

Applied Clickstream-Analysis with phpWebSite

René C. Kiesler <<http://www.kiesler.at/>>, Martr. Nr. 9525452, Study Code 933, 2005-04-30

Abstract

This is an overview of known Clickstream-techniques, phpWebSite, the Visitors Module and how one could merge all of them.

When running a website the webmaster or the supervising manager usually wants to know the impact on that particular site. Until now, webmasters only utilize log file analyzers like Analog, AWStats or Webalizer. That way, they can find successful pages but not the whole story behind them. How did a visitor reach these successful pages? Which pages did they visit next? Is there a noticeable pattern behind the visit of certain pages? A Clickstream analysis would give answers to all of that.

Content

Applied Clickstream-Analysis with phpWebSite	1
Abstract.....	1
Content	2
Definitions.....	3
Hit.....	3
Page Views	3
Session	3
CMS (Content Management System)	3
PhpWebSite.....	3
What does that mean?	4
The Visitors Module	4
Visitors: Real-time mode	4
Visitors: Analysis mode	5
Classifying Web-metrics	5
Describing Clickstream-analysis approaches.....	6
Click / Session / sub-session fact approaches	6
Hypertext Probabilistic Grammar	7
The Hybrid approach.....	8
How to implement that	8
Visualization	9
Conclusion	9
Further Reading	10
Referred Publications	10

Definitionsⁱ

Hit

"The sending of a single file, whether text, graphic, audio or other type of file. When a page request is made, all elements or files that comprise the page are recorded as hits on a servers log file. While there is no accurate formula for determining the number of visitors to a page or site based on the number of hits -- one visitor could go back and forth twenty times or twenty people could visit a single time each -- a hit at least indicates somebody was there. Thus, hits can be far more valuable than the tracking devices in any other media."

Page Views

"Number of times a user requests a page that may contain a particular ad. Indicative of the number of times an ad was potentially seen, or "gross impressions." Page views may overstate ad impressions if users choose to turn off graphics (often done to speed browsing)."

Session

"A series of transactions or hits made by a single user. If there has been no activity for a period of time, followed by the resumption of activity by the same user, a new session is considered started. Thirty minutes is the most common time period used to measure a session length."

CMS (Content Management System)ⁱⁱ

"A program for generating, organising, storing and publishing content for a dynamic website. The use of a content management system means that website content can be kept up to date by editors and other contributors who do not have to have special knowledge about HTML or other Web technologies."

PhpWebSite

phpWebSite is an open source CMS that has a rather long historyⁱⁱⁱ. It is based a branch of the very popular Nuke by Francisco Burzi and maintained by the Appalachian State University. phpWebSite can be downloaded from Appstate's site^{iv}.

The classical Nuke concept differs quite a bit from other CMSs. One of the differences that you might notice very early is the module-orientation. Other CMSs are often sitemap-, or category-oriented. What does this mean? Let me explain this with a rough sketch of the phpWebSite architecture.

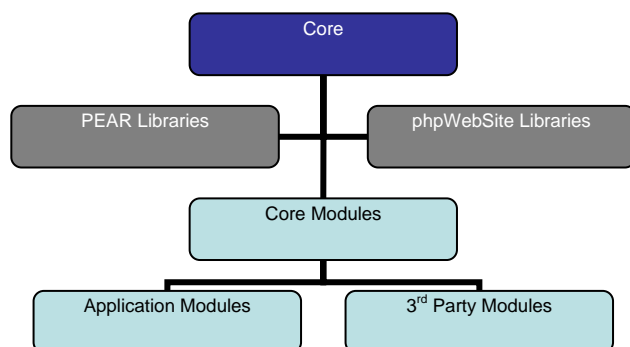


Image 1: phpWebSite is very modular. Its Core only has a few hundred lines of code; the rest is divided amongst libraries and modules.

phpWebSite has a rather small core that is only about 500 lines of code. It is mainly used for dispatching and bootstrapping modules. A minimal installation, consisting only of the

core modules,¹ doesn't seem to do anything but manage users, themes and categories. Installed only with the core modules, phpWebSite does not provide any means of actually entering content.

That's where the Application- and 3rd-Party modules come in. Developers can use all the core modules and libraries, which gives them a rich API. Note also that about everything a user touches in phpWebSite is a distinct module. Often, content also has a category, provided by "Fatcat."

What does this Mean?

There is no defined site structure. Every module can be jumped to directly by a standardized URL²; every module has a well-defined meaning. This gives us a benefit when analyzing user-behaviour: with the module-name and a sequential list of URLs alone, we already have more information than we ever could get through a simple logfile analysis.

But there's more. By watching the search-module, for example, we can find out the terms used to find documents within our site. The user-module tells us whether the user is logged in or not. If a user is, we also know their email-address as well as whether they are Admin³, Deity⁴, of a certain user-group or have special rights.

The language-module can tell us if a (registered) user prefers a certain language and the layout-module, whether they prefer a certain design.

The Visitors Module

One of the many phpWebSite 3rd-Party Modules is Visitors^v. It is basically a mixture of a community- and analytics-tool. It has a real-time and an analysis mode. The real-time mode is shown all the time, whereas the analysis part is only available by demand from deities.

Visitors: Real-Time Mode

The real-time shows, depending on the user-role:

- How many people are online and how many of them registered (as guests)
- How many guests are online and who the registered users are (for registered users) or even
- Online users together with current module as well as their respective IP-address (for deities). And a link to a detailed-view, which shows the user-agent string, the resolved hostname, and the Clickpath for the current session of the selected user.

¹ Approval, Boost, Controlpanel, Fatcat, Help, Language, Layout, Search, Security and Users

² For example, to access the boost module, you'd call `./index.php?module=boost`

³ Only Admins are allowed to enter content that's not a Forum thread or a comment

⁴ Deities are the Superusers in phpWebSite. They can manage users, grant rights, change the layout, install modules and so on.

Visitors: Analysis Mode

The analysis-mode of Visitors as of the current version (1.1) has three dimensions:

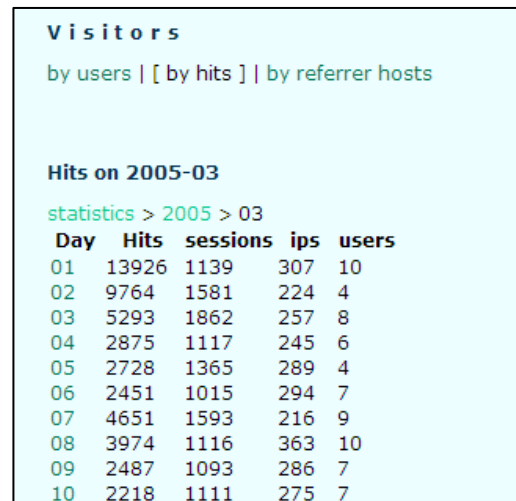
- hits (strictly speaking: page views)
- users and
- referrers

All three can be drilled up and down, the Visitors module defaults to the hits of the current month. Visitors supports a couple of intervals:

- all (e.g. by clicking on "statistics")
- per year (e.g. by clicking on "2005")
- per month
- per day (e.g. by clicking on "01" or "10")

The hits-screen includes:

- hits (page-views)
- distinct sessions
- distinct IPs
- distinct users



The screenshot shows the 'Visitors' module interface. At the top, it says 'Visitors' and 'by users | [by hits] | by referrer hosts'. Below that, it says 'Hits on 2005-03' and 'statistics > 2005 > 03'. The main part of the screenshot is a table with the following data:

Day	Hits	sessions	ips	users
01	13926	1139	307	10
02	9764	1581	224	4
03	5293	1862	257	8
04	2875	1117	245	6
05	2728	1365	289	4
06	2451	1015	294	7
07	4651	1593	216	9
08	3974	1116	363	10
09	2487	1093	286	7
10	2218	1111	275	7

The users' and the referrer hosts list the users / referrer hosts of that period together with their amount of hits.

Classifying Web-Metrics

Clickstream Analysis is in essence a technique that shows how visitors navigate through a site. But there are a couple of other ways of measuring the usage pattern of a site in a technical way:

- 1) **Not at all.**
This of course would be the easiest way. You just put your site online and don't care about how many people visit it.
- 2) **With a page hit counter⁵.**
Various companies offer a page hit service for free. You insert a bit of code in your site and see a page hit counter in your Browser in return.
- 3) **Through logfile-analysis⁶.**
This option is a bit more demanding than the previous two. Here, you need access to a web server to collect its log files. They usually give you a line of information per page hit, but don't relate the hits within a session.
- 4) **Through session analysis.**
As of the current version 1.1, the Visitors module we will discuss later on supports session analysis. Hits are not viewed independently but sessions are taken in

⁵ For example: <http://www.webcounter.com/>, <http://www.amazingcounters.com/>, <http://xcounters.com/>, <http://www.rapidcounter.com/>, etc.

⁶ For example with <http://www.analog.cx/>, <http://www.awstats.org/>, <http://www.mrunix.net/webalizer/>, ...

account as well.

5) **Through Clickstream analysis**⁷.

Basically, a Clickstream is nothing but an ordered list of events that occur within a session. There are a couple of ways of storing this information, the Click fact and Session fact^{vi}, Sub-session fact^{vii} and the Hypertext Probability Grammar approach. There's also a *Hybrid-approach*^{viii}, which combines HPG with the click fact approach. Visitors, as of the current version 1.1, provides a click fact table but is very limited in reporting. More about that later on.

6) **Through Mouse pointer tracking.**

It is possible to track the cursor movement of users through JavaScript, Flash, Java or some other way and send the movements together with their timestamps back to the server. While this gives rather good information, it is a rather big effort.

7) **Through Eye tracking.**

Some Usability-Labs have facilities to track eye-movement. While this is not feasible on a larger scale, this might be interesting for large corporations who want to know about the reactions to their sites and how people use them.

8) **Via Bio-feedback.**

In a Lab, it would of course be possible to stick some electrodes on the participants as well. This obviously exceeds the scope of this paper.

Describing Clickstream-analysis approaches

Click / Session / sub-session fact approaches

These three are all database-driven. The *Click-fact* approach works like a log-file. It has one line per user-hit. Visitors works that way and stores information that differs a bit from the common logfile format defined by Apache:

- id (unique id)
- ip (either REMOTE_ADDR, part of HTTP_X_FORWARDED_FOR or HTTP_CLIENT_IP, calculated)
- timestamp
- *module (current module as of hit)*
- session_id
- *user (name of user)*
- *user_id*
- REMOTE_ADDR, REMOTE_PORT
- HTTP_CLIENT_IP
- HTTP_X_FORWARDED_FOR
- HTTP_USER_AGENT
- HTTP_REFERER
- HTTP_ACCEPT_LANGUAGE
- QUERY_STRING, REQUEST_URI
- HTTP_HOST

⁷ For example with <http://www.clicktracks.com/>, <http://www.datanautics.com/>, <http://www.omniture.com/s2/>, <http://www.sas.com/solutions/webanalytics/>, <http://www.sane.com/products/NetTracker/>, <http://www.spss.com/pwa/>, ...

phpWebSite gives us the attributes written in italics (module, user and user_id); all the others come from either php or MySQL.

While one can calculate the order of events in a session through the timestamp or the (strictly monotone rising) id, doing so is quite expensive. Add parsing of the QUERY_STRING, which contains, for example, the visited items, and you'll soon find out that analysis on a large scale is not feasible that way. Then again, adding meta-info (like our user-name) is very easy.

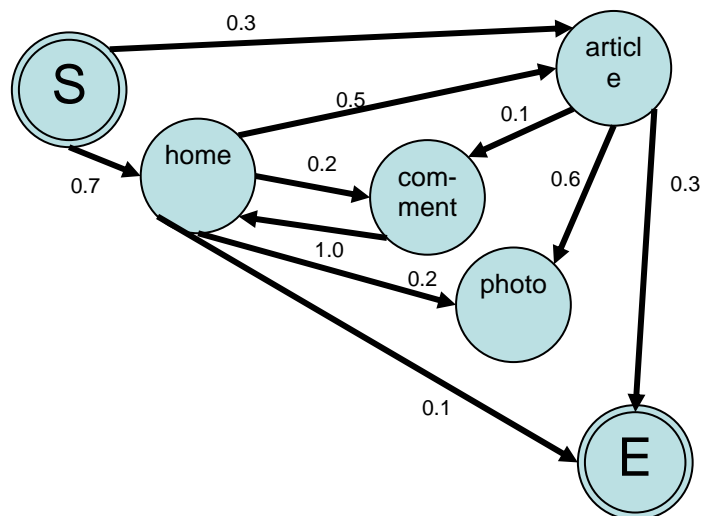
The *Session-fact* approach scales better. But alas, it also takes away a lot of information as it only stores the start and end page of a session, together with referrer, user agent and a timestamp. This of course makes us unable to do a detailed click analysis, as we don't know about the pages visited between start and end page. On the other hand, this approach is very fast.

That's where the *Sub-session fact approach* comes in. It stores all possible combinations of start- and end pages in a session, which gives on the one hand, the high performance of the session-fact approach, and on the other hand, the granularity of the click-fact approach without the speed-penalty. Then again, this of course can take up quite a significant amount of space in our database.

Hypertext Probability Grammar

The HPG works on a more abstract level. It sees our site as a couple of states (nodes) with a bunch of edges. Each edge contains a value which indicates the probability of the start node being left in favour of the end node reached by the edge.

Let's have a look at an example. Here, I've drawn a hypothetical HPG for 4 phpWebSite modules. One can easily see that this syntax can get quite complex. We only see 4 states here, whereas my installation on <http://www.kiesler.at/> currently has 33 modules installed, which of course can all interact with each other. But that's not all.



What if we want to have additional meta- data in our model as well? Take for example the Clickpath of someone coming from the US, from a certain referrer or even a distinct IP. I currently have

- 9734 distinct IPs and
- 9934 distinct HTTP_REFERRERS

in my Visitors table. Potentially, these two meta-attributes would lead us to 9734*9934*33 states. That's 3.191.019.348 states, more than 3 Billion! This is of course is not feasible as soon as you want to differentiate multiple meta-informations.

That's where the next approach comes in.

The Hybrid approach

If you've understood the click- and the HPG-approach, you've understood the Hybrid approach as well. Basically, the Hybrid approach only creates a very simple, restricted syntax based on user constraints. It then enriches the result with various meta-data.

Jespersen, Thorhauge and Pedersen for example took demographic data like "estimated age", "estimated income", and so on to find out about the behaviour of visitors to the Zenarias company website.

I think that the Hybrid approach would be best suited implemented by Visitors. But first, we'd need a bit of caching implemented to get speedy response times.

How to Implement It

We have for example only request URLs in the database, together with the module names. What we certainly would need is at least an item number and maybe even an item title.

Also, note that we don't provide any pre-calculated DNS lookups right now, we do them just in time. That's of course very slow when done in a large scale as needed here. From my point of view we'd need to cache at least:

- Element⁸, together with previous and next Element, including Module and Category, per Session
- Module, together with previous and next Module, per Session
- Category, together with previous and next Category, per Session
- IP to DNS mapping
- maybe Country (could be mapped from the IP)
- Browser Type
- Operating System
- Is the user a Bot (if so, which one)?⁹
- Time between clicks
- Time between sessions on a per-user basis
- Amount of clicks in that session

We would also want to analyze the user search behaviour, as this can give insights about needed content / navigation improvements.

- What were the search-strings used for users coming from a search engine? Which page did he reach through his search? Does it match his intentions?
- Which search-strings did the user use on our internal search, where did it lead him?

From all of that we could, on the one hand, categorize users by interest, on the other hand, even guess about their intent^{ix}:

- Is he/she just browsing?
- Is he/she searching something in particular?
- Was that search successful?
- Does he/she want to write about something?

⁸ An element would be a content-item, a category or a menu-item.

⁹ can be determined through a combination of agent-analysis and DNS-analysis

Further versions of Visitors could also contain a lot more of information, which we could for example gain through the intelligent use of JavaScript:

- Screen-Resolution¹⁰
- Content-Resolution¹¹
- Colour depth
- JavaScript yes / no, which version?
- Java yes / no, which version?
- Flash yes / no, which version?
- How many sites are in the history?

Visualization

As soon as we have the data at hand, we could visualize it as well. Visitors, as of the current version 1.1, doesn't have any graphical output. It only shows a couple of tables with numeric or other collected information in them.

The ClickViz^x tool, as implemented by Blue Martini, is capable of doing this. The tool shows:

- site topology
- traffic flow and
- segmented site traffic data, based on meta-data

Basically, the visualization of Blue Martini equals the HSG seen before, together with a bit of meta-data. As long as we don't want to animate the HSG, we probably could implement something similar with the php GD library¹² in Visitors.

We could of course implement other, more trivial statistics as well. Like Top-Referrer along a timeline and flow from top-referrers to pages on our site.

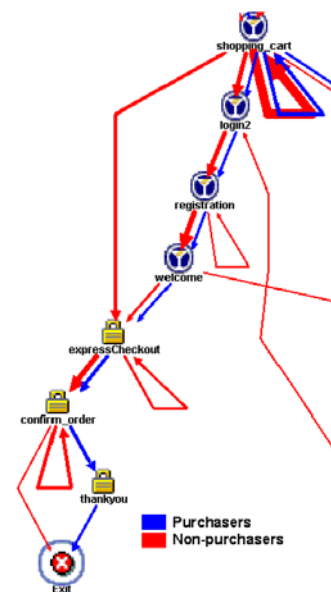


Image 4: this clickstream-visualization shows the checkout-process of a shopping-cart

Conclusion

By implementing all of the bullets mentioned two sections earlier, Visitors would be able to give webmasters much more high quality information as compared to regular logfile analyzers. Even when compared to common Clickstream-analysis tools¹³, Visitors could be an interesting choice, as it is tightly integrated in phpWebSite and knows, for example, about the module-concept.

Given the time and proper funding, the needed information should be collectable and help deities optimize their sites.

¹⁰ What's the actual resolution the user runs his/her monitor at?

¹¹ How much content can be fitted in the browser-window at the moment?

¹² What is the GD-library? see <http://www.boutell.com/gd/>

¹³ I can't compare them in this paper as we have a limit of 2.500 words

Further Reading

- Montgomery, Li, Srinivasan, Liechty (2004):
"Modelling Online Browsing and Path Analysis Using Clickstream Data"
- Computer Adept (2000):
"Data Warehouse Architecture and Project Management: Clickstream Data Warehousing"
- Kohavi (2000):
"An Ideal E-Commerce Architecture for Building Web Sites Supporting Analysis and Personalization",
<http://www.bluemartini.com/>

Referenced Publications

- i Lazworld.com (2005):
"Internet Marketing Glossary",
<http://www.lazworld.com/glossary.htm>
- ii Ameris (2004):
"Glossary of Terms",
http://www.ekeda.com/glossary_of_terms.cfm
- iii René C. Kiesler (2005):
„Comparing TYPO3 with phpWebSite“,
<http://www.kiesler.at/article174.html>
- iv Appalachian State University (2005):
„phpWebSite“,
<http://phpwebsite.appstate.edu/>
- v René C. Kiesler (2005):
„Visitors for phpWebSite“,
<http://www.kiesler.at/article148.html>
- vi Kimball, Metz (2000):
"The Data Webhouse Toolkit"
- vii Andresen, Giversen, Jensen, Larsen, Pedersen and Skyt (2000):
"Analyzing Clickstreams using sub-sessions"
- viii Jespersen, Thorhauge, Pedersen (2002):
„A Hybrid Approach To Web Usage Mining“
- ix Bucklin, Lattin, Ansari, Bell, Coupey, Gupta, Little, Mela, Montgomery and Steckel (2001):
"Choice and the Internet: From Clickstream to Research Stream"
- x Brainerd, Becker (2001):
„Case Study: E-Commerce Clickstream Visualization“,
<http://www.bluemartini.com/>