Remote web usability testing

Andres Baravalle, Vitaveska Lanfranchi
Department of Computer Science, University of Turin

Measuring behaviour 2002
Amsterdam, The Netherlands
27-30 August 2002

Outline

- Focus of the research
- Problems in usability testing
- OpenWebSurvey
- OpenWebSurvey architecture
- OpenWebSurvey testing model
- Case study

Focus of the research

- Record user navigation in the web
- Evaluate web sites usability.

Problems in usability testing

- It is common to run usability tests in research laboratories but:
  - There are situations in which it is not possible to run tests in a research lab
  - It can be difficult to find a sufficient and significant number of users willing (or able) to participate
- For example, it is very difficult for sick persons to come to a research laboratory.

Problems in usability testing (2)

- An alternative solution is to run research tests *in loco*:
  - It can take a lot of time
  - There are places where the researcher cannot access
- There are places where it is difficult for the interviewer to access, as some hospital units, or some (ri)educational centers.

Why remote usability tests

- **Remote usability testing** allows to:
  - **record** users behaviour remotely **evaluating** the usability of web sites gathering information from remote users.

OpenWebSurvey

- Web-based software (written in PHP) able to:
  - **supply a survey**
  - record **quantitative data** about the user behaviour while surfing a site
- Monitors web navigation storing data on:
  - **visited pages** (load time and some client side actions)
  - the **entire session** (total visit time, page visit time, general information about the user system)
  - **survey answers**
- Able to record, store, share and process data for web usability analysis.

OpenWebSurvey advantages

- How it works? It **rewrites the code** of the site under investigation!
- The user (usually) does not perceive **any difference** while surfing the site under investigation
- Easy support for multiple languages both in the interface and in the surveys.
- **No** need of **installing** any software or hardware components, either in the client computer or in the web site server.

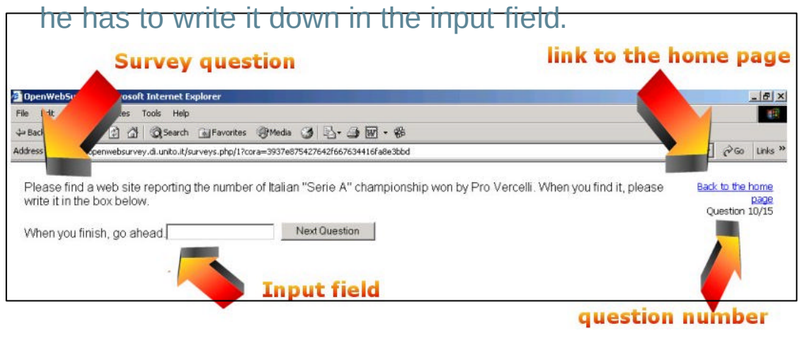
OpenWebSurvey testing model

- The user connects to the OpenWebSurvey server, surfs a site under investigation (and answers to some questions)
- The testing interface is a **double framed** browser window:
  - The **upper frame** contains the **survey questions** and (eventually) the input field for the answers
  - The **lower frame** contains the **site** under investigation, modified in order to record surfing data.

OpenWebSurvey interface

Interface – upper frame

- Example of a survey question requesting a complex task to be performed
- When the user has found the requested information, he has to write it down in the input field.



Interface – lower frame

- Frame containing the site under investigation
- The web site **looks the same**!

OpenWebSurvey data storage

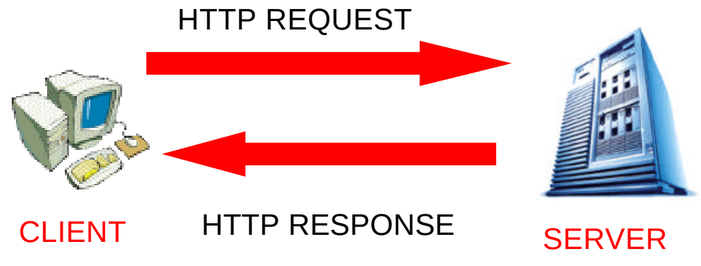
- Survey questions are stored in standard **triple-s** xml format (xml dialect for describing surveys)
- Triple-s has been extended (with **backward-compatibility**) in order to represent more complex information (e.g. to represent the name of the site relative to the question).

Analysis stage

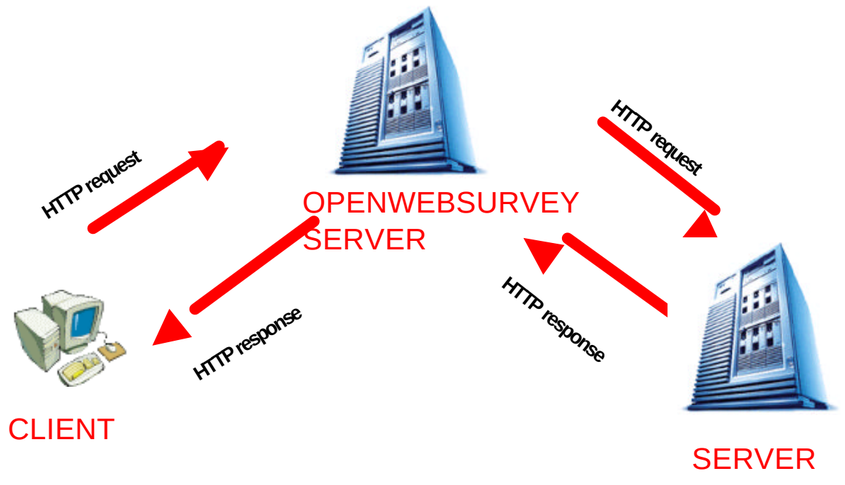
- In the analysis stage, researchers can use OpenWebSurvey data to:
  - Have **statistical reports**
  - Make **assumptions** about site usability
  - Infer **psychological motivations** that could have induced certain actions, using collected data about survey answers and users behaviour.

Internet architecture

- Usually a client sends a request to a server, that replies with the requested page.



OpenWebSurvey architecture

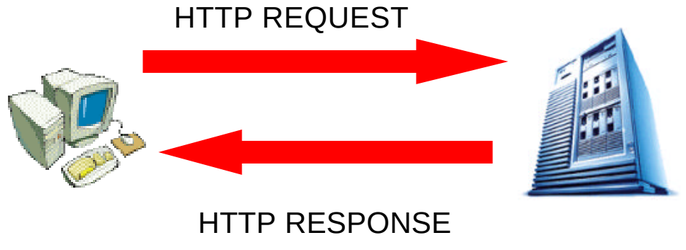


OpenWebSurvey architecture

- OpenWebSurvey is based on a server that:
  - **Intercepts** (most) URL requests
  - **Connects** to the requested URL
  - **Retrieves** the expected page
  - **Rewrites** it:
    - every URL is rewritten like:
      - http://openwebsurvey.di.unito.it/test.php/http://it.yahoo.com/nw/we at
    - OpenWebSurvey, as it is running in the server side, is able to handle redirects and any kind of HTTP headers
  - **Compresses** it: if the client supports compression, text data are compressed before delivery.
  - **Sends back** to the client the requested page, modified in order to record user behaviour
- Note that **every HTML link or HTTP redirect is rewritten** by OpenWebSurvey server.

OpenWebSurvey perceived architecture

- The user **does not perceive** the real architecture!



OpenWebSurvey performance

- OpenWebSurvey server is pretty fast!
- It performs some operations in order to **improve the performance**:
  - uses **compression** algorithms to send the page back to the client
  - uses **caching** for serving part of the contents.

OpenWebSurvey capabilities

- OpenWebSurvey is able to process:
  - **HTML pages** with any kind of standard object:
    - sounds
    - images
    - movies
  - **JavaScript code**, if it finds complete URLs (e.g.: http://www.mydomain.com/mypage.html)
- OpenWebSurvey can collect informations both in the client and in the server side. Cookies (or querystring) are used to identify sessions.

Open problems

- OpenWebSurvey is still not able to process:
  - binary data:
    - **Flash** or **director** movies
    - **Java** applets
  - **JavaScript** code with dynamically generated URLs like:
    - Document.Write (“http://www.mysite.com/” + “about.html”);
- In most of these cases OpenWebSurvey fails gently:
  - The user does not perceive the failure
  - The software does not record the user behaviour on that page

Case study

- An alpha release of the software is being tested at Turin University, Department of Computer Science, to investigate **web search strategies**
  - http://openwebsurvey.di.unito.it

Case study (2)

- The search engine analysed is Virgilio Junior and the target are children from 5 to 12 years old.
- The domain examined is the Italian one
- The goal of this research is to understand if there is a mutual relation between:
  - The knowledge about the domain and the success in finding information
  - The type of task and the strategy of research used (keywords or categories)
  - The layout of the page and the strategy of research.

THE END