Automation of Distribution Management System

Submitted By

Prasang Misra

13MCEC14

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Automation of Distribution Management System

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Prasang Misra

(13MCEC14)

Guided By

Prof. Gaurang Raval

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 **”Automation of Distribution Management System”** submitted by **Prasang Misra (Roll No: 13MCEC14)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of 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. Gaurang Raval
Guide & Associate 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, **Prasang Misra**, Roll. No. **13MCEC14**, give undertaking that the Major Project entitled "**Automation of Distribution Management System**" 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: Nirma University,Ahmedabad

Endorsed by
Prof. Gaurang Raval
(Signature of Guide)

Acknowledgement

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Gaurang Raval**, Associate 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 would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Prasang Misra
13MCEC14

Certificate

I hereby declare that the work which is being presented in the Major Project titled **”Distribution Management System”** in partial fulfillment of the requirement for the award of degree of Master of Technology in Computer Science Engineering of Institute of Technology, Nirma University, Ahmedabad is an authentic record of my own work carried out under the supervision of Mr. Alok Mishra, External Guide and Mr. Sanjeev Bansal, Manager. The matter presented in this thesis has not been submitted in any other University/Institute for the award of my degree.

Date:

Place:

Prasang Misra

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Mr. Alok Mishra
Mentor and External Guide,
TRnD-CCDS,
STMicroelectronics,
Greater Noida (U.P)

Mr. Sanjiv Bansal
Manager,
TRnD-CCDS,
STMicroelectronics,
Greater Noida (U.P)

Abstract

This system manages deliveries and distributes IPs to different departments, according to customers requirements. The Distribution Management System not only processes the products according to their media but also manages each and every request using a web interface.

The distribution team in the infrastructure Services group is responsible for delivery of products to internal as well as external customers of ST. Requests for delivery of products come in large numbers every day. So we have automated the process. At the starting level, we thought of using Selenium. Selenium automates browsers. Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks. But lastly we automated the process using the Perl scripts and modules. There are two types of customers:
1.ST Internal Divisions
2.External partner
ST Internal divisions of designers can ask any type of request; it may be web download and transfer products in their repository. There is one more customer called as External Partner, this request made by external customers through internal departments in ST. Finally the delivery will be transfer to external customer through data transfer gateways like SIFT and SFT.
Distribution team takes care to process all type of requests. There are several types of methods, used to process different kind of customer Requests.

The TRnD department creates libraries which can be reused for present and future projects. Other divisions in ST or external customers who need these libraries for their projects make a request for these libraries on a web interface known as DRT (Distribution Request Tracking). Distribution team takes care to process all the DRT requests. There are several types of methods used to process different kind of DRT.

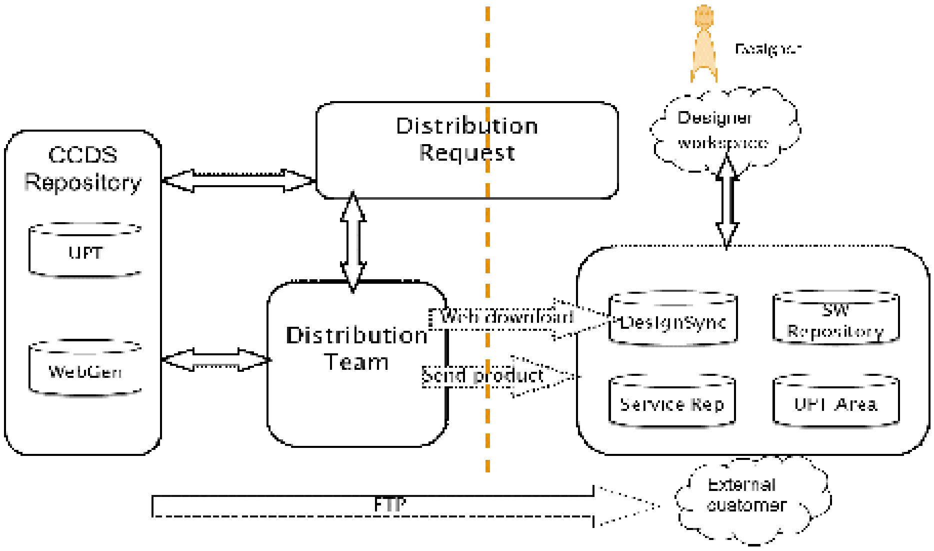


Figure 1: Distribution System Architecture

–

Contents

Certificate **iii**

Statement of Originality **iv**

Acknowledgement **v**

Certificate **vi**

Abstract **vii**

List of Figures **1**

1 Project Definition and Overview **3**
- 1.1 Project Definition. 3
- 1.2 Objective of Project 3
- 1.3 Terms in Definition. 4
  - 1.3.1 Automation 4
  - 1.3.2 Build 5

2 Literature Survey **6**
- 2.1 IP-Intellectual property 6
- 2.2 DRT 7
  - 2.2.1 Submitted: 8
  - 2.2.2 Assigned: 8
  - 2.2.3 Pending Approval: 8
  - 2.2.4 Pending Support: 10
  - 2.2.5 Approved: 10
  - 2.2.6 Processed: 10
- 2.3 Getting Approval 12
  - 2.3.1 Approval Mail 12
  - 2.3.2 UOLBE 12
- 2.4 Medias 13

3 Tools and Technogies used **17**
- 3.1 Setting up a fresh environment 17
- 3.2 UPT 18
- 3.3 Design Sync 18
- 3.4 LSF-Load Sharing Facility 20
- 3.5 Shell Script 21

3.6 PERL 21
3.7 JSP 21
3.8 Servlets and Ajax 22
3.9 Selenium 22

4 End-To-End Process **23**
4.1 Prototype Model 23
4.1.1 Prototype Approach 23
4.1.2 Evolutionary prototyping 25
4.1.3 Throwaway prototyping 26
4.2 Assumptions Made 27
4.3 Design and Implementation 27
4.3.1 Gathering Information 28
4.3.2 Distribution Tracking Page 29
4.3.3 Approval Management System 31
4.3.4 Distribution 33

5 Conclusion **35**

List of Figures

1 Distribution System Architecture viii

1.1 Software Production line. 4

2.1 Submitting a request 8
2.2 Request assigned 9
2.3 Request changed to pending approval 9
2.4 Penging support section 10
2.5 Approved 10
2.6 Approved 11
2.7 Processed 11
2.8 Approval Mail 12
2.9 Flow of DRT 13

3.1 Architecture of UPT 18

4.1 Prototype of process 25
4.2 Design of IP Distribution Management System 28
4.3 DRT Operator Tracker 30
4.4 DRT Operator Tracker 30
4.5 Approval Management System 32
4.6 Distribution Dashboard Approval frame 33
4.7 Distribution Dashboard Approval completed 33

Chapter 1

Project Definition and Overview

1.1 Project Definition.

Automation of the Distribution Management System is done to achieve and enhance features in the processing such as :

1. Efficiency
2. Reliability
3. Robustness
4. Fault Tolerance
5. High quality

1.2 Objective of Project

Basically, Software Production line or Software Develpoment life cycle as commonly known consists of following phases

1. Requirement analysis.
    - It totally relies on what the present scenaio is, and what is to be expected from the further version of the product.
2. Design.
    - Designing needs to be schematic and in a planned manner.

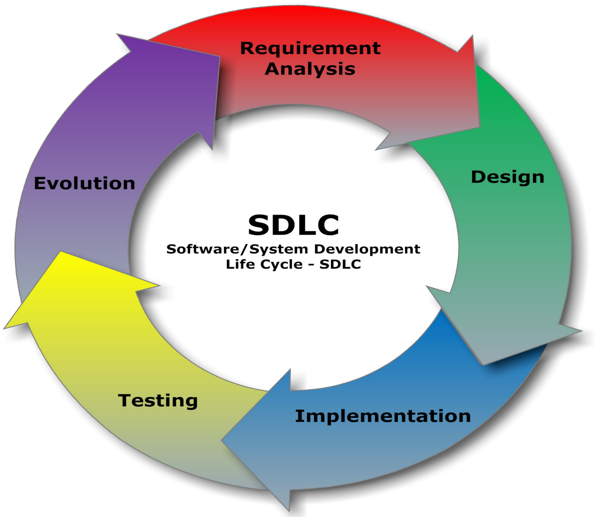


Figure 1.1: Software Production line.

3. Implementation.

    - Implementing those tasks. Coding phase begins from here. Mainly an implementation language is chosen to code the tasks.

4. Testing and QA

    - Automation plays a very important role in testing phase and during QA. Testing scripts and codes with all the possible test cases is an important part before any project can go live.

5. Evolution and Maintainence.

    - Once the code is devloped and tested in controlled environment, it is run in actual Production environment.
    - Automation scripts are written and remote installation is done on large number of hosts simultaneously using those scripts.

1.3 Terms in Definition.

1.3.1 Automation

Automation can be defined as"**A series of recorded steps that can be replayed again to perform same task**".The Project includes automation of:

- Pulling of code from source control.

- Compiling code.
- Packing the artefacts.
- Enable deploying of package using Web Logic.
- Running various test scripts against them.

Advantages of Automation:

- Automation is process of allowing repetitive and tedious tasks to be done by itself.
- Some tedious tasks can be made independent using automation.
- Economic method for manual testing.Reduces cost of maintainence.
- Quality check can be kept for softwares using automation scripts

1.3.2 Build

- Build is the task of converting code snippets into independent artefact that can be run on a computer by itself.[1].
- It involves many distinct functions[1] :
  1. Version Control : The version control function is used for tasks such as creation of workspace where actual code files and related files can be stored,baselining and notifying. It allows build to run in a proper environment that is needed for success of the code.
  2. Code Quality : It is also called code analysis that actually makes sure that a developer has not coded against rules of organization and followed all standards required for a production quality code.
  3. Compilation : To turn files to be compiled to direct executable.These executable are platform independent.

Chapter 2

Literature Survey

2.1 IP-Intellectual property

SYSTEM: An electronics system consists of three important components, namely ALU-Processor, Memory and Communication system. ALU-Processor performs the arithmetic and logical calculations while memory provides storage for series of commands and their intermediate results. Communication systems take-in input data and take-out results to the outputs ports. SYSTEM-ON-CHIP (SOC): Due to modern day advancements in Electronics Fabrication and (Computer Aided) Design, a complete system is being realized in single chip/IC by the Semi-Conductor companies. This fact leads to the result of calling the IC as System-on-Chip (or its acronym SOC). Architecture of a SOC depends on its and functionality and specification which in turn is governed by the application where its being used. So, type of ALU, type and size memory used and communication topology is different from Soc to Soc. Hence, in industry, these three (ALU, Memory and Communication) components are designed by independent teams before being integrated. This independent design blocks are called IPs. IP: It is acronym for intellectual property. This IPs are designed to be configurable so that SoC designer can configure it according to the need while integrating it with other IPs.
IPs is classified into two types:

Soft IP: When IP team is asked to provide a Soft IP, then, design team delivers the design (that could be a Processor or Memory or Communication System) in the form a HardWare Description Language (HDL)i.e., in simple can be said as a piece of code or in the form a structured logic gates.

Hard IP: When IP team is asked to provide a Hard IP, then, design team realizes the HDL into actual logic gates. These logic gates are then realized into layout. This layout again is realized into GDS. Layout is graphical format of the logic gates showing them to most abstract level as much as channel, substrate and oxide. GDS is their corresponding text format. Following figure is an example of a layout.

2.2 DRT

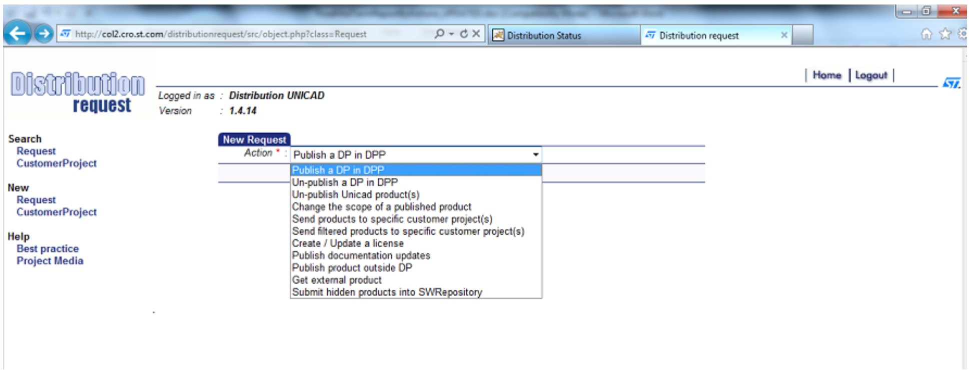
DRT stands for "Distribution Request Tracker".The TRnD department creates libraries which can be reused for present and future projects. Other divisions in ST or external customers who need these libraries for their projects make a request for these libraries on a web interface known as DRT (Distribution Request Tracking). Distribution team takes care to process all the DRT requests. There are several types of methods used to process different kind of DRT.

At present DRTs are being processed by little manual intervention .But this causes wastage of time, depends upon availability of person, involve high risk of manual mistakes and produce inefficient outputs.

The objective is to make the things automated to reduce the manual mistakes and produce efficient outputs.

They are as follows states are involved to start and complete a request.

1. Submitted
2. Assigned
3. Pending Approval
4. Pending Support
5. Approved
6. Processed



http://col2.cro.st.com/distributionrequest/src/object.php?class=Request
Distribution Status
Distribution request
Distribution request
Home
Logout
Search
Request
CustomerProject
New
Request
CustomerProject
Help
Best practice
Project Media
New Request
Publish a DP in DPP
Un-publish a DP in DPP
Un-publish Unicad product(s)
Change the scope of a published product
Send products to specific customer project(s)
Send filtered products to specific customer project(s)
Create / Update a license
Publish documentation updates
Publish product outside DP
Get external product
Submit hidden products into SWRepository

Logged in as : Distribution UNICAD
Version : 1.4.14

Action * : Publish a DP in DPP

Figure 2.1: Submitting a request

2.2.1 Submitted:

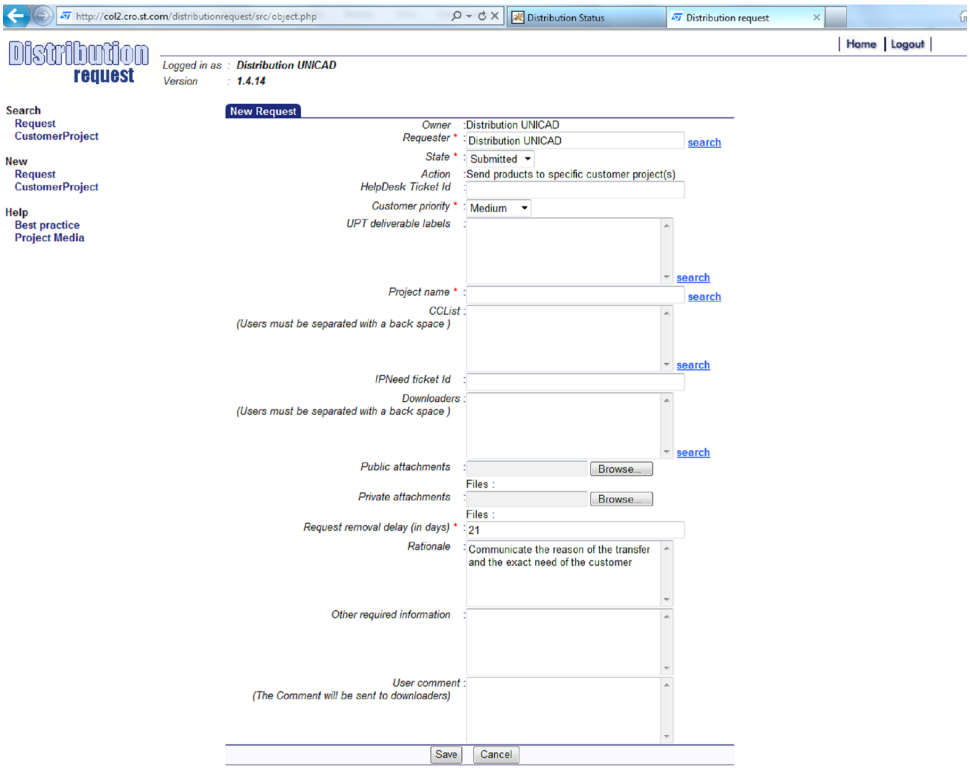
It is a state where requester request for IPs to deliver. Whenever requester access the Distribution Request page with his/her own credentials, clicks the Request link under New label at left side of the page shown in below screenshot, then requester selects type of request which are already given in dropdown list. When requester clicks next button one form will be open with appropriate information to be entered which are needed for Distribution Team. Distributed IPs are placed by the requester with respect to UPT deliverable labels corresponding with the project name, cc list, downloader list, other required information and user comment.

2.2.2 Assigned:

Once requester submits request a mail will be received by distribution team under dedicated mail box. It is just like notification to Distribution Team as new request arrived to process. When request arrives operator analyses then someone from our team will be responsible to handle the request till end of the process.

2.2.3 Pending Approval:

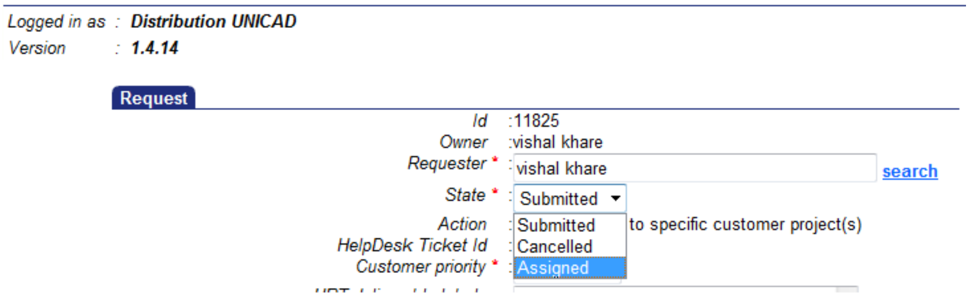
Every product before giving privileges to someone we need take approval from owner of the product. There are a group of people who can approve the type of product. Operator having an excel sheet that contains correlation between type of product and approval group. To get approval from those people operator has to compose a mail with description about request like type of media, products list, and project etc. Once the operator sends a mail to group of people, he/she can changes request to pending approval state. As shown in below screen shot.



Logged in as : Distribution UNICAD
Version : 1.4.14

Owner :Distribution UNICAD
Requester * : Distribution UNICAD
State * : Submitted
Action :Send products to specific customer project(s)
HelpDesk Ticket Id :
Customer priority * : Medium
UPT deliverable labels :
Project name * :
CCList :
(Users must be separated with a back space)
IPNeed ticket Id :
Downloaders :
(Users must be separated with a back space)
Public attachments :
Files :
Private attachments :
Files :
Request removal delay (in days) * : 21
Rationale : Communicate the reason of the transfer and the exact need of the customer
Other required information :
User comment :
(The Comment will be sent to downloaders)

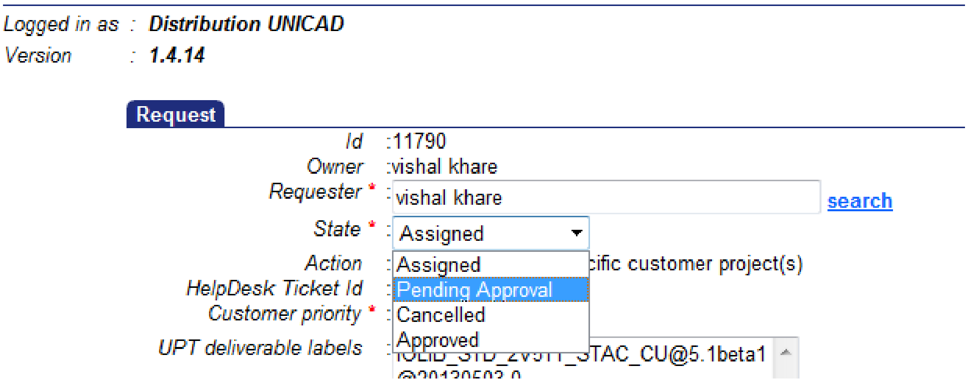
Figure 2.2: Request assigned



Logged in as : Distribution UNICAD
Version : 1.4.14

Id :11825
Owner :vishal khare
Requester * : vishal khare
State * : Submitted
Action : to specific customer project(s)
HelpDesk Ticket Id :
Customer priority * :

Figure 2.3: Request changed to pending approval



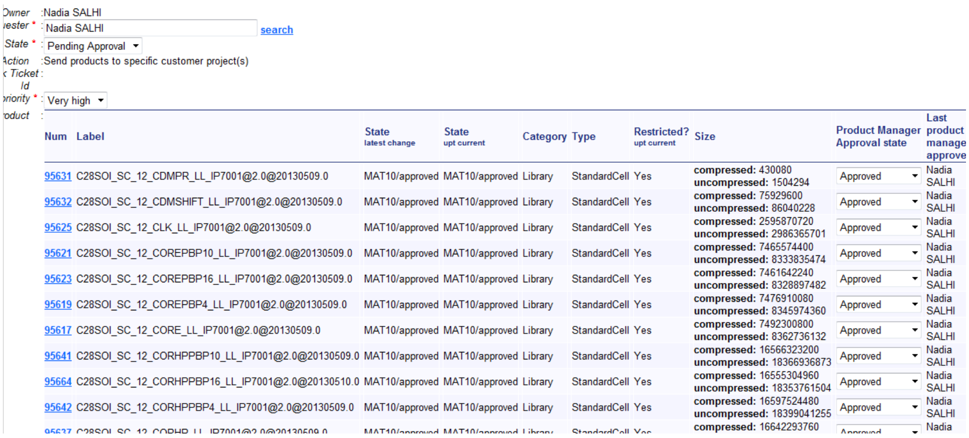
Logged in as : Distribution UNICAD
Version : 1.4.14

Id :11790
Owner :vishal khare
Requester * : vishal khare

State * : Assigned

Action : cific customer project(s)
HelpDesk Ticket Id :
Customer priority * :
UPT deliverable labels : TOLIB_STD_2V5TT_STAC_CU@5.1beta1 @20130503.0

Figure 2.4: Penging support section



Owner :Nadia SALHI
uester * : Nadia SALHI

State * : Pending Approval
Action :Send products to specific customer project(s)
k Ticket Id :
priority * : Very high
roduct :

Num	Label	State latest change	State upt current	Category	Type	Restricted? upt current	Size	Product Manager Approval state	Last product manage approve
95631	C28SOI_SC_12_CDMPR_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 430080 **uncompressed:** 1504294	Approved	Nadia SALHI
95632	C28SOI_SC_12_CDMSHIFT_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 75929600 **uncompressed:** 86040228	Approved	Nadia SALHI
95625	C28SOI_SC_12_CLK_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 2595870720 **uncompressed:** 2986365701	Approved	Nadia SALHI
95621	C28SOI_SC_12_COREPBP10_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 7465574400 **uncompressed:** 8333835474	Approved	Nadia SALHI
95623	C28SOI_SC_12_COREPBP16_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 7461642240 **uncompressed:** 8328897482	Approved	Nadia SALHI
95619	C28SOI_SC_12_COREPBP4_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 7476910080 **uncompressed:** 8345974360	Approved	Nadia SALHI
95617	C28SOI_SC_12_CORE_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 7492300800 **uncompressed:** 8362736132	Approved	Nadia SALHI
95641	C28SOI_SC_12_CORHPPBP10_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 16566323200 **uncompressed:** 18366936873	Approved	Nadia SALHI
95664	C28SOI_SC_12_CORHPPBP16_LL_IP7001@2.0@20130510.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 16555304960 **uncompressed:** 18353761504	Approved	Nadia SALHI
95642	C28SOI_SC_12_CORHPPBP4_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 16597524480 **uncompressed:** 18399041255	Approved	Nadia SALHI
95637	C28SOI_SC_12_COREHP_LL_IP7001@2.0@20130509.0	MAT10/approved	MAT10/approved	Library	StandardCell	Yes	**compressed:** 16642293760	Approved	Nadia

Figure 2.5: Approved

2.2.4 Pending Support:

After getting approval mail from any one of the group, attach text format of the mail in DRT page. Change all products state into approved state and as well as change request state to pending support. If any one of the product rejected then cancel the request.

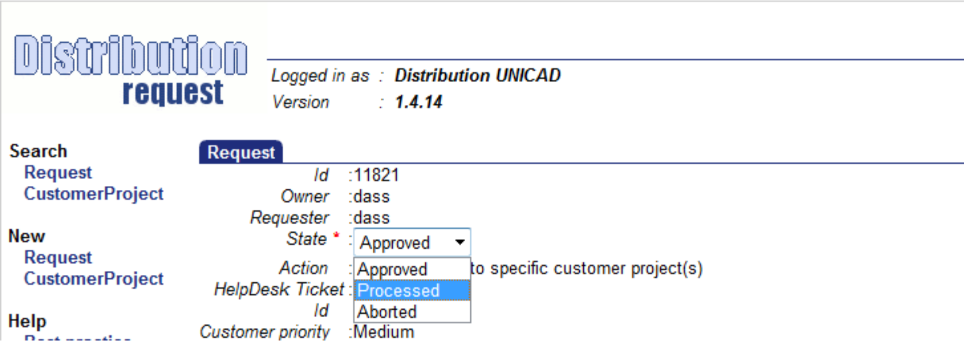
2.2.5 Approved:

This state shows all products are approved from one of the approver. Then start distributing request according to their media.

2.2.6 Processed:

This is final state it shows delivery is completed as per the standard procedure and customer gets the products successfully.

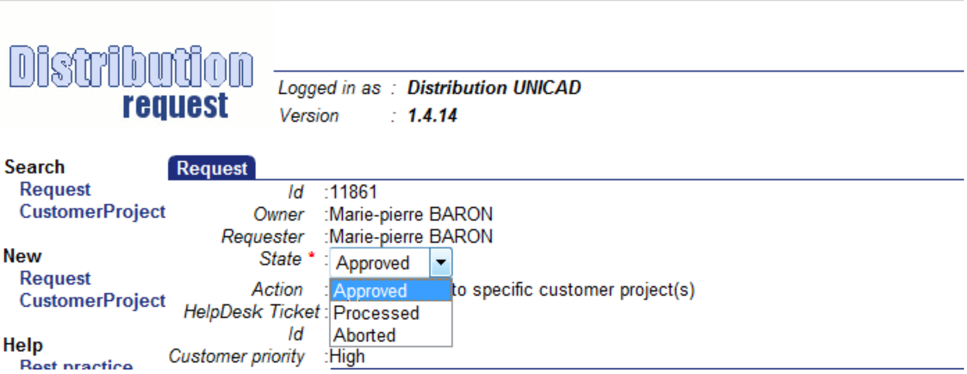
All the above states are changed manually on DRT. But now the DRT plus is going to be use. In which all the states related to product is going to change automatic if the entire request processed successfully and after that the processed button is clicked. All



Logged in as : **Distribution UNICAD**
Version : **1.4.14**

Id :11821
Owner :dass
Requester :dass
State * : Approved
Action : to specific customer project(s)
HelpDesk Ticket :
Id
Customer priority :Medium

Figure 2.6: Approved

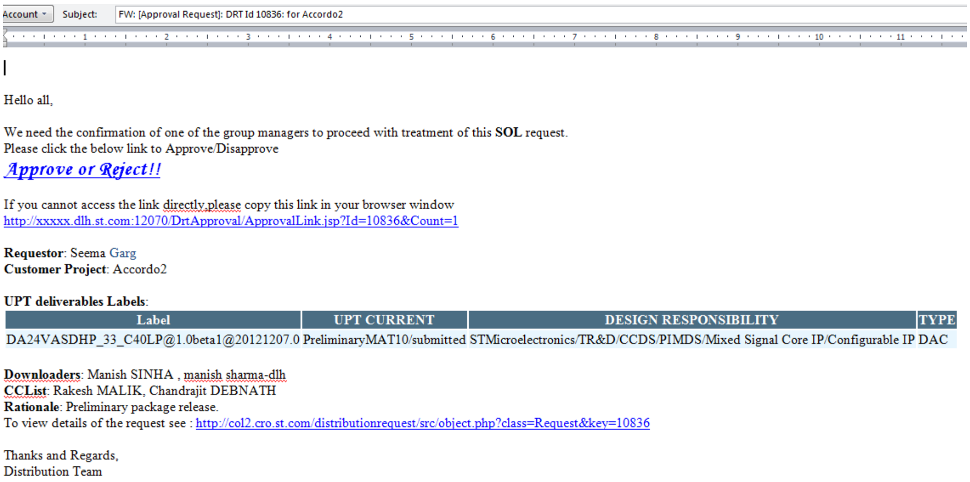


Logged in as : **Distribution UNICAD**
Version : **1.4.14**

Id :11861
Owner :Marie-pierre BARON
Requester :Marie-pierre BARON
State * : Approved
Action : to specific customer project(s)
HelpDesk Ticket :
Id
Customer priority :High

Figure 2.7: Processed

the states are going to be automate in DRT Plus.



FW: [Approval Request]: DRT Id 10836: for Accordo2

Hello all,

We need the confirmation of one of the group managers to proceed with treatment of this **SOL** request.
Please click the below link to Approve/Disapprove
Approve or Reject!!

If you cannot access the link directly,please copy this link in your browser window
http://xxxxx.dlh.st.com:12070/DrtApproval/ApprovalLink.jsp?Id=10836&Count=1

Requestor: Seema Garg
Customer Project: Accordo2

UPT deliverables Labels:

Label	UPT CURRENT	DESIGN RESPONSIBILITY	TYPE
DA24VASDHP_33_C40LP@1.0beta1@20121207.0	PreliminaryMAT10/submitted	STMicroelectronics/TR&D/CCDS/PIMDS/Mixed Signal Core IP/Configurable IP	DAC

Downloaders: Manish SINHA , manish sharma-dlh
CCList: Rakesh MALIK, Chandrajit DEBNATH
Rationale: Preliminary package release.
To view details of the request see : http://col2.cro.st.com/distributionrequest/src/object.php?class=Request&key=10836

Thanks and Regards,
Distribution Team

Figure 2.8: Approval Mail

2.3 Getting Approval

2.3.1 Approval Mail

The approval system starts after assigned state and ends with approved state, it starts once the operator finalizes request made by requester has been correct by checking all parameters are mentioned in the web form and type of request (whether it is filter or non-filter). Selecting media is depends on type of project, for every project there is a corresponding type of media There are a group of people who can approve the type of product. Operator having an excel sheet that contains correlation between type of product and approval group. To get approval from those people operator has to compose a mail with description about request like type of media, products list, and project etc. Here is one example of the composed mail given in below. The approval group given in To list, body contains list of products and link to particular request. Downloaders are nothing but end users, who are requesting privileges of products from owner of the products.

2.3.2 UOLBE

UolBE is a backend tool which is used to distribute the libraries among various ST customers. Libraries are used in different projects and these projects are distributed using different media. UolBE supports different types of media.

1. FTP

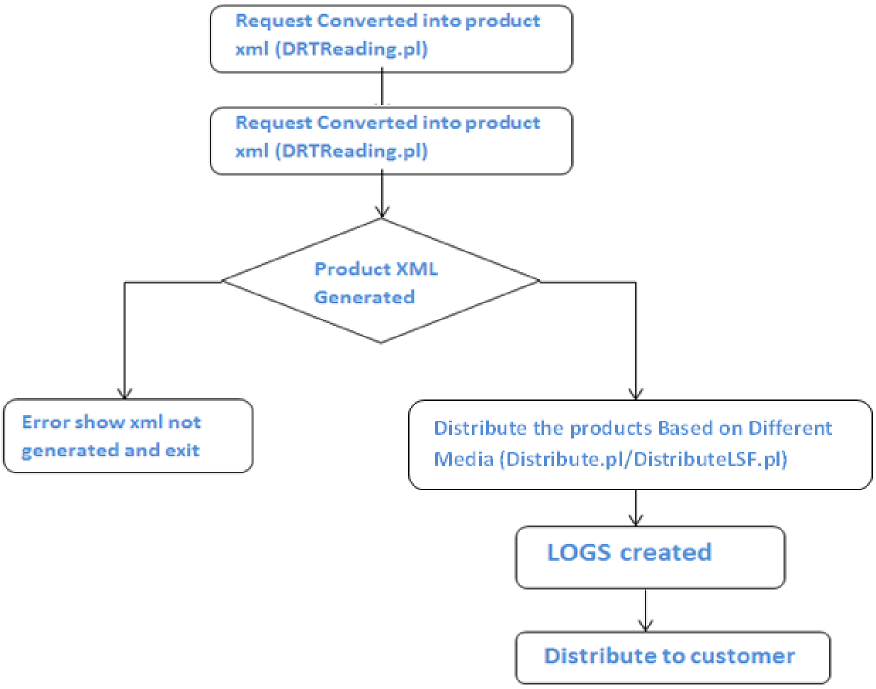

Request Converted into product xml (DRTReading.pl)
Request Converted into product xml (DRTReading.pl)
Product XML Generated
Error show xml not generated and exit
Distribute the products Based on Different Media (Distribute.pl/DistributeLSF.pl)
LOGS created
Distribute to customer

Figure 2.9: Flow of DRT

2. WebgenPlus

3. Design Sync

4. SOLPLUS

5. Service Repository

6. Servicerep-nonupt

7. IPOL-IPCatalog

8. PPOD

9. Filter, if required in any media

2.4 Medias

- FTP:- The objective of this media is to provide all requested libraries to external company or partner. These libraries are downloaded, packaged and encrypted and then put on SFT server using distribution account. To deliver ST products to external customers, we need:

- Request should be approved.
- ST FTP site: used to transfer data (example: ftp-eu.st.com)
- Installation of SFT setup.
- Sqlite3 3.0.8 or later(http:www.sqlite.org-sqlite-3.3.16.tar.gz)
- zlib-1.2.3 or later (http:www.zlib.net/)
- Python 2.5 or above (http:www.python.org/download/)
- pycurl-7.15.4
- PyXML-0.8.4
- pycrypto-2.6
- Twisted-12.1.0
- ZopeInterface-3.0.0
- pyasn1-0.1.3
- Perl package
- External downloader list
- Rationale: comment why this delivery is requested(optional)

- WebgenPlus:-
  - This media is similar to FTP.
  - In WebGen plus we create a complete tar package of compressed products and deliver it as atomic product on SFT server.
- Design Sync:-
  - For this media, we deliver the libraries on Design sync distribution buffer.
  - Using UptPlus Grant, Access grant is given to the authorized customers on Design Sync buffer.
  - Replication infra is used to replicate the data at different area and sites.
  - Access grant is given to the authorized customers on Design Sync buffer.
  - User can download the product using Sol+ client at their end or can get it from design sync cache.

- SOL PLUS:-
  - For this media, we deliver the libraries on Design sync distribution buffer.
  - DS upt2ds API is used to deliver the libraries on Design sync distribution buffer.
  - Using UptPlus Grant, Access grant is given to the authorized customers on Design Sync buffer.
  - User can download the product using Sol+ client at their end.
- Service Repository:-
  - Products are downloaded by using NonDSMedia module.
  - The products are uploaded to service repository area.
  - Products are installed in service repository using standard ucdprod generation
  - From where the customer can use the products.
- Servicerep-nonupt:-
  - This media is similar like service repository.
  - For this media, Requestor sends products to distribution team using SIFT server.
  - The products are uploaded to service repository.
  - Products are installed in service repository using standard ucdprod generation
  - From where the customer can use the products.
- IPOL-IPCATALOG:-
  - The Objective of this type of delivery is to make install of all requested IPs in specific server
  - From where the customer can use the products.
  - Products are downloaded by using a module i.e. NonDSMedia
  - Products published at the specific area.
- PPOD:-

  - The Objective of this type of delivery is to publish the product in /sw/unicad area
  - /sw/unicad area is standard repository for products accessed by different divisions of ST.
  - From where the customer can use the products.

- Filter:-
  - Filter is required when confidential data is not to be send to customers.
  - It is mentioned in DRT request if filter is required for any product.
  - Filter is generally done for ST external customers.
  - But if filter is required for ST internal customers the DRT request media is changed to FTP.
  - Filter Specs are:-
  - DPdeliverySpecName : xxxxxxxxxxxxxxxxxx
  - DPdeliverySpecVersion : xxxxxxxxxxxx
  - ViewFilter : xxxxxxxxxxx
  - CornerFilter : xxxxxxxxxx
  - Filter Specs are different for Different DRT Request.

Chapter 3

Tools and Technogies used

3.1 Setting up a fresh environment

All the tasks here are carried out on Windows Machine. Many softwares needs to be installed,their paths need to be set in environment variables.

1. **JAVA Setup**

   - Download JAVA version 7 (jre 7 and jdk 7)
   - To set the PATH variables permanently, add the full path of jdk1.7.0,"bin directory" to the PATH variable.
     - Click Start,then Control Panel, then System.
     - Click Advanced, then Environment Variables.
     - Add the location of the bin folder of the JDK installation to the PATH variable in System Variables. The following is a typical value for the PATH variable:
       C:\WINDOWS\system32; C:\WINDOWS; C: \Program Files\Java\jdk1.7.0 \bin

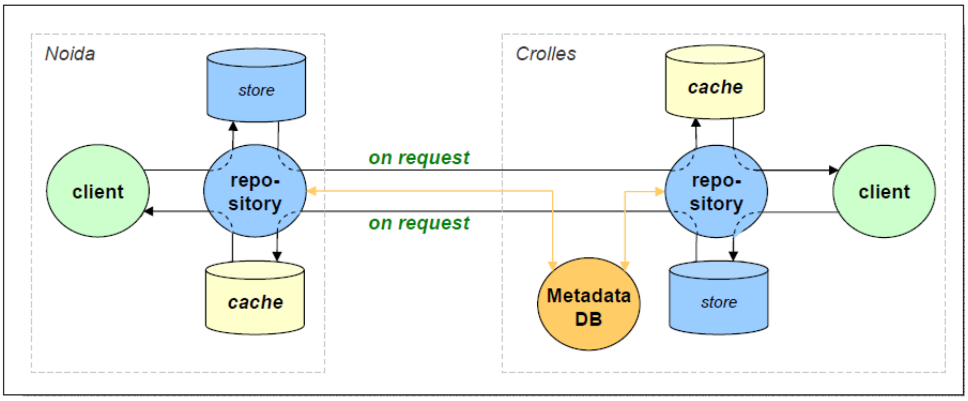


Figure 3.1: Architecture of UPT

2. **Tomcat localhost server**

    - To test for the web services before deploying them to QA or Production Servers.

3. **Instace of MySQL Workbench and access to the database connections.**

4. **Instance of VNC on Servers of NOIDA and Grenoble**

3.2 UPT

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.

3.3 Design Sync

Design synchronicity is a version control system maintains a record of all changes on a file. Types of different version control tools are Subversion, CVS, Clear case, Synchronicity. Here a few points discussed about how difficulty without a version control system.

- How do you keep track of changes in your design?
    - You can remember what changes you made, but it is difficult

  - Periodically save backups like file.h, file2.h to maintain history and for rollback
  - Maintain one excel sheet for who did what, when, and why?

- How do you coordinate who has the authority to change a file?
  - Exchange mail I'm working on file.h, so please don't touch it.
  - How to work with geographical teams?
  - Every time compress the data and send it through ftp and get updated data for next changes.

DesignSync deals with Data Management and Configuration Management. It can be used with DesignSync GUI or DesignSync Commands. Whenever user wants to upload/checkin data into DesignSync server before that user has to create a module in server. A module is a project, which is composed of one or more independently folders.
Module Instance Numbers, each time a module is populated into a root directory, a unique instance number is assigned. It is possible to have multiple instances of the same module populated into the same root directory, but in different base directories. The first (and default) instance number is For example, you might tag the current version of your design files "Alpha" when you have reached the Alpha Release is used for baseline and for delivery it is an instance of a product which is distributed to other teams mostly outside of the development team. Releases cant be removed and renamed.
Branch Suppose you are working on a IP and after sometime other product team asks for same IP with some modification. In this situation, you need to create a branch to work simultaneously on same IP. A hierarchical reference is a connection between two independent modules. It is used to create link between project data and IPs. In addition to content (folders and files) a module version can include a Hierarchical Reference (href), which is a URL to another module version, or branch. If a module version which contains one or more hrefs is populated to a workspace, the hrefs are followed, and the referenced module versions are populated into the workspace as well. A module version which is populated as a result of following a href may contain hrefs as well.
There is no limit to the depth. In this manner, a module hierarchy can be fetched using a single populate command. A href specifies a containment relationship between a module and another module (sub-module). The base directory into which the sub-module is

populated is specified by a path relative to the base directory of the parent module.

3.4 LSF-Load Sharing Facility

A compute farm is composed of a network of computers organized into a seamless system, to which multiple jobs can be submitted. The system will then find the best available machine to run the job, schedule the job, execute it and inform (if requested) when the job is completed. The software to provide this functionality is called LSF (Load Sharing Facility).

Advantages of LSF :

- A network of computers can be used as a single virtual system.
  For e.g. if we connect 20 machines, it is equivalent to have a set of 20 CPUs which are available to run jobs for everyone, without having to login into each machine and run the job manually.
- LSF automatically selects the best available host to run the job and schedules the job automatically on that host.
- If there are a number of jobs to run, one can submit all the jobs to LSF and LSF will keep on scheduling and running the jobs one after another as soon as the resources become available.
- LSF maintains full control over the jobs, including the ability to suspend and resume jobs based on load conditions of the host machine.

LSF jobs have the following states:

- PEND Waiting in a queue for scheduling and dispatch
- RUN Dispatched to a host and running
- DONE Finished normally with zero exit value
- EXITED Finished with non-zero exit value
- PSUSP Suspended while pending
- USUSP Suspended by user

- SSUSP Suspended by the LSF system
- POST_DONE Post-processing completed without errors
- POST_ERR Post-processing completed with errors
- WAIT Members of a chunk job that are waiting to run

3.5 Shell Script

A shell script is a text file written for the shell, or command line interpreter that contains a sequence of commands for a UNIX-based operating system. It's called a shell script because it combines into a "script" in a single file a sequence of commands that would otherwise have to be presented to the system from a keyboard one at a time.
Shell script has been used in this project for configuring tools, working on files, directories and connection with another server.

3.6 PERL

Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). Its major features are that it's easy to use, supports both procedural and object-oriented (OO) programming, has powerful built-in support for text processing, and has one of the world's most impressive collections of third-party modules.
Perl is needed where integrating with web, reading web pages, Data Base connection, taking inputs from web interface. Process each media until products are distributed.

3.7 JSP

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSP can be used independently or as the view component of a server-side modelviewcontroller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller.

3.8 Servlets and Ajax

A servlet is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers.

Ajax (also an acronym for Asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behaviour of the existing page. Data can be retrieved using the XML Http Request object. Despite the name, the use of XML is not required (JSON is often used instead), and the requests do not need to bisynchronous.

3.9 Selenium

The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases. Its an easy-to-use Firefox plug-in and is generally the most efficient way to develop test cases. It also contains a context menu that allows you to first select a UI element from the browsers currently displayed page and then select from a list of Selenium commands with parameters pre-defined according to the context of the selected UI element. This is not only a time-saver, but also an excellent way of learning Selenium script syntax.

Chapter 4

End-To-End Process

This chapter is mainly about the discussion of the project implementation. This project is implemented by Evolutionary Prototyping. Prototyping and Evolutionary prototyping is discussed below. The various issues that were encountered during the design and development of the project are given below.

4.1 Prototype Model

Prototyping approach, also known as evolutionary approach, came to picture because of failures that occurred in the final version of the software application developed using the waterfall approach. The failure generally occurs because of the changes in the requirement of the proposed system or because of the gap in understanding the customer requirement by the development team. A gap in the first version of the developed application, inevitably leads to the need for redoing the application. To overcome these limitations, the concept of prototyping was introduced.

4.1.1 Prototype Approach

A prototype is a sample implementation of system that shows the limited and main functional capabilities of any proposed system. After any prototype is built, it is delivered to a customer for evaluation. The prototype helps customers to determine how the feature will function during the final software release. The customer provides suggestion and improvements over the prototype. The development team then implements those suggestion on the new prototype, which is , again evaluated by the customer. The process continues until the customer and the development team understands the exact requirement of the proposed system. When a final prototype is developed, then requirement is considered frozen.

This prototyping approach is also used in requirement gathering and during analysis phase so that they capture the exact requirement of proposed system. After the requirements are frozen, the remaining phases of development process need to be executed to complete the overall development of software system.

An e-commerce website, such as a shopping site is one example where you can implement prototyping approach. You can develop a prototype of various web-pages of the site such as a catalogue page, a product order page etc., and then, present it to the customer for an approval. If the customer approves this prototype of the site, requirements are states again and the design of the web site gets initiated. If the customer does not approve to the web site, then development team revisits the prototype and resubmits it to customer for an approval. This process continues until the prototype gets approved.

- Not a standalone, or a complete development methodology, but rather an approach select parts of a large, traditional development methodology (i.e. incremental, spiral, rapid application development (RAD)).

- Attempts to reduce any inherent project risks by breaking projects into smaller segments and provide ease-of-change during development process.

- User involved throughout the development process, which increase the likelihood of user acceptance of final implementation.

- Small-scale mockups of the system developed followed by an iterative modification process until prototype evolves to meet the user requirements.

- While most prototypes get developed with an expectation that they will get discarded, it is possible in some cases to evolve from a prototype to a working system.

- A basic understanding of fundamental business problem is necessary to avoid solving a wrong problem. Outline of the prototyping process

The process of prototyping involves the following steps.

- Identify basic requirements: Determine basic requirements including the input and output information desired. Details, such as security, can typically be ignored.

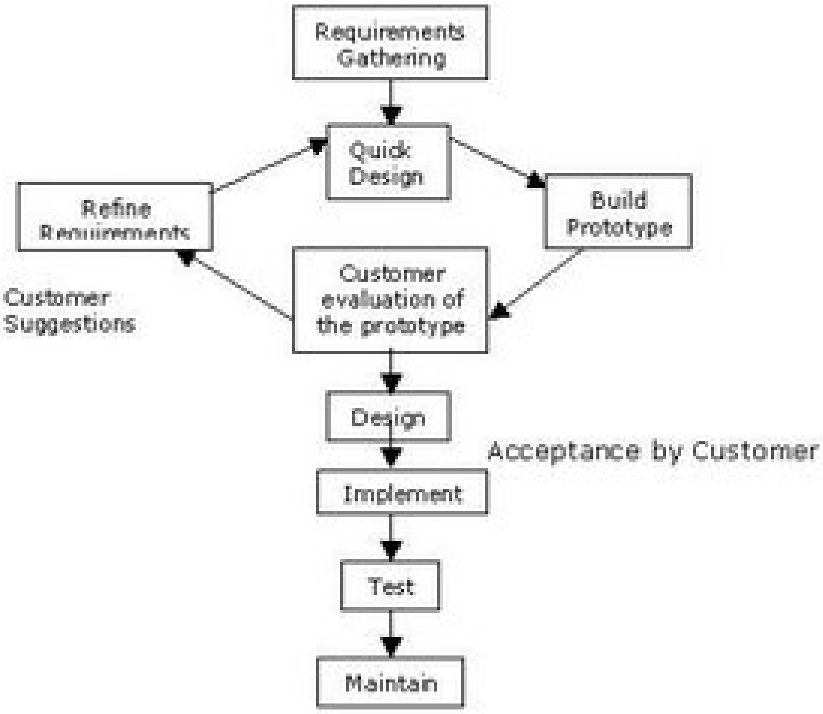


Figure 4.1: Prototype of process

- Develop Initial Prototype: The initial prototype is developed that includes only user interfaces.

- Review: The customers, including end-users, examine the prototype and provide feedback on additions or changes.

- Revise and Enhance the Prototype: Using the feedback both the specifications and the prototype can be improved. Negotiation about what is within the scope of the contract/product may be necessary. If changes are introduced then a repeat of steps 3 and 4 may be needed.

4.1.2 Evolutionary prototyping

Evolutionary Prototyping (also known as breadboard prototyping) is quite different from Throwaway Prototyping. The main goal when using Evolutionary Prototyping is to build a very robust prototype in a structured manner and constantly refine it. The reason for this is that the Evolutionary prototype, when built, forms the heart of the new system, and the improvements and further requirements will be built.

When developing a system using Evolutionary Prototyping, the system is continually refined and rebuilt. Evolutionary prototyping acknowledges that we do not understand all the requirements and builds only those that are well understood. This technique allows the development team to add features, or make changes that couldn't be conceived during the requirements and design phase. It is not unusual within a prototyping environment for the user to put an initial prototype to practical use while waiting for a more developed version. The user may decide that a 'flawed' system is better than no system at all. In Evolutionary Prototyping, developers can focus themselves to develop parts of the system that they understand instead of working on developing a whole system.

To minimize risk, the developer does not implement poorly understood features. The partial system is sent to customer sites. As users work with the system, they detect opportunities for new features and give requests for these features to developers. Developers then take these enhancement requests along with their own and use sound configuration-management practices to change the software-requirements specification, update the design, recode and retest.

4.1.3 Throwaway prototyping

It also called close-ended prototyping. Throwaway prototyping is where the objective of the evolutionary development process is to understand the customer's requirements and hence develop a better requirements definition for the system. The prototype concentrates on experimenting with the customer requirements that are poorly understood.

The most obvious reason for using Throwaway Prototyping is that it can be done quickly. If the users can get quick feedback on their requirements, they may be able to refine them early in the development of the software. Making changes early in the development lifecycle is extremely cost effective since there is nothing at that point to redo. If a project is changed after a considerable work has been done then small changes could require large efforts to implement since software systems have many dependencies. Speed is crucial in implementing a throwaway prototype, since with a limited budget of time and money little can be expended on a prototype that will be discarded.

With Throwaway Prototyping, code is developed to explore factors critical to the system's success, and then that code is thrown away. The prototyping implementation uses programming languages or development practices or both are much faster than the target language and practices. The user interface is prototyped far more commonly than any other part of the system, but other parts of some systems can also benefit from being prototyped. When used as a requirements specification aid, the Throwaway Prototyping practice can accelerate projects based on traditional lifecycle models, such as DoD projects. It can be initiated at either a management or technical level.

4.2 Assumptions Made

The following are a few assumptions that hold true in all cases.

- Any request has to get approval from two or more approval groups then PM (Project Manager) team approves the request.

- When project is not found in DB for media and project correlation, operator add project and corresponding media either an SOL if the request is not for an external company or checking previous request get right media.

- Filter parameters for filter request getting from other required information mentioned in Distribution request web form if not then assume it is not a filter request.

- Create an xml file with all input data which are needed to process request till end of the state. Keeping this xml file between two phases of implementation, acts like mediatory between these phases.

4.3 Design and Implementation

The following diagram explains the design of complete system. The system majorly consists of 4 modules. They are,

- Information Gathering : In this phase reading the data from DRT tool page using PERL packages at back end and it is intern connected with MySQL data base where the data is extracted and keeping in local Data Base, Then given to the further phase.

- DRT Tracker: The information which is extracted from the first phase is displayed in a web interface. Which shows status of request and necessary information which are needed.

- Approval Management System : Every request has to be approved from the owner of the products before distributing them. This is done by approval management system automatically.

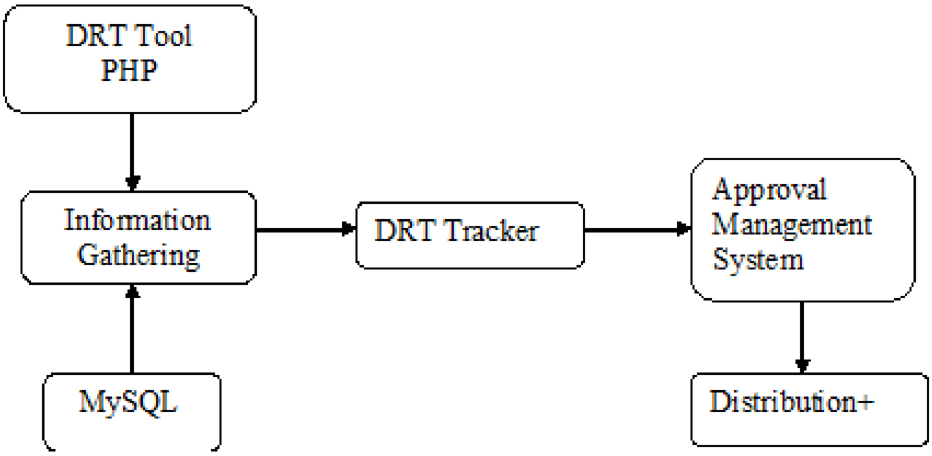


Figure 4.2: Design of IP Distribution Management System

- Distribution : Once product has been approved then operator distributes the products according to their media.

4.3.1 Gathering Information

Distribution request has its own Data Base but to process a request we need a few information instead whole DB. Information which we need keeping in another DB in Noida for distribution application. The DB which is in Noida is just like a copy of Main DB with a few modifications and necessary information. The modifications of data before inserting into DB are done at back end using PERL scripts.
List of tables involved :

- State Table contains all information fetching from main DB which is given by requester. It is used in DRT to display status as well as information of the request. The above table consist of 7 entities, they are :
  - request id: it is the primary key, requested by the requester it is unique for all requests
  - State: it tells status of the request
  - Requester: the one who request
  - Operator: the one who operating particular request
  - Media: type of request
  - Project_Name: type of project

  - Other required Information: it tells significance of request

- cclist which contains request id and mail id. These mail ids are used when getting approval from a group of people.

- downloader which contains request id and downloaders mail id. These mail ids are used when getting approval from a group of people. The distributed products are used by these downloaders.

- product which contains request id, products and uptType. These are the products requested by the requester corresponding with the UPT Type.

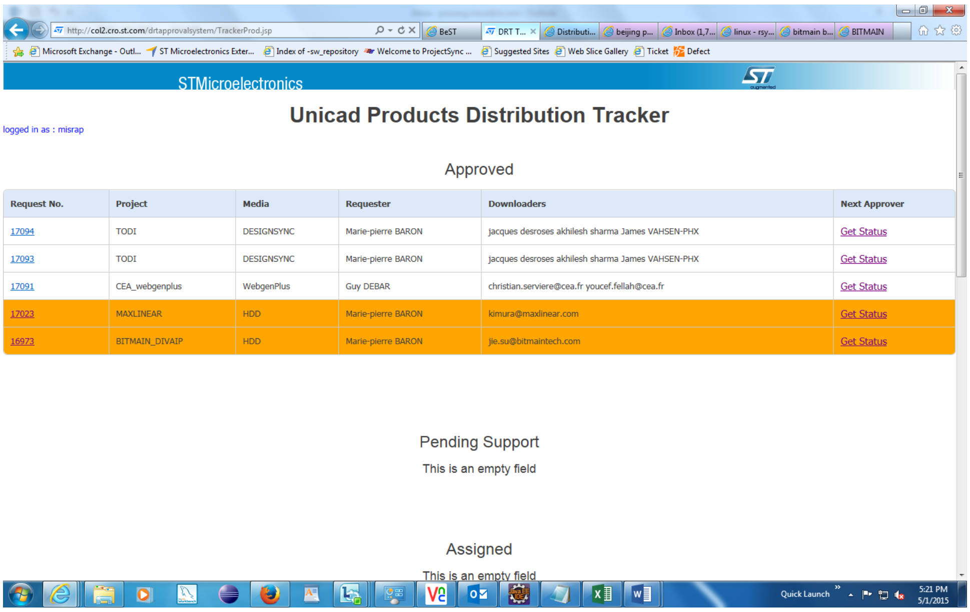
- ”Approval Status” consist of s.no, ReqId, ID, AppovalStatus, ApprovalTime and ApproverName. Here recoding of approver name, the one who approved the type of products and time at which it has been approved

4.3.2 Distribution Tracking Page

This is a front end Web application developed using JSP, Servlets and Ajax. It is difficult to keep track of each request arrival and present status of all the delivery requests. As discussed in old methodology every request notification comes through a mail to distribution team. The operator has to check in frequent intervals whether new request arrived and status of old request, it is an overhead and also difficult to track.
DRT (Distribution Tracking Page) places all submitted and pending requests in one page with relevant information. Here instead of tracking mails just querying to database to provide submitted and pending requests. At the back end it is filtering all requests which are in state processed, cancel and abort, displaying rest of the requests in a tabular form having 10 columns of related information. Each and every request starts with assign state and end with processed state. These process notifications showed in second column in below screen shot.

Here, Through the ”Get Status” feature, we can get the live update from the DB regarding what all products need an approval or what all products have already been approved.

Here all requests are sorted according to request number that is given in first column. When a new request arrives the state of the request is submitted, for every request before



Unicad Products Distribution Tracker

logged in as : misrap

Approved

Request No.	Project	Media	Requester	Downloaders	Next Approver
17094	TODI	DESIGNSYNC	Marie-pierre BARON	jacques desroses akhilesh sharma James VAHSEN-PHX	Get Status
17093	TODI	DESIGNSYNC	Marie-pierre BARON	jacques desroses akhilesh sharma James VAHSEN-PHX	Get Status
17091	CEA_webgenplus	WebgenPlus	Guy DEBAR	christian.serviere@cea.fr youcef.fellah@cea.fr	Get Status
17023	MAXLINEAR	HDD	Marie-pierre BARON	kimura@maxlinear.com	Get Status
16973	BITMAIN_DIVAIP	HDD	Marie-pierre BARON	jie.su@bitmaintech.com	Get Status

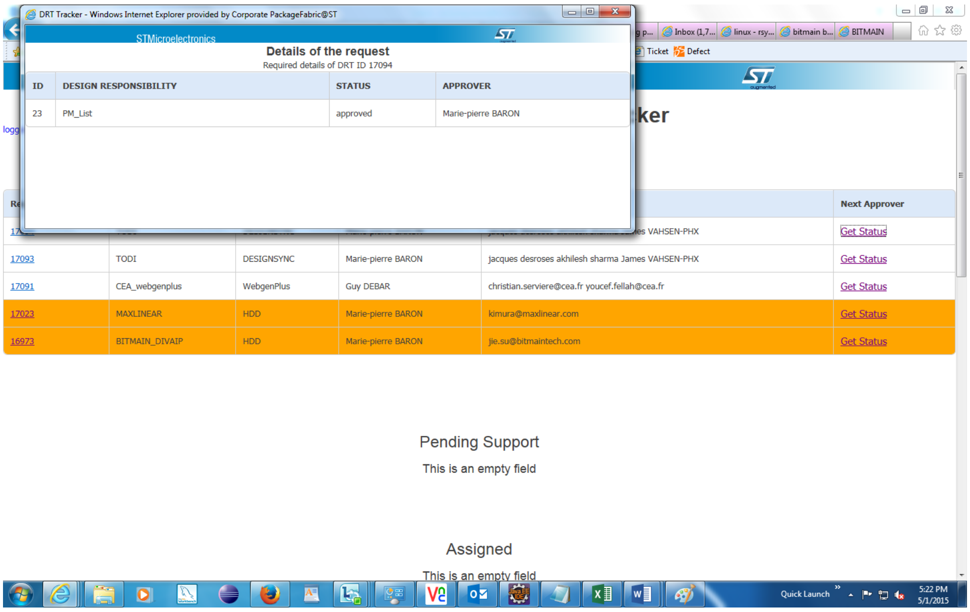
Pending Support

This is an empty field

Assigned

This is an empty field

Figure 4.3: DRT Operator Tracker



Details of the request

Required details of DRT ID 17094

ID	DESIGN RESPONSIBILITY	STATUS	APPROVER
23	PM_List	approved	Marie-pierre BARON

ker

logg

Re					Next Approver
17				es VAHSEN-PHX	Get Status
17093	TODI	DESIGNSYNC	Marie-pierre BARON	jacques desroses akhilesh sharma James VAHSEN-PHX	Get Status
17091	CEA_webgenplus	WebgenPlus	Guy DEBAR	christian.serviere@cea.fr youcef.fellah@cea.fr	Get Status
17023	MAXLINEAR	HDD	Marie-pierre BARON	kimura@maxlinear.com	Get Status
16973	BITMAIN_DIVAIP	HDD	Marie-pierre BARON	jie.su@bitmaintech.com	Get Status

Pending Support

This is an empty field

Assigned

This is an empty field

Figure 4.4: DRT Operator Tracker

going to process need to get approval. In the second column corresponding to state of the request a button will be popup. If it is submitted state a button will be popup with the name change to pending approval, when operator click on button if the media of the specified project is identified as SOL/FTP/Service Repo/JDP an Automatic Approval mail is sent to a group of people.

Once getting approval the state of request change to approved. When state is identified as approved one more button is appeared it is called Process button. Once click on process button at back end corresponding scripts will be launch. Every process button has their own functionality corresponding to media. A link will be appear after clicking the process button that shows log and status of the each product.

4.3.3 Approval Management System

The purpose of taking approval from approval group before processing a request. Here the approval management system has been automated since sending a mail until getting approval from owner. Distribution Tracking Page is central point of managing all type of requests where operator can take approval and process a request. Approval management system is one of the parts in whole project.

Here discussing about automation has been done on approval management system after clicking change to pending approval button on DRT. In Noida we maintain a separate Database by querying to central DB which is in another site.

It is very difficult to send a number of mails to get approve for one request. If any request has to get more than one group approval then PM (Project Manager) team will approve the products. One automatic mail has been sent to approver for approve/reject of customer products. A link has given in mail it takes into web interface with appropriate information to take decision whether to accept or reject.

As shown in above screen shot, it has two radio buttons to take decision whether approve or disapprove. Once approver approves the products the state of project manger approval state becomes approved and then state of request becomes approved until all products has been approved. The state of request as well as products has been changed using PERL at back end. It reading entire web page and changing it to next state.

If approver is not an authorized to approve, a warning message will be displayed on interface as shown in above screen shot. Authorization is takes place comparing with the

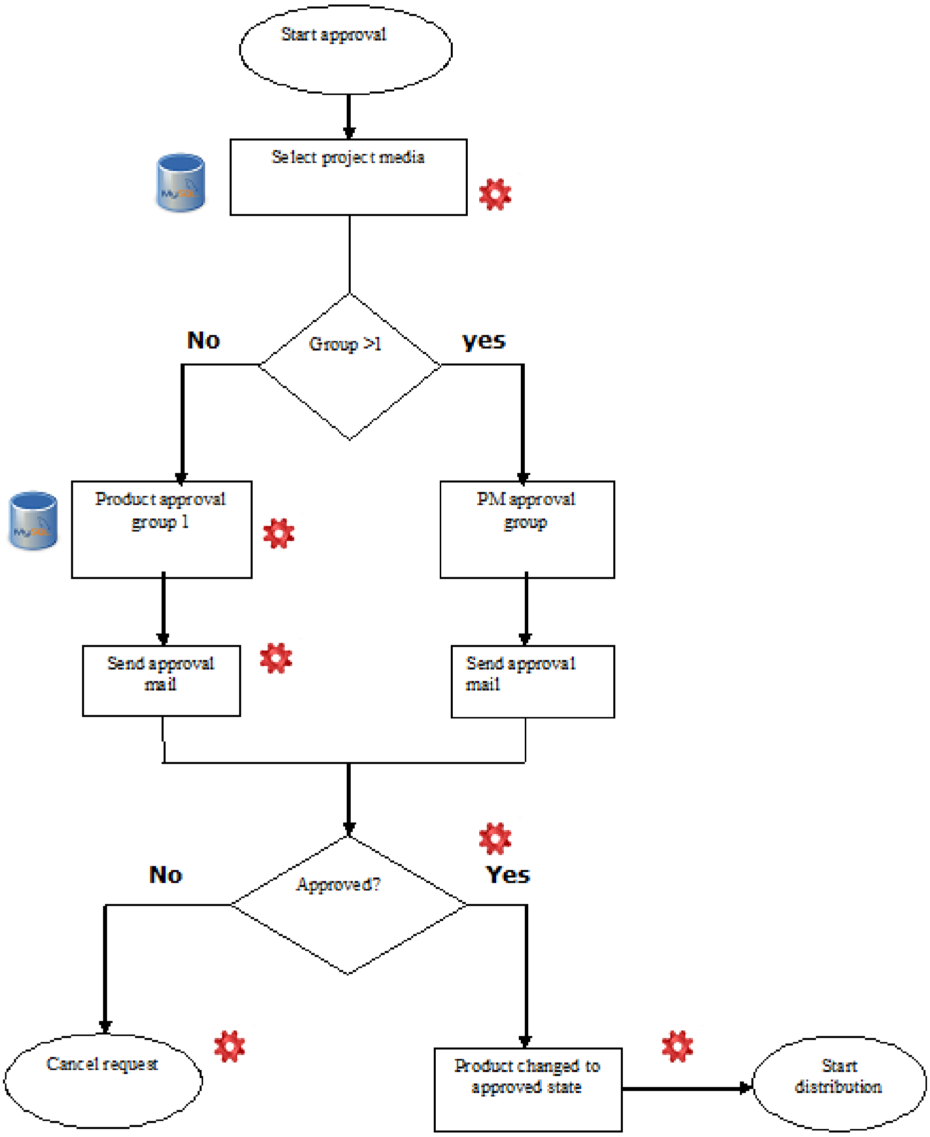
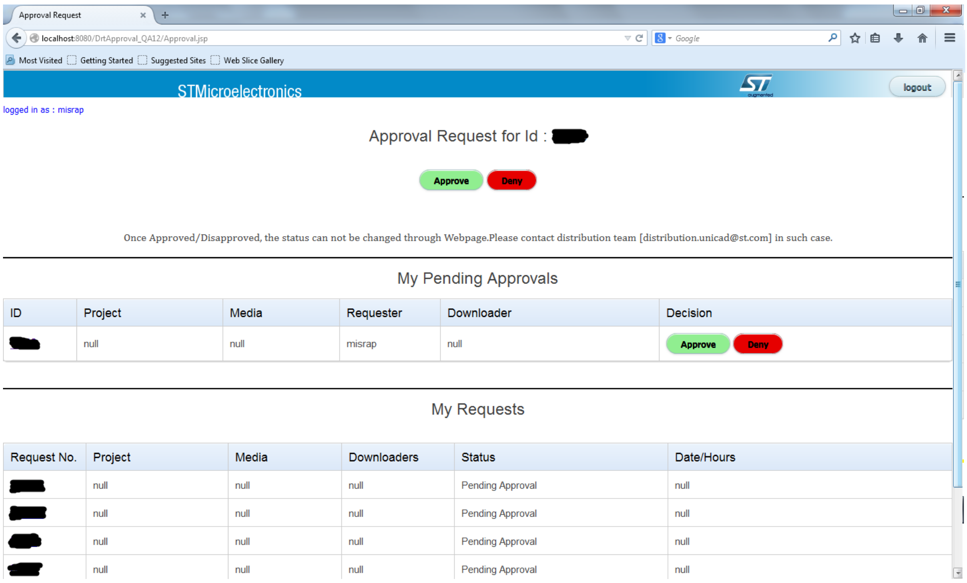


Figure 4.5: Approval Management System



logged in as : misrap

Approval Request for Id : [redacted]

Approve Deny

Once Approved/Disapproved, the status can not be changed through Webpage.Please contact distribution team [distribution.unicad@st.com] in such case.

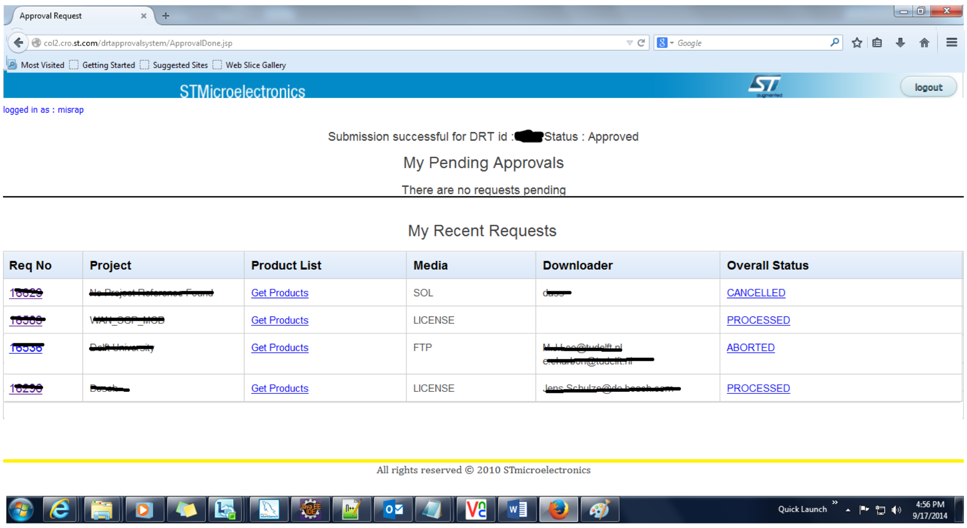
My Pending Approvals

ID	Project	Media	Requester	Downloader	Decision
[redacted]	null	null	misrap	null	Approve Deny

My Requests

Request No.	Project	Media	Downloaders	Status	Date/Hours
[redacted]	null	null	null	Pending Approval	null
[redacted]	null	null	null	Pending Approval	null
[redacted]	null	null	null	Pending Approval	null
[redacted]	null	null	null	Pending Approval	null

Figure 4.6: Distribution Dashboard Approval frame



logged in as : misrap

Submission successful for DRT id :[redacted]Status : Approved

My Pending Approvals

There are no requests pending

My Recent Requests

Req No	Project	Product List	Media	Downloader	Overall Status
[redacted]	[redacted]	Get Products	SOL	[redacted]	CANCELLED
[redacted]	[redacted]	Get Products	LICENSE		PROCESSED
[redacted]	[redacted]	Get Products	FTP	[redacted]	ABORTED
[redacted]	[redacted]	Get Products	LICENSE	[redacted]	PROCESSED

All rights reserved © 2010 STmicroelectronics

Figure 4.7: Distribution Dashboard Approval completed

user window names which are already collected in one table corresponding with approval mail id.

4.3.4 Distribution

Objective is to take entire system into web interface and operate every delivery trough web interface. To build a web interface as well functioned, it need a better back end system to make it run at any circumstances. In old methodology there is no common system to process any type of media and as well as any type of delivery (either filter or non filter). Inputs which are needed for scripts used enter by the operator. If there is any filter request arrived that has to be taken hold and processed an independently.

Distribution+ is a new system developed at back end to distribute products at right place. It is divided into two phases. First phase will run one time per a request and second phase will run repeatedly until successful completion of distribution.

- Reading inputs from DRT (Distribution Request Tracker)
- Launch on LSF (Load Sharing Facility)

First phase provide input to second phase as input capture from DRT as well as from Data Base. Input for this phase is a delivery request id according to that it login into DRT page using distribution credentials. After login into system it reads the web form of the requester and parallel it login into Data Base get some more required information. After getting all information it will keep entire metadata by creating a new xml file per a product under separate directory. There are two types of requests filter and non filter, to process a filter request it need filter specifications. Populate filter specifications from Design Sync by providing proper name and version. This is one time operation for all products given in request. If request is not a filter request, just jump into second phase. For example if a request has 10 products it creates 10 xml files. These xml files are input to the next phase.

Coming to second phase it is repeatedly and parallel launching on LSF until product has been delivered successfully. LSF is a load sharing facility to do parallel jobs on server. Here keeping this xml file between two phases of implementation, acts like mediatory between these phases. In this phase first it reads xml file and get product metadata. If product is available in SW UNICAD it copies product into local area or else it will perform an UPT get operation to download products into local area. It checks filter parameters given in xml. If it is found any filter parameters it will perform filter on each products. After that it will call respective media (FTP, JDP, and Service Repository) files corresponding to the request. Due to some reasons if LSF job failed it will again launch the second phase for failed products.

Chapter 5

Conclusion

Hence the basic understanding of Distribution of IPs has been done. Steps involved processing a request and also types of requests (Internal or External) and deep understanding of each type of media and getting approvals for each request and also Purpose of each tool such as (LSF, Design Sync etc.). Considering above deviations in the system, finally we decided to develop a good front end where requester place a request. We already developed a back end system irrespective of type of delivery. This front end does not allow any type of free text and having necessary checks for input fields.

References.

[1] Selenium Webdriver Projects *http://www.seleniumhq.org/projects/webdriver/*

[2] Setting up Selenium with eclipse *http://selftechy.com/2011/05/31/setting-up-selenium-with-eclipse*

[3] Prototypes *http://www.slideshare.net/rhspcte/software-engineering-ebook-roger-s-pressman*

[4] XML parser in perl *http://search.cpan.org/ msergeant/XML-Parser-2.36/Parser.pm*

[5] Perl Introduction Documentation *http://www.perl.com/pub/2000/10/begperl1.html*

[6] Basic of shell scripting *http://linuxcommand.org/writing_shell_scripts.php*