

Webgen Compiler Installation and Automation

Submitted By
Garima Joshi
13MCEC05



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Webgen Compiler Installation and Automation

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Garima Joshi

(13MCEC05)

Guided By

Prof. Zunnun Narmawala



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Certificate

This is to certify that the major project entitled ”**Webgen Compiler Installation and Automation**” submitted by **Garima Joshi (Roll No: 13MCEC05)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Prof. Zunnun Narmawala
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Vijay Ukani
Associate Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Garima Joshi**, Roll. No. **13MCEC05**, give undertaking that the Major Project entitled "**Webgen Compiler Installation and Automation**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by
Prof. Zunnun Narmawala
(Signature of Guide)

Acknowledgements

Firstly I sincerely thank to my external guide **Mr. Rohan Khanna**, and project manager **Mr. Ashu Talwar**, ST Microelectronics, Greater Noida for providing me necessary information regarding the project and I owe them lots of gratitude for having a profound impact on this report.

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Zunnun Narmawala**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I am grateful to my father **Mr. Ramesh Kumar Joshi** who provide guidance but do not deny one the freedom to experiment with ones ideas. I express my gratitude to my mother **Mrs. Mamta Joshi**, for her emotional support. My parents and teachers were a tremendous source of inspiration. I dedicate this work to them.

- **Garima Joshi**

13MCEC05

Abstract

Optimize and automate the manual memory generator installation process. Memory generators are used to generate various kinds of memories with various parameters like (Cut, pvt etc). The aim of the project is to automate the installation process and also provide the customer, transparent view of the installation process using Web interface. With the help of Webgen, cuts which is an area on the silicon chip where different components are being embedded and to generate them in a logical fashion, Webgen portal is used. Cuts are being generated which are being used by Memory team for memory generation specifications. Various compilers are being installed on this platform so that the library configs pertaining to various technologies can be incorporated and used by the customers. This project also contains the steps to optimize the process of memory installation. It also incorporate method to clean up the areas which are no longer used by the customers. Memory generator automation will be implemented in Java, Unix shell scripting and partly in perl technology.

—

Contents

Certificate	iii
Statement of Originality	iv
Acknowledgements	v
Abstract	vi
List of Figures	ix
List of Tables	1
1 Introduction	2
1.1 Introduction	2
1.2 AIM of the Project	2
1.3 Bottlenecks of the manual system	3
2 Literature Survey	4
2.1 Webgen Overview	4
2.2 Webgen Terminology	5
2.3 Types of Configurations	6
2.4 Webgen Architecture	6
2.5 Advantages of MVC architecture	7
2.6 Components of Webgen application	8
2.7 WebGen Hierarchy and Authentication	8
2.7.1 Webgen Hierarchy	9
2.7.2 Authentication	9
3 Method to Install the Configuration on compute-farm.	10
3.1 Log on Grenoble machine	10
3.1.1 Execute InstallProductsFromServer command	10
3.2 Create Library Configuration on Webgen	11
3.2.1 Expectation	11
3.2.2 Create the configuration in Webgen	11
3.3 Set Configuration Compatibility	13
3.4 Execute INTEGRATION Test	14
3.4.1 Expectations	14
3.4.2 Selecting or creating integration project	14
3.4.3 Selecting or creating integration library	15

3.5	Publish Configuration	16
3.5.1	Publish the configuration in Webgen	16
3.6	Publish the configuration in UPT	17
3.7	Create Authorization group	17
4	Current Methodology	18
4.1	Current work flow	18
4.2	Current V/s New Automation	19
5	Webservice	21
5.1	Intoduction	21
5.2	Types of web service	22
5.3	Comparison of SOAP and RESTful	23
5.4	Creation of XML Files	24
5.5	Parsing input XML file	24
5.5.1	DOM parser	25
5.5.2	Algorithm for parsing XML table	26
5.6	Task of automation (web service)	26
6	Algorithm used in Web service	29
6.1	Algorithm	29
6.2	Development Tools	30
6.2.1	Vnc (LINUX)	30
6.2.2	Perl	30
6.2.3	TCL	31
6.2.4	Bash	31
6.2.5	Csh	31
6.2.6	Postgre sql	31
6.2.7	Apache Tomcat Server	32
6.2.8	JSP, Servlet, Ajax	32
6.2.9	Eclipse IDE	32
7	Results	33
7.1	Execution time	33
7.2	Speed up	34
7.3	Comparative Result	35
8	Conclusion and Future Work	36
	References	37

List of Figures

2.1	Architecture of UPT	5
2.2	MVC Architecture	7
2.3	Interaction between Webgen components	8
3.1	Creation of Library Config	12
3.2	Compatibility of Config	13
3.3	Integration project	14
3.4	Integration Library	16
4.1	Current Work Flow	18
4.2	Current v/s Automated Work flow	19
5.1	XML Format	25
5.2	DOM Parser	25
5.3	Algorithm of Parsing	27
6.1	Algorithm used by web service	29
7.1	Comparative execution time	35

List of Tables

5.1	Comparison of SOAP and RESTful web services	23
7.1	Comparison of execution time (in min)	33
7.2	Time taken to create xml files (in min)	34

Chapter 1

Introduction

1.1 Introduction

Memory generator Installation system is being used to generate various kinds of memories (Memory cuts) with various parameters. The aim of the project is to automate the installation and also provide the customer the transparent view of the installation process using Web interface. This project also contains the steps to optimize the process of memory installation. Memory generator automation is implemented through a web service which is in JAVA . Inputs to this webservices will be fetched through Unix shell scripting (bash and csh) and partly through perl. We are doing Publication and Distribution of Memory generators which would be used by worldwide customers to generate designs and blueprints to fabricate silicon chips.

1.2 AIM of the Project

The aim of the project is to automate the manual installation process. Memory generator Installation system is being used to generate various kinds of memories (Memory cuts) with various parameters. Part of the responsibility of my team is to install memory generators on Web portal. After installation, these memory generators are available on web portal for the customers to generate their desired memory specification schematics. We are doing Publication and Distribution of Memory generators which would be used by

worldwide customers to generate designs and blueprints to fabricate silicon chips. Some parts of the process is partially automated and all other is still manual.

1.3 Bottlenecks of the manual system

In manual installation system we need to Perform all repetitive work manually. This process is very time consuming, but according to SLA agreement the team has to install the memory generator within 24 hours from requested time. Checking status of installation every 20-30 minutes. Checking the correct version and the state for every new primary. we need to check the correct version and primary from UPT, which is also a time consuming process. Creating versions manually if the version is not present, which is again a very time consuming and tedious job. Chances of error are more as human intervention is more as compared to automation.

Chapter 2

Literature Survey

2.1 Webgen Overview

Webgen is a web based application which is used to generate memory based projects and specifications according to the library configs in each project. Memory generators are used to generate various kinds of memories with various parameters. Memory generator Installation system is being used to generate various kinds of memories (Memory cuts) with various parameters. Various compilers are being installed on this platform so that the library configs pertaining to various technologies can be incorporated and used by the customers. The CAD-Installation team does the work of installing the memory generators using the webgen portal.

So, broadly speaking, webgen interface is:

- Webgen Interface is the web based user interface available to internal team and customers to generate their libraries.
- Consists of Webpages that are linked to the backbone (webgen Compute farm) to transfer information (inputs) and show output.
- Webgen is accessed through web browser to all the customers.

Webgen Compute Farm is a backbone of Webgen. It consist of cluster of computers on server system. It can be accesses through linux remote access tools like VNC.

2.2 Webgen Terminology

[1]

Config :- A complete package, which is a collection of products that is used to generate schematics for a specific type of memory chip fabrication.

Webgen :- Webgen is a web portal, on which memory configs(as defined above) are installed to enable company-wide usage for generation of schematics for user desired type and configurations.

Maturity :- Different maturity level have different set of validations and therefore confirmation about the validity of partial or all the desired generate-able schematics. Some of the used maturity are prelim, mat09 (ST internal views validated), mat10 (production) etc.

UPT :- UPT stands for **UNICAD Product Tracking** is a central repository where all IPs (standard cell, IO etc.) are present in one place. It doesnt mange version control system whereas Design Sync is a development server and it manages version control. That means it has all deployed IPs which are tested and using in production. If anyone needs a particular IP they can get it from central repository called UPT. There are some commands to work on UPT just like SQL queries.

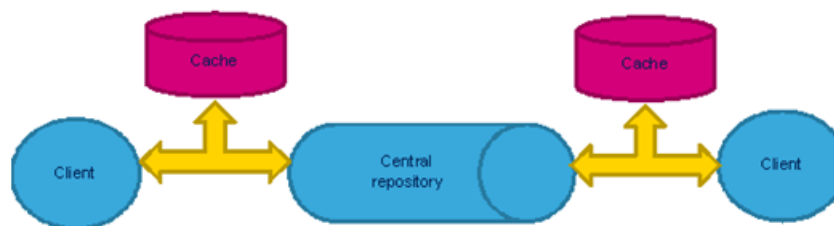


Figure 2.1: Architecture of UPT

2.3 Types of Configurations

Generation :- The configurations that are used to generate schematics for memory chips generation.

Bist :- BIST stands for Built-in Self Test. Cluster of different types of cuts and pvts of Generation configs contained in a project.

Verification :- Once the memory chips are designed, they are required to fulfill some checking criterion. The configurations defining these checking criterion are defined as verification configs.

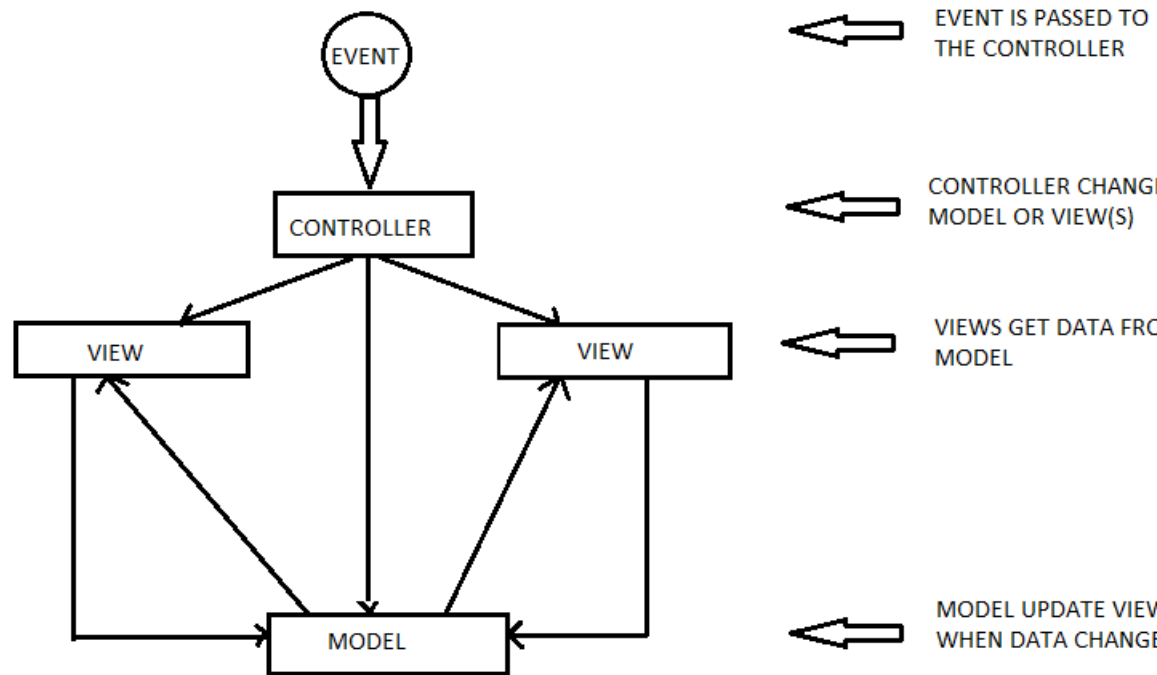
2.4 Webgen Architecture

Webgen is a web based portal, and it is based on MVC Architecture, and uses struts framework. MVC architecture has three pillars as shown in the below figure 2.2, they are **Model View and Controller**. It is a architecture design pattern that separates the all the three pillars from each other .No overlapping between business logic, presentation logic and data. [1]

Controller acts as an interface between View and Model. Controller intercepts all the incoming requests. Mainly servlets represent controller part in web application. It translates the user actions and user inputs into business method calls on model and selects appropriate view based on the user preferences and model state.

Model depicts the logic part of the application i.e. data. It is the structure of data in the application. It performs application specific operations on the data.

View represents the presentation of data i.e. UI(User Interface).For eg JSP, HTML. It also represents data in some form to the user in context of some business function.



[8]

Figure 2.2: MVC Architecture

2.5 Advantages of MVC architecture

There are basically three main components of any web application. In the traditional approach, the 3 components were clubbed in a single entity called class. Different JAVA components were used to club together to make an application.[8]

In MVC approach following are the advantages:-

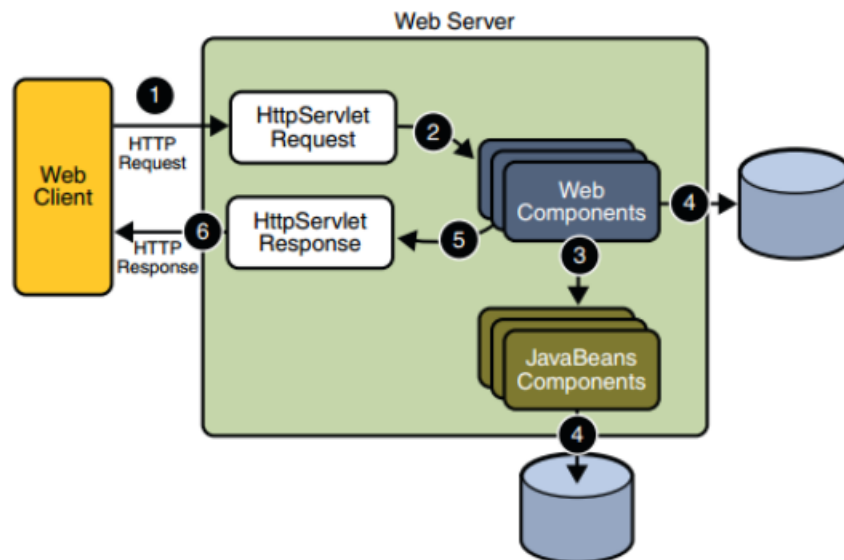
- By separating the Model and View(ie. Data representation and presentation): It means we are separating the behaviour and presentation of data of an application. This makes it easier for a developer to represent the same data with different user interface. It also facilitates adding new types of data presentation as technology develops.[8]
- By separating the Controller and View(ie. Application behavior from presentation): This permits runtime selection of appropriate views based on workflow, user preferences or model state.
- By separating Controller and Model(ie. Application behavior from data representation): Allows configurable mapping of user actions on the controller to application

functions on the model.

2.6 Components of Webgen application

[3] Typical web application consists of the following parts:

- Web Components(such as Servlets,JSP,HTML,XML)
- Web Container(Deployment Descriptor such as web.xml)
- Static Resource files(like images)
- Helper Classes(EJB)
- Libraries
- Interaction between different components



[3] Figure 2.3: Interaction between Webgen components

2.7 WebGen Hierarchy and Authentication

Webgen Interface is the web based user interface available to internal team and customers to generate their libraries. It consists of Webpages that are linked to the backbone to transfer information (inputs) and show output. This is usually accessed through web browser.

2.7.1 Webgen Hierarchy

Webgen portal is based on MVC architecture, in which Servlets, JSP, java files play a major role. Webgen has been deployed in three different phases:

- Prod Webgen: Used mainly by the customers who access the application to generate memories according to their given specifications. memory generators are installed on webgen prod, so that customer can generate their cuts.
- QA webgen: Used mainly by the testing team to test the application and provide frequent bugs detected by the users. They usually raise a ticket in concern of the development team so that the detected bug could be fixed.
- Dev Webgen: Used only by the development team for development activity, not live for user access.

2.7.2 Authentication

The authentication for webgen is provided in 2 steps

- By LDAP(Lightweight Directory Access Protocol) : Based on the users LDAP ID, he is signed in the portal. To access the LDAP service, the LDAP client first must authenticate itself to the service. That is, it must tell the LDAP server who is going to be accessing the data so that the server c

—

an decide what the client is allowed to see and do. If the client authenticates successfully to the LDAP server, then when the server subsequently receives a request from the client, it will check whether the client is allowed to perform the request. This process is called access control.

- By Internal access checker: ST has its internal access checker for all its applications. The authentic users credentials have been stored in ST Database.

Chapter 3

Method to Install the Configuration on compute-farm.

[2]

3.1 Log on Grenoble machine

After creation of the configuration in the UPT, following command will install the configuration on the web portal compute farm.

In Linux, login to Grenoble machine by using authorised username and passFile.

```
ssh -l <userName> <hostName>
```

3.1.1 Execute InstallProductsFromServer command

Install the products on Web portal compute-farm with creation of .productFile with the use of InstallProductsFromServer command.

InstallProductFromserver- It is an automation that would take care of the following actions :-

- Installing products on web portal.
- Declaring them in web portal interface (database entry),

```
 /<path of the script >/InstallproductsFromServer -user <username>  
 -password <encrypted password File> -technology <technology Name>  
 -config <configuration Name> -target webportal
```

- Making links for newly installed CAD-Tools which are missing from product repository and creating the list of products (.productfile) for this UPT Configuration.
- Defining new products on webgen interface.

3.2 Create Library Configuration on Webgen

3.2.1 Expectation

To create configuration on webgen following steps are expected.

- Webgen technology should be aligned with UPT.Primary and Webgen design flow should be aligned.
- UPT and Webgen maturity model must be aligned (to be checked).Add specific field in UPT for primary.
- Asic Kit: Mandatory (get from unigen),Flow Type: Mandatory (get from unigen) views filter: Optional (get from unigen) library filter: Mandatory (get from upt)

3.2.2 Create the configuration in Webgen

In Webgen,

- Click the link in the left "Library config",Choose Design Flow,purpose and enter the UPT configuration in name(excluding version and date information) then Click the link 'Search'.
- Result of search command will be either The list of configuration or no result.

[2]

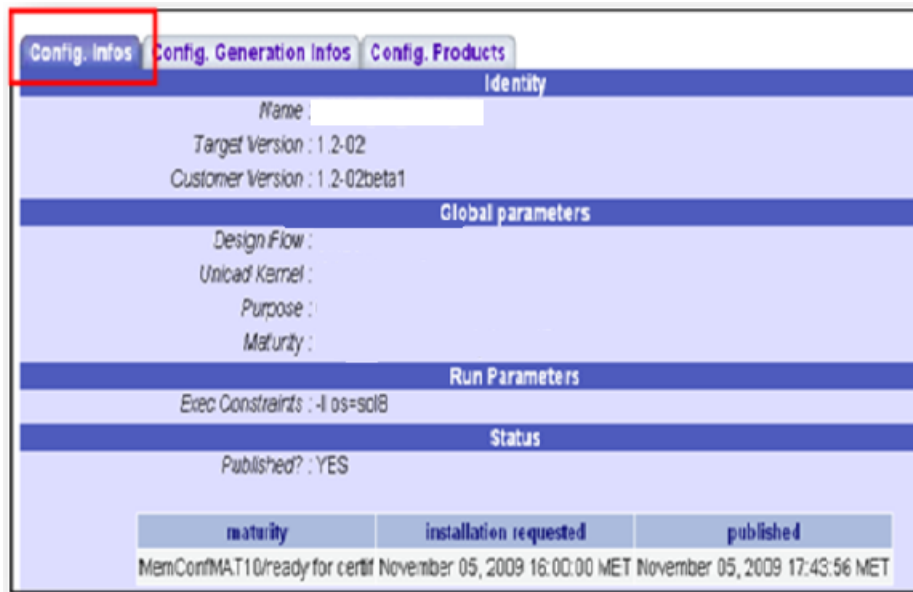


Figure 3.1: Creation of Library Config

- If the name of configuration already exists, choose the latest version and modify it by clicking 'EDIT'. Else create a new configuration by clicking 'NEW'.
- In config Infos tab, fill all the required information like Config Name ,Target Version , Design flow,Purpose, Monocad Thread(based on the version of last product of product file),Maturity and exec-constraints.

In Config. Generation Infos tab, you must indicate in the field:

In Config. Generation Infos tab, you must indicate in the field (For Generator and BIST)

- Identifier: the configuration name.Asic Kit ,Flow Type ,views filter ,library filter,Must Edit Before Generation needs to be entered.Get value of the "Generator-ParameterChanged" from the "CoordinatorOK" signature of the primary deliverable.It will be set to yes if "GeneratorParameterChanged =YES " else set No.
- In the window 'Config Products' you must put the list of the necessary products to .productfile by clicking .productfile editor.
- Copy the .productfile in ".productfile content" from the webgen setup area.Click 'SAVE'
- Wait for 2min and Click 'Refresh' till generation status becomes OK.

3.3 Set Configuration Compatibility

Following are the steps to find the compatibility of configuration with other versions.

- Click the link "config compat" to open the compatibilities lookup page. Choose the "design flow" according to technology of the PRIMARY. Enter the name of configuration from Webgen configuration. Choose source purpose and target purpose from UPT. Choose source customer version. Click "Search".

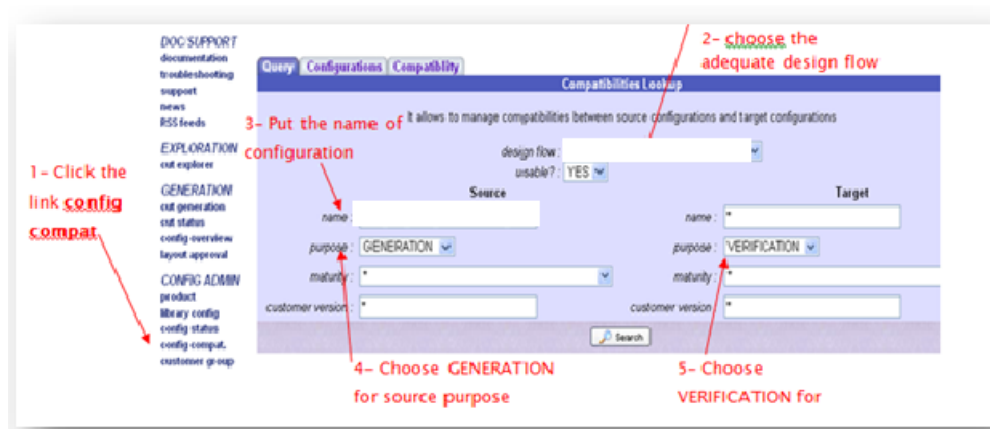


Figure 3.2: Compatibility of Config

[2]

- If the result is empty then go to next step. i.e Execute INTEGRATION Test. Else select proper configuration name and version as defined in Webgen Click "Check compatibility and then Edit.
- Open the primary deliverable (as received in the mail) in UPT and then click on signed in front of "Co-ordinator OK" signature and copy the verification platform Name. If the configuration being installed has purpose "Generation". and there exist a value then Set 'Yes' for this verification primary, 'No' for all others.
- If the configuration being installed has purpose "verification", then set compatibility to YES for all versions that are mentioned in comment of verification primary signature. For all other compilers available for the particular techno, set it to NO.

3.4 Execute INTEGRATION Test

3.4.1 Expectations

Project Format must be: TESTS INTEGRATION <configurationdesignflowname>

3.4.2 Selecting or creating integration project

- Click the link Cut Generation on webgen
- Select the Design Flow and Enter in name of project *TESTS INTEGRATION* and Click Search.If the result of search is not empty then check the format of the result it should be like "TESTS INTEGRATION <configurationdesignflowname>" then click on the project name.If the result is empty then Create the integration project by clicking "new". fill all the entries , Click "Finish" and click on the newly created project.

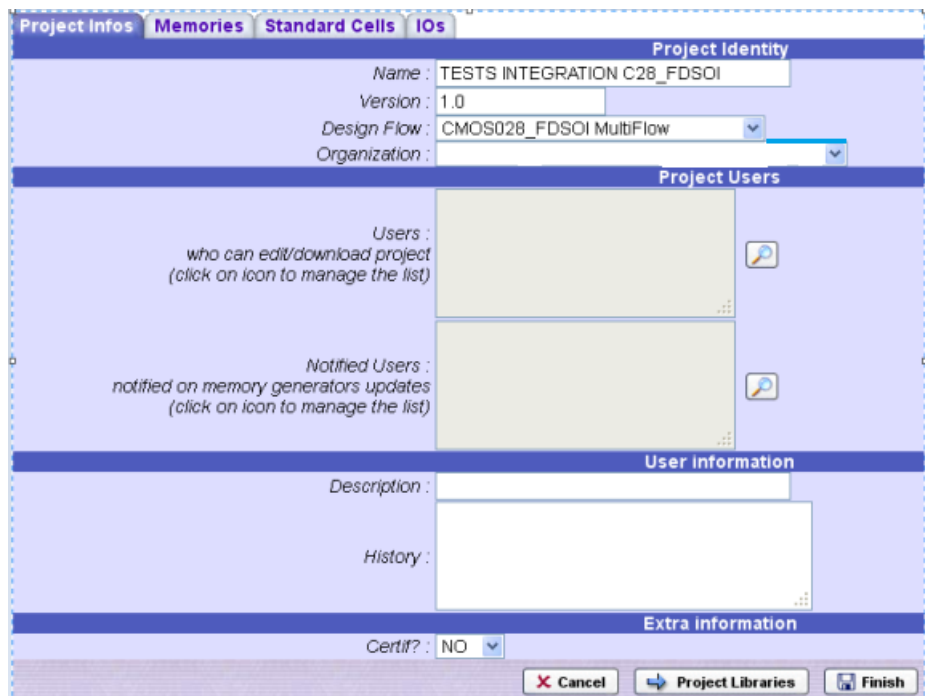


Figure 3.3: Integration project

[2]

- Go to memory tab(if primary is generation type or verification type) and if the primary is of Bist type the go to bist tab.

3.4.3 Selecting or creating integration library

For Generation

- If new configuration then add the library else click the existing library name and select appropriate library parameters , operating condintions ,View window etc.and save the library and project. To generate the library , select the library name and click on "Generate the selected libraries".

For Verification

- Run all the compatible libraries in the integration test by updating the verification environment for the "TESTS INTEGRATION <technodesign >" project.
- Click in the 'Generate selected libraries' to launch the Integration test, then Click "Generate WITHOUT layout".

[2]

For BIST

- In bist tab, we need to update version of library config for this Click on library config version , one window will open then select appropriate version (latest) then save
- Select your config name under "bist controller library name" and then a popup will open.No need to change "bist control parameter" and select "bist collar parameter".Select all lines and delete it. Add one line for each memory type. And set collar chain order for each in ascending order starting from 1. After that select "memory instances" and then add one row. If memory name exists then press ok else note down the name of memory type, do this for all the rows.For the memory type which we noted we need to mail that to corresponding person and wait for

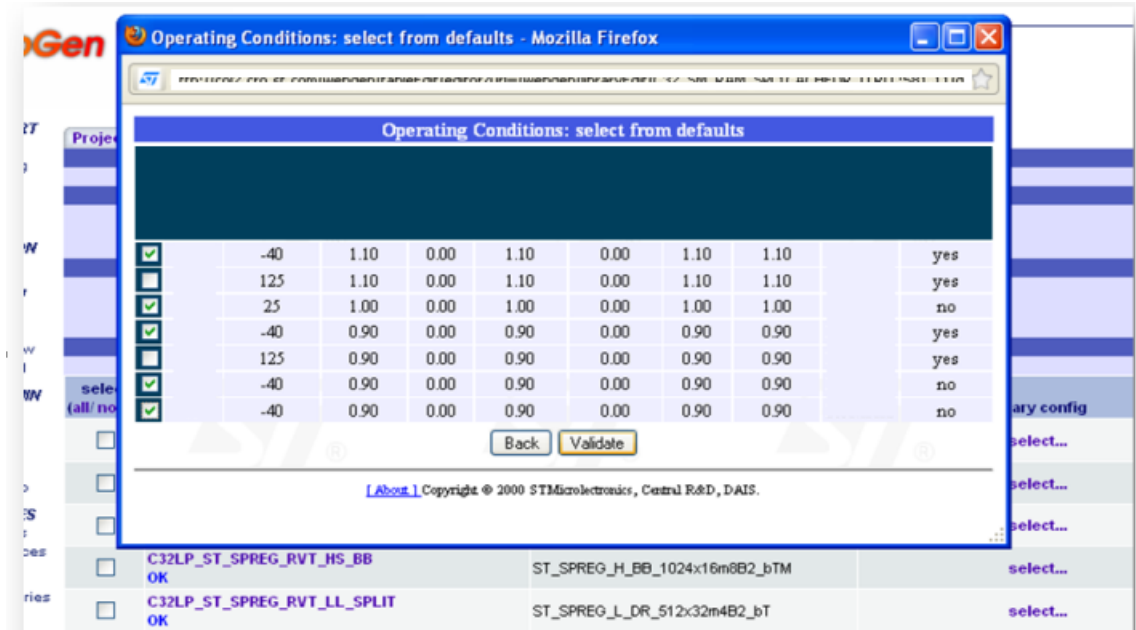


Figure 3.4: Integration Library

his reply and act accordingly. If no error then Select all rows and "compute cut name from selected lines". Then In view selection window, make sure all views are selected. Then ok and Save. Select the updated bist library and click on "generated selected". Then you will see "bist generating" and after some time press refresh till the process gets completed.

- Note :- You receive a notification mail from WEBGEN SERVICE. If the Webgen test is not OK, the result must be communicated to the Memory team via HD.

3.5 Publish Configuration

To publish the configuration in Webgen and UPT (if the Webgen test is OK):

3.5.1 Publish the configuration in Webgen

In Webgen,

- Click in the 'config status'then Select the configuration to publish.Click in the link 'Accept selected',Fill the installation request date with the date of the last of the 3 signatures for UPT PRIMARY.Fill the external comment with the comment in NewVersionHighlight field in the signature of CoordinatorOK in HTML format (Replace all carriage return/new line by break tag)then Click in the link 'Publish'.

3.6 Publish the configuration in UPT

- In UPT, configuration page,Switch the configuration to the state "ready for project" by signing the "readyforproject" approval .Indicates the "Webgen version[customer version]" in "Webgenversion" field.Search in Webgen "library config" the configuration and get the customer version.On the "library Config" page, this is found in the field "Customer Version" on Config. Infos tab. then save.

3.7 Create Authorization group

In authorization group Click on the app/Webgen domain then "Groups" panel .Search the UPT configuration name (only name)Result will be a list of configuration names.If current configuration name does NOT exists , Click [New group] Enter the configuration name then click on [Save]Return to the "Groups" page and click on your newly created group (configuration) click on "Permissions" panel. Click [New Permission].Choose appropriate class name for Class and action name for Action [2] Put the appropriate authentication condition and click on "SAVE".

Chapter 4

Current Methodology

4.1 Current work flow

Following figure depicts the methodology of webgen in current scenario.

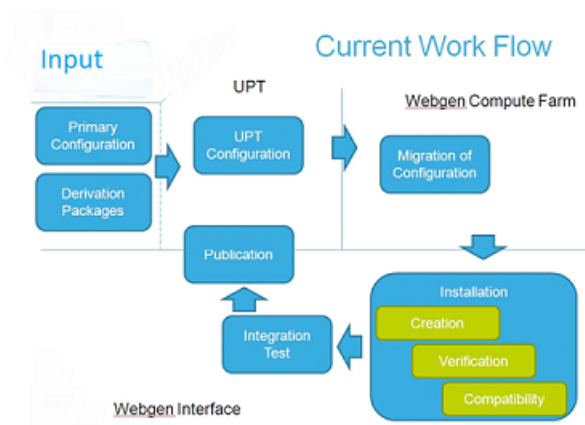


Figure 4.1: Current Work Flow

[4]

As demonstrated in the above figure 4.1 primary configuration and derivation packages are combined and are submitted to the UPT with "submitted" as the current state for intermediate storage purpose. These intermediate storage will finally be updated when the memory generators are installed on webgen. Then a mail is sent to the person, who is responsible for the installation process.

. In webgen compute farm all the products that are required for our memory generator

are checked one by one, whether they are present or not. If the product is present with

the required version then another copy of the same product will not be downloaded in this case. But if the product is not present in webgen farm area then that product will be downloaded with the required version. .Ucdprod is the file that holds all the required product along with its version. After downloading this step, Library config is created, then its compatibility is verified. To launch a library Integration test is performed. After successful generation of library it is published for the customers to use and fabricate their silicon chips. Right after publishing state of the memory generator is changed from submitted to ready for project.

4.2 Current V/s New Automation

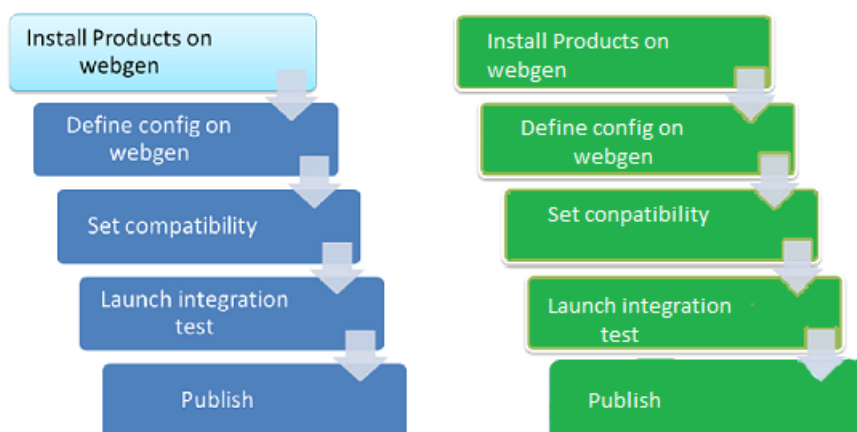


Figure 4.2: Current v/s Automated Work flow

[3]

Currently first step, that is InstallProducts on Webgen is partially automated. More precisely, it downloads all the products with appropriate version in the project area which are currently not present. This all happens during the execution of one script. Execution of the script is done manually. Rest all the steps are which are marked as blue are fully manual. These steps are to be performed by the person who is responsible for installation. But according to the web service, all the steps marked as green will be fully automated.

Automatically first module will be called by InstallProducts at the end when library config completes its task, it will return to the calling script. Then main calling script

will call compatibility web service. Same as library config it will return to the calling script after completing its task. Calling script will then call Integration tests web service , it will launch the integration test and return to the calling script. Calling script will now wait for the the integrating test to complete. In this step only library will be generated. When the library is generated system will get a notification that library is generated successfully or not. If its generated successfully without any error, then calling script will call the publish library web service at last. If the library generation is failed then it will return error message to the calling script.so no library will be punished, this whole process will be finished.In this case a ticket is raised to memory team for analysis of failure.

Chapter 5

Webservice

5.1 Introduction

A webservice is based on client server model which is basically used for communication. It is basically a kind of program or functionality written in some programming language which does the same task as the web GUI. Webservice is a method of communication between client and server over network. By using webservice a java application can communicate with php application. So it is interoperable in nature.^[7] A webservice is a collection of standards for communicating information between two application.

Concisely, webservice can be defined as :

- A kind of program or functionality written in some programming language which does the same task as the web graphical user interface.
- Based on client server model for communication between two machines over a network.
- is a software system for interoperable machine to machine communication.
- is a collection of standards or protocols for exchanging information between two devices or application.

5.2 Types of web service

Broadly, Web service is divided into 2 categories [7]

- **SOAP web services** :- SOAP stands for **S**imple **O**bject **A**ccess **P**rotocol. It is used for accessing web services and it is based on xml based model. It is both platform as well as language independent protocol. SOAP will give the flexibility to interact with other programming languages.

Advantage of SOAP is that it defines its own security which is called WS Security. SOAP web services gives freedom that a web service can be written in any programming language . Though it is platform independent, so the web service can be executed in any platform. Main disadvantage of using SOAP is it is a bit slow, consumes more bandwidth. It also define some standard which are mandatory for the web application to follow. SOAP can't use REST because it is a protocol. SOAP uses WSDL(Web Services Description Language) method to find the web service. WSDL serves as an interface between web service application

- **RESTful web services** :- Here REST stands for **R**epresentational **S**tate **T**ransfer. REST is not a protocol like SOAP, but it is more like a architectural style. Some of the advantages of RESTful web services is that it is faster as compared to SOAP.

It also uses less resource and consumes less bandwidth. It is both Language as well as programming independent. REST can use SOAP web services because it is a concept and can use any protocol like HTTP, SOAP. It can also use features of SOAP in the development of web service. Can be used with wide variety of formats like, HTML, plain text, XML etc.

5.3 Comparison of SOAP and RESTful

Following tables shows comparison in between SOAP web service and RESTful web service [9]

Property	SOAP	RESTful
Nature	protocol	Architectural style
Full form	Simple Object Access Protocol	REpresentational State Transfer
Usability	SOAP can't use REST	REST can use SOAP
Expose to business logic	Through services interfaces	Uses URI
JAVA API	JAX-WS	JAX-RS.
Follow standards	defines standards to be followed strictly	Does not define too much standards like SOAP.
Bandwidth	more	less
Resource	more	less
Security	defines its own security	inherits security measures from the underlying transport
Data format	XML data format only	Different data format such as Plain text, HTML, XML etc
Preferred	less	more

Table 5.1: Comparison of SOAP and RESTful web services

5.4 Creation of XML Files

Actual tags of Create config xml file are ConfigName, Customer Version, Upt Configuration Name, Design flow, Monocad thread, Maturity Name, Exec constraints, Asic kit, flow type, views filter, library filter, must edit, .productFile contents. Fetching of Config name , Customer version and UPT configuration name is done from the information present in the notification mail.Design flow is fetched by querying the database.Monocad thread and exec constraint are calculated from a products version present in product file. Asic kit, flow type, views filter, library filter are extracted from internal directory structure's setup file.must edit is fetched by querying the database. .productFile is the output file of InstallproductfromServer.

Same as tags of Config compatibility are config name, Source purpose,target purpose,Design flow, Verification platform name, and souce version of verification config.Souce purpose is fetched from upt query. To fetch target purpose mapping is provided.verification platform name is extracted bt web scrapping method (mechanize).

Further tags of Integration test is setupDb file which contains information about cuts , pvt, views and library parameters.

Tags of publish library are install Request date, external comments config name.In publish library external comments and install request date both are fetched through web scrapping technique (using mechanize in perl).

5.5 Parsing input XML file

Inputs are to be specified by the automation engineer. Usually the inputs are taken in the form of XML files. These xml files are output of some scripts written in bash, csh, perl. Perl is used only for web scrapping method. The input XML tags should contain all the inputs which a user has to enter while doing the same task on GUI.This xml is passed as an input to the web service. Web service will parse this xml with the help of some parser.

Following is the Demo format of XML file.

```
<CreationConfig>
  <Name>AB_CDE</Name>
  <PTConfigName>AB_CDE@1.0@21APR</PTConfigName>
  <Technology>AB</Technology>
  <Version>1.0</Version>
  <CustomerVersion>C1.0</CustomerVersion>
  <Maturity>MAT_10</Maturity>
  <SignedBy>GARIMA</SignedBy>
  <ProductFileContent>
    Asd v1
    Qas v2
    Mno v1.2
  </ProductFileContent>
</CreationConfig>
```

Figure 5.1: XML Format

In my project I have used DOM Parser.

5.5.1 DOM parser

DOM stands for Document object model. DOM defines an interface that enables programs (in which DOM is implemented) to parse and modify the structure and contents of XML documents. When DOM parser parses an XML document, the outcome of parsing is a tree like structure. [6]

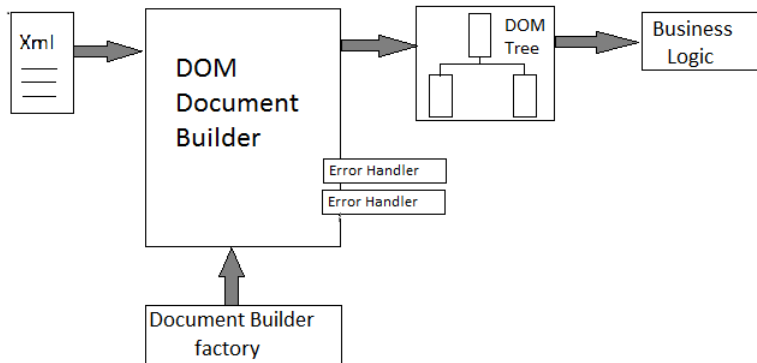


Figure 5.2: DOM Parser

[6]

Document.getDocumentElement() method will return the root node of the tree. Root node will be the first tag or the main node of the xml document.

Following are the methods that DOM Parser uses in traversing the tree.

- Node.getFirstChild() - This method will return the first child of a given Node.
- Node.getLastChild() - This method will return the last child of a given Node.
- Node.getNextSibling() - This method will return the next sibling of a given Node.
- Node.getPreviousSibling() - This method will return the previous sibling of a given Node.
- Node.getAttribute(attrName) - This method will return the attribute with the requested name.

5.5.2 Algorithm for parsing XML table

Following is the algorithm used in XML table to parse which is passed as an input to web service.

The whole logic by which we can parse the input xml file is written in fromDOM() which is a method in java class. There are some private variables in that java class which have getters and setters and as we go on parsing the XML file we put the values in those variables by setters. The values in the XML file are put into these private variables and when they are accessed by their getters we get the value of these variables. This function returns the class type variable, which is an EJB object.

5.6 Task of automation (web service)

This is the main task to be done for installations on webgen. The manual installations are very time consuming. Thus some automated system should be there so that this task could be completed in few seconds rather than 20-30 min. With automation, all the steps which the user has to repeat on the GUI are saved.

All the tasks which are performed on GUI are coded to be done at the back end using

- Take input which will be an XML file.
- Import XML-related packages.
 - `import org.w3c.dom.*;`
 - `import javax.xml.parsers.*;`
 - `import java.io.*;`
- Create a DocumentBuilder.
 - `DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();`
 - `DocumentBuilder builder = factory.newDocumentBuilder();`
- Create a Document from a file or stream
 - `StringBuilder xmlStringBuilder = new StringBuilder();`
 - `xmlStringBuilder.append("<?xml version='1.0'?> <class> </class>");`
 - `ByteArrayInputStream input = new
ByteArrayInputStream(xmlStringBuilder.toString().getBytes("UTF-8"));`
 - `Document doc = builder.parse(input);`
- Extract the root element
 - |
 - `Element root = document.getDocumentElement();`
- Examine attributes , sub-elements, next child until last child is traversed.

Figure 5.3: Algorithm of Parsing

some specified functions. Output of `fromDom()` function is captured in the servlet which is the main center of logic for this web service. We call the `fromDom` function of java class from servlet and as at the end we are doing the work to post the data from xml file to the database we use `doPost(HttpServletRequest request, HttpServletResponse response)` method of the servlet. When we call `fromDom` function of java class from the servlet, we get the class type object of the class where `fromDom` function is defined. That class has many private members which can be accessed only by getters and setters.

Now, as we have data from XML file, we can use that data to do further processing. In the processing we send the information got from xml file to another function that searches the corresponding config into the database. Search library config will give some result. We need to populate the fields with the value which we got from XML parsing. Further processing can be done by the servlet by updating the helpers of catalog class, which is a globally used class for making different functionalities work. we are doing the work to post the data from xml file to the database, for this we use `doPost(HttpServletRequest request,`

HttpResponse response) method of the servlet. Thus the work which needed to be done by a person using GUI in lets say 20 minutes, can be done in few second using this web service.

The work which was previously done in 20-30 minute are now completing in less than 2 minutes, with a single command. The command takes only input xml file, and try to run the servlet. Command to be executed is:

```
cat xmlFile.xml — lwp-request -m POST -c application/xml 'http://myapplication  
/webservicepath/webserviceName'
```

Chapter 6

Algorithm used in Web service

6.1 Algorithm

Following is the algorithm used in development of Java Web service [7]

```
Receive xml, which is passed as an input
Parse the xml to populate local variables
Search the Database for object with desired fields.
If the object exist then
    Load the EJB Object
    Update the EJB Object for the new desired values which are available in local variables.
    Save the EJB Object to the database.
else
    Create a new EJB object
    Initialize the values to desired data.
    Save EJB object to database.
Endif
If webserviceName != publish, then
    Call next webservice
Else
    Exit
Endif
```

Figure 6.1: Algorithm used by web service

Firstly, input which is an xml file is received. It is then parsed using DOM parser and populate local variables with the data of xml. According to the algorithm it then search the primary key in the database. If it finds an entry in the database, then it creates new

version and update that version with the values that are in local variables. Save the ejb object. If no result is found while searching the databases then create a new ejb object then populate all the fields with the values in local variable and save the ejb object in the database.

LOAD is defined in catalog which gets all associated data from various tables into EJB Object. Save is defined in catalog which saves all values in various associated relational tables. Enterprise JavaBeans (EJB) is an architecture for setting up program components, written in the Java programming language, that run in the server parts of a computer network that uses the client/server model. Main advantage of using EJB object is this it can control change at the server rather than having to update each individual computer with a client whenever a new program component is changed or added. Web services will be called sequentially. One web service will trigger the execution of second web service and second will trigger the execution of third web service and so on.

6.2 Development Tools

6.2.1 Vnc (LINUX)

VNC stands for **V**irtual **N**etwork **C**omputing. Vnc basically is graphical desktop sharing system. Vnc is Based on client server model. Therefore, many clients can connect to main server simultaneously. Variants of VNC are also available. VNC is widely used to provide technical support remotely or when someone has to access some files at different locations. RealVNC uses the RFB protocol. RFB Stands for **R**emote **f**rame**b**uffer. It is a simple protocol used to access graphical user interfaces remotely.

6.2.2 Perl

Perl is a high-level language. It is not compiler based language but, its a interpreter based general-purpose, language. Real Advantage of perl are it processes string and text files very faster. Perl is used for graphics programming. We have used Perl as a scripting language in our project i.e. compiler installation system.

6.2.3 TCL

Tcl stands for **T**ool **C**ommand **L**anguage is a very fast, dynamic and powerful language. It is easy to learn programming language capable of doing extensive variety of employments, It includes web applications and desktop applications, networking, testing and many more. It is an Open source and business-friendly language. Tcl is a mature yet evolving language . It is a cross platform, easily deployed and highly extensible language.

6.2.4 Bash

Bash stands for **B**ourne-**a**gain **s**hell. Bash as the name suggests it is a replacement of shell. Bash is a charge processor that commonly runs in a content window, where the client types command that cause activities. The larger part of Bourne shell scripts can be executed by Bash without change. Bash order language structure incorporates thoughts that are derive from the Korn shell (ksh) and the C shell (csh).

6.2.5 Csh

Csh stands for **C** shell had been composed solely in C, so the C shell's first goal was a command language that was all the more elaborately predictable with whatever is left of the framework. The keywords, the utilization of brackets and the C shell's implicit interpretation sentence structure and backing for shows were all emphatically impacted by C. Csh uses Ad Hoc Parser, which is the main drawback of csh.

6.2.6 Postgre sql

Database used is PostgreSQL which is an ORDBMS. ORDBMS stands for **O**bject-**R**elational **D**atabase **M**anagement **S**ystem. Its primary function is same as any other relational database management system like store the data securely, when user wishes it can retrieve or fetch the data. It can handle varied work load. For example single machine small applications to large multi-machine applications. Some of the advantages of Postgresql is it support ACID properties and uses multi version concurrency control mechanism to avoid locking control issues. No dirty reads problems. It handles more complex queries

than any other database. Postgresql also supports functions and stored procedure.

6.2.7 Apache Tomcat Server

Apache Tomcat is an open source web and servlet compartment grew by the Apache Software Foundation (ASF).Tomcat implements the Java Servlet and the JavaServer Pages (JSP) details from Sun Microsystems, and gives an "immaculate Java" HTTPweb server environment for Java code to run in.

6.2.8 JSP, Servlet, Ajax

JavaServer Pages (JSP) is an innovation that helps programming engineers make dynamically created web pages taking into account HTML, XML, or other document types. JSP is expected to show status of the every request and to process ask for through web interface. Ajax will upgrade every request status block rather than the whole page. Servlets accepts each input coming from web interface and process according to them To deploy and run JavaServer Pages, a perfect web server with a servlet holder, for example, Apache Tomcat is needed.

6.2.9 Eclipse IDE

Eclipse is a multi-language Integrated development environment (IDE) embodying a base workspace and an extensible module framework for customizing the environment. It is written mostly in Java. It can be utilized to create applications in Java and, by means of various plug-ins, other programming languages including C. C++.

Chapter 7

Results

7.1 Execution time

Following are the comparison of execution time of each module as well as whole process.

	Manual Process	With XML file	Automated service (Web Service)
Creation of Config	3.50	2.22	0.16
Compatibility of Config	1.43	1.05	0.11
Launch Integration Test	4.02	2.57	0.40
Publishing Generated Library	1.59	1.07	0.13
Whole process	12.04	7.31	1.20

Table 7.1: Comparison of execution time (in min)

Manual process means we are fetching the data manually from different locations, then updating then installing the memory generator on web portal. It is very time

consuming process. With XML file means, we need not to fetch the data from different locations rather just use XML table values to install the generator on web portal. By using Automated services that is webservice, it takes minimum time among the three methods. Webservice uses XML file as input and then does the rest of the processing in few seconds.

Following is the table that demonstrates the time that is required to create XML tables.

	XML file Creation
Creation of Config	0.09
Compatibility of Config	0.06
Launch Integration Test	0.07
Publishing Generated Library	0.07
Whole process	0.29

Table 7.2: Time taken to create XML files (in min)

7.2 Speed up

$$S = \frac{T_{new} - T_{old}}{T_{old}} \quad (7.1)$$

Here,

S is resultant Speedup

T_{old} is old Execution time

T_{new} is new Execution time

Following is the speed up of using web service with respect to Manual process.

$$S_{WM} = 88.95\% \quad (7.2)$$

Following is the speed up of using web service with respect to XML file process.

$$S_{WX} = 82.26\% \quad (7.3)$$

Following is the speed up of using xml File with respect to Manual process.

$$S_{WM} = 37.71\% \quad (7.4)$$

7.3 Comparative Result

Graph showing execution time of all Stages of Compiler installation in three modes :- Manual process, With xml and web service execution time.

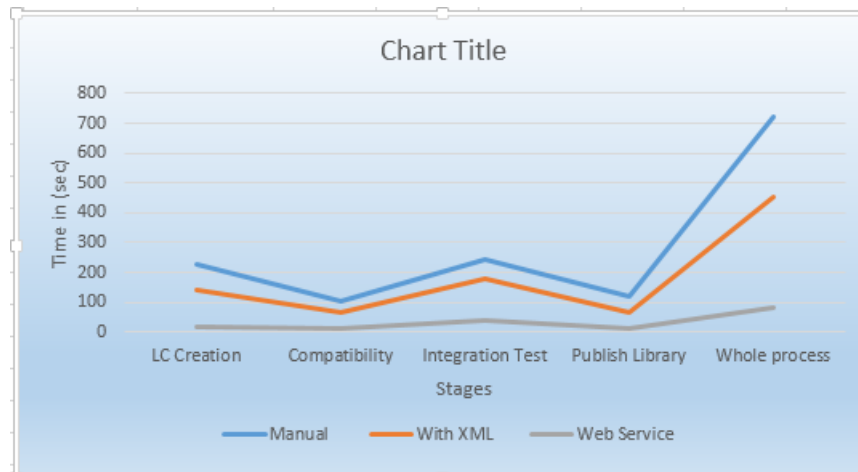


Figure 7.1: Comparative execution time

Chapter 8

Conclusion and Future Work

After the Development of a this web service for all the four main activities of web service, we need not to worry about the memory generator installation anymore. We will now be only responsible for changing the state on UPT databases as well as to define authentication rule (basically authentication).Efficiency of installation process has improved 20-30 minutes per installation to less than 2 min per installation. No manual intervention is required and manual resources can be utilized somewhere else. Previously it was a blocking factor if number of installations came more than 10 per day , but now its no longer a blocking factor.

Developed websevice is useful only for "GENERATION" type memory generators, not for BISTS and VERIFICATION.Future work of this project contains webservice of "BIST" and "VERIFICATION" type installation request.But 95 percentage request which we receive are of Generation type rest 4.5 percentage are of Bists type and 0.5 percentage are of Verification type.

References

- [1] ST proprietary document for webgen Architecture.
- [2] "ST Proprietary Document for memory generator Installation on webgen.
- [3] ST internal documents and Training manuals on Automation
- [4] ST internal powerpoint presentation for webgen Automation flow
- [5] Practical Programming in Tcl and Tk, Fourth edition, Ken Jones and Jeff Hobbs."
- [6] "<https://docs.oracle.com/javase/tutorial/jaxp/dom/readingXML.html>"
- [7] "<http://www.javatpoint.com/web-service-components>"
- [8] "<http://www.codeproject.com/Articles/620195/Learning-MVC-Part-Introduction-to-MVC-Architecture>"
- [9] "<http://www.javatpoint.com/soap-vs-rest-web-services>"