steps according to changes.
6. $T^* = T^* - tu$ delete all remaining unused test cases (tu) which are already executed in the testing of previous version of WS.
7. Return the T^* (reduced test case) evaluated to be $T^* = ti + tm - td - tu$.

3. Application of AWSCM

Table 1. Case studies of ORTWS & PRTWS on Web services

Web service Project	CWSDL for ORTWS			PRTWS	
	DWSDL testing	RWSDL testing	UWSDL testing	Dynamic Black Box Testing	Static White Box Testing
Eucalyptus	Y*	Y*	Y*	X	Y*
SaaS	Y	Y	Y	Y	Y
BookService	Y	Y	Y	Y	Y
Amazon WS	Different versions of WSDL is not available	Y*	Code is not available	Y*	Code is not available
Bible WS		Y		Y	
Currency Conversion WS		Y		Y	
Weather WS		Y		Y	

• *Threat to validity* because case studies is performed without the proper test data for Eucalyptus and AWS.

→ AWSCM is just a prototype, an actual realization can be done when *ORTWS* and *PRTWS* is added as feature in the standard WS testing tools.

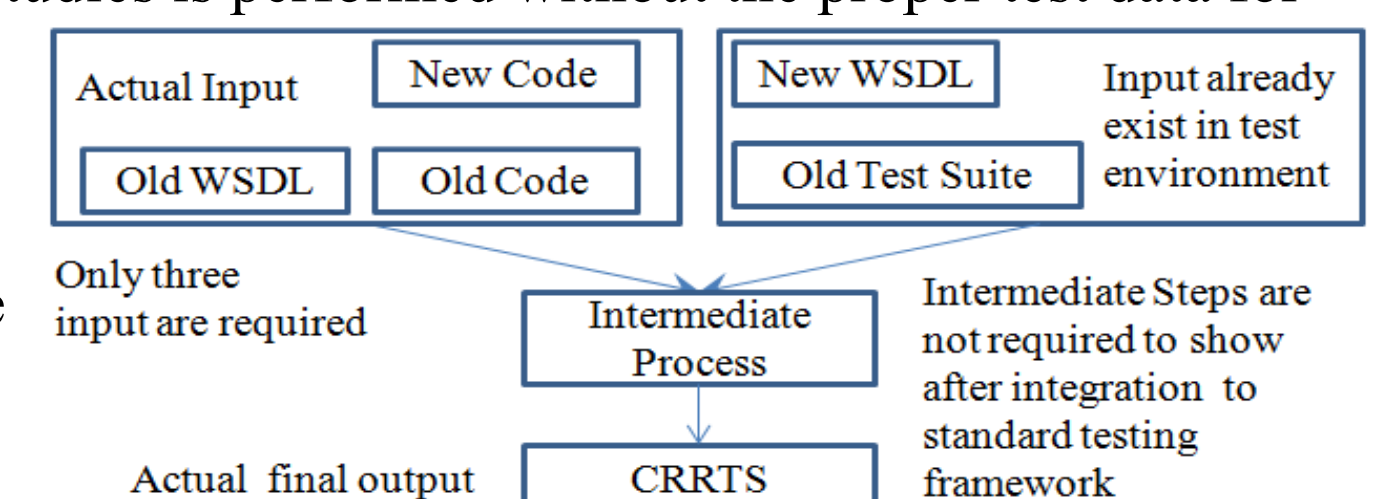


Fig. 4. AWSCM integration with SoapUI and JMeter.

4. Addition to AWSCM

→ The paper gives insight on computation of change impact as well as mapping them to their test cases.

→ Discussion on details of algorithm for the construction of reduce regression test suite.

→ AWSCM can also be useful for top down development of web services using the subset operations in WSDL.

→ Other Published Articles on AWSCM

1. Chaturvedi Animesh. "Change Impact Analysis Based Regression Testing of Web Services." *arXiv preprint arXiv:1408.1600* (2014).
2. C. Animesh and G. Atul, "A Tool Supported Approach to Perform Efficient Regression Testing of Web Service," 7th IEEE MESOCA 2013.
3. Chaturvedi Animesh, "Subset WSDL to access Subset Service for Analysis", 6th IEEE CloudCom 2014.
4. Chaturvedi Animesh, "Automated Web Service Change Management AWSCM - A Tool" 6th IEEE CloudCom 2014.
5. Chaturvedi Animesh, "Reducing Cost in Regression Testing of Web Service," CSI 6th CONSEG, 2012 on IEEE.