WEB WORKLOAD GENERATORS
A Survey Focusing on User Dynamism Representation

Raúl Peña-Ortiz
Computer Engineering Department (DISCA), Polytechnic University of Valencia, Camí de Vera s/n, Valencia, Spain

Julio Sahuquillo, José A. Gil, Ana Pont
Computer Engineering Department (DISCA), Polytechnic University of Valencia, Camí de Vera s/n, Valencia, Spain

Keywords: Web performance evaluation, Dynamic users' navigation, Dynamic contents, Modeling dynamic workload.

Abstract: The evolution of the World Wide Web from hypermedia information repositories to hypermedia distributed applications and services oriented architectures (SOA) has introduced new features in the current and incoming web. An important feature is the dynamism of its contents and services, which induces the dynamism of the client behavior. This feature represents a major constrain when modeling and generating current web workload.

In this paper, we first review the state of the art for web workload generation, focusing on the approach based on workload models. After that, we analyze a representative subset of the state of the art workload generators that use workload models, concentrating on those model characteristics that represent dynamism in the workload generation. The study reveals that five generators present some capabilities to reproduce this dynamism, but only the *GUERNICA* approach improves the dynamic workload generation by using users' behavior models. Finally, we discuss *GUERNICA* and describe how it generates dynamic workload by addressing both user behavior and workload distribution.

1 INTRODUCTION

As a consequence of the recent and incessant changes in web technology, new types of hypermedia distributed applications, hereafter web applications, and distributed services have become more frequent and familiar to web users. For instance, e-commerce, on-line booking systems, on-line auction systems, Google Mashups and Amazon Web Services are only few examples that manifest how web sites are evolving from hypermedia information repositories to hypermedia distributed applications and distributed services. Due to the increase and popularity of these applications and services, typical of the Web 2.0, dynamic contents and services are becoming more and more frequent, suggesting the review of widely accepted paradigms and models. The Web 2.0 technology involves important changes in the web site architecture (Treese, 2006), for instance, AJAX, flash, tagging, social network, or services oriented architectures (SOA) (Erl, 2005). These changes are more relevant and meaningful in the recently incoming Web, also refereed to as Future Internet (Tselentis, 2009). This evolution also implies significant changes in web users' behavior (O'Reilly, 2005) because new applications promote a dynamic navigation, providing an experience much closer to desktop applications than the traditional static web pages, e.g., syndication of website content or web information systems (Houben et al., 2004).

As any system that is continuously changing, both in the offered applications and in its infrastructure, performance studies are important to evaluate novel proposals when designing new web-related systems, such as services oriented architectures, web servers, or proxy policies. These studies require from the use of an accurate and representative workload models in order to guarantee the representativeness and validness of the results.

In the case of the Web, an appropriate web workload model is a major concern in order to reproduce client workloads. These models can be used to evaluate the system performance and to test applications and services since they reproduce the behavior of their

Peña-Ortiz R., Sahuquillo J., A. Gil J. and Pont A..
WEB WORKLOAD GENERATORS - A Survey Focusing on User Dynamism Representation.
DOI: 10.5220/0003274801190126
In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST-2011), pages 119-126
ISBN: 978-989-8425-51-5

potential clients. In this context, a web workload model should reproduce the HTTP requests that a real web server typically receives in daily working sessions.

Nevertheless, the implicit dynamism in users' behavior and in the generation of web content makes hard to design accurate web workload models that represent users' navigations. On the one hand, users require customized information and advanced services from web applications. This requirement implies that the contents are dynamically generated from different data sources. On the other hand, along a typical navigation, users use to select pages to visit depending on the previously visited ones, and particularly on their contents or characteristics such as the response time. Thus, the dynamism in the content generation implies that the users' navigation may differ depending on the contents generated by the web application during a given period of time. In summary, the dynamic content introduces dynamism in users' navigation, that is, it produces dynamic users' behavior. For example, a high percentage of users' navigation sessions begins searching a dynamic resource in a specialized site and then it continues visiting one or more sites, depending on the results of the previous searches.

In general, the main challenges when modeling web workloads are: i) how to model users' dynamism, ii) how to represent the different roles that users play in the web, and iii) how to model continuous changes in users' behavior (Weinreich et al., 2006).

In a previous work (Peña-Ortiz et al., 2005) we proposed the *GUERNICA* approach to model the dynamism of the web workload based on users' behavior models. This paper extends that work in several ways. First, we review the state of art on web workload generators, software products that are designed and implemented to generate web workload, and classify them in three main groups according to their capability to generate web workload and/or to model the users' dynamic behavior. Second, we compare *GUERNICA* to other web workload generators. The study reveals that five web workload generators present some capabilities to reproduce this dynamism, but only our approach improves the dynamic workload generation by using users' behavior models.

The remaining of this paper is organized as follows. Section 2 reviews, analyzes and classifies a representative subset of workload generators. Section 3 summarizes highlights some GUERNICA characteristics. Finally, Section 4 presents some concluding remarks.

2 WEB WORKLOAD GENERATORS

Dynamic web applications and services have induced continuous changes in users' navigation patterns, thus making it difficult to characterize the web workload.

Currently, we can reproduce the client behavior by using either traces or workload models. Traces log the sequence of HTTP requests and commands received by a web application or a web server during a certain period of time and under specific conditions. Therefore, traces are obtained in a particular environment (e.g., server process speed or network bandwidth) for a specific application. This means that if any system parameter changes (e.g., new contents in a news portal), the resulting trace might differ. Therefore, the main concern in trace-based study is the representativeness potentially achieved, especially when the requests received by a given web server exhibit a high variability because they only reproduce a subset of their clients. Consequently, these models are not appropriate to model changes in the client behavior.

On the other hand, parameterizable workload models are abstractions of the real workload that hide those characteristics not relevant for a particular study. Unlike the previous models, this kind of models is accurate enough when reproducing the client behavior or when evaluating the performance of web applications. Parameterizable workload models generate sequences of HTTP requests similar to the real sequences and different scenarios can be configured by properly setting the corresponding parameters.

The representation of the dynamic users' behavior has been addressed in some web workload characterization studies when defining dynamic web site benchmarks (Amza et al., 2002) or when studying the performance and scalability of the technology used to develop dynamic sites (Cecchet et al., 2002). However, user dynamism is complex by nature, and current results are still far from being precise and satisfactory. Therefore, more research efforts must be done in this direction in order to provide more accurate tools to model and evaluate web performance. In this sense, the Customer Behavior Model Graph (Menascé and Almeida, 2000) was introduced to be used as input to a workload model of an e-business site.

Workload models are the basics of workload generators. They are software products designed and implemented to generate HTTP requests. They are flexible tools useful to address tuning or capacity planning studies but, unfortunately, current generators only represent the users' dynamic behavior in a partial way; for instance, they cannot represent user navigation changes as a response to the quality of the content, the quality of its generation, or the character-

istics of the last visited pages.

Comparing workload generators is a laborious and difficult task since they offer a large amount and diversity of features. In this paper we compare generators according to a wide set of features and capabilities. Below, the twelve features and capabilities used are defined to ease the understanding of the comparison study.

- **Distributed Architecture.** It refers to the capability to distribute the processes of generation among the different nodes. The distribution of the workload generation significantly helps to improve the workload accuracy.
- **Mathematical-based Architecture.** This feature represents the capability to use mathematical models to define the workload. These models allow to improve the workload quality by using them as workload parameters (e.g., users' behavior models or simulation architectures.)
- **Business-based Architecture.** This refers to the capability to model web application business logic. When defining a testing environment, the architecture simulator should implement the same features as the real environment (e.g., e-commerce architectures typically include a catalog, a product searcher, or a payment gateway).
- **Client Parametrization.** This is the ability to parameterize generators nodes (e.g., number of users, allowed navigations set, or changes between navigations). In general, web dynamism highlights the need for a workload characterization based on parameters, and specially the related user behavior.
- **Workload Types.** Some generators organize the workload in categories or types, each one modeling a given user profile (e.g., searcher or buyer user profiles).
- **Modeling of the Web Application Business Logic (Business Test)**. This capability permits to define functional tests related to a real web application. These tests allow to guarantee the application correctness (i.e., the application provides the defined functionality, which fulfills the quality and assurance requirements).
- **Multiplatform.** Is referred to a computer software that are implemented and inter-operate on multiple computer platforms.
- **Differences between LAN and WAN.** Simulations usually run in LAN environments, and most of the current simulators cannot model differences between LAN and WAN, where applications are usually located.
- **Ease of Use.** The generator should be a friendly application carrying out usability guidelines, mainly in commercial products.
- **Performance Reports.** The results elaborated by the generation process are usually presented by using graphical plots.
- **Open Source.** This feature allows the developers' community to develop extensions or different generation alternatives over the generator architecture.
- **Dynamic Workload.** This is the main feature we are interested in, because the dynamism in contents and users is the most relevant characteristic in the current web.

Table 1 summarizes the studied software packages used to generate web workload as well as the grade (full or partial) in which they fulfill the features described above. By nature, this software can be classified in benchmarks and generators. In this paper, the latter group is broken down according to the generator offers capabilities to model the dynamic workload or not. In other words, as observed in Table 1, three main groups have been identified as discussed below.

- *Group I: Benchmarks that model the client and server paradigm in web context, but they do not consider dynamism in web workload generation.*

  Webstone was designed by Silicon Graphics in 1996 and it is currently commercialized as a Mindcraft product. It bases its operation on units of execution located on the client. These units simulate the clients' behavior when accessing web resources. These units can run on several machines in order to generate requests to the server resources, which can be classified in different categories according to their size.

  SPECweb99 was designed to measure the performance of systems offering services in the web. This generator, commercialized by SPEC, generates both static and dynamic requests. The workload is distributed among several clients, which are managed by a central client that collects data from the other ones. The clients generate requests according to certain categories that are defined after studying different web applications. In addition, clients verify the result of each request. This generator evaluates the architectural performance in order to provide a generic profile of web applications.

  SURGE was developed by Barford and Crovella (Barford and Crovella, 1997) with the goal of measuring the server behavior while varying the user load. The generator performs an analytical

Table 1: Workload generators and grade in which features are fulfilled.

Feat./Cap. \ Gen.	Webstone	SPECweb99	SURGE	Polygraph	S-clients	Webjamma	Deluge	Hammer	PTester	Siege	Httperf	Autobench	TPC-W	Webload	Mercury	JMeter	GUERNICA
Mathematical-based arch.	♦	♦	♦	♦													▲
Dsitributed architecture	♦	♦												♦	♦	♦	♦
Business-based arch.				▲	♦								♦	▲	▲	▲	▲
Client parametrization	♦	♦		▲		♦		▲					♦	♦	♦	♦	♦
Workload types		♦															♦
Business test													♦				▲
LAN and WAN					♦												▲
Multiplatform	♦	♦	♦	♦	♦	♦	♦	▲	♦	▲	♦	♦	♦	♦	♦	♦	♦
Ease of use														♦	♦	♦	♦
Performance reports		♦		♦			♦	♦	♦	♦	♦	♦		♦	♦	♦	♦
Open source	♦		♦	♦	♦		♦	♦	♦		♦	♦	♦			♦	▲
Dynamic workload													▲	▲	▲	▲	♦
	Group I				Group II								Group III				

♦ Full support ▲ Partial support

characterization of the user load and a set of mathematical models that generate the HTTP requests in the server.

Web Polygraph was developed at the California University (Rousskov et al., 1999) and it is a workload generator based on mathematical models which simulates both the client and the server. It also allows to introduce real components instead of mathematical models.

- *Group II: Workload generators that reproduce web clients behavior without considering dynamism.*

  S-Clients was developed by Banga and Druschel (Banga and Druschel, 1999). It generates load sporadic tips to improve the benefits provided by the server. S-Clients splits the process of generating traced HTTP requests in two subprocesses: one for obtaining the connection and the other for recovering the content, so enabling a relative parallelism.

  Webjamma was developed by Virginia Tech's Network Research Group. It is aimed at serving as a baseline for developing a future generator. This generator works in a simple way by taking a URL file that provides the source of the HTTP requests to be generated, so it cannot represent dynamic users' behavior. To this end, it uses a multiprocessing architecture based on generation nodes.

  Deluge was developed by Thrown Clear Productions. It is a basic and open source solution that includes three main components: i) `dlg_proxy` records HTTP requests, ii) `dlg_attack` generates workload by reproducing recorded users' requests, so it does not allow dynamic HTTP requests generation, and iii) `dlg_eval` elaborates statistics from the generated results.

  Scripts and tools, which are usually basic and open source approaches, only allow to capture HTTP requests and to reproduce them for stressing purposes; e.g., *HAMMERHEAD*, *PTESTER*, *SIEGE*, *HTTPERF* or *AUTOBENCH*.

- *Group III: Generators that model web workload considering the dynamism.*

  TPC-W (Smith, 2000) was defined by TPC. It is oriented to e-commerce web transactions, providing both models of business-client (B2C) and business-business (B2B). It is aimed at evaluating performance of architecture on a generic profile of web applications. It examines real features of e-commerce applications: security, catalog, etc. The application profile is configured by choosing the desired features, so allowing to represent different user behaviors. TPC-W partially models dynamic users' behavior by means of these users' profiles.

  Webload (RadView Software, 2003) is web workload generator commercialized by RadView. It is oriented to explore the performance of critical applications, quantifying resource usage in the server. Webload is aimed at evaluating the correctness of a given application (testing). This generator includes two main modules: the first module is on the client side and enables it to execute *request agendas* (navigation scripts) on the server side; the second module is on the server side and estimates performance gains. Webload presents

some capability to model users' behavior by using agendas.

MERCURY LoadRunner is a software, commercialized by Mercury, for functional and performance web testing. It generates the specific workload of a web application by simulating the concurrent navigations of a typical users set. The generator supports the recording of the navigation and the definition of users' families to generate web workload. The software also includes a versatile module that permits the graphic representation of the representation. LoadRunner allows to define users' navigations and partially dynamic users' behavior by using Javascript language to define these navigations.

JMeter is a solution for generating workload and evaluating web server performance presented by Apache Project. It is written entirely in Java and provides an easily configurable and visual API, evaluating the performance of client side web applications. It presents partial capability to model dynamic workload by defining a navigation test based on patterns (e.g., regular expressions).

GUERNICA is a web workload generator designed to generate dynamic workload by modeling dynamic users' behavior. In addition, its design and implementation is intended to achieve a solid baseline generator as a first step to support future improvements and extensions with the aim of becoming a commercial product.

Most of the presented generators do not model dynamic workload because: i) they are simulation approaches that do not reproduce real workload (e.g., Webstone or Web Polygraph), ii) they are based on HTTP traces (e.g.,Webjama, S-Clients or the scripts and tools solutions) or iii) they are based on workload models that do not consider dynamism as a parameter (e.g., SPECWeb99 or SURGE).

In summary, we can conclude that only five of the studied approaches (i.e., Webload, TPC-W, JMeter, Mercury and GUERNICA) present some capability to model users' dynamic behavior and to generate dynamic workload, but only GUERNICA presents fully capability. This generator totally supports seven of the twelve described features and partially supports the remaining ones.

3 GUERNICA

GUERNICA (Universal Generator of Dynamic Workload under Web Platforms) is a workload generator based on the Dweb model (Peña-Ortiz et al., 2009), a recent workload model able to characterize the web dynamism in an accurate way. Three main concepts allow Dweb to deal with dynamic workload: *navigation*, *workload test*, and *workload distribution*. The two former permit to characterize the workload dynamism by means of web users' behavior models. The latter concept allows to improve the workload accuracy by distributing its generation among different machines.

Its design is the result of the cooperation among the *Web Performance Research Group (Polytechnic University of Valencia)*, *Intelligent Software Components* and the *Instituto Tecnológico de Informática*; thereby, bridging the gap between academia and industry.

GUERNICA applications and Dweb model concepts permit to carry out the workload test process by considering different levels of dynamism in users' behavior when characterizing web workload. This process consists of four main phases (see Figure 1) detailed below.

1. **Navigation Definition.** This phase defines the first level of dynamism by modeling the users' dynamism when interacting with the dynamic web contents (typical of dynamic web). For instance, when a user runs a web searcher to make a query, he or she usually continues visiting one or more sites determined by the search outcome. If the query is successfully resolved (i.e., a list of results is obtained), it is likely that the user will either i) visit the first site of the list, or ii) refine the search. On the other hand, if the response time is too longer, the query might be canceled. In other words, each user request depends not only on the response itself but also on other issues related to the quality of service (e.g., response time length or content amount).

   Owing to dynamic web contents, the users' dynamism is modeled by using the *navigation concept*. Figure 2 shows the navigation tree corresponding to a *Google search navigation*.

   Two main parts can be distinguished in the navigation:

   (1) The upper part of the diagram (before reaching branch *b1*) shows the two ways in which the search can be initiated:
   - a. On the left side, the user makes use of a search toolbar of a web browser (e.g., Google toolbar for Mozilla Firefox) to make the query directly.
   - b. On the right side of the figure, the user reaches branch *b1* after the *Google.HOME* node, where the user requests the main page (http://www.google.com) to web searcher engine. After that, she or he waits for a while

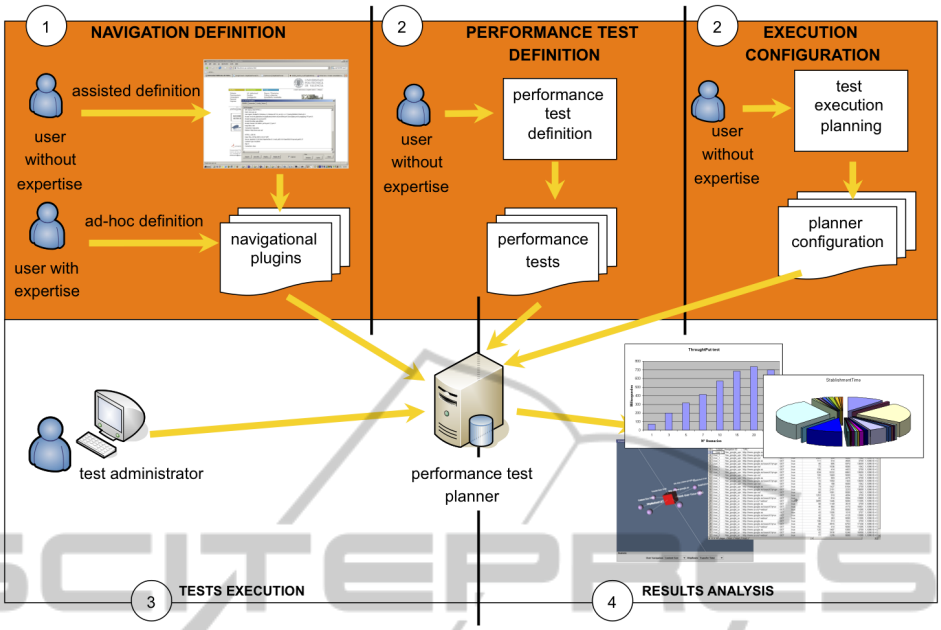

Figure 1: Testing phases in GUERNICA.

(time referred to as the user thinking time) and then the user makes the query.

(2) In the below side of the diagram, the user analyzes the results:

a. If the web search engine provides results (path from *b1* to *b2*) the user analyzes them. After that, she or he can refine the results by making a new query, *refined query* in the figure (path from *b2* to *b1*), or access the top n sites provided and finish the navigation (path from *b2* to black dot through *X_TH_RESULTS.HOME* node).

b. On the contrary, that is, if no results are provided (path from *b1* to *b3*), the user can make a new query, *other query* in the figure (path from *b3* to *b1*), or finish the process (path from *b3* to black dot).

CARENA (Niño et al., 2005) is a Mozilla plugin that helps GUERNICA to define users' navigations. It captures real users' navigations that GUERNICA converts to its internal representation.

2. **Performance Test Definition and Execution Configuration.** It defines the workload of the target site by using the *workload test* and *workload distribution* concepts.

Workload test concept allows to define the second level of dynamism. This level is related to the roles that users play in the web and the continuous changes of these roles defining users' behaviors.

For example, assume people at the Import and Export Department in a typical multinational company that use to access the web during their job. Assume also that the company has got a web ERP[1] (intranet) which allows web access. Most of the time, workers use Internet for professional purposes (e.g., intranet navigations, suppliers sites navigations, or professional web searches), but sometimes they use the web for leisure purposes (e.g., reading the news with Google Reader, performing personal searches, or checking the mail). So we can distinguish between two roles when a department member navigates the web: *working behavior* (professional navigations), and *leisure behavior* (personal navigations).

The Dweb model uses the *workload test* concept to model user behaviors and changes between behaviors. Figure 3 defines the working and leisure behaviors of the example, and the likelihood to change between behaviors by using balanced arcs (the arc weight is the probability to change from the source behavior to the destination behavior). These behaviors are defined as automatons, where their nodes represent navigations, and their balanced arcs indicate the transitions between navigations (the arc weight indicates the probability to take that arc).

Finally, *Workload distribution* concept allows to

[1]Enterprise Resource Planning (ERP) is an application used by large organizations to manage inventory, resources, and business processes across departments.

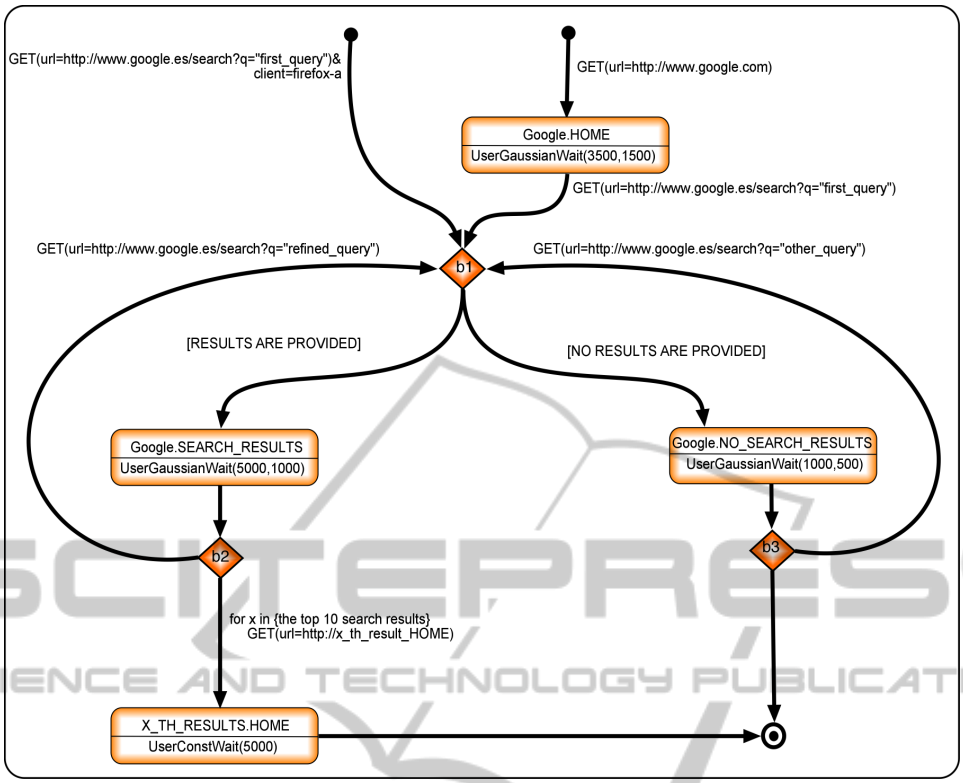

Figure 2: Google Search navigation pattern.

improve the workload accuracy by distributing its generation among different machines.

3. **Tests Execution.** This phase executes workload tests gathering performance statistics. The *planner application* controls and plans the workload test. The *probe client application* is aimed at evaluating from the point of view of a typical user (e.g., response time or application correctness) the major functionalities of a given web application, while it is stressed by the *workload generators*. Results collected by generators and probe clients are given to the planner, which groups, classifies and reaches a consensus among them in order to obtain an uniform set of results.
4. **Results Analysis.** Finally, GUERNICA analyzes the performance of the system under test and represents the performance indexes by using graphical plots. Regarding 3D representation, it represents the results in an external format based on an ontology that SemViz (Blázquez et al., 2008) (implemented as an applet) can read and display graphically. SemViz visualizes knowledge based on ontologies by using 3D technology.

4 CONCLUSIONS

Due to the increase in the number and popularity of web applications and distributed services such as e-commerce, social networks or Amazon Web Services, dynamic contents are becoming more and more frequent. For example, Web 2.0 applications mainly focus on users who are the potential clients by providing them an experience closer to desktop applications. One of the most important mechanisms to achieve this goal is the dynamism present in the web content (e.g., personalized contents or publicity), which induces the dynamism of users' behavior in the current web. This feature is the main shortcoming when modeling and generating real web workload.

In this paper we have analyzed the characteristics of a representative subset of the state of the art workload generators, focusing on the capability to represent user dynamism in the current and incoming web. We found five workload generators that present some capability to represent workload dynamism although only *GUERNICA* allows to fully represent this dynamism. With the aim of illustrating how this fact has been achieved, we described the workload test process used in the generator side to address both user

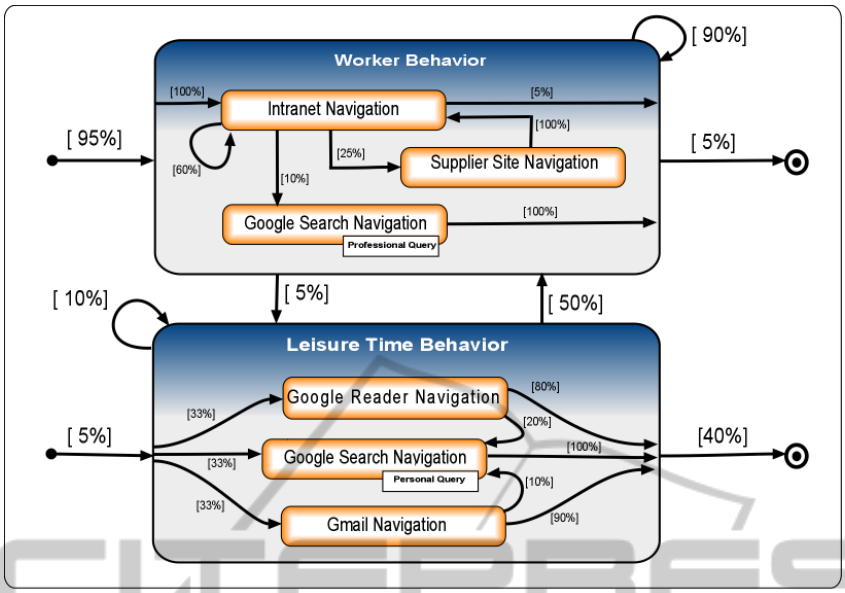

Figure 3: Working and leisure behaviors workload test.

dynamic behavior and workload distribution.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Government Grant (CICYT TIC2001-1374-C03-02 and FIT-340000-2004-236), Spanish Ministry of Education and Science and the European Investment Fund for Regional Development Grant (TSI 2005-07876-C03-01) and IMPIVA Grant (IMIDTD/2004/92, IMIDTD/2005/15).

REFERENCES

Amza et al. (2002). Specification and Implementation of Dynamic Web Site Benchmarks. In *5th Workshop on Workload Characterization*.

Banga, G. and Druschel, P. (1999) Measuring the capacity of a Web server under realistic loads. In *8th WWW*, 2(1-2):69–83.

Barford, P. and Crovella, M. (1997). An Architecture for a WWW Workload Generator. In *WWW Consortium Workshop on Workload Characterization*.

Blázquez et al. (2008). *Trends on Legal Knowledge, the Semantic Web and the Regulation of Electronic Social Systems*, Chapter Visualization of Semantic Content.

Cecchet et al. (2002). Performance and scalability of EJB applications. *SIGPLAN Not.*, 37(11):246–261.

Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR.

Houben et al. (2004). Modeling User Input and Hypermedia Dynamics in Hera. In *Web Engineering - 4th International Conference*, Vol. 3140, 60–73.

Menascé, D. A. and Almeida, V. A. F. (2000). *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*. Prentice Hall PTR.

Niño et al. (2005). Carena: A tool to capture and replay web navigation sessions. *Third IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services*.

O'Reilly, T. (2005). What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. *O'Reilly Online Publishing*.

Peña-Ortiz et al. (2005). Modeling continuos changes of the user's web dynamic behavior in the WWW. In *Fifth International Workshop on Software and Performance*, 175–180.

Peña-Ortiz et al. (2009). Dweb model: Representing Web 2.0 dynamism. *Computer Communications*, 32(6):1118–1128.

Rousskov et al. (1999). The first ircache web cache bake-off. Technical report, National Laboratory for Applied Network Research.

Smith, W. D. (2000). TPC-W: Benchmarking An Ecommerce Solution. Technical report, Intel Corporation.

Technical report, RadView Software. (2003). WebLOAD 6.0 The WebLOAD difference.

Treese, W. (2006). Web 2.0: Is It Really Different ? *netWorker*, 10(2):15–17.

Tselentis, G. (2009). *Towards the Future Internet: A European Research Perspective*. IOS Press Inc.

Weinreich et al. (2006). Off the beaten tracks: exploring three aspects of web navigation. In *15th WWW*, 133–142.