International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-3, Issue-4)

Research Article | **April 2014**

Spotlight Feature and Infrastructure Development

Prasanna Kumar G B
Information Science& VTU
India

Srinidhi Adiga
LSI R&D Pvt.Ltd
India

Dr.Radhika K R
.Information Science &VTU
India

Abstract—

Electronic design automation (EDA or ECAD) is a category of software tools for designing electronic systems such as printed circuit boards and integrated circuits. Spotlight is an EDA Intellectual Property (IP) and Design Checker tool from LSI for 65 nm (nm scales "transistor gate length") and higher technologies. Spotlight is an EDA Intellectual Property (IP) and Design Checker tool from LSI for 65 nm (nm scales "transistor gate length") and higher technologies.This paper consists of modules to enhance the service quality and maintainability of Spotlight. It encompasses all the tasks as addition and customization of rules, Spotlight Quality Drive, Spyglass Tool Qualification, Spotlight Release Automation and building database and GUI support. The "Spotlight Infrastructure Development" part is all about developing scripts and building support like maintainable as well as efficient database, frontend web application for easy and desired retrieval of information from database. Spotlight Infrastructure Support comprises of following modules: Spotlight Dashboard,Spotlight Run Analysis,Spotlight Report Consolidation.

Keywords— Spotlight, Spotlight Dashboard, Spotlight Run Analysis, Spotlight Report Consolidation ,Model View Controller.

I. INTRODUCTION

Spotlight is a wrapper over which combines LSI developed custom audits and augments the native Spyglass engine. Briefly, Spotlight checks whether the circuit design satisfies specific recommended design guidelines.

Spotlight Dashboard: The Spotlight Dashboard Support is to store complex and huge XML data to an easily accessible database and retrieve it using an effective real time web application keeping in mind the developers and project managers as audience. It is accompanied with multiple tables dedicated to different attributes and elements of XML.As we know, RDBMS is very much developed and effective technique for data store and retrieval.It encompasses development of a fully flexible and efficient GUI to get the current and historical trends of data about the Spotlight Runs.

Spotlight Run Analysis: There is always a need to capture the data post software run, process it and show it in presentable form for further enhancement of product quality. Spotlight Run Analysis is an effort to make it as simple as possible for developers. The reports generated post Spotlight Run is periodically uploaded to database server and all the run information and metadata is available to developer. The developer is not bothered with complexities of querying a huge database for information retrieval. All possible flexibility and ease is provided to developer with features like exporting filtered results to excel and use of dynamic graphs and timelines for better visualization.

Spotlight Report Consolidation: Until now, it was very cumbersome task to read the output reports generated as a result of Spotlight run. The user has to look in multiple subdirectories e.g. spotlight_errors, spotlight_warnings and spotlight_info in Spotlight output directories and extract the desired information. So a tool was needed to automate the task of consolidation of reports to user friendly spreadsheets with added option to customize the format of spreadsheet and output of tool. This module mainly consist of development of a Linux based command driven tool to parse and extract the desired information spread in Spotlight output directories containing the output reports and populates multiple spreadsheets.

II. METHODOLOGY

Model–view–controller (MVC) Pattern:

Model–view–controller is a software architecture pattern which separates the representation of information from the user's interaction with it. The **model** consists of application data, business rules, logic, and functions.A **view** can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a bar chart for management and a tabular view for accountants.The **controller** mediates input, converting it to commands for the model or view. The central ideas behind MVC are code reusability and separation of concerns.In addition to dividing the application into three kinds of components, the MVC design defines the interactions between them.A **controller** can send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document). It can also send commands to the model to update the model's state (e.g., editing a document).A **model** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the

software platform does not support them. A **view** requests from the model the information that it needs to generate an output representation to the user. The projects "Spotlight Dashboard" and "Spotlight Run Analysis" typically follow the MVC Architecture of web development.

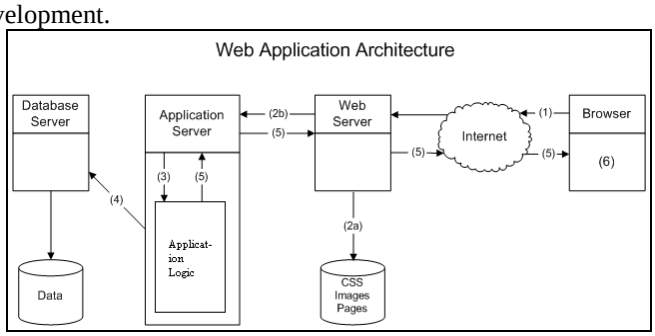

Figure 1: Web Application Architecture

III. IMPLEMENTATION

Spotlight Dashboard: The flow of events occurring from origin of XMLs' from Spotlight Run till the retrieval of organized and desired info using the web application can be understood by dividing all operations into two categories i.e. On Client Machine and On Server Machine

On Client Machines :

1. Changes made in Spotlight for Spotlight Dashboard:
2. Moving **XML** generated to Temp Dir
   - Do FTP
   - Set the credentials for remote FTP server
   - Check if FTP server exist
   - If connection is ON
   - If file to be FTP exist
3. Write the progress in log file and store to spotlight_analyis Output Directory
4. parse spotlight.log and get the run rime

On Server Machine :

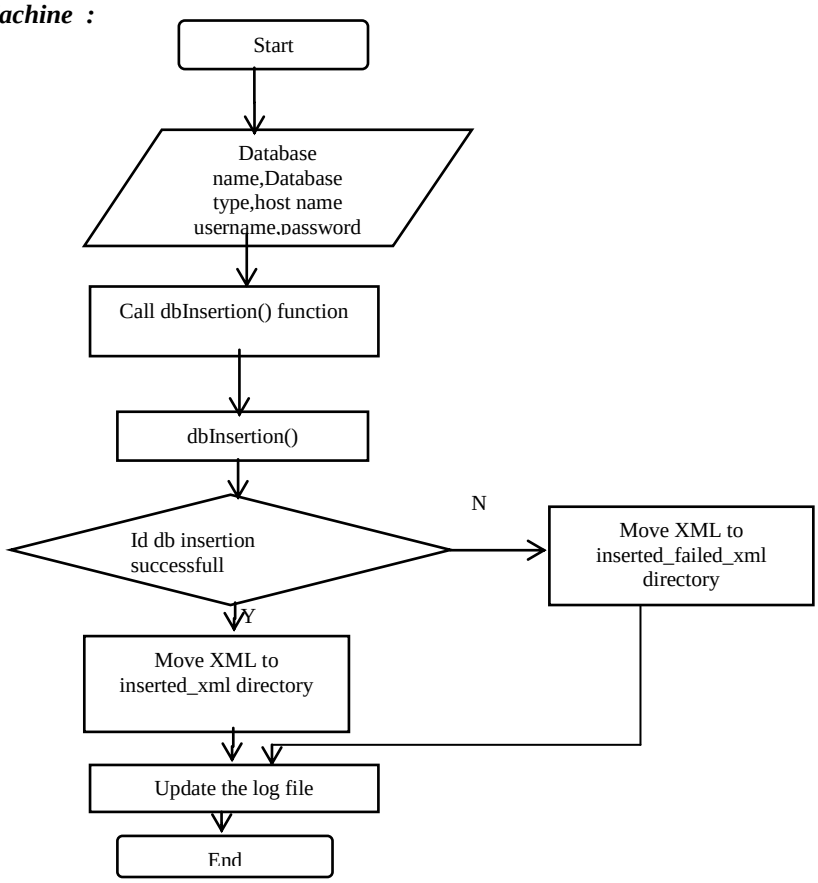

Figure 2: Flowchart showing events on server for spotlight dashboard module

Entity Relationship Diagram for the "SpotlightDB" database

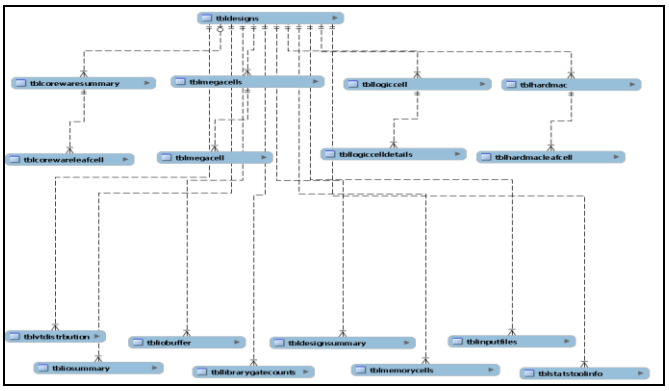

Figure 3: SpotlightDB (Dashboard) database ER Diagram

Spotlight Run Analysis: The flow of events occurring from origin of CSVs' from Spotlight Run till the retrieval of organized and desired info using the web application can be understood by dividing all operations into two categories i.e. On Client Machine and On Server Machine.

On Client Machines :

Changes made in Spotlight for Spotlight Run Analysis Project are as follows

1. check if fast_flow enabled
2. check for template names in run
3. call to shell script to create csvs' to be sent via FTP
4. concatenate all rule files to one to send via FTP
    - Do FTP
    - Set the credentials for remote FTP server
    - Check if FTP server exist
    - If connection is ON
    - If file to be FTP exist
5. Write the progress in log file and store to spotlight_analyis Output Directory
6. parse spotlight.log and get the run rime

On Server Machine :

Algorithm for Spotlight Run Analysis

- Provide variables like database name, database type, host name, user name, password for connection to database server.
- Search for data.csv and corresponding rpt.csv in spotlight_csv directory.
- Parse data.csv line by line and get data and insert in corresponding columns of "sum_data" table.
- Set the foreign key value for "sum_rpt" table.
- Parse rpt.csv to get data and insert in corresponding columns of "sum_rpt" table.

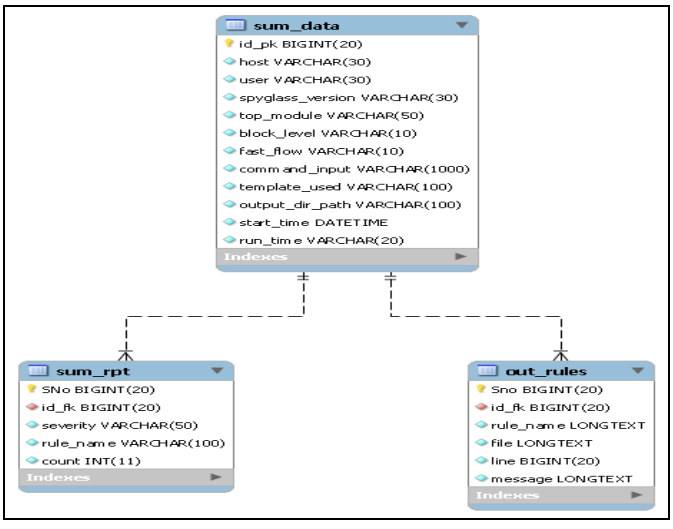

Figure 4 : SpotlightAnalysis Database ER Diagram

Spotlight Report Consolidation : A Spotlight Report Consolidation tool is a standalone tool written in Perl. The tool is run in Linux distribution to parse an input list of "spotlight_analysis" (contains the ouput reports of Spotlight Run). The output of the tool is a number of Excel Spreadsheets populated by the rule name, violation message, file in which violation found, line number on violation etc in corresponding columns.

Algorithm for Spotlight Report Consolidation

1. Check the options for validity
2. Call **detailed_reports**() function if option disable_detailed_report not used
3. To create summary.xlsx
   {
4. Shell script to parse summary file for getting Error, warning and info counts and finally populate csv files
5. Call **populateHash**(file_name) function
6. Call **insertToSummaryExcel**(summary_workbook_name)
   }
7. Zip all the workbooks and log file to a directory
8. Mail the zip folder to the user email id if provided using option mail_id.

The final deliverable to the client is a directory containing dependency packages, consolidation tool and user document for reference. The delivered directory contains following:

- Excel Perl Package
- Archive Perl Package
- Spt_ consolidate_reports.pl
- User_doc.htm
- Read Me

IV. RESULTS AND SCREENSHOTS

The result can again be divided in parts corresponding to the modules. The result of the web applications i.e Spotlight Dashboard and Spotlight Run Analysis can be presented by the screenshots of the web pages. The outcome of Spotlight Report Consolidation is shown with screenshots of different screens of Linux and Excel worksheets created as a result of tool run.

Input :

Following command is used to run spotlight over source files using various liberty files as input files

spotlight -test_spotlight_lib $SPT_INT_REL_AREA -"$SPOTLIGHT_REGRESSION" -verilog Sc32_post_scan.v Sc32_stp_proc0_stp.vs Sc32_stp_proc0_sms.vs -sglib density_libraries -sglib performance_libraries -sglib 9020pd -policies=erc,lsi_netlist_audit -rules=NGMemoryChecks -allow_module_override -connfile_debug=NGMemoryChecks -hhm_mode=yes -top Sc32 -template lsi_post_test -wdir ../other_files

output :

Above run will create below folders as output

- consolidated_reports
- spotlight.vdb.data
- spotlight_reports
  - spotlight_errors
    - ErrorAnalyzeBBox.OUT
    - HiDeFSetupRule.OUT
    - HiDeFSgdcCheck.OUT
  - spotlight_warnings
    - FaninLogicCone.OUT
    - WarnAnalyzeBBox.OUT
    - FanoutLogicCone.OUT
  - spotlight_info
    - Async_06.OUT
    - HiDeFUnloadedOutputPort.OUT

- CMD_sglib03.OUT
- INFO_1007.OUT
- summary.rpt
- spotlight.out
- spotlight_logs
- spotlight_spysch

Spotlight Dashboard

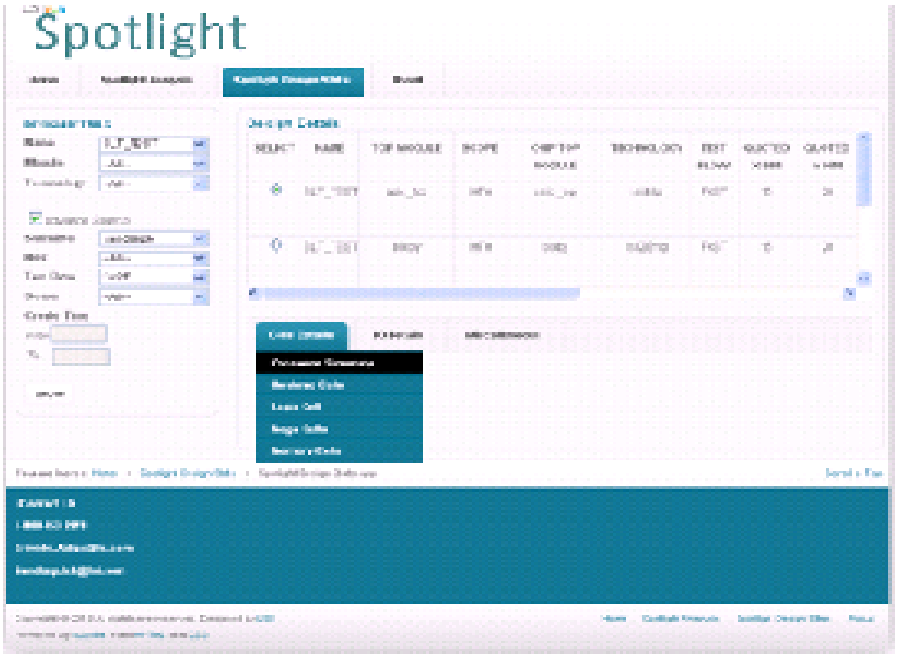

Figure 5: Spotlight Dashboard Page

Spotlight Run Analysis

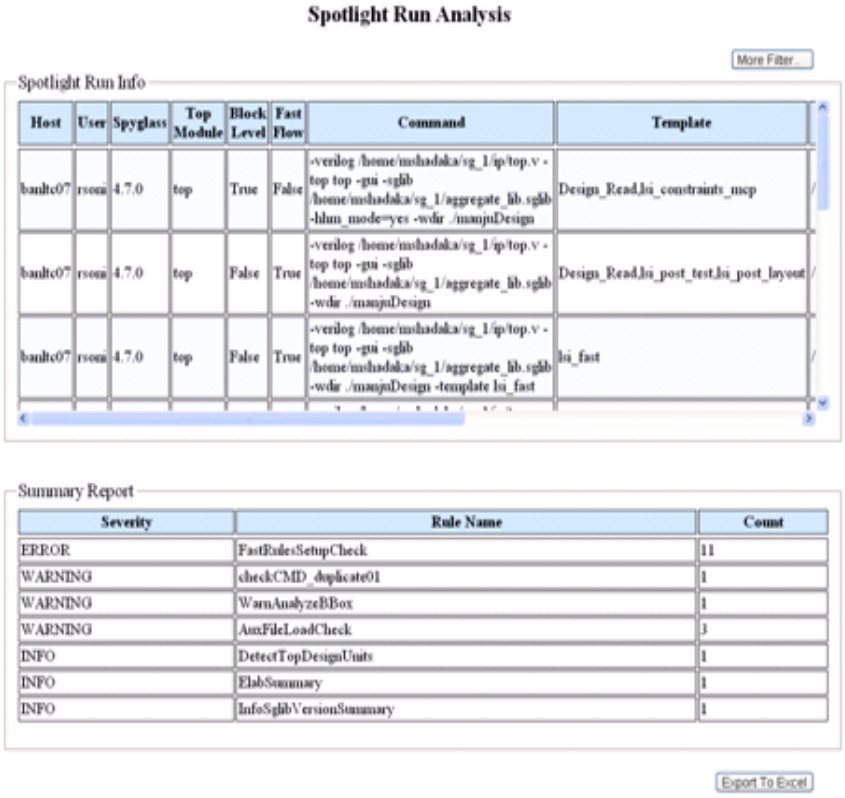

Figure 6: Spotlight Run Analysis result

Spotlight Reports Consolidation

Figure 7: Timeline for Spotlight Violations Count

V. CONCLUSIONS AND FUTURE SCOPE

Conclusion : This paper helped in getting a industry oriented experience of analysis of problem, planning for solution and development of professional applications and tools. A in-depth knowledge of web technologies as PHP, JavaScript, JQuery, Google Charts APIs' as well scripting languages like Shell script and Perl script is achieved. The Spotlight Infrastructure Development has evolved very nicely from its infant state to a well organized and developed state. All the three modules are under intense development and all have a very good scope for future enhancements.

Future Scope :There is a vast scope of development for all components in all modules . The web application oriented tool may accommodate more functionalities and features in itself while the tool can be enhanced further working and better user experience.

ACKNOWLEDGMENT

I am thankful to LSI Reasearch & Development Pvt.Ltd, Bangalore and my mentor Srinidhi Adiga, for his constant encouragement, guidance and support since beginning, which has transformed me as a "Subject Matter Expert". Last but not least, I would like to extend my gratitude to all those who indirectly supported me for the success of this paper.

REFERENCES

[1] file:///tools/spyg-5.1.1/SPYGLASS_HOME/htmlhelp/wwhelp/wwhimpl/js/html/wwhelp.htm
[2] A design and implementation of the active socket programming, K.L.E. Law and Leung, R.; Dept. of Electr. & Comput. Eng., Toronto Univ., Canada
[3] The Socket Programming and Software Design for Communication Based on Client/Server, Ming Xue ,Changchun Institute of Technology and Changjun Zhu ,HeBei University of Engineering, China
[4] Programming Perl, Tom Christiansen, brian d foy, Larry Wall, Jon Orwant.