Information Visualization Attributes and Operators for a Web Log Based Management System

Florin Zamfir
Univ. de Aveiro
Campus de Santiago, Aveiro
florin@ca.ua.pt

Óscar Mealha
Univ. de Aveiro
Campus de Santiago, Aveiro
oem@ca.ua.pt

Beatriz Sousa Santos
IEETA / Univ. de Aveiro
Campus de Santiago, Aveiro
bss@ieeta.pt

José Nunes
Univ. de Aveiro
Campus de Santiago, Aveiro
jnunes@ca.ua.pt

abstract

This paper presents the theoretical basis of a web log based management system used for research purposes. The main focus is related with information attributes and operator definitions used for information visualization analysis and manipulation. Web sites and portals are becoming complex organic systems difficult to monitor usage patterns and control navigational and structural problems. In many day to day routines, severe inefficient and ineffective situations are ignored or understood a little too late. The system mentioned in this paper represents a research framework with the goal of solving information and communication management problems in a web site/portal, using an information visualization approach, with direct manipulation adjustment and problem identification mechanisms.

Although the specific contribution of this paper is theoretical, some system usage screenshots were added to clarify the purpose of some attributes, definitions and operators.

keywords

information visualization, information and communication management, web logs.

1. INTRODUCTION

Starting from the following premise presented by [Ivory2002] "*Using quantitative measures of the informational, navigational, and graphical aspects of a Web site, a quality checker aims to help nonprofessional designers improve their sites.*" and using a complementary approach, our goal is to analyze existing hypermedia structures and provide legible visual feedback for their usage patterns. Usability issues / clues are discovered using quantitative measures of informational, navigational and graphical aspects of a website, enriched with a subset of qualitative based measures. Qualitative measures are derived from usability guidelines, combined with metrics derived from empirical analysis of web sites involving web site dynamics, navigational paradigms, usage scenarios, contents classification, etc. In order to achieve this, information attributes must be defined, information visualization operators conceptualized and system components developed into a unified framework with tightly coupled inspection methods.

For the purpose of this paper, that presents the theoretical basis of a web log based management system used for research purposes, it is important to mention that information visualization is a complex process that transforms raw data into visual features. It combines several techniques to collect, filter and represent the information. According to [Healey2001], scientific visualization is the conversion of collections of strings and numbers into images that allow viewers to perform visual exploration and analysis. The information has to be collected and stored in accessible formats for interrogations. During this process, filters are applied to validate the contents. Afterwards, the information is available for exploration and inspection in a natural form (raw data) or organized in logical relational structures. Depending of the amount of information to be represented, several techniques are available to transform raw data into visual features.

One important contribution of this work is related with the approach to combine quantitative and qualitative measures for website structure, contents and usage into a integrated automated analysis tool, meant to help decision makers inside institutions reorganize their institutional websites according to the feedback provided by such a tool. The identification and association of several

attribute sets to page, text, link and graphical elements, as detailed in section 3 – Theoretical Information Framework, and the association of a visual transformation function to a unique structural attribute is somehow innovative and simplifies the representation of information.

We use visual features to provide help in answering the following general question: How is the site used? Starting from this general question, we can detail a specific sub-set of questions, whose answers might provide useful feedback: Who is using the site? What are the site areas / sectors / pages of interest? What statistical information can be obtained from the log files? What navigational behaviors can be detected by analyzing the associated usage statistics?, etc.

Exploring the enormous capacities of the human visual system, we present the information through different visualization methods, to provide help for answering the following concrete questions: Which are the areas with problems? What usability problems can be identified and at what level (content relations, semantic, navigational, design layouts, etc.)? How does the visual organization of the site influence user's navigational decisions?

The following sections describe the conceptual model of a proposed web log based data collection and visualization system and its theoretical information framework that includes; visualization metrics and taxonomy, data representation and raw mapping functions to visual features. System usage screenshots are used as examples to complement theoretical definitions and representations.

2. CONCEPTUAL SYSTEM MODEL

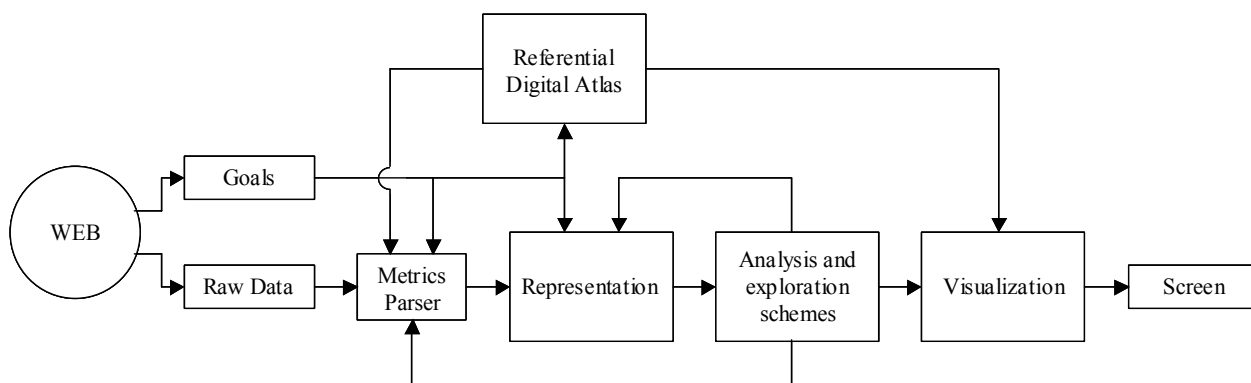A conceptual model for the visualization subsystem, as introduced by [Mealha2004], is represented in Figure 1.

Figure 1 Conceptual System Model

2.1 Description of modules

1. Web: represents the website with all its available services, pages and organization;

2. *Goals*: represents the set of specific results to be obtained and all the intermediary steps required for each of the goals to be attained;

3. *Raw Data*: represents the basic information available on its primary state. The information is available from several sources as: website contents, usage log files, eye tracking, motion tracking or interception systems, etc. This involves all the information as stored on its primary source, without any preliminary processing or filtering applied;

4. *Metrics Parser*: this module is designed to filter and extract valid information from the entire store of raw data. One example of information filter and extraction is the detection of usage session based on a specific set of attributes that uniquely identify a user session as a time window within the active time of a specific client, or "A delimited set of user clicks across one or more Web servers." [WCTD2006].

   Considering: *(1)* $\Delta T_k = \sum_{i=1}^{n} \Delta T_{j,i}$, the sum of all *n*

time windows of a specific client *j* that accessed the web server, $\Delta t_k \in \Delta T_{j,i}$ = the time periods between subsequent requests $t_{k,i}$ of a specific client's time window *i*; a request $t_{k,i}$ is considered as part of the same session *i* if *(2)* $\Delta t_{k,i+1} - \Delta t_{k,i} \leq \delta$, where δ = the maximum accepted time window for two subsequent requests to be considered in the same session. In this case, $\Delta T_{j,i}$ is considered a user session.

A 30 minutes δ represents the most common value for the definition of a time window (session) for most of the implementations: according to IFABC Global Web Standards [IFABC2006] a *Session* (or *a Visit*) is a "series of one or more Page Impressions, served to one User, which ends when there is a gap of 30 minutes or more between successive Page Impressions for that User."

5. *Referential Digital Atlas*: represents the information library that contains all relevant information for the system. It is represented using a specific format that supports the identification of all A_i, $A_{T,j}$, $A_{L,k}$, $A_{G,m}$ elements presented in the next section – Theoretical Information Visualization framework;

6. *Representation*: this module includes all available methods for representation of information used to produce visualizations. Several visualization methods are available (2D, 3D, etc.), each design within a specific set of goals;

7. *Analysis and exploration schemes*: the representations obtained from the previous module are combined with additional attributes to support direct manipulation of specific parameters that have a direct impact on the results produced by the visualization. This module provides interaction means with the visualization methods, allowing customizable representation to be produced;

8. *Visualization*: this module combines the results of all previous modules with the aim to display a visual representation of the processed information, which provides adequate interaction and inspection mechanisms;

9. *Screen*: represents the media used to display the information and can be implemented using traditional screens or sophisticated representation technique as virtual 2D/3D glasses, etc.

A technical transpose of the conceptual model is presented in Figure 2. As presented in [Zamfir2004] and [Mealha2004], the system model is composed of five active components, one database and a parameterization module used to filter the information. Three of the active modules (analyzer, compiler and interceptor) are used to gather/intercept the information stored / vehiculated on / to the web server, while the fifth component, the visualizer, maps the functionality of the visualization component of the conceptual model.

Modules *Web, Goals, Raw Data* and *Metrics Parser* have direct mapping to *web server, analyzer, interceptor* and *compiler* components, module *Referential Digital Atlas* is represented by the *database*, while modules *Representation, Analysis and exploration schemes, Visualization* and *Screen* are represented by the *visualizer* component.
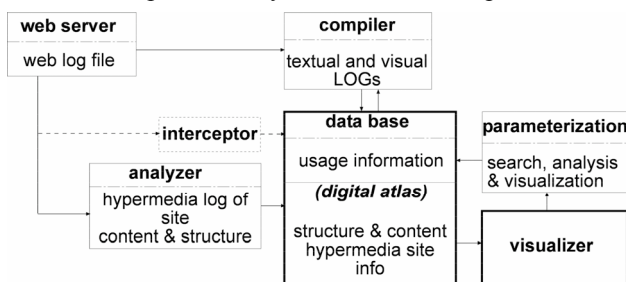

Figure 2 System Model – Conceptual Components

3. THEORETICAL INFORMATION FRAMEWORK

Our goal here is to represent the website contents as it is provided to its final users, in a web browser. For this purpose, we considered the structure of a website as a union of two sets: pages and interconnections, usually called hyperlinks: *(3) W = (P, H)*.

P is the set of pages: *(4) $P = \{P_1, ..., P_n\}$* where $n =$ number of site pages; H is the set of all possible interconnections between all website pages.

H is the set of hyperlinks: *(5) $H = \{H_1, ..., H_m\}$*, where $m =$ number of website interconnections. A hyperlink H_i is defined as an edge in the graph W that represents the website: $H_i = (p, q) \in P$, where p represents the origin page (also called *referrer* page) and q represents the child page (also called *referenced* page).

3.1 INFORMATION ATTRIBUTES

One potential approach inspired by Healey's [Healey2001] was introduced in [Mealha2004] to associate attributes to each hyperlink element present on a webpage: each page $P_t = \{h_{t,1}, ..., h_{t,n}\}$, contains n hotspots; each hot-spot, $h_{t,n}$, of any specific page has at least $m > 1$ attributes, $A = \{A_1, ..., A_m\}$. Using Ivory's [Ivory2002] approach, a page is composed of *Text elements, Link elements, Graphic elements* and a page P_i can be specified as the union of the three sets:

(6) $P_i = T_{i,p} \cup L_{i,q} \cup G_{i,r}$, where $L_{i,q}$ represents the set of hotspots. According to Faraday's [Faraday2000] classification, several page elements capture user's attention during the first visual scan of a page, some attributes like color, size, position, text style, etc., being able to directly influence user's navigational decision.

For the purpose of this works, an approach similar to *(6)* was adopted, even if all three presented solutions are conceptually similar. In conformance with previous definitions, the attributes definition can be extended to all elements that correspond to a web page: considering the page P_i with content elements $T_{i,p} \cup L_{i,q} \cup G_{i,r}$, a set of attributes can be associated to each of these

(7) $A_T = \{A_{T,1}, ..., A_{T,u}\}$,

(8) $A_L = \{A_{L,1}, ..., A_{L,v}\}$,

(9) $A_G = \{A_{G,1}, ..., A_{G,w}\}$.

$A_T =$ the set of attributes associated to Text elements, $A_L =$ the set of attributes associated to Link elements and $A_G =$ the set of attributes associated to Graphical elements; $u, v, w =$ number of elements associated to Text, Link or Graphical elements.

Seven structural attributes were identified for the web page itself and another three sets, each of five structural attributes, for each of text, link and graphical elements, as follows:

• $A_1 =$ URI (*Unique Resource Identifier*) of the page. For dynamic or personalized pages, a unique identifier can be generated to cope with the uniqueness criteria;

- A_2 = dimensions of the page in a fixed size browser window (e.g. a possible reference resolution can be considered 1024x768 browser window sizes);
- A_3 = minimum *site level* the page belongs to;
- A_4 = page raw content size in bytes;
- A_5 = page raw stored copy;
- A_6 = snapshot image of the page that is referenced as it is presented to the user;
- A_7 = dynamism type (static, dynamic, personalized, etc.).

For Text elements, a set of five structural attributes for the A_T set were identified:

- $A_{T,1}$ = URI (*Unique Resource Identifier*) of the page it belongs to;
- $A_{T,2}$ = font characteristics (size, style, spacing, color, paragraph, reading order);
- $A_{T,3}$ = position, type and size of the area occupied by the text in the page;
- $A_{T,4}$ = culture (language);
- $A_{T,5}$ = dynamism type (static, dynamic, personalized, etc.).

For Link elements, a set of five structural attributes for the A_L set were identified:

- $A_{L,1}$ = URI (*Unique Resource Identifier*) of the page the hotspot (link) belongs to;
- $A_{L,2}$ = position and size of the area occupied by the hotspot in the page;
- $A_{L,3}$ = type of hotspot (textual link, image link, composed graphical link, etc.);
- $A_{L,4}$ = URI of the page that is referenced;
- $A_{L,5}$ = dynamism type (static, dynamic, personalized, etc.).

For Graphical elements, a set of five structural attributes for the A_G set were identified:

- $A_{G,1}$ = URI (*Unique Resource Identifier*) of the page it belongs to;
- $A_{G,2}$ = occupied area type, position, size, colors;
- $A_{G,3}$ = motion / movie presence;
- $A_{G,4}$ = interactive or not;
- $A_{G,5}$ = dynamism type (static, dynamic, personalized, etc.);

Basically, we used Healey's [Healey2001] definitions to achieve visualization using a function $M(V, \phi)$ that transforms raw-information into a visualization feature V_j on the visual workspace: $V = \{V_1,...,V_r\}$ represents the visual features that are used to display the attribute(s)

(10) $A_j.\phi_j : A_j \rightarrow V_j$, in other words, maps the A_j domain into the viewable feature domain V_j; A_i domain is the domain of attributes specific for a web page, $A_{T,j}$, $A_{L,j}$, $A_{L,j}$ are the domains of attributes for text, link and graphical elements of a page j.

(11) $\begin{bmatrix} M_1(V_1,\phi_1) & M_2(V_2,\phi_2) & ... & M_f(V_f,\phi_f) \end{bmatrix}$ represents a vector of functions that transform raw information into visual features and in our particular case f depends of the number of visualization methods that can be achieved based on each of the attribute sets to be visually represented.

According to Farraday [Faraday2000], the visual analysis of a webpage is achieved in two phases: search and scanning. These phases can be analyzed based on the visual attributes that uniquely identify each page element. Subsets of the translation functions domains *(10)* and *(11)* can be used to code these visual attributes and visually highlight content classification based on focusing and attention capture information.

Unlike Farraday's approach that is based on guidelines derived from the use of eye-tracking techniques that identify which interface elements attract user attention, for our case of study, the information can be achieved by analyzing live experiments data produced by an eye / motion tracking or interception system, as exemplified by Card [Card2001].

To code the focusing and attention levels of page elements during search and scanning phases, a subset of attributes A_F can be defined as a subset of:

(12) $A_F \in A_T \cup A_L \cup A_G$.

The structural attributes sets can be extended accordingly to usage information that can be extracted from the log files and / or controlled experiments, [Drott1998], [Zamfir2004], [Fraternali2003], or usage information collected using visual / motion tracking or interception systems. The aim here is to be able to visually represent website usage information, navigational patterns, interface design layouts or other types of raw information.

For the webpage and its contents, structural attributes are extended with statistical usage information to support statistically rich information transformations of statistical raw data into visual features:

For the webpage, the following extended attributes were identified:

- A_8 = number of times this webpage was visited during a specific time period;
- A_9 = frequency of requests for the webpage in the given time interval;
- A_{10} = minimum amount of time the page was analyzed by the clients, using the $\Delta t_{k\min} \in \Delta T_{j,i}$ of two subsequent request;
- A_{11} = maximum amount of time the page was analyzed by the clients, using the $\Delta t_{k\max} \in \Delta T_{j,i}$ of two subsequent request;
- A_{12} = most analyzed areas of the page (potential information generated using an eye / motion tracking or interception system);

For Text elements, a sixth and seventh attributes were identified for the A_T set:

- $A_{T,6}$ = the average time users spent to analyze the text element: $\Delta T_{kta} \in \Delta T_{j,i}$;

- $A_{T,7} \in A_F$ = classification of the text element on a scale, regarding focusing and attention capture.

For Link elements, four other attributes were identified for the A_L set:

- $A_{L,6}$ = the average time users spent to analyze the link element: $\Delta T_{kla} \in \Delta T_{j,i}$;

- $A_{L,7}$ = average time to go from one page to another using this hotspot: $\Delta T_{klj} \in \Delta T_{j,i}$;

- $A_{L,8}$ = number of times this link element was clicked (visited) during a specific time period;

- $A_{L,9} \in A_F$ = classification of the link element on a scale, regarding focusing and attention capture.

For Graphical elements, two other attributes were identified for the A_G set:

- $A_{G,6}$ = the average time users spent to analyze the link element: $\Delta T_{kga} \in \Delta T_{j,i}$;

- $A_{G,7} \in A_F$ = classification of the link element on a scale, regarding focusing and attention capture.

Note that some of these extended attributes are complex entities that are represented using subsets of elementary attributes: *(13)*. Elementary attributes have direct mapping to relational structures as relational databases or similar entities. Relational or Object Oriented Databases (RDB or OODB) are ones of the most adequate systems for this purpose.

$$(13) \quad A_Z = \sum_{i=1}^{n} a_{w,i} \mid a_{w,i} \rightarrow \Delta_{x,i}, \text{ where } A_z \text{ is the com-}$$

plex attribute z, $a_{w,i}$ are elementary sub-attributes, and $\Delta_{x,i}$ represents the domain of atomic data types that can be directly represented in a relational data system.

Each visualization method makes use of a limited subset of structural and extended attributes to represent its information. Visual inspection methods may use a subset or complete set of visual elements defined by $M_f(V_f, \phi_f)$ to map raw information into visual features at a time. Complementary information can be represented at the same time, to highlight specific goals.

Based on 0 and 0, a potential representation for the web structure might consider some of the attributes sets previously described. Four visual elements $V_1 = \{V_{1,1}, V_{1,2}, V_{1,3}, V_{1,4}, V_{1,5}\}$ can be defined to represent the website structure and usage information as follows:

- $V_{1,1}$ = representation of a page as a tree node with image thumbnail or symbol;

- $\phi_{1,1}$: represents the page as a tree node with optional page image thumbnail obtained from A_6 attribute information or as / (or on) a basic 2D/3D geometric symbol (color circle or square);

- $V_{1,2}$ = connecting line between pages or nodes;

- $\phi_{1,2}$: uses $A_{L,1}$ and $A_{L,4}$ attribute information to represent and connect nodes as directional lines, uses $A_{L,8}$ to calculate line thickness and statistical connectivity;

- $V_{1,3}$ = visits info (statistical page visits, analysis times, etc.);

- $\phi_{1,3}$: visits information is visually embedded into the page representation using color, thickness or shape coding techniques, or it can be driven by a roll-over event over the page object, visual feature that pops up a window with statistical information. Several extended attributes can be used to highlight visits information: A_8, A_9, A_{10}, A_{11};

- $V_{1,4}$ = connectivity info (statistical, URLs, etc);

- $\phi_{1,4}$: driven by a roll-over event on the connecting line, this visual feature feedbacks a pop up window with statistical information, URLs, etc. related to the nodes or to their inter-connectivity. Different graphical representations are user: textual, percentage, progress bars, etc. Several attributes can be used to highlight usage information: $A_8, A_9, A_{10}, A_{11}, A_{L,6}$, etc.;

- $V_{1,5}$ = an image thumbnail;

- $\phi_{1,5}$: driven by a roll-over event, it draws a thumbnail with the contents of the page represented by the network or tree node with a semi-transparent factor. It is always visually linked or related to the node. Several contents attributes can be used to overlay statistical visual feedback on the page thumbnail: $A_{12}, A_{T,6}, A_{T,7}, A_{G,6}, A_{G,7}, A_{L,6}, A_{L,7}, A_{L,8}, A_{L,9}$.

An example of usage might consider the average of time users analyzed a webpage: if the value tends to on $\Delta t_{k\min}$, the page might have been used only for navigation purposes; if the value tends to $\Delta t_{k\max}$ the page might have been a contents store of interests for the users). This information can be easily represented using $V_{1,3}$ and $V_{1,4}$ visual element. Shape thickness or color coding can be applied to produce valuable insights.

3.2 Information Layers

The attributes sets are visually represented as needed. They are not necessarily used simultaneously on one visualization scheme but are useful for a tightly coupled

representation and interpretation between different visualization scenarios. For this purpose, visual features are classified in several classes of representation, called *Information Layers (IL)*. An Information Layer is a logical collection of web data abstractions that can be laid over a visual representation [Chen2004]. This approach was inspired from GIS (Geographical Information Systems) [GIS2005] where several map layers are manipulated to attain visual representations of the information.

To visually represent the $A \cup A_T \cup A_L \cup A_G$ attributes sets we divided the visual workspace in several information layers, using a similar approach as introduced by Chen [Chen2004]. However, some of our information layers consist of distinct visual features represented using transparency. These information layers are achieved using visual cues such as color, size, shape and thickness to code information from specific attributes sets. We can exemplify with the usage of color, transparency and thickness of link elements to represent connectivity information $V_{1,4}$; in addition, the same cues are suitable to represent visual paths followed by users overlaid on the *visual workspace* of the website (visual workspace = the information layer that combines all visual scenarios of the website's UI, mapped on the same viewport as used for analysis purposes).

Our work aims not only to make a representation of the website structure (or portions of it), as introduced by Chen [Chen2004] with the *Web Image*, but also for several representations of website's interface design layouts as visual workspace, visual representation of concurrent website usage sessions virtually mapped over the website structure, and dynamic goal-oriented interconnections of website pages. Multi-tier representations can be achieved by combining statistical information, extracted from the extended attributes sets, with structural representations that make use of structural attributes.

Each IL user a subset of the structural and extended attributes sets: $A_{ILj} \in A \cup A_T \cup A_L \cup A_G$ and depends on a subset of the complete set of visual translation functions $M_f(V_f, \phi_f)$.

By using the information that can be extracted from the website structure, its classified contents and usage information and combining most of the structural and extended attributes sets, several information layers can be considered for our case:

1. *Reference Map (RM)* or *Base Image (BI)* – is the layer that uses structural attributes to visually map a possible representation of the Website Structure (WS), page design layouts in a Visual Workspace (VW), etc. It makes use of shape, color, thickness and transparency to code A_1, A_2, A_3, A_6, A_7, $A_{T,1}$, $A_{T,2}$, $A_{T,3}$, $A_{T,5}$, $A_{L,1}$, $A_{L,2}$, $A_{L,3}$, $A_{L,4}$, $A_{L,5}$, $A_{G,1}$, $A_{G,2}$, $A_{G,3}$, $A_{G,4}$, $A_{G,5}$ attributes sets. BI is usually the background layer for most of our visual representations: Figure 3;
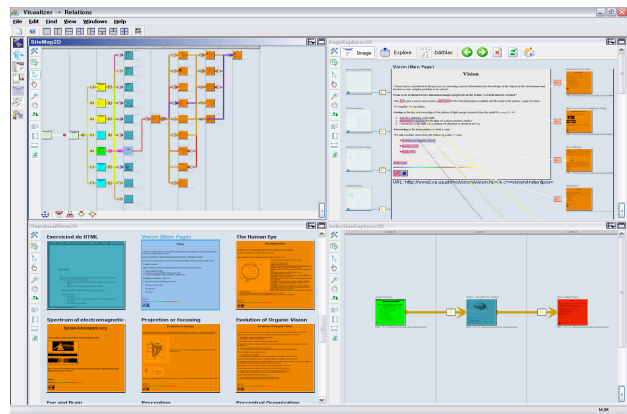

Figure 3 Reference Maps with additional attributes

2. *Number of Visits (NoV)* – represents the layers class that displays statistical information for page visits and link visits Figure 4. It uses the following extended attributes: A_8, A_9, $A_{L,8}$. Object shape, thickness, color and transparency visual clues are used to highlight page visits and link visits information. These visual clues are usually mapped on top of the *Base Image* layer – BI, using A_2, A_3 and A_6 attributes to represent it;
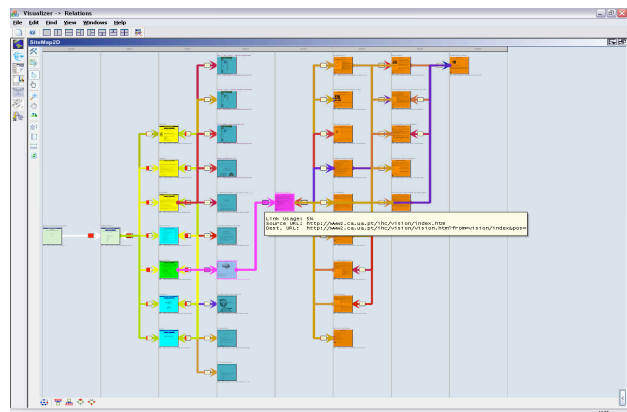

Figure 4 Visits information

3. *Links Usage (LU)* – represents the layers class that codes usage information for inter-page connection elements Figure 5. This layer visually represents the $A_{L,8}$ attribute and uses object shape, thickness, color and transparency as visual clues. It can be overlaid on top of a BI composed of the website structure as links elements between website pages, a BI that represents the visual workspace of the website as positional coded visual clues, or a BI that represents webpage contents as statistically coded hotspots information;
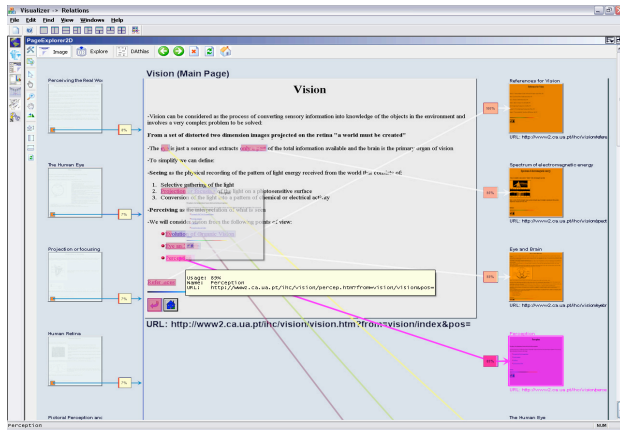
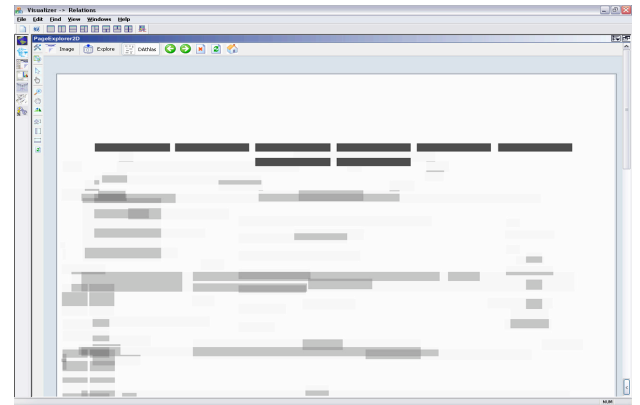**Figure 5 Hovered additional link information**

4. *Viewing Time (VT)* – is one of the most complex layers class and uses the information collected from eye / motion tracking or interception systems to represent A_{10}, A_{11}, A_{12}, $A_{T,6}$, $A_{T,7}$, $A_{L,6}$, $A_{L,7}$, $A_{L,9}$, $A_{G,6}$, $A_{G,6}$ attributes sets. The usage of visual clues to code these attributes depends of the type of information to be represented, e.g. thickness, direction, transparency and color can be used to represent $A_{L,8}$ mapped on top of the BI described by $A_{L,2}$. Several attributes as $\Delta t_{k\min}$, $\Delta t_{k\max}$, ΔT_{kta}, ΔT_{kla}, ΔT_{klj}, ΔT_{kga} can be used for the VT IL, and then represented on top of an adequate BI;

5. *Inspection Density Map (IDM)* – represents the layer that uses shape and color to represent statistically derived information based on $A_{T,3}$, $A_{G,2}$, $A_{L,2}$ structural attributes – inspection and interaction areas. This IL is usually represented on top of the visual workspace BI and codes statistical usage information identified by A_{12}, $A_{T,6}$, $A_{T,7}$, $A_{G,6}$, $A_{G,7}$, $A_{L,9}$, $A_{L,6}$, $A_{L,7}$ and $A_{L,8}$ extended attributes.

These five classes of information layers might depend on the actual system implementation. Some might support dynamic filtering and interaction mechanisms while others are only supposed to represent a static view of the information. However, several derivations of these might consider additional information and / or inspection mechanisms.

The combination of a Reference Map – RM (BI) with one or several Information layers – IL produces a *Visual Representation (VR)* Figure 6 and Figure 7, as a final target for an analysis and visualization system:

$$(14)\ VR_f = RM_m \bigcup \sum_{i=1}^{n} IL_i,\ \text{where}\ VR_f\ \text{is a specific}$$

VR to be achieved, RM_m is the reference map (base image) used as background for the representation, and each IL_i is a specific information layer obtained from the combination of on or several extended attributes and visual translation functions $M_f(V_f, \phi_f)$.

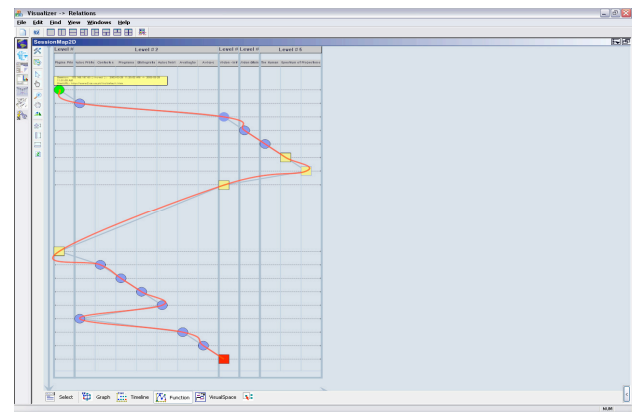**Figure 6 Visual Representations (I)**

Figure 7 Visual Representations (II)

3.3 IL operators and operations

A possible definition of elementary sub-attributes, applied to the set of extended attributes, with each elementary sub-attribute being used to represent a set of atomic values in a specific active domain, can be:

$$(15)\quad A_w = \sum_{i=1}^{n} a_{w,i}\ |\ a_{w,i} \to \lambda_{w,k} \in \nabla_{w,i}\quad \text{where each}$$

elementary sub-attribute $a_{w,i}$ can have a $\lambda_{w,k}$ value, defined on a specific active domain of atomic values $\nabla_{w,i}$, within the range of possible values $[\delta_{\min}, \delta_{\max}] : \delta_{\min} \leq \lambda_{w,k} \leq \delta_{\max}$.

In following paragraphs we make use of definition *(15)* to explain how each information layer depends on the domain of values $\nabla_{w,i}$ for each elementary attribute mapped into a visual feature.

Note that the range of possible values $[\delta_{\min}, \delta_{\max}]$ for each $a_{w,i}$ can be used as parameters on visual mapping functions ϕ_f to control the results for the visual representation, e.g. remapping a visual feature to a distinct domain of value might produce different results that show

or hide subsets of values from the represented domain. For this purpose, we can define ϕ_f as follows:

(16) $\phi_{f,w} \rightarrow \phi_{f,w}(\lambda_{w,k}, \delta_{min}, \delta_{max})$, where each translation function $\phi_{f,w}$ depends of at least three parameters:

$(\lambda_{w,k}, \delta_{min}, \delta_{max})$.

A complex set of parameters can be used to filter information, usually called *threshold* Figure 8. The specific set of parameters depends on the representation of each elementary attribute.

Figure 8 Example of Threshold Limits

Similar to the *Web Graph Algebra* introduced by Chen [Chen2004] and inspired by *Map Algebra* [Tomlin1994], we introduced a set of operators to manipulate information layers. These operators allow web usage mining and data operations on information layers. Unary operators are used to filter information while binary operators are used to combine the information on two distinct layers and obtain a new layer. Applying successive operators on information layers might produce distinct results, depending on what operators are used and on the intended results.
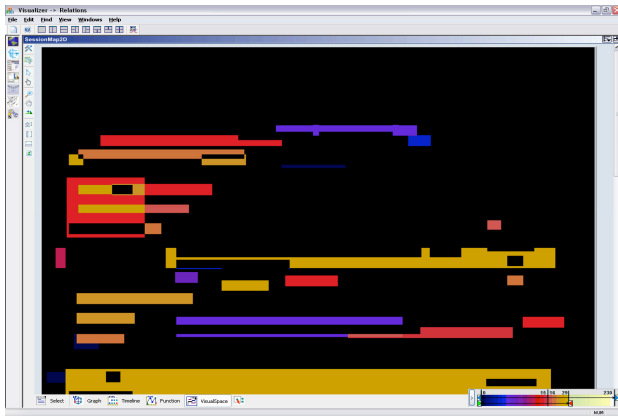

Figure 9 Modified Threshold Results

Operator FILTER is a unary operator that manipulates one information layer according to a specific threshold and filters information not in the specific domain of values specified by threshold parameters:

(17) $IL_{\theta} = FLT_{threshold}(IL_{\alpha})$, where IL_{α} is filtered using the threshold parameters and the resulted content is placed into a new IL_{θ};

Operator ADD is a binary operator that adds two distinct layers of information:

(18) $IL_{\theta} = IL_{\alpha} + IL_{\beta}$ selects the objects that exist in IL_{α} and IL_{β}, and transposes them into IL_{θ}. If the objects exist in both layers, the resulted value is the sum of their respective content values from both layers, otherwise is the value associated to the object; e.g. considering the object O_x present in two distinct layers, with two distinct numeric values associated to each layer, *x, y*, the result of the ADD operator represents the object O_x with a numeric value associated, calculated as the sum of $x + y$;

Operator MINUS is a binary operator that makes the difference of two distinct layers:

(19) $IL_{\theta} = IL_{\alpha} - IL_{\beta}$ selects the objects that exist in IL_{α} and IL_{β}, and transposes their difference into IL_{θ}. If the objects exist in both layers, the resulted value is the difference between their respective content values from both layers, otherwise the object in the second layer is not considered. Negative values can be obtained with MINUS operator, but these values can be either considered zero or differently coded in the representation;

Operator COMMON is a binary operator that makes the intersection of two distinct layers:

(20) $IL_{\theta} = IL_{\alpha} :: IL_{\beta}$ selects the objects that exist in both IL_{α} and IL_{β}, and transposes their intersection into IL_{θ} with the respective content values available in both layers only. The resulted values can be considered: the minimum, the average or the maximum values of the intersected content in two layers;

Operator EXCEPT is a binary operator that selects the objects in only one of two distinct layers:

(21) $IL_{\theta} = IL_{\alpha} \ni IL_{\beta}$ selects the objects that exist only in IL_{α} and do not exist IL_{β}, and transposes their content values into IL_{θ};

The feature that allows dynamically applying of filters and interaction with the representation domains of atomic values is designated as *Dynamic Visual Inspection and Manipulation* (*DVIM*). Several parameters can be manipulated to control the behavior of visual mapping functions $M_f(V_f, \phi_f)$.

Example: To exemplify a possible usage of operators, we can consider a simple visualization method used to make a visual inspection of interface design (visual workspace) coherence [Mealha2004], presented in Figure 10.

This visualization methods uses attribute mapping on the original 2D visual workspace, where pages were viewed. It needs to map $A_{L,2}$ attribute of link elements as visual elements transposed as their real geometrical active area on the 2D visual workspace. In this case, the information

is presented to the viewer after being transformed with "raw information → visual feature" function $M_i(V_i, \phi_i)$. This visualization scheme can be decomposed in 2 basic visual elements $V_i = \{V_{i,1}, V_{i,2}\}$ with:

- $V_{i,1}$ = geometrical area of attribute $A_{L,2}$ that varies in color according to "*hot-spot usage*" calculated from attribute $A_{L,8}$, in the range that goes from $\delta_{\min}$ to $\delta_{\max}$ according to the selected filtering threshold, with $\phi_{i,1}$: that transforms the usage statistics with a linear or logarithmic function into a color for the geometrical area. For this particular example, one considered all areas as pixels, the usage associated to each pixel is calculated as sum of the usage values for all hotspots that contain the pixel. $\phi_{i,1}$ can receive as parameters the limits $\delta_{\min}$ and $\delta_{\max}$ to remap the visual outputs according to the selected values. $\delta_{\min}$ and $\delta_{\max}$ represent the minimum respectively the maximum values the $A_{L,8}$ can have for the considered set of hotspots;

- $V_{i,2}$ = connecting line between two visual representations of $A_{L,2}$ attributes, reference and the referenced page (two hot-spot subsequently clicked during a usage session), $\phi_{i,2}$: draws a line connecting two visual representations of $A_{L,2}$ attributes using three visual clues to represent $A_{L,1}$, $A_{L,4}$ structural attributes and $A_{L,6}$, $A_{L,7}$, $A_{L,8}$ extended attributes: direction to code $A_{L,1}$, $A_{L,4}$, thickness to code $A_{L,8}$, and transparency to code $A_{L,6}$ and $A_{L,7}$. $\phi_{i,2}$ can receive as parameters the limits $\delta_{\min}$ and $\delta_{\max}$ to remap the visual outputs according to the selected values.

$V_{i,1}$ represents the Reference Map (RM) layer and, at the same time, Inspection Density Map (IDM) layer, while $V_{i,2}$ is the additional information layer that combines features of Links Usage (LU) and Viewing Time (VT) layers.
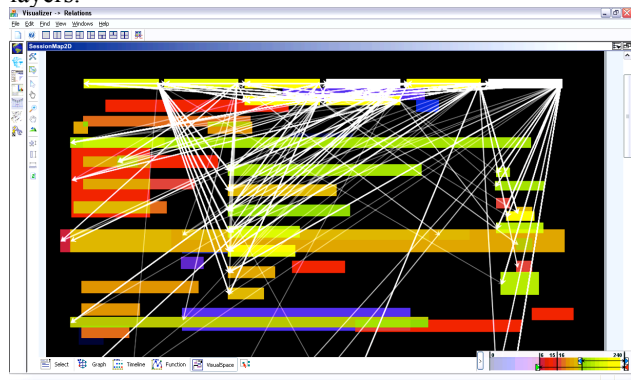

Figure 10 **Visual inspection of interface design (visual workspace) coherence**

First step is to use **FILTER** operator for each IL with the threshold specified by $\delta_{\min}$ and $\delta_{\max}$ parameters. The goal is to suppress the objects not in the specified usage interval. The second step is to use **ADD** operator for the two layers to obtain the representation by adding the objects in $V_{i,1}$ and $V_{i,2}$ to the representation. Note that the result depends of the threshold parameters applied for **FILTER**.

4. CONCLUSIONS

This paper contains a contribution for any researcher in the field of information visualization in the context of web based systems. The major theoretical dimension of the paper describes information attributes, information representation schemes and the possible information layers that can be used. The characterization of the specific attributes is contextualized in hypermedia systems, specifically, web information and technologies. The information visualization schemes and layers are based on research work and also have as reference many other authors of this research field.

This work represents the conceptual core for a research framework that also contains an experimental web log based management system in its second version. Some of this system's usage screenshots are used to clarify the need and/or purpose of some attributes and operators.

Several combinations of attributes, operators, operations and visual transformations of raw information into visual features have been already tested and good feedback has been collected during experimental usage scenarios.

The focus here is to combine the definitions introduced by this work to formally describe effective inspection instruments for the analysis of generic website structure, contents and usage patterns.

5. BIBLIOGRAPHY

[Card2001] Card K. Stuart, Pirolli, P., Van Der Wege, M., Morrison, J. B., Reeder, R. W., Schraedley, P. K., and Boshart, J. 2001. Information scent as a driver of Web Behavior Graphs - results of a protocol analysis method for web usability. In Proceedings of the SIGCHI *Conference on Human Factors in Computing Systems* (Seattle, Washington, United States). CHI '01. ACM Press, New York, NY, 498-505.

[Chen2004] Chen Jiyang, Sun, L., Zaïane, O. R., and Goebel, R. 2004. Visualizing and Discovering Web Navigational Patterns. *In Proceedings of the 7th international Workshop on the Web and Databases*: Colocated with ACM SIGMOD/PODS 2004 (Paris, France, June 17 - 18, 2004). WebDB '04, vol. 67. ACM Press, New York, NY, 13-18.

[Drott1998] Drott M. Carl. 1998. Using web server logs to improve site design. *In Proceedings of the 16th Annual international Conference on Computer Documentation*

(Quebec, Quebec, Canada, September 24 - 26, 1998). SIGDOC '98. ACM Press, New York, NY, 43-50.

[Faraday2000] Faraday Pete. 2000. Visually Critiquing web Pages. *Proceedings of HFWeb'00* (Austin, TX, June 2000). Online at: *http://www.tri.sbc.com/hfweb/faraday/faraday.htm.*

[Fraternali2003] Fraternali Piero, Maristella Matera, Andrea Maurino. 2003. Conceptual-level log analysis for the evaluation of web application quality. *First Latin American Web Congress* (LA-WEB'03), pp 46.

[GIS2005] Geographic Information Systems. 2005. U.S. Geographical Survey. *http://erg.usgs.gov/isb/pubs/gis_poster/*, Last visit: May 2006.

 [Healey2001] Christopher G. Healey, Robert St. Amant, and Jiae Chang. 2001. Assisted Visualization of E-commerce Auction Agent, *presented at Graphics Interface 2001 – Canadian Human-Computer Communications Society*, pp. 201-208, 7-9 June 2001.

[IFABC2006] IFABC Global Web Standards. *http://www.ifabc.org/standards.htm*. Last visit: March 2006.

[Ivory2002] Ivory Melody Y., Marti A. Hearst. 2002. Improving website design. *IEEE Internet Computing*, v.6 n.2, p.56-63, March 2002.

[Mealha2004] Óscar Mealha, Beatriz Sousa Santos, Nunes José, Florin Zamfir. 2004. Integrated Visualization Schemes for an Information and Communication Web Log Based Management System. *Proceedings of International Conference of Information Visualization – IV04*, London. July 2004.

[Tomlin1994] Tomlin C. Map algebra-one perspective. *Landscape and Urban Planning*, pp. 30(1-2):3–12, Octomber 1994.

[Zamfir2004] Zamfir F., Nunes J., Teixeira L., Mealha Ó., Sousa-Santos B., 2004. Visual Application for Management of Web-Based Communication and Information Systems. *Proceedings of IADIS International Conference Applied Computing 2004*, pp. II 119–125. Lisbon, Portugal.

[WCTD2006] Web Characterization Terminology & Definitions Sheet - *World Wide Web Consortium*. *http://www.w3.org/1999/05/WCA-terms/*. Last visit: March 2006.