

phpModeler – a web model extractor

Josip Maras

Faculty of Electrical Engineering,
Mechanical Engineering and Naval
Architecture,

University of Split, Croatia
00 385 21 305 832

josip.maras@fesb.hr

Maja Štula

Faculty of Electrical Engineering,
Mechanical Engineering and Naval
Architecture,

University of Split, Croatia
00 385 21 305 852

maja.stula@fesb.hr

Ivica Crnković

School of Innovation, Design and
Engineering,
Malardalen University, Vasteras,
Sweden

00 46 21 103 183

ivica.crnkovic@mdh.se

ABSTRACT

This paper presents phpModeler, a tool for reverse engineering of legacy php web applications that generates static UML diagrams showing resources the current web page is using, its functions and dependencies it has on other web pages. Once the models describing individual web pages have been generated, phpModeler can analyze them and generate dependency models that for each entity in every page model show all entities dependent on it. phpModeler can also be used to highlight the difference between page models – a feature that, when combined with a SVN repository shows the way how the current web page has evolved over time. phpModeler is a plugin for the Eclipse IDE.

Categories and Subject Descriptors

D.2.7 [Software engineering]: Distribution, Maintenance, and Enhancement – *reverse engineering tool*.

Keywords

Reverse engineering tool, maintenance, architecture recovery, web applications.

1. INTRODUCTION

In the last two decades, web applications have made a tremendous leap forward: from simple static web pages developed only in HTML to full-fledged dynamic web applications developed using server-side technologies such as php, asp.net, java, ruby, python and etc., that extensively use web services, databases and client-side technologies such as JavaScript, flash, and silverlight. Unfortunately, because of the very short time-to-market, these applications are often developed in an un-disciplined way; i.e. in php this approach can often lead to whole web pages developed as a mixture of SQL, php, HTML and JavaScript – four different languages in a single file, mixing business, presentation and data access logic. In turn, this often means that considerable effort is necessary to maintain these applications [1]. To improve development and maintenance efficiency also these pages should be modeled and their design should be designed using standard design or modeling languages, such as UML. Modeling and documentation of existing pages is as important as design of new applications.

In this paper we present a Reverse Engineering (RE) tool – phpModeler that statically analyses web application source code (php scripts, html pages, and JavaScript libraries) and generates

models that can be used as a basis for architecture recovery and that facilitate maintenance of legacy web applications.

2. PHPMODELER

phpModeler is divided into two modules:

- *Model generator module* that generates static UML diagrams representing resources the current web page is using (database tables, files, JavaScript libraries), its functions, properties and dependencies on other web pages.
- *Difference analyzer module* that shows differences between web page model versions.

Model generator module comprises the following modules: code processor, dependency analyzer and UML generator.

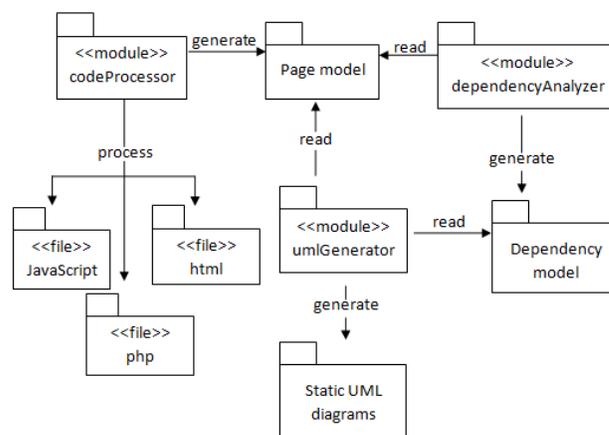


Figure 1 Model generator modules

Code processor is a page parsing module that parses php, html and JavaScript code. Its main functionality is to generate a web page model based on the adapted Connalen UML extensions[2].

Dependency analyzer is a module that analyses all generated models and that for each web page entity (JavaScript library, database table, file, php library, web page, etc.) finds all other entities dependent on it; for every database table these models show all web pages that access them; for function libraries they show web pages that use those functions, etc. This simplifies the process of change, because for each change the maintainer can easily see where the change propagates.

UML generator is a module used to generate UML diagrams in a standard format used by the majority of UML tools [3]. As an input it uses models generated by the code processor and the dependency analyzer modules.

phpModeler can also be used to highlight the differences between page models. For that functionality it uses the difference analyzer module and the EMF Compare tool [4]. We have connected this module to a SVN repository and in that way gained the functionality of modeling page evolution. The model differences are shown in a way similar to [5].

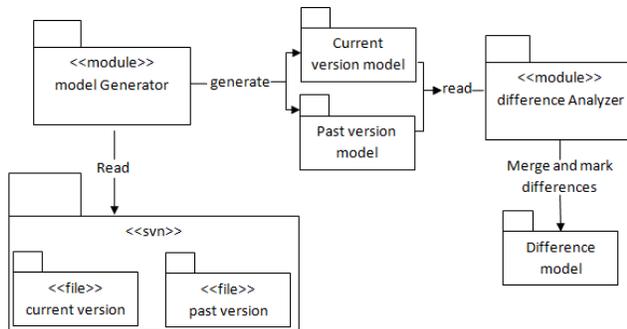


Figure 2 Modeling page evolution with phpModeler

The process of a web page model generation goes as follows. In phpModeler, the user selects the root folder or individual web pages and starts the modeling process. The code processor module parses the selected web pages and all referenced web pages, and generates their models. The dependency analyzer module then analyses the generated models and for each entity (database table, file, web page) generates models showing other entities dependent on it. After the model generation phase, the UML generator component generates UML diagrams.

3. RELATED WORK

In the last decade several web application RE tools have been made, some of which are: WARE [6], ReWeb [7], WebUml [8], Enterprise Architect[9], Visual Paradigm for UML[10] .

WARE is a RE tool that adopts the UML extensions proposed by Conallen and that uses static, dynamic and behavioural analysis to generate class diagrams representing the architecture of web applications; sequence and collaboration diagrams to represent the dynamic model and use case diagrams to represent web application behaviour.

ReWeb is a RE tool that statically analyses web application source code and generates models that represent the web application as a graph structure.

WebUml is a RE tool that generates class diagrams, used to describe the structure and components of a Web application and state diagrams that represent the behaviours and the navigational structure of the Web application. Web UML models a web application using modified Conallen UML extensions.

Enterprise Architect is a commercial general modeling tool that, among other things, enables RE of object-oriented php into

UML and change synchronization between source code and UML models. It can be used to create class diagrams representing classes defined in php, and sequence diagrams that show what PHP classes use and how they are used.

Visual Paradigm for UML is a commercial UML CASE tool that can reverse engineer object-oriented php into UML models.

phpModeler’s functionality is similar to other web application RE tools based on the Conallen’s web extensions for UML. It offers an automated way to generate models that facilitate architecture recovery and maintenance of legacy web applications. Compared to other, similar web application RE tools, the benefits of using phpModeler are:

- phpModeler simplifies web page maintenance because it can be used to generate dependency models that show the way in which change propagates through the web application.
- phpModeler has the capability to highlight the difference between page models. This functionality combined with the capability to access to SVN repository can be used to show web page evolution.
- phpModeler can be used from the Eclipse IDE, which is the IDE often used for developing php applications – so one gets the benefit of modelling and coding web pages from the same environment.

REFERENCES

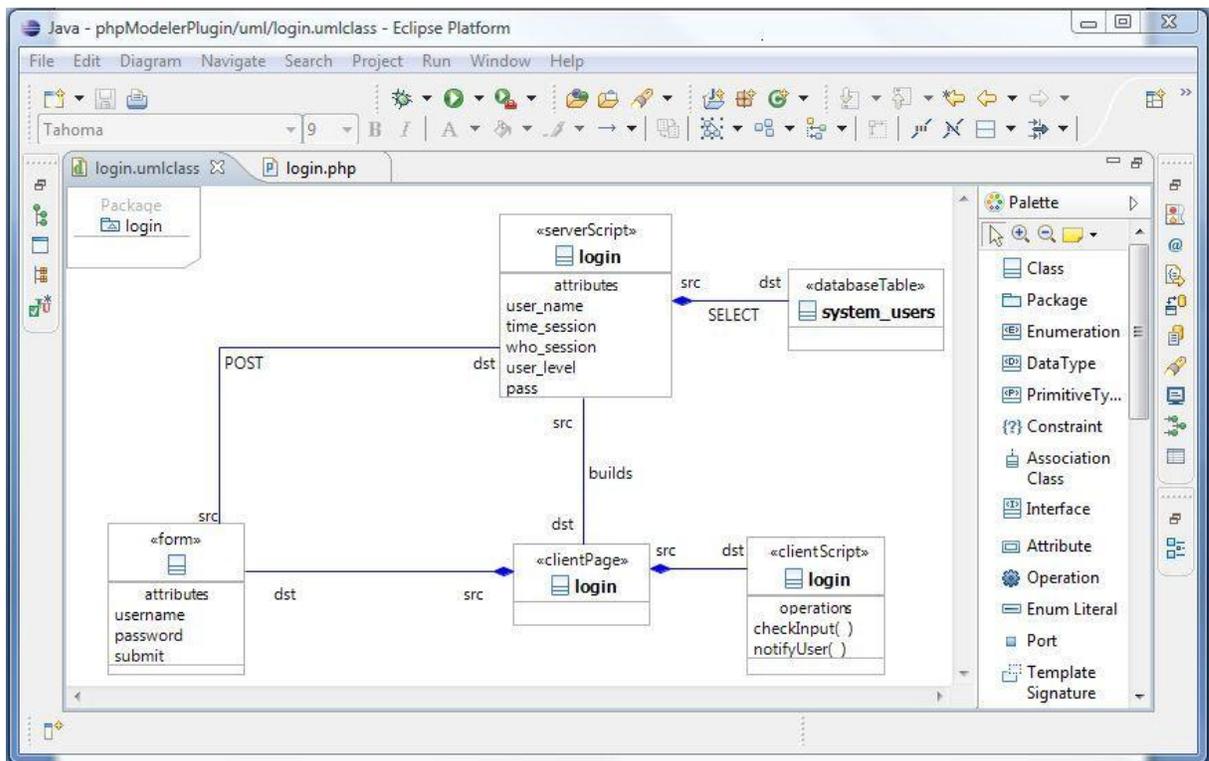
- [1] Mikkonen, T. Taivalsaari, A.; 2008; Web Applications – Spaghetti Code for the 21st Century; Software Engineering Research, Management and Applications
- [2] Conallen, J.; 2003; Building Web Applications with UML Second edition. Addison-Wesley
- [3] XMI – XML Metadata Interchange; 25.05.2009; <http://www.omg.org/technology/documents/formal/xmi.htm>
- [4] Niere, J.; 2004; Visualizing differences of UML diagrams with Fujaba; In Proc. Of the 2nd Fujaba Days
- [5] EMF Compare; 15.06.2009. ; http://wiki.eclipse.org/index.php/EMF_Compare
- [6] Di Lucca, G.; Fasolino A.R.; Tramontana P.; 2004 Reverse engineering Web applications: the WARE approach, Journal of Software maintenance and evolution: Research and practice
- [7] Ricca F., Tonella P.; 2001; Understanding and restructuring Web sites with ReWeb; IEEE Multimedia
- [8] Ballettini, C.; Marchetto A.; Trentini A.; 2004; WebUml: Reverse Engineering of Web Applications, ACM Symposium on Applied Computing
- [9] Enterprise Architect; 25.05.2009; www.sparxsystems.com.au/platforms/php_uml.html
- [10] Visual Paradigm for UML; 25.05.2009; www.visual-paradigm.com

Appendix: Tool Demonstration Description

The demonstration of the phpModeler tool will be focused on three examples showing reverse engineering of a user authentication process. The first example is intended to illustrate the basic principles, the second is a more complex example with dependencies that illustrate tool usage for complex web pages, and the third example will demonstrate management of differences between different module versions.

Example 1: Modeling web page static structure

This is a simple example that will show static models generated by the phpModeler tool. It will also be used to introduce the basic concepts of the web page modeling UML extensions.



1. Figure 1 Static diagram of the login page example

Figure 1. shows the static diagram of a simple login web page. In this example the login page is a php server script that builds a html client page containing a web form by which the user submits user credentials back to the login page. The login page then reads sent user data and tries to find it in the “system_users” database table. If it finds the matching data it redirects the user to the main part of the system and if not the user is notified that the used credentials are invalid.

Example 2: Generating dependency models

In this example several web pages will be modeled, and for each web page entity of each modeled web page, dependency models will be generated.

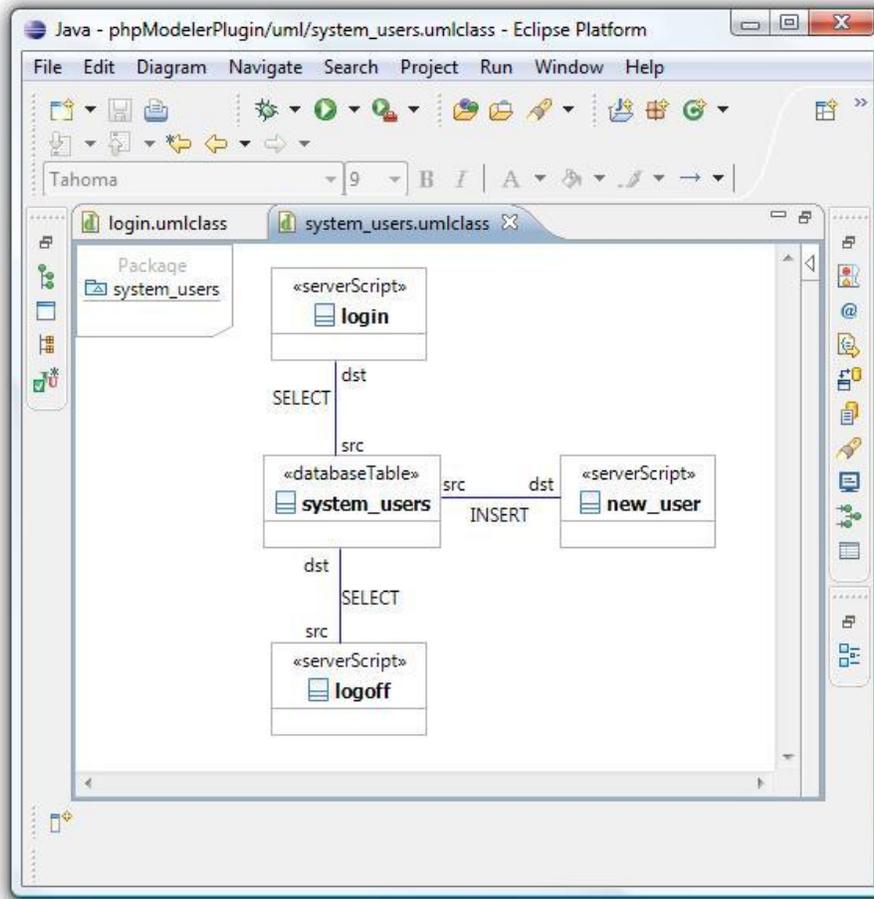


Figure 2 Example dependency model

Figure 2. shows one of the results of the dependency modeling phase: php server scripts that are dependent on the “system_users” database table. With these diagrams the person in charge of system maintenance can easily see that if the “system_users” database table has to be changed, that change will most likely propagate to the “login”, “logoff” and “new_user” php scripts.

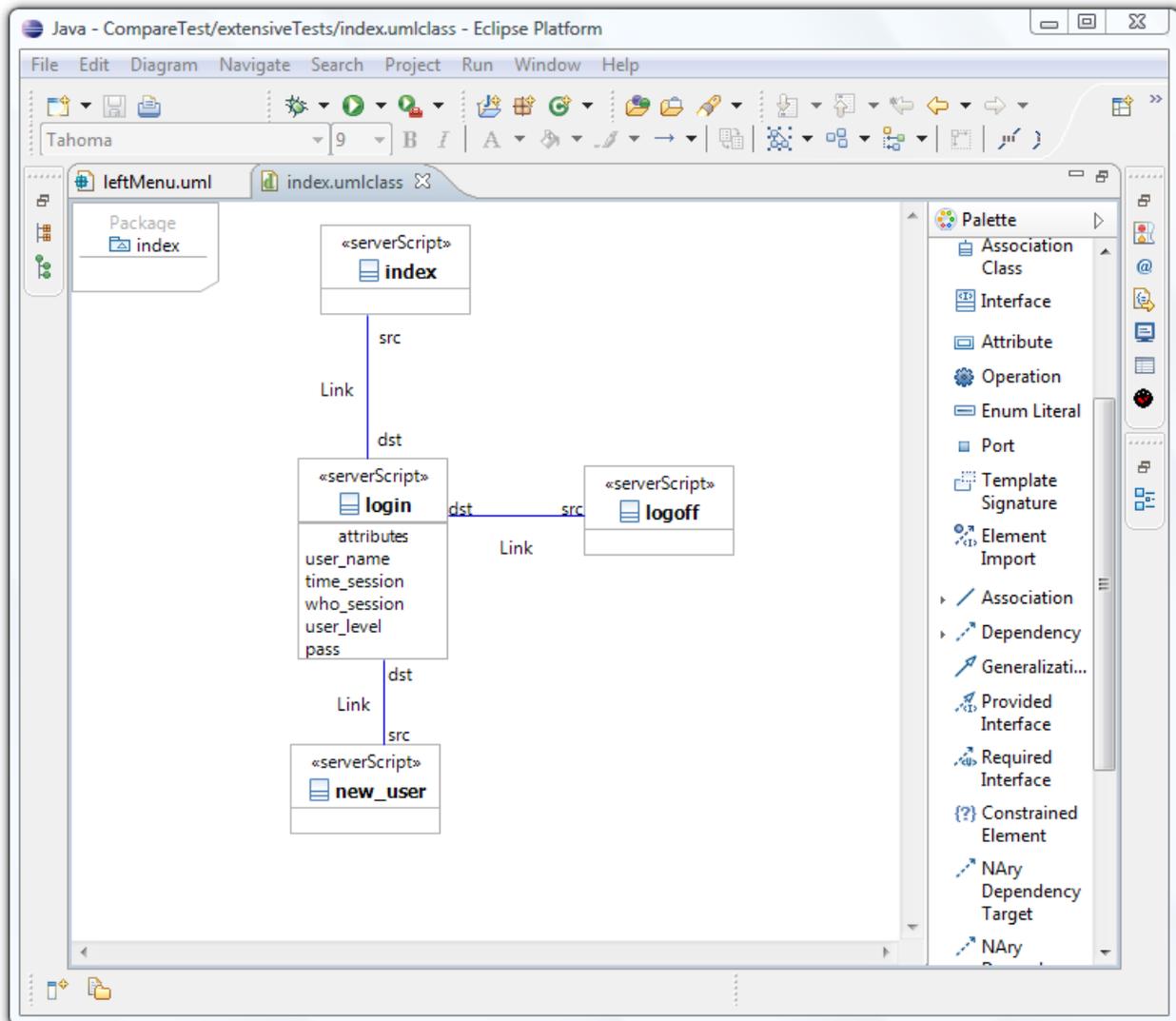


Figure 3 Example Dependency model

Figure 3 shows web pages dependent on the login page. If the user opens the main index page and is not logged in to the system the index page redirects him/her to login page. The same happens when trying to access the new_user and the logoff web page.

Example 3: Modeling page evolution

This example will show how page evolution can be modeled with phpModeler. Several versions of the same web page will be checked out from the subversion repository and their differences will be modeled.

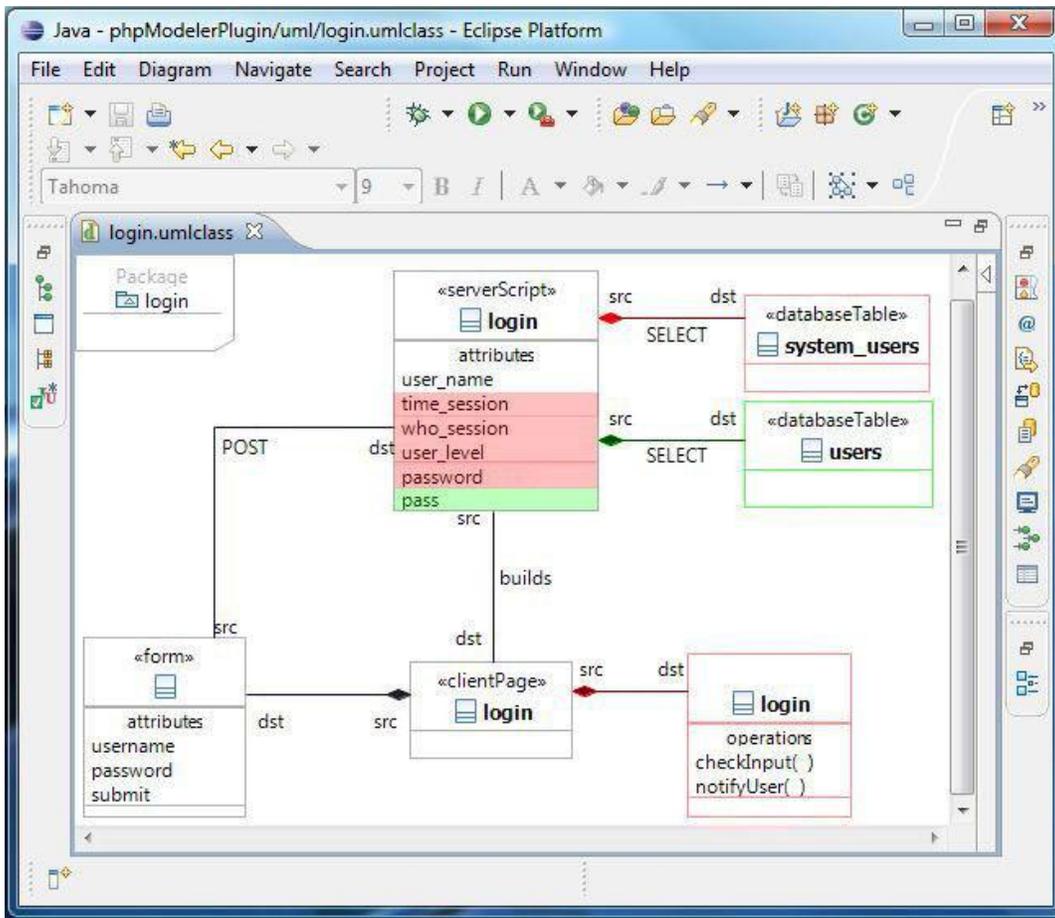


Figure 4 Model Difference Example

Figure 4. shows the results of the model comparison phase for two versions of the login web page. Two models are merged and parts detected as similar are shown in black color, parts found only in the earlier version are shown in green and parts found only in the newer version are shown in red.

Current tool state

The functionalities of generating static web page models and dependency models are completed and the functionality of displaying model differences is currently in a refinement phase. The tool is available on the <http://www.fesb.hr/~jomaras/phpModeler> web page as a preliminary version.