



Florin Laurentiu Zamfir Website Visualizer - uma ferramenta para análise visual de utilização de sítios web: desenvolvimento e avaliação

Website Visualizer - a tool for the visual analysis of web site usage: development and evaluation



Florin Laurentiu Zamfir Website Visualizer - uma ferramenta para análise visual de utilização de sítios web: desenvolvimento e avaliação

Website Visualizer - a tool for the visual analysis of web site usage: development and evaluation

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica, Telecomunicações e Informática, realizada sob a orientação científica da Prof. Doutora Beatriz Alves de Sousa Santos, Professora Associada do Departamento de Engenharia Electrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientação do Prof. Doutor Óscar Emanuel Chaves Mealha, Professor Associado do Departamento de Comunicação e Arte da Universidade de Aveiro.

o júri

presidente

Prof. Doutor António Manuel Melo de Sousa Pereira
professor catedrático do Departamento de Engenharia Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Prof^a. Doutora Maria Beatriz Alves de Sousa Santos
professora associada do Departamento de Engenharia Electrónica, Telecomunicações e Informática da Universidade de Aveiro (Orientadora)

Prof. Doutor Óscar Emanuel Chaves Mealha
professor associado do Departamento de Comunicação e Arte da Universidade de Aveiro (co-Orientador)

Prof. Doutor João Manuel Gonçalves Duarte Cunha
professor catedrático convidado da Faculdade de Ciências da Universidade de Lisboa

agradecimentos

Gostaria, em primeiro lugar, de agradecer aos meus orientadores - Prof. Beatriz Sousa Santos e Prof. Óscar Mealha- por terem tornado possível a minha modesta contribuição para a vasta área que é a usabilidade e a visualização. O agradecimento não se limita ao seu exímio contributo profissional mas, principalmente, às genuínas preocupações demonstradas na minha integração na comunidade aveirense, fazendo-me sentir em casa e em família mesmo a 4000Km do meu país. As suas generosas contribuições ajudaram-me a tomar as decisões profissionais correctas durante estes quatro anos de residência em Portugal.

Gostaria também de agradecer ao Prof. José Nunes por todo o seu apoio profissional e amizade infinda, provando ser mais um irmão que um colega, reforçando as minhas decisões correctas e, quando necessário, corrigindo onde estava errado.

Gostaria também de agradecer ao Prof. Doutor João Cunha para a relevância dos comentários e feedback considerados neste documento, que representam uma contribuição relevante para a sua legibilidade e qualidade final.

O meu agradecimento vai também para o Professor Fernando Delgado, o meu supervisor directo no emprego durante a frequência do Mestrado, que sempre compreendeu as minhas necessidades e, generosamente, diminuiu a carga de trabalho para que eu pudesse terminar a dissertação.

Obrigado ao Kevin Tetu, o supervisor do meu actual emprego, por demonstrar genuínas preocupações e simpatia ao dar-me liberdade para terminar a dissertação.

Ao IEETA e ao DeCA que me acolheram durante o ano e meio de investigação que precedeu a presente dissertação, financiando todo esse período.

A todos os colegas e amigos com quem trabalhei e que me ajudaram a realizar o meu trabalho final, nomeadamente aqueles que directamente e indirectamente participaram nos testes e/ou estudos laboratoriais: Prof. Doutor Carlos Ferreira, Prof. Doutor Joaquim Madeira, Prof. Leonor Teixeira, pessoal do CICUA, estudantes do DETI e outros amigos.

Queria também expressar a minha profunda gratidão pela minha família: Felicia, Ramona, Vasile, Andrei, fam. Olaru, fam. Marcu, fam. Nicola, fam. Barbu, entre outros, que tiveram a capacidade de me apoiar na minha ausência, que coloriram a minha vida nos poucos momentos que passei com eles, e que fortalecem a minha alma quando penso neles.

"Last but not least", quero agradecer aos meus amigos Cristina, Elsa, André, Carlos, Jorge, Pedro, Hélder, Leonor, que tiveram a "coragem" ☺ de me apoiar sempre, fazendo-me sentir em casa.

A todos estes e aqueles que não mencionei, um muito obrigado.

acknowledgements

First of all, I want to thank Prof. Beatriz Sousa Santos and Prof. Óscar Mealha who supervised my work and made possible my small contribution to the measureless research field of usability and information visualization. Not only their professional collaboration, but also their parental concerns with my integration in Aveiro's community made me feel at home, with a distance of 4000 Km away from my family and friends. Their everlasting collaboration helped me support my professional decisions during the last four years of staying in Portugal.

I thank Prof. José Nunes for his professional support and endless friendship, who proved to be more like an older brother than a colleague, who supported me when I needed it most.

I thank Prof. João Cunha for his relevant comments and feedback taken into account in this document, representing a clear contribution for its final legibility and quality.

I thank Prof. Fernando Delgado, my direct supervisor on the job I worked in during my Master Course, who understood my needs and gave me the opportunity to complete my dissertation by providing me less workload.

I thank Kevin Tetu, the direct supervisor on my current job, who was concern and sincerely kind to provide me enough freedom to complete this dissertation.

To the Institute of Electronics and Telematics Engineering of Aveiro (IEETA) and Department of Communication and Art (DeCA) that hosted me during the one and a half years of research that preceded this dissertation, financing the entire period with a grant.

To all colleagues who worked with me and helped me accomplish my final work, those who directly or indirectly participated to tests and/or laboratory studies: Prof. Dr. Carlos Ferreira, Prof. Dr. Joaquim Madeira, Prof. Leonor Teixeira, employees of CICUA, students at DETI, and others.

I want to express my deepest gratitude to my family in Romania: Felicia, Ramona, Vasile, Andrei, Olaru family, Marcu family, Nicola family, Barbu family, etc. who had the power to support my decisions while I was gone, who made life warmer in the few moments I spent with them, and nourish my soul while I remember them.

Last, but not least, I want to thank my friends Cristina, Elsa, André, Carlos, Jorge, Pedro, Hélder, Leonor, who had the guts ☺ to support me all the way here and made me feel at home.

To all these and those not mentioned, thank you all.

palavras-chave

visualização de informação, avaliação de usabilidade, gestão de informação e comunicação, web logs, desenvolvimento e avaliação de aplicações.

resumo

Os sítios web estão incorporados em organizações para sustentar a missão das mesmas e para garantir uma difusão eficaz de informação num quadro de fluxo de trabalho eficiente. Neste contexto, os gestores de conteúdo e informação tem que monitorizar constantemente as necessidades inerentes à missão institucional e reflecti-las na estrutura, conteúdos e paradigmas de interacção das respectivas intranets e extranets. Esta tarefa de monitorização e análise não é de todo trivial, nem automática, sendo difícil garantir a sincronização dos sítios institucionais com as efectivas necessidades da sua missão em dado momento.

O objectivo fundamental deste trabalho traduz-se nos exercícios de conceptualização, desenvolvimento e avaliação de uma aplicação capaz de relatar um cenário de análise e visualizar padrões de interacção em sítios institucionais suportados em tecnologias web, que seja capaz de realçar as áreas mais críticas, com base na análise da estrutura, conteúdo e hiperligações. Para este efeito, propôs-se um modelo conceptual e uma arquitectura, bem como um conjunto de métodos de visualização que facilitem essa análise.

De forma a validar o modelo conceptual, a arquitectura, as estruturas de informação e os diversos métodos de visualização propostos, desenvolveu-se um protótipo que já comporta algumas fases de avaliação e aferição. Este protótipo pode ser considerado como uma plataforma de suporte à investigação capaz de integrar e testar esquemas específicos de visualização e procedimentos de correlação visual. Em suma, é parte integrante de um dos projectos de investigação da Universidade de Aveiro.

Mais especificamente, este trabalho introduz uma arquitectura por camadas que suporta vistas multiplas sincronizadas, bem como novos métodos de visualização, inspecção e interacção. O prototipo integra estes métodos de visualização numa aplicação capaz de capturar, compilar e analisar informação relacionada com a estrutura e conteúdo do sítio web, bem como padrões de utilização.

O protótipo destina-se fundamentalmente a dar apoio a especialistas de usabilidade ou gestores de conteúdo na organização do espaço de informação de um sítio institucional. Contudo, não se destina a produzir directamente soluções para problemas de usabilidade encontrados, mas sim a ajudar a tomar decisões com base nos problemas de usabilidade diagnosticados, identificados e sinalizados durante o processo de análise.

keywords

information visualization, usability evaluation, information and communication management, web logs, applications development and evaluation.

abstract

Websites are incorporated in organizations to support their mission and guarantee effective information delivery within an efficient information workflow framework. In this context, content managers have to constantly monitor the business needs and reflect them on the structure, contents and interaction paradigm of the institutional websites. This task is not trivial, nor automated, being difficult to guarantee that these websites are synchronized with the actual business requirements.

The overall goal of this work is the conceptualization, development and evaluation of an application able to assist usability experts in the analysis and visualization of interaction patterns of organizational web based systems. It should be able to highlight the most critical website areas, based on the analysis of website structure, contents and interconnections. For this purpose, a conceptual model and architecture has been proposed, as well as a set of visualization methods designed to facilitate that analysis.

In order to validate the proposed conceptual models, the architecture, information structures and several visualization methods, a prototype was developed, evaluated and refined. It can be considered as an experimental research platform, capable of integrating and testing specific visualization schemes and visual correlation procedures, and is part of an ongoing research program of University of Aveiro.

Specifically, this work introduces a layered architecture that supports simultaneously synchronised multiple views, as well as novel visualization, inspection and interaction mechanisms. The prototype integrates these visualization methods in an application able to capture, compile and analyze the information related to the structure, contents and usage patterns of a website.

This work is meant mainly to help usability experts or content managers to organize the informational space of an institutional web site. However, this application is not supposed to directly provide solutions for the usability problems of the site but to offer the means to help its users take decisions based on the interpretation of the usability problems identified and highlighted during the analysis process.

CONTENTS

Chapter 1. Introduction	11
1.1. Overview	11
1.2. Motivation and objectives	12
1.3. Main contributions	14
1.4. Dissertation outline.....	14
Chapter 2. State of the Art	17
2.1. Website classification.....	22
2.1.1. Hypermedia structure.....	22
2.1.2. Semantic content and information clustering	30
2.2. Usage Information.....	33
2.2.1. Gathering	33
2.2.2. Filtering	35
2.2.3. Classification	35
2.3. Visual correlations of structure, contents and usage	37
2.3.1. Visualizing structure	38
2.3.2. Visualizing content	39
2.3.3. Visualizing usage	42
2.4. Dynamic information exploration.....	45
2.5. Automated reporting and suggestion of usability improvements.....	47
2.6. Applications.....	48
2.6.1. Basic logs analysis.....	49
2.6.2. Structure and contents analysis	56
2.6.3. Usage analysis.....	59
2.7. Discussion.....	62
Chapter 3. Conceptual System Model and Visualization Methods	79
3.1. Conceptual model	80
3.2. Visualization methods	82
3.2.1. Visualization methods for visual workspace coherence.....	83
3.2.2. Visualization methods for website structure and session analysis.....	90
3.3. Visual correlations of visualization methods	97
Chapter 4. The Prototype: Objectives and Implementation.....	101
4.1. General objectives and system overview.....	101
4.2. The Prototype Implementation	107

4.2.1.	Background	108
4.2.2.	Prototype evolution strategy	108
4.2.3.	Application architecture and technologies used	110
4.3.	Site Analyzer	113
4.4.	Compiler	116
4.5.	Interceptor	119
4.6.	Visualizer	121
4.6.1.	Application layered model	121
4.6.2.	User Interface conceptual model.....	124
4.6.3.	Relational data structures.....	129
4.6.4.	Implementation building blocks and synchronization details	130
4.6.5.	Implementation of visualization methods	136
4.6.6.	Visual correlation strategies	144
Chapter 5.	Evaluation and Results	147
5.1.	Evaluation methods	147
5.2.	Results	150
5.3.	Proposed improvements.....	154
5.4.	Discussion	155
5.5.	Application to a Real Case	157
5.5.1.	Step 1: Information gathering	157
5.5.2.	Step 2: Information filtering	158
5.5.3.	Step 3: Classification and definition	158
5.5.4.	Step 4: Visualization, statistical analysis and results interpretation.....	159
Chapter 6.	Conclusions and Future Work	171
6.1.	Summary	171
6.2.	Conclusions	172
6.3.	Future Work.....	174
Chapter 7.	Bibliography	177
Annexes		187
1.	Session detection algorithm	187
2.	Database model and application framework	190
2.1.	Database Structure	190
2.2.	Application framework.....	194
3.	Evaluation details	199
3.1.	Procedures and Measures	199
3.2.	Exemplified evaluation tasks.....	201

LIST OF TABLES

Table 1.1 Website analysis and visualization - tools and techniques	69
Table 1.2 Website analysis and visualization - tools and techniques (continued)	70
Table 1.3 Website analysis and visualization - tools and techniques (continued)	71
Table 1.4 Website analysis and visualization - tools and techniques (continued)	72
Table 1.5 Website analysis and visualization - tools and techniques (continued)	73
Table 1.6 Website analysis and visualization - applications	74
Table 1.7 Website analysis and visualization - applications (continued)	75
Table 1.8 Website analysis and visualization - applications (continued)	76
Table 9 Number of users that completed or didn't do tasks without any question.....	151
Table 10 Number of users that completed correctly, incorrectly or did not do tasks having a question	152
Table 11 Evaluation task example.....	203
Table 12 Final questionnaire example.....	205
Table 13 Example task with check items.....	206

LIST OF FIGURES

Figure 1 Andy Cockburn and Steve Jones proposed taxonomy for web usage experience problems	18
Figure 2 Summary of the Melody Ivory's and Marti Hearst's proposed taxonomy for classifying usability evaluation methods	19
Figure 3 Chronological representation of website monitoring and analysis systems and studies.....	20
Figure 4 IBM's Haifa Research Lab - Mapuccino Site Map Application (Vertical Tree vs Fish Eye).....	23
Figure 5 Spreadsheet with <i>Disk-Tree</i> representation of site structure and usage	24
Figure 6 <i>Time-Tube</i> with <i>Disk-Tree</i> representations.....	25
Figure 7 <i>Cone-Tree</i> representation of usage patterns and evolution	25
Figure 8 <i>WebKIV</i> focus view (left) and context window (right).....	26
Figure 9 Layered <i>web Image</i> Visualization (ViewTime, ProbabilityUsage, ExitPoints).....	26
Figure 10 Usage sessions with site usage data.....	27
Figure 11 <i>VisVIP</i> - Path laid (left) over a website structure (right)	28
Figure 12 The H3 3d hyperbolic browser	28
Figure 13 Filippo Ricca and Paolo Tonella's hierarchical representation of the site.....	29
Figure 14 Directed Graph (left) and Hierarchical (right) representations	29
Figure 15 Robert Cooley's – Web Usage Mining process	31
Figure 16 <i>WARE</i> : class diagram of the web application ' <i>Juridical Laboratory</i> '	32
Figure 17 UI of <i>LinkViewer</i> : Successful vs. unsuccessful examples of page analysis....	33
Figure 18 Browser extension for event logging used by <i>WebRemUSINE</i> system.....	34
Figure 19 Page content exploration and event interception.....	40
Figure 20 Example of WBG for specific search tasks	40
Figure 21 Design Advisor's visual clues.....	41
Figure 22 WebTango UI analysis tool taxonomy	42
Figure 23 Example of surfing animation (sample slides)	43
Figure 24 <i>WebRemUSINE</i> : display of page visit timings	43

Figure 25 Frequent access patterns (white graph, left image) and the addition of frequency of pattern attribute rendered as thickness (right image)	44
Figure 26 <i>WebRemUSINE</i> : display of a user groups task performance.....	45
Figure 27 Web Behavior Graphs	45
Figure 28 Ivory's website structure taxonomy	47
Figure 29 <i>AWStats</i> example of statistics graphical outputs.....	51
Figure 30 <i>Wusage 8</i> - Example of report organization	51
Figure 31 <i>Webtrends</i> example - classic reports represented by graphs	52
Figure 32 <i>Sitelogz</i> example report.....	52
Figure 33 <i>Sawmill</i> example report	53
Figure 34 <i>FastStats</i> report example	53
Figure 35 <i>Opentracker</i> country identification report example	54
Figure 36 <i>Deep Log Analysis</i> hierarchical interactive presentation	55
Figure 37 <i>iWEBTRACK</i> example report	55
Figure 38 <i>ISA Server 2004</i> traffic report example	56
Figure 39 <i>LiveSTATS</i> reports examples	57
Figure 40 <i>ClickTracks</i> overlaid usage information.....	57
Figure 41 <i>ClickTracks</i> synchronized view with overlaid usage data.....	58
Figure 42 <i>Opentracker</i> online visitors report example	58
Figure 43 <i>Webtrends</i> example - Overlaid web usage data.....	59
Figure 44 <i>FastStats</i> hyperlink tree view example	59
Figure 45 <i>Webtrends</i> example - <i>scenario analysis</i>	60
Figure 46 <i>Webtrends</i> examples - Contents effectiveness of usage data through path analysis.....	61
Figure 47 <i>Ehtnio</i> proof of concept.....	61
Figure 48 <i>Google Analytic</i> funnel visualization and site overlay	62
Figure 49 Conceptual System Model.....	80
Figure 50 System Model – Conceptual Components	81
Figure 51 <i>Page Areas</i> visualization	83
Figure 52 <i>Interactive Zones</i> 2D visualization.....	84

Figure 53 <i>Interactive Zones</i> 3D visualization	85
Figure 54 <i>Page Relations</i> visualization	86
Figure 55 <i>Hovering Tips</i> visualization	87
Figure 56 <i>Eye-Tracking Layers</i> visualization	88
Figure 57 <i>Mouse-Tracking Layers</i> visualization	88
Figure 58 <i>Interaction Workspace</i> visualization	89
Figure 59 <i>Site Pages</i> visualization	90
Figure 60 <i>Site Structure 2D</i> visualization	91
Figure 61 <i>Site Structure 3D (Complex Hierarchical 3D View)</i> visualization	92
Figure 62 <i>Site Structure 3D (Holistic 3D View)</i> visualization on the left, the conceptualized version on the right	93
Figure 63 <i>Linkage Elements</i> visualization	93
Figure 64 <i>Path to Goal</i> visualization	94
Figure 65 <i>Session History</i> visualization	94
Figure 66 <i>Session History 3D</i> visualization	95
Figure 67 <i>Tree-Structured Traversing in Time</i> visualization	96
Figure 68 <i>Page Linkage</i> visualization	97
Figure 69 Tightly coupled synchronized views concept	99
Figure 70 Simplified model of the system	104
Figure 71 Analysis timeline	106
Figure 72 Website hierarchical representation	107
Figure 73 Layered System Architecture	111
Figure 74 System Simplified Architecture	111
Figure 75 Main view of the <i>Site Analyzer</i> prototype	114
Figure 76 Portal architecture example	116
Figure 77 <i>Compiler</i> prototype - user interface	118
Figure 78 Layered conceptual model of <i>Visualizer</i>	121
Figure 79 Visualization modules	123
Figure 80 User Interface model	125
Figure 81 Interface objects flexibility	126

Figure 82 Possible interface manipulations	127
Figure 83 Wizard steps (left to right, top to bottom).....	128
Figure 84 User interface aspects.....	128
Figure 85 Visualizations hierarchy.....	132
Figure 86 Visualization elements hierarchy	133
Figure 87 Filter selection threshold with color lookup table	133
Figure 88 Visualization lifecycle.....	134
Figure 89 ID_SHOW_ALL_PATHS message routes.....	136
Figure 90 Offline Site Explorer visualization	138
Figure 91 Relational visualization of site structure	139
Figure 92 Path to goal visualization.....	139
Figure 93 Interactive zones visualization.....	140
Figure 94 Page relations visualization	141
Figure 95 Tree-structured traversing in time visualization	142
Figure 96 Visual workspace coherence visualization	143
Figure 97 Synchronized visualizations triggered by the same event.....	144
Figure 98 Boxplots corresponding to times in all tasks for Users#1 and #2.....	151
Figure 99 Boxplots corresponding to times and satisfaction task by task	153
Figure 100 Overall website statistical information example.....	158
Figure 101 Case Study - Detailed Site Structure.....	160
Figure 102 Threshold that depicts intense backwards navigation from the third to second levels.....	161
Figure 103 Semantically related contents prove average usage statistics	161
Figure 104 Page inspection for interface design coherence combined with statistical information.....	162
Figure 105 Possible navigational problem with the usage of browser history	163
Figure 106 Website visual interaction workspace.....	164
Figure 107 Visual workspace coherence for users inside the University network.....	165
Figure 108 Visual workspace coherence for users outside the University network.....	166
Figure 109 Visual workspace coherence for teachers and administrators	166

Figure 110 Visual workspace coherence for all users and all sessions	167
Figure 111 Browser's history back navigation example	168
Figure 112 Backward navigation example on tree-structured traversing in time	168
Figure 113 Backward navigation to highly connected pages	169
Figure 114 Application maintenance information entities	191
Figure 115 Website information entities.....	192
Figure 116 Webpage information entities	193
Figure 117 Framework – network communication	195
Figure 118 Framework – application subsystem, data access and image manipulation	196
Figure 119 A more detailed view of framework integration	197
Figure 120 Framework – user defined controls	197
Figure 121 Roles of the students	200

Chapter 1. Introduction

1.1. Overview

Nowadays, worldwide institutions developed and supported important informational frameworks, based on the latest hypermedia presentation systems. As a result, an emerging problem related to the management and understanding of these vast informational workspaces has occurred.

From the technical point of view, the latest trends of Internet based technologies gave the institutions the means to develop large websites, meant mainly to support their strategic and operational decisions. However, the management of such systems became more and more difficult, due to the unpredictable growth facilitated by these technologies. The Internet became an efficient instrument to support business decisions, the prediction and understanding of its potential representing an advantage for every institution. Considering that organizations can be understood as organic systems, composed of people, processes and procedures, the intranets / extranets are incorporated in organizations to support their mission and guarantee effective process delivery within an efficient framework. These technological platforms also provide internal asynchronous communication services that help in process development and contribute to a more efficient outcome of overall mission results.

One of the main goals of web usability experts is to analyze the relations between the web site structure, its contents and the usability issues that might influence the overall usage experience of a website. Contents and navigational paradigms determine website users to make good or bad navigational decisions or interact with the website's user interface (UI) in a manner that was not predicted by the designers. Understanding how contents and UI layouts influences user's decisions is a priority.

In this context, good structural and usage analysis and diagnosis tools are required to support the managers in decision-making. They must be designed to cope with those sophisticated hypermedia infrastructures, allowing the identification of user-system mismatch at the human-computer communication level. These tools must support quantitative and qualitative correlated representation and inspection features of the analyzed websites; however, the development of such applications is not an easy task. Potential problems should be located and handed on to the design or redesign team.

1.2. Motivation and objectives

In the context of constantly growing hypermedia structures, motivated by structural organization and decisional behaviors of worldwide institutions, an important role is assigned to information and content managers, their task is to constantly monitor business needs and reflect them on the structure, contents and interaction paradigm of the associated intranet and extranet websites. However, this is not trivial, nor automated, the sophisticated dynamics and information flows of nowadays businesses making even more difficult to focus and capture the dynamic evolution of all areas of the websites. An emerging problem occurs at this level, to guarantee that the institutional websites are synchronized with the actual business requirements.

In this context, good feedback instruments for automated problem/solution identification are fundamental. Some applications do exist; however, they tend to be biased by classical technical metrics for technical tuning, not for organizational communication and information analysis [Tauscher1997], [Bieber1997]. Some recent solutions are focused on some specific areas of website representation and do not promote a complete system able to perform both types of analysis, quantitative and qualitative: [Chi1998], [Drott1998], [Chi1999], [Becker1999], [Faraday2000], [Card2001], [Ricca2001], [Chi2002], [Ivory2002], [Niu2003], [Eick2004], [Ruffo2004], [Chen2004]. Other recent commercial applications have also focused on usability problems discovered as a result of the analysis of website usage log files, such as: [AWStats2005], [ClickTracks2005], [Deep2005], [Ethnio2005], [FastStats2005], [GoogleAnalytics2005], [ISAServer2004], [IWEBTRACK2005], [LiveSTATS2005], [Opentracker2005], [Sawmill2005], [Webtrends2005], [Wusage2005]. These solutions tend to be more technical and provide quantitative measures mainly for the usage of site areas, partially for some elements of site pages, or even for information flows derived from the processing of quantitative usage information, however, they do not provide qualitative or interface design level usability analysis.

The objective of this thesis is to introduce innovative visualization methods, and to conceptualize, implement and evaluate a prototype of an application able to deal with the analysis and representation of the complex hypermedia structures and information flows collected inside an institution, mainly from the analysis of the web site structure and site usage logged information (either obtained during natural site usage or controlled experiments). This application is meant mainly for helping usability experts or content managers to organize the informational space of an institutional web site. However, it is not meant to directly provide solutions for the usability problems of the site; instead, it offers the means that help its users take decisions based on the interpretation of the usability problems identified and highlighted during the analysis process.

The aim of this application is to provide help for answering the following general question:

- How is the site used?

Starting from this general question, a specific sub-set of questions can be detailed, whose answers might provide useful information:

- 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?

The solution adopted is to present the information through different visualization methods that explore the enormous capacities of the human visual system, in order to provide help for answering the following additional 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 users' navigational decisions?

In a first stage of this work we consider existing similar applications and then continue with the overall organization we propose for our research framework, the evolution of the development cycle, the technical presentation of the implementation, a complete evaluation cycle and an application to a real scenario. To carry on the proposal, at least one prototype of the application has to be implemented and evaluated, to help us determine which aspects are to be maintained on our research agenda and which have to be improved.

This thesis presents the author's contribution to an ongoing research program of University of Aveiro, Portugal. It focuses the conceptualization of several innovative visualization methods, and follows the complete life-cycle for the implementation of a prototype that integrates the visualization methods, more specifically, conceptualization, development and evaluation of the prototype and the proposed visualization methods. The main difference between this thesis and the work presented by Nunes in [Nunes2006] is that this one focuses more on the development and evaluation of the prototype itself and the integrated visualization methods proposed, while the other focuses more on the theoretical aspects and conceptualization of the proposed research framework as a whole. However, some conceptual aspects of the visualization methods, implementation and evaluation have to be presented in both cases, as a result of a tight collaboration.

1.3. Main contributions

A brief enumeration of the most important contributions of this thesis can be summarized as follows:

- A combination of several quantitative and qualitative measures was introduced to analyze not only the website usage patterns, but also the interface design coherence or navigation coherence;
- The usage of visual reporting capabilities was considered, to pinpoint possible inconsistencies regarding the website structure, navigation or interface design;
- Some inspection tools (e.g. statistical image tips) and visual synchronization mechanisms were introduced to enrich complementary visualization methods. The interaction effects of active view is simultaneously propagated to all views, with complementary different visual effects on each other;
- A innovative website design and maintenance tool was introduced, to provide the means to redesign the website structure and contents with less effort, as a result of a preliminary analysis session;
- Several innovative visualization methods are of a great importance for our proposed system as “Interactive Zones and Page Relations”, “Visual Workspace Coherence” and “Inspection of tree-structured information traversing in time”.

1.4. Dissertation outline

The thesis is divided into six chapters. The first two chapters have an introductory character, while the remaining chapters are directly related to the proposed visualization methods, implementation and evaluation of a prototype application.

The first chapter, Introduction, gives a general overview of the motivation, the objectives of this thesis and some of the main concepts promoted; it also introduces the structure of this document.

Chapter Two, State of the Art, discusses the current research trends on the field of website analysis and representation, as well as recent commercial applications available on the market. It is subdivided in six sections, each one dedicated to a specific topic of interest: Website , Usage Information, Visual correlations of structure, contents and usage, Dynamic information exploration, Automated reporting and suggestion of usability improvements and Applications. This latest section is subdivided in three topics of interests: Basic logs analysis, Structure and contents and Usage analysis. The timeline included in this chapter shows a relevant part of the chronological evolution of this research area and highlights some of the Aveiro research group papers published during the 1997-2006 period.

The third chapter, Conceptual System Model and Visualization, presents the conceptual model of the proposed application, theoretical aspects regarding website representation and information visualization, the conceptual information structures required for visualization and analysis purposes, as well as a conceptual description of the proposed visualization methods and inspection mechanisms.

The fourth chapter, The Prototype: Objectives and Implementation, focuses on the general objectives, implementation details and the evolution of the proposed prototype. This chapter discusses the objectives, proposed functionality and development strategy, gives an overview of the implementation and presents the goals for the current prototype version, the implementation and preliminary evaluation, as well as the latest development stages. The four conceptual modules of the application are briefly described: Site Analyzer, Interceptor, Compiler and Visualizer. Since Visualizer represents the focus of this thesis, it is discussed in detail. Finally, the evaluation of the main aspects of the user interface and the visualization methods implemented by the Visualizer application are addressed.

The fifth chapter, Application to a Real Case, focuses on an example of usage of the prototyped system to analyze a real website. Data preparation phase, visualization, statistical analysis and results interpretation are discussed and some examples of highlighted usage patterns or identified problems are shown.

The sixth chapter, Conclusions and Future Work, gives an overview of the main achievements and critiques of this thesis, and presents possible improvements and additional functionalities to be considered in the future evolutions of the application.

Finally, the Bibliography is presented in the seventh chapter and the latest chapter, Annexes, introduces some technical details of the implementation.

Chapter 2. State of the Art

Even if there were several success stories, the e-commerce failure in the late 90's could not be avoided [Nielsen2001]. There were several reasons that influenced this fact, one of the most important being the overall poor design of websites. As [Chi2002] highlighted, "usability can make or break a website".

Lately, website usability became a very important issue on the agenda of every respectable website designer and / or system administrator. Several website usability techniques have been proposed during the last decades, each providing a specific measure for usage success or failure of a website. All these techniques can be easily classified accordingly to the following aspects of the website involved:

- i.* Structural website organization;
- ii.* Content classification (web mining);
- iii.* Visual representation of the content;
- iv.* Navigation paradigm;
- v.* Information availability.

Each and every of the previously presented aspects provides important information that helps a usability expert understand the impact the website has on its final users:

- The *structural website organization* can influence the user's access to the information; a poor website structure might hide some aspects of its contents;
- The *content classification* has an important impact on the perception of the website, the organization of the information being a critical issue. A content-chaotic website might cause serious problems for its users and, therefore, the use of web mining tools for the optimization process is a difficult task to be performed by the evaluator(s); when personalization is used [Cooley2003] the analysis is even more difficult;
- The *visual representation of the content* of a website is crucial for its success. All the aspects regarding the representation of the content have a direct influence on the site user's satisfaction, therefore, a poor design might create frustration or even rejection by the users [Ivory2002];
- The *navigation paradigm* can easily confuse a website user, the lack of feedback of navigational context or a poor web navigation scheme might bring frustration to the users, therefore, negatively influence the decision of the users to revisit the website [Card2001], [Ricca2001];

- Last but not least, the *information availability* of a website can easily influence the exploration decisions of its users. Poor response time or temporarily unavailable web pages can make an anxious user leave the current navigational context or even never come back to that context [Ruffo2004].

In [Cockburn1997] the authors present an evaluation of tools meant to augment browser navigation support based on providing visual history support and potential discovery of web links, using link previews. Twelve visualization tools were analyzed with the purpose of discovering novel visualization methods of the web hyper-space, in the end, the authors highlight the new features brought by each visualization tool. They proposed three origins as taxonomy for browsing experience problems (Figure 1):

- Origin 1: Browser User Interfaces;
- Origin 2: WWW Subspace Design; and
- Origin 3: WWW Subspace Description Language.

While analyzing the twelve browser tools, the authors focused on one approach meant to improve the previously presented problems as augmentation of browsing applications with visualizations of WWW subspaces.

Origin of Problem	Proportion of users affected	Example problems	Potential solutions
Browser Interfaces (Origin 1)	Very high (all WWW browsing is carried out through a handful of browsers).	Misunderstanding of client's facilities (for example, Back and Forward).	Improved system image to better communicate the system's facilities to the user.
		Range of client's facilities (for example, the absence of an interactive history list).	Extended facilities for user support.
Page Design (Origin 2)	Small for each site, but errors of page design are common across many sites.	Poor structure within a WWW site (promoting "lost in WWW space").	Tools to support structured WWW design (see section 2.2).
		Poor graphical design (consistency in representation, legibility and readability, visual appeal, etc).	Guidelines for page designers (for example, [NS95]).
Hypertext Markup Language Features (Origin 3)	Very high (almost all pages are written using a few dialects of HTML).	Restricted range of expressible hypertext facilities. Inability to affect browser state (such as the history list).	Solutions to these problems are unlikely. There are strong conflicts between the need for standardisation, for advanced features, and for security.

Table 1: *Origins of user problems in WWW navigation, the proportion of users affected, example problems, and potential solutions.*

Figure 1 Andy Cockburn and Steve Jones proposed taxonomy for web usage experience problems [source: [Cockburn1997], Table 1, page 3]

Because of the dimensions, complexity and dynamics of nowadays websites, automated tools are needed, to cope with the requirements of large amounts of information

manipulations meant for analyzing these enormous structures. Even if the software design technologies continuously evolved in the last decades, such completely-automated tools were not fully implemented; dynamic website generation, personalization and content classification being some of the most important factors that influenced the design of such sophisticated tools.

[Drott1998] proposed descriptive methods for evaluating websites, focused on statistical analysis of usage patterns discovered in server logs. The author introduced three task-oriented methods for the detection of usage patterns by analyzing website usage log files.

[Ivory2002] proposed a possible taxonomy for the classification of user interface evaluation techniques (as shown in Figure 2), and presented an extensive survey of usability evaluation methods, organized according to a new taxonomy that emphasizes the role of automation.

To facilitate the analysis of the state of automation in usability evaluation, the authors grouped usability evaluation methods along the four dimensions:

1. *Method Class*: describes the type of evaluation conducted at a high level (e.g. usability testing or simulation);
2. *Method Type*: describes how the evaluation is conducted within a method class, such as thinking-aloud protocol (usability testing class) or information processor modeling (simulation class);
3. *Automation Type*: describes the evaluation aspect that is automated (e.g., capture, analysis, or critique); and
4. *Effort Level*: describes the type of effort required to execute the method (e.g., model development or interface usage)."

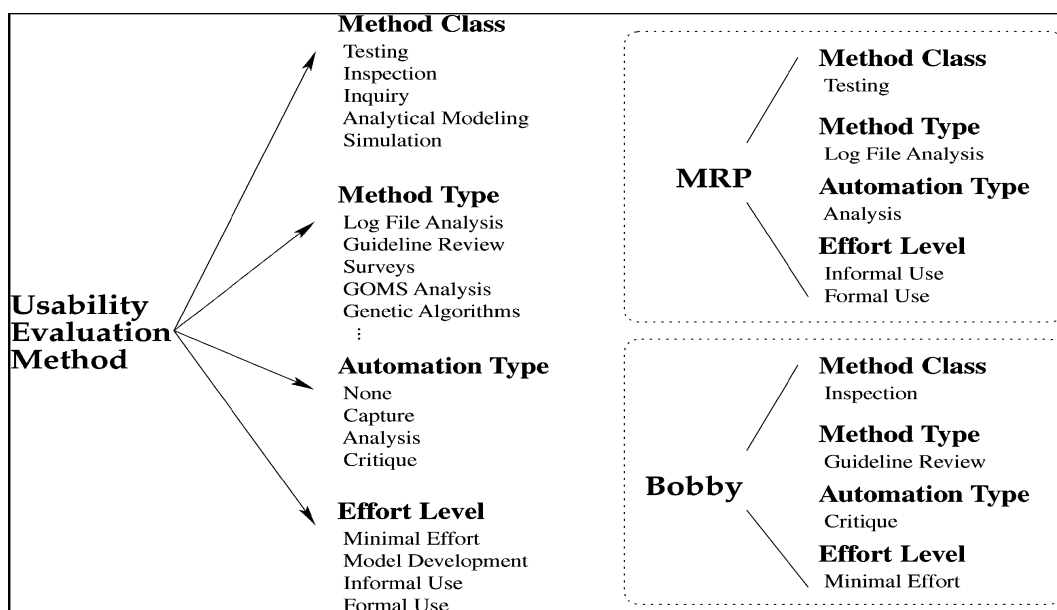


Figure 2 Summary of the Melody Ivory's and Marti Hearst's proposed taxonomy for classifying usability evaluation methods [source: [Ivory2002], Figure 1, page 474]

Using the classification of website properties for measuring purposes (as introduced in the beginning of this chapter) and considering several automation, visualization and interaction aspects, we classified the analyzed website visualization and usability evaluation techniques in five major classes that deal with:

1. *Website Classification* – methods for the classification of *hypermedia structure* of websites and the *semantic content and information clustering techniques* used to classify the web content – meant to distill and classify the large amounts of information concentrated in websites. Representation transcribes the way the website is analyzed and represented, using breadth or depth first algorithms, etc.;
2. *Usage Information* – information analysis techniques that can be classified according to three major areas of interest: *gathering, filtering and classification*. These techniques implement automated solutions for the analysis of server stored web usage log files, and discovery of usage patterns based on the information provided by these logs. In addition, some of them implement automated solutions for intercepting live (real-time) website usage, discovering and analyzing usage patterns by processing the interactions and events produced by clients during real-time navigation;
3. *Visual Correlations of Structure, Content and Usage* – techniques used for the analysis of visual organization of website structure and contents and the impact of these structures on the navigational paradigms embedded in the website design, in close relation with its usage. In addition, means of interaction are included in this class. The visual correlations are used to emphasize the discovery of usage patterns;
4. *Dynamic Information Exploration* – techniques of exploration and interaction mechanisms provided by the visualizations we considered for analysis;
5. *Automated Reporting and Suggestion of Usability Improvements* – automated methods for the analysis and reporting of usability aspects of the websites and suggestion of possible improvements or highlight problems. They are based on sophisticated algorithms able to analyze the web page clutter, design layout, content and several other important aspects.

However, the considered website analysis and visualization techniques tend to have some limitations because of the implementation or because of the selected user interface paradigm. Therefore, the evaluation of these analysis frameworks proposed is a key factor for their validation.

A chronological representation of website monitoring and analysis systems and studies is shown in Figure 3.

The following sections briefly describe the most important proposals / applications we have analyzed, provide an overview of the past and current trends in automated website

usability analysis, as well as web structure and content information visualization techniques. Some of these proposals cover several topics discussed in this chapter, thus, they are presented several times, accordingly to the focus of each section.

2.1. Website classification

During the past decade, several visualizations were proposed to visualize the website structure. Basically, all the visualizations start from the entrance point of the site (usually called the root page) and use a breadth first (BFS) or depth first (DFS) traversal of the graph associated to the site (the most obvious representation of a website can be considered a set of graph nodes connected by directional links). BFS proved to be a more reliable traversal in terms of balance of the tree [Najork2001], [Chen2004]. The approach of traversal is possible for common websites, in which a unique node of the graph is represented by the corresponding URL / URI (Unique Resource Locator / Identifier). Lately, the approach of uniquely identified resource proved to be an issue for dynamic websites that use the same URL to represent different types of contents, based on user interaction and personalization. Since the actual parameters of a page can be hidden in POST requests, it is difficult to discover all possible paths within a dynamic website; we also consider here the dynamic personalization. This is a constraint since several usage analysis preprocessing steps cannot be completed without having a stable classification of the website structure.

2.1.1. Hypermedia structure

Starting from the premise that all possible nodes of the website representation graph have been discovered and classified, several visualizations of the same information can be used, based on the dimension of the representation space and layout, e.g.:

- i.* Two dimensional space (2D):
 - Hierarchical tree visualizations;
 - Radial tree visualizations;
 - Clustered visualizations;
 - Hyperbolic visualizations; etc.
- ii.* Three dimensional space (3D):
 - Clustered spherical visualizations;
 - 3D Hyperbolic visualizations;
 - Row by column visualization with circuits;
 - Time tube – combined radial tree slices coded in time;

- Hemisphere hierarchy visualization;
- Cone tree visualizations; etc.

[Dodge2003] surveyed several applications able to construct site maps for the visualization of the hypermedia structure. Several visualizations were addressed, from 2D representation as hierarchical tree (Figure 4 – left quadrant), radial tree (Figure 4 – right quadrant – Fish Eye visualization), (Figure 5, Figure 6, Figure 8, Figure 9) and hyperbolic (Figure 12), to 3D representations of the website. Some of these visualizations and some others were the basis for several commercial and noncommercial implementations discussed in the section 2.6 - Applications.

In [Eick2004], Stephen G. Eick described several website visualizations with the purpose of understanding the activity of online users and their behaviors. The author concluded that the website analysis and visualization tools require a deeper research on the following areas: *Scalability*, *Support Action* (“Information visualizations do not by themselves create value but lead to valuable insights. Turning insights into decisions and activities that add value requires an action step”) and *Taxonomies that define the appropriate analysis problems as information visualization might address*.

[Andrews2002] introduced an important survey of information visualization techniques highlighting several techniques used to represent hierarchies, which are the basic organizational units of websites. In this work, the author presented detailed functional aspects of each analyzed visualization, behavior and visualization being the focused areas of interest.

[Benford1999] surveyed different approaches to create 3D visualizations of the World Wide Web. Several issues were addressed: visualizations of the structure of the web, the history of browsing sessions, searches, the evolution of the web, and the presence of other users on the web. The author focused on presenting aspects regarding visualization styles and techniques to manage scalability, interaction and information sharing.

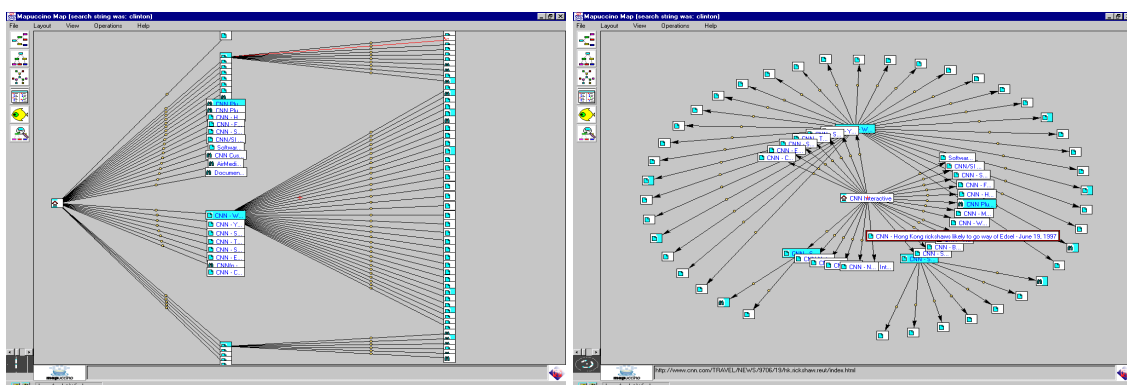


Figure 4 IBM's Haifa Research Lab - Mapuccino Site Map Application (Vertical Tree vs Fish Eye)
[source: [Dodge2003], Figure 7, page 3]

Ed H. Chi's *Disk-Tree* [Chi1998], [Chi2002] uses the 2D and 3D representations of the information based on a radial representation of the site graph (Figure 5 and Figure 6); the representation starts from the root page, as the center of the graph, and continues with the website levels concentrically represented. The *Disk-Tree* was the first to display the web usage and structure information together by mapping the usage data on the structural objects. The graph representing the website structure is collapsed into a "disc" using a breadth-first search algorithm. If a page is linked from many other pages, only the first link found with the breadth-first search is kept and represented. This technique has been used to visualize website evolution, web usage trends over time, and evaluation of information foraging. The website usage information is displayed using color and thickness coding techniques (Figure 5).

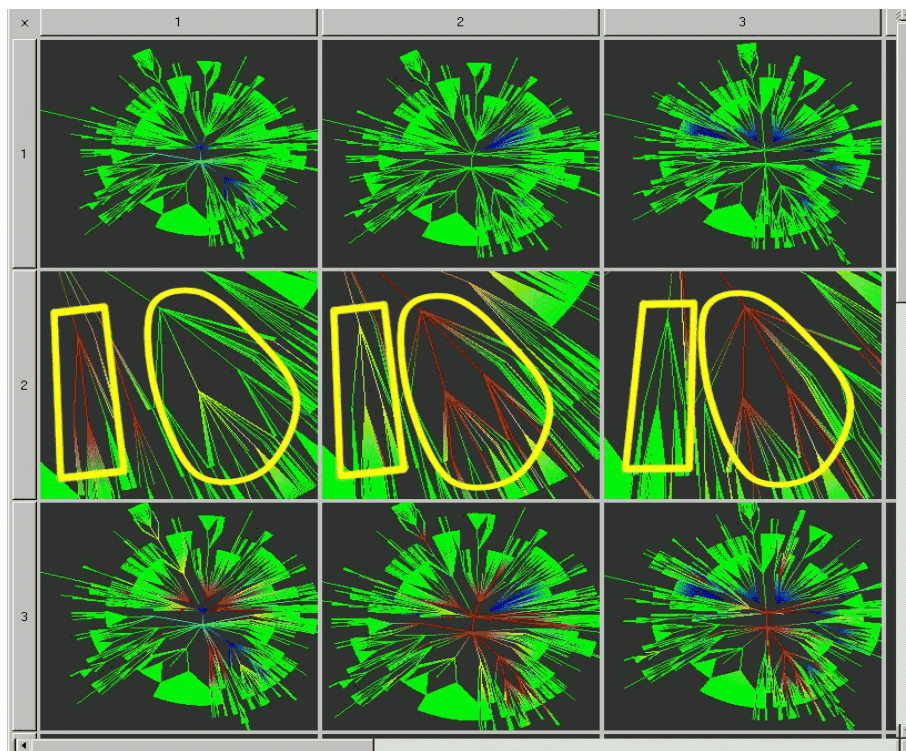


Figure 5 Spreadsheet with *Disk-Tree* visualization of site structure and usage [source: [Chi1999], Figure 3.16, page 62]

Another feature is to map the evolution of the website structure using color and thickness, to highlight its changes (Figure 6). The concept of *Time-Tube* visualization was proposed as a series of representations constructed at different time periods aligned together for comparison. It is then used to identify each version of the website by aligning several *Disk-Trees* on the third dimension, the time (Figure 6).

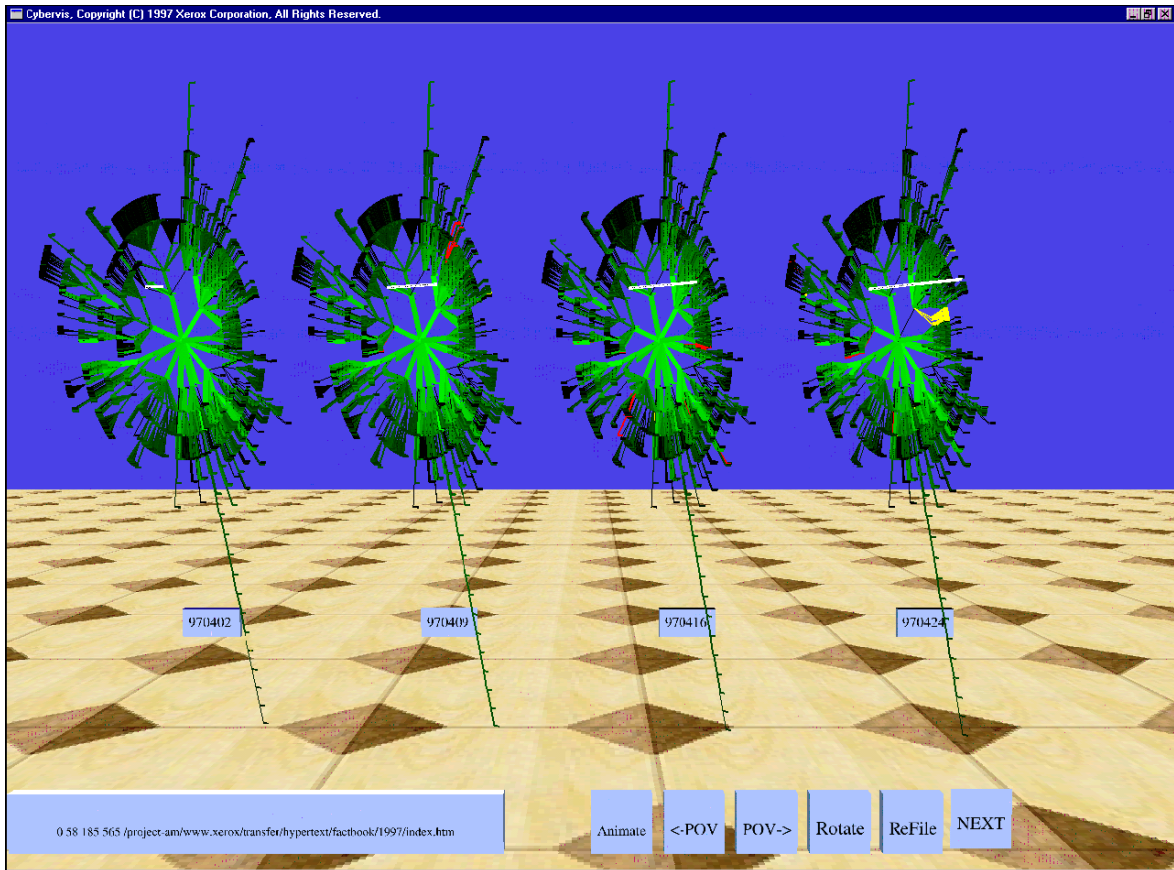


Figure 6 *Time-Tube* with *Disk-Tree* visualizations [source: [Chi2002], Figure 3, page 68]

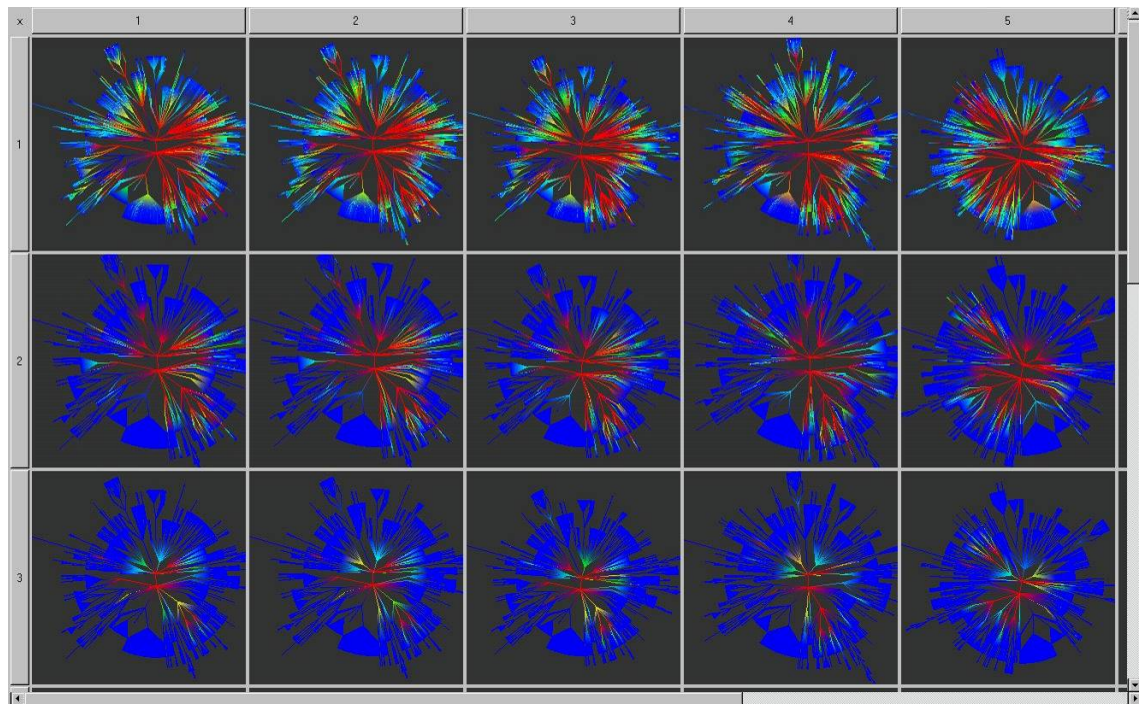


Figure 7 *Cone-Tree* visualization of usage patterns and evolution [source: [Chi1999], Figure 3.14, page 59]

Yet, another proposal of the same author [Chi1999] is the *Cone-Tree* visualization technique which is a 3D radial representation of *website levels*, each area of the site being modeled as a cone in the three dimensional space (Figure 7). Even if it provides similar results as the *Disk-Tree* visualization (Figure 5), occlusion might hide some of the details.

An improved version of *Disk-Tree* was proposed in [Niu2003] as the *WebKIV - Web Knowledge and Information Visualization* tool. The proposed tool is able not only to display a static structure of a website, but also to visualize web usage data and allow interaction. Multiple user session can be animated to provide feedback of site usage patterns. This tool is able to represent websites with a structure of up to 70000 pages, scaling up the capabilities of *Disk-Tree* proposed by Ed H. Chi. *WebKIV* uses a visualization of aggregate and individual navigation usage behavior, and the comparative display of navigation improvements (Figure 8). It combines several visualization strategies from other web visualization tools, to provide a single method of visualizing web structure, and the results of web mining on that structure.

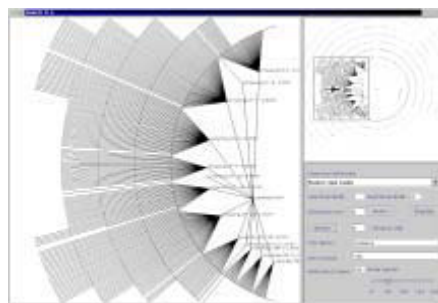


Figure 8 *WebKIV* focus view (left) and context window (right) [source: [Niu2003], Figure 2, page 3]

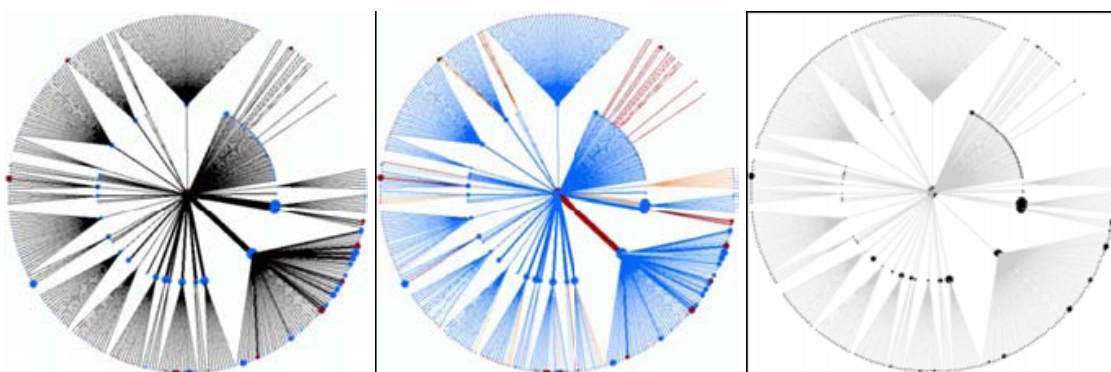


Figure 9 Layered *web Image* Visualization (ViewTime, ProbabilityUsage, ExitPoints) [source: [Chen2004], Figure 1 - C, D, page 15, Figure 5 - B, page 17]

Perhaps the most complete radial representation of website structure was introduced by [Chen2004]. The authors proposed a multi-tier taxonomy (Web Knowledge Visualization

and Discovery System - WEBKVDS) that divides the visualization space in several layers used to represent different types of information in which the web graph is represented as a multi-tier object (Figure 9). The first tier, called a *Web Image*, is a tree representing the structure of a given website (or subset of it), and is represented as the background of the visualization. Each other tier, called an *Information Layer*, represents some statistics about web pages or the links connecting them. Combining the tiers allows putting the navigational data in its web structural context. The new features are represented by the proposed web graph algebra that combines different information layers to obtain visual insight about the web structure, usage data and whatever usage patterns discovered during the mining process. Color and thickness are then used on separate layers to code different quantitative and qualitative measures such as: *Number of Visits*, *Link Usage*, *View Time* (Figure 9– left quadrant) and *Probability Usage of a Link* (Figure 9 – middle).

According to the authors: “The *Web Image*, also referred to as *bare graph*, is a tree representing the structure of a website or a subset of it. It has a root, which is a starting page, and a certain given depth. Each node in the tree represents a web page and an edge represents a hyperlink between pages. A website is in reality a graph, but it is collapsed into a tree. When visualized, the tree is displayed as a disc with the centre being the root, and the nodes of each level displayed on a circular perimeter, each level successively away from the centre.”

[Youssefi2003] presents a preliminary proposal that uses 3D visualizations rather than a 2D layout techniques to visualize the web structure and map it with web usage data. The visualizations of the website combine circular, hierarchical and clustered visualizations of the website structure, each enriched with web usage information (Figure 10).

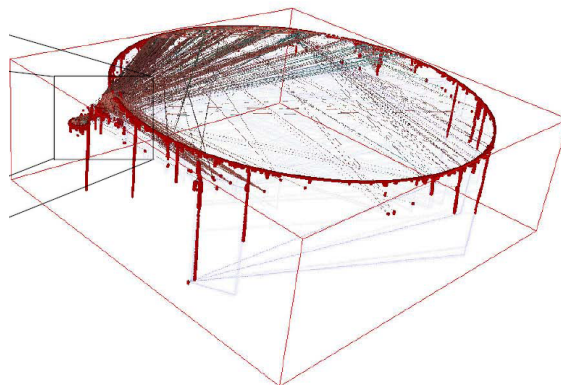


Figure 10 Usage sessions with site usage data [source: [Youssefi2003], Figure 2, page 3]

VisVIP [Cugini1999] is a graphical tool that superimposes the paths followed by the users over the structural website visualization (Figure 11). The 3D space was chosen for semi-clustered representations of the website structure and its interconnections, using balanced trees. It implements suitable exploration mechanisms to simplify highly

connected websites by suppressing all the edges leading to a selectable node or by simplifying the graph to a tree, whose root is represented by an arbitrary selectable node.

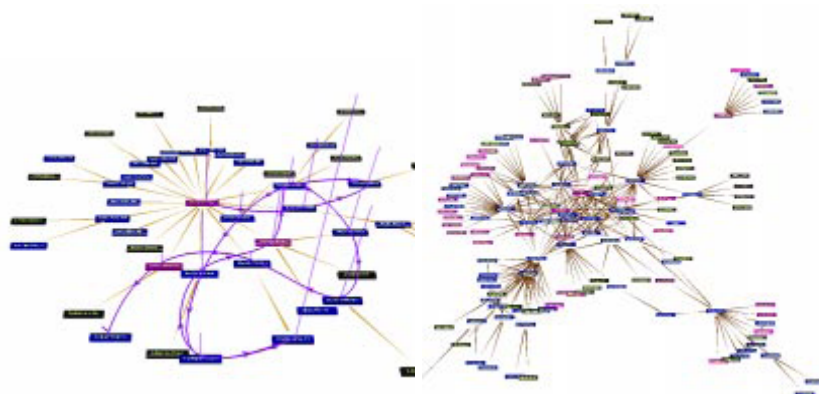


Figure 11 *VisVIP* - Path laid (left) over a website structure (right) [source: [Cugini1999], Figure 1, Figure 2, page 3]

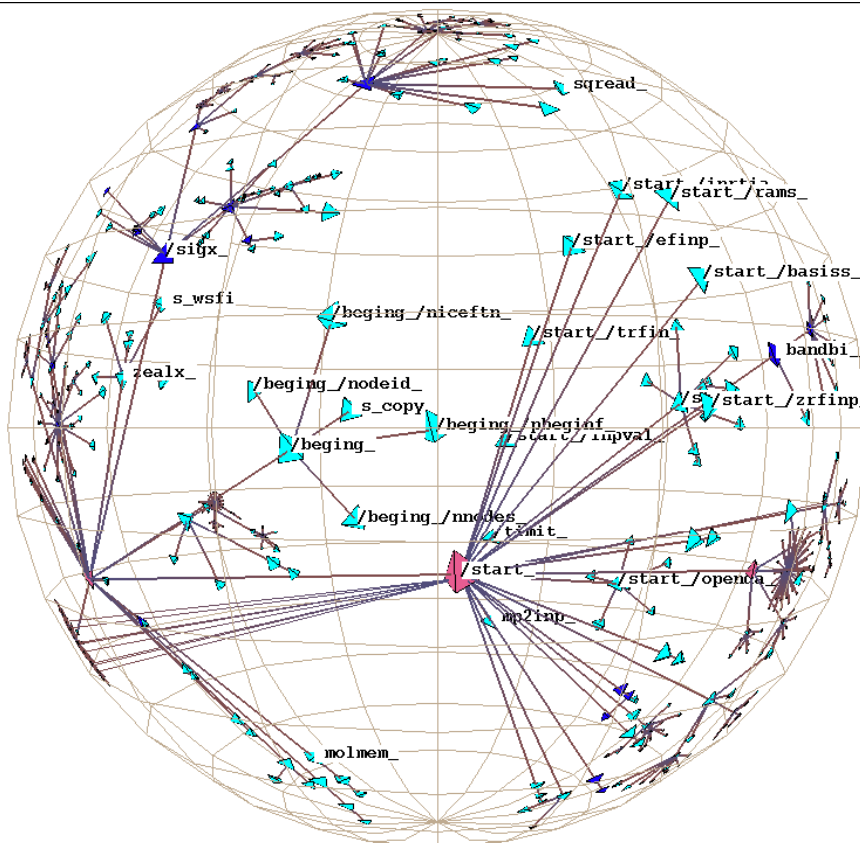


Figure 12 The H3 3d hyperbolic browser [source: [Munzner1997], Plate 2, page 9]

[Munzner1997] introduced the H3 algorithm for representing large informational structures in 3D hyperbolic space (Figure 12). The cone tree layout was optimized for the 3D hyperbolic space by placing children on a hemisphere around the cone extremities instead of on its perimeter. The volume of hyperbolic 3D-space increases exponentially,

as opposed to the familiar geometric increase of Euclidean 3D-space. This type of visualization is suitable for large hypermedia structures, clustering being one of the valuable aspects.

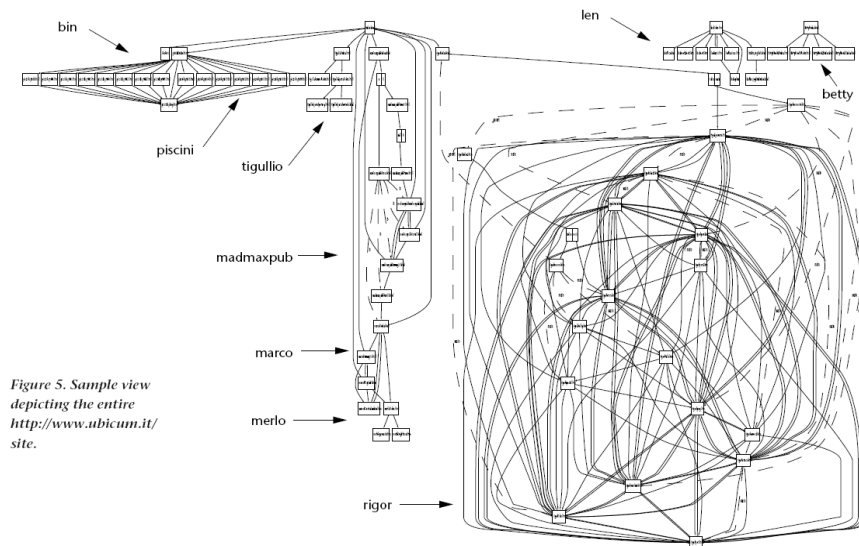


Figure 5. Sample view depicting the entire <http://www.ubicum.it/> site.

Figure 13 Filippo Ricca and Paolo Tonella's hierarchical visualization of the site [source: [Ricca2001], Plate 5, page 46]

[Ricca2001] used a clustered hierarchical visualization for the website with the purpose of analyzing its usage patterns. Each potential important root area is represented as a root of a new cluster (Figure 13). The *ReWeb* tool was designed for the analysis of the structure and history of the websites. It uses different versions of the website and uses graphs to visualize the differences.

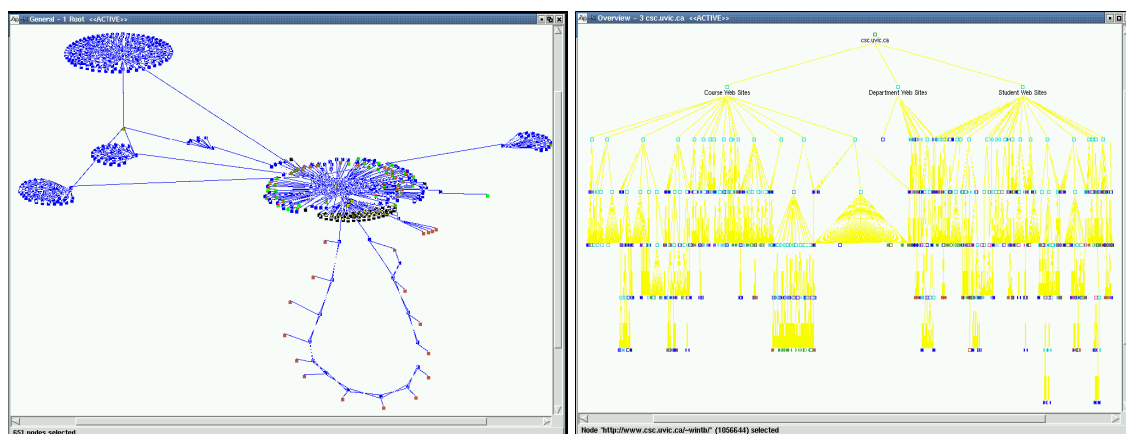


Figure 14 Directed Graph (left) and Hierarchical (right) visualizations [source: [Martin2001], left - Figure 2, page 4, right- Figure 6, page 6]

[Martin2001] used directed graph and hierarchical visualization with the purpose of classification and analysis of the web contents, using reverse engineering tools. Directed graphs use clustering for representing the semantics (Figure 14 – left quadrant).

2.1.2. Semantic content and information clustering

The process of extracting the semantic information related to the websites is a complex topic, several research areas being applied. This process is also referred as web mining. The information that characterizes a website, as well as the semantic classification of information structures can be mapped on the representation of the site; the results are sets of information clusters, used to identify the semantic content of the website. The first steps in this process are to apply filtering and cleaning processes, to ensure the validity of the subsequent processes. Then, several techniques can be applied for mining purposes, discovering navigational behavior or usage patterns being the final point of interest in our case.

The *WebKIV - Web Knowledge and Information Visualization* tool [Niu2003] previously described uses several web mining techniques to collect the structure and usage information of the websites. Its authors proposed three levels of functionality for the classification of information:

1. *“Web structure visualization.* *WebKIV* provides tools for visualizing small and large web structures, with controls that support the display of both detailed and abstract structure;
2. *Web navigation visualization.* *WebKIV* provides static and dynamic display of both individual and aggregate user navigation patterns;
3. *Web mining results comparison.* *WebKIV* provides a way of overlaying web navigation patterns, and comparing those constructed from the application of machine learning to navigation improvement.”

However, *WebKIV* lacks the functionality to operate on the graphs, presents a static graph, and uses the same web topology-rendering algorithm as Ed Chi’s *Disk-Tree*.

In addition to the content of every web page, the frames inside the pages can make a difference while classifying modern websites; “due to the presence of frames, the number of potential page views for a website can be vast” [Cooley2003]. This author proposed the following taxonomy for representing web pages:

“The structure of a website needs to be stored as a set of frames, \mathcal{F} , with a list of associated links, \mathcal{L} , and targets. A target is the page area that the link should be loaded into in the browser display. A single link can lead to the replacement of between one and all of the frames in a page view. A formal definition of a site structure map can be as follows, where \mathcal{M} is a site map, h_i is an HTML file, r is a link type, and g_j is a target area.

Web Usage Mining preprocessing consists of converting the usage, content, and structure information contained in the various available data sources into the data abstractions necessary for pattern discovery.” Figure 15 a visual summary description of the process of web usage mining.

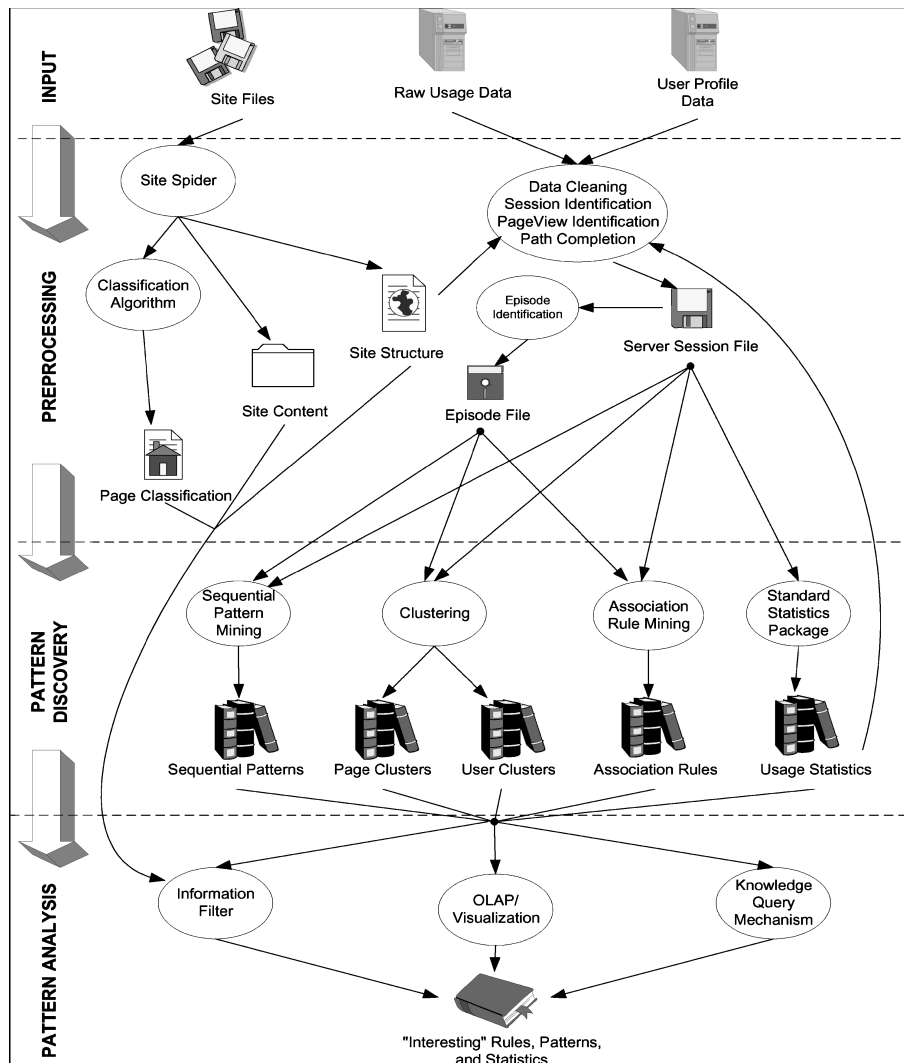


Figure 15 Robert Cooley's – Web Usage Mining process [source: [Cooley2003], Figure 3, page 102]

[Ricca2001] classified the websites in four levels of abstraction according to a taxonomy characterized by dynamism, page decomposition, and data flow:

1. Level 0: static pages without frames
2. Level 1: static pages with frames
3. Level 2: dynamic pages without data transfer from client
4. Level 3: dynamic pages with data transfer from client

The tool proposed by Rica [Ricca2001], *ReWeb*, is able to analyze and classify the content found on the first two levels of abstraction. It is meant to analyze the structure and history of websites and uses graphs to visualize the differences (page and link additions, modifications, and deletions) between versions.

[DiLucca2002] introduced WARE: a tool that performs static analyses on web applications, stores the extracted information into a database and then uses such an information within a reverse engineering process to construct UML diagrams that semantically describe the website. It uses a special representation of the web application called Intermediate Representation Form (IRF), implemented with a set of tagged files, one for each source file of the application. An example of a possible output is presented in Figure 16.

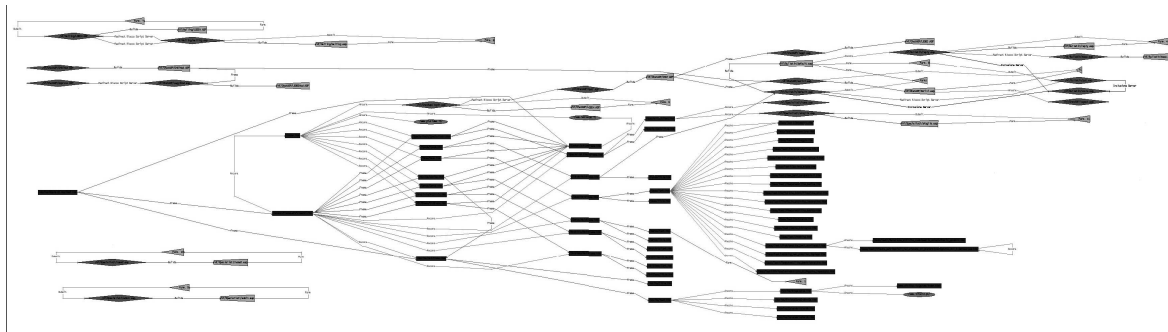


Figure 16 WARE: class diagram of the web application '*Juridical Laboratory*' [source: [DiLucca2002], Figure 5.1, page 8]

[Nomura2002] proposed a visual visualization of the HITS (Hyperlink-Induced Topic Search) web search algorithm used for discovering topic-bound web communities. The proposed visualization tool, *LinkView*, uses the meta-information of hyperlinks, as a subset of the page content, to represent connection between semantically related web contents. The tool was developed to study the cases of large and densely linked set of unrelated web pages that proved the HITS algorithm does not successfully apply. Figure 17 reveals two cases of usage, one for discovering the "Artificial Intelligence" topic, the other for the "Harvard" topic. While the first case was a successful one, the second failed due to highly linked sets of unrelated pages that contained the topic.

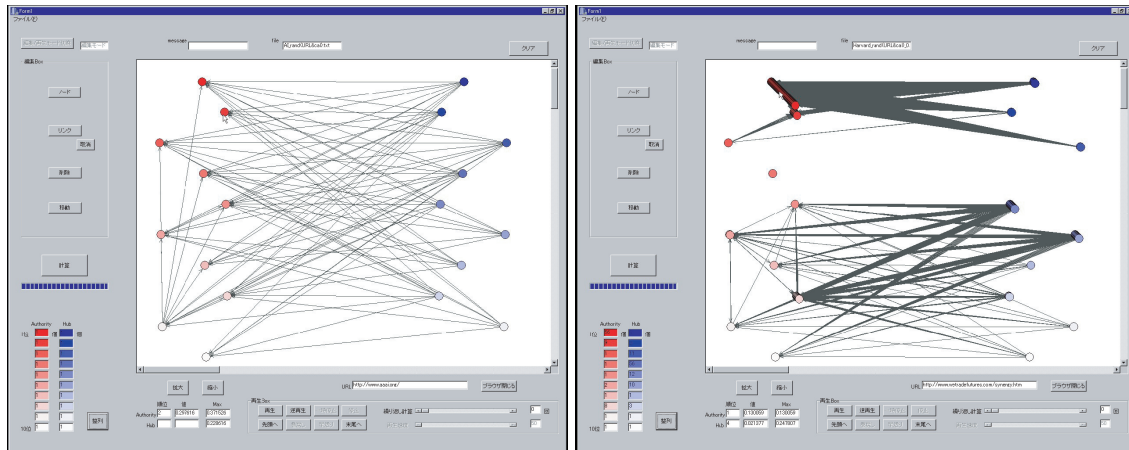


Figure 17 UI of *LinkViewer*: Successful vs. unsuccessful examples of web page analysis [source: [Nomura2002], left - Figure 2, page 3, right - Figure 3, page 4]

2.2. Usage Information

We classified the usage information retrieval processes in three phases: *Gathering* the required information; *Filtering* the information and *Classifying* or *Defining* the information.

2.2.1. Gathering

Log file analysis methods automate analysis of data captured during formal or informal website usage scenarios. Since web servers automatically log client requests, log file analysis is a heavily used methodology for evaluating web interfaces [Ivory2002].

[Cooley2003] proposed three definitions for the data involved in web usage information retrieval (mining):

1. *Content*. The real data in the web pages, that is, the data the web page was designed to convey to the users. This usually consists of, but is not limited to, text and graphics;
2. *Structure*. Data that describes the organization of the content. Intra-page structure information includes the arrangement of various HTML or XML tags within a given page. The principal kind of inter-page structure information is hyperlinks connecting one page to another;
3. *Usage*. Data that describes the pattern of usage of web pages, such as IP addresses, page references, and the date and time of accesses. Typically, the usage data comes from an Extended Common Log Format (ECLF) Server log”.

Another way of collecting usage information is to intercept the events generated by website users during controlled experiments. [Card2001] proposal is based on the combination that uses browser logging tools and eye trackers. The log files produced during controlled experiments with users are then analyzed, using a specific tool -

WebLogger, the results being visually represented and used to understand the users' behavior.

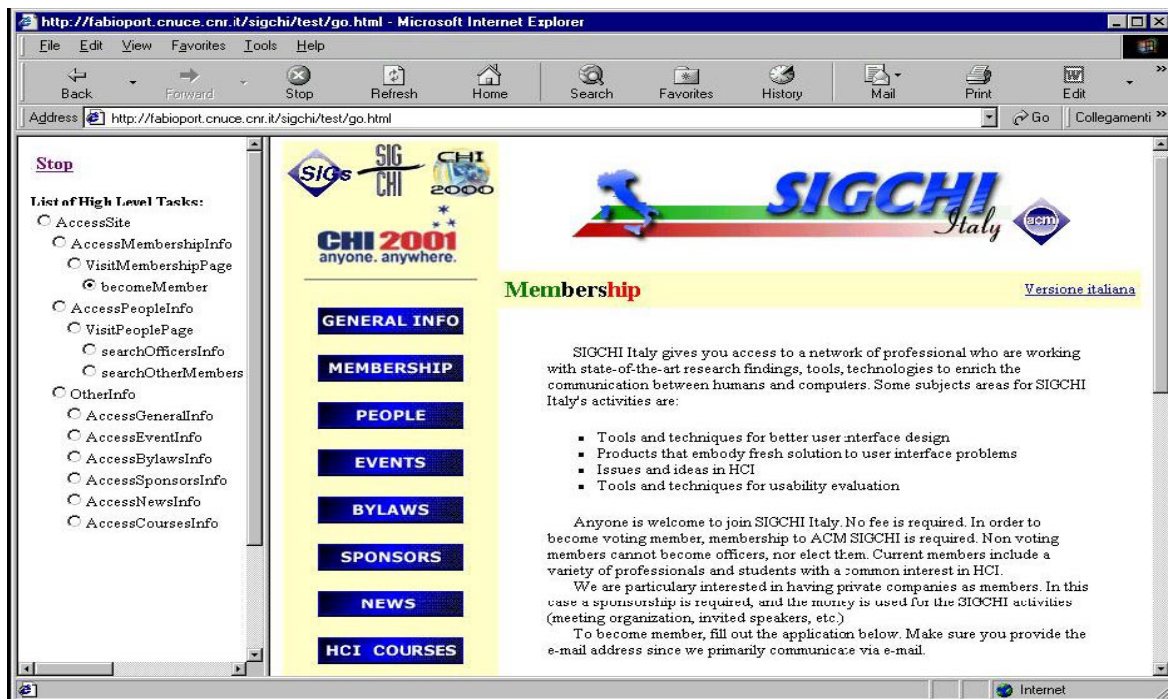


Figure 18 Browser extension for event logging used by *WebRemUSINE* system [source: [Paganelli2002], Figure 1, page 4]

[Paganelli2002] presented a set of tools able to perform intelligent analysis of web browser logs collected from both sources: application logs and the information contained in the task model of the application (obtained during the preparation and evolution phases of live experiments) (Figure 18). The log analysis tool, *WebRemUSINE*, combines both the information stored on the web server logs with the information obtained from informal experiments over the website usage to obtain insights.

[Heer2002] used as primary source of data the access logs stored on the application server. The authors focused on applying several mining techniques on the pre-processed web logs, in order to discover navigational patterns and present them using visualization techniques.

[Fraternali2003] focused on the analysis of both type of log files: *application server logs* and *runtime logs*. The former, the access logs from the application server registered in ECLF (Extended Common Log Format), the latter, the log files generated by the design framework that uses *WebML* language for the design of the website [Ceri2000]. The analysis process can be performed in two major phases:

1. “*Design Schema Analysis* verifies the correctness and consistency of design specifications, to enhance the quality of conceptual schemas by looking for design inconsistencies and irregularities in the application of design patterns. This phase

focuses only on a static description of the application and does not take into account dynamic usage aspects; and

2. *Web Usage Analysis* operates on log data dynamically collected at runtime and produces quality reports on content access and navigation sequences. This analysis exploits the so-called conceptual-level log files, defined as "enriched" web logs that integrate the conventional HTTP data about page requests with information about the database objects used to populate pages and about the elementary page units and link paths accessed by the users".

2.2.2. Filtering

[Spiliopoulou2000] proposed an interesting algorithm for website usage mining and pattern discovery, based on the identification of web usage sessions. The first step in web usage mining is always to filter and clean the data. The most significant problems in this step usually involve the extraction of web usage sessions.

[Youssefi2003] introduced 3D visualization techniques for representing the results of Web mining over the website structure and access logs. The results of two algorithms were used to produce the data used for visualization: *Sequence Mining* [Zaki2001] and *Tree Mining* [Zaki2003]. The former (site graphs) are semi-static snapshots of the website structure; the latter (access logs) capture the dynamic behavior of surfers visiting a website. On both cases, the process of mining starts with a set of filter applied to raw data.

[Heer2002] focused on visualizing data from web access logs and visualizing patterns derived from web mining processes, providing means to interactively play with visualization objects in order to perform ad hoc visual data mining and interpretation of the discoveries. The navigational dataset represents statistical information extracted from pre-processed web logs, navigational patterns being discovered with web mining techniques while the context of these is simply the structure of the website.

2.2.3. Classification

[Cooley2003] classified the information provided by the data sources of analysis with the purpose of constructing a data model consisting of several data abstractions: *users*, *page views* (all of the files that contribute to the client-side presentation seen as the result of a single mouse "click" of a user), *click-streams* (the sequence of page views that are accessed by a user), and *server sessions* (the click-stream for a single visit of a user to a website). The author proposed the Web Usage Mining (Figure 15) preprocessing that consists of converting the usage, content, and structure information contained in the various available data sources into the data abstractions necessary for pattern discovery. After this process, several steps are required for classifying the information:

- i. Data cleaning;

- ii. Session identification
- iii. Page view identification;
- iv. Path completion; and
- v. Episode identification.

In [Youssefi2003] a specific set of questions was highlighted to link directly the raw information available for analysis and classification with the high-level abstraction of website-user behavior analysis:

1. “What is the typical behavior of a user entering our website?”
2. What is the typical behavior of a user entering our website in page A from ‘Discounted Book Sales’ link on a referrer web page B of another website?
3. What is the typical behavior of a logged in registered user from Europe entering page C from link named “Add Gift Certificate” on page A?
4. What is the typical behavior of a user who comes in our website 1 day to 3 weeks before Christmas and buys something, versus one who didn’t buy anything”?

In order to support these questions, the authors proposed a specific set of data mining techniques, for extracting new insights and measures, and several visualization techniques to obtain an overall picture correlating static site structure with (dynamic) access patterns. They defined the notion of a *user session* as a temporally compact sequence of web accesses by a user. Partially, the goal of their web mining techniques is to work on these sessions for better visualization toward useful information.

[Fraternali2003] are using the *WebML* design method that proposes a sequence of steps for assembling the *data schema* and the *hypertext schema*, which assume that web applications can be abstracted as complex arrangements of elementary *sub-schemas*, i.e., pairs of *structural diagrams*, and *hypertext diagrams* (describing composition of pages and navigational patterns). This approach allows the authors classify the content of a website during the design phase, having as results a better classification of the structural, semantic and content of the website. The whole process proposed can be considered a dynamic workflow, starting from the design phase of each page of a website and ending with the interpretation of usage patterns discovered from the analysis of usage logs.

[Heer2002] represented each session as a vector that describes the session's sequence of transactions. For example, in a space (A,B,C,D,E) which corresponds to a website that consists of 5 pages labeled A through E, a session consisting of page views A→B→D could obtain a vector (1,1,0,1,0). Then, a number of possibilities for assigning vector values were explored, e.g. viewing time of each page can be used to assign vector values: (10, 25, 0, 15, 0), this case, in seconds.

2.3. Visual correlations of structure, contents and usage

Although there are many analysis and visualization techniques (good examples are included in [Card1999], [Bederson2003]), and many systems that use them to visualize large amounts of information, there are comparatively few studies on the evaluation of those techniques and systems more adequate for the analysis and visualization of websites. This is perhaps due to the inherent complexity of this evaluation. A good example was introduced by Ivory [Ivory2002] as a State of the Art in Automated Usability Evaluation of User Interfaces.

While there is not yet a body of knowledge on information visualization evaluation, we can find in literature some works explicitly using evaluation in the process of designing a visualization system, evaluating specific systems and visualization techniques, as well as comparing alternative visualizations. Examples can be found in [Hix1999], [North2000], [Sebrechts1999], [Wiss1998], [Kobsa2001], and [Barlow2001]. Moreover, there are also some authors recognizing the importance and making an effort to develop more systematic approaches to the complex problem of evaluation in information visualization [Freitas2002], [Brath1999], and [Grinstein2002]. No matter how interesting these works may be (and we believe that they indeed are), those who want to evaluate visualization techniques and systems, still struggle with a lack of specific techniques and methodologies to conduct the evaluation. Currently, it seems that a reasonable approach could be to adapt the well-known and already widely used methods of Usability Engineering [Nielsen1993], by taking into account the specificities of the techniques and systems that one is trying to evaluate. However, a wealth of techniques has been developed, and is applied to solve real problems, it is important to know if these analysis and visualization techniques actually work for websites. Furthermore, there is a need to know under what circumstances they should be used, how they compare and what tasks they best serve.

In [Eick2004], Stephen G. Eick focused on three areas in which information visualization makes a significant contribution to understanding online visitor activity:

1. *“Visualizing site structure as a visitor navigation aid;*
2. *Showing paths and flow through the site to help designers build a more effective site; and*
3. *Monitoring the site’s real-time activity to help site operators run their businesses more efficiently”.*

In our case, three major areas are of a significant importance for the study:

- a. *Visualizing the Structure* of the website and all interconnections of website pages;
- b. *Visualizing the Content* of the website as web pages content and/or information clusters (semantic classification); and

- c. *Visualizing the Usage* of the website using quantitative and qualitative measures for the user success / insuccess;

Several website visualizations have been developed, most of them related to the visualization of the website structure, some based on its semantic content classification.

2.3.1. Visualizing structure

[Chi1998], [Chi2002], [Niu2003] and [Chen2004] combined a radial visualization of the website structure, the root of the website being the center of the representation, the pages being represented as nodes, and the links as edges, the so called *Disk-Tree* visualization presented in Figure 5, Figure 6, Figure 8 and Figure 9. [Chi1998], [Chi2002] also presented the *Time-Tube* that, basically, is a 3D representation of the *Disk-Tree*, multiple disk-trees being represented on the time axis, to highlight the differences between different versions of the website (Figure 6).

[Niu2003] added more dynamics to website visualization by combining context exploration. Using the same radial representation, [Chen2004] presents a visualization framework that combines a multi-tier taxonomy that divides the visualization space in several layers used to represent different types of information. The base of the visualization layers is the website structure, every information layer being superimposed on this one.

VisVIP is a graphical tool proposed by [Cugini1999] that visualizes the paths followed by the users during the site visit. It implements one way to show paths for an individual visitor and superimposes a trace of the pages visited on top of a site map, using this page-tracing technique for path and timing data (Figure 11).

[Munzner1997] uses a hyperbolic visualization of the website structure, the information clusters being visualized in 3D spherical representation (Figure 12).

In [Youssefi2003], the authors combined various 2D and 3D visualization techniques for the representation of the website structure, enriched with usage information resulted of data mining techniques being applied over the application server's log files. The fashionable 3D visualizations proposed lack in expressiveness and interaction, occlusion being one of the major problems for the 3D space.

Even if there are several visualization suitable for medium to large websites, nowadays websites tend to be dynamically generated, up-scaled and personalized; without the paradigm of a static organization of contents, it difficult to represent and analyze a constantly "morphing" structure, means of interaction being adequate and necessary for filtering these large representations of semantically related content.

2.3.2. Visualizing content

The constant evolution and sophistication of website structures and the need to understand them, determined worldwide usability experts focus more and more on the semantic relations of website contents:

- How is the content of the site influencing the user's behavior?
- What kind of direct mappings do exist between the semantic content and usage patterns?
- Which of the clusters of semantically related contents have more visitors and why?

The answers to all these questions and many others are challenges for the development of automated usability analysis tools that might use visual representations to give insights of the relation between the semantic content and visitor's behavior. Page clutter, structural organization, visual pattern, navigational paradigm and many other factors have a deep impact on the analysis of usage behavior.

In [Card2001], the authors introduced a combined analysis technique of the website content and its influence on the website users' behavior. They proposed a technique of auditing the events generated by the users while exploring website contents, during empirical experiments. Several measures were taken into account: keyboard and mouse events, visual impact (eye-tracking devices were used). Figure 19 presents a portion of the results of the analysis of a simple session meant to search for a specific content on a specific website.

According to the authors, the information space is organized in patches:

- i. Web;*
- ii. Website;*
- iii. Page.*

A *Page* might contain *Link descriptors* (textual or image-based), *Content elements* and other elements. For the specific tasks of information discovery and retrieval, search for content elements can proceed by a search through spaces composed of sets defined by: *Link descriptors, URL and Keywords.*

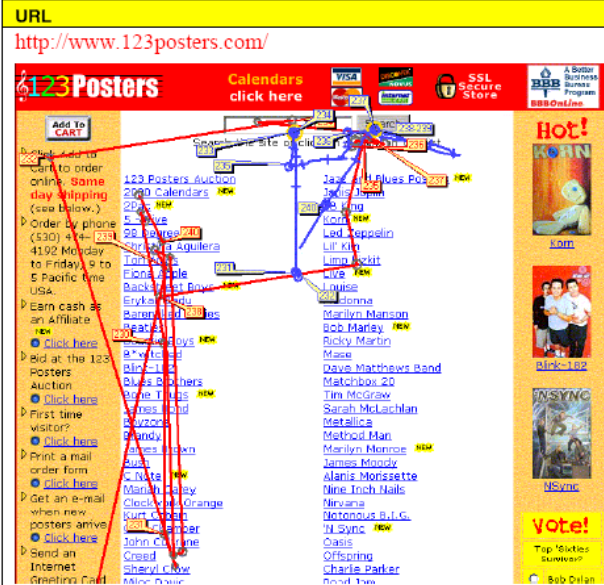
URL	Observed Actions & Transcript	Model Interpretation
	<p>Let's see what they've got for me. Woah! Search for Antz..</p> <p>230.191 (SCROLL) 230.842 (MOVE-MOUSE) 232.555 (MOUSE-CLICK Search-Box) 233.777 (TYPE: antz) 234.638 (MOVE-MOUSE) 236.891 (MOUSE-CLICK Search-Button)</p> <p>Mmm, what are we gonna get?</p>	<p>(O*SEARCH SITE 123posters.com null "Antz")</p>

Figure 19 Page content exploration and event interception [source: [Card2001], Figure 2, page 501]

Starting from these definitions and some others, the authors then proposed the results of the analysis of the logged information as a *Web Behavior Graph (WBG)*, meant to help a usability specialist visualize the behavior of the users while performing specific tasks. Figure 20 presents an example of two specific tasks of searching the web for *Antz* and *City* movie posters and the associated legend of color coding mappings.

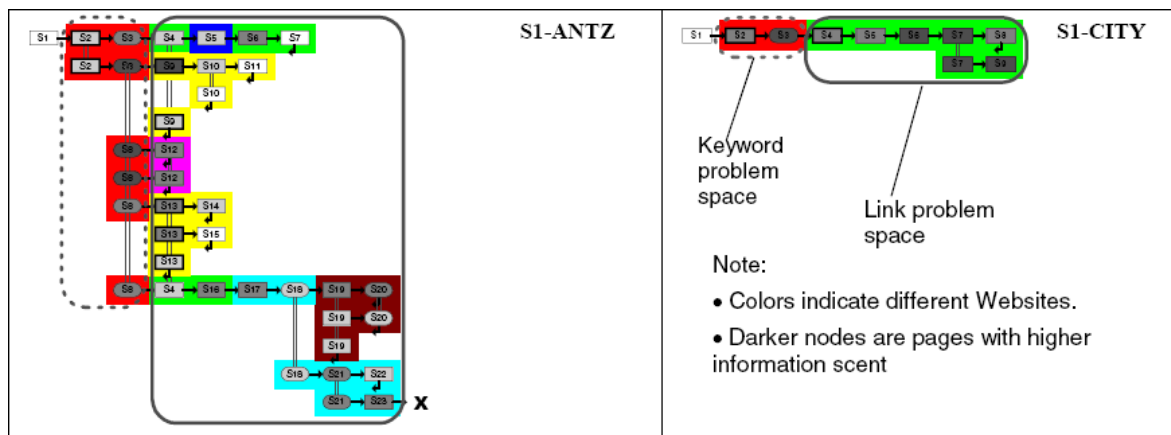


Figure 20 Example of WBG for specific search tasks [source: [Card2001], Figure 4, page 503]

[Faraday2000] proposed the *Design Advisor* tool that enables visual analysis of web pages. The tool is based on guidelines derived from the use of eye-tracking techniques that identify which interface elements attract user attention (animations, images, colors, etc.) by identifying the scanning path on the web page. The author suggests that visual processing of web pages forms a distinct visual hierarchy in which certain perceptual elements have priority; the process of reading a web page is divided into two phases : the

first phase is termed 'search', the second is 'scanning'. The former is used to find an entry point of the page, the latter to scan and extract information. The following guidelines have been identified for the search phase and are ordered as follows: *Motion, Size, Images, Color, Text Style and Position*. This represents the order of a page being scanned by subjects. The scanning phase starts after finding an entry point and is influenced by *Area and Proximity & Reading Order*.



Figure 21 Design Advisor's visual clues [source: [Faraday2000], Figure 6]

Based on these guidelines, the tool is able to overlay the scanning paths over the page content, resulting from the processing of page contents. Visual clues are then used by usability experts to critique the page (Figure 21).

In [Ivory2002] the authors proposed a tool based on empirically derived measures computed over thousands of web pages. Then, these measures, which characterize the informational, navigational, and graphical aspects of a website, were converted into profiles for a variety of site types. The tool is able to compare features of the submitted design to features of highly rated sites and tends to signalize a set of rules as predictions, similarities, differences and suggestions (Figure 22). Even if there is no direct visual mapping for these rules, the authors promise an evolution able to suggest a better design with direct linkage to the good designs used for the analysis.

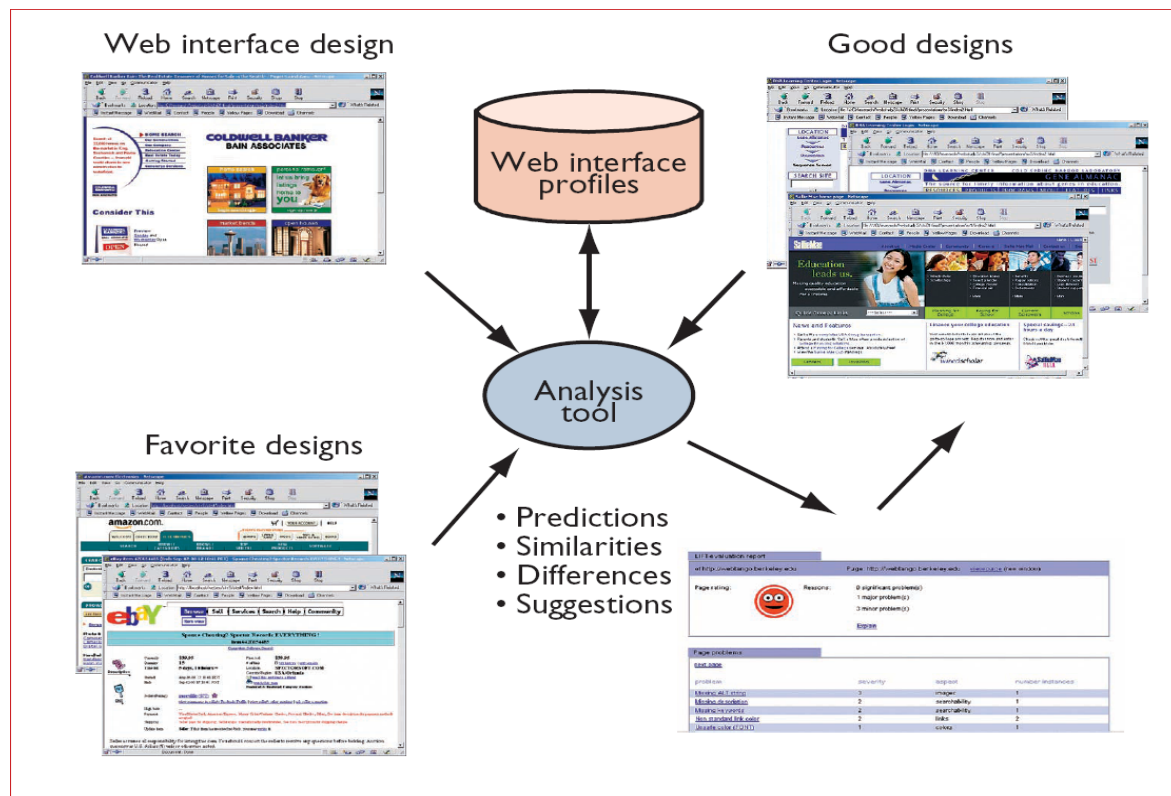


Figure 22 WebTango UI analysis tool taxonomy [source: [Ivory2002], Figure 3, page 62]

2.3.3. Visualizing usage

The usage information of a website is retrieved from the application server logs (obtained during natural website usage) and logs produced during controlled experiments. The challenge is to represent the quantitative and qualitative measures of website usage, combined with other aspects for a direct visual mapping of these measures on the website structure. As an example, the user's satisfaction level is an important qualitative measure, subjective and difficult to collect under regular circumstances of website usage.

A. Quantitative

[Chi2002] proposed a technique of mapping the quantitative usage information using color and thickness to highlight usage patterns on a radial representation of the website structure (Figure 5). In addition, the *Time-Tube* visualization [Chi1998] was designed to cope with the sophisticated evolution of the website structure and its users behavior, several versions of the website structure and navigational patterns being compared within the division of time (Figure 6).

In addition to the static structures proposed by [Chi2002], [Niu2003] introduced a type of animation for the website usage, the tool *WebKIV* being able to replay the natural usage using animated transitions of the representation (Figure 23). The visualization is able to

represent quantitative measures of website usage by using color-coding techniques, line thickness and animation.

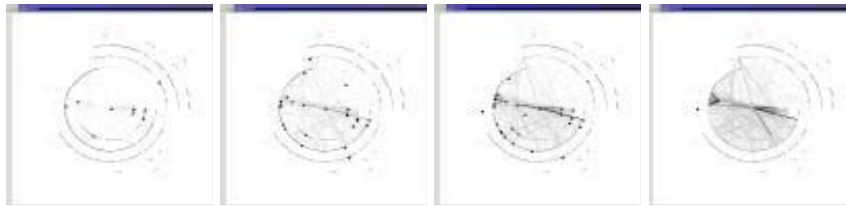


Figure 23 Example of surfing animation (sample slides) [source: [Niu2003], Figure 6, page 5]

The mapping techniques of usage information on the website structure introduced by [Chen2004] might be considered one of the most valuable techniques of representing quantitative usage information with the purpose of extracting qualitative measures of website usage and navigational patterns. Color and thickness are used on separate layers to code different quantitative and qualitative measures as *Number of Visits*, *Link Usage*, *View Time* (Figure 9– left quadrant) and *Probability Usage of a Link* (Figure 9 – middle).

[Paganelli2002] presents a tool able to perform intelligent analysis of web browser logs collected from both sources: application logs and the information contained in the task model of the application (obtained during the preparation and evolution phases of live experiments). It uses an empirical evaluation method for websites, involving the users of the website in an experiment meant to measure the performance of their interactions while executing a specific set of tasks. This tool, *WebRemUSINE*, is capable of analyzing the events that occur on the monitored browser and use these events for the discovering of the website usage patterns via its real user interactions. The Figure 24 presents a quantitative measure of usage (page visit timings).

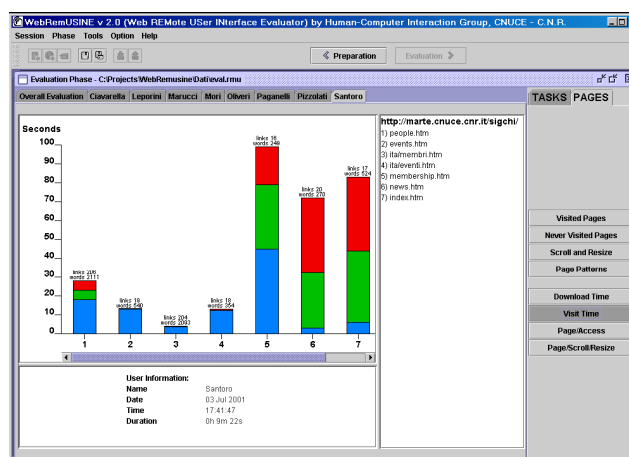


Figure 24 *WebRemUSINE*: display of page visit timings [source: [Paganelli2002], Figure 5, page 117]

In [Youssefi2003] the tool presented in and mentioned in section 2.1.1, the results of usage sessions and/or different usage patterns are presented using color coding, thickness, special disposition, etc (Figure 10 and Figure 25). Given that 3D space is used for the visualizations, occlusion might be a serious issue for the observer. In addition, the complexity of the visualizations and a low or no identification means of the visually represented information might prevent appropriate perception of the same.

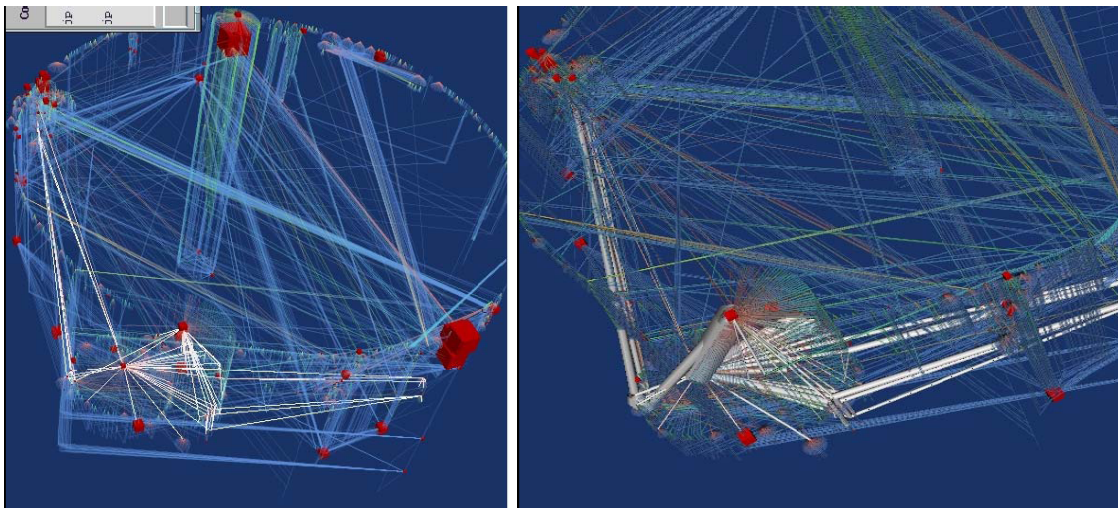


Figure 25 Frequent access patterns (white graph, left image) [source: [Youssefi2003], Figure 6, page 6] and the addition of frequency of pattern attribute rendered as thickness (right image) [source: [Youssefi2003], Figure 7, page 6]

B. Qualitative

In [Chen2004], a visualization framework able to apply web-mining techniques on the data obtained from the analysis of website access log files is presented. Using a multi-tier taxonomy, it divides the visualization space in several layers used to represent different types of information. In addition to visualizing data from web access logs and visualizing patterns derived from web mining processes, the main goal steering the design of the system is to provide means to interactively manipulate visualization objects in order to perform ad hoc visual interpretations of the same. The proposed web graph algebra combines different information layers to obtain visual insight about the web structure, usage data and whatever usage patterns can be discovered during the mining process. Figure 9 presents some of the qualitative measures deduced from the superimposed information layers, each presenting a type of quantitative information.

The *WebRemUSINE* tool [Paganelli2002] mentioned in subsection 2.3.3 is able to produce qualitative evaluations of the user-performed tasks, presenting a qualitative measure of the user groups per task performance, as shown in Figure 26.

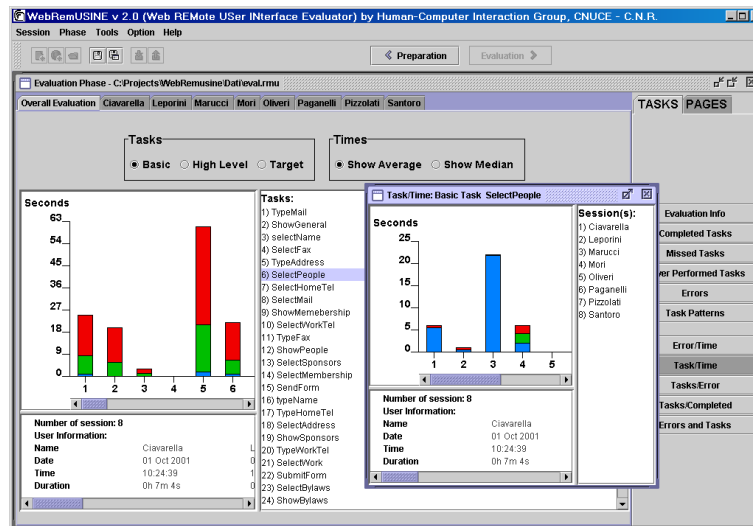


Figure 26 WebRemUSINE: display of a user groups task performance [source: [Paganelli2002], Figure 6, page 117]

The tool proposed by [Card2001] is based on the combined use of browser logging tools and eye trackers. After performing controlled experiments with sets of users, the *WebLogger* log files are analyzed and the results are represented as a *Web Behavior Graph* (Figure 27).

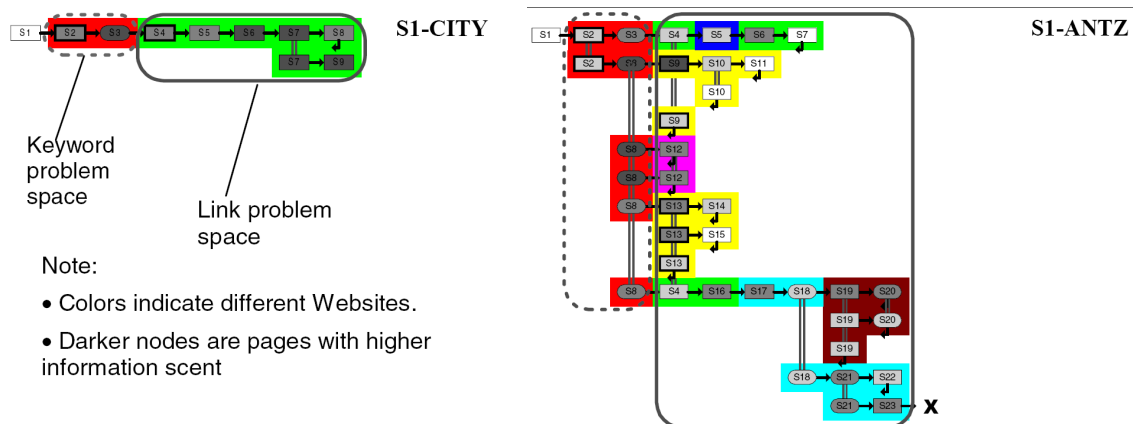


Figure 27 Web Behavior Graphs for users in the study. Solid enclosing lines indicate that the states and operators are part of the Link problem space. Dotted lines enclose the keyword search problem space. Square dotted lines enclose the direct URL typing problem space [source: [Card2001], Figure 4, page 503]

2.4. Dynamic information exploration

As website users require fairly well designed web pages to complete their tasks, usability experts require well designed instruments to help them analyze possible design malfunctions that might affect the popularity and accessibility of websites. A good

visualization method is useless without the proper interaction mechanisms that allow its users better interact and observe the represented information. In the following paragraphs, the dynamic information exploration support will be highlighted for part of the previously presented visualization techniques as well as some others.

In [Chi1998], [Chi2002] and [Chi1999] the authors proposed three types of representations for the website structure, evolution and usage patterns discovery. The three dimensional representations have the advantage of being capable of representing large amounts of information, but the occlusion problem might affect the perception of the represented information. According to [Chi1999], these visualizations offer the following interactions:

- i.* Time-Tube: Focus on a slice; Bring slices back into the Time-Tube; Zooming focus on the connectivity of a node by right-clicking on it; Rotate slices; Brushing on pages by highlight URL on all slices; Animate through the slices;
- ii.* Spreadsheet with Time-Tubes or Disk-Trees: Dynamic view-filter; Change object position and orientation; Pixel image addition between cells; Geometric object addition between cells; Animation; Coordinated direct manipulation; Apply geometric operators; Detail on-demand Zoom;
- iii.* Disk-Tree and other tree types: Focus node; Hide sub-tree; Change orientation and position of tree; Apply Dynamic level filtering.

[Niu2003] introduced new representation techniques meant to facilitate the analysis of the presented information: *Web Surfing Animation and Usage Aggregation*, *Web Mining Evaluation*, *Subtree Zoom-in*, etc.

In [Chen2004] the *Web Image* representation allows dynamic layout of the radial tree and time-based filtering of the displayed information.

In [Youssefi2003] fashionable visualization techniques are used, implemented using open source library called the Visualization ToolKit (VTK) [Schroeder1998] that allows basic manipulations of the 3D representations. It turned out that the final proposed tool provides poor interaction mechanisms and feedback, with large sets of information due to the lack of information identification mechanisms.

The *VisVIP* tool [Cugini1999] allows dynamic layout and simplification of the representation by remapping to another root, multiple selections and manual dynamic suppression of nodes.

The Hyperbolic browser (Figure 12) [Munzner1997], in addition to the basic manipulations of nodes as dynamic information clusters, offers new visually interactive techniques for selection of nodes, providing transitions based on animation and optimized re-clustering of the out-of-focus nodes.

2.5. Automated reporting and suggestion of usability improvements

In [Ivory2002] the authors used website predefined design profiles to measure the compliance of analyzed websites to the “good” designs techniques, as described in section 2.3.2 – Visualizing content. According to these authors (Figure 28), “a website interface is a complex mix of text, links, graphic elements, formatting, and other aspects that affect the site’s overall quality. Consequently, website design entails a broad set of activities to address the following diverse aspects:

- i. *Information design* focuses on identifying and grouping content items and developing category labels to reflect the site’s information structure;
- ii. *Navigation design* focuses on developing mechanisms (such as navigation bars and links) to facilitate interaction with the information structure;
- iii. *Graphic design* focuses on visual presentation;
- iv. *Experience design* encompasses all three of these categories, as well as properties that affect the overall user experience (download time, ads, popup windows, and so on).”

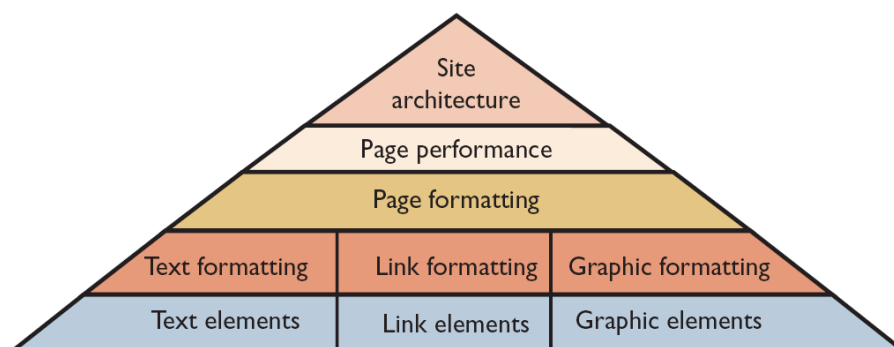


Figure 28 Ivory's website structure taxonomy [source: [Ivory2002], Figure 1, page 57]

The proposed design-checking approach is not intended to replace usability testing, but rather to complement it. The tool can be used to address potential design issues before conducting usability testing. The results may be helpful in identifying aspects to focus on during testing, such as text readability or whether page layouts facilitate information search.

The tool *WebRemUSINE* presented by [Paganelli2002] is able to highlight basic usability problems, allowing usability experts interpret the results of the analysis of experimental usage sessions to obtain insight.

[Fraternali2003] introduced a set of tools for the design and analysis of websites in two phases, as presented in sections 2.2.1 - Gathering and 2.2.3 - Classification. The design-

time conceptual schemas (structural diagrams and hypertext diagrams describing composition of pages and navigational patterns) of the application and the usage data collected at runtime are analyzed and the results are sets of rules and navigation patterns. Each quality attribute considered in the analysis is associated with a set of pattern descriptions and an evaluation method. The evaluation method comprises a metrics computation function, generating aggregated numerical values that quantify the level of satisfaction of the considered quality attribute, and a condition-checking rule, which highlights potential problems. The application then displays the pattern descriptors highlighting the paths between the different conceptual schemas involved in the pattern.

2.6. Applications

The survey presented by Andy Cockburn and Steve Jones in [Cockburn1997] approached ways to augment browser navigation using several visualization techniques; starting from this premise, the focus was on the analysis of several existing tools. Some of the tools analyzed were capable of displaying the navigation history, an important issue while analyzing website usage.

Even if many important analysis and visualization techniques have been proposed by the worldwide research laboratories in the latest years, only a few managed to reach the value of being integrated in existing applications. One of the reasons for this might be the tremendous complexity involved by such sophisticated algorithms and analysis techniques. Most of the current commercial solutions for website analysis are directing their efforts to quantitative measures of Log analysis and / or network traffic (bandwidth) related quantitative measures. Few of them address website content analysis and the correlation between its design, structure, content and usage patterns. Some of the more common information used by web log analyzers are related to web server log tags: date, time, client IP address, username, client URL request, client URL request parameters, request status, bytes sent, client user-agent, client cookies, and client referrer. Note that, some tags may be absent on some server or client configurations. This information is subjected to some statistical processing and the results are usually presented using tables and / or graphics [Nunes2003].

Given the previously presented facts, we decided to classify the current implementations of website analysis tools in three major areas of intervention:

- i. Log Analysis Tools – for the analysis of the log files stored on the application server;*
- ii. Website Analysis Tools – for the analysis of the structure, contents and semantics of the web site; and*
- iii. Usage Analysis Tools – for the analysis and discovery of usage patterns for the website, based on the analysis of web logs, website structure and contents.*

These three classes of commercial implementations are presented in the following sections. We focused on the description of the main features for each of the analyzed tools, as well as the provided visualizations and interaction mechanisms.

2.6.1. Basic logs analysis

Before exemplifying with real-estate applications, we should define the basis for what kind of information can be extracted from the application-server log files. Since most of the log files have a common standardized format, mostly, they store information about the requests that came to the application server during natural usage of the website. An example of a common log file format (CLF) [W3C2005] can be considered as including the following information:

```
{<Requesting host IP address or DNS name>,
<The remote logname of the user.>,
<User authenticated name>,
<Date and time of the request>,
<The request line exactly as it came from the client>,
<HTTP status code returned>,
<Bytes transferred>}
```

Starting from the premise that every application server logs a similar type of information, it is trivial to observe what kind of information can be extracted from such raw data:

- i. First, we would start with the measures of action success and unsuccess by measuring the *<HTTP status code returned>* values;
- ii. Next, we would proceed with applying clustering techniques on the *<Requesting host IP address or DNS name>*, *<The remote logname of the user.>*, *<User authenticated name>* fields to identify specific users or user groups, perhaps demographics if extra information on IPs is available;
- iii. Then we would segment the results in time using the values provided by the *<Date and time of the request>* field;
- iv. At last, we would do some calculus to obtain quantitative measures of bandwidth using the values represented in the *<Bytes transferred>* field;
- v. If the log file format permits, information like: *time taken for the operation, the referrer page of the request, the type of the referrer* (regular page, spider, search engine, etc.) *client browser type and version, operating system, client screen size, etc.*, might be of interest to be processed and interpreted.

After the first steps of raw data processing and filtering, some other kind of associations and aggregations would be of interests for the global study. We would:

- i.* associate the time with the bandwidth and obtain important information of which periods of day / week / month / year are most intensive;
- ii.* associate the referrer with the request to see where are the users coming and heading to;
- iii.* associate the client identification with the requests to obtain navigational behavior;
- iv.* associate the time taken with the requests to discover latency problems and low response times on the website;
- v.* associate the type of operating system or browser with the bandwidth to check how they handle the transfers;
- vi.* associate client identification with requests and timings to observe the time spent by clients on specific areas of the website, etc.

These rules are what we call *common statistical results* log analysis software might provide as output, regularly using some sort of formatted reports. The following paragraphs present some of current applications of web log analysis software tools.

AWStats [AWStats2005] - is a free log analyzer tool for advanced web, streaming, ftp or e-mail server statistics that presents the results graphically. The tool is able to analyze most common log file formats within any sizes (using a temporary file) and produce statistics based on the information found on these files. The visual output of the tool can be easily classified in tables and bar graph charts (Figure 29). An example of such outputs can be observed at <http://ns3744.ovh.net/awstats/awstats.pl?config=destailleur.fr>. Several categories of statistical information are available (Figure 29 – left quadrant), most of them presenting numerical values for quantitative measures of website usage. These analysis categories are organized in five major areas of interest: *When*, *Who*, *Navigation*, *Referrers* and *Others*. A full list of features, compared with other solutions, can be found at http://awstats.sourceforge.net/docs/awstats_compare.html.

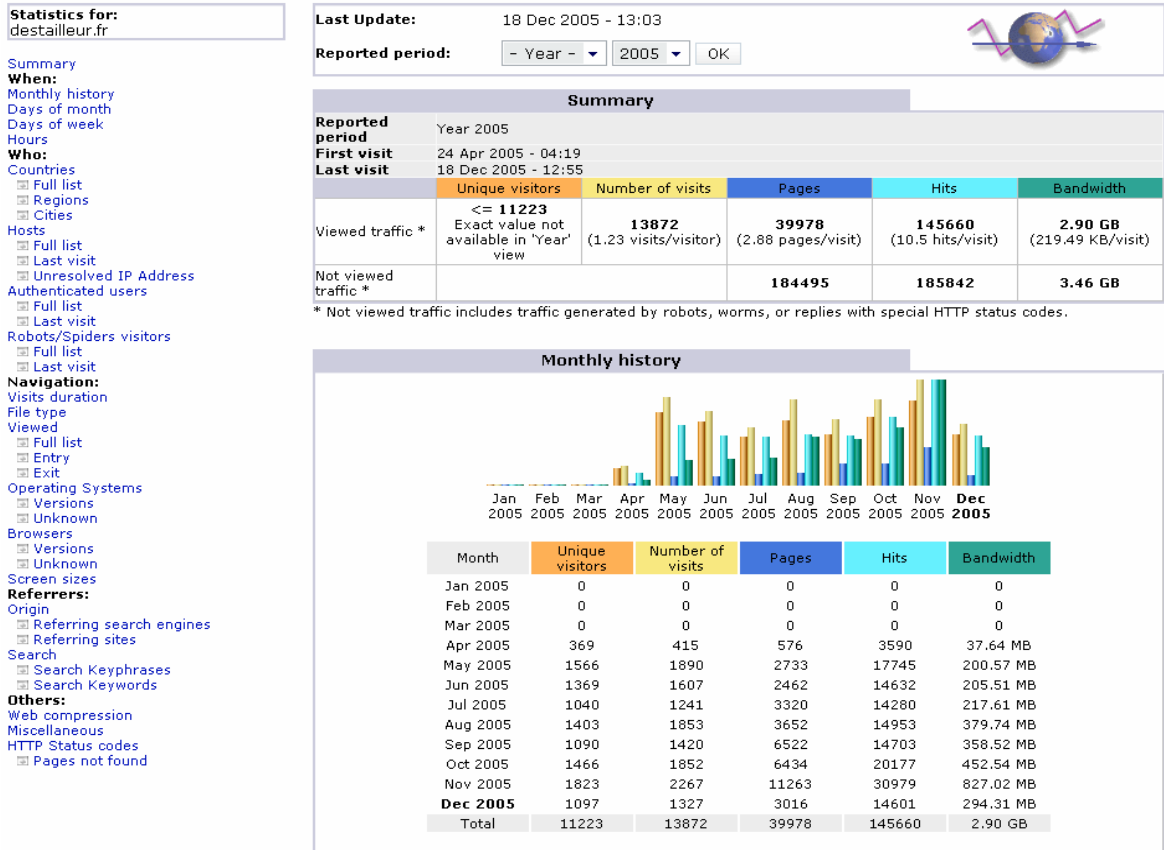


Figure 29 AWStats example of statistics graphical outputs [source: [AWStats2005], <http://awstats.sourceforge.net/>, Last visit: November 2005]

Wusage [Wusage2005] – is a commercial web log analysis software capable of measuring the popularity of the hosted documents, as well as identifying the sites that access the web server most often. The tool displays the results in a regular fashion, using tables, bar and pie charts (Figure 30).

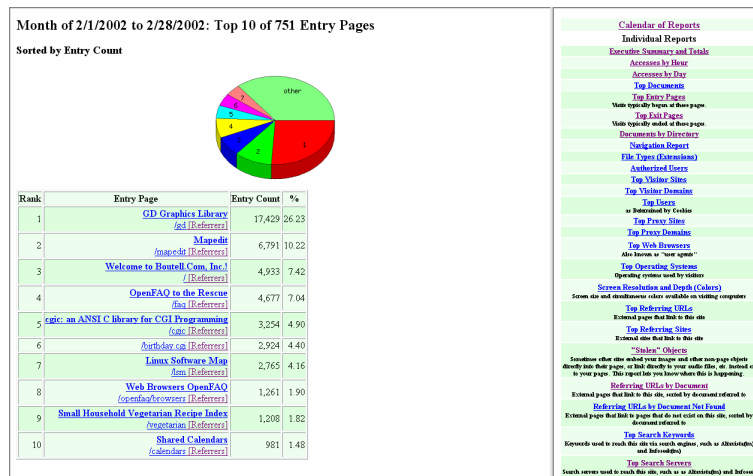


Figure 30 Wusage 8 - Example of report organization [source: [Wusage2005], <http://www.boutell.com/wusage/example/daily/2002/02/15/index.html>, Last visit: December 2005]

Webtrends [Webtrends2005] - is a commercial website analysis solution that combines several features of web log analysis and, most important, is one of the few solutions able to analyze the content of the website, catalog its visitors and compute semantic relations between web contents and their usage patterns. The tool is capable of representing regular log analysis reports (Figure 31), as well as innovative sophisticated visual outputs for usage behaviors.



Figure 31 Webtrends example - classic reports represented by graphs [source: [Webtrends2005], <http://webtrends.breezecentral.com/webtrends7/>, Last visit: December 2005]

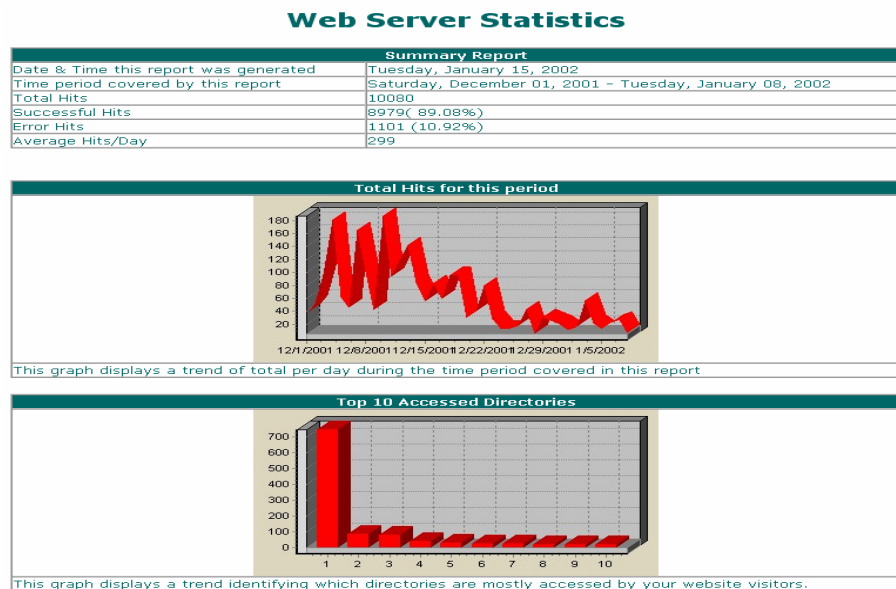


Figure 32 Sitemlogz example report [source: [Sitemlogz2005], <http://www.sitemlogz.com/sitemlogz/sample.php>, Last visit: December 2005]

Sitelogz [Sitelogz2005] - is a real-time log analyzer solution that provides common types of quantitative measures for website usage, extracted from log files. The outputs of the tool are formed of tables and charts (Figure 32).

Sawmill [Sawmill2005] – is another log analysis tool capable to implement the common features for a log analysis tool, yet with some distinctive aspects linked to session identification and path analysis. A sample report is shown in Figure 33.

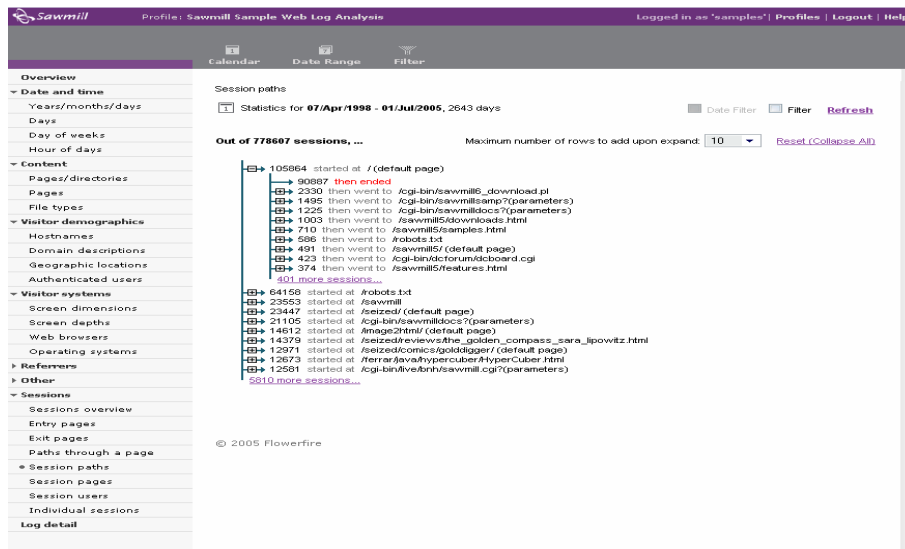


Figure 33 Sawmill example report [source: [Sawmill2005], http://www.sawmill.net/cgi-bin/sawmill7/samples/sawmill.cgi?dp+templates.profile.index+p+sawmill+web+log+analysis+sample+volatile.display+reports+true+webvars.username+user_26955+webvars.password+sawmill, Last visit: December 2005]

FastStats [FastStats2005] – is a log analysis tool capable to implement the common features for a log analysis tool, the reports being represented in a common manner: tables and charts (Figure 34). Of particular interest is the clustering technique used for navigational purposes.

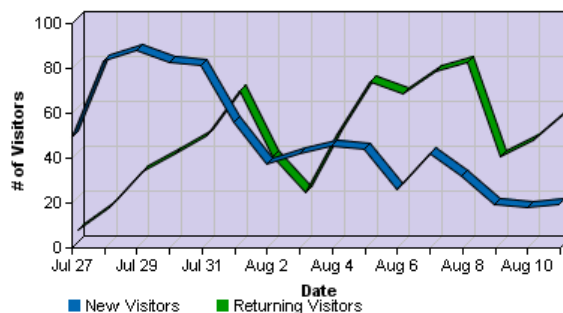


Figure 34 FastStats report example [source: [FastStats2005], <http://www.mach5.com/products/analyzer/cpc-profitable.php>, Last visit: December 2005]

Opentracker [Opentracker2005] – is a web log analysis and website tracking system. Part of the system enables the analysis of the application-server log files and the extraction of common information. The information is then presented in a regular fashion, using tables and charts (Figure 35). Another part represents the real-time analysis and monitoring system, which is discussed in section *Structure and contents*.



Figure 35 Opentracker country identification report example [source: [Opentracker2005], http://www.opentracker.net/en/demo/screenshots/screenshot_countries.jsp, Last visit: December 2005]

Deep Log Analyzer [Deep2005] – is a web log analysis tool that provides the common analysis features and other new features as a scripting engine that allows customization and an interactive hierarchical presentation system (Figure 36).

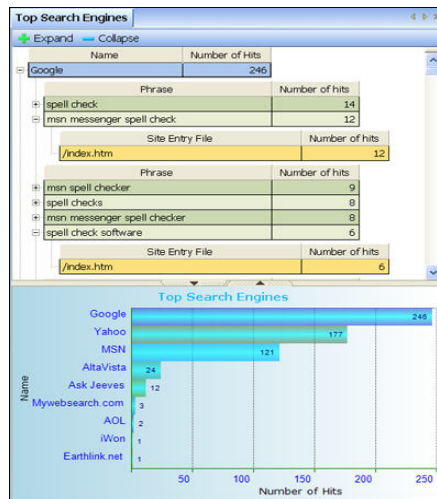


Figure 36 Deep Log Analysis hierarchical interactive presentation [source: [Deep2005], <http://www.deepsoftware.com/default.asp>, Last visit: December 2005]

iWebTrack [IWEBTRACK2005] - is a log analysis solution able to produce common types of reports for the website usage. It combines visual enhancements of report presentations and some features of integration with other applications (Figure 37). It is also capable of analyzing campaigns based on keywords and implement website availability monitoring.

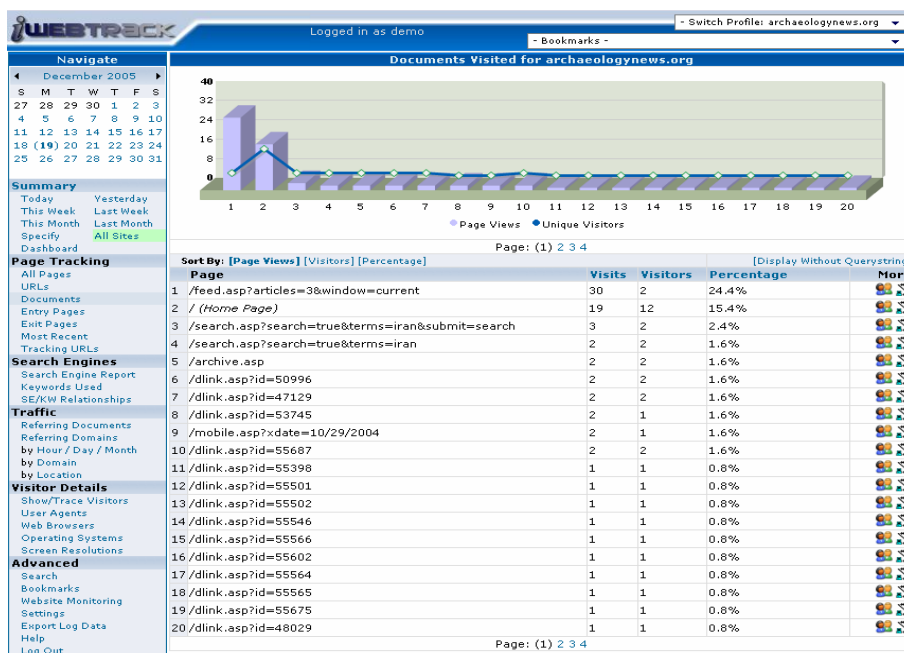


Figure 37 iWEBTRACK example report [source: [IWEBTRACK2005], <http://www.iwebtrack.com/login/documentsvisited.asp>, Last visit: December 2005]

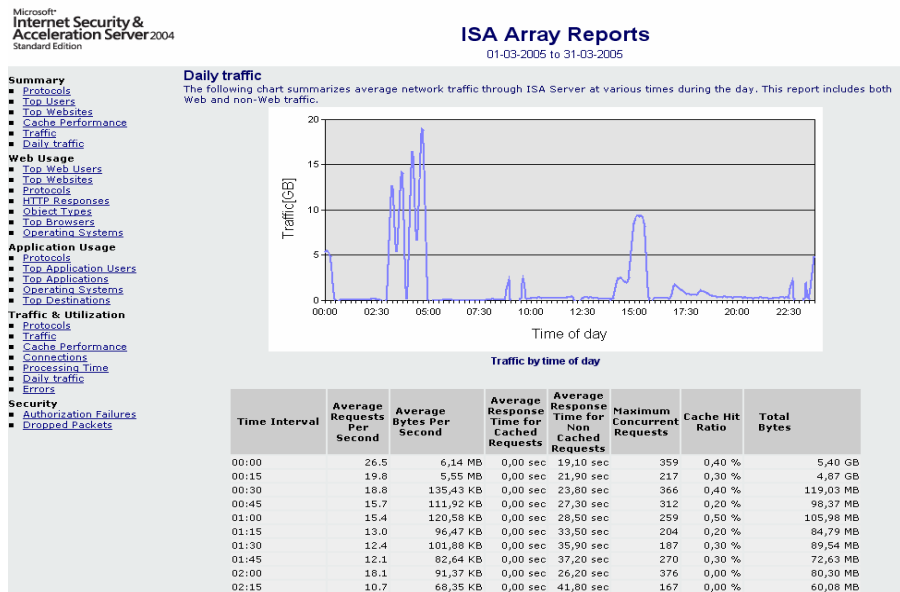


Figure 38 ISA Server 2004 traffic report example [source: [ISAServer2004], Last visit: December 2005]

Microsoft ISA Server 2004 [ISAServer2004] – is a solution that provides several services, one of which being web server access logging and analysis. During the daily usage of websites, the application logs clients’ requests. Starting from the analysis of the log files, a custom, manual or automatic report can be produced, containing common statistical information regarding quantitative measures for the website traffic. The output is a set of HTML reports that use common layouts – tables and charts (Figure 38).

2.6.2. Structure and contents analysis

LiveSTATS [LiveSTATS2005] – is a solution that allows the analysis of three profiles of websites: .BIZ – for business customers, .NET – for regular websites, .XPS – for service providers. It provides the common log analysis functionalities and an extended set of profile-oriented analysis techniques for business, regular or service provider websites in real-time. Content filtering and analysis is provided, in addition, historical analysis and evolution of the website (Figure 39). On the right side of the Figure 39 an example of the representation of the hyperlinks with the corresponding usage information overlaid is shown.

ClickTracks [ClickTracks2005] – is a web analysis solution able to pinpoint usage information linked to the website contents. It is able to catalog the evolution of the website contents through statistically significant variances and ignoring simple fluctuations; in addition, the reporting mechanisms can display visual representation of usage data by overlaying it over the page contents (Figure 40). This tool is also able to provide synchronized views of website versions, with overlaid hyperlink usage information (Figure 41).



Figure 39 LiveSTATS reports examples [source: [LiveSTATS2005], <http://www.deepmetrix.com/livestats/net/tour/>, Last visit: December 2005]

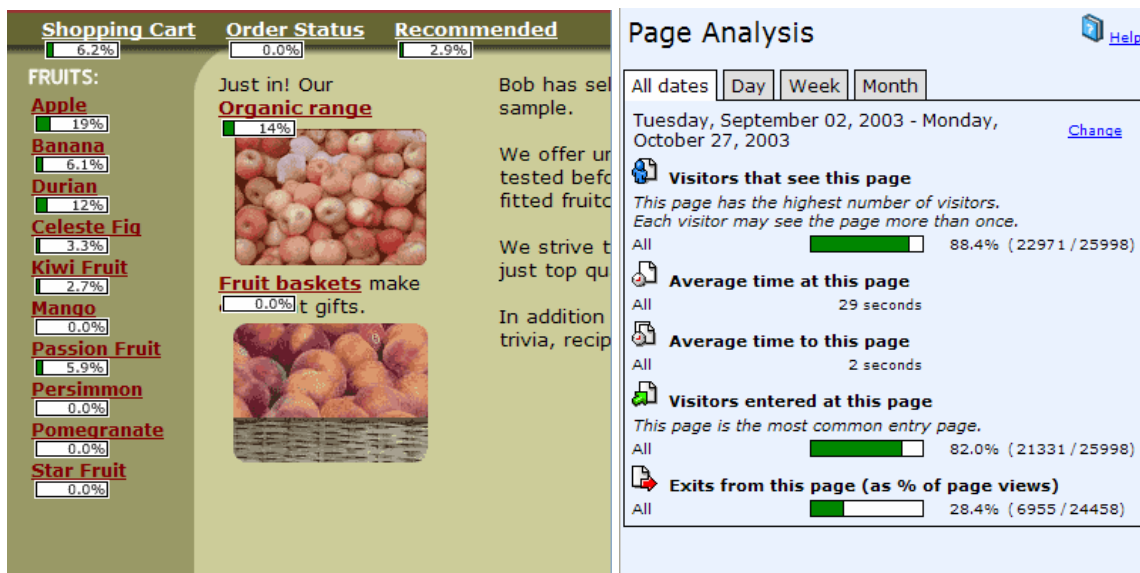


Figure 40 ClickTracks overlaid usage information [source: [ClickTracks2005], http://www.clicktracks.com/demos_small/navreport.php, Last visit: December 2005]

Opentracker [Opentracker2005] – is a web log analysis and website tracking system capable to perform real-time analysis of website usage. The system uses a tracking system implemented as scripts that run at client side and analyze the client’s behavior while exploring the website. Perhaps, this approach is the most accurate for session identification and usage behavior analysis. Another important new feature of the tool is the capability of monitoring the website registered online visitors (Figure 42).

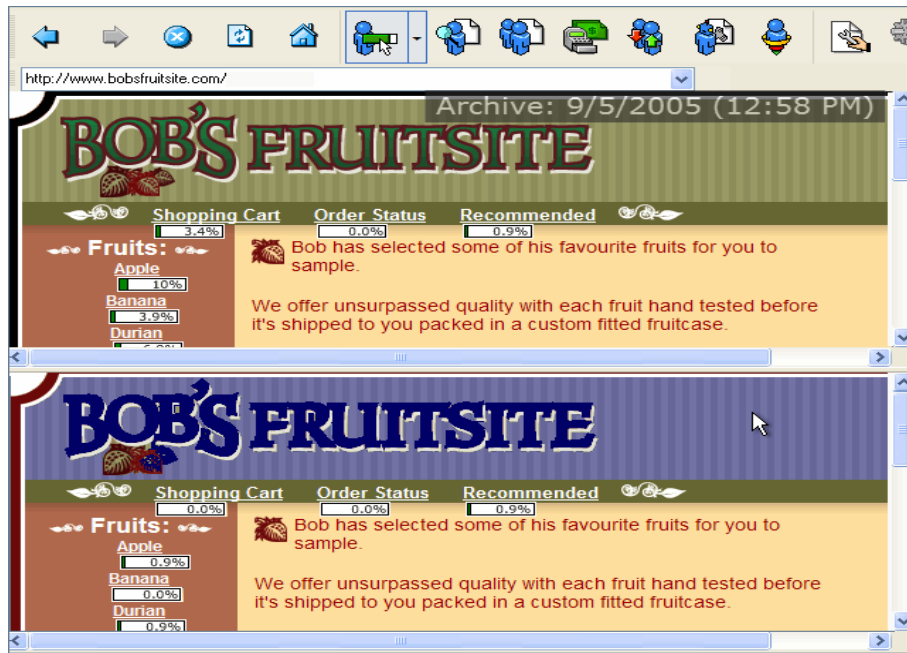


Figure 41 ClickTracks synchronized view with overlaid usage data [source: [ClickTracks2005], http://www.clicktracks.com/demos_small/navreport.php, Last visit: December 2005]

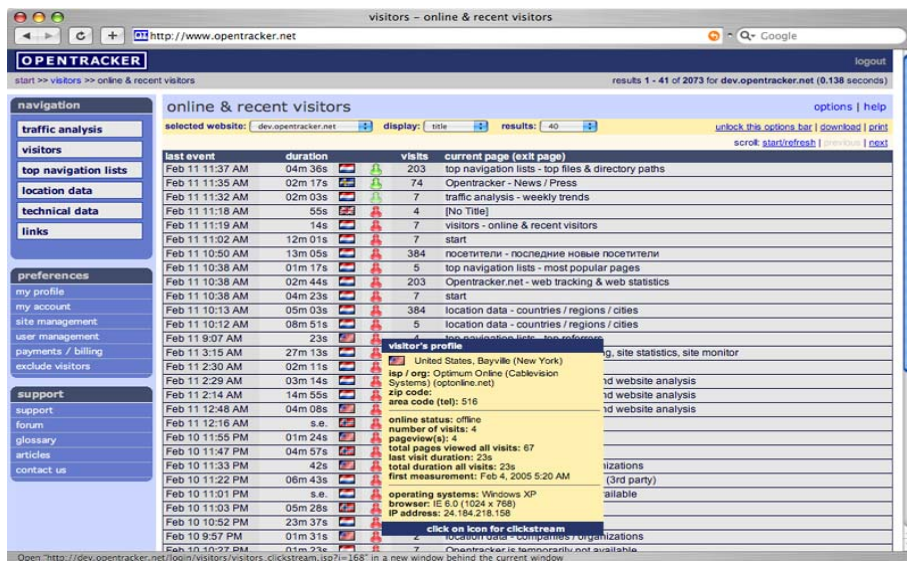


Figure 42 OpenTracker online visitors report example [source: [Opentracker2005], http://www.opentracker.net/en/demo/screenshots/screenshot_visitors_online.jsp, Last visit: December 2005]

Webtrends [Webtrends2005] - is a complete website analysis solution capable to analyze the content of the website, catalog its visitors and compute semantic relations between web contents and their usage patterns. This tool is able to analyze the website structure and usage logs and to combine the resulting information into visual representations used to determine usage patterns. Figure 43 presents the overlay technique used to

superimpose usage information over the real website content. A more detailed presentation of the tool is presented in section Usage analysis.

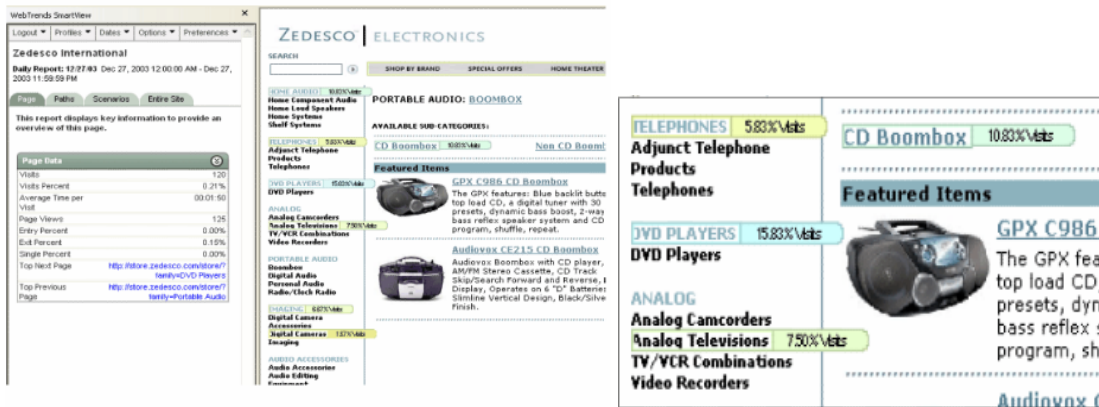


Figure 43 Webtrends example - Overlaid web usage data [source: [Webtrends2005], <http://webtrends.breezecentral.com/webtrends7/>, Last visit: December 2005]

2.6.3. Usage analysis

FastStats [FastStats2005] is capable to display a dynamic diagram of content exploration pattern, presenting the browsing history as a “butterfly” workflow of Referrers→Page→Children (Figure 44).

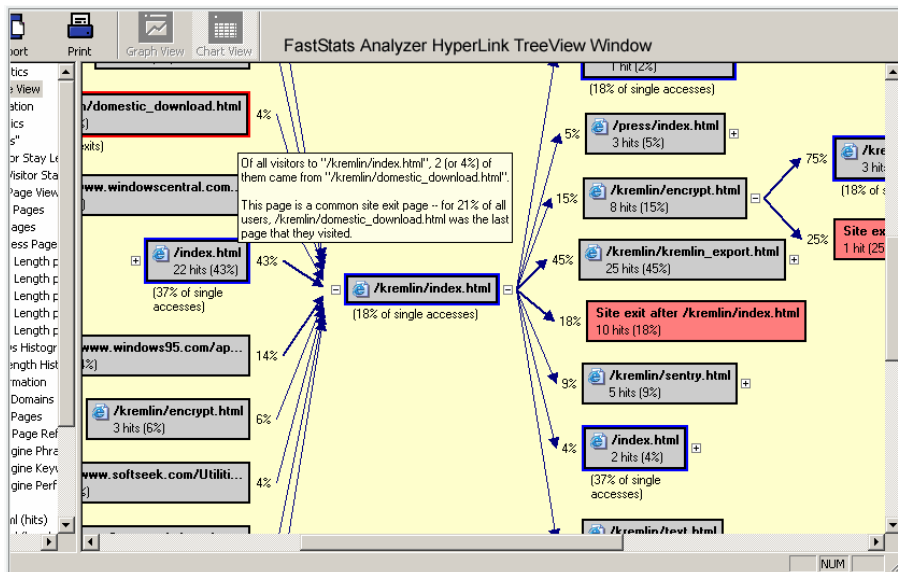


Figure 44 FastStats hyperlink tree view example [source: [FastStats2005], <http://www.mach5.com/products/analyzer/hyperlink-mo.php>, Last visit: December 2005]

Webtrends [Webtrends2005] - is a completely innovative commercial website analysis solution that combines several features of web log analysis and, most important, is one of the few tools capable to analyze the contents of the website, catalog its visitors and compute semantic relations between web contents and their usage patterns. The tool is

capable of representing regular log analysis reports (Figure 31) and innovative visual outputs for usage behaviors (Figure 43, Figure 45 and Figure 46). Some of the innovative features of the tool are:

- Semantic analysis of website contents and its relations with the usage patterns (the so called *Marketing Campaign Analysis*) - allows the analysis of a portion of the website in specific circumstances;
- Scenario analysis – meaning the complete analysis of user’s behavior related to a specific set of tasks, providing a visual representation of the conversion of the user’s habits (Figure 31 – right quadrant);
- Overlaid quantitative values of usage measures for web pages (the so called *Smart View*) implemented with overlaid web usage data for each hyperlink of the web page (Figure 43 – left quadrant);
- Usage path to goal analysis represented by the most popular paths followed by users to achieve a specific goal (the so called *contents effectiveness*) (Figure 46 – left quadrant);
- User behavior classification (the so-called *segmentation and retention*) represented by the capability of segmenting the overall site users into user groups, identified by different interests, preferences and usage habits within the website. Then, segmentation rules can be applied for the different groups, to allow experts identify special users and deliver suitable user needs;

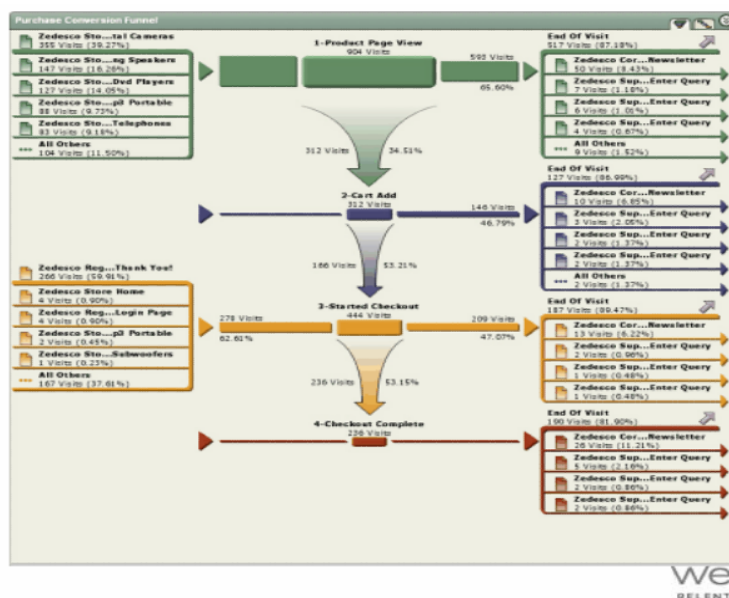


Figure 45 Webtrends example - scenario analysis [source: [Webtrends2005], <http://webtrends.breezecentral.com/webtrends7/>, Last visit: December 2005]

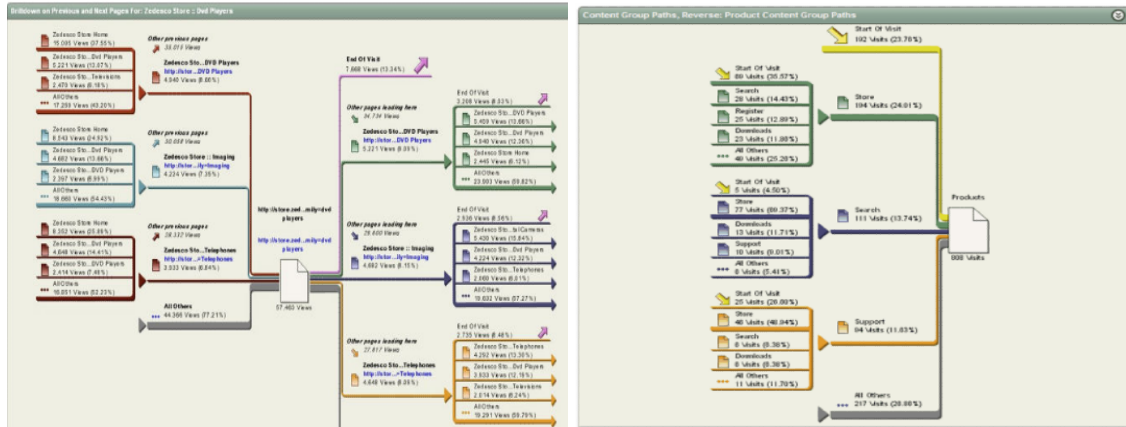


Figure 46 Webtrends examples - Contents effectiveness of usage data through path analysis [source: [Webtrends2005], <http://webtrends.breezcentral.com/webtrends7/>, Last visit: December 2005]

Ethnio [Ethnio2005] – is a remotely moderated usability testing software that allows to:

- i. “Easily observe and record usability behavior from participants anywhere in the world;
- ii. Capture click stream data and desktop video;
- iii. Create and manage recruiting screeners;
- iv. Capture live users’ requests as they visit a site.”

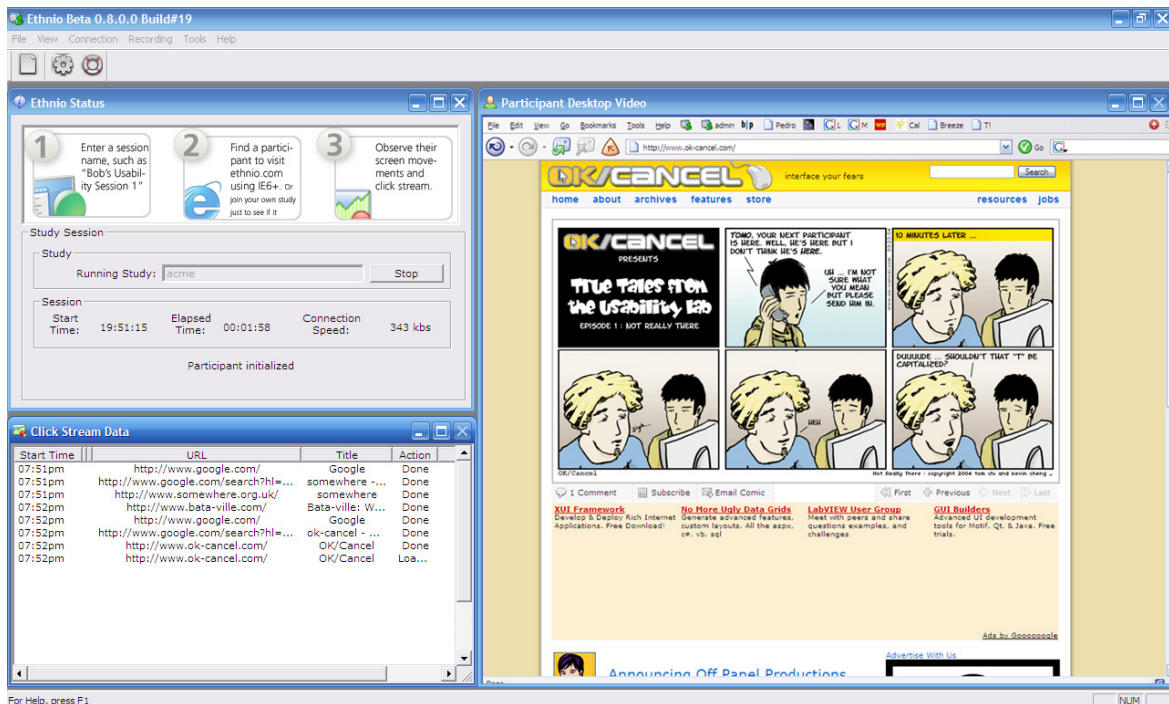


Figure 47 Ethnio proof of concept [source: [Ethnio2005], <http://ethnio.com/>, Last visit: December 2005]

Google Analytics [GoogleAnalytics2005] - is a free website analysis solution based on a tracking system that must be included on every page of the website as a script. The solution tends to identify website contents and relate it with online usage statistics collected from both live usage and central repositories. It is able to identify campaigns by keyword and is integrated with the Google's *AdWords* promotion system, to perform advanced visitor segmentation, demographical localization, to report trends and to represent trends and usage patterns using innovative visualization techniques (Figure 48 – left quadrant). The *funnel visualization* (Figure 48 – left quadrant) is meant to analyze and visually represent the steps required for the user to perform a specific task, combining visually represented quantitative measures for entrance and exit points of each step. It performs visualization techniques of website pages superimposed with click and conversion data for each link (Figure 48 – right quadrant) and comes with over 80 predefined reports.

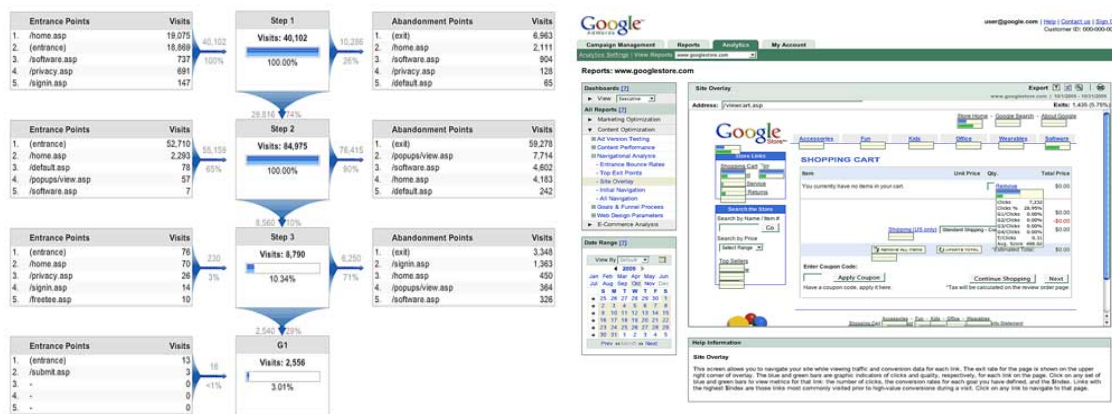


Figure 48 Google Analytic funnel visualization and site overlay [source: [GoogleAnalytics2005], left - http://www.google.com/analytics/feature_funnel.html, right - http://www.google.com/analytics/feature_overlay.html, Last visit: December 2005]

2.7. Discussion

We classified the past and current trends in web analysis and information visualization techniques using five predominant aspects based on the properties of the websites, previously introduced in the introduction of this chapter.

The representation of a website is a sensitive issue due to the enormous quantities of interconnections that need to be captured, filtered, classified and represented. Usage information collected from natural site usage or controlled experiments is another sensitive topic of interest that involves complex algorithms used for session detection, information filtering, clustering and representation.

Most of the presented methods and applications lack visual information background and do not have the usability data reporting on source page contents neither on its

hierarchical structure. Many advantages result from the representation of the page elements as images, contextualized on the site representation:

- On the synchronized hierarchical site representation, users can easily identify the focused page;
- Page elements and usability information give more feedback if they are overlaid on each other, helping the user better perceive the usability of the page;
- Contextualized visual feedback for page design, obtained based on statistical information on individual page usage, is a powerful option that helps the detection of possible navigational problems and information interpretation; etc.

Based on the conceptual and functional aspects of each presented technique or application, a set of conclusions is raised:

1. The representation of a website involves a primary step meant to discover the structure of a website using several crawling techniques (combined with both BFS and DFS traversals of the graph that represents the website structure [Najork2001], [Chen2004]). Note that this process is very complex, dealing with enormous sets of information and several website creation technologies based on dynamic content creation or personalization; these approaches make the classification of a website as a set of nodes identified by a unique URI (Unique Resource Identifier) more difficult. Next, the information is organized and filtered, with the purpose of identifying unique nodes of the graph, which can be considered as self-contained entities. The result is then represented using a visualization technique that combines hierarchical, radial or other 2D and 3D representation of the graph associated with the website structure. Hierarchical representations as vertical or horizontal organizations of website-associated graphs were mentioned by [Dodge2003], [Ricca2001] and [Martin2001]. The hierarchical representation has the advantage of being very close to logical organization of a website. The main disadvantage is that complex websites cannot be represented simultaneously, making this type of representation less readable. [Chi1998], [Chi2002], [Chi1999], [Niu2003] and [Chen2004] used radial representations of the website-associated graphs, combining 2D and 3D in some cases. Radial representation is suitable for representing medium to large structures, giving more visibility to the entire site structure. Sometimes, access to subset of the graph facilitates users' navigational task. Usage information is superimposed on top of the representation to give insight of usage patterns. Given the functionalities of the implementations of this radial representation technique, one may consider it as the most valuable representation available. [Chen2004] and [Ricca2001] combined a set of hierarchical representations based on clustering, which highlight distinguishable information clusters within the website. The hyperbolic tree presented in [Munzner1997] is suitable for contextual focused explorations of the website structure; clusters of structural information are represented using a spherical 3D space, the

navigation being facilitated by translations between information clusters into the active user context of analysis. Yet another representation of the website structure and its usage information is proposed in [Youssefi2003]. The authors concentrated more on eye-candy visualization techniques, resulting of mining the information extracted from website logs, and neglected their usage to extract useful information about the website usage. Direct mapping of website information could have helped improve the overall identification of the represented objects.

2. Contents classification and information clustering might represent one of the most demanding task in terms of processing raw data obtained from the website structure and usage logs. [Niu2003], [Ricca2001], [Chen2004] and [Cooley2003] have done important work on the field of identifying the semantic contents of a website, discover its information clusters and map the resulting information on the structure to obtain insight of usage patterns and navigational paradigms. Yet, this field of research is becoming a more and more demanding task, given the complexity and dynamism of nowadays websites.
3. Usage data / information represent a very important topic for the analysis of websites. Three major steps are required to collect and process this type of information: *Gather*, *Filter* and *Classify* or *Define* the resulted information. Mainly, two important sources for usage information are analyzed, or a combination of them: application server log files [Heer2002], [Fraternali2003], [Fraternali2003]; controlled experiments logs [Paganelli2002], [Card2001] (which also included user-tracking devices as eye or mouse tracking logging). The information collected from the log file has to be filtered before the analysis process can proceed with session identification and usage pattern discovery, as referenced by [Spiliopoulou2000] and [Heer2002]. Both, the website structure and the access logs need to be filtered using specific algorithms as the ones presented by [Youssefi2003]. The classification of usage information represents another important step before any usage pattern can be discovered. Several mining and semantic data analysis techniques can be used to identify and classify clusters of information and to track users' behavior by identifying usage sessions, as presented by [Cooley2003], [Youssefi2003] and [Fraternali2003].
4. Visual representation are required to help interpret the information related to website structure, usage and usage / navigational patterns. The visualization of such information can be accomplished considering three major areas of website information visualization techniques used to visualize the website: structure, contents and usage data quantitative or qualitative measures. The structure of a website can be represented using several types of site maps, as classified by [Dodge2003]. However, the structural representation techniques need to provide interaction mechanisms and visual filter for the information; [Niu2003] and [Chen2004] proposed several means of interaction with the represented information, providing several filtering and context

manipulation techniques. [Chi1998] provided a way to study the evolution of a website by representing its structure on a time-line, the so-called *Time-Tube* visualization. Another important navigation paradigm was introduced by [Munzner1997] with the hyperbolic browser. One thing is to have access to the website structure, yet another is to access its contents so that navigational malfunctions can be identified by means of semantically associations of contents and structure. [Card2001] introduced the metaphor of associating the content with the logging of user events and his/her focus of attention. Eye tracking and input tracking systems were used to track the navigational experience deep on the origins of human visual perception system. Then, the authors coded the user experience using color, thickness, shape combined with overlay techniques over the contents of the explored web pages. [Faraday2000] introduced a tool for the analysis of which interface elements attract user attention. The result is a set of scanning paths overlaid on the page contents, used to identify which are the top focused areas of a page. Another method of analysis is to compare visual features of the target design to features of highly rated sites, and then signalize a set of rules as predictions, similarities, differences and suggestions, as introduced by [Ivory2002]. Even if there are many techniques that display usage information mapped over the website structure, many of them provide quantitative measures for usage, as presented by [Chi2002], [Chi1998], [Niu2003], [Chen2004], [Paganelli2002] and [Youssefi2003], not behaviors. In addition, [Chen2004] uses several information layers combined with a *web graph algebra* that provides visual insight of qualitative measures obtained from combining several information layers with mathematical filtering and computational techniques identified by the algebra. [Paganelli2002] presented a tool able to produce qualitative evaluations of the user-performed tasks, while [Card2001] introduced a tool able to synthesize the results as a *Web Behavior Graph*, used to interpret users' experience.

5. Dynamic information exploration represents the means of interaction provided by each discussed website analysis technique. The combination of two and three-dimensional representations might improve navigational experience of the user that uses a analysis tool, but it might also decrease the perception of the real meaning of the represented information (occlusion being one of the issues). A visualization tool needs to provide a rich set of interactions with the represented information, directed focus, context zoom-in or other technique being appropriate to this purpose [Chi1999]. Aggregation and animation are yet other useful interaction tasks, as introduced by [Niu2003]. Dynamic layout of the radial tree and time-based filtering of the displayed information were introduced by [Chen2004] to improve the observation process. [Munzner1997] introduced transitions based on animation and optimized re-clustering of the out-of-focus nodes.
6. Even if good steps have been given towards the automation of the process of website analysis and usability evaluation, few of the actual techniques manage to provide a

set of suggestions based on the anomalies detected during the analysis process. Design malfunctions from either UI perspective or navigation are difficult to be obtained by humans if no appropriate tools are provided to help the synthesis and automation of the analysis process. Nevertheless, good tools cannot completely replace the human agent, but they can highly increase productivity by providing the appropriate techniques for processing the enormous quantities of information produced during daily website usage or controlled experiments. It might be difficult to automatically provide solutions for specific situations, but good inference analysis rules might help the analysis of page clutter, some UI validations or even to provide some suggestions based on empirical experiences. Following this idea, *Design Advisor* [Faraday2000] implements a few of these automated critiquing techniques, focused on the detection of highly attention attractive elements of a web page. [Ivory2002] proposed a tool that can be used to address potential design issues before conducting usability testing. Note that this design-checking approach is not intended to replace usability testing, but rather to complement it. The tool proposed by [Paganelli2002] is able to highlight basic usability problems, allowing usability experts to interpret the results of the analysis of experimental usage sessions to obtain critical usage insights.

Several applications for website logs or usage analysis were analyzed with the purpose of clarifying the relation between the research area on one side and the actual worldwide implementations of the concepts for the open market. As previously described in the beginning of the section 2.6 - Applications, we classified nowadays commercial and noncommercial applications in three classes: *Basic log analysis*, *Structure and contents analysis* and *Usage Analysis*.

1. *Basic log analysis* – most of the applications discussed in this class serve a common purpose: analyze web application-server log files and compute quantitative measures of website access statistical information. Many of these applications display the resulting information using sets of tables, charts or a combination of these visual representations: **AWStats** [AWStats2005], **Wusage** [Wusage2005], **Webtrends** [Webtrends2005], **Sitelogz** [Sitelogz2005], **Sawmill** [Sawmill2005], **FastStats** [FastStats2005], **Opentracker** [Opentracker2005], **Deep Log Analyzer** [Deep2005], **iWebTrack** [IWEBTRACK2005] and **Microsoft ISA Server 2004** [ISAServer2004]. Of particular interest are **Webtrends** [Webtrends2005] and **FastStats** [FastStats2005] that, in addition to the analysis of server logs, are able to analyze the website contents and to semantically relate usage patterns with the contents. The **Opentracker** [Opentracker2005] also introduces a real-time analysis module, able to display the real-time activity analysis of a particular website. **iWebTrack** [IWEBTRACK2005] is able to analyze so called campaigns (contents promoted on a specific time period) based on keywords filtering. Most of these tools are able to capture and represent information as network traffic during time and its relations with usage, user

identification and profile clustering, etc. (related with the computation metrics presented in the beginning of the 2.6.1 - Basic logs analysis subsection). However, only some of the tools are able to relate the statistical information extracted from the log files with the content of the website.

2. *Structure and contents analysis* – this class includes the applications able to analyze the structure of the website, to classify its contents using mining techniques, and compute statistical information based on content classification, usability metrics and website usage information. Most of these tools are able to highlight usage patterns based on the content of each page, combining the usage information with sophisticated algorithms meant to detect usage trends and predictions. Then, the information is presented using overlay techniques over the contents of each page of the website: **LiveSTATS** [LiveSTATS2005], **ClickTracks** [ClickTracks2005], **Webtrends** [Webtrends2005]. Some of these tools are able to implement contents filtering, providing profile-oriented analysis techniques for business customer needs, in addition, might combine historical information of website evolution. However, most of the tools lack in techniques for mapping local information on top of overall representation of website structure, or provide synchronized navigational mechanisms for the exploration of the information. The visual layout of the website user interface is not analyzed by any of the presented tools, being yet a unexplored area introduced with the proposal of our *Website Analysis Tools for Communication and Information Management* applications framework.
3. *Usage analysis* – represents a class of applications able to combine the information resulting from the analysis of website structure, usage logs, contents classification, to obtain valuable usage patterns. The applications that implement these techniques are able to combine several quantitative measures of usage information, with the purpose of obtaining qualitative measures for website user's behavior. Several representation techniques are used to display the information related to the identification of usage sessions, clustering techniques being applied to semantically related information "islands". Perhaps, the path-to-goal representation (also known as content effectiveness - **Webtrends** [Webtrends2005] or "butterfly" representation - **FastStats** [FastStats2005]) is one of the most suitable for the analysis of usage behavior (Figure 44, Figure 46). The so called "*funnel visualization*" introduced by **Google Analytics** [GoogleAnalytics2005] is meant to analyze and visually represent the steps required for the user to perform a specific task, combining entrance points statistics with failure exit points in a visually represented manner. **Ethnio** [Ethnio2005] represents one of the few tools able to remotely capture usage behaviors during usability testing sessions for given websites. The advantage of such a tool it manipulates the real usage logging information analyzed directly, without passing into a process of filtering and session identification like in the case of web usage log analysis applications.

Even if there are many analysis tools on nowadays markets, few manage to comply with specific usability guidelines, or to provide flexible and easy-to-use means of interaction with the represented information. There is a tremendous gap between research trends and actual implementations of the concepts. A united and flexible user interface that combines many of the analysis features is yet in a stage of requirement specification.

Tables, Table 1.1 to Table 1.8, present the synthesis of the most important features provided by actual research trends and real implementations of website analysis tools and techniques.

Table 1.1 Website analysis and visualization - tools and techniques

Website Classification							
Name	Data Sources	Data Gathering and Filtering	Visual Abstraction	Visual Representation	In View Interaction	Overall Views Synchronization	Comments
Mapuccino Site Map - IBM's Haifa Research Lab	Hypermedia structure of the website	Create tree layout / radial graph (called fish eye) from the results of crawling website structure	Breadth first transformation into hierarchical tree layout / radial graph (called fish eye) from the hypermedia structure	Create tree / graph using definition of levels of hyperlinked nodes = pages	Selection of nodes, translation, zooming	Single view display at a time	The tool is able to display several representations of the website structure, only one at a time.
Disk-Tree, Cone-Tree	Hypertext or web linkage structure; application server usage log files	Extract website hypermedia structure into graph. Mining techniques are used to transform raw data into potentially usage information	Do breadth first traversal and use a visualization abstraction: tree hierarchy	Cone tree: Layout using 3d cones; layout using disk tree	Focus node; hide subtree; change orientation and position of tree; apply dynamic level-filtering	Disk-Tree is used in synchronized visualization spreadsheets	This visual representation of the website combines the hypermedia structure as well as usage information mapped on the website structure.
Time-Tube	Hypertext or web linkage structure; application server usage log files	web structure evolving over time and its associated usage statistics (content, usage, and topology of the web site)	Do breadth first traversal with global node position over time. Create graph from web structure by crawling the web site	Create time tube, which is represented using an aggregation of disk trees (invisible tube-like shelf)	Recognize gestures for: focus on a slice; bring slices back into the time tube; zooming focus on the connectivity of a node by right clicking on it; rotate slices; brushing on slices	Single view display at a time	Good feedback for the evolution of the website.
WebKIV	Hypertext or web linkage structure; application server usage log files	Extract website hypermedia structure into graph. Mining techniques are used for filtering	Breadth first transformation into hierarchical tree layout / radial graph (called fish eye) from the hypermedia structure	Layout using disk tree	Selections, zooming, multiple sessions animation, replay the natural usage using animated transitions of the representation	Synchronize to selection, synchronize selected viewport	Introduced animation techniques to replay natural website usage. Synchronized views that allow focusing on a specific sub-context.

Table 1.2 Website analysis and visualization - tools and techniques (continued)

Website Classification							
Name	Data Sources	Data Gathering and Filtering	Visual Abstraction	Visual Representation	In View Interaction	Overall Views Synchronization	Comments
Web Knowledge Visualization and Discovery System	Hypertext or web linkage structure; application server usage log files	Extract website hypermedia structure into graph. Mining techniques are used to filter raw data	Do breadth first traversal and use a visualization abstraction: tree hierarchy. Uses a web image with overlaid information layers	Combined a radial representation of the website structure, the root of the website being the center of the representation, the pages being represented as nodes, and the links as edges	Focus node; hide subtree; change orientation and position of tree; apply dynamic level-filtering; apply	-	Very important for the overall feedback provided. The concept of information layers helps combining several information sources to attain valuable usage information and detect usage
VWM - Visual Web Mining	Hypermedia structure of the website and the application server log files	Use mining techniques to filter website structure, the raw information in the log files and to detect usage sessions	Several fashionable 3D representations using clustering, radial representations or other combinations	Uses clustering techniques and circular representations for the website structure and usage information. Display usage sessions as falling clusters on a radial	Several generic 3D manipulations provided by the VTK -Visualization Toolkit	-	Impressive visualizations but less effective because of the missing mappings of the represented information on the website structure and content.
VisVIP	Hypermedia structure of the website and the application server log files	Uses site crawling and log mining combined techniques	Uses balanced trees and 3D layouts to represent paths followed by users	Uses semi-clustered representations of the website structure and its interconnections, using balanced trees in a three dimensional space	Allows to simplify highly connected websites by suppressing all the edges leading to a selectable node or by simplifying the graph to a tree, whose root is represented by an	-	Interesting for observing user's sessions and the interconnections between the selected pages.
H3 - 3D Hyperbolic Browser	Hypertext or web linkage structure	Uses web crawling to gather the information	Uses cone trees transposed in a hyperbolic space to represent hyperlinks as clusters	Uses hyperbolic 3D cone trees by placing children on a hemisphere around the cone mouth instead of on its perimeter. <i>Butterfly</i> representation of visited nodes	Layout using hyperbolic tree, provides transitions based on animation and optimized re-clustering of the out-of-focus nodes	-	Provides a better usage of the available space and an interesting navigation mechanism.

Table 1.3 Website analysis and visualization - tools and techniques (continued)

Website Classification							
Name	Data Sources	Data Gathering and Filtering	Visual Abstraction	Visual Representation	In View Interaction	Overall Views Synchronization	Comments
Directed Graph	Website hypermedia structure and contents	Design time layouts, crawling and mining of the website structure and contents	Breadth first traversal with clustering support	Semantic clusters of pages represented using 3D layouts	-	-	Directed graphs use clustering for representing the semantics.
WARE - Web Engineering Reverse Application	Website hypermedia structure and contents	Reverse engineering of website structure and contents classification	Breadth-first traversal	Hierarchical topology of related classes obtained from the reverse engineering of each page	-	-	Uses reverse engineering techniques to analyze the website and generate a class hierarchy of concepts that represent the website.

Table 1.4 Website analysis and visualization - tools and techniques (continued)

Website Contents Visualization and Usage Capturing						
Name	Data Sources	Data Gathering and Filtering	Area of Analysis	Visual Representation	Comments	
Web Behavior Graphs	Website contents and usage events capturing in live experiments	Contents analysis, user produced events logging using input devices logging and eye tracking techniques	Usage behavior analysis by relating content semantic to usage patterns	Use of color-coding and transparency to display the user's interaction with the webpage and the adjacent timings required for each operation. The usage behavior graph displays color-coded sequences of actions performed by the user to attain a specific goal.	The approach combines several information sources to detect the behavior of users as sequences of actions required to perform specific tasks. The results are represented by two visualization techniques: one for representing the visual impact of the page contents over the decisions of the user, another to represent the sequences of actions performed to get to a specific goal.	
Design Advisor	Webpage contents	Analysis of each webpage for overall colors used, font dimensions and styles, content disposition, etc.	Webpage elements reading order analysis based on guidelines produced by eye-tracking studies	Uses overlaid graphs to display the reading order of the page elements	Important guidelines for detecting design anomalies of web pages.	
WebTango	Website contents and design layouts	The contents of the website. The comparison information gathered from several websites well rated as having good design	Website design and layouts.	Uses basic outputs as text and images to highlight similarities, differences, predictions or suggestions for a specific website	Introduces a good approach for checking if a specific website complies with the so-called "good design" layouts.	
WebRemUSINE	Application server logs and logs obtained during evolution phases of live experiments	Filters captured events that occur on the client's web browser application during live experiments navigation	User behavior analysis related to sequences of tasks to be performed on live experiments	Sets of textual descriptors of task completion success rates and quantitative or qualitative evaluations of the results	Important for defining a set of tasks to be performed in an evaluation session and to automatically collect the results of the session by analyzing the events that occurred on the client's browser during the session.	

Table 1.5 Website analysis and visualization - tools and techniques (continued)

Applications for Website Logs and Content Analysis							
Application Name	Analysis Field	Data Sources	Analysis Measures	Visual Feedback	Application Features	Overall Visual Interactions	Comments
AWStats	LOG Analysis	Application server Log files	Common statistical results involving: Historical classifications, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Demographical localization, Content classification	Commonly uses numerical values within tables and charts	Able to analyze most common log files within any sizes and identify the geographical location of users	Reports navigation within web browser based interface	Common functionality.
Wusage	LOG Analysis	Application server Log files	Common statistical results involving: Historical classifications, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts	Able of measuring the popularity of the hosted documents, as well as identifying the sites that access the web server most often	Reports navigation within web browser based interface	Common functionality.
Webtrends	LOG Website Analysis, Usage Analysis	Application server Log files, Website contents	Common statistical results, website contents analysis, usage analysis, usage behaviors discovery, semantic content relation with website usage patterns	Common tables and charts and sophisticated visualization methods for displaying usage patterns, trends, etc., overlaid usage information on top of the page	Able to analyze the content of the website, catalog its visitors and compute semantic relations between web contents and their usage patterns	Visual representations exploration and interaction within the application and reports navigation within web browser based interface	Perhaps the most complete commercial solution meant to analyze a website in terms of contents and usage behavior. Provides several visualization techniques and content / usage analysis methods.
Sitelogz	LOG Analysis	Application server Log files	Common statistical results involving: Historical classifications, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts	real-time log analyzer solution	Reports navigation within web browser based interface	Common functionality.

Table 1.6 Website analysis and visualization - applications

Applications for Website Logs and Content Analysis							
Application Name	Analysis Field	Data Sources	Analysis Measures	Visual Feedback	Application Features	Overall Visual Interactions	Comments
Sawmill	LOG Analysis	Application server Log files	Common statistical results involving: Historical classifications, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts	Some distinctive aspects linked to session identification and path analysis	Reports navigation within web browser based interface	Common functionality.
FastStats	LOG Analysis	Application server Log files	Common statistical results involving: Historical classifications, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts	Of particular interests is the clustering technique used for navigational purposes. Presents the browsing history as a "butterfly" workflow	Reports navigation within web browser based interface	Common functionality.
Opentracker	LOG Analysis	Application server Log files	Common statistical results involving: Historical classifications, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts	Common analysis for offline log files and real-time analysis and monitoring system	Reports navigation within web browser based interface	Common functionality.
Deep Log Analyzer	LOG Analysis	Application server Log files	Common statistical results involving: Historical classification, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts	Scripting engine that allows customizations and an interactive hierarchical presentation system	Reports navigation within web browser based interface	Common functionality.

Table 1.7 Website analysis and visualization - applications (continued)

Applications for Website Logs and Content Analysis							
Application Name	Analysis Field	Data Sources	Analysis Measures	Visual Feedback	Application Features	Overall Visual Interactions	Comments
iWebTrack	LOG Analysis	Application server Log files	Common statistical results involving: Historical classification, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts. Introduces visual enhancements of report presentations	Capable of analyzing campaigns based on keywords and implement website availability monitoring	Reports navigation within web browser based interface.	Common functionality.
Microsoft ISA Server 2004	LOG Analysis	Application server Log files	Common statistical results involving: Historical classification, Users identification, Bandwidth measures, Referrers classification, Keywords filtering, Content classification	Commonly uses numerical values within tables and charts. Introduces visual enhancements of report presentations	Has the advantage of manipulating the data logged by the same engine, during the daily based usage of a website. Most of the information is related to traffic measures and	Reports navigation within web browser based interface.	Common functionality.
LiveStats	LOG Website Analysis, Usage Analysis	Application server Log files, Website contents	Common statistical results, website contents analysis, usage analysis, usage behaviors discovery, semantic content relation, extended set of profile-oriented analysis techniques for business, regular or service provider websites	Common tables and charts and sophisticated visualization methods for displaying usage patterns, trends, etc, overlaid usage information on top of the page	Content filtering and analysis is provided, in addition, historical analysis and evolution of the website.	Visual representations exploration and interaction within the application and reports navigation within web browser based interface	An important tool that apply profile-oriented techniques to discover usage patterns and map it on the website content.
ClickTracks	LOG Website Analysis, Usage Analysis	Application server Log files, Website contents	Common statistical results, website contents analysis, usage analysis, semantic content relation and versioning analysis.	Common tables and charts and sophisticated visualization methods for displaying usage patterns, trends, etc, overlaid usage information on top of the page	Able of discovering the evolutions of the website content through statistically significant variances and ignoring simple fluctuations.	Visual representations exploration and interaction within the application and reports navigation within web browser based interface. Synchronized views of website	Page analysis and contents relation is a good feature when combined with overlaid usage information.

Table 1.8 Website analysis and visualization - applications (continued)

Applications for Website Logs and Content Analysis							
Application Name	Analysis Field	Data Sources	Analysis Measures	Visual Feedback	Application Features	Overall Visual Interactions	Comments
Google Analytics	LOG Analysis, Website Analysis, Usage Analysis	Application server Log files, Website contents, Public ranking databases	Common statistical results, website contents analysis, usage analysis, usage behaviors discovery, semantic content relation with website usage patterns.	Common tables and charts and sophisticated visualization methods for displaying usage patterns, trends, task evolution, etc, overlaid clicks information on top of the page hyperlinks	Identifies website contents and relate it with online usage statistics collected from both live usage and central repositories, identifies campaigns by keyword, performs advanced visitor segmentation, demographical	Visual representations exploration and interaction within the application and reports navigation within web browser based interface	Provides several visualization techniques and content / usage analysis methods supported by good visual feedback. Integrated with Google AdWords.
Etmio	Usage Analysis	Live usage data	Captures usability evaluation sessions events for post processing phase	During evaluation session tasks are presented in a multiple views layout that enables task oriented evolution	Observe and record usability behavior from session's participants, capture click stream data, desktop video, recruiting screeners and live	Task oriented interactions on a synchronized multiple views user interface.	Important tool used to capture and analyze live information in remote usability sessions.

Chapter 3. Conceptual System Model and Visualization Methods

This chapter introduces the theoretical fundamentals, concepts and definitions for what we define as an *automated tool for visual analysis of website usage*. It includes a conceptual model for the data collection and visualization system, a mathematical representation for website characteristics, a methodological classification of visualization techniques that can be used to represent the website, details on each visualization technique proposed for each specific set of goals, several inspection mechanisms and some proposed correlation methods for the visualization techniques.

Information visualization is a key factor on the website analysis process. It should be applied to the pre-processed website related information with the purpose to extract insight on usage patterns and visual layouts. Therefore, several visualization techniques should be considered, each with the purpose to highlight specific types of information or results that might be relevant for the analysis process.

According to Spence [Spence2001] “Information visualization is the process of uncovering fundamental relations in large volumes of data, which will support insight into them”. It is a complex process that transforms raw data into visual features and 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 or queries. 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 on the amount of information to be represented, several techniques are available to transform raw data into visual features.

The first section of this chapter briefly describes a conceptual model of the proposed data collection and visualization system. The second section introduces the website representation in terms of structural contents, defines the information that can be extracted from the log files, the conceptual data structures used to represent the information and some characteristics of the manipulated information. The third section describes the methodology we used for obtaining visualizations. The last two sections present the visualization techniques of interest of our system, a classification of these techniques, the description of each visualization method and possible correlations among them. The visualization techniques discussed in this chapter are meant to facilitate the

exploration of website structure, contents, navigational paradigms, as well as the design of interaction and the analysis of user interface coherence.

Some of the concepts discussed in this chapter emerged as a result of a tight collaboration with Nunes and other members of the team, as presented in [Mealha2004], [Nunes2006] and [Santos2004], and therefore, these aspects are referenced in this chapter as needed. One example is related to the attributes used to represent the website, another are the image mockups used to simplify the perception of the proposed visualization methods.

3.1. Conceptual model

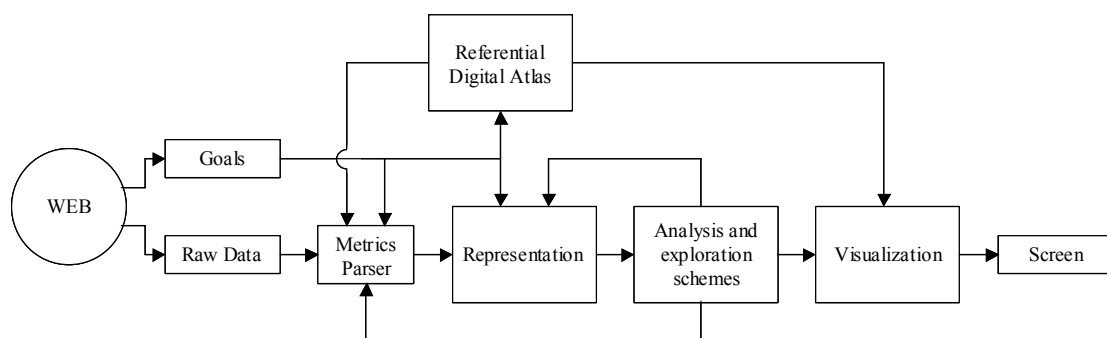


Figure 49 Conceptual System Model

A conceptual model for the visualization subsystem, as introduced by [Mealha2004], is represented in Figure 49, having the following modules:

- I. *Web*: represents the website with all its available services, pages and organization;
- II. *Goals*: represents the set of specific results to be obtained after the analysis process of a website and all the intermediary steps required for each of the goals to be attained;
- III. *Raw Data*: represents the basic information available on its primary state, obtained 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;
- IV. *Metrics Parser*: this module is designed to filter and extract valid information from the 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 session;
- V. *Referential Digital Atlas (RDA)*: represents the information library that contains all information relevant for the system;

- VI. *Representation*: this module includes all available methods for representation of information used to produce visualizations. Several visualization methods are available (2D, 3D, etc.), for specific set of goals;
- VII. *Analysis and exploration schemes*: the representations obtained from the previous module are combined with additional attributes to support live 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 representations;
- VIII. *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;
- IX. *Screen*: represents the media used to display the information and can be implemented using traditional screens or more sophisticated displays as virtual glasses, etc.

A possible implementation of the conceptual model is shown in Figure 50. 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 on, or transferred to/from, the web server, while the fifth component, the *visualizer*, implements the functionality of visualization from the conceptual model. Modules I, II, III and IV have direct mapping to the *web server*, *analyzer*, *interceptor* and *compiler* components, module IV is represented by the *database*, while modules VI, VII, VIII and IX correspond to the *visualizer* component.

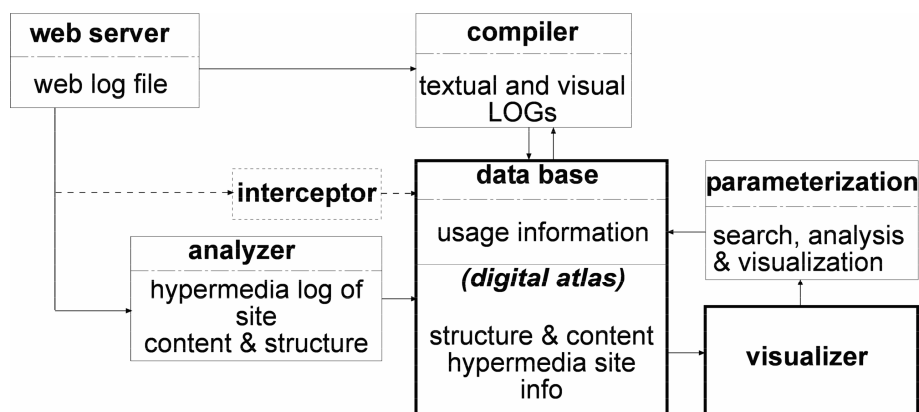


Figure 50 System Model – Conceptual Components

A detailed discussion of these models is included in section Chapter 4 – The Prototype: Objectives and Implementation Details.

3.2. Visualization methods

This section presents the fundamentals for the visualization of website related information, several visualization methods introduced being designed to visually represent its structure, contents, linkage and usage information.

Usability experts require analysis methods that easily provide insight for potential usability problems that can be discovered during the analysis of websites. Visual clues are of great help since they give a rapid response to specific issues. Four possible topics of interest were identified by [Ivory2002] for website analysis purposes: *information design*, *navigation design*, *graphic design*, and *experience design*. The ability to provide an integrated analysis framework that combines all these topics in an integrated visual interface represents an ongoing challenge of worldwide scientists. As mentioned in Chapter 2, integrated visualization frameworks for website analysis and visualization were introduced by: [Chi1998], [Chi2002], [Niu2003], [Chen2004], [Cugini1999], [Munzner1997], [Youssefi2003] for website structure; [Card2001], [Faraday2000], [Ivory2002] for website contents; [Chi2002], [Niu2003], [Paganelli2002], [Youssefi2003], [Chen2004] for usage information.

A *visualization method* is defined as a graphical representation of abstract data usually relayed in text and numbers, with the addition of interaction features. Several graphical elements and features can be combined to achieve a good visualization method, usually used to transform large amounts of information into a set of visual features.

This section describes the visualization methods we elected as most suitable for implementation in our analysis system.

The visualization methods consider the following three classes of analyzed information:

- Information structure – methods that analyze and visually represent the structure / organization of a website;
- Visual workspace coherence – methods that analyze and visually represent the design layouts of one or several web pages;
- Navigational behaviors – methods that analyze and visually represent the evolution and effects of one or several usage sessions.

Structural website representations provide an overview of the analysis space, usually a reference map that needs to be complemented with additional information. Navigational patterns and possible navigation inconsistencies can be highlighted by combining the website structure with the information collected from natural or laboratory usage sessions. An example is a structural website reference map combined with an additional information layer that shows the path followed by a specific user or group of users; additional statistical information can be coded in a second layer, highlighting the most viewed pages during the selected period and set of users.

Visual inspection of interface design combines the analysis of page design layouts, page contents classification, interaction history and user tracking information to provide the means to determine good vs. bad design layouts.

Starting from the classification of the visualizations and using structural attributes of site and page contents, we proposed several visualization methods for analyzing the website structure, contents, navigational and usage patterns and the design layouts visual coherence. The following two sections present a general overview of these visualization methods.

Note that several images in this section are simplified mockups that expose the concept behind the representation, and not the real implementations as found in 4.6.5 – Implementation of visualization methods , where implementation details might occlude some visual clues. Some of the mockups were produced in collaboration with other team members, and are also presented in [Nunes2006], [Mealha2004], [Santos2004].

3.2.1. Visualization methods for visual workspace coherence

3.2.1.1. Page Areas

A webpage is a union of link, text and graphical elements [Ivory2002]. These elements can be organized in information blocks that allow the identification of some specific design aspects, using several available criteria, one of which is the focusing and attention capture classification of page elements proposed by Faraday [Faraday2000].

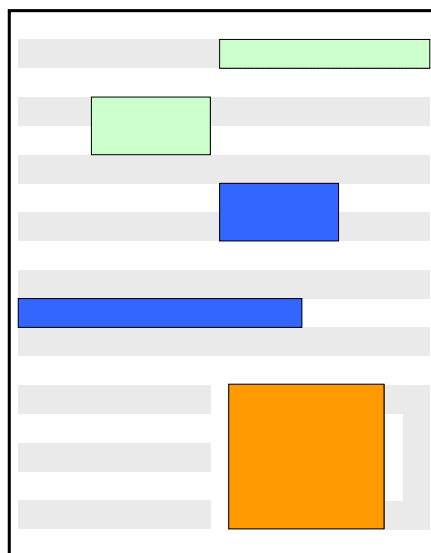


Figure 51 Page Areas visualization [image source: [Nunes2006], Figura III-9]

The base for the representation is the area defined by the dimensions and the snapshot image of the page of the page itself that forms the Base Image (BI) for the representation. All other elements are represented on top of the BI within their natural positions (as

viewed in the browser by the clients). For simplicity purposes, only rectangular shapes were considered for the representation of page elements. Color coding techniques are used to represent distinct element types; additionally, transparency was considered to resolve occlusion problems. Interactive zones and several other attributes are coded to highlight specific design layouts. Figure 51 presents a schematic overview of the visual representation.

This representation is suitable to identify design problems when additional statistical information, related with usage, is represented.

3.2.1.2. Interactive Zones

From the whole set of elements of web pages, only some are interactive and allow the user to navigate from one context to another: the link elements. Using *Page Areas* visualization as reference, *Interactive Zones* represent additional information that consider all pages that can be reached based on the interaction with the link elements of the considered webpage. These are *child pages*, represented using small thumbnails, positioned on the right side of the reference page, the link elements of the page (hotspots) being connected to these thumbnails, as presented in Figure 52.

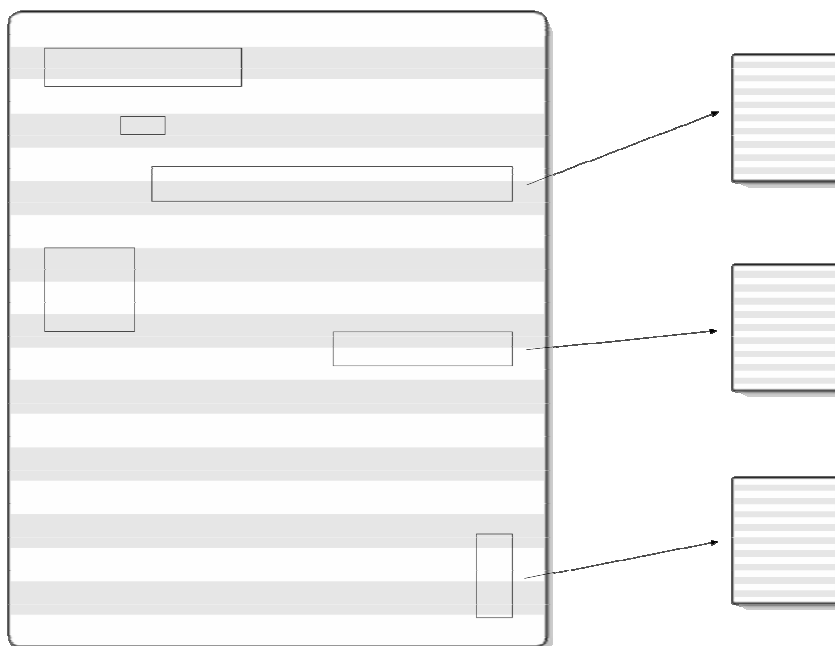


Figure 52 *Interactive Zones* 2D visualization [image source: [Nunes2006], Figura III-10]

Some other structural and statistical attributes can be considered, as well as the usage of filter to suppress or display information. A three dimensional representation of the same visualization considers additional different visual and navigational aspects that can be easily represented within a 3D space to improve the visibility of the information and its relations; one example is presented in Figure 53.

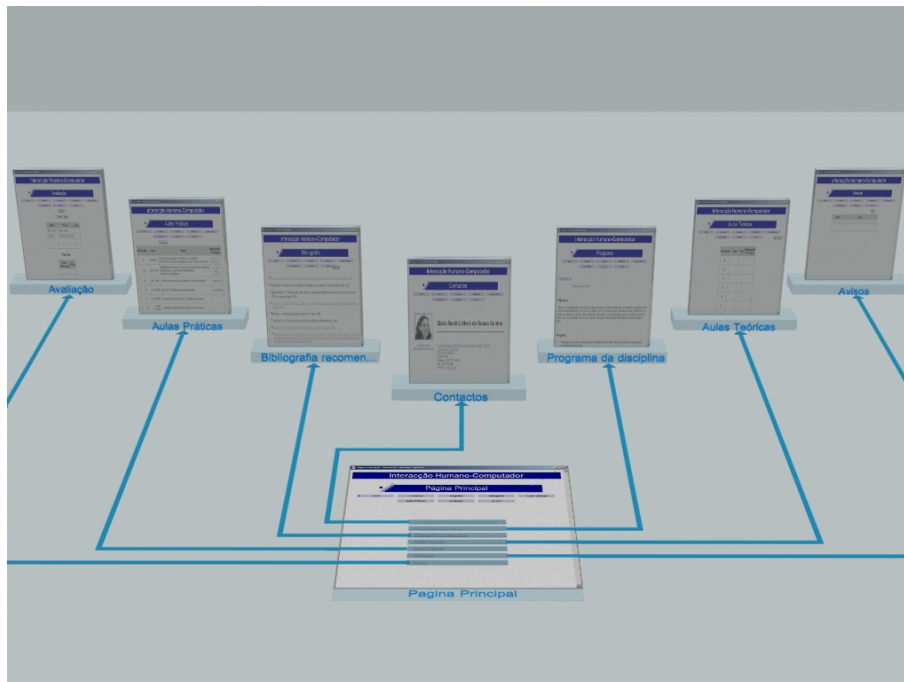


Figure 53 *Interactive Zones 3D visualization*

3.2.1.3. Page Relations

This visualization considers the structural attributes of link elements to identify and represent all possible referrer (parent) pages for the considered page. This visualization extends the *Page Areas* and *Interactive Zones* visualizations with the addition of the URI of the page the link element belongs to, position and size of the area occupied by the link element and the URI of the page that is referenced. These attributes values are visually positioned on top of the corresponding referrer page thumbnails, represented on the left side of the visualization. Each referrer page thumbnail is connected to the analyzed page as presented in Figure 54 using some additional statistical visual feedback elements.

This visualization places the analyzed page in the center of the representation, the link elements on the page are connected to their corresponding pages represented on the right, while the referrer pages are represented on the left side of the figure using connections from the link elements that redirect the context to the analyzed page. It forms a so-called “butterfly” visualization effect, as introduced by [Munzner1997] and [FastStats2005].

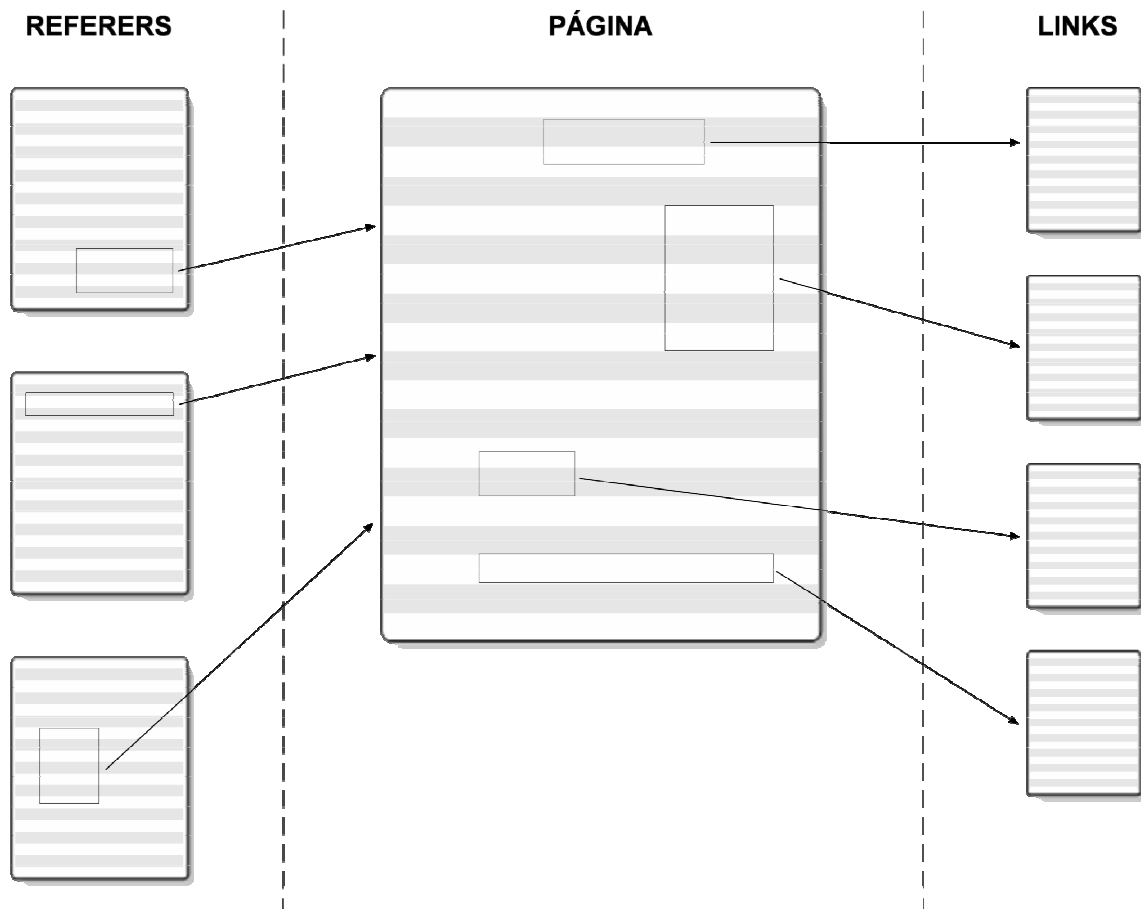


Figure 54 *Page Relations* visualization [image source: [Nunes2006], Figura III-12]

3.2.1.4. Hovering Tips

This is a complementary visualization for most of the visualizations, activated by a user request to represent custom details on hovered items. It makes use of several structural and statistical attributes (e.g. text and graphics) to represent the information. It is activated by a rollover event, being dynamically generated based on the information that describes the hovered item. Figure 55 presents three examples of *Hovering Tips* – tooltip windows and thumbnail on top of hovered items used to represent additional information.

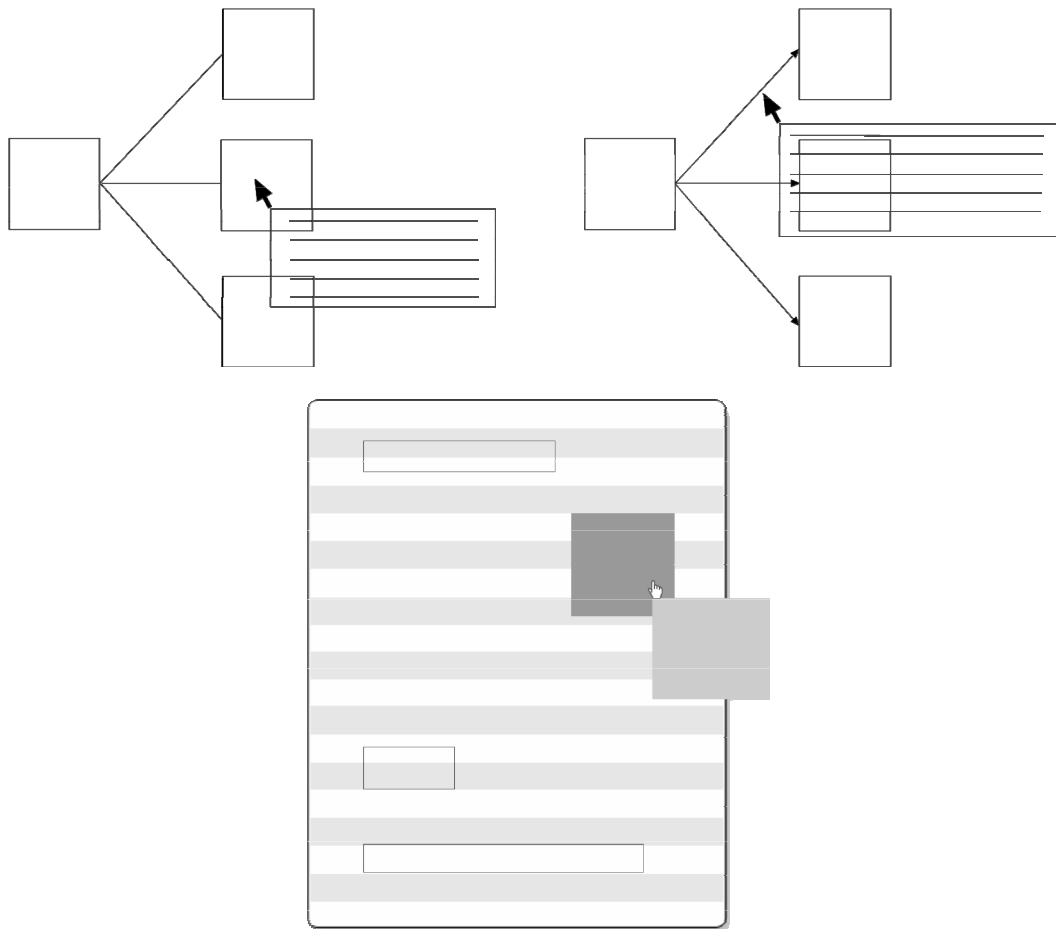


Figure 55 *Hovering Tips* visualization [image sources: [Nunes2006], Figura III-26, Figura III-28 and Figura III-14]

3.2.1.5. Eye-Tracking Layers

Eye-Tracking Layers is a complementary visualization for *Page Areas* and its derived visualizations, activated by visualization users to represent custom statistical eye-tracking details on the page usage. It uses the eye-tracking information captured during webpage usage sessions, visually coding the events as symbolic related icons. In order to provide visual feedback, all events generated by the eye-tracking device are represented over the page reference map as an additional information layer. Subsequent events are connected with lines and the events are represented on the same position the eyes were positioned at the time the event occurred, as presented in Figure 56. The time the user spent to visualize a page area is coded using the icons dimension, the lesser time the user spent on a specific area, the smaller the diameter of the icon. This type of information is very important to determine if a specific page layout is scanned and analyzed by users as predicted by the designer.

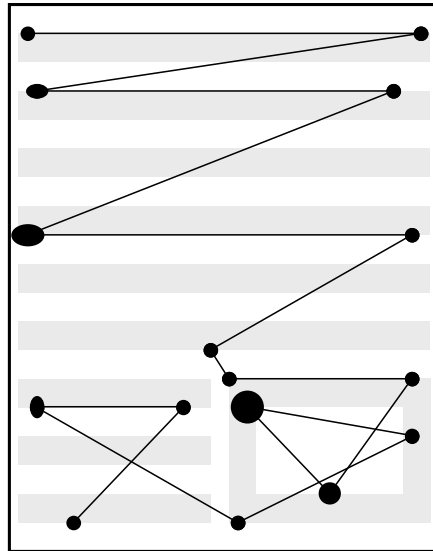


Figure 56 *Eye-Tracking Layers* visualization [image source: [Nunes2006], Figura III-15]

3.2.1.6. Mouse-Tracking Layers

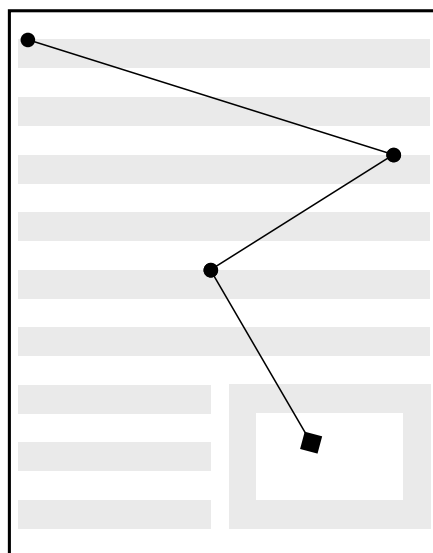


Figure 57 *Mouse-Tracking Layers* visualization [image source: [Nunes2006], Figura III-16]

Mouse-Tracking Layers is a complementary visualization for *Page Areas* and its derived visualizations, activated to represent custom statistical mouse-tracking details on the page usage. It uses the mouse-tracking information captured during webpage usage sessions, visually coding the events as symbolic related icons. To achieve visual feedback, all events generated by the pointing device are represented over the page reference map as an additional information layer. Subsequent events are connected with lines and the events are represented on the same position the pointing cursor was positioned at the time the event occurred, Figure 57. Different pointing device events are

coded using the different icons shapes. This type of information is very important to determine if a specific page offers the same interaction as predicted by the designer.

3.2.1.7. Interaction Workspace or Visual Workspace Coherence



Figure 58 *Interaction Workspace* visualization [image source: [Nunes2006], Figura III-19]

Interaction Workspace is yet another visualization derived from *Page Areas*. Even if all particular areas of pages are useful for analysis, a different type of feedback can be extracted from the composition of the visual interaction workspace of a website. The visualization represents link elements in their natural position so that all link elements from all pages are represented over a base image produced from the combination of the area of all pages of the website. The additional information layer is produced by combining all link elements shapes and positions, using color coding techniques. Other page elements can be represented as needed; however, the mixture of several element types might produce confuse visualizations.

Figure 58 presents an example of an interaction workspace that uses link elements and blue color-coding scale. In this figure, two intense interaction zones can be observed: the top and bottom areas of the representation. These areas suggest that they contain more interaction elements than all other areas of the visual workspace. The intensity of top area highlights the idea of the presence of a menu, even if no details of the analyzed pages are presented.

The same concept can be used to represent one or several sessions and determine usage patterns based on the visual insight provided by the visualization.

The rectangle on the top of the representation is the viewport of the user – given by the resolution of its display. Lower interactive areas suggest that the user has to scroll down several pages to reach the desired information, which might be annoying for the users.

3.2.2. Visualization methods for website structure and session analysis

3.2.2.1. Site Pages

Site Pages visualization allows the identification of website pages or sessions, using an iconic visual form that represents the page thumbnail as the areas and the snapshot image of the page. All pages are represented on a grid (bi-dimensional matrix), and the representation provides several types of filters. In addition, several sorting and ordering criteria can be applied to organize the representation; e.g. use the number of visits as a criteria to display most visited pages first. Figure 59 presents such a visual representation of all website pages.

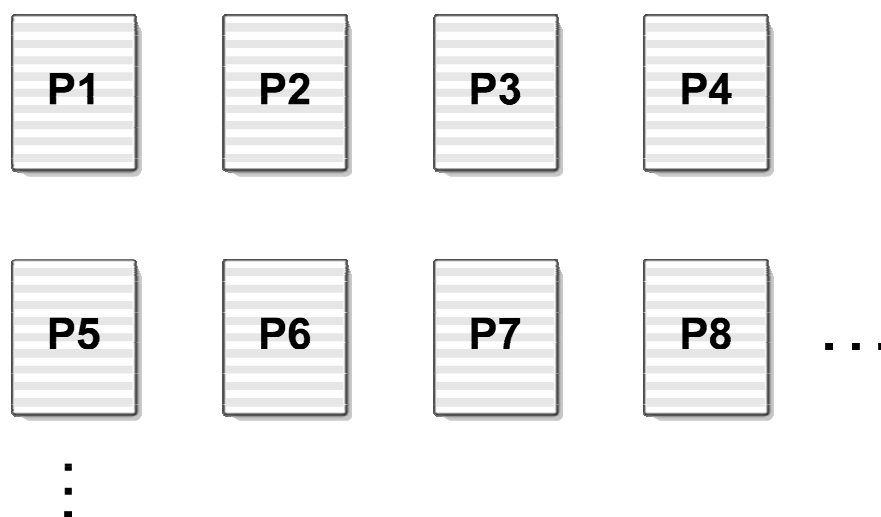


Figure 59 *Site Pages* visualization [image source: [Nunes2006], Figura III-21]

3.2.2.2. Site Structure 2D

Site Structure 2D visualization is meant to represent a holistic view of the website. Generally, a website contains an *entrance page* (usually called *home page*) that contains link elements allowing a subset of the website pages. Each page on this subset contains link elements that provide the linkage to another subset of pages, and so on successively. This type of organization allows the definition of *site levels*: the first level contains the main page, the 2nd level contains the pages accessible from the first page, the 3rd level contains the pages accessible from all pages on the 2nd level, and so on. It is similar to the representations presented in [Dodge2003], [Ricca2001], [Martin2001]. This visualization uses structural attributes of the page and link elements to represent the website.

This type of organization of a website is similar to a representation of a multi-dimensional tree, where each page is accessible via the hyperlinks that connect link elements to the corresponding pages, called child pages. The page on the first level is the *root* page. The representation guarantees that each page is accessible via its unique parent link.

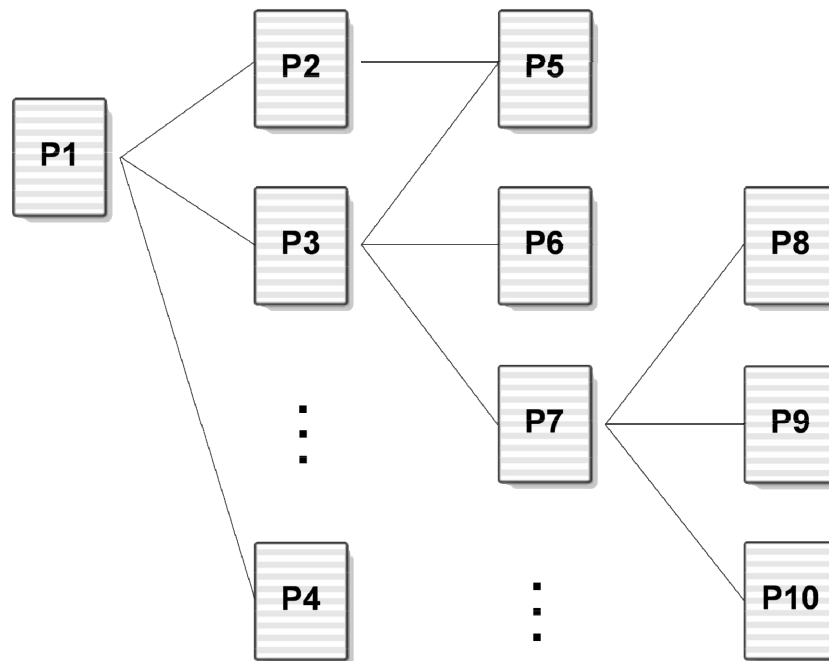


Figure 60 Site Structure 2D visualization [image source: [Nunes2006], Figura III-22]

Moreover, each page can be accessed alternatively, according to the amount of pages that contain link elements that refer to it. Thus, on larger sites, if a page is reachable from several distinct pages, it can be represented several times and in several levels; such a representation can easily turn unreadable or confuse, due to the large number of pages and interconnections of a site. To correct these aspects, a page should be represented only once on the lowest level it first appears (the shortest distance from the root to the page). Other references to the same page can be represented only as backward connectors to the represented page. The connectors can be represented as flowcharts, on top of each other, to avoid visual load on the visualization. However, this aspect might introduce occlusion that can be overcome if flexible interactive connectors and filtering options are used.

Figure 60 presents a possible visual representation of a website that contains several levels. Web pages are represented using iconic thumbnails determined by the occupied area of the page.

Link connectivity and statistical usage of linkage elements can be highlighted or represented as needed, driven by rollover events or other type of interactions.

3.2.2.3. Site Structure 3D (Holistic 3D view)

Site Structure 3D (Holistic 3D View) visualization uses the same concepts as *Site Structure 2D*, without the restriction for the pages to be represented only once, given the fact that a three dimensional space can easily represent more objects. The presence of many connectors might produce occlusion while the 3D projection can introduce distortion. Several implementations of the visualization are possible: spherical, conical, disk or hierarchical being some of the most important.

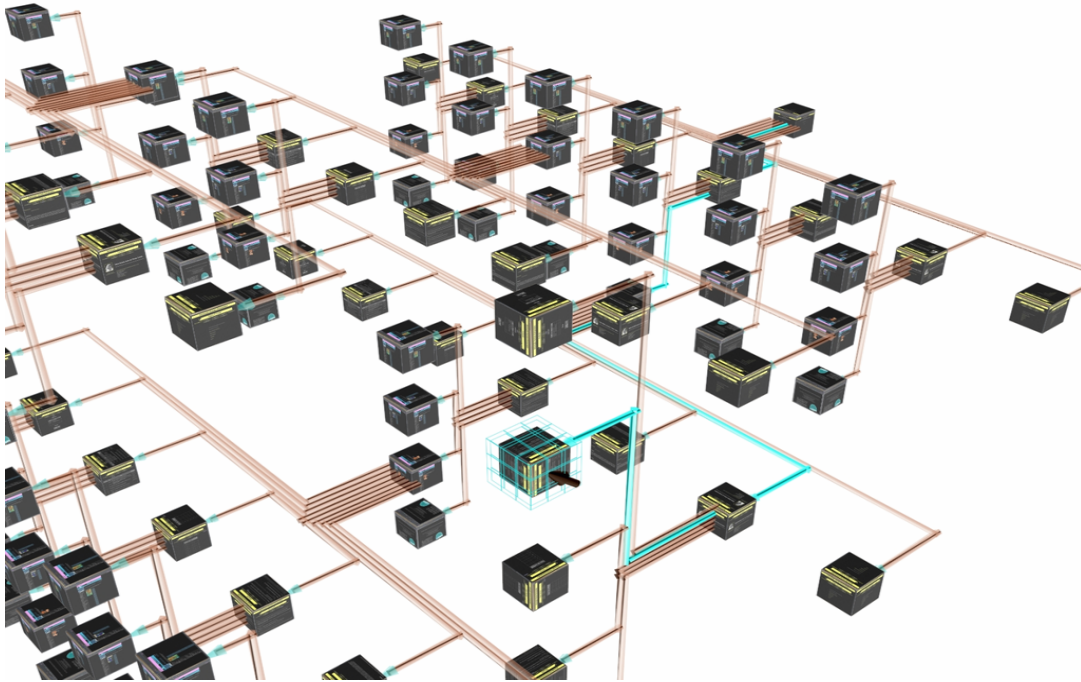


Figure 61 Site Structure 3D (Complex Hierarchical 3D View) visualization

We selected for our test implementation only the *Hierarchical 3D View* that considers the pages in subsequent site levels represented perpendicular to each other (Figure 62). This organization is suitable to represent large numbers of highly interconnected pages using as less space as possible. However, occlusion and poor visual perception are the main disadvantages of this visualization. Figure 61 presents an example of a complex hierarchical 3D view while Figure 62 presents a less complex site structure. Link connectivity and statistical usage of linkage elements can be coded into the visualization or can be represented as needed, driven by a rollover event.

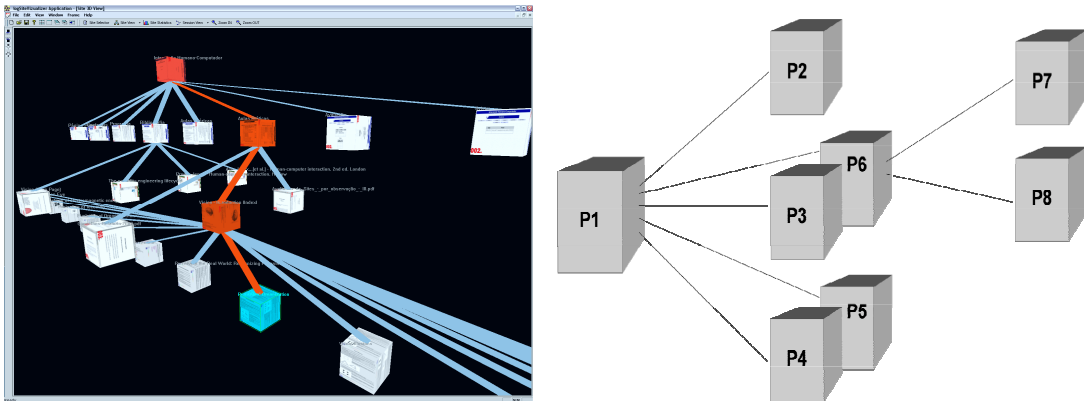


Figure 62 *Site Structure 3D (Holistic 3D View)* visualization on the left, the conceptualized version on the right [image source: [Nunes2006], Figura III-25]

3.2.2.4. Linkage Elements

Linkage Elements is also a complementary visualization for most of the site and session visualizations. It uses several structural and statistical attributes of page and link elements to overlay information on the hovered connector between two pages. It is usually driven by a rollover event, highlights the source and destination of connector, as presented in Figure 63, and overlays a complementary tooltip window.

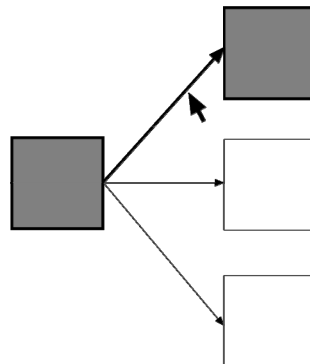


Figure 63 *Linkage Elements* visualization [image source: [Nunes2006], Figura III-29]

3.2.2.5. Path to Goal

Most websites provide several alternative paths to visit the same page, some naturally created, others introduced to improve navigation experience. It is common for different users to use different paths, accordingly to the context, the visibility of some linkage elements, information availability (the time spent for the page to load from the server), etc. It is important for a navigational website designer to have feedback on how distinct paths for the same page are used.

Path to Goal visualization represent all possible connections between two specified pages. This visualization uses structural attributes of page and link elements to identify and represent all possible connections of two dynamically selected pages.

First selected page is considered the start point and the second is the goal page (the end of navigation). All possible connections between the two selected pages are represented on a grid, sorted by the number of page jumps required for a user to get from the start page to the goal page, as shown in Figure 64.

Additional sorting methods can be provided to determine the optimum path. The minimum number of page jumps does not always mean that the considered path is the optimum solution for the navigation. Some times, the lack of feedback, visibility problems of linkage elements, or availability might influence the decision of the users and they can abort the path. This type of anomalies are very important to be detected and corrected.

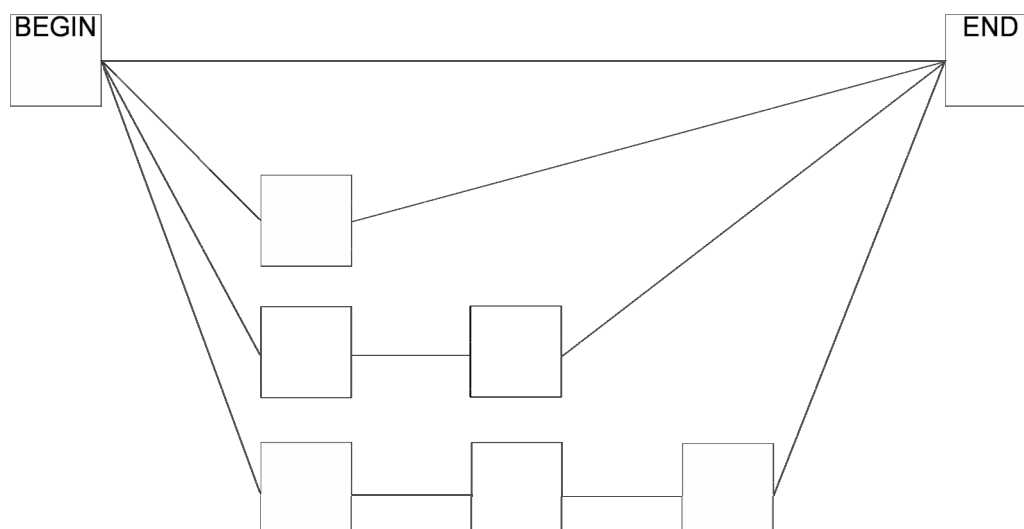


Figure 64 Path to Goal visualization [image source: [Nunes2006], Figura III-31]

3.2.2.6. Session History

Session History visualization displays the pages visited by the user during a session, considering the visiting order. The visualization uses structural attributes of page and link elements to represent the connections between the visited pages and the additional statistical information associated to each subsequent page jumps of the session. Visually, the pages are represented using linked thumbnails, the link elements selected by the user being highlighted for subsequent pages, as presented in Figure 65. The timings for each visit can be coded as the distance between subsequent page jumps.



Figure 65 Session History visualization

3.2.2.7. Session History 3D

Session History 3D visualization is similar to the two-dimensional version with the representation in a three dimensional space of the whole page screenshots, not only the thumbnails. Timing information is coded as distances on Z-axis. This scheme is suitable to represent additional statistical information for each represented page but the perception of details in a three dimensional space might be negatively influenced by occlusion or distortion. If transparency is used, the representation turns even less perceivable. Figure 66 presents a simple conceptual representation of this visualization.

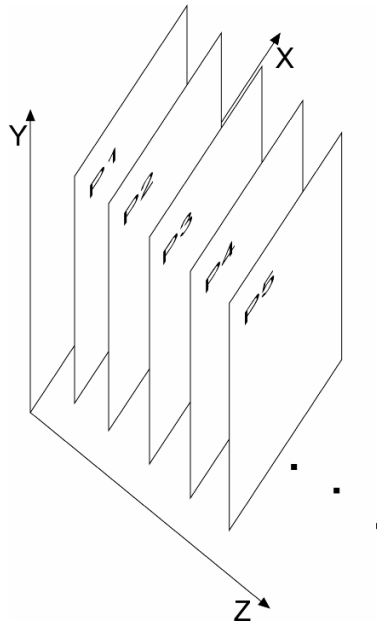


Figure 66 *Session History 3D* visualization [source: [Nunes2006], Figura III-35]

3.2.2.8. Tree-Structured Traversing in Time

Tree-Structured Traversing in Time visualization combines the representations of the website structure and of the session to obtain a temporal visual representation of a user session. This visualization maps each visited page to its corresponding level on the website structure; the levels are represented as rows in a tabular structure (Figure 67: left side of the figure shows site levels as rows). The timings between subsequent pages visits are coded using distance ($\Delta t_1 - \Delta t_n$ in Figure 67), each visit being represented as a new column in the overall table. Additional information is coded using color and shape, to represent unique page visits like the beginning and the end of the session, or to highlight discontinuous visits: when the user goes out of the context of the website domain or selects a jump from the browser history. The visualization can use either small page thumbnails or coded icons to represent the pages. Figure 67 presents the coded icons version of the visualization and a legend is included on the bottom side of the figure.

This visualization is suitable for detecting atypical jumps inside or outside the site structure: a jump back over several levels might highlight an inconsistency on the site

navigation, the user being unable to find the desired page (as in the example of Figure 67, where the visit from fifth to sixth page indicates a jump back over three site levels).

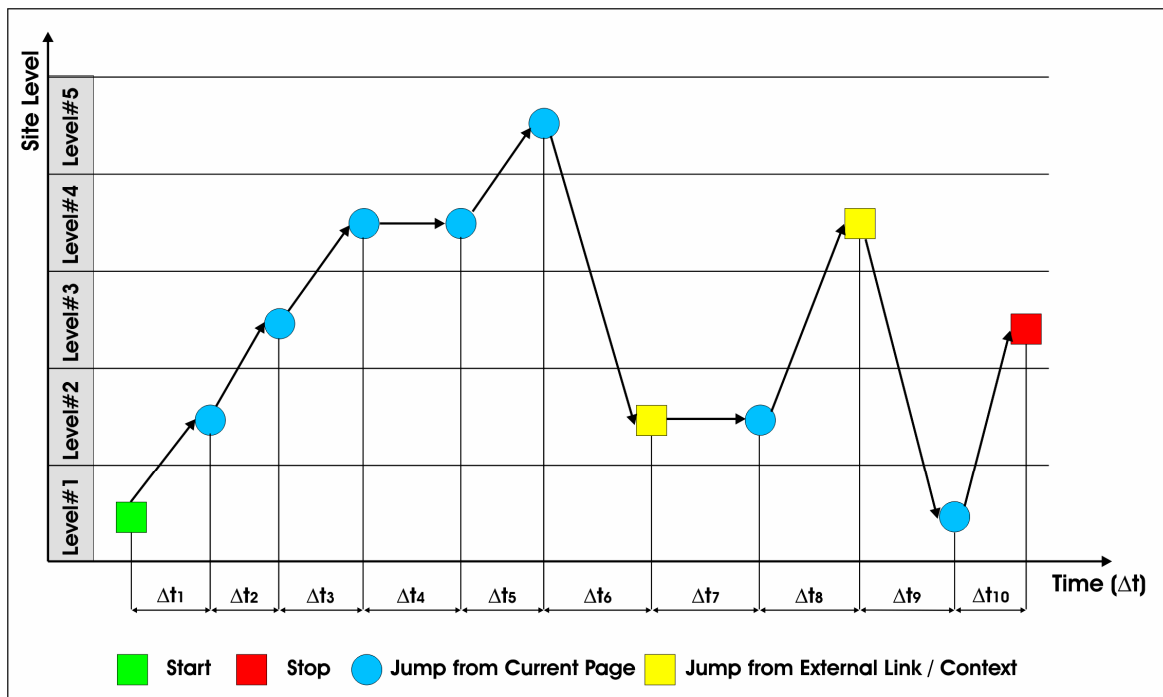


Figure 67 Tree-Structured Traversing in Time visualization

3.2.2.9. Page Linkage

Page Linkage visualization combines page and link element attributes to represent all possible connections for a dynamic set of selected pages. Several thumbnails are displayed for each page in the selected set. The thumbnails are coded icons represented in a two dimensional space and the linkage information is represented as oriented connectors. The connections between pages are obtained based on the URI of the page the link element belongs to, as well as the URI of the page that is referenced by the link element. For each pair of attributes a connector is represented, having the direction from the specified page to the referenced page. Additional structural and statistical information is coded based on the number of times the webpage was visited and the number of times this link element was clicked during a specific time period, to represent the number of visits, as shown in Figure 68.

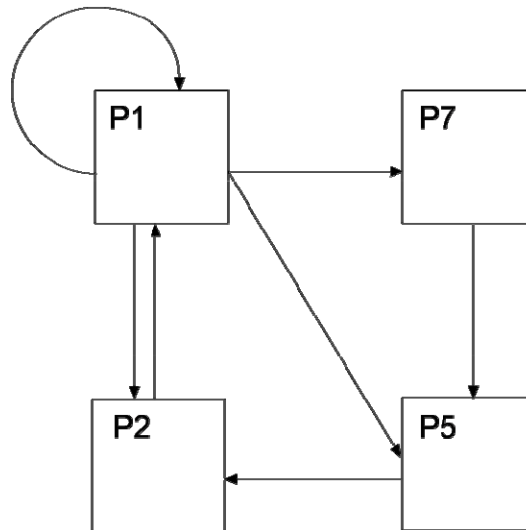


Figure 68 Page Linkage visualization [source: [Nunes2006], Figura III-38]

3.3. Visual correlations of visualization methods

A unique visualization method allows its users to observe a specific process, phenomenon or information. However, there are several situations when a visualization method does not provide all the information required to observe and understand the meaning of the represented information. To overcome these situations, several visualization methods can be used simultaneously, allowing several simultaneous views or observation techniques of the same problem on the same representation screen. Thus, it is vital to produce a unified user interface (UI) that provides the visualization methods required to represent the information in a manner that facilitates the extraction of insight. However, for the process to be effectively efficient, a key aspect is that the visualizations should be interconnected. Considering these aspects, the UI must be synchronously integrated and should instantly propagate the interactions on the active visualization method to all others.

To achieve these objectives, we considered some fundamental aspects inspired by a set of principles introduced by Ben Shneiderman in [Shneiderman1998]:

- I. A visual correlation of visualization methods and inspection mechanisms is required. All visualization methods must implement a similar user interface, sharing the same interaction paradigm and results presentation;
- II. It is important to synchronize the visualization methods based on global or local specific objectives (e.g. show only the sessions filtered for a specific path to goal visualization);
- III. Similar color coding tables should be used on various visualization methods. The same color scheme must be used to classify various objects with the same meaning, similar for color maps;

- IV.** Prioritize the visual context and reduce the cognitive effort using one active visualization at a time, from a variety of available synchronized visualization methods. Visualization methods visible at the same time have to be tiled and tightly coupled to give the impression of a unique representation, even if several are visible. Moreover, the visualizations have to be synchronized and promote the results of user interactions synchronously and independently;
- V.** Reduce the necessity of short-time memory usage, allowing simultaneously observation of multiple tiled views that do not overlap each other. Permanent visibility of simultaneously technique reduces the usage of short-time memory required for the cases when a representation has to be replaced by another while changing the window;
- VI.** Allow the user customize the number of visible visualizations and the positioning/organization of the visual layout, from the entire set available;
- VII.** Screen size and resolution might condition the amount of information to be simultaneously visualized. In addition, the number of screens is also a factor to consider when using several simultaneous representations. More screens of better resolution can improve the visibility by allowing more information to be represented or organized accordingly.

In the context of this thesis, the system integrates various visualization methods, dialogue and feedback information units into one unified interface, as shown in Figure 69. In some situations, the interface depicts simultaneous visual representations with different but complementary information inspection methods. Some retrieve qualitative visual information and others quantitative statistical and structural information.

An inspection exercise starts with the specification of a goal that can consider as starting page the site home page or some other and ends with the specification of a final page where the information is located and is specified as a usability evaluation goal. An articulated and tightly coupled set of views are generated for deeper inspection analysis of inter-page related statistics and interface design parameters. All these schemes are triggered and redrawn by the same event and use the same raw information space to represent their visual features.

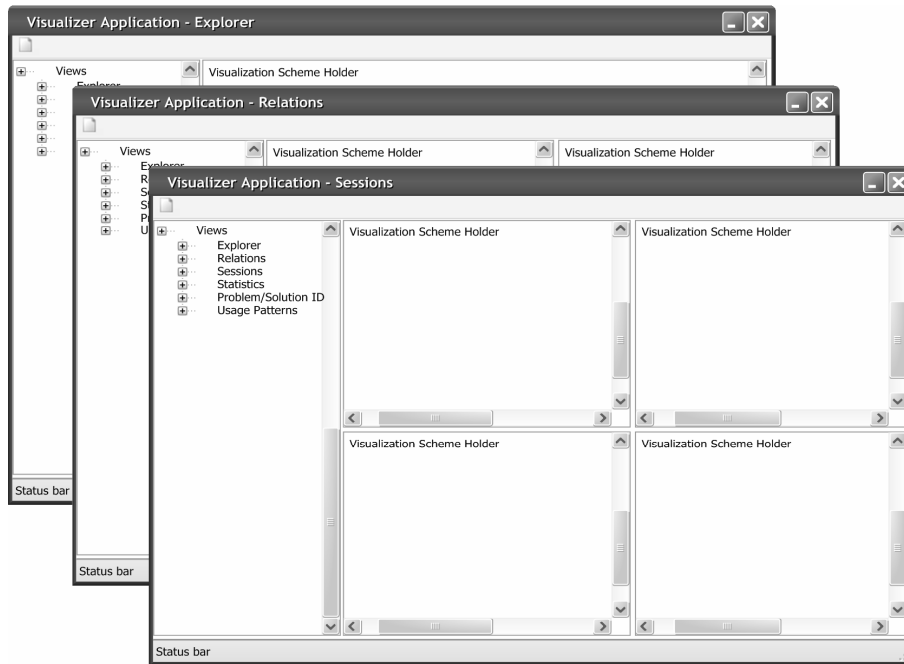


Figure 69 Tightly coupled synchronized views concept

[Andrews1999] presents a case study using a tightly coupled multiple window visual inspection method with 2D and 3D representations. Multiple view windows are fundamental for a coherent inspection of the information, but the context is lost in some situations when the view window is switched. Interface design strategies in this situation should guarantee a coherent context with all relevant multiple views present at one time.

Chapter 4. The Prototype: Objectives and Implementation

This chapter describes the general objectives and implementation details of the prototype used to demonstrate and eventually validate the visualizations and interaction mechanisms proposed by this work. The first section gives an overview of the general objectives and development lifecycle, while the following sections describe the actual system architecture and implementations details. Visualizer is of particular interest since it describes the core application of the proposed visualization system.

4.1. General objectives and system overview

This section presents the objectives of the system we proposed, some key functionalities, specific goals, as well as a system overview.

As a result of a prior study concerning the goals that our application should meet and the corresponding functionality to be included [Nunes2003], [Zamfir2004], [Mealha2004], [Santos2004], we have identified as central to our prototyped analysis framework the following generic functionalities:

- analyze the web site structure and catalog each page in terms of content and structural information;
- analyze the log files of the specified site, which includes several steps as filtering, identification of users, sessions, statistical information, etc.;
- analyze and interpret the information collected from controlled experiments, as mouse tracking, eye tracking, etc.;
- represent the information regarding the site structure;
- represent the usage information, session information and other additional information;
- represent statistical information collected from the log files and controlled experiments;
- visualize the presented information easier to perceive and interpret;
- analyze and process the structural and logged information, using predefined patterns and algorithms, in order to identify usability problems;
- offer a flexible and easy-to-use user interface, in order to allow an easy manipulation of the information visualizations and user interactions.

Several other contributions and commercial applications have been analyzed in order to identify possible issues that this type of systems might involve: [Cockburn1997], [Card2001], [Chi1998], [Chi1999], [Chi2002], [Chen2004], [Cooley2003], [Drott1998], [Faraday2000], [Fraternali2003], [Heer2002], [Ivory2002], [Ivory2002], [Keim2001], [Najork2001], [Niu2003], [Paganelli2002], [Ricca2000], [Ricca2001], [Ruffo2004], [Spiliopoulou2000], [AWStats2005], [ClickTracks2005], [Deep2005], [Ethnio2005], [FastStats2005], [GoogleAnalytics2005], [ISAServer2004], [IWEBTRACK2005], [LiveSTATS2005], [Opentracker2005], [Sawmill2005], [Webtrends2005], and [Wusage2005]. These contributions are detailed in section 2.6 – Applications, page 48, of chapter State of the Art.

We classified the goals of our application in three different major areas of interest:

- *Site Representation* and exploration;
- *Session Representation* and exploration;
- *Statistical Information*.

The following sections present the goals identified for each of the previous areas:

I. *Site Representation* goals:

- view/manipulate the site structure;
- identify/manipulate site levels;
- identify/manipulate a page/group of pages/areas of the site;
- identify/manipulate links (all possible links) between pages/areas of the site;
- identify/manipulate semantic content classification for the site;
- identify/manipulate statistical information about the site: extended attributes as the number of times the webpage was visited during a specific time period, the average time users spent to analyze the text element, the average time users spent to analyze the link element, the average time users spent to analyze the graphical element, etc.;
- identify/manipulate site classification according to a specific criterion;
- identify page-level statistical information and generate possible representations like: site visual layout coherence, etc.

II. *Session Representation* goals:

- view/manipulate the session representation;
- identify/manipulate a page/group of pages/areas of the session;
- identify/manipulate session areas according to a specific goal;

- identify/manipulate the additional information of a session like eye or mouse tracking;
 - identify/manipulate the statistical information mapped to a usage session;
 - identify/manipulate a group of session, compare and represent multiple sessions;
 - synchronize the session(s) information and highlight the session(s) on the site context;
 - extract and manipulate user patterns from the session information and classify the information;
 - identify and highlight usability problems by processing the additional information about the session;
 - identify page-level statistical information based on session information and generate possible representations like: visual workspace coherence, etc.
- III. *Statistical Information* goals:
- process and represent the statistical information collected from the log files and controlled experiments;
 - allow generic representations for the presented information (e.g. hovering tips visualizations);
 - highlight the information selected in one visualization propagated to different visualization methods;
 - allow reports creation and manipulation.

A system model has been identified based on the functionalities required and on the set of goals defined for each of the previously introduced areas of interest. As an introduction, Figure 70 presents an overview of the system integration, its conceptual components, the roles of each component and the information flows.

The system model is organized to achieve two main goals focused on intranet usage. One is related to the site structure, the other to the interface design (visual workspace) organization. It must provide solutions that help the user detect structural problems revealed by visual inspection of usage patterns with the possibility of having simultaneously, qualitative and quantitative visual inspection methods. Another vital area where severe system problems occur is at user interface level. The visual workspace where the user takes her/his navigational decisions is another concern and the usage pattern at this level is represented and analyzed.

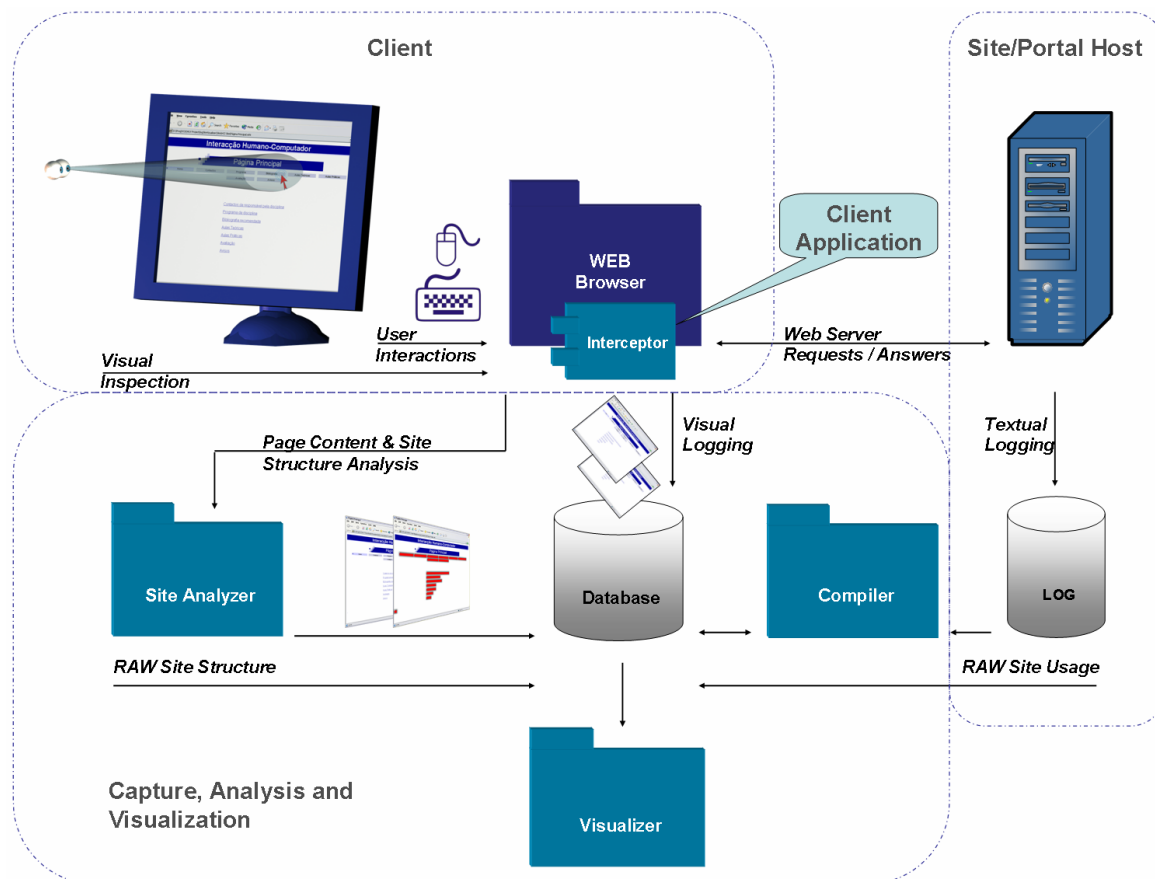


Figure 70 Simplified model of the system

Four main modules have been identified for the implementation of our system:

- I. *Site Analyzer* - meant to analyze the site in terms of structure and interconnections, to capture and classify the content of each page, to make a screen content capture of each page, to identify and classify the hotspot areas of each page and to store the resulting information in our database;
- II. *Interceptor* - meant to intercept the site user events at the client side (mouse, keyboard and eye-tracking events), in a controlled experiment, to make a screen content capture for every context change (when a link element is clicked) and to send the collected information to our server using a protected connection;
- III. *Compiler* - meant to analyze the *log files* stored on the site server, to interpret the information collected from these files or controlled experiments. It uses the session identification window to organize the statistical information related to site usage, site users, page content, etc., and then store the resulting information in our database;
- IV. *Visualizer* - meant to present the overall processed information in a way that is more adequate to the human perception and understanding. This module uses a set of visualization methods specialized on showing different types of information to the user, having different/complementary goals. It gives the user the possibility to choose

different visualization methods for displaying the same information and switch among different representations. It does not modify our primary processed data; it only accesses it and transforms numerical and textual data into visual representations.

The analysis process of a web site is a resource and time-consuming operation, and different steps are required to complete the entire process. The phases required to analyze a website, starting with the data collection and finalizing with the visualization and problem-solution identification, as illustrated in Figure 71, are:

1. Firstly, use the *Site Analyzer* to analyze the structure and content of the site, and to catalog each page. This is a complex and time consuming process, involving automated and semi-automated tasks. As examples consider, on one hand the attributes page, link element, text element and graphics element dynamism type if a specific subset of page elements is personalized, the crawler might not have the possibility to explore the personalized contents; on the other hand, the automated semantic classification of page contents might vary according to the personalization level or might involve a sophisticated mining process;
2. define a time period to monitor the site usage, or perform a controlled experiment, with the final goal to collect live information from the site usage and use the *Interceptor*, considering the possibility to organize the collected information directly on the analysis system database;
3. use the *Compiler* to analyze the natural website usage logged information (*log files*) plus controlled experiments logged information, and to identify the mappings of the collected information throughout the website structure and statistical meters of the website usage;
4. finally, use the *Visualizer* to load the information stored on the database, to create specific visual representations of the selected information, to analyze and identify possible problems (usage behaviors, design layout, conceptual and implementation issues).

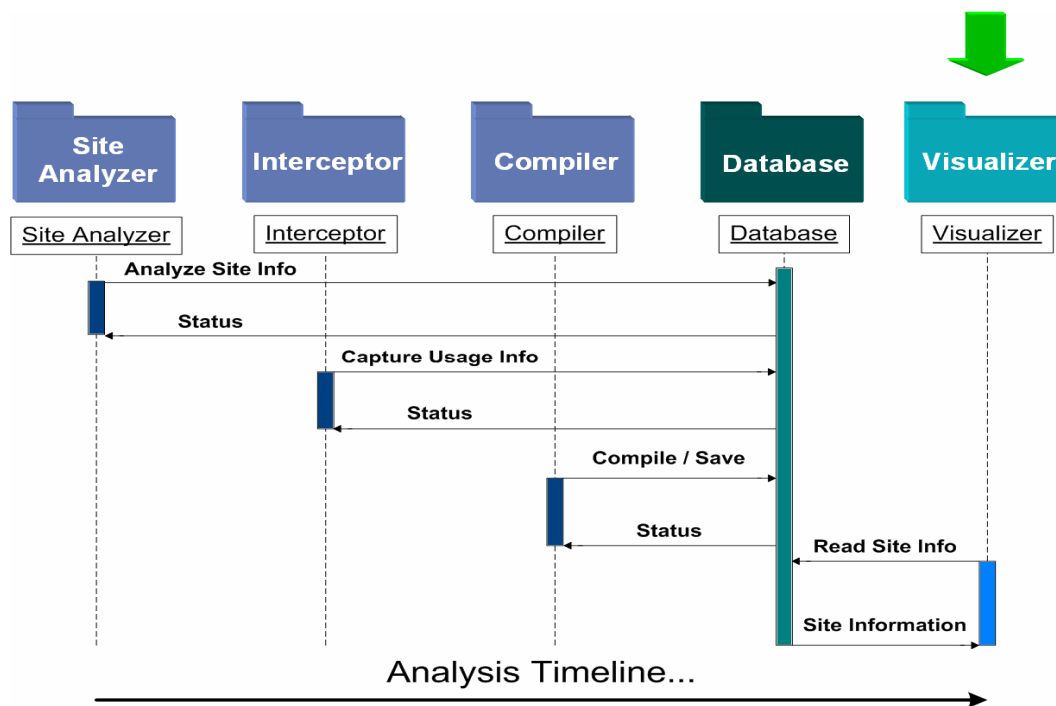


Figure 71 Analysis timeline

Starting from this general structural organization, we implemented a prototype meant to validate some of the concepts previously presented.

A first approach was to merge the *Compiler* and the *Visualizer* with the purpose of analyzing the *log files* at runtime. This was possible since the logs used to test the prototype had a small size. Thus, no *Interceptor* application was implemented, even if our data structures considered the data structures required by the *Interceptor*.

Finally, the resulting prototype used a *relational database* to store the information and implemented all four components as proposed on the system model (two of them only partially).

This prototype considered only the representation of a website as presented in Figure 72, which shows a logical hierarchical representation of a website. For crawling purposes, we considered a website that contains web pages, each page organized hierarchically, in a tree-based structure, having a set of attributes and containing a set of objects. The objects contained by a page are web-based specific objects, starting from text, images, tables, borders, hotspots (hyperlinks), etc.

Each page is classified in a unique site *level*, defined as the shortest path, from the entrance point of the site, to that specific page, following a breadth-first algorithm [Najork2001]. This way we can uniquely identify the shortest path from the entrance point in the site to a specific page. This identification is useful to validate the concept of *goal-oriented exploration*, meaning that a user explores the site, having a specific goal (target),

and the shortest path shows one possible optimal (or non-optimal) solution for the user to reach the desired page/area.

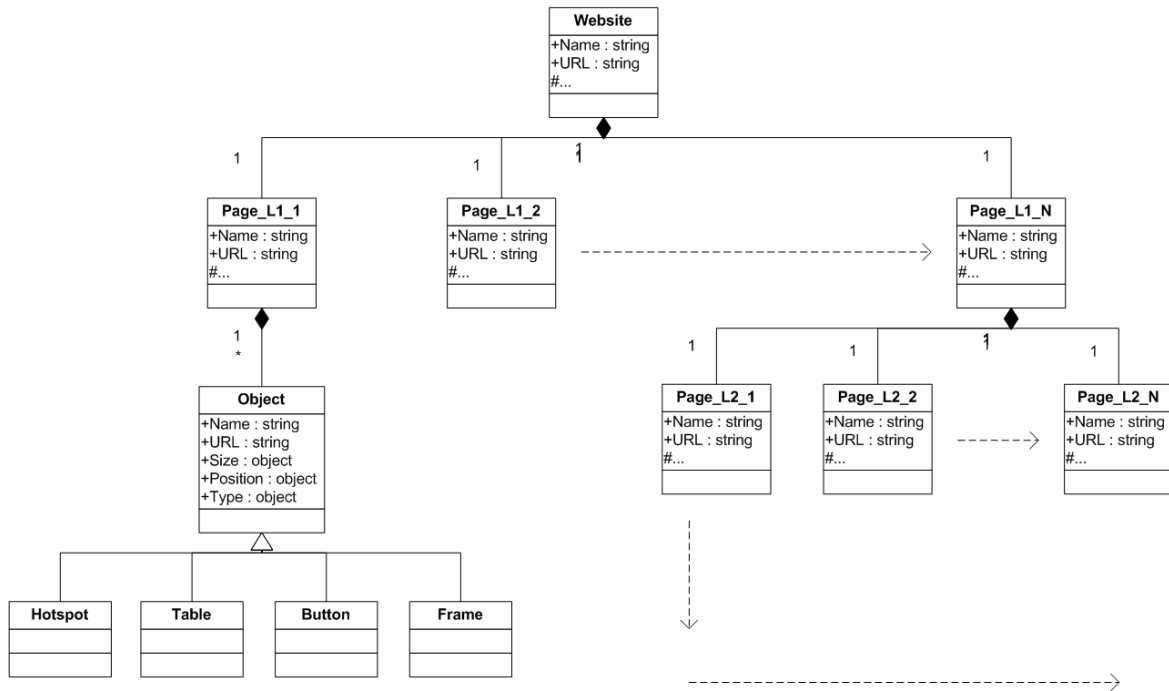


Figure 72 Website hierarchical representation

4.2. The Prototype Implementation

For the entire system and modules, the development strategy was tightly related with the predefined system goals. We followed the development process accordingly to the priority we assigned to some specific tasks chosen for implementation and test. Given the complexity of the concepts involved, we decided to firstly implement only the modules that we considered of greater relevance to the results of our test system.

On a first stage, we decided to identify a subset of functionalities to be implemented, from the entire set of functionalities proposed in the previous section, afterwards we selected the visualization methods, identified and specified their proposed functionalities, interface manipulations and user interactions to be implemented. Starting from these premises, we identified twelve visualization methods or combinations to be implemented.

A preliminary prototype was implemented to test a site analysis and capturing application, the session identification algorithms as well as to validate some of the two-dimensional versus three-dimensional visualizations. The final prototype was a more sophisticated application that focused more on the visualizations, and less on the data collection.

4.2.1. Background

A preliminary version of the prototype was implemented, to address some basic tasks like visualization of site structure using both two and three dimensional representations, user interface paradigm, interaction mechanisms information identification and navigation in 3D representations versus 2D, etc. The following paragraphs briefly describe the results obtained with this preliminary implementation.

The identification of pages and navigation within three-dimensional space turned to be a difficult and demanding task. Most of the users did not guessed that the combination of mouse buttons and movement would have the effects of panning and camera rotation. Almost none of them intended to use arrow keys to perform the navigation. These key combinations turned to be unknown for the users, even if they implemented the main navigation on most of the 3D games available at that time.

Yet another difficult task turned to be the hovering. We noticed that if we asked users to discover hovered information in the beginning of the experiment, they would have failed to do it. Instead, if we gave them some time to accommodate to each visualization, they would have performed the task very well.

Another demanding task for the users was to arrange the multiple views to be simultaneously observed. Navigation on a MDI – Multiple Document Interface turned to be confusing if no predefined layouts are available.

All these aspects and some others gave us valuable feedback to consider in the final implementation of the prototype:

- use well-known key combinations to perform well-known or similar actions;
- use predefined windows layouts and dispositions to facilitate navigation;
- switch to an unified interface with tightly coupled visual inspection methods instead of using MDI;
- remove 3D implementations of visualizations as they turned to be very confusing for the users;
- use navigation tools similar to well-known image processing applications;
- make the application icons more intuitive, etc.

4.2.2. Prototype evolution strategy

4.2.2.1. Conceptual phases

The development of the prototype focused on the conceptualization and implementation of a flexible application, based on the experience obtained from the preliminary

implementation, although, we decided to implement everything from scratch, even if the process was more complex. Some of the main phases involved to:

1. analyze and classify the results of the evaluation of the preliminary prototype;
2. model a new user interface *framework* and application *engine*;
3. conceptualize the newly introduced visualization methods and refine the standard ones;
4. create and evaluate visualization mockups and test them to obtained feedback before implementing;
5. create the adequate model for the relational database structure used to store the information;
6. implement and test a application *framework* that provide generic functionality;
7. implement the database on a real DBMS (Database Management Systems) and fill it with real data collected from the analysis of a test website;
8. implement user interface conceptual model and define interaction strategies with the application;
9. define the synchronization mechanisms between simultaneously represented visualizations;
10. make use of a client-server-like architecture for the visualizations, the visualization server;
11. evaluate preliminary designs and mockups and obtain insight;
12. implement all modules required to make the prototype functional;
13. conceptualize and conduct formal and informal evaluations of the prototype and analyze of the results; and
14. implement application refinements based on evaluation results.

The decision to implement everything from scratch allowed us to improve some aspects regarding the data structures, models of user interface, visualization methods and proposed interaction paradigm.

4.2.2.2. Implementation challenges

We divided newly introduced concepts and improvements in four classes of major tasks for the development cycle:

- A.** general application architecture:
 - create a more adequate layered application architecture;
 - detach the user interface from the visualization schemes implementation;

- implement the *Visualization Server* to work as a message dispatcher and server-like component for all the different visualization schemes;

B. user interface:

- switch to an unified interface with tightly coupled visual inspection methods, implemented as a multiple top-level document interface, easy to manipulate and configure;
- improve the way of interaction with the represented information, using specialized tools as standardized interaction models (similar to the tools present on image manipulation applications, etc.);
- allow an easier manipulation of the interface views through a specific window-based framework and predefined layouts for windows disposition;

C. relational database:

- identify all possible elementary attributes and their dependencies for the website pages, page contents and interconnections;
- find an adequate data representation, optimized for a relational representation as a relational database;

D. visualization methods, information manipulation, perception and overall integration:

- identify the most suitable visualization methods to be integrated in the new version of the prototype;
- make the represented information more legible within the visualizations;
- classify the perception mechanisms and identify new ones that allow a better understanding of the specific information representation;
- define the overall integration and synchronization of the visualization methods used.

Being aware of all these aspects, we were able to refine the architecture of our system as presented in the following paragraphs. The following sections describe the technologies we used to implement the prototype, the system architecture, and give an overview of the prototype and a detailed description of the implementation.

4.2.3. Application architecture and technologies used

The application has been developed for Windows Platform and as development tools we used Microsoft Visual Studio .NET and Microsoft SQL Server for our relational database.

We decided to use a layered architecture [MSDN2004], which allows a greater independency from the development platform and the operating system, as presented in Figure 73. The *Business Layer (Framework)* represents the link between the

Development Platform and the Presentation Layer (PL), which includes all mechanisms for user interaction.

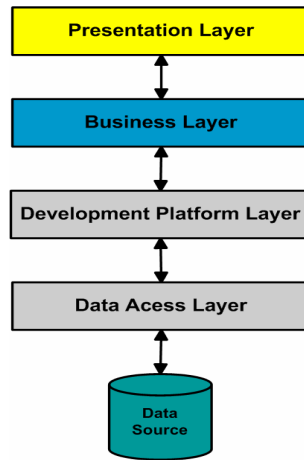


Figure 73 Layered System Architecture

We followed a top down approach for the development process and used UML (Unified Modeling Language) [Booch1998] and [Jacobson1999] to represent the architectural models of our application.

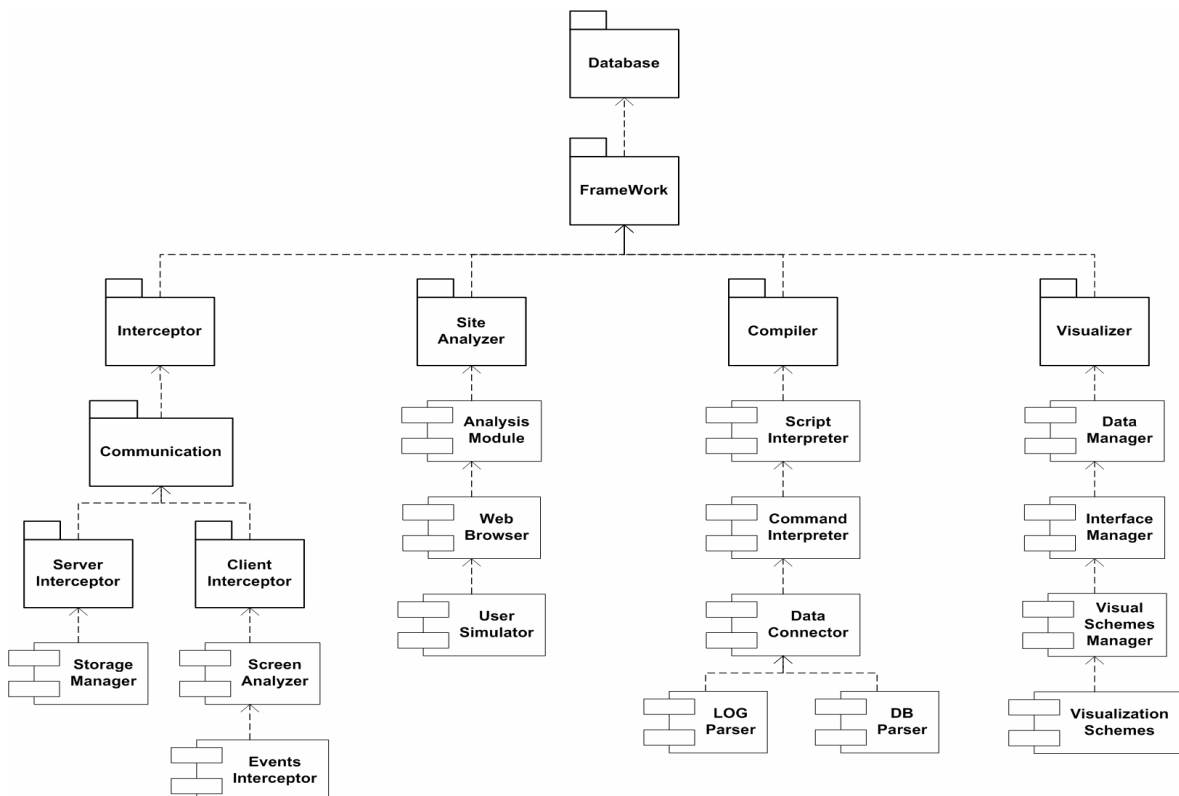


Figure 74 System Simplified Architecture

A simplified model of the application is presented in Figure 74; this model integrates a relational database for storing the information, a framework used as an intermediate layer

and the four main components of the application. As we can observe in the figure, the four main components are distinct entities, independent and interdependent only via the information stored on the database.

The four components of the application are as mentioned before:

- *Site Analyzer*;
- *Interceptor*;
- *Compiler*; and
- *Visualizer*.

The *Compiler* was developed based on the implementation of a preliminary prototype, being responsible of organizing the statistical website usage information resulted from the identification and analysis of user sessions. *Compiler*, *Interceptor* and *Visualizer* were developed using Visual C++ MFC (Microsoft Foundation Classes) programming language, for Windows platform. *Site Analyzer* was developed using Visual Basic and C++, respectively Visual C# on a Microsoft .NET development platform.

Specific details on the database model, its entities, relations, and the model used for the development of the application framework can be found in Annex

2. Database model and application framework.

The following sections present the main components of our application, their functionalities and programming models.

4.3. Site Analyzer

Two versions of the application have been developed. The former focused on collecting the website related information, the latter on the dynamic reverse engineer and manipulation of portal structures implemented using Microsoft SharePoint Portal Server.

The main functionality of the first version of *Site Analyzer* is to analyze the website by analyzing its page interconnections, the content of each page (e.g. identify page objects, hyperlinks, object position, the three sets of page elements: text, graphical and link elements), to make screenshots of every page and to save this organized information in the database.

The analysis process cannot be fully automated for all types of existing technologies; for specific technologies, it is a semi-automated process, involving a human to take the major decisions and to help the correct identification of pages, areas, etc.

It is possible to organize the website in different areas of interest, accordingly to the content of specific areas, a task that can be performed only manually in this implementation.

The extraction of information for the analysis of a website involves the classification of the information on three classes: contents, structure and usage, as proposed by Cooley [Cooley2003]. *Site Analyzer* performs the extraction of the first two classes:

- Contents - the real data in the web pages, that is, the data the web page was designed to convey to the users. This usually consists of, but is not limited to, text and graphics;
- Structure - data that describes the organization of the content. Intra-page structure information includes the arrangement of various HTML or XML tags within a given page. The principal kind of inter-page structure information is hyperlinks connecting one page to another.

For the first implementation of the application, we used a browser tool that allows the retrieval of each page of the website and gives access to the collections of objects from within the page. Once we access the hyperlinks collection of the web page, we can use a breadth-first algorithm [Najork2001] to retrieve all connected pages. Once the page is retrieved from the server, it can be parsed and all content can be identified and classified accordingly. This process is called crawling and the more complex the targeted websites are, the more complex the crawling process is. Personalization and dynamic page

contents can be considered as demanding challenges for the analysis process to succeed.

As mentioned, the analysis of a website can be a time-consuming process, involving a human at a decisional level. Large quantities of information are to be catalogued and organized to reflect the site structure and interconnection. One of the major problems while analyzing large websites is that the analysis process cannot be synchronized with the site evolution, meaning that specific parts of the site can be dynamically modified before the analysis process finishes. To resolve this problem, the information collected should be versioned using timestamps, and the overall information should be synchronized with the correspondent version. This option provides the possibility to track the evolution of the site and understand it.

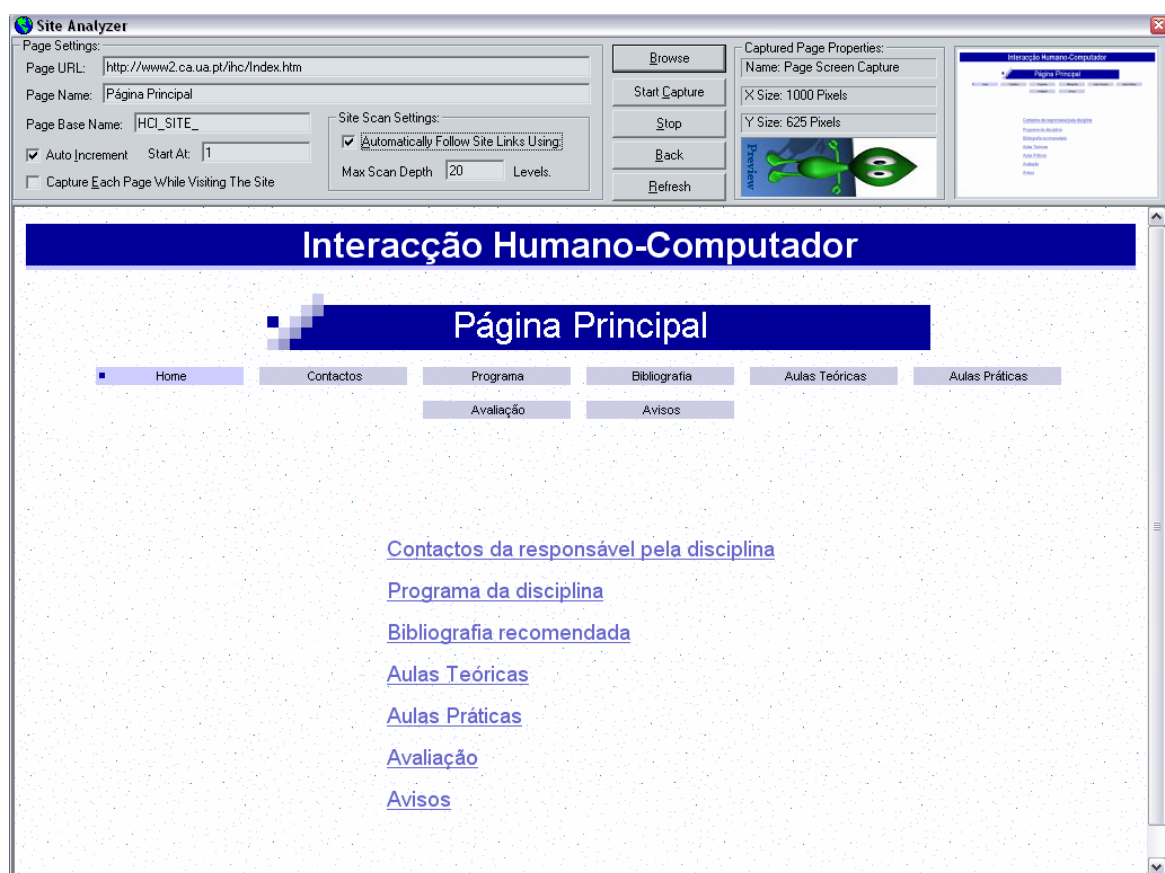


Figure 75 Main view of the *Site Analyzer* prototype

Figure 75 presents a screenshot of the *Site Analyzer* application. Besides the option of automated/semi-automated crawling and analysis of website structure and each page contents, the application creates a screen-shot of each page, using a predefined image resolution (mainly the width is important) unique for the entire process of website analysis.

At the time of analysis, the most common resolution of the monitors was 1024x768 pixels; supposing that the web browser is maximized and excluding the margins of the window,

the result is a viewable area of a width of the real page equal to 1000 pixels. We considered this specific width for each analyzed page of the analyzed sites we have captured.

A second version of the *Site Analyzer* was implemented and focused on additional functionalities as well as the integration with Microsoft SharePoint Portal Server 2003 and Windows SharePoint Services, a portal technology, flexible, easily deployable, having a reduced cost, and offering an efficient way to organize the institutional information spaces.

SharePoint Designer and Analyzer is the implementation of an application that provides a way to visually design or reverse engineer, implement and maintain an intranet *Portal* structure. We decided to use SharePoint since the technology provides a very high level of accessibility and integration. It is flexible enough to let us programmatically redesign UI layouts or site structure, since it provides atomic components and dynamic contents management.

Besides the integration with SharePoint, the *SharePoint Designer and Analyzer* is highly flexible and might permit integration for other types of modular and non-modular website technologies to be reverse engineered, redesigned and updated. The core of the application uses Xml to define the contents, independently of the technology used to implement the website.

The application provides the reverse engineer functionality for a SharePoint portal that uses a breadth-first algorithm [Najork2001] to retrieve all connected structures and substructures, but the main purpose is the portal structure and contents design, considering all aspects that define a portal: sites and sub-sites, areas and sub-areas, pages, page contents, personalization, access control, etc.

A simple scenario for our case study is to reverse engineer a portal, analyze its usage information using *Visualizer* and at the end redesign or update the structure and/or contents based on the analysis results. Figure 76 provides an example of the organization of an educational portal that includes several sites and sub-sites. In a detailed view, each site is composed of several pages and each page of one or several content elements.

A more detailed description of the implementation is presented in the Annexes included on the CD support that accompanies this dissertation.

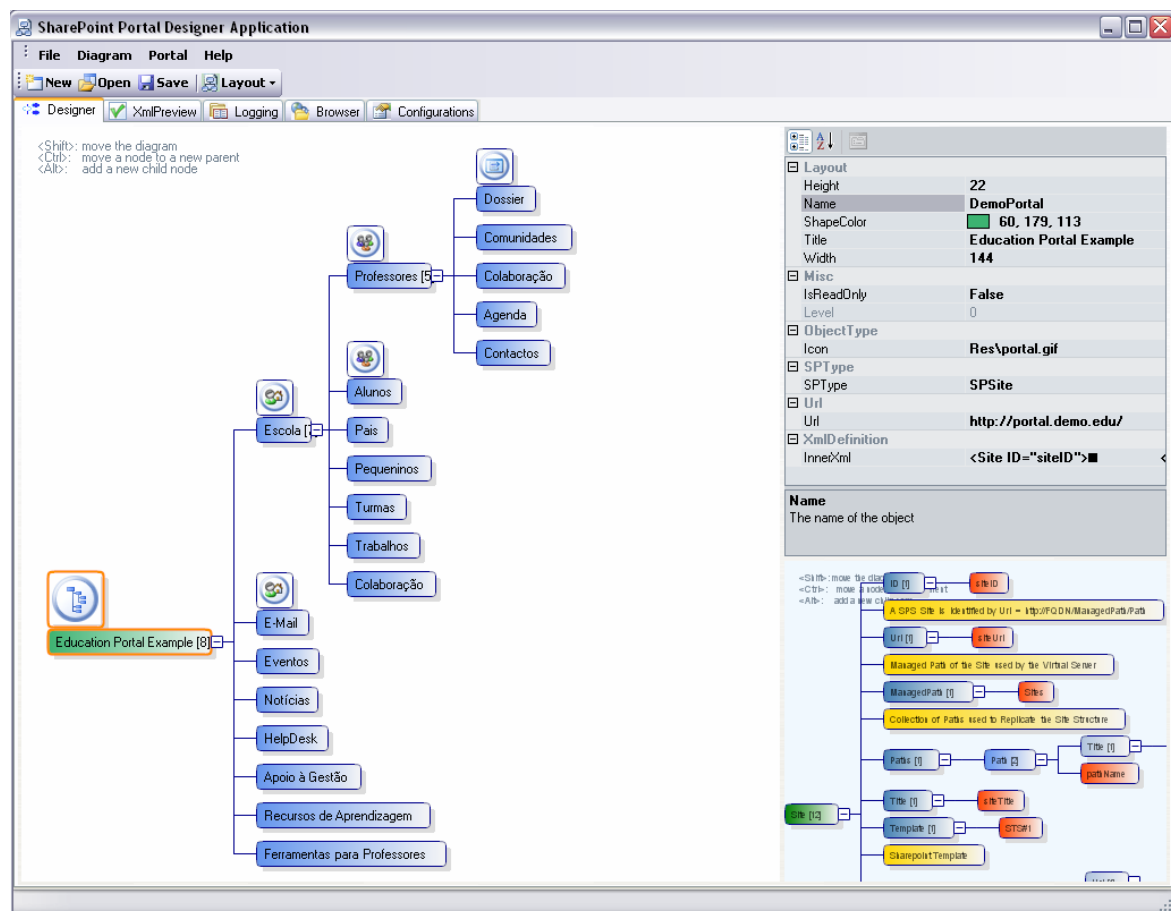


Figure 76 Portal architecture example

4.4. Compiler

The main role of the *Compiler* is to analyze the *log files* resulted from natural site usage or controlled experiments (files stored on the web server side or application database). It is meant to identify the statistical site usage data, the user sessions, and to convert this data in more organized and processed information to be used by the *Visualizer*. This process is time consuming and fully automated.

As introduced in section 2.2 – Usage Information, there are several steps applied to the analysis of application-server log files to determine usage patterns: information gathering, filtering and classification are the most important. Mostly quantitative measures of website interactions and, in some cases, some qualitative measures and semantic contents classification can be extracted by automated analysis systems. Log files analysis is a preliminary step on every web analysis system and many authors proposed similar approaches for the analysis of the log files: [Cooley2003], [Card2001], [Paganelli2002], [Heer2002], [Fraternali2003], [Spiliopoulou2000], [Niu2003], and [Youssefi2003], as previously mentioned.

In the scenario of this work, the compiler application needs to perform several tasks:

- load the site structure and construct its hierarchical on memory map;
- parse application runtime log files, filter the information and detect sessions and the session identification;
- parse live experiments logs (if offline logs are used instead of database) and save the information directly to the database since sessions are already defined in live experiments logs;
- identify referrers if not present, considering subsequent requests of the same session as referrers;
- identify and map session requests to site structure, based on *Url* and *Query* parameters of each request, for both runtime and live experiments logs;
- save session information, for both runtime and live experiments logs;
- identify unique users and user groups;
- identify usage information for all pages based on the identified mapped requests Url, for both runtime and live experiments logs;
- identify usage information for each link element in A_L set based on the referrer information of each request, for both runtime and live experiments logs;
- save usage information;

During the development of the application, a special algorithm had to be developed for detecting a specific session on a log file. The algorithm for detecting a session is reflecting exactly the concept of session – a time window where a specific user is making requests to the web server.

Figure 77 shows a screenshot of the main UI of the compiler application.

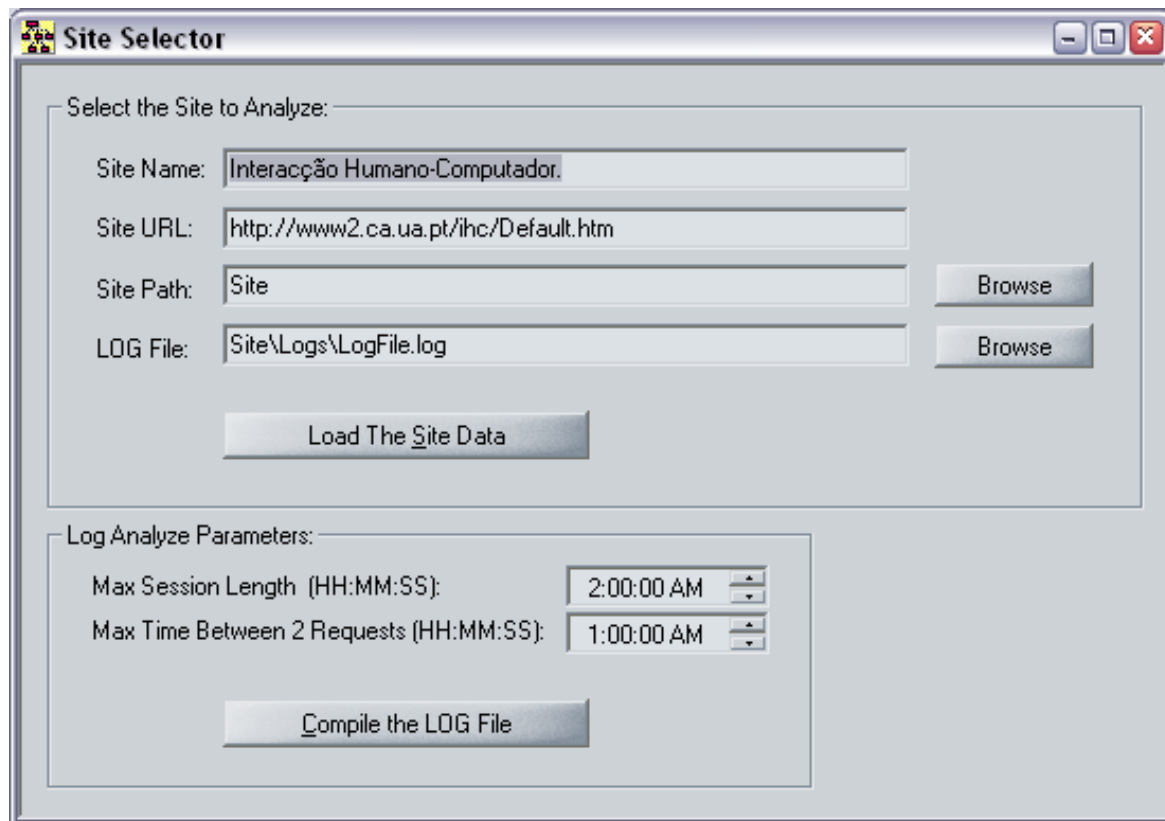


Figure 77 Compiler prototype - user interface

Sessions can be uniquely identified by the following set of parameters:

- SessionID – the unique identifier for the session, numeric format;
- SiteID – a pointer to an object that uniquely identifies a web site;
- UserID – a pointer to an object that uniquely identifies a web site user;
- Cookie – a unique string used by the browser to identify a session;
- StartURL – the starting point URL of the session;
- EndURL – the URL where the session ends;
- StartTime – the date-time identification for the session beginning;
- EndTime – the date-time identification for the end of the session;
- Length – the session length (time interval);
- Requests – a pointer to the array of requests that have been made during the session;
- ReferrerURL – the URL of the session referrer.

However, if some information is missing (in some cases, not all the previously presented fields are logged), sessions can be identified using only the following information:

- user name;
- user IP;

- computer name;
- cookie;
- referrer.

To define the session length, the following constraints are used:

- maximum time between two consecutive requests should be less than 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;
- maximum time for the session window should be less than $\Delta t_k \in \Delta T_{j,i}$ = the time periods between subsequent requests $t_{k,y}$ of a specific client's time window i .

Session identification varies from one logging file format used to another, the most common being Extended Common Log Format – ECLF (as defined by W3C – World Wide Web Consortium). We selected ECLF for the current implementation of the compiler, even if the file format can be changed by adding a new module. The algorithm for detecting user sessions based on log files in ECLF is discussed in Annex 1. Session detection algorithm.

4.5. Interceptor

The *Interceptor* allows the interception of user interactions with the analyzed website, during controlled usage experiments, and the creation of logging information based on these interactions. Interception and logging of mouse, keyboard and eye-tracking events is its primary task (depending on the requirements and additional hardware present in the laboratory). The second task is to make a screen content capture for every context change (when a link element is clicked). The information can be stored locally until the experimental session ends and then directly sent to a specific server for later processing.

One of the main reasons to use an interception system was to help us correctly identify the beginning and the end of a specific usage session, as well as to be able to log all additional usage events that occur at the client side. Most application servers that use common log formats do not include additional information on clients requests, but only information regarding the requested documents.

A possible scenario of usage is to install the *Interceptor* on a test machine among the hardware and software for eye tracking, mouse tracking and key logging. Pre-configure the server connection credentials and the websites to be monitored, then perform usage sessions with a test user at a time. A sequence of tasks performed during a unique usage scenario/session can be:

- initialize interception component when the client browses a page on the monitored site domain:

- initialize interception component, server connections or logging system, etc;
- initialize a new client session by assigning a new session identification;
- log the client computer identification, client name, etc.;
- for each page requested by the client:
 - initialize browser information as position, window size, scroll position, etc.;
 - access the page contents collections;
 - identify the link element clicked by the client to request a new page and store the information;
 - make a screenshot of the entire client screen and one for the page itself;
 - synchronize the eye-tracking information with the page elements visualized by the user;
- log mouse, keyboard and eye-tracking events;
- store the information on the logging system or directly on the server before a new context changes;
- close the opened session when the interval between two requests overpasses $\Delta t_k \in \Delta T_{j,i}$ = the maximum periods between subsequent requests $t_{k,y}$.

The *Interceptor* is important in the preliminary phase of information gathering and preparation phase. It is of tremendous importance to capture the intermediate events that occur on the client-side between two subsequent requests in order to identify what determined the user's navigational decision. This type of information can be determined by analyzing the interaction of the user with the client browser and eye-tracking devices that can highlight the focus and attention of the client during the website navigation session. Card [Card2001] and Paganelli [Paganelli2002] proposed similar approaches that intercept user events.

A preliminary *Interceptor* is implemented as a *BHO – Browser Helper Object* module for integration within *MSIE – Microsoft Internet Explorer* browser. Implementation details are not of relevance in this context. However, it is important to mention that first version intends to support only mouse tracking, keyboard and eye-tracking being scheduled for a later stage. Currently, the preliminary development version focuses on the interception of context change events, creation of screen-shots of the contexts and storage of the information within a direct connection to the database server.

4.6. Visualizer

The *Visualizer* represents a set of interconnected analysis and visualization schemes in a unified user interface. It is the only component of our system that accesses the database stored information and transforms it into visual representations, perceivable by the human eye and easily understandable.

The following sections describe the modules, relational data structures, the user interface paradigm and implementation, as well as the visualization methods with the associated interactions, correlations and synchronizations used in this application. Note that three visualization terms are used: visualization techniques, visualization methods and visualization schemes. Visualization techniques and visualization methods terms are more related to conceptual visualization; the former focus on ways of generating visualizations, the latter addresses more the resultant visualization. Subsequently, visualization schemes are the real implementations of the concepts. Therefore, this chapter addresses visualization schemes and not methods.

4.6.1. Application layered model

The application uses a layered model with tightly coupled, interdependent and interactive layers. Figure 78 presents the general model of the *Visualizer*. As observed in this figure, two major layers are present: one for data access and another for information presentation.

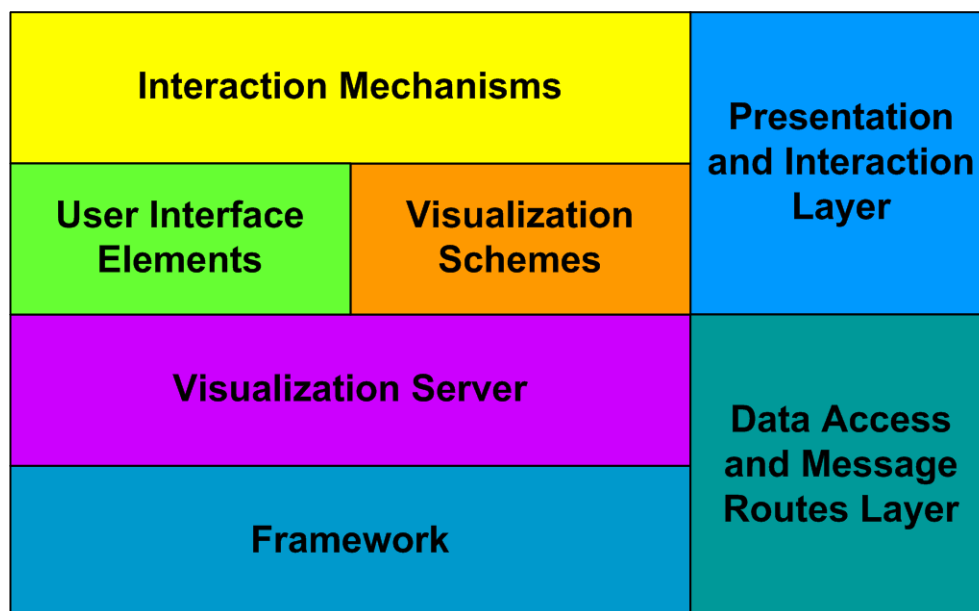


Figure 78 Layered conceptual model of *Visualizer*

The following paragraphs describe the main blocks presented in Figure 78:

1. *Framework* represents the link between the development platform and the application functional objects. It facilitates the access to the information stored in the database by providing a rich set of tools able to interrogate and manipulate the information;
2. *Visualization Server* has the functionality to manage the creation and removal of *visualization schemes*, to manage their information access needs, data communication and synchronizations. This functional block feeds the *visualization schemes* with information, retrieved from the database via framework, and acts as the message dispatcher role between all the connected *visualization schemes*. It is called server since its role is similar to the server functionality in a client-server topology;
3. *User Interface Elements* represents a set of standard or functional enriched user interface controls used to build the user interface of our application. These objects can be windows, menus, toolbars, web browser controls, etc.;
4. *Visualization Schemes* represent a set of functional blocks, each combining a rich set of user interface controls and interaction mechanisms, altogether implementing a specific visualization method or a combination of methods. The *Visualization Schemes* entity is defined as a standalone entity because of its functional and logical aspects. As from the implementation point of view, a *Visualization Scheme* is a user interface element, that can be selected and manipulated whenever and wherever users want to place it, with the condition to be up and running. The *visualization schemes* represent the central purpose of our application and, because of their complexity, we describe each of them in a separate section 4.6.5 – Implementation of visualization methods , page 136;
5. *Interaction Mechanisms* are all possible means of interaction with the application. The user interface interactions are interventions of the application user, meant to reveal, highlight or present a specific set of information. Every interaction induces a set of effects on the application, as a change of the context, aspect or content of the currently active representations.

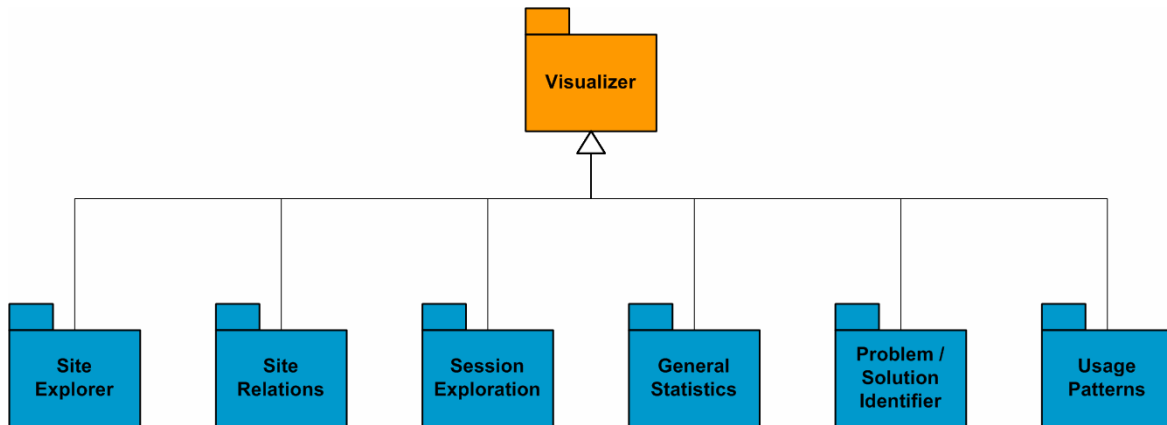


Figure 79 Visualization modules

The Visualizer includes six different modules (Figure 79):

- *Site Explorer* – where the user has the opportunity to explore offline the structure of the selected site. Each page of the site is stored in the database and every time the user selects a link element from a page, the corresponding page is loaded from the database and presented through the browser;
- *Site Relations* presents the information related to the site structure. Several representations of the information can be shown at a time, selected at any moment from the set of visualization methods available. Not only structural but also statistical information is presented, as usage patterns. Different types of interactions with the information are offered here, with the results reflected by the intention of the user to reveal a specific information, or to highlight other, etc. (e.g. showing the possible paths between two selected pages);
- *Sessions Exploration* – the session information can be observed using several representations of the information at a time. In addition, multiple sessions can be observed simultaneously. Overall usage statistical information is represented here. The interaction mechanisms presented here allow the usage of filters for the represented information;
- *General Statistics* is designed to show the statistical information related to the site usage. Different visualization methods can be represented in this area;
- *Problem/Solution Identifier* is designed to show information concerning usability problems related to the site structure (e.g. broken links) and the result of using metrics to assess the compliance with specific usability guidelines (e.g. usage of colors, font size, overall clutter of the pages, etc.). By comparing the actual information with the paradigm associated with the site section, the tool will be able to suggest (or not) a solution for the identified problem(s). Unfortunately, at this stage of development, these functionalities are not implemented;

- *Usage Patterns* is designed to access the information related to user / user-group site usage, highlighting usage patterns related to site navigation, information flows or communication traffic.

Each one of the above modules is designed to cope with the overall design of the Visualizer: synchronization, visibility and perceptibility of the represented information.

Note that given the large amount of concepts involved and the selected implementation goals for the current prototype, only some of the presented groups were implemented, some others being still under development or conceptualization.

4.6.2. User Interface conceptual model

As mentioned earlier, our proposal involves the representation of large amounts of data. Based on our previous expertise, on the results presented in [Munzner1997] , [Dodge2003], [Chi1998], , [Chi1999] [Chi2002], [Niu2003], [Chen2004], [Youssefi2003], [Martin2001] , [Nunes2003], considering the survey of [Benford1999] and taking into account human-computer interaction and information visualization principles [Dix1998], [Spence2001], we have tried to find solutions for effectively representing those great amounts of information. To attain our proposed goals, we have to visually represent the information in a perceivable manner, and offer means of interaction with it. Considering these aspects, the results of the preliminary prototyped user interface, and as a result of prior investigation on UI design and interaction [North2000], [Sebrechts1999], [Wiss1998] we were able to design the user interface for our application as described in the following paragraphs.

The graphical user interface is based on a multiple top-level window engine (Figure 69), which allows the manipulation of data representations, in an easy and practical way. The experience obtained from the preliminary prototype has proved that the manipulation of each different view, as an autonomous entity, is preferable. For a better flexibility, we decided to implement a set of controls (docking windows, splitter windows, etc.) that allow the manipulation of the spatial position and size of the selected visualization representation window.

An important aspect of Visualizer is that the user has the possibility to observe the information, using different representations simultaneously. The synchronization between the representations offers good feedback to the application user, allowing a better observation and inspection of the selected information. The partition of the interface in several areas of interest, as presented in Figure 80 and Figure 69 (section 3.3 – Visual correlations of visualization methods), and the possibility to apply direct spatial manipulations to these areas, give a better visibility to the represented information. The window selection area allows changing the current context, which means to change the active window, each context corresponding to a separate window and a different type of information represented.

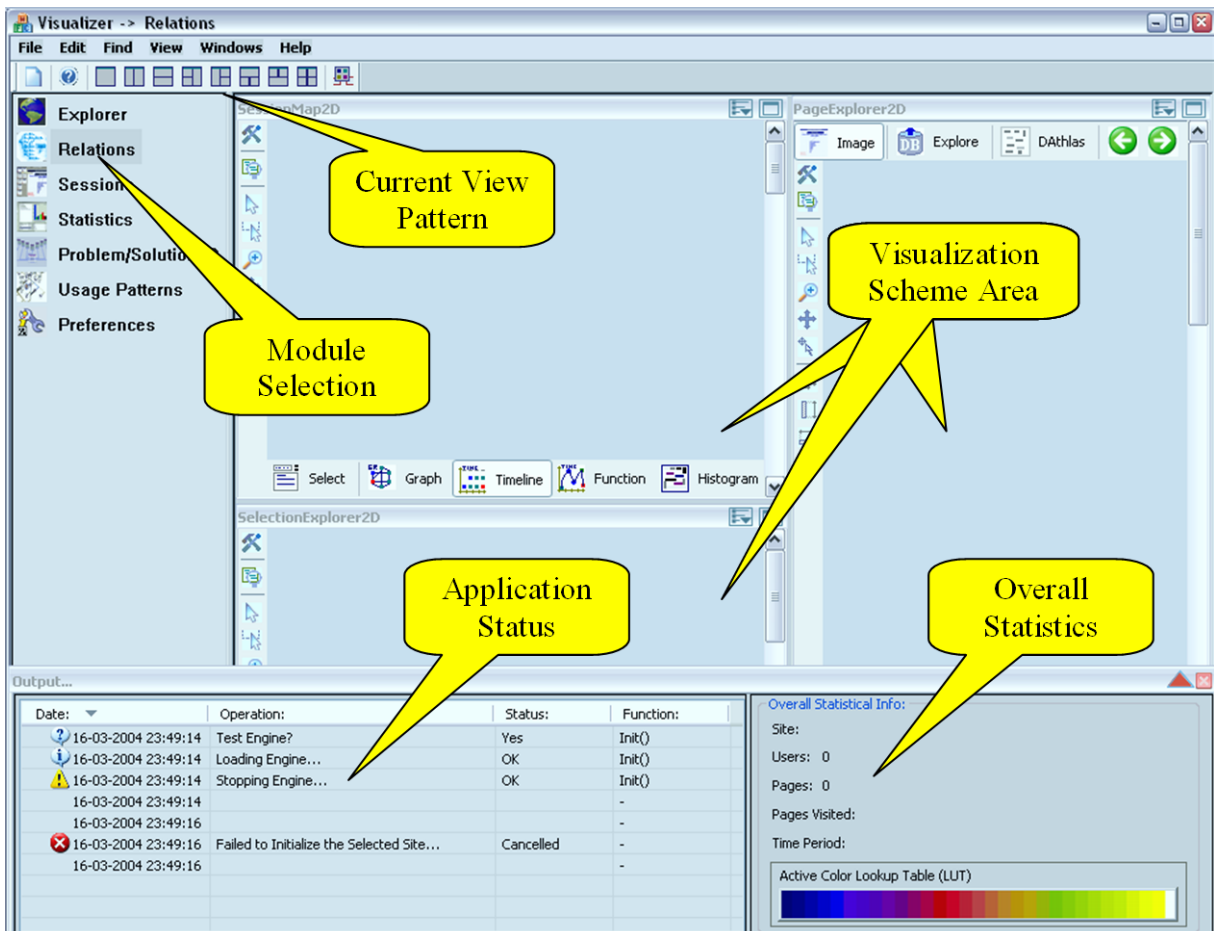


Figure 80 User Interface model

The application status window helps the user to have a better understanding about the actual state of the system (Figure 80– bottom left); it provides information related to the description of the current context, the user interactions with the system and the effects of these actions. In addition, overall statistical information is presented in this window, information regarding the overall site usage, content and analysis period, for the selected site.

Each visual and functional aspect of the application is configurable; the possibility of manipulating the user interface entities, and the interaction among these entities, also contributes to the overall flexibility. For example, each functional element of the UI can be spatially relocated, capable of two or three degrees of freedom, like docking windows (Figure 81 and Figure 82 present some possible manipulation of interface objects).

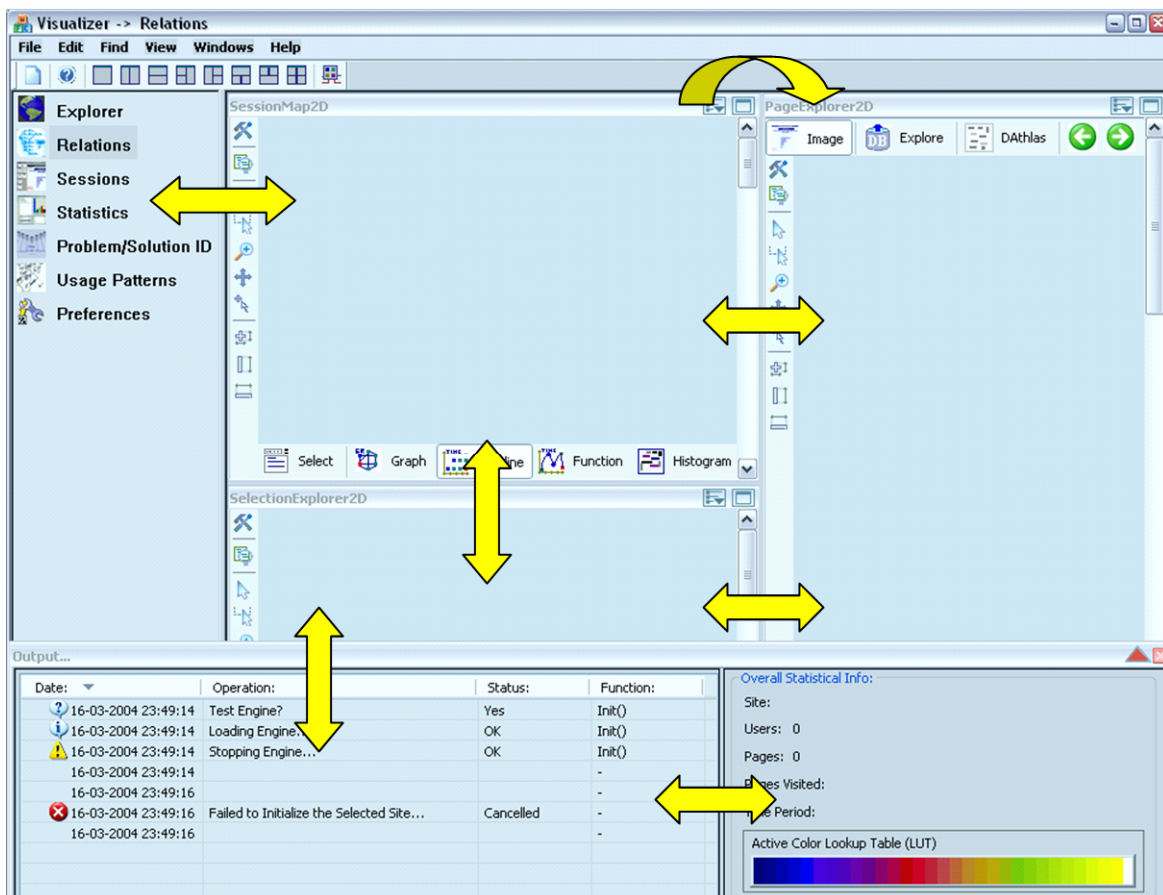


Figure 81 Interface objects flexibility

All functional modules of the Visualizer are designed to fit in the user interface above description. Thus, we can speak about a standardization of the user interface among the Visualizer modules presented in Figure 79.

As a multiple top-level window architecture, the user interface uses specialized views whose functionality defines the basics for its purpose and providing some standard functionality. All of the six modules of the Visualizer inherit the functionality of a specialized view, implementing predefined behavior and appearance. Each specialized view of the interface holds a set of predefined user interface elements, which can be *visualization schemes*, navigation objects, configuration panels, etc. Currently, each one of these views can hold up to four different *visualization schemes* at a time, with the possibility of changing their spatial disposition or interchanging their places. Navigating between these views becomes a task as simple as selecting another view from a navigation panel positioned on the left side of each window.

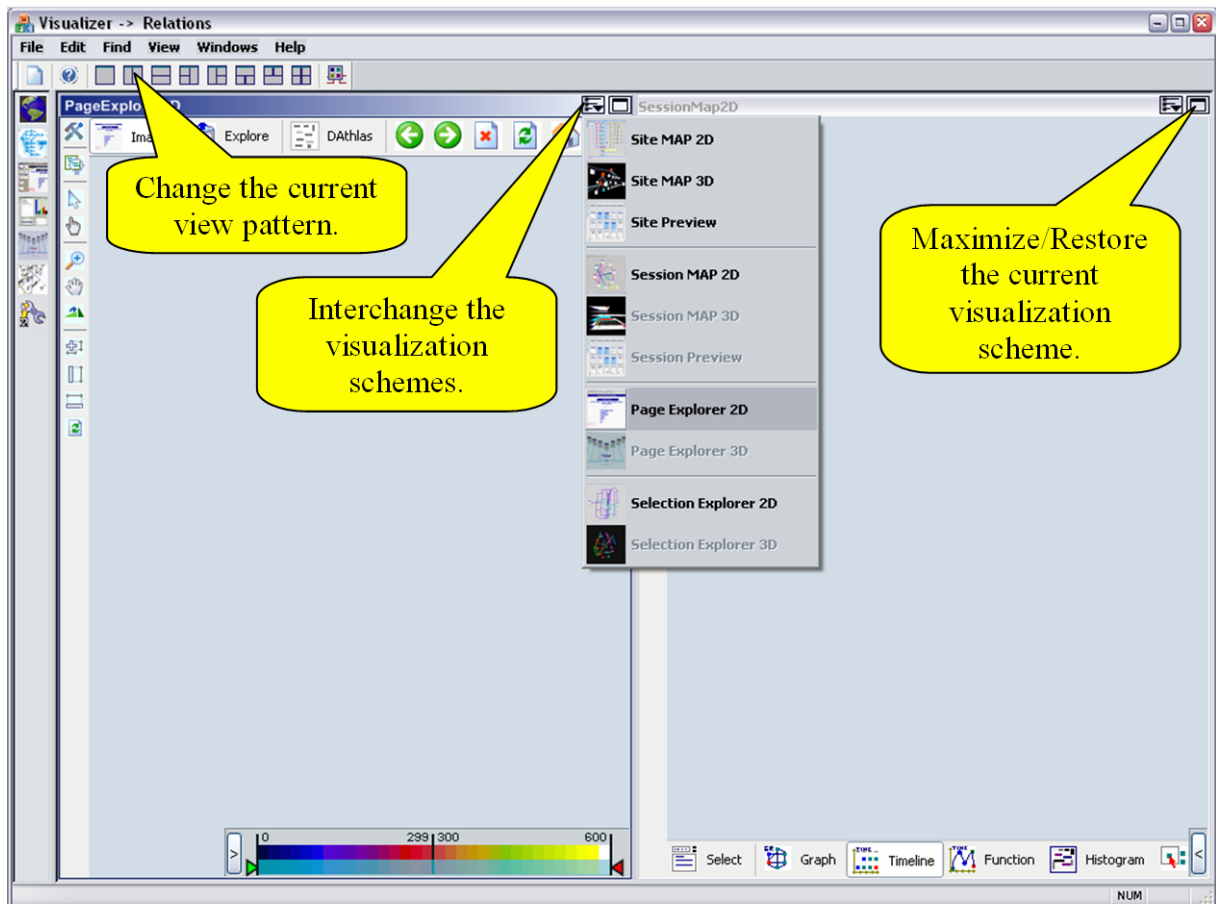


Figure 82 Possible interface manipulations

The main entry point of the application, as well as other specific tasks are implemented using a wizard-based interface [Dahlback1993]. This approach offers the user more intuitive and easier methods for completing a specific set of tasks. Figure 83 presents the four steps of the application initialization wizard to select a website, from left to right, top to bottom.

As part of the UI standardization, for interaction purposes, we can highlight that all *visualization schemes* contain a generic toolbar that offers a unique set of interactions with the information visualized (Figure 84, left half of the figure, left toolbar). It is a generic toolbar for spatial manipulation of the contained objects, usually placed on the left side of the visualization window; additionally, it provides the access to some other standardized functionality as search and properties panels.

In addition, each *visualization scheme* provides a unique set of configurable aspects whose values are stored in the configuration area of the application database, for each application user – personalization (Figure 84, right side of the figure).

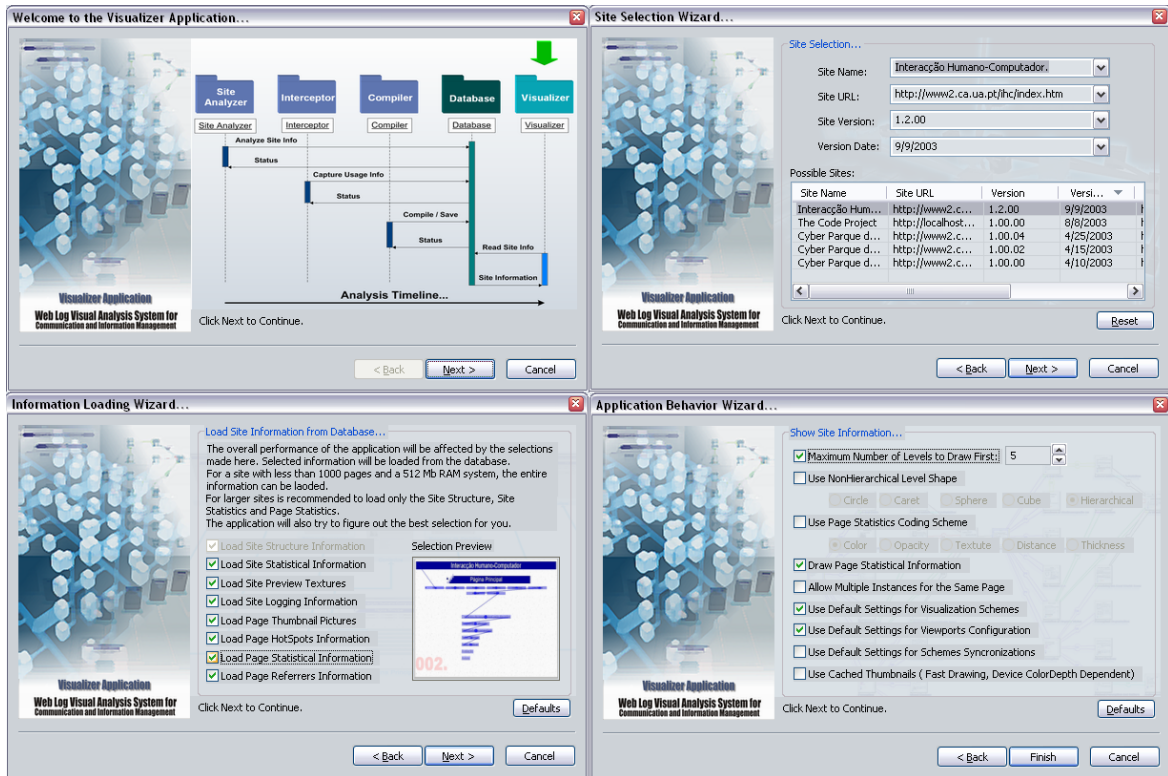


Figure 83 Wizard steps (left to right, top to bottom)

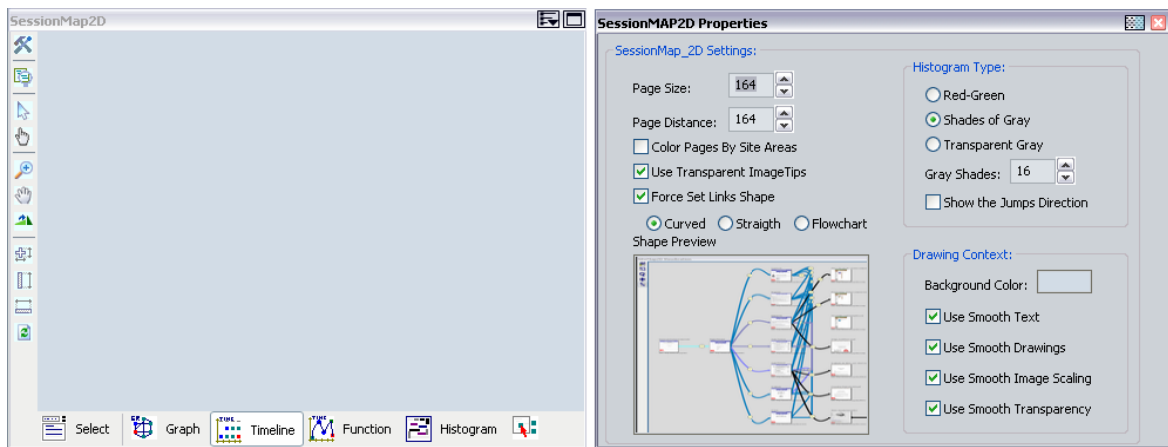


Figure 84 User interface aspects

The application was designed to include a visual logging functionality; it is capable of storing the application context (screenshots and event logging) each time the user changes it. This functionality is designed to allow us observe the usage of the application from the UI perspective, user performed actions, application responses, etc. The system proved its value during the preparation phase of the pre-evaluation sessions we performed with a set of unspecialized users, providing helpful feedback about Visualizer usage behavior and responsiveness.

4.6.3. Relational data structures

The information processed by the Visualizer is organized in several groups as follows:

- application configuration information (user rights and configurations, general application settings, overall application resources, etc.);
- website information (websites, site users, groups, areas, etc.)
- webpage information (content, page elements, statistical information, etc.)
- session information (log request, user event, session event, mouse-track or eye-track event, or other information collected either during natural site usage or controlled experiments)

The data structures used by the application had to be designed and optimized for the manipulation of large quantities of information of a small to medium website. Depending on the size of the website, thousands of web pages might require expensive hardware resources. Considering that for each webpage we have to store a lot of information, we had to be very careful with the memory allocation algorithms, the memory being one of the most important resources we had to deal with. To resolve this issue, we decided that we should load all the information related to the webpage only when necessary and use only the basic information required for the overall representations of the website.

For each webpage, we would have to store:

- the web content of the page as seen in the browser;
- a full size copy of the page and a thumbnail with its screenshot;
- a binary compressed texture for 3D representation of the page;
- the information regarding each object of the page;
- the hotspots of the page and their interconnections;
- the statistical information related to the page usage;
- etc.

A detailed view of all data structures of the application can be found in the Annexes included on the CD support that accompanies this dissertation.

Another important issue we had to resolve is related to the representation of a large amount of objects, in different rendering contexts, each object having or not an associated image thumbnail (we have to remember that for the 3D representations of web pages, specific textures of the page thumbnail are required). This problem was resolved by using the *Visualization Server* that stores all the information needed and shares it with each connected *visualization scheme*. It is described in section 4.6.4.1 – Visualization server.

Part of the relations between the data structures and their conceptual meaning are:

- a website can have a unlimited number of users;

- a website can have a unlimited number of levels, defined as the minimum distance from the entry point of the website to the pages of the level;
- a website can have a unlimited number of site areas, defined as a group of pages with a similar conceptual/semantic meaning and content;
- each level or area of the website is composed of a (un)limited amount of web pages;
- each webpage has one or more referrers defined as parents holding links to the page;
- each webpage can hold one or more link, text or graphical elements, etc.;
- each usage/user session contains a limited number of requests, user events, mouse or eye tracking events.

4.6.4. Implementation building blocks and synchronization details

The development and testing of large software systems, often involving different conceptualization / development teams, is a process that requires a very good communication protocol and notation. The Unified Modeling Language (UML) was created with this purpose, to offer standardized and integrated notation for software modeling/development teams worldwide [Booch1998], [Larman1998]. Following the latest information representation standards and driven by the need of conceptualizing a medium size software system, we were able to create mockups of our system using UML. Because of the complexity involved in such representations and the nature of this document, the following sections present only some simplified representations.

We used object oriented modeling / design (OOM/OOD) patterns to identify the architecture of our application and we highlight the following conceptual and functional distinct building blocks:

- *Data access and manipulation components* (parts of the framework) deal with the database information manipulation, image/texture manipulation, log data manipulation, etc.;
- *Standardized user interface elements* (parts of the framework) are used implement the application main UI and the functional views;
- *Visualization schemes* (parts of Visualizer) visually represent the information and allow its manipulation, ways of interaction and synchronization mechanisms;
- *Event and context synchronization component* (part of Visualizer), represented by the *Visualization Server*, is responsible for the synchronization of various concurrent *visualization schemes* and representing the data source for all of them

All these building blocks are integrated, functionally related and interdependent.

The *standardized user interface elements* are the basics of the user interface, a conceptual model being introduced in Figure 69, page 99 and in section 4.6.2 – User Interface conceptual model, page 124.

Data access and manipulation components represent a set of elements meant to manipulate the information stored in the database. The processed information can be numerical, textual or binary information (e.g. images, binary content of web pages, archived log files, etc.). The most important are: database access, image manipulation, logging and network communication.

Visualization schemes are all visualizations produced by Visualizer, their related data structures, aspects and functionalities. They share common functionalities and are synchronized via the *Visualization Server*. They are registered to the *Visualization Server* each time an instance is created, to share the data and communication channel (used for messages synchronization and information transfer).

The *visualization schemes* are conceptually organized in a hierarchy of eight levels, meant to specialize each scheme accordingly to its purpose and to share as much common functionality as possible from the upper levels, as presented in Figure 85. While the lower levels are responsible for basic window manipulations, the middle level is responsible for the definition of basic shared visualization functionalities and the upper levels are specialized accordingly to each visualization scheme purpose.

Event and context synchronization component (the *Visualizer Server*) is designed to perform the synchronization between the *visualization schemes*; in addition, it uniquely stores one copy of the shared information units. Whenever a *visualization scheme* requires information from the database, this component processes the request and returns the information. The message dispatching role is accomplished as needed, to satisfy the synchronization needs between the *visualization schemes*. This component is one of the most important parts of the application, therefore, it is described in detail in section 4.6.4.1 – Visualization server, page 134;

The application user interacts only with the presentation layer, while the *visualizations* have two types of interactions: one with the application user and one with the *Visualization Server*.

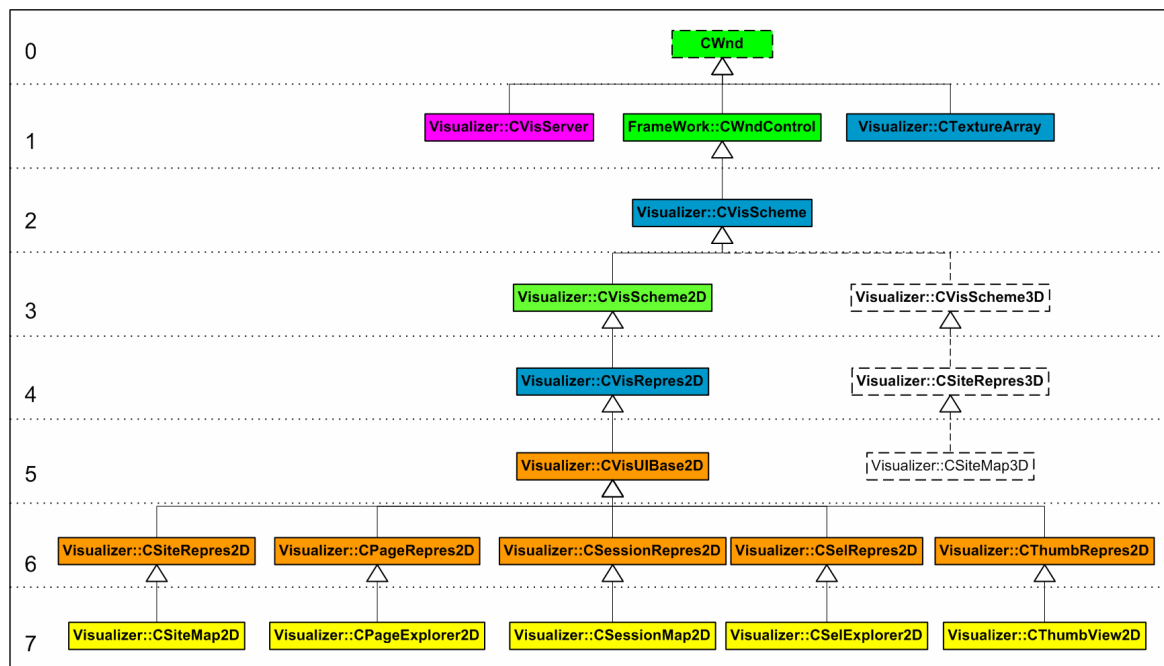


Figure 85 Visualizations hierarchy

Figure 85 presents a simple model of the *visualizations* hierarchy, using color-coding representations equal to the application components code used in Figure 78. This representation highlights all the intermediate levels of abstraction for the *visualizations* implementation. While level 0 represents a basic level provided by the operating system, the other levels correspond to:

1. *Level 1* – the components of this level inherit the basics of a window; `CWndControl` represents the base control for all the *Visualizations*, `CVisServer` is the *Visualization Server* and the `CTextureArray` is used to share the 3D textures between different *OpenGL* rendering devices of the application. All these components are part of the application framework;
2. *Level 2* – the `CVisScheme` represents the basic interface for all the visualizations of the application. It is responsible for the connection to the *Visualization Server* and for the synchronization with the container window;
3. *Level 3* – the `CVisScheme2D` implements the basic functionalities for a two dimensional representation (zooming, translation, rotation) and provides a drawing context, which can be a window or a memory block (using double-buffering);
4. *Level 4* – the `CVisRepres2D` represents the basic holder for all the object instances of a visualization. Basic searching capabilities and object identification are implemented in this level; the possibility of creating search result sets, selecting, hiding or identifying objects is offered to the upper levels. Figure 86 presents a list of possible objects that can be represented and manipulated by our visualization schemes. All these objects are derived from a base object that offers a common interface for

manipulation. Each object has an ID and a name that uniquely identifies it and includes attributes as position, translation or rotation values, etc.;

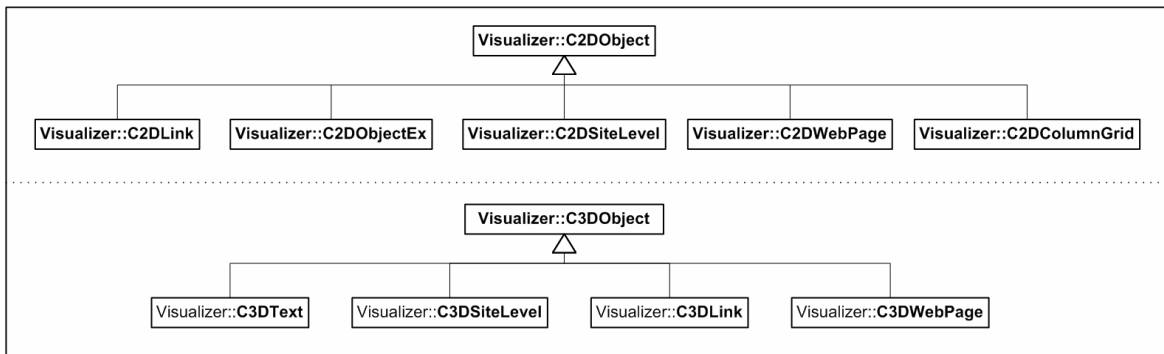


Figure 86 Visualization elements hierarchy

5. *Level 5* – the `CVisUIBase2D` implements the basic and default interaction mechanisms for all the visualizations. This component is the first one in the hierarchy that manipulates directly the representation objects, the drawing device and responds to the events produced by user interactions with the represented information. The following basic operations can be highlighted here:

- change the scheme properties;
- display the *Find Objects* window for usage;
- select objects with synchronization;
- move objects within the representation;
- zoom, translate, rotate functions for the representation;
- reset and fit the representation to the active windows;
- hide / show objects / groups of objects;
- export the representation as an image.

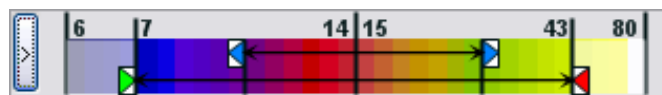


Figure 87 Filter selection threshold with color lookup table

In addition, a basic functionality is present here as the possibility to apply filters that affect the properties of objects in representation, one example is presented in Figure 87 that shows a filter applied on the CLUT (Color Lookup Table), that hides the objects not in the threshold selected by the user;

6. *Level 6* – this level specializes the functionality of the `CVisRepres2D`, accordingly to the purpose and goals of each visualization scheme. Many of the basic data

manipulation and identification functionalities are specialized, also new specific functionalities and representation objects are maintained in this level. Specific drawings, object manipulations, etc., are implemented in this level of specialization;

7. *Level 7* – this level of specialization corresponds to completely functional visualization schemes. Most of the user interactions are specialized here. In addition, a rich set of functionalities is added to each scheme, to provide the right interaction means and tools for each scheme purposes. One example of additional functionalities is that the SiteMAP2D representation provides one tool, among others, that highlights the possible paths between two selected pages (as presented in Figure 64 of section 3.2.2 – Visualization methods for website structure and session analysis, page 90).

As observed in Figure 88, the *Visualization* lifecycle is controlled by the existence of the *Visualization Server*. Whenever a *Visualization* is created, it tries to find a valid *Visualization Server* to connect to it. After the connection is established, the scheme is able to share information and synchronization messages.

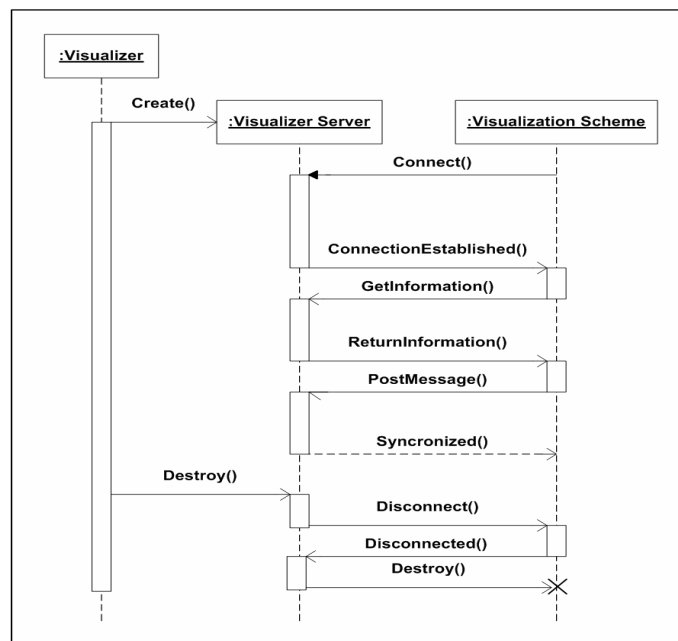


Figure 88 Visualization lifecycle

4.6.4.1. Visualization server and synchronization events

Visualization Server is the component meant to deal with the synchronization between the visualization schemes and is responsible to maintain unique instances of the shared information units. This means that this component holds all the information related to the selected website, usage session, textures for the website pages objects, visualizations synchronization information. Currently, the information concerning the usage sessions and webpage extras is loaded on demand; this is due to the impressive number of usage

sessions that can be logged and the large amount of information related to each webpage.

The information this component contains and maintains is:

- website information, meaning all information regarding the selected website's structure and interconnections, statistical information, etc.;
- usage sessions information representing the logged information related to the website usage;
- 3D textures used for the three dimensional representations. To save the video memory somehow, we implemented the TextureArray storage component that shares the textures with all the OpenGL device contexts as a simple concept of "load once, use many" paradigm.

The synchronizations are possible via a programming interface implemented by all visualization schemes. Whenever a visualization needs to synchronize events (like object selection or properties change), it just calls the public member of the visualization server, responsible for dispatching a specific message to the connected visualizations.

Whenever a visualization needs to update the status of the application, because of a user interaction, it uses the visualization server and sends the synchronization event and the corresponding information. Each interaction with one of the visualizations has as effects the synchronization of all affected visualizations by changing their content or aspect. This functional aspect implements the concept of complementary related and synchronized visualizations with correlation mechanisms detailed in section 3.3 – Visual correlations of visualization methods, page 97.

To synthesize, the actions sequence for synchronization is:

1. the user interacts with a representation and produces an event (select, unselect, hide objects, etc.);
2. the visualization scheme analyzes the user event and, if needed, uses the visualization server to broadcast the conceptual effects of user's action;
3. the visualization server analyzes the event, then generates and configures an internal message to be broadcasted (part of the message information identifies the original sender of the event);
4. the visualization server broadcasts the message to all connected visualizations;
5. each visualization receives the message and processes it; after processing the message, the scheme might change some objects properties or the content of the representation; finally, an action result/code is returned.

To implement these functional aspects we had to define a set of messages used to synchronize the application components. An example of synchronization is presented in Figure 89, for the `ID_SHOW_ALL_PATHS` message. Figure 64 of Visualization methods for website structure and session analysis section presents the effects of the message in `CSelectionExplorer2D` representation.

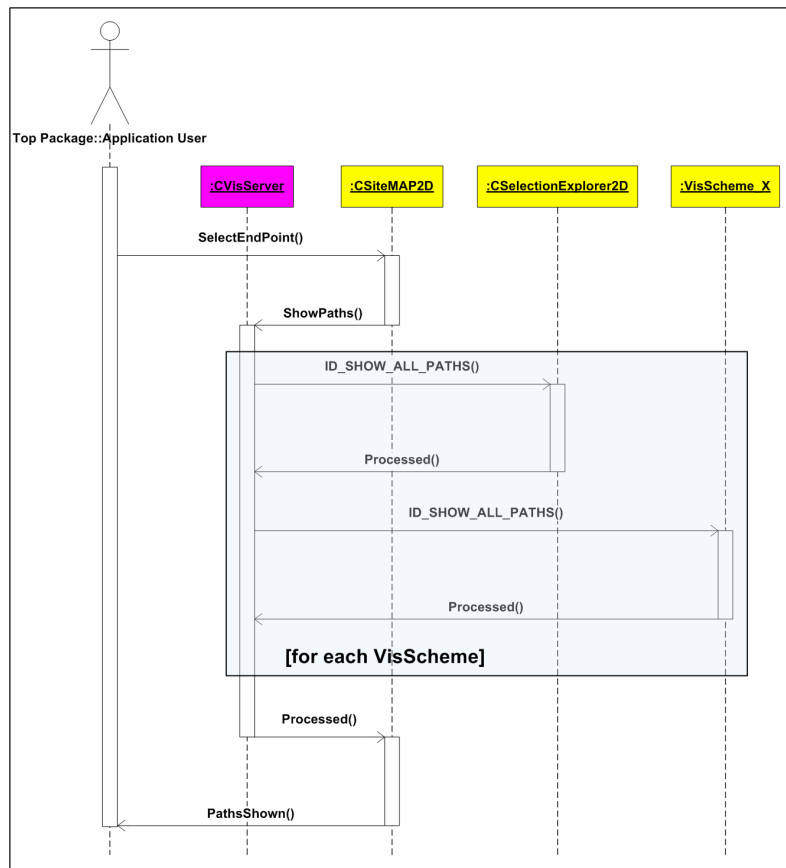


Figure 89 `ID_SHOW_ALL_PATHS` message routes

4.6.5. Implementation of visualization methods

This section describes the visualization methods used in our system. First we describe the subdivision of the visualization tasks in three main areas of interests. Then we present the visualization methods and some related application examples. Finally we deal with visual correlation strategies and presents an example of the unified interface with articulated and tightly coupled complementary visualization methods. Note that for simplicity we some times refer to visualization scheme as visualization or scheme.

Visualization functions/tasks can be subdivided according to three main concerns, at this stage of conceptualization and development:

1. Visual inspection of website structure and its efficiency from the navigational point of view;

2. Visual inspection of interface design coherence, specifically, for navigational and information retrieval purposes; and
3. Representation of tree-structured information traversing in time.

A web page is registered internally in the referential digital atlas (discussed in section 3.1 – Conceptual model) and presented visually, with the information it hosted when it was recalled from the server and shown to the user. The page can be represented in various sizes and spatial resolutions (using scaling) throughout the system depending on the visualization context.

Statistical and usage information are represented in some situations with color. The mapping function uses a color scale; complementary color scales are used accordingly to the needs of the *visualization schemes*. Due to the nature of the statistical information related to website usage, some of the *visualizations* use a linear mapping function others a logarithmic function.

Note that all visualizations provide a set of generic object manipulation tools, similar to ones used in image processing or CAD, as follows:

- Tool to activate the properties panel;
- Tool to activate “Find Objects” panel to look for named objects using their name, Url or other properties;
- Objects Selection tool;
- Drag/Drop tool;
- Zooming tool;
- Translation tool;
- Rotation tool;
- Fitting tool (default aspects, horizontal and vertical fit); and
- Refresh/redraw tool.

In the following sections, the visualizations schemes (methods) and their actual implementation in our test application are presented, specific functional description being detailed for each of them.

The screenshots in the following sections show some preliminary results corresponding to information gathered from an e-learning community intranet related to the Human-Computer Interaction course offered at the Electronics, Telecommunication and Informatics Department, University of Aveiro – Portugal. A detailed application to a case study for this site is presented in Chapter 5, section 5.5 – Application to a Real Case, page 157.

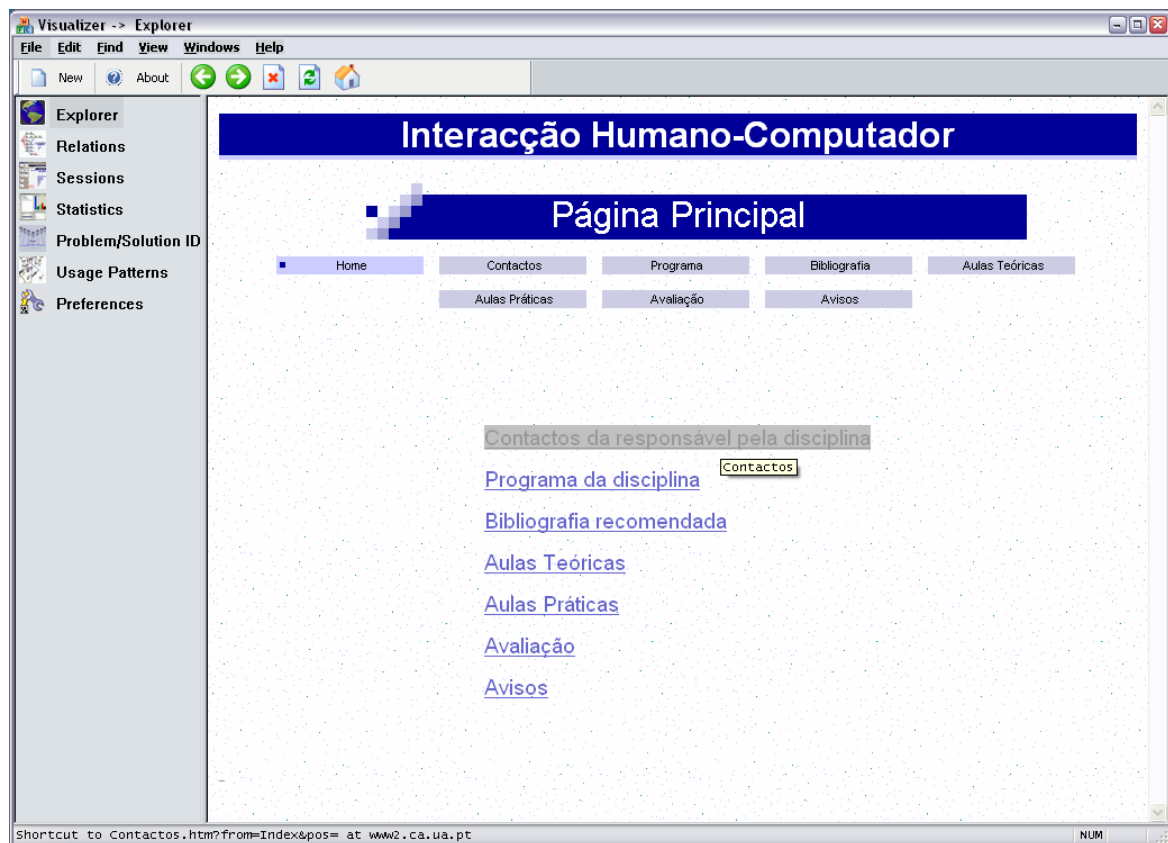


Figure 90 Offline Site Explorer visualization

4.6.5.1. Offline Site Explorer

The *Explorer* module implements the basic web browser functionality, the main difference being in the way the information is obtained. While a regular web browser is supplied with live information retrieval mechanisms, in our case, the information is retrieved from the application database. Every request for a new webpage is translated in a request for the database. All this is possible because we keep a raw copy/image of each page of the site, in our database. This allows us to analyze the exact structure of a website at a specific moment of time, having the possibility of tracking the website evolution. Figure 90 illustrates the implementation of this module.

4.6.5.2. Site Structure 2D (SiteMAP2D)

This visualization is appropriate for network type or tree-structured data with parent (referred) and child (referenced) pages represented at each “node” or intranet page, as presented in Figure 91, similarly to the representations presented in [Dodge2003], [Ricca2001], [Martin2001]. It is normally used to indicate the “goal path”, as shown in Figure 92, with the initial page in green and final page in red. Any other inter page relational path can also be analyzed here, with home page as the default first page if no other is chosen as starting point. Different site areas are distinctly color-coded.

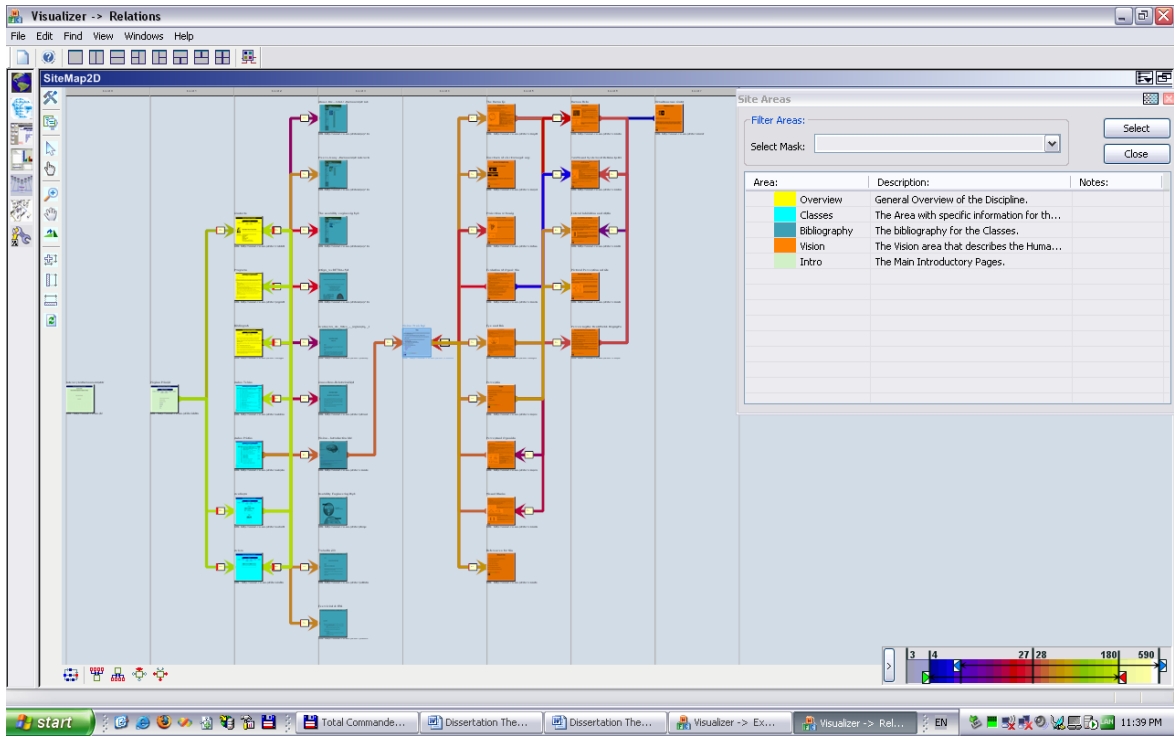


Figure 91 Relational visualization of site structure

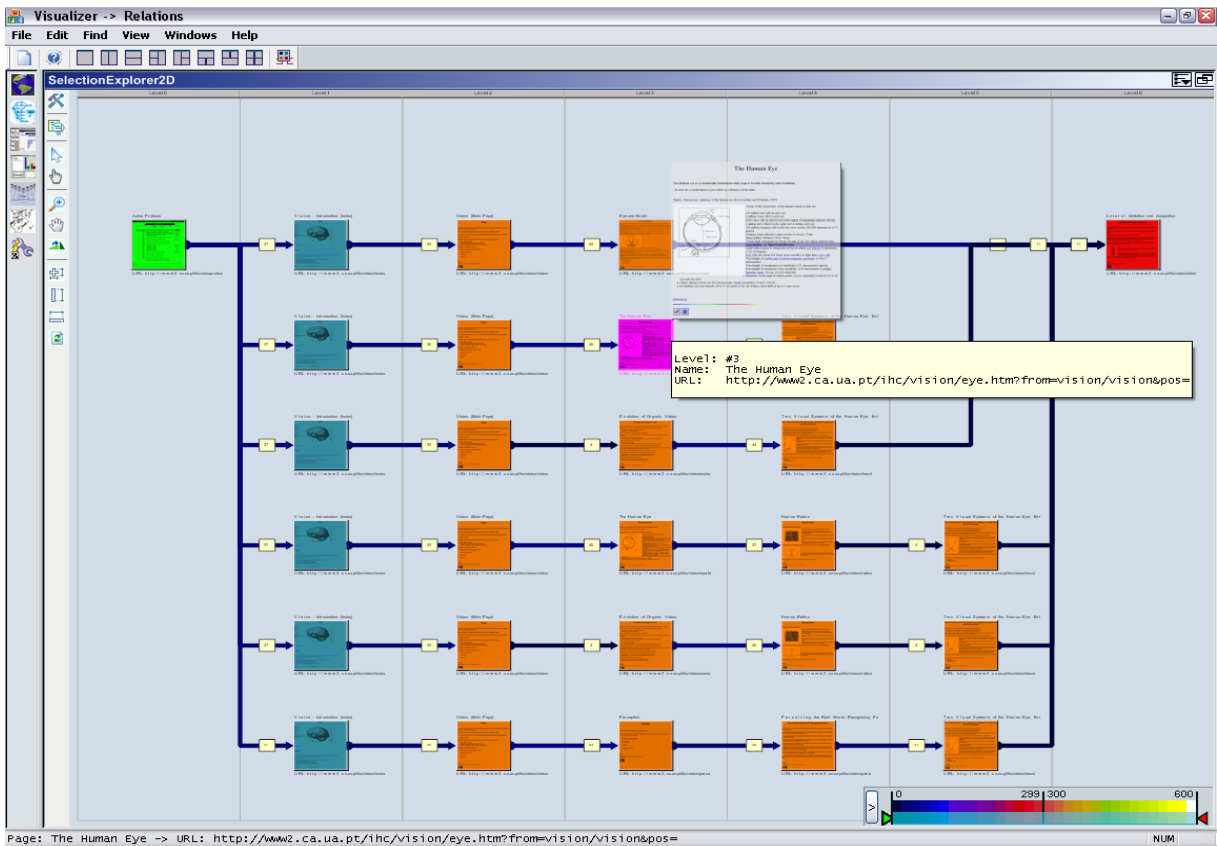


Figure 92 Path to goal visualization

4.6.5.3. Path to Goal – Interconnections (SelectionExplorer2D)

A first criterion for efficiency classification to achieve a goal is related to the minimum “goal path” between an initial (green) and end (red) page, the second is related to the amount of time taken to go from one page to another using a specific link element. Visual inspection of these parameters can be done as shown in Figure 92, where alternative paths to goal page are shown with roll-over functionalities to obtain statistical inter page information.

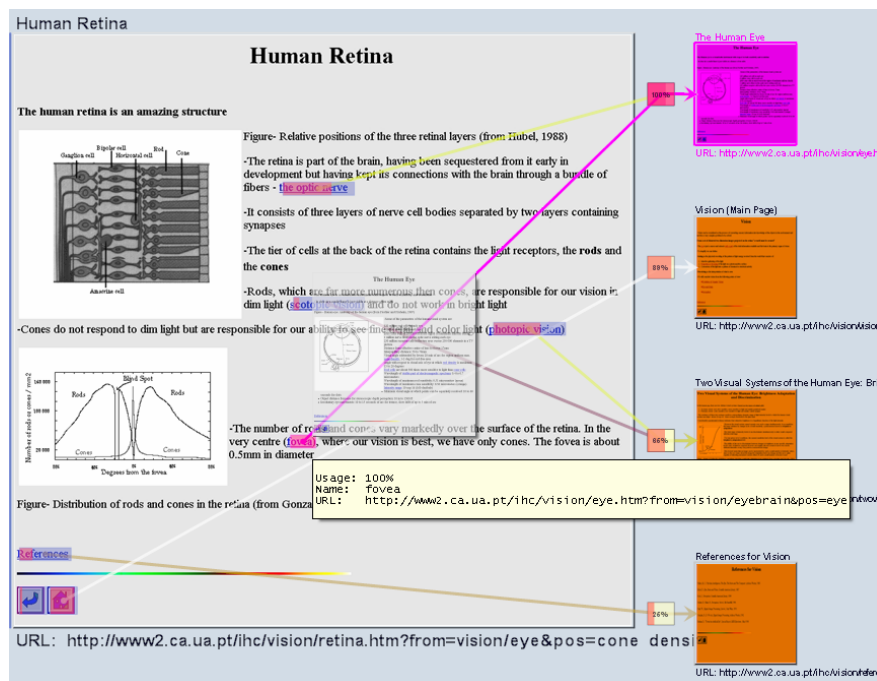


Figure 93 Interactive zones visualization

4.6.5.4. Interactive Zones and Page Relations (PageExplorer2D)

This scheme is used to visually inspect the link elements position on the page, hypermedia connections and statistical information of a page (Figure 93).

Five visual elements are used for this purpose and can be described as follows:

- representation of an intranet page as a base image information layer (discussed in section 3.2.2 – Visualization methods for website structure and session analysis, page 90), represents all the information of a page as seen by the user, obtained from the image of the page it belongs to (as it was presented to the user);
- representation of all the page hot-spots areas as presented to the user; uses attribute information to delineate the hot-spot outline on the page information;
- representation of all the page thumbnails referenced by each hot-spot; draw all thumbnail images of the pages referenced by the hot-spots with information obtained

from the image of the page that is referenced as it was presented to the user. Multiple referenced pages are represented only once;

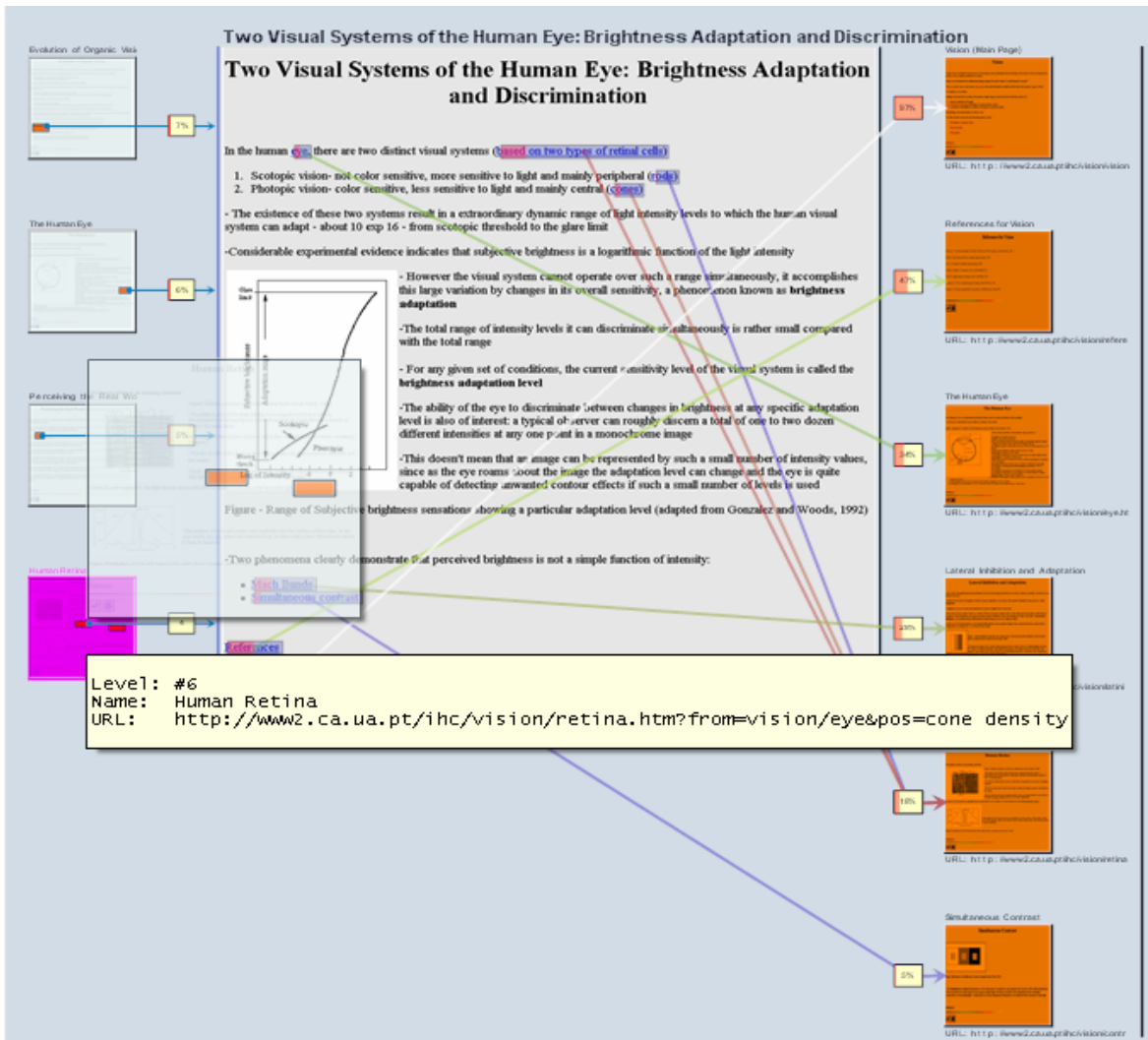


Figure 94 Page relations visualization

- connection line between hot-spot and referenced page thumbnail; it uses the position and size of the hot-spot in the page and the equivalent thumbnail position to represent the connecting line and the number of times this hot-spot was visited to calculate line thickness and statistics;
- connectivity info obtained from the structural and statistical attributes defined for page and link elements: driven by a roll-over event on the connecting line, this visual feature constructs a pop up window (image tip) with statistical information, URLs, etc. related to the nodes or to their inter-connectivity. The visual feature is obtained from the URL of the page it belongs to, number of times this hot-spot was used, URL of the page that is referenced, and time to go from one page to another using this hot-spot.

Whenever a page thumbnail is selected in any other visualization scheme, the visualization shown in Figure 94 is updated to highlight the page relational hypermedia properties and in context usage statistics.

4.6.5.5. Tree-Structured Traversing in Time (SessionMap2D – Timeline)

The usage activity representation shown in Figure 67 or Figure 95 may lead to a discrete function, considering time on one axis and tree level on the other. The green square symbol represents the starting page and the red is the last page where the usage session ended. The various visited pages are represented chronologically according to:

- elapsed time (between page views) and
- their position in the tree structure constructed with a breadth first algorithm.

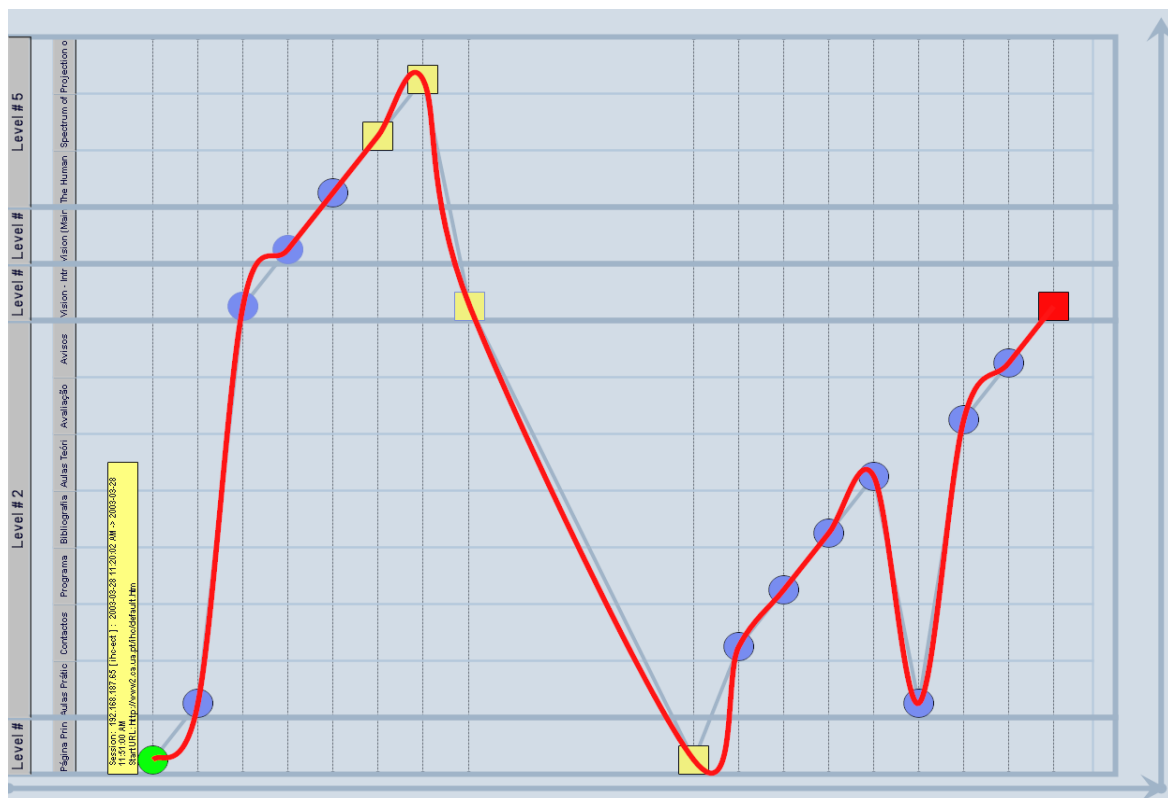


Figure 95 Tree-structured traversing in time visualization

It seems a promising representation if analyzed according to the function first derivative in time where abrupt changes may lead to jumps between pages depicting problems in information structure or on the interface. Jumps back over more than one level might indicate that the users cannot find the desired information and have to use the browser's history to go back to a previously visited page. These are preliminary research results and still need further theoretical research and empirical validation. However, informal evaluation sessions highlighted user's interest for such situations when the exploration sequence is not coherent. The coding of the amount of time between subsequent jumps

as distance provides better visual clues on the time users spent to analyze specific pages. A possible evolution for this visualization is to map usage session information to a specific subset of website pages, as selected by the user.

4.6.5.6. Visual Workspace Coherence (SessionMap2D – VisualSpace)

Path representation considering hot-spots as network nodes also uses hypermedia properties as mentioned in [Bieber1997]. Inter-node connections are represented using one color for a line arriving at a node (referenced) and another color for a line leaving a node (referred). Overlapped rectangular representations of the hot-spots that were used by the users during the analyzed session are represented one in top of another, color coded according to usage, as shown in Figure 96, which is similar to some of the proposals presented in [Card1999], specifically, [Becker1999] and some commercial tools as [LiveSTATS2005], [ClickTracks2005] and [Webtrends2005].

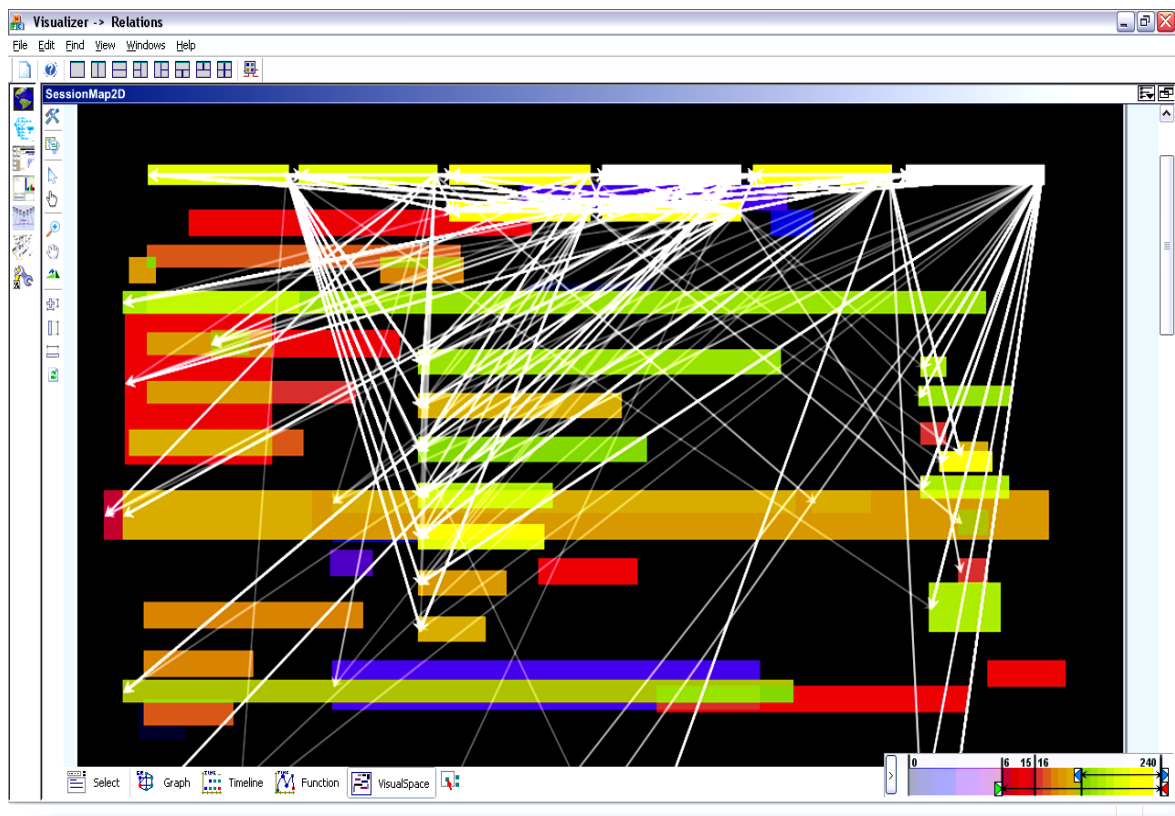


Figure 96 Visual workspace coherence visualization

Important interface design ambiguities can be revealed by this visualization. For example, analyzing Figure 96, an interface design analyst can discover a usage pattern on the visual workspace's right side (thicker lines and lighter colors that reveal intense activity), that might illustrate some interface design ambiguity which should be revised. It thrives for consistency with a main interface page centered design strategy. The left, and some

bottom low usage hot-spots are related to an intranet module rarely used and that is not in conformance with the overall interface design strategy.

4.6.6. Visual correlation strategies

Our system uses a user interface strategy that integrates various visualizations, dialogue and feedback information units into one unified interface, as shown in Figure 69. In some occasions, the interface depicts simultaneous visualizations with different but complementary information inspection methods. Some visualizations present qualitative visual information and others quantitative statistical and structural information. [Andrews1999] presents a case study using a tightly coupled multiple window visualization inspection method with 2D and 3D representation schemes. The tightly coupled multiple view windows are fundamental for a coherent inspection of the information but context is lost in some situations with the inter view window jump. Interface design strategies in this situation should guarantee a coherent context with all relevant multiple views present at one time. This proposal delivers an articulated and contextualized visual inspection method sustained on simultaneously present views. The visual correlation of qualitative and quantitative network information properties is considered.

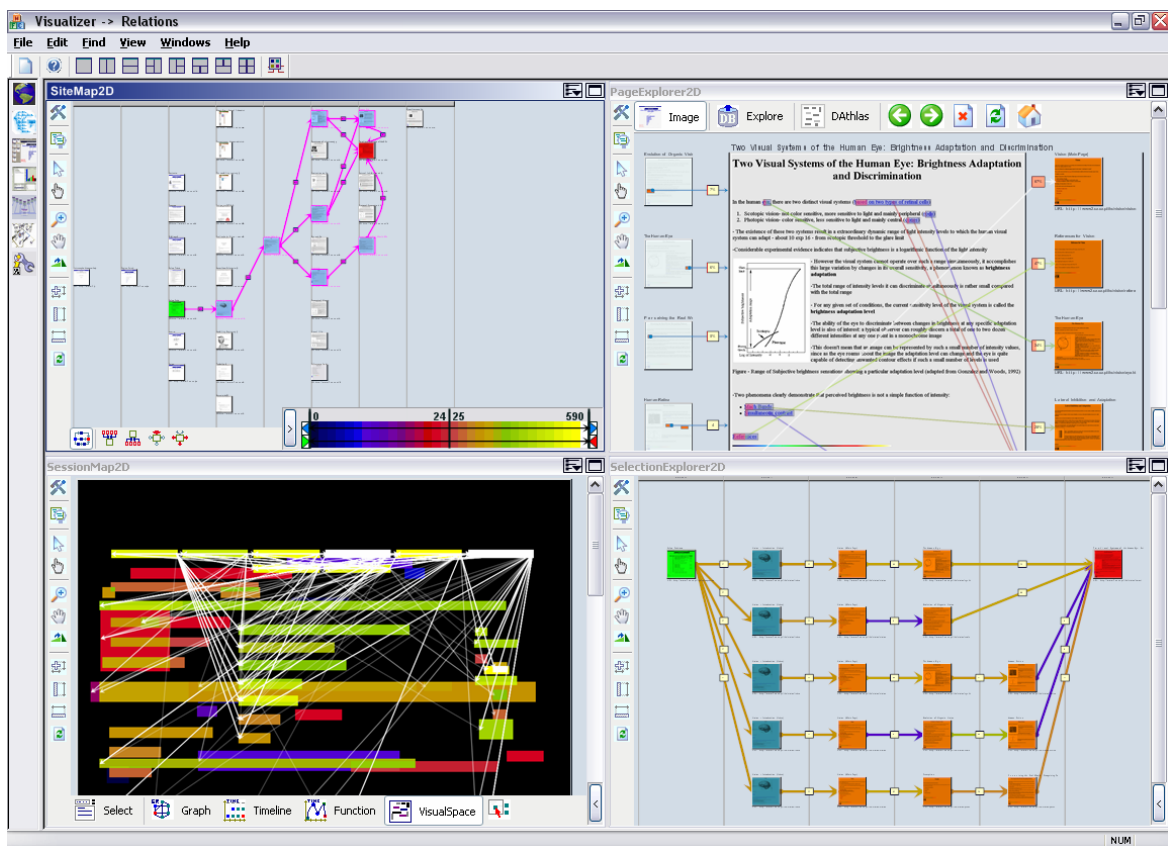


Figure 97 Synchronized visualizations triggered by the same event

An inspection exercise starts with the specification of a goal that can consider as starting page the site home page, or some other, and ends with the specification of a final page where the information or contents is located and is specified as a usability evaluation goal. An articulated and tightly coupled set of views are generated to facilitate the inspection and analysis of inter-page related statistics and interface design parameters. All these schemes are triggered and redrawn by the same event and use the same raw information space to represent their visual features: see the visualization in Figure 97 where the top left, top right and bottom right visualizations present the results of the same event triggered in top left visualization.

Chapter 5. Evaluation and Results

This section presents and discusses the evaluation of the Visualizer application: theoretical aspects, preparation phase, users, equipments, evaluation tasks, general and specific results obtained as well as improvements proposed as a consequence of the evaluation. An application to a real case is also presented and the usage of the prototype is exemplified and discussed for this case.

5.1. Evaluation methods

During the development of the *visualizations* and user interface, we have conducted informal evaluation sessions, using preliminary prototypes, with several people. These people were mainly volunteer Computer Engineering students, which in general had some experience as web developers. These sessions were very useful and allowed us to discard some ideas and refine others. Two members of the team, the ones less involved with prototype implementation, have also made some heuristic evaluations of the user interface features and some informal evaluation of the *visualizations*. After this first evaluation cycle, we developed a more sophisticated prototype including six main *visualizations* and the mechanisms to use them synchronously which was presented in the last chapter. Then, we started a second evaluation cycle. In this cycle, we adopted a different, more structured, approach that is briefly presented in the following sections.

This second cycle of evaluation had two major goals: to evaluate the user interface main aspects and to evaluate the *visualizations*, including the interaction mechanisms provided to users so that they can interact with data through the visual representation. According to [Freitas2002], to evaluate visual representations we can use cognitive complexity, spatial organization, information coding and state transition, and to evaluate interaction mechanisms we can use orientation and help, navigation and interrogation and data set reduction.

We performed several evaluation sessions, with two types of users, using mainly observation and query based techniques to evaluate both the user interface features and some aspects of the *visualizations*. These techniques are widely used in usability evaluation of user interfaces, but they have also been considered appropriate to evaluate some aspects of visualizations [Ware2000]. The collected data was analyzed using Exploratory Data Analysis.

Users and Observers

In order to perform the two kinds of evaluation, i.e., user interface and *visualizations*, we have asked for the collaboration of two different types of users: thirty-two Computer

Engineering students, currently attending an introductory course on Human-Computer Interaction, and five professionals that work at the Centre of Informatics and Communications of our University (CICUA). These professionals have several years of experience as web developers, web/network managers and three of them have attended the same course on Human-Computer Interaction, during their graduate studies. The profile of these professionals made them not only representative of our target users, but also capable of understanding well what kind of feedback we would need from them, in the scope of this evaluation. While the students are not, in general, web developers/managers, they have a profile that makes them also reasonably suitable as subjects for the evaluation of our user interface, since all of them have experience as web users and computer programmers. Therefore, we have conducted an evaluation more focused on the user interface with the help of the students and an evaluation of the *visualizations*, as well, with the help of the five professionals.

Since the students had been practicing user interface evaluation through different methods, we decided to profit from their capabilities and ask them to act as observers as well as users. This procedure would allow us to obtain observation data from a larger number of users, provided that we would ask the observers to register simple enough information (since they are not very experienced). We considered this would provide an interesting practice for the students and that was also a motivation to have all the students act both as users and as observers.

Therefore, while half of the students would perform some predefined tasks, the other half would observe them and register times, task completeness, as well as other relevant information. After some predetermined time they would change roles. Obviously, the students that would act first as observers and later as users would have a greater acquaintance with the interface than the others, a different level of awareness, and could be considered as more experienced users. Hence, for the purpose of data analysis, the students were divided in two classes of users: less experienced users and slightly more experienced ones. We should notice, however, that in spite of the fact that we have two groups of users, the performed evaluation is non-experimental, in the sense that there is no control group, nor has been defined any hypothesis. Its main purpose was to gather ideas that can be further explored in later stages of this work.

The basic demographic data of each participating user was collected before the tasks, through a questionnaire, including also some questions meant to assess their experience with information visualization applications and as web developers or managers. Analyzing the collected data we found that they were between 19 and 31 years old (median value=21 and two outliers aged 30 and 31), 3 females and 29 males, having no difficulties in color perception. Moreover, concerning their experience as web designing and evaluation, most of them were able to produce web pages of moderate complexity and were acquainted with the basic methods of evaluation. Finally, concerning experience

with applications using 2D visualization, most of them use frequently several packages (e.g. Macromedia Suite, Adobe Suite and MatLab).

After completing the tasks, a post-task questionnaire was given to the users aiming the assessment of their satisfaction and opinion on several issues.

Database

An internal site containing information (such as program, studying material, practical assignments, etc.) corresponding to the Human-Computer Interaction course, offered to approximately forty Computer Engineering students, in 2002/03 at the Department of Electronics, Telecommunications and Informatics of the University of Aveiro, was developed specifically to collect usage data. The database used contained the information collected during the whole semester both in normal usage and in controlled sessions.

Equipment

The evaluation sessions were performed in a laboratory classroom equipped with PC computers running the prototype and an SQL Server for the Database.

Evaluation tasks

We defined a set of tasks for the users to perform during the evaluation sessions that were relatively simple, nevertheless regarded as representative of typical operations end-users will perform with the visualizations and the user interface.

Keeping tasks simple makes it easier to analyze user performance; however tasks should not be so simple that their ecological relevance is unclear (i.e., we have to ask how frequently do those tasks actually occur in real-world tasks, and how significant are they in the overall task solution process).

Each user had to complete ten tasks within a given time window. The tasks were related both to the evaluation of the user interface and the visualization schemes. The following are some of the tasks (a more complete example is shown in

Table 11 and Table 13):

- Manipulate and navigate among the visualization windows;
- Use a menu option or tool button to obtain a given functionality;
- Select a given site;
- Select a given session;
- Find and select a given page of the site;
- Find how many times a link was followed between two given pages;
- Show possible paths between two given pages;
- Find the number of pages corresponding to the shortest path between two given pages;
- Find how many external jumps has a user performed during a session.

The first four tasks of the previous list, are very simple and directed to the evaluation of some user interface features (change viewing conditions or use functionality through buttons or menu options); the other tasks are combined with some of the previous ones in more complex tasks focused on evaluating the performance of the user using the *visualizations* to extract some qualitative or quantitative information through interaction with the data. “Find and select a given page of the site” and “Find how many times a link was followed between two given pages” are tasks oriented to evaluate the interrogation features of the *Visualization*. “Find the number of pages corresponding to the shortest path between two given pages” and “Show possible paths between two given pages” are intended to evaluate the data set reduction feature, according to the evaluation criteria proposed in [Freitas2002] to evaluate interaction mechanism of visualization schemes.

Two distinct procedures were selected to perform the evaluation: one with the students, another with the professionals; additionally, several types of measures, user satisfaction and opinions were collected. Details can be found in the Annex 3.1. Procedures and Measures.

5.2. Results

In this section we describe the results obtained with the students, which are more quantitative and more focused on the user interface, as well as the results obtained with the professionals, which are more qualitative and focused on the evaluation of the *visualizations* and overall interest of the application. STATISTICA [STATISTICA1999] was used to process the collected data.

Results obtained with students

As mentioned before, after the first round of sessions (TS1 and TS2) we had the impression that some of the students had not fully understood what they were supposed to do; thus, we simplified the fill-in-forms for the observers as well as the tasks phrasing.

Then, we decided to analyze, as a first approach, the following items collected only during the evaluation sessions TS3 and TS4:

- the time spent in each task;
- if the task was completed;
- if the answer was correct (in some of the tasks);
- the user satisfaction.

Figure 98 shows the box plots obtained for the times corresponding to all tasks performed by Users #1 and #2. The median values for the times are 60s, and 32s, respectively; using a Wilcoxon test we have found the difference between these values significant ($p < 0.00001$). While Users #1, as well as Users #2, had about the same amount of experience with the application before the beginning of this round of evaluation sessions, and we were not expecting to observe a significant difference in performing the tasks, there was a difference in the median time. This seems to mean that Users #2 learned how to perform the tasks faster just by observing their colleagues. However, no significant difference on the number of correct answers was observed between the two types of users.

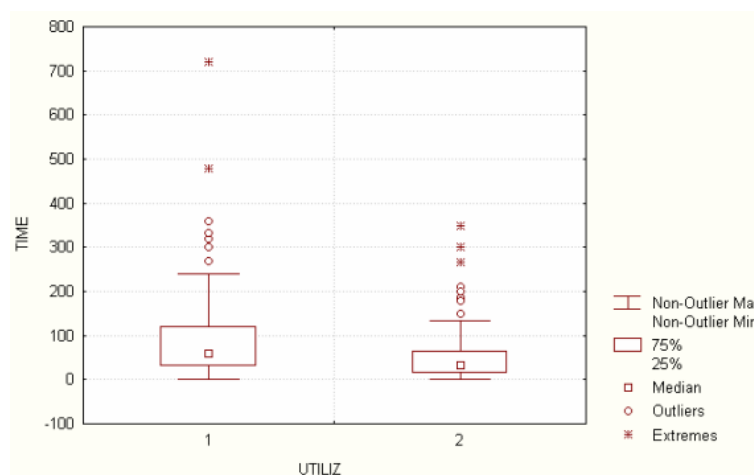


Figure 98 Boxplots corresponding to times in all tasks for Users#1 and #2

As to satisfaction, the median values for Users #1 and #2 are respectively 3 and 4. Analyzing user satisfaction and time, task by task (Figure 99), we informally noticed that the users are less satisfied when they take more time to complete the tasks. This result was confirmed as significant in tasks 2, 4, 5, 6, 7, according to the Spearman correlation coefficient.

Table 9 Number of users that completed or didn't do tasks without any question

Task	Completed	Didn't do
1	32	0

2	32	0
4	31	1
6	31	1
8	29	3

Table 9 shows, for each task that had not any question to answer, the number of users that completed or did not complete the task. This table shows that the great majority of the users were able to complete the tasks. In fact, 97% of the tasks were completed. While for these tasks we cannot know if the users have completed them correctly, for the tasks that have a question to answer we have this information.

Table 10 Number of users that completed correctly, incorrectly or did not do tasks having a question

Task	Correctly	Incorrectly	Didn't do
3	24	4	4
5	26	5	1
7	17	12	3
9	29	0	3
10	16	2	14

Table 10 shows, for each of these tasks, the number of users that completed correctly, completed incorrectly and did not complete each task. Counting the total number of correct answers to the questions, we found that a high percentage of users were able to find the correct information through the visualization schemes: 70 percent of the tasks having a question were completed correctly.

In task 3 and 7 users were supposed to find how many times a given link had been followed, using two different *visualizations* (SiteMap2D and Page Explorer2D). From the 32 users, 24 completed correctly task 3, 4 users completed it but obtaining a wrong answer and 4 users didn't perform the task. As to task 7, 17 users completed correctly the task, 12 have completed it but obtained a wrong answer and 3 didn't do the task. This difference in user performance was confirmed as statistically significant using the non-parametric sign test. This suggests that SiteMap2D could support better this kind of task than PageExplorer2D. It is interesting to note that, while the users show a better performance in task 3, their satisfaction is higher in task 7. This result confirms the notion that a higher user satisfaction does not necessarily mean a better performance. This could be related to the total time to complete the task, which is lower for task 7 (median time=30s) than for task 3 (median time=60s).

In task 5 users had to find the number of pages corresponding to the shortest path between two given pages. For this task they had to select the pages using SiteMap2D and then observe the result using another *Visualization* displayed on another

synchronised window. From the 32 users, 26 were able to obtain the correct number of pages, 5 obtained an incorrect number (which means that they probably were not able to select the right pages on SiteMap2D), and just 1 user was not able to perform the task. Moreover, the median time to complete it is the 2nd lowest time (median=30s) among all tasks. This seems to suggest that using these two synchronised *visualizations* is reasonably obvious to the users.

Task 9 implied using another *Visualization* (SessionMap2D) in order to find the number of jumps to external pages. In this task, 29 of the 32 users were able to complete the task and obtain the correct number of jumps; only 3 users didn't complete it. This seems to suggest that SessionMap2D supports adequately this task.

Task 10 was much more difficult, intentionally devised to see if the students would be able to perform a complex task using the application. The median time was 268s and the median satisfaction was 2 (the lowest value of all the tasks). Even so, 16 of the 32 users were able to do it correctly obtaining the right answer, 2 completed it but obtained a wrong answer and 14 didn't do it. Whereas it was necessary to give a hint on how to perform the task to some of the students, we were expecting worse results in this task.

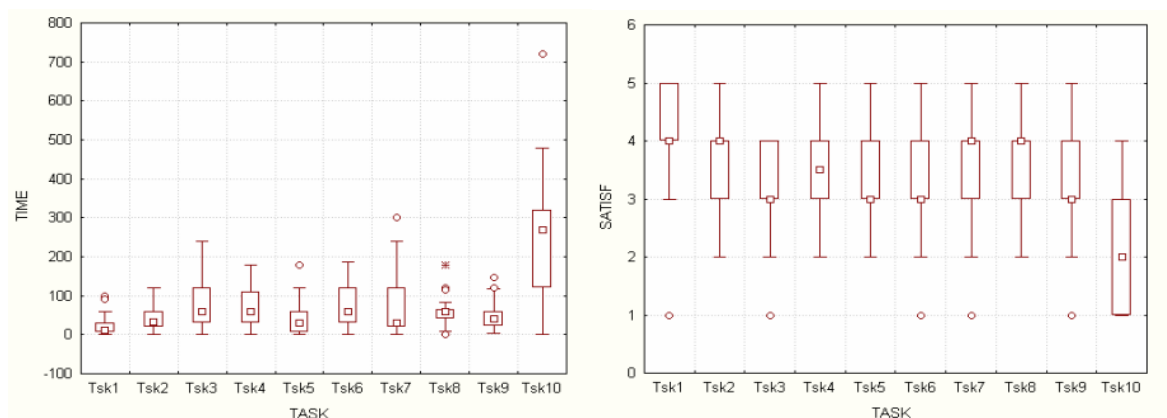


Figure 99 Boxplots corresponding to times and satisfaction task by task

Even if most of the tasks were relatively simple, these results are encouraging since the students were not experienced users and were able to perform correctly a high percentage of tasks. Also, the median value of the overall satisfaction was 3 for Users #1 and 4 for Users #2.

Results obtained with the professionals

During the sessions performed with these users, we collected interesting low level feedback concerning specific features of the user interface, and visualization methods included in the prototype, but also more general, high level information related to the interest and usefulness of the application.

Concerning the user interface, some icons, the color scales and some dialog boxes, were considered as the weakest points. According to some users the synchronized representations of the same data is the most interesting feature of the application; however it makes it more complex.

These users generally considered the application as having a great potential and were interested in using it as soon as we could produce a consolidated first version.

5.3. Proposed improvements

As a consequence of the performed evaluation we obtained results that can be classified in three major classes:

1. Conceptual issues as a collection of user suggestions or observations, resulted by direct observation of application usage (each test had a set of persons evaluating the usage of others);
2. Interface issues as a collection of suggestions and observations of both users and evaluators; and
3. Architectural issues as results of a new conceptual analysis and refinement of the system, based on the current application functionality and the results of the evaluation.

Conceptual issues

- divide the application in different conceptual interconnected sub-components ;
- implement filters for the displayed / analyzed information;
- divide the site into areas of interest (good for exploring large sites);
- integrate some of the functionality of dedicated log analysis tools and browser add-ons (e.g. *OMNIWEB* [OMNIWEB2004]);
- show color-coded usage information on hotspots and links on page representation;
- make the color LUT dynamic and interactive (use it as a filter) and select the color coding scale similar to web design / analysis software;
- divide the *SessionMAP* representation and functionality into sub-components and add more interactivity to visual workspace coherence visualization;
- allow identification of pages using general filters on find results (improve actual search engine);
- color-code the usage on the page objects, allowing different levels of detail;
- make the tooltips configurable for every type of object in the visualizations;
- refine the visualization of tree traversing in time session usage accordingly to the time jumps, using distance-coding features.

Interface issues

- hide the status window by default;
- correct the flickering when scrolling;
- show feedback about which visualization is focused;
- correct the zoom tool functionality;
- add scroll bars and update them accordingly while zooming;
- make the default button active on search panels;
- put the link usage number on the link tooltip;
- display feedback after search / when finishing searching;
- change the size and content of the icons, making them more intuitive;
- make possible paths tool more intuitive;
- use the default image processing shortcuts and functionality;
- correct the drawing cycle flickering while many paths are displayed;
- display more information while moving the mouse over the hotspots;
- implement a tool for highlighting specific paths while showing all paths;
- allow multiple selections to be synchronized with all the representations; and
- sort the linked pages, on page visualization, accordingly to the usage.

Architectural issues

- refine the framework classes and some functionality;
- refine the visualization classes accordingly to the common functionality;
- update the database model to store more information about a specific page / group of pages;
- optimize the *Visualization Server*, message routes and information organization; and
- refine the interface classes to offer more flexibility to the user.

5.4. Discussion

In this chapter we described the evaluation of the visualization application introduced by this work. We used observation and querying techniques and asked two types of users to perform a set of tasks meant mainly to evaluate the usability of certain user interface features and of the interaction mechanisms of the visualizations. Not only these formal evaluation sessions, but also informal sessions helped us understand the implications of such a complex system.

The two types of users, professionals and students, revealed many user interface problems and possible improvements regarding icons, menus, color scales, toolbars layout and functionalities, visualizations synchronizations, segmentation of objects, etc. as detailed in section Proposed improvements of this chapter.

Another type of discussion is concerned the visualization methods we proposed and, as expected, the feedback came mostly from the professionals. They addressed more conceptual topics as the usage of filters that apply to overall analysis process or to specific areas only, visualizations synchronization and their effects, the segmentation of visualizations in different components to be integrated in web pages, as well as more technical issues concerning the architecture of the application and the implementation.

Additionally, the results of the evaluation sessions performed with the students helped us understand the impact of the presented UI and concepts for non-initiated users. They were mainly discovering UI issues and, in a few cases, conceptual issues regarding the synchronizations of the visualizations or the awareness of the represented information. Thus, we concluded that this type of users is not fully adequate to test the concepts involved by such an application.

The analysis of both formal and informal discussions with non-initiated users or professionals revealed the synchronization of visualizations, UI navigation and integration as the key contributions of our application. However, for some of the users, the synchronized views had an increased the complexity, in their opinion, making the analysis more difficult. Additionally, page exploration, visual workspace coherence and the inspection of tree-structured information traversing in time represented the key visualizations of a great interest for the users (mostly the professionals that had more experience with similar of visualization).

These evaluation sessions revealed some important visibility or interaction problems that we corrected or tried to correct in a later development cycle. Some of these aspects concerned the perception of usage information coded on hovering tips, as an example being the usage of a scale to code usage information without giving any feedback of the representation domain. We tried to correct this aspect by providing a segmentation tool able to inform the user of the current domain of the representation for additional statistical information, as well as the coding scheme used.

It looks that the information presented on hovering tips over inter-page linkage elements lack in details, the textual representation being confusing for some users. However, the same information is visually represented on the page exploration visualization and seems more adequate for this purpose.

One important issue addressed by professionals was the ability to analyze and classify dynamically generated web pages, or, even more complex, the personalization issue that allows users redesign their personal views of web pages as they need to. These aspects might reveal some architectural problems we might address and were scheduled on our research agenda.

5.5. Application to a Real Case

This section discusses a complete usage scenario with some preliminary results corresponding to information gathered from an e-learning community intranet related to the Human-Computer Interaction (HCI) course offered at the Electronics, Telecommunication and Informatics Department, University of Aveiro – Portugal. The prototyped application, discussed in 4.6.5 – Implementation of visualization methods, has been used for the visual analysis scenario.

Using the timeline presented in Figure 71 and discussed in section 4.1 – General objectives and system overview, page 101, the phases required to analyze a website begin with data collection (gathering, filtering, classification and definition), proceed to the processing phase to collect statistical information and end with the visual representation that helps problem and solution identification. A preliminary step is to define a time interval in which the site usage is considered for analysis, then:

1. use the *Site Analyzer* component to analyze the site structure and page contents;
2. use the *Compiler* application to analyze the website log files plus monitoring information (corresponding to the given time interval);
3. finally, use the *Visualizer* to create specific visual representations of the selected information, to analyze and identify possible problems (usage behaviors, design layout, conceptual and implementation issues).

The following sections exemplify these phases for our case study.

5.5.1. Step 1: Information gathering

The first step involved making an offline capture of the website structure and contents, as well as the identification and analysis of the website structure and web page elements.

We used the *Site Analyzer* to capture the website structure and analyze each page contents, in a semi-automated process that revealed 35 web pages. For the complete set of 35 pages 199 link elements were discovered, corresponding to the hotspots that connect the web pages of the site. For all pages of the website, a total set of 2524 page elements have been identified and stored in the database. The first version of the Site Analyzer did not specifically distinguished text and graphical elements as separate entities, being identified by a type attribute; however, link elements were considered separately, being denominated as hotspots.

The analysis process produced a set of formatted text files, images, configuration files, page snapshots (manually organized), that were analyzed and inserted in the database using a specific module. During this phase, we had to manually adjust some functionality of the tools with the purpose to filter and synchronize the information collected manually with the one collected automatically.

No monitoring was performed since, at the time of the analysis, the *Interceptor* was not fully functional and eye-tracking system was unavailable.

We used a filtered log file for analysis purposes, with a time window defined as the interval from 2003-02-17 to 2003-06-11, which corresponds (more or less) to the second semester of a scholar year at the University of Aveiro.

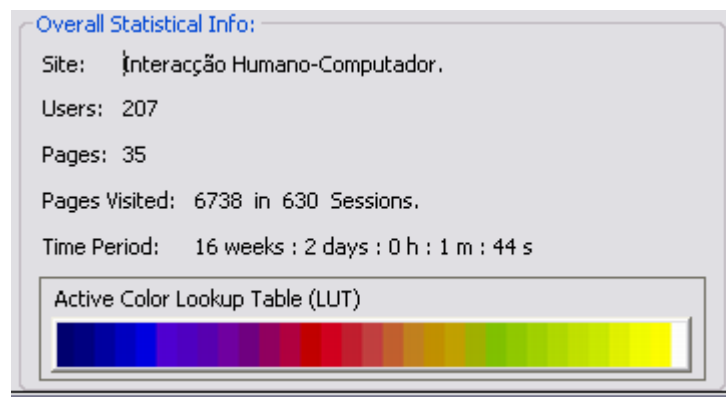


Figure 100 Overall website statistical information example

5.5.2. Step 2: Information filtering

Since some of the pages might have associated a set of URIs described by query parameters, we determined a sub-set of 98 additional URIs for the main set of 35 pages. This was the first filtering process we applied to the collected information to determine unique pages referenced by multiple URIs.

A second filter was applied to the log files to filter only the requests in the specified time window and the specific domain we analyzed. Some other filters were applied during the logs analysis phase as described in following section.

5.5.3. Step 3: Classification and definition

This phase was the last preparatory phase and involved the processing of the log files, detection of sessions and definition of semantic site areas. The processing of the log files has been implemented using the *Compiler* specifically designed for this purpose (discussed in section 4.4 - Compiler). The filtered log file contained 17379 requests from the 41008 requests registered on the first half of 2003. A number of 207 users (distinguished by the computer name or IP) and 630 sessions were identified (see Figure 100).

During this phase, the compiler also established the synchronizations between the website structure and content classification, previously produced by Site Analyzer and some additional tools. Synchronization means the identification of pages specified by each session request, identification of link elements (hotspots) as requested/clicked by users, identification of referrers for each session request if not already present,

computation and update of statistical information for each page and link element based on session requests, etc.

Of a significant importance here is the presence of scripting link elements that force the navigational context to go back one level into the history, e.g. `javascript:history.back()`. This type of situations have to be considered as deadlocks for the automation process, since one cannot predict the subsequent page navigated from the current context since it depends on the user's behavior.

The definition of semantic site areas was a manual process, finalized with the help of the site designer. The complexity of development or integration of an automated semantic content analysis system (semantic mining) makes it unaffordable for the purpose of this work.

Once this third step of the data preparation process was complete, structural website, contents and usage information was available for querying in the relational database associated to the test system we implemented. From this step forward, the information for this specific version of the website capture was prepared for automated processing, visual representations and interpretation, using the *Visualizer* application (discussed in section 4.6 - Visualizer).

5.5.4. Step 4: Visualization, statistical analysis and results interpretation

In this section we focused on three areas:

- Visual inspection of website structure and efficiency;
- Visual inspection of interface design coherence; and
- Inspection of tree-structured information traversing in time.

5.5.4.1. Visual inspection of website structure and efficiency

Using the representation of the studied website, as presented in Figure 101, the structure of the website appears as a balanced tree divided in two sections, levels 1-4 and levels 5-8.

One can observe an intense connectivity on the third level of the website, which suggests that each page on this level contains direct link elements to all others. For this level, the statistical information (coded as color of the connecting lines) suggests a balanced navigation scheme that leads quickly from one point to another (see lighter color codes on the color lookup table on the bottom-right side of the figure).

A very poor connectivity on the fifth level can be observed, where a unique page links to two semantic areas/segments of the site (the pages in the two segments are differently colored): this factor suggests a poor navigation scheme from one segment to another.

The same problem occurs with the page “Vision – Introduction (Index)” that links the fourth level (a level is a column in the representation) to the fifth (the page inside the blue ellipse). The presence of only one link element from segment defined by the fourth level to the segment defined by the fifth level might lead to navigational problems, this due to the impossibility of the user to navigate directly to the page in level four. In the actual site implementation, the page on level five contains a link element that activates the browser’s history and goes back to previous page: *javascript:history.back()*, an action that depends on user’s navigation, which means that we cannot predict if the action leads to the page on fourth level or to a page in the sixth level.

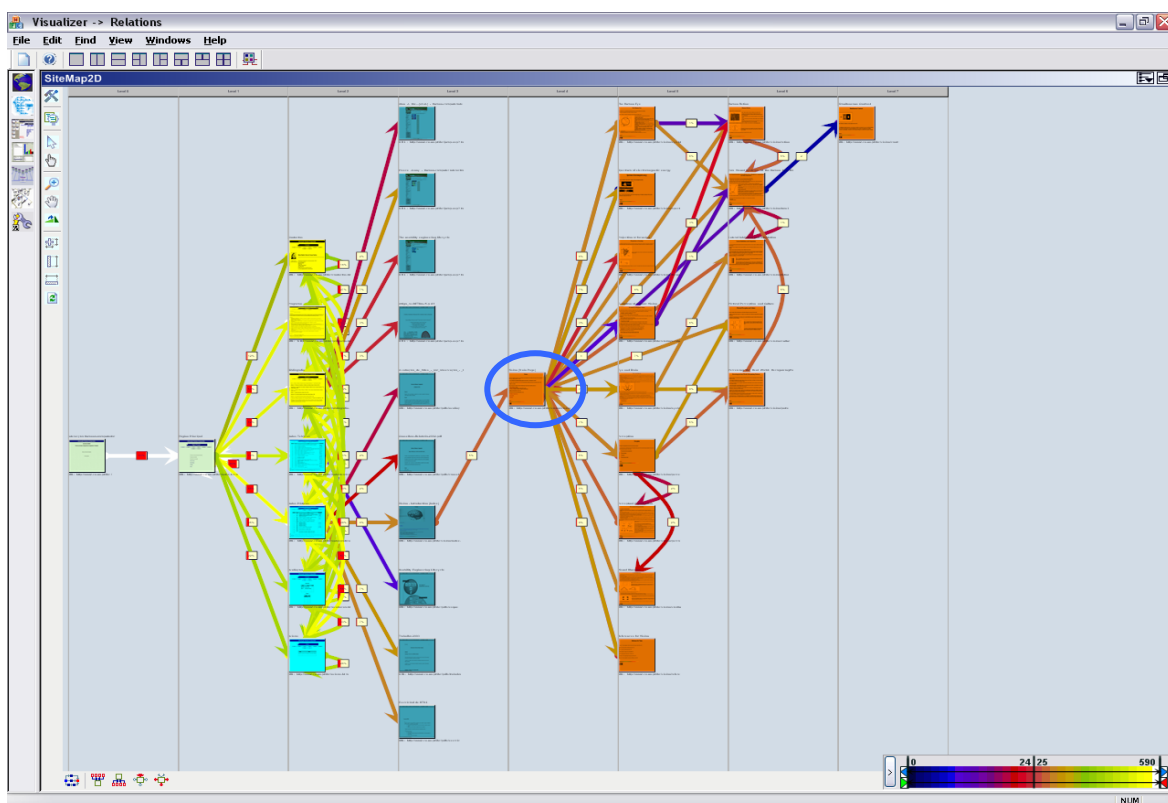


Figure 101 Case Study - Detailed Site Structure

Changing the threshold and mask present on the color LUT reveals some important navigational patterns. The threshold presented in Figure 102 depicts intense backwards navigation from the third to second levels of the site; this is due to the intense connection network in level three. A potential problem might be highlighted by this context: users do not attain their goals and have to go back and restart navigation.

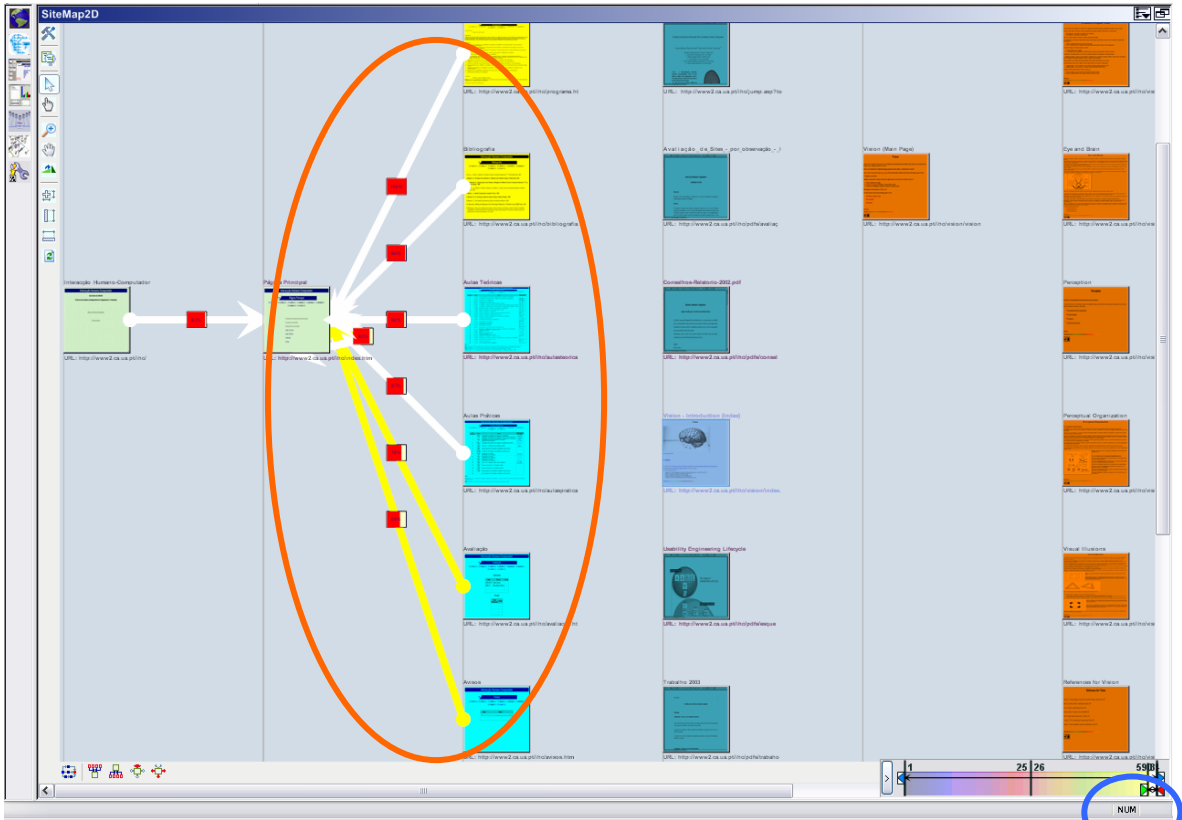


Figure 102 Threshold that depicts intense backwards navigation from the third to second levels

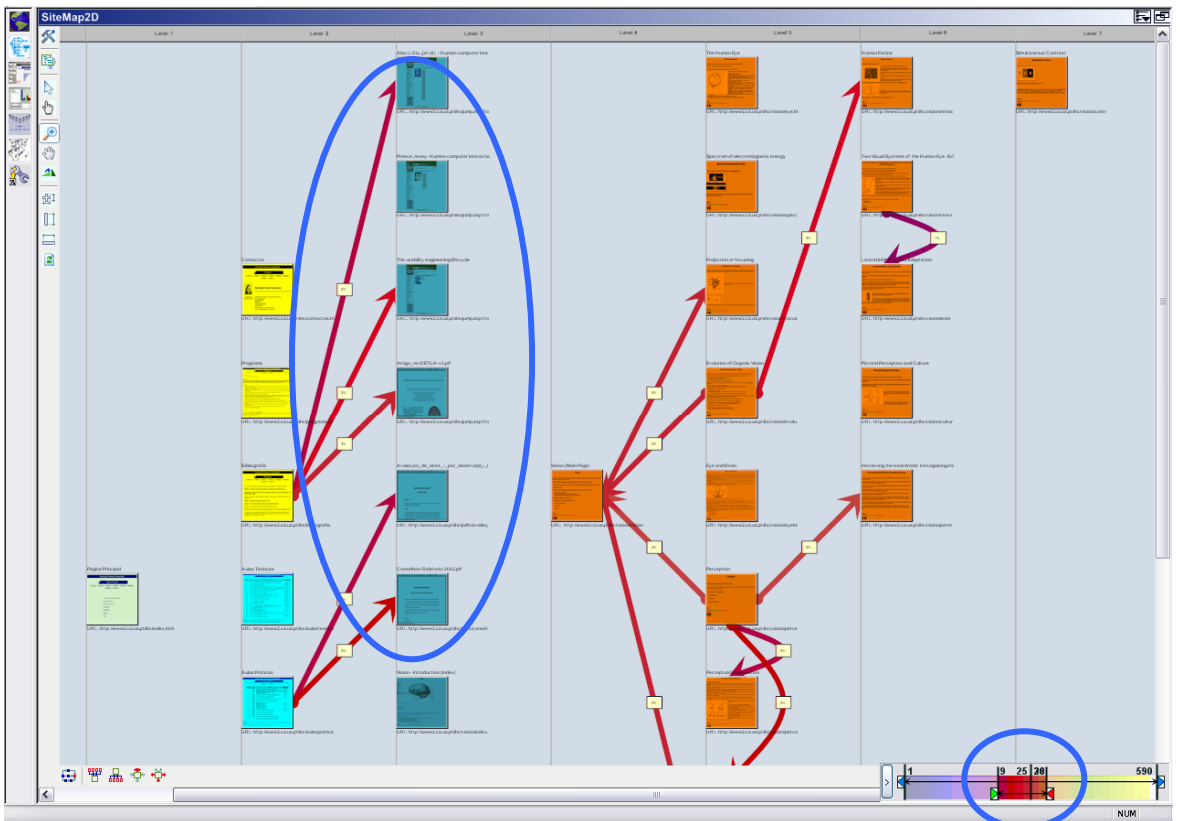


Figure 103 Semantically related contents prove average usage statistics

Figure 103 shows the situation where a threshold is activated to show only the values in the middle of the scale. The action reveals that semantically related contents in level four prove average usage statistics on the logarithmic scale used in this case. A possible explanation might be that the dark green colored pages represent theoretical bibliography in PDF format, less visited by the attendees.

To attain more insight, the visual inspection of website structure needs to be complemented with additional visualizations and inspection mechanisms page contents, layouts and session analysis.

5.5.4.2. Visual inspection of interface design coherence

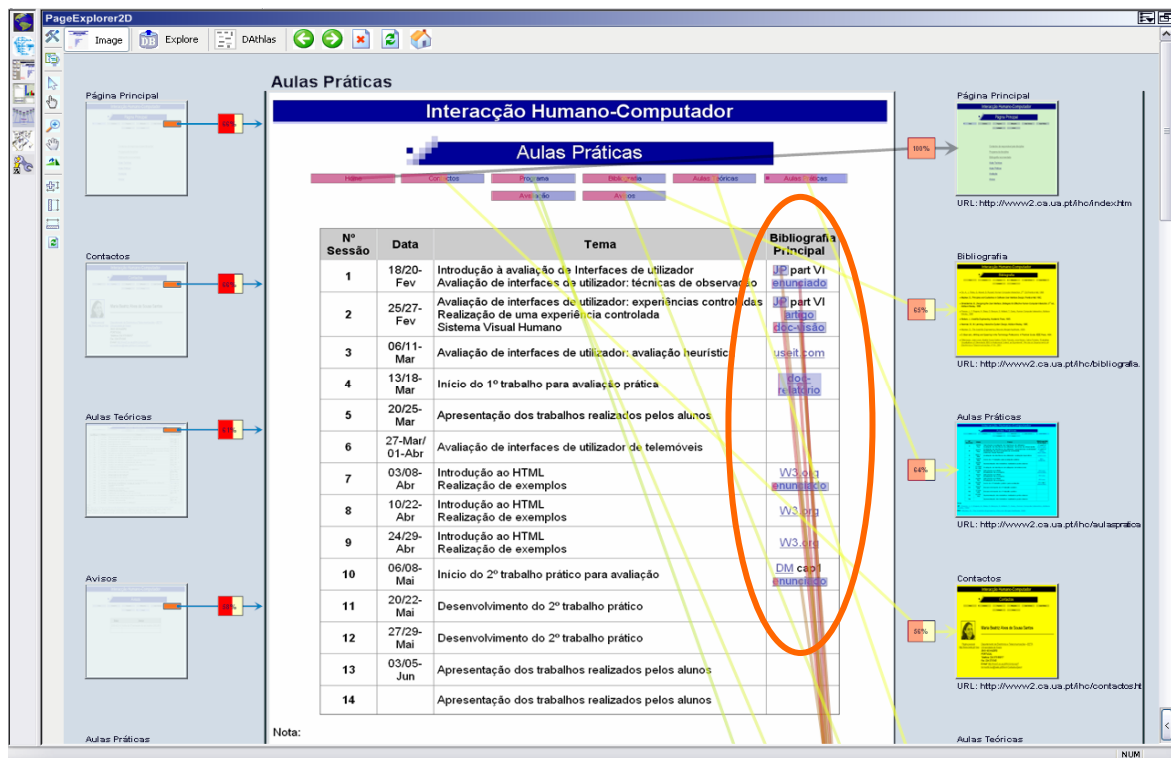


Figure 104 Page inspection for interface design coherence combined with statistical information

Figure 104 represents the visual workspace of the page called “Aulas Práticas”. The navigation of the page (as of the entire third level) is implemented with a top-frame menu containing eight menu items as the number of pages contained in third level. However, the navigation paradigm is broken by a set of hyperlinks situated on the right-most column of the main table of the page. Statistical information is more or less balanced, users showing more interest for “Bibliografia” and “Aulas Práticas” pages. The hyperlinks on the table have little usage, proving the users were not attracted by the linked contents.

Figure 105 presents the case discussed in the previous section, when the navigation is forced to the browser history. This type of navigation can be unpredictable since the page is directly reachable from several pages in levels six, seven and eight.

Figure 106 presents the visual interaction workspace obtained from all hotspots of all pages of the website. It is clearly visible that the interaction areas of the website are concentrated on the top (as a menu represented with darker areas), the central and the left areas of the representation. Darker areas signal more hotspots.

Starting from the idea of interaction workspace and combining usage information obtained from session requests, we selected three sets of user groups as three scenarios for analysis purposes. These three groups represent general students within the local University network, general students outside the University network and a third group represented by professors and site administrators. Figure 107, Figure 108 and Figure 109 present the results for analysis of visual interaction workspace behaviors for the selected user groups.

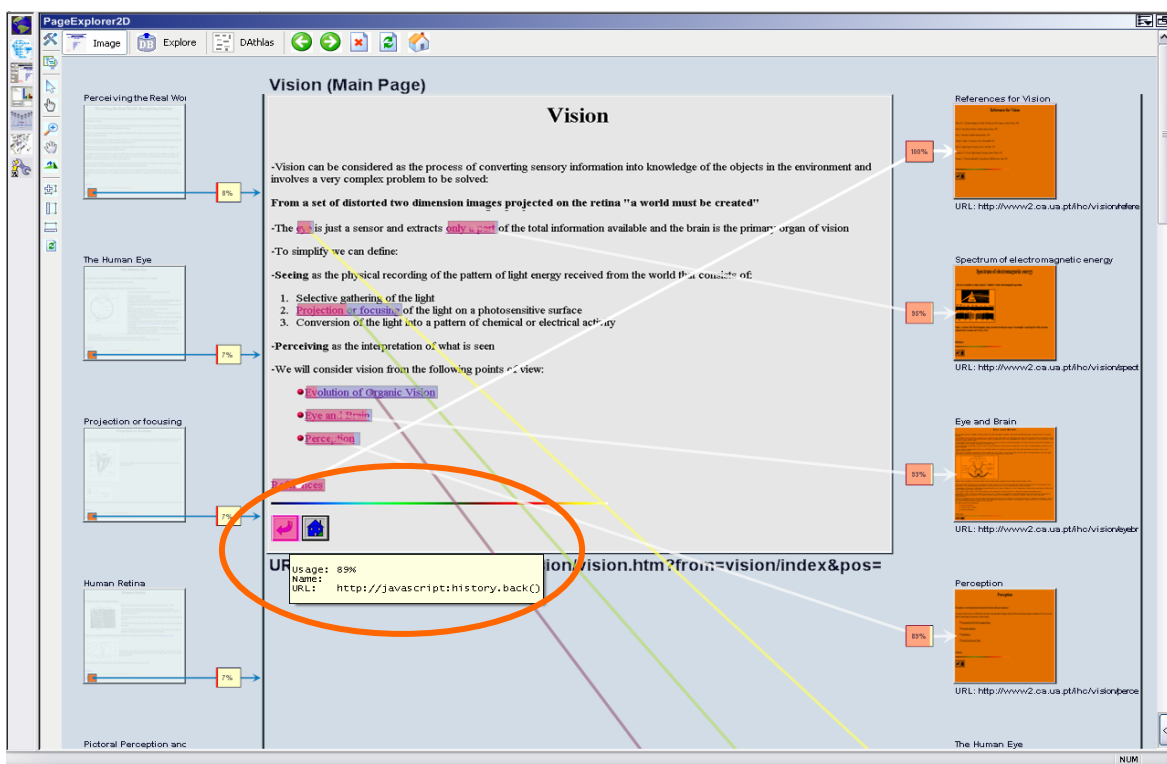


Figure 105 Possible navigational problem with the usage of browser history

All representations used the statistically derived inter page jumps, represented as transparent lines from one hotspot location to another (the less transparent and the wider the line, more users selected the hotspots for navigation). These visual clues reveal that students were navigating more efficient by using the “T” like structure present on the center of the figures (Index page), combined with intensive navigation on the bibliography hyperlinks provided as the pages “Aulas Téóricas” and “Aula Práticas”. At the same time, professors and administrators of the site had a more robust navigational pattern, directed to “Bibliography” pages.



Figure 106 Website visual interaction workspace

The left side of Figure 106 reveals an interaction area of average intensity, but the representations that use statistically derived information (Figure 107, Figure 108, Figure 109, and Figure 110) show no or few interest for this area. This might happen because of the semantics of the related areas or might depict a serious navigational problem.

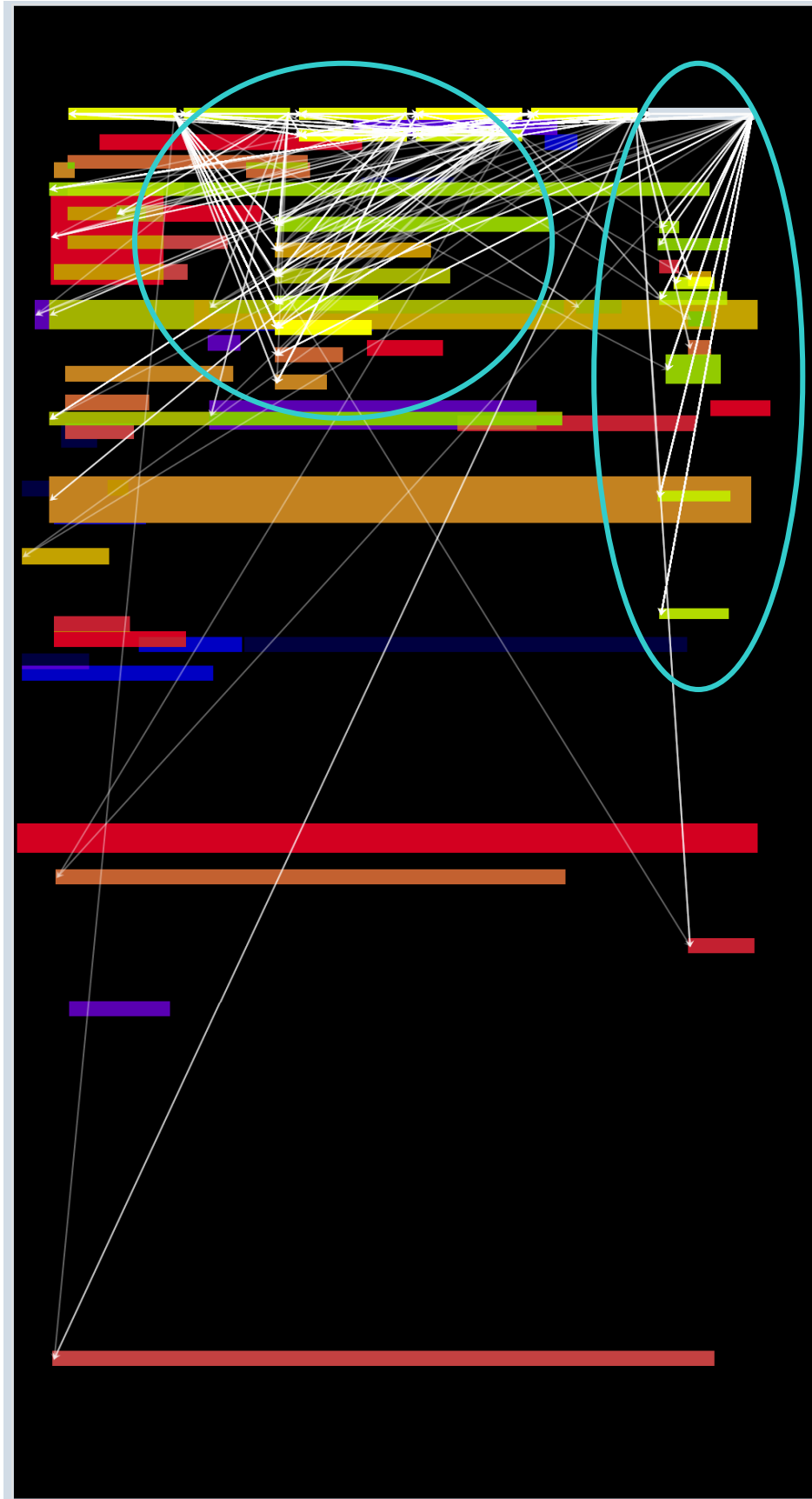


Figure 107 Visual workspace coherence for users inside the University network

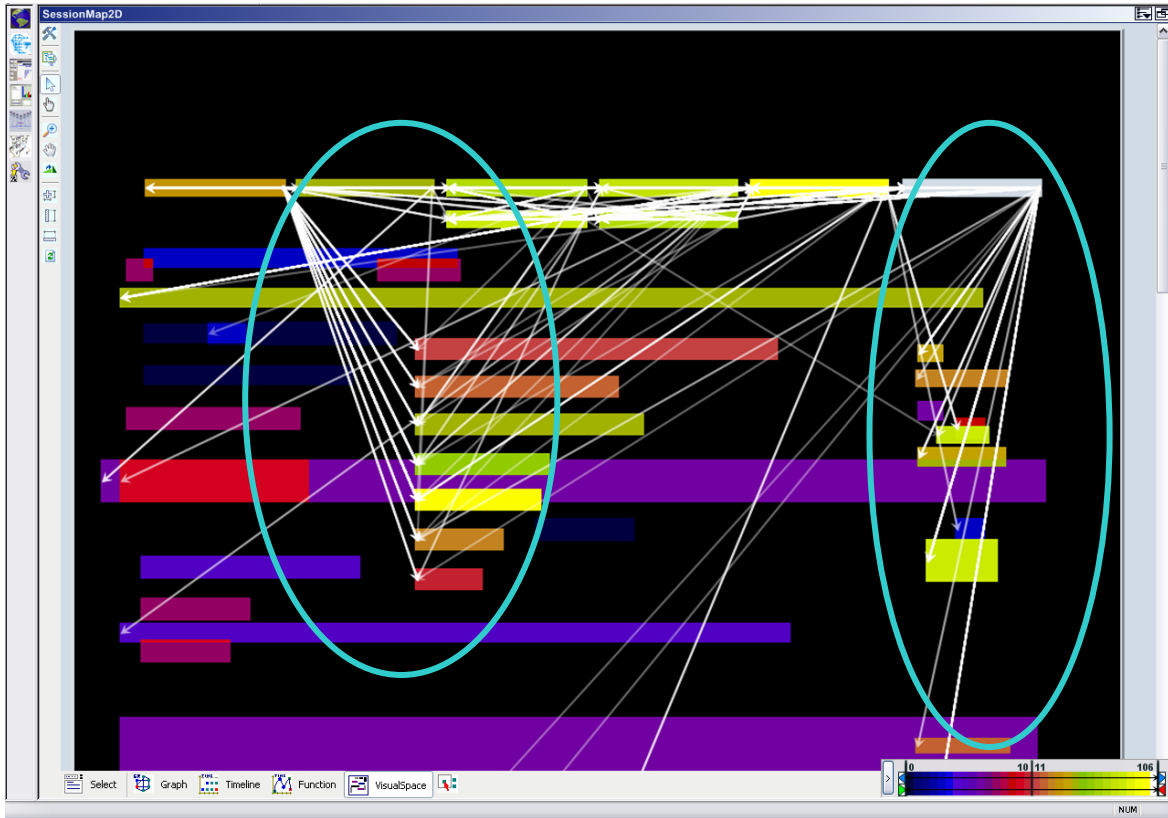


Figure 108 Visual workspace coherence for users outside the University network

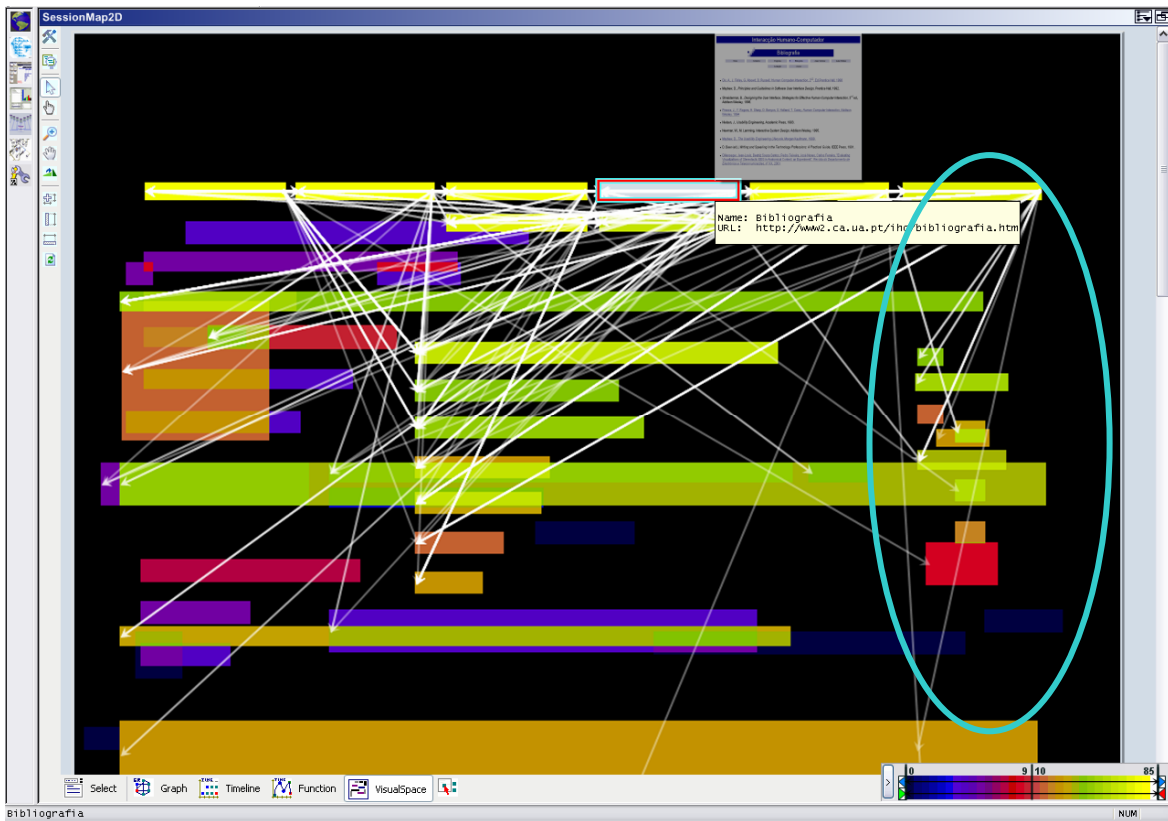


Figure 109 Visual workspace coherence for professors and administrators

Figure 110 represents the visual workspace coherence for all users of the website, for all analyzed session. It is clearly visible that the navigational paradigm is not shared when it comes to different user profiles with different goals. However, the representation demonstrates that the interaction paradigm of the website is not clearly concentrated on some specific areas like menus, toolbars, etc.; moreover, the representation in Figure 107 confirms that the existence of “long” page breaks the organization of the navigational space provided to the user (the “longer” the page is, the less visited are the areas on the bottom of the page).

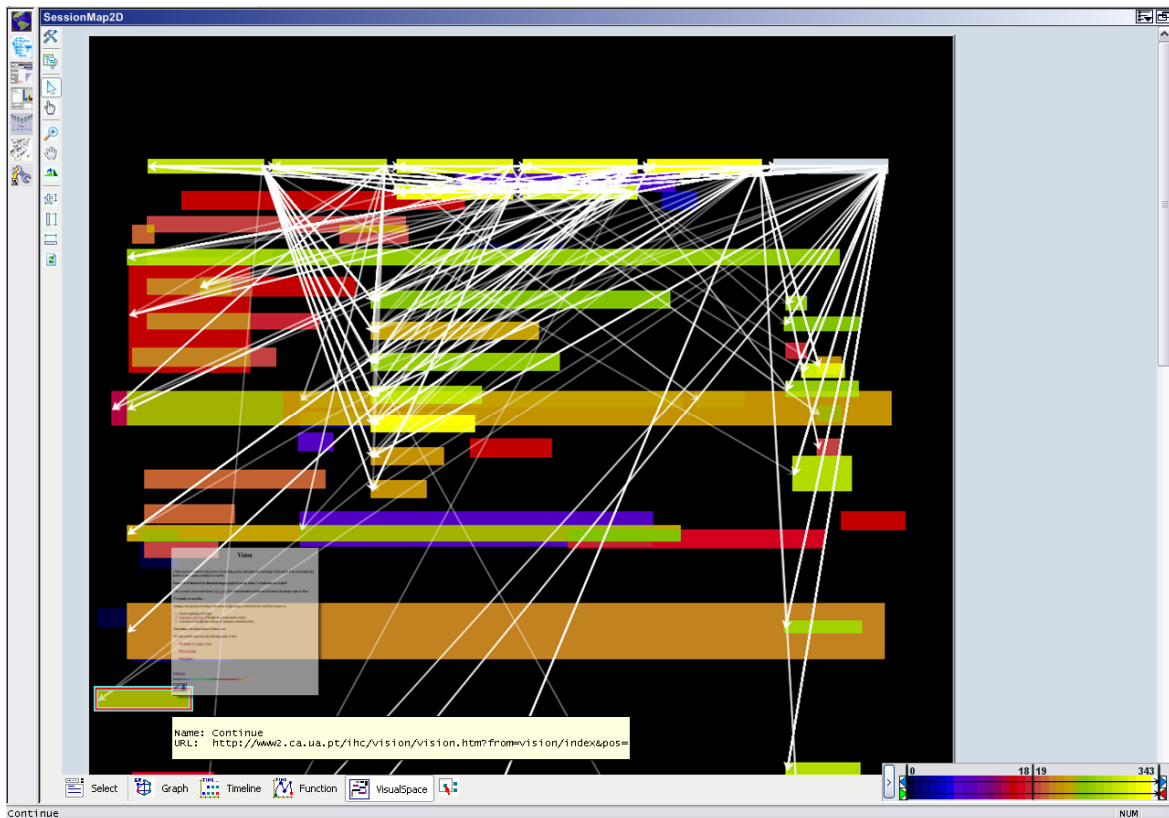


Figure 110 Visual workspace coherence for all users and all sessions

5.5.4.3. Inspection of tree-structured information traversing in time

Tree-structured information traversing in time can be used to detect navigation anomalies. The problem of using java script to force browser history navigation (discussed in section Visual inspection of website structure and efficiency, represented in Figure 105) is visually represented in Figure 111. Navigation out of context is coded as yellow squares because of the usage of back buttons present on the pages in the sixth level. The user has to go back to a parent page every time he/she needs to proceed to a subsequent level.

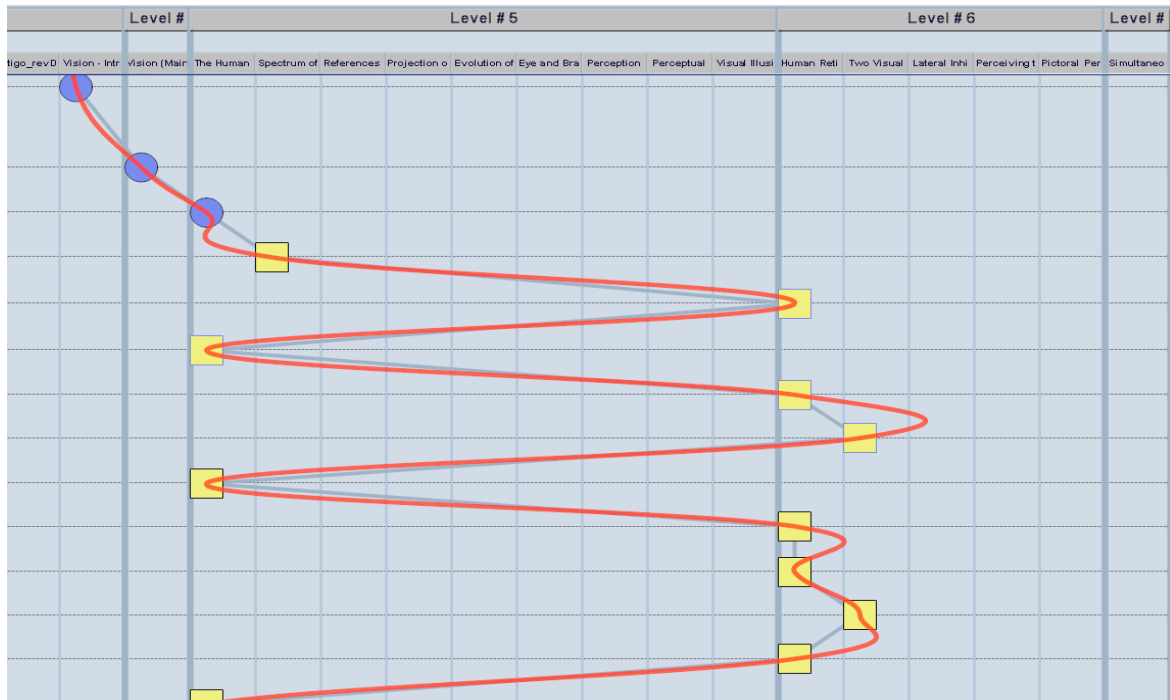


Figure 111 Browser's history back navigation example

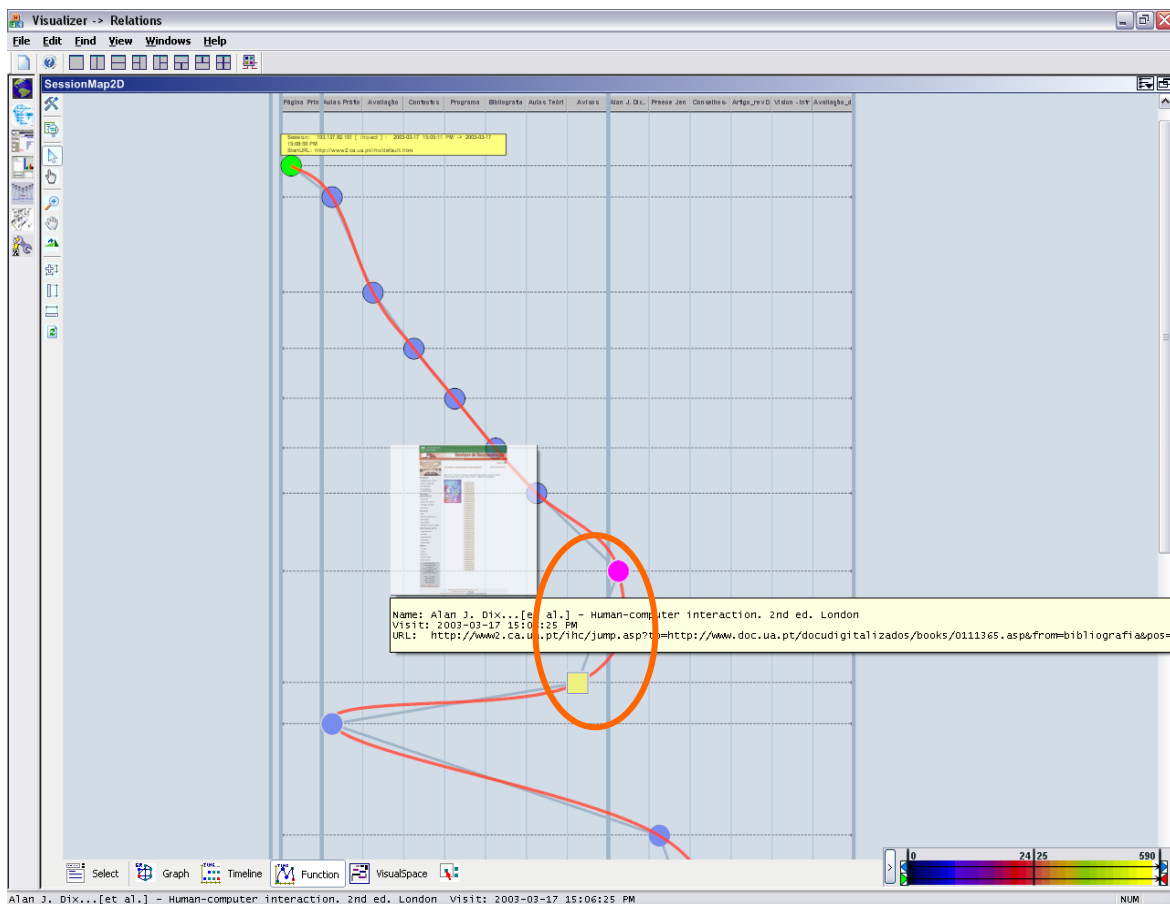


Figure 112 Backward navigation example on tree-structured traversing in time

Figure 112 highlights a navigation to and from outside the context of the analyzed site. The purple circle represents a jump to a page hosted on an external domain (library domain); the request that followed was clearly selected from the browser history or manually typed, since it is not present on the session history.

Another backward navigation example is presented in Figure 113, as the request from the seventh level is followed by a request on the second level, most probably selected from the browser history. Following requests navigate among the highly connected pages on the second level, as discussed in section Visual inspection of website structure and efficiency.

A deeper analysis of several usage sessions revealed that most users behave in similar way: they visited the second level until they access “Aulas Teóricas” and “Aulas Práticas” associated pages, read some of the referenced documents, and then visited the page from the semantic area called “Vision”.

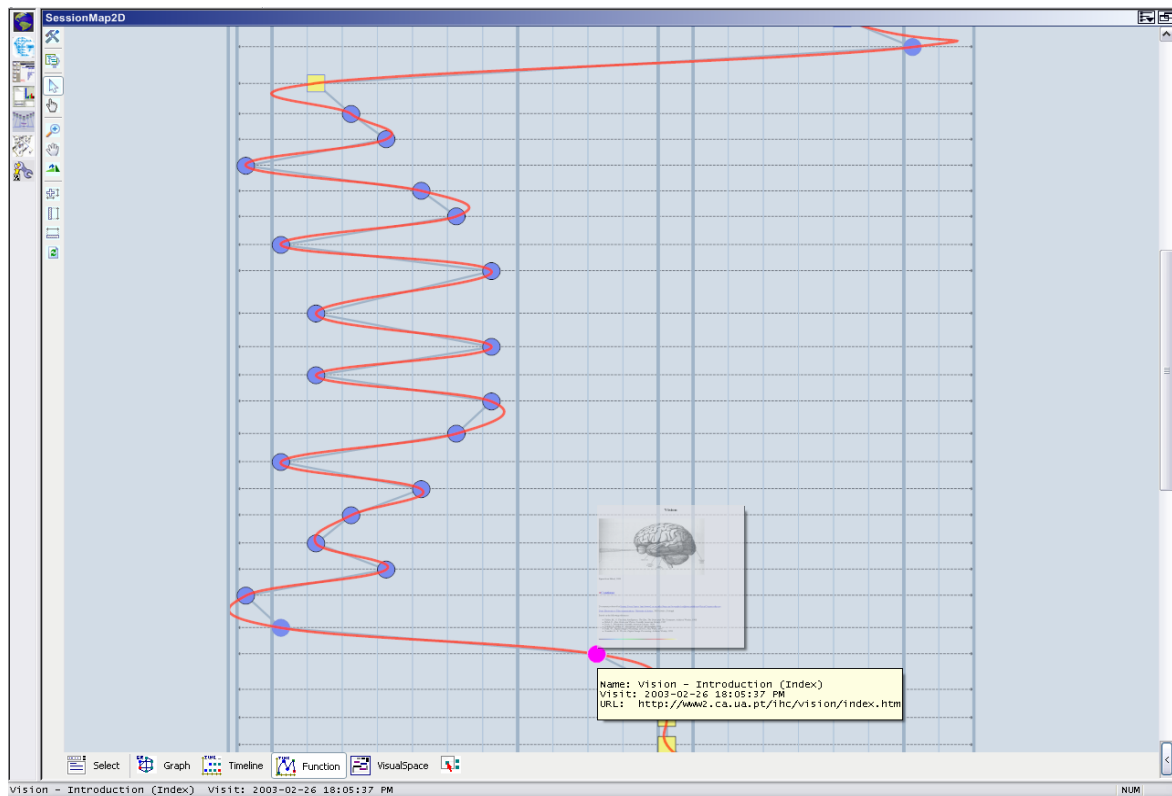


Figure 113 Backward navigation to highly connected pages

Chapter 6. Conclusions and Future Work

6.1. Summary

We introduced the latest evolution of a research spin-off application under development intended to help usability experts analyze usage patterns of an institutional website. Our proposal is especially concerned with the representation of the collected data, mainly related with the website structure, contents, and site usage logged information (either obtained during natural site usage or controlled experiments). We present the information using specific visualization methods exploring the enormous capacities of the human visual system.

Our research framework involves the representation of the information based on attributes of web pages, page elements, and the statistical information. The addition of user interactions and exploration capabilities with visual representations is expected to improve the overall insight of the presented data. Innovative visualizations and user interface interaction mechanisms had to be developed to comply with the requirements to support our ideas, some based on the existing concepts, others completely new.

We proposed an application organized in four main modules: the Site Analyzer, the Interceptor, the Compiler and the Visualizer. The Visualizer is the most complex module and therefore, a detailed version of the application is presented in this work. It offers the user a rich functionality and a choice among several synchronized representations of the data obtained using different visualization methods.

The application's design and implementation was performed in two steps:

- the former involved the implementation of the analysis algorithms, definition of the preliminary data structures and basic functional requirements;
- the latter involved the implementation of a completely new application supported by a more flexible framework and by new design patterns, meant to optimize the analysis procedures.

The user interface introduced by the proposed application, includes a multiple view option for each of its analysis areas/schemes resulting in a coherent and synchronized visualization solution that offers a simultaneous information inspection set of tools. The visualization schemes described represent some of the solutions for these inspection tools. Some of the preliminary schemes undergoing test, integration and optimization, such as the hot-spot usage with path representation of usage pattern and the discrete function obtained from tree-structured information traversing in time, require further investigation.

We also described the evaluation of our proposed visualization application, starting by an informal evaluation and proceeding with a more structured evaluation. We asked two types of users to perform a set of tasks meant mainly to evaluate the usability of certain user interface features and of the interaction mechanisms of the visualizations. As a consequence, some of the visualization schemes and user interface aspects had to be redesigned. These evaluations also provided feedback that resulted in new ideas to improve our application. Moreover, we were encouraged by the preliminary results obtained with both the students and the professionals.

6.2. Conclusions

This section highlights the main contributions of this work among with the pros and cons we identified along the conceptualization and development of the prototype.

This works produced several important contributions:

- The approach to combine quantitative and qualitative measures for the analysis of website structure, contents and usage into a unique integrated visual analysis interface, meant to facilitate institutional websites maintenance to content managers;
- The transformation of raw data into visual features (using visual transformations), mean to provide visual clues for the information analyzed;
- Several innovative or improved adapted versions of inspection tools and visual synchronization mechanisms as *Interactive Zones*, *Page Relations*, *Hovering Tips*, *Interaction Workspace*, *Tree-Structured Traversing in Time* visualizations or *Visual Workspace Coherence*, meant to facilitate the overall experience of the application;
- The unified user interface with multiple synchronized views, meant to improve the perception of complementary related information represented in each view. The innovation comes from the interaction effects propagated simultaneously to all views, but with complementary different visual effects on each one of them;
- The new website/portal maintenance framework, *SharePoint Designer and Analyzer*, capable to perform not only the analysis of a portal, but also to reorganize and update its structure. The workflow Reverse Engineer → Visualize → Reorganize → Update is very important for the maintenance of large and dynamic hypermedia structures as portals, based on dynamic visual configurations of structure and contents.

Several innovative visualization methods are important for the topics of this work, as follows:

- Path to Goal – Interconnections (SelectionExplorer2D) highlight a possible solution to discover the shortest path from one point to another (less clicks does not necessarily means less time);

- Interactive Zones and Page Relations (PageExplorer2D) is definitely a useful navigational and page design analysis tool, the workflow Referrer → Page → Children being of a great importance while looking for insight on navigational patterns;
- Visual Workspace Coherence (SessionMap2D – VisualSpace) is probably the most important page design analysis tool, combining web page structural information with usage information, to foster insight on visual workspace coherence and discovery of usage patterns based on visual clues;
- Inspection of tree-structured information traversing in time seems a promising approach for the prospective analysis of usage behavior, to discover possible navigational inconsistencies due to website structure and/or design.

Given the complexity of the concepts involved in this work and the amount of time we spent with the development and evaluation process, we can conclude that we managed to establish the basis for a website analysis framework, from both points of view communication / information sharing infrastructure and its understanding. Many of the components introduced in this work are in a preliminary phase, further investigations being the challenge for our actual implementation.

Probably, the most important limitation of the prototype was the inability to automatically address dynamic page contents or personalization scenarios, in addition to the semantic classification of website contents. Thus, these aspects were considered as secondary preoccupations for the purpose of this thesis, given the fact that only the concept of crawling a website can be considered an important research topic by its own.

From the prototype evaluation point of view, the most demanding tasks involved the validation of the visualization schemes. Valuable feedback was not easy to obtain, given the complexity of the involved concepts. On the contrary, the evaluation of the user interface aspects was straightforward.

Some of the most important evaluation issues that we noticed were:

- During the preliminary evaluation sessions we realized that three-dimensional representations were actually only eye-candy, however they are not easy to navigate or suitable for information retrieval and inspection. One of the reasons might have been the poor navigational features provided, not intuitive, nor standardized, combined with the increased complexity of three-dimensional representations;
- Yet, another negative aspect of the evaluation was the amount of work required for the preliminary data preparation phase. It turned out to be a bottleneck to the usage of the application for the analysis of medium to large websites. Semi-automated database synchronized entries slowed down the overall process.

Finally, the most important achievement of this work can be considered the conceptual organization and finalization of a research, development and evaluation cycle of a complex website analysis system.

6.3. Future Work

To continue the development and evaluation of the application introduced in this thesis, we identified the following topics as headlines of our research and development agenda:

- finalize the implementation of the components less explored by this work (some because of the priorities we associated to each component, others because of the complexity of the concepts involved);
- update the visualization components by introducing and implementing new methods of user interaction, manipulation and identification of the visualized information;
- further investigate visualization methods, their relations and application to the real world environments:
 - hot-spot and usage path overlapped visualization needs additional dynamic inspection tools such as a visualization slider to segment the low usage representations to a certain level. This might help the inspection of the most common navigation paths, and is expected to reveal interface design ambiguity or mismatch with design goals. This interface design inspection scheme should probably include distance between starting page and goal page, information that can be helpful to understand effort and efficiency or to highlight possible optimizations;
 - the tree-structured information traversing in time scheme seems a promising approach for the prospective analysis of usage behavior during tree-structured information traversing;
- Update the second version of Site Analyzer to address technologies other than SharePoint, and to synchronize the collected information with the database;
- identify new application requirements by organizing more evaluation sessions with the expected application users as information managers, designers, web masters, etc.. Empirical evaluation sessions might be useful in a later moment of the prototype development lifecycle:
 - perform more evaluation sessions with the professionals to identify how the proposed visualization schemes respond to their needs, as they become more experienced users, within the functionality of our application;

- evaluate the visualizations concerning the visual representation only. Until now, we have judged informally the complexity of the representation and the time it takes to rebuild it after user interaction;
- evaluate the adequacy of some aspects (e.g. colors, icons, spatial organization and coherence) with a graphics designer that would help identify possible design problems.

Chapter 7. Bibliography

- [Andrews1999] Andrews Keith. 1999. *Visualising Cyberspace: Information Visualisation in the Harmony Internet Browser*, In Readings in Information Visualization: Using Vision to Think, (editors) Stuart K. Card, Jock D. Mackinlay and Ben Shneiderman. San Diego, CA, Morgan Kaufman, pp 493-502.
- [Andrews2002] Andrews Keith. 2002. *Visualization Notes*. IEEE Symposium on Information Visualization (InfoVis).
- [AWStats2005] AWStats Log Analyzer. 2005.
Online at: <http://awstats.sourceforge.net>; last visit: November 2005.
- [Barlow2001] Barlow T., Neville P. *A Comparison of 2D Visualizations of Hierarchies*. IEEE Symposium on Information Visualization, InfoVis01, pp 131-138.
- [Becker1999] Becker R.A., Eick S.G. and Allan R. Wilks. 1999. *Visualizing Network Data*, In Readings in Information Visualization: Using Vision to Think, (editors) Stuart K. Card, Jock D. Mackinlay and Ben Shneiderman. San Diego, CA, Morgan Kaufman, pp 215-230.
- [Bederson2003] Bederson B., Shneiderman B. 2003. *The Craft of Information Visualization, Readings and Reflections*. Morgan Kaufman.
- [Benford1999] Benford Steve, Taylor, I., Brailsford, D., Koleva, B., Craven, M., Fraser, M., Reynard, G., and Greenhalgh, C. 1999. *Three Dimensional Visualization of the World Wide Web*. ACM Comput. Surv. N°31, pp 25.
- [Bieber1997] Bieber M., Vitali F., Ashman H., Balasubramanian V. and Oinas-Kukkonen H. 1997. *Fourth generation hypermedia: some missing links for the World Wide Web*, in International Journal Human-Computer Studies, N°47, pp 31-65.
Online at: <http://ijhcs.open.ac.uk/bieber/bieber.pdf>; last visit: July 2006.

- [Booch1998] Booch G. et al. 1998. *The Unified Modeling Language User Guide*. Reading (MA): Addison-Wesley, New York, USA.
- [Brath1999] Brath R. 1999. *Concept Demonstration Metrics for Effective Information Visualization*. IEEE Symposium on Information Visualization, InfoVis97. pp 108-111.
- [Card1999] Card S. K., Mackinlay J., Shneiderman B. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufman.
- [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). ACM Press, New York, NY, pp 498-505.
- [Ceri2000] Ceri S., P. Fraternali, and A. Bongio. 2000. *Web Modeling Language(WebML): a Modeling Language for Designing Websites*. Proceedings of WWW9 Conference, Amsterdam.
- [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). WebDB '04, vol. 67. ACM Press, New York, NY.
- [Chi1998] Chi Ed H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R., and Card, S. K. 1998. *Visualizing the evolution of web ecologies*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Los Angeles, California, United States, April 18 - 23). Editors: C. Karat, A. Lund, J. Coutaz, and J. Karat, Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, pp 400-407.
- [Chi1999] Chi Ed H. 1999. *A Framework for Information Visualization Spreadsheets*. PhD Thesis.
- [Chi2002] Chi Ed H. 2002. *Improving web usability through visualization*. IEEE Internet Computing 6, 2 (March), pp 64-71.

- Online at: <http://dx.doi.org/10.1109/4236.991445>; last visit: April 2006.
- [ClickTracks2005] ClickTracks Web Analyzer. 2005.
- Online at: <http://www.clicktracks.com/products/analyzer/>; last visit: December 2005.
- [Cockburn1997] Cockburn A, Steve J. 1997. *Design Issues for World Wide Web Navigation Visualization Tools*. Proceedings of RIAO'97: The Fifth Conference on Computer-Assisted Research of Information. McGill University, Montreal, Quebec, Canada, June.
- Online at:
<http://www.cosc.canterbury.ac.nz/andrew.cockburn/papers/riao97.pdf>.
- [Cooley2003] Cooley R. 2003. *The use of web Structures and Content to Identify Subjectively Interesting web Usage Patterns*. ACM Trans. Inter. Tech. 3, 2 (May), pp 93-116.
- [Cugini1999] Cugini J, Scholtz J. 1999. *VISVIP: 3D Visualization of Paths through websites*. In Proceedings of the 10th international Workshop on Database & Expert Systems Applications (September 01 - 03). DEXA. IEEE Computer Society, Washington, DC, pp 259.
- [Dahlback1993] Dahlback N., Jonsson A., and Ahrenberg L. 1993. *Wizard of oz studies - why and how*. In W Gray, W. E. Heey, and D. Murray, editors, Proceedings of the 1993 International Workshop on Intelligent User Interfaces. Association of Computing Machinery, Inc.
- [Deep2005] Deep Log Analyzer. 2005.
- Online at: <http://www.deep-software.com/default.asp>; last visit: December 2005.
- [Di Lucca2002] Di Lucca G. A., A. R. Fasolino, F. Pace, P. Tramontana, U. de Carlini. 2002. *WARE: a tool for the Reverse Engineering of web Applications*. csmr, p. 0241, Sixth European Conference on Software Maintenance and Reengineering.
- [Dix1998] Dix A. et al. 1998. *Human Computer Interaction*, 2nd. Prentice-Hall, London, England.

- [Dodge2003] Dodge M. 2003. *The Atlas of Cyberspace: Maps of websites*.
Online at: http://www.cybergeography.org/atlas/web_sites.html; last visit: December 2005.
- [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.
- [Eick2004] Eick G. Stephen. 2004. *Visualizing Online Activity*. Commun. ACM 44, 8 (Aug. 2001), 45-50.
- [Ethnio2005] Ethnio. 2005.
Online at: <http://www.boltpeters.com/ethnio/index.html>; last visit: December 2005.
- [Faraday2000] Faraday P. 2000. *Visually Critiquing web Pages*. Proceedings of HFWeb'00 (Austin, TX, June).
Online at: <http://www.tri.sbc.com/hfweb/faraday/faraday.htm>; last visit: April 2006.
- [FastStats2005] FastStats. 2005.
Online at: <http://www.mach5.com/products/analyzer/index.html>; last visit: December 2005.
- [Fraternali2003] Fraternali P, Matera M., Maurino A. 2003. *Conceptual-level log analysis for the evaluation of web application quality*. First Latin American Web Congress (LA-WEB'03), pp 46.
- [Freitas2002] Freitas C. S., Luzziardi P., Cava R., Winckler M., Pimenta M., Nedel L. *Evaluating Usability of Information Visualization Techniques*. Proceedings 5th Symposium on Human Factors in Computer Systems IHC2002, Fortaleza, Ceará.
- [GIS2005] Geographic Information Systems. 2005. U.S. Geographical Survey.
Online at: http://erg.usgs.gov/isb/pubs/gis_poster/; last visit: May 2006.
- [Grinstein] G. Grinstein, P. Hoffman, S. Laskowski, R. Pickett. *Benchmark*

- Development for the Evaluation of Visualization for Data Mining.*
 Online at: <http://home.comcast.net/~patrick.hoffman/VIZ/benchmark.pdf>; last visit: March 2004.
- [Grinstein2002] Grinstein G., Hoffman P. E. and Pickett R. M. Benchmark. 2002. *Development for the Evaluation of Visualization for Data Mining.* In Fayyad, U., Grinstein, G. G., et al. (eds.), *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann, San Francisco, pp 129-176.
- Online at: <http://home.comcast.net/~patrick.hoffman/VIZ/benchmark.pdf>; last visit; June 2006.
- [Google Analytics2005] Google Analytics. 2005.
- Online at: <http://www.google.com/analytics/index.html>; last visit: December 2005.
- [Healey2001] Healey C. G., Amant St. R., and Chang J. 2001. *Assisted Visualization of E-commerce Auction Agent.* Presented at Graphics Interface 2001 – Canadian Human-Computer Communications Society, pp. 201-208, 7-9 June.
- [Heer2002] Heer Jeffrey, Ed H. Chi. 2002. *Separating the Swarm: Categorization Methods For User Sessions On The Web.* Proceedings of CHI2002.
- [Hix1999] Hix D., J. E. Swan II, Gabbard J., McGee M., Durbin J., King T. 1999. *User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment.* In Proceedings IEEE Virtual Reality 99. pp 96-103.
- [Jacobson1999] Jacobson I., Booch G., and Rumbaugh J., 1999., *The Unified Software Development Process.* Addison-Wesley, New York, USA.
- [IFABC2006] IFABC Global Web Standards.
- Online at: <http://www.ifabc.org/standards.htm>; last visit: March 2006.
- [Ivory2002] Ivory M. Y., Hearst M. 2002. Improving website design. *IEEE Internet Computing*, v.6 n.2, p.56-63, March 2002.
- [Ivory2002] Ivory M. Y., Hearst M. 2002. *The State of the Art in Automated Usability Evaluation of User Interfaces.* *ACM Computing Surveys (CSUR)*, v.33 n.4, p.470-516, December.

- [ISAServer2004] Microsoft ISA Server. 2004. Microsoft Internet Security and Acceleration Server.
- Online at: <http://www.microsoft.com/isaserver/evaluation/overview/default.msp>,
Last visit: December 2005.
- [IWEBTRACK2005] IWEBTRACK. 2005. Online at: <http://www.iwebtrack.com/>; last visit: December 2005.
- [Keim2001] Keim A. D. 2001. *Visual Exploration of Large Data Sets*. Communications of the ACM, vol. 44(8), pp. 38-44.
- [Kobsa2001] Kobsa A. 2001. *An Empirical Comparison of Three Commercial Information Visualization Systems*. IEEE Symposium on Information Visualization, InfoVis01. pp 123-130.
- [Larman1998] Larman C. 1998. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. Prentice-Hall.
- [LiveSTATS2005] LiveSTATS Deepmetrix. 2005.
- Online at: <http://www.deepmetrix.com/>; last visit: November 2005.
- [Martin2001] Martin Johannes, Ludger Martin. 2001. *Website Maintenance With Software-Engineering Tools*. 3rd International Workshop on Web Site Evolution (WSE'01), pp 126.
- [Mealha2004] Mealha Ó, Sousa Santos B., Nunes J, Zamfir F. 2004. *Integrated Visualizations for an Information and Communication Web Log Based Management System*. Proceedings of International Conference of Information Visualization – IV04, London. July.
- [MSDN2004] MSDN. February 2004. *Enterprise Development, User Interface Design and Development, Graphics and Multimedia*. Microsoft.
- [Munzner1997] Munzner T. 1997. *H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space*. In Proceedings of the 1997 IEEE Symposium on information Visualization (infovis '97) (October 18 - 25). INFOVIS. IEEE Computer Society, Washington, DC, 2.
- [Najork2001] Najork M, Wiener I. J. 2001. *Breadth-First Search Crawling Yields High-Quality Pages*. In Proceedings of the 10th International World

- Wide Web Conference. Hong Kong. pp 114-118.
- [Nielsen2001] Nielsen J. 2001. *Did Poor Usability Kill E-Commerce?*. Alertbox, 19 August.
- Online at: <http://www.useit.com/alertbox/20010819.html>; last visit: June 2006.
- [Nielsen1993] Nielsen J. 1993. *Usability Engineering*. Academic Press. 1993.
- [Niu2003] Niu Y., Zheng T., Chen Z., Goebel R. 2003. *WebKIV - Visualizing structure and navigation for web mining applications*. IEEE/WIC International Conference on Web Intelligence (WI'03), pp 207.
- [Nomura2002] Nomura S, Oyama S., Hayamizu T., Ishida T. 2002. *Analysis and Improvement of HITS Algorithm for Detecting Web Communities*. Proceedings of the 2002 Symposium on Applications and the Internet (SAINT'02).
- [Nunes2003] Nunes J., Zamfir F., Mealha Ó., Sousa Santos B. 2003. *Web LogVisualizer: A Tool for Communication and Information Management*. Proceedings of the 10th International Conference Human-Computer Interaction – HCI International 2003, Vol. 3 (Human-Centred Computing: Cognitive, Social and Ergonomics Aspects), Crete-Greece. 824-828, June.
- [Nunes2006] José Nunes. 2006. *Visualização de Interação em Cenários de Comunicação Humano-Computador*, Tese de Doutoramento (Versão provisória), Universidade de Aveiro.
- [North2000] North C., Schneiderman B. 2000. Snap-Together: *Can Users Construct and Operate Coordinated Views?* Int. Journal Human-Computer Studies, 53, 5. 715-739.
- [OMNIWEB2004] OMNIWEB for Apple, version 5.0, Apple Inc., 2004.
- Online at: <http://www.omnigroup.com/applications/omniweb/>; last visit: July 2006.
- [Opentracker2005] Opentracker. 2005.
- Online at: <http://www.opentracker.net/index.jsp>; last visit: December 2005.

- [Paganelli2002] Paganelli L., Paternò F. 2002. *Intelligent analysis of user interactions with web applications*. In Proceedings of the 7th international Conference on intelligent User interfaces (San Francisco, California, USA, January 13 - 16, 2002). IUI '02. ACM Press, New York, NY, 111-118.
- [Ricca2000] Ricca F., Tonella P. 2000. *We Site Analysis: Structure and Evolution*. IEEE.
- [Ricca2001] Ricca F., Tonella P. 2001. *Understanding and Restructuring websites with ReWeb*. IEEE MultiMedia 8, 2 April, 40-51.
- [Ruffo2004] Ruffo G., R. Schifanella, M. Sereno. 2004. *WALTy - a user behavior tailored tool for evaluating web application performance*. Network Computing and Applications, Third IEEE International Symposium on (NCA'04), pp. 77-86.
- [Santos2004] Sousa Santos B., Zamfir F., Ferreira C., Mealha Ó., Nunes J. 2004. *Visual Application for the Analysis of Web-Based Information Systems Usage: A Preliminary Usability Evaluation*. Proceedings of International Conference of Information Visualization – IV04, London. July.
- [Sawmill2005] Sawmill. 2005.
Online at: <http://www.sawmill.net/>; last visit: December 2005.
- [Sebrechts1999] Sebrechts M., Vasilakis J., Miller M., Cugini J., Laskowski S. 1999. *Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces*. ACM Conf. Research and Development in Information Retrieval, ACM SIGIR 99, 3-10, California.
- [Shneiderman1996] Shneiderman B. *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. Presented at IEEE Symposium on Visual Languages, Bouldes, Colorado, USA, pp. 336-343, 3-6 September.
- [Schroeder1998] Schroeder W., K. Martin, B. Lorensen. 1998. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice Hall.
- [Sitelogz2005] Sitelogz. 2005.
Online at: <http://www.sitelogz.com/sitelogz/index.php>; last visit:

December 2005.

- [Spence2001] Spence R. 2001. *Information Visualization*. Addison-Wesley, 2001.
- [Spiliopoulou2000] Spiliopoulou M. 2000. *Improving the Effectiveness of a website with web Usage Mining*. In Revised Papers From the international Workshop on Web Usage Analysis and User Profiling B. M. Masand and M. Spiliopoulou, Eds. Lecture Notes In Computer Science, vol. 1836. Springer-Verlag, London, 142-162.
- [STATISTICA1999] STATISTICA for Windows, version 5.5, StatSoft Inc., 1999.
Online at: <http://www.statsoft.co.uk/>; last visit: July 2006.
- [Tauscher1997] Tauscher L. and Greenberg S. 1997. *How people revisit web pages: empirical findings and implications for the design of history systems*. Int. J. Human-Computer Studies, 47, pp 97-137.
- [Webtrends2005] Webtrends. 2005.
Online at: <http://www.webtrends.com/Products/WebTrends7.aspx>; last visit: December 2005.
- [Wiss1998] Wiss U., Carr D., Jonsson H. 1998. *Evaluating Three-Dimensional Information Visualization Designs: a case Study of Three Designs*. *Proceedings Information Visualization 98*. IEEE. pp 137-144.
- [Wusage2005] Wusage. 2005.
Online at: <http://www.boutell.com/wusage/>; last visit: December 2005.
- [Youssefi2003] Youssefi Amir H., D.Duke, M.Zaki, and E.Glinert. 2003. *Visual Web Mining*. In Proceeding of Visual Data Mining at IEEE Intl Conference on Data Mining (ICDM), Florida.
- [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.
- [Zaki2001] Zaki M. J. 2001. *Spade: An efficient algorithm for mining frequent sequences*. Machine Learning Journal, 42:31–60.

Bibliography

- [Zaki2003] Zaki M. J. 2003. *Efficiently mining trees in a forest*. In ACM SIGKDD.
- [Ware2000] Ware Colin, 2000. *Information Visualization: Design for Perception*. London: Morgan Kaufmann.
- [W3C2005] Logging Control In W3C HTTPD. 2005.
Online at: <http://www.w3.org/Daemon/User/Config/Logging.html>; last visit: November 2005.
- [WCTD2006] Web Characterization Terminology & Definitions Sheet - World Wide Web Consortium. Online at: <http://www.w3.org/1999/05/WCA-terms/>; last visit: March 2006.

Annexes

1. Session detection algorithm

The fields of the textual log format, for CLF logging format, are as follows:

date	Date
time	Time
c-ip	Client IP Address
cs-username	Client User Name
s-sitename	Server Site Name
s-computername	Server Name
s-ip	Server IP
s-port	Server Port
cs-method	Method
cs-uri-stem	URI Requested
cs-uri-query	URI Query
sc-status	Protocol Status
sc-win32-status	Win32 Status
sc-bytes	Bytes Sent
cs-bytes	Bytes Received
time-taken	Time Taken
cs-version	Protocol Version
cs-Host	Server Host
cs(User-Agent)	User Agent
cs(Cookie)	Cookie
cs(Referer)	Referrer

A line in the W3C log file looks like the following:

```
2003-02-18 10:12:38 192.168.187.211 ihc-ect W3SVC4 CPJ-SRV 193.137.85.3
80 GET /ihc/Contactos.htm - 200 0 8225 511 16 HTTP/1.1 www2.ca.ua.pt
Mozilla/4.0+ (compatible;+MSIE+6.0;+Windows+NT+5.0;+.NET+CLR+1.0.3705)
ASPSESSIONIDSAQDDADD=BEEJIDABBOOMKMBGHPIIINDC
http://www2.ca.ua.pt/ihc/Index.htm
```

The conceptual algorithm used to identify usage sessions is:

START

*Open log file

```

*Init sessions array
*Initialize temp data structures
do
{
  *Get a line from log file
  *Convert the line to a request structure
  if (*Is valid request)
  {
    bool Request_IP_Found_In_ASession = false
    for (*Every session in the sessions array)
    {
      if (Session->UserIP = Request->UserIP)
      {
        Request_IP_Found_In_ASession = true;
        if (*Referrer page present)
        {
          if (Session->Referrer = Request->Referrer)
          {
            if (*User name present)
            {
              if (Session->UserName = Request->UserName)
              {
                if (*Cookie present)
                {
                  if (Session->Cookie = Request->Cookie)
                  {
                    if (*time between requests is less than maxi time between requests)
                      *Add the request to current Session
                    else
                      *Finalize current Session
                  }//Cookies equals
                }
                else
                  *Create new Session
              }//Cookies present
            }
            else
              if (*time between requests is less than maxi time between requests)
                *Add the request to current Session
            else
              *Finalize current Session
          }//Users equals
        }
        else
          *Create new Session
      }//Users present
    }
    else
      if (*Cookie present)
      {
        if (Session->Cookie = Request->Cookie)
        {
          if (*time between requests is less than maxi time between requests)
            *Add the request to current Session
          else
            *Finalize current Session
        }//Cookies equals
      }
      else
        *Create new Session
    }//Cookies present
  }
  else
    if (*time between requests is less than maxi time between requests)
      *Add the request to current Session
    else
      *Finalize current Session
}//Referrers equals
else
  *Create new Session
}//Referrer present
else
{
  if (*User name present)
  {
    if (Session->UserName = Request->UserName)
    {
      if (*Cookie present)
      {

```

```

        if (Session->Cookie = Request->Cookie)
        {
            if (*time between requests is less than maxi time between requests)
                *Add the request to current Session
            else
                *Finalize current Session
        } //Cookies equals
        else
            *Create new Session
        } //Cookies present
        else
            if (*time between requests is less than maxi time between requests)
                *Add the request to current Session
            else
                *Finalize current Session
        } //Users equals
        else
            *Create new Session
    } //Users present
    else
        if (*Cookie present)
        {
            if (Session->Cookie = Request->Cookie)
            {
                if (*time between requests is less than maxi time between requests)
                    *Add the request to current Session
                else
                    *Finalize current Session
            } //Cookies equals
            else
                *Create new Session
        } //Cookies present
        else
            if (*time between requests is less than maxi time between requests)
                *Add the request to current Session
            else
                *Finalize current Session
        } // Else No referrers present
    } //IP equals
} //For

if (Requet_IP_Found_In_ASession = false)
    *Create new Session

} //Valid request
} while (*end of file)

return *sessions array

STOP

```


2. Database model and application framework

2.1. Database Structure

The database is structured in two major areas: one for storing the information related to the user interface settings, user rights, language settings, etc.; the second that stores the website related information, for different versions of the website.

The database was implemented as a relational database, using Microsoft SQL Server 2000 as DBMS.

I. *Application maintenance information*

Since the application uses a relational database for storing the information, the general application settings are stored inside the database. For a better integration of the application and easier portability, all raw information used by the application, represented by binary resources, is also stored inside the database. Using the application framework, all this information is accessed and manipulated very easily. The architecture of the framework integrates the database access and manipulation with a set of data access classes. Following sections present the main tables and their specific meaning for our application. Figure 114 presents the user/groups tables, general application settings, user rights, application version, logging and logistics tables.

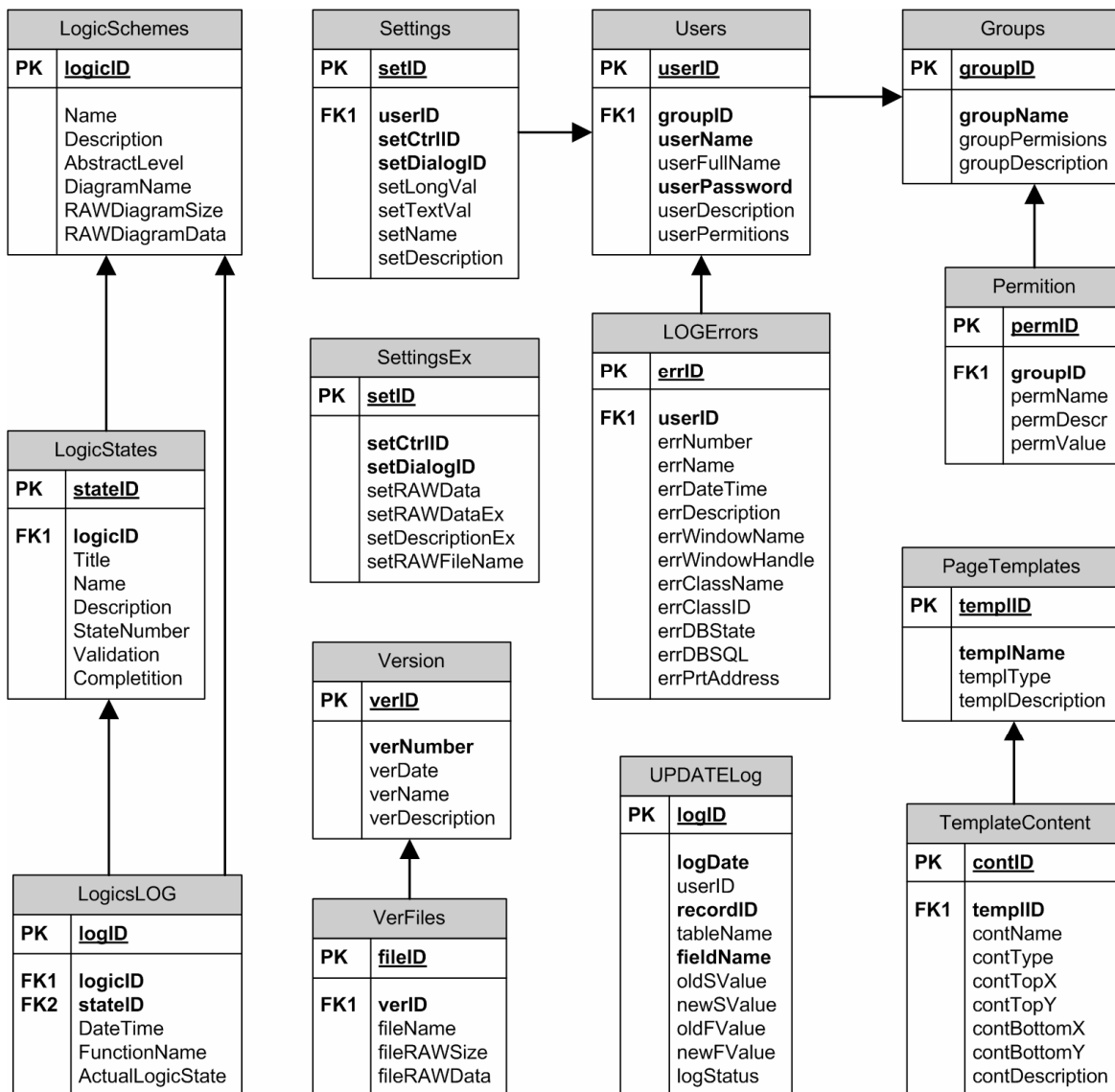


Figure 114 Application maintenance information entities

A textual description of the most important entities is:

- *Users* area stores information about the users of the application, user groups and what rights do they have;
- *Logging* area stores information about the system status at a specific time, usually when system errors occur or important actions are completed;
- *Version* area stores information about the versions of the application/database and a description the changes history;
- *Logic* stores information about the logical states of the application and a *logging* subsystem for storing the current logical state of the application at a specific time;
- *Page Templates* section stores information used by the application to identify and classify the content of a webpage.

II. Website information

The website related information is stored in the database, separately for each version of the site. The structure of the information related to the website is specifically designed to permit high-level interrogations for the presentation layer applications. The website information area represents the primary data that is processed and manipulated by the visualization application.

Most of the structural attributes associated to the website pages are directly stored in the database, some others are dynamically calculated at runtime using queries.

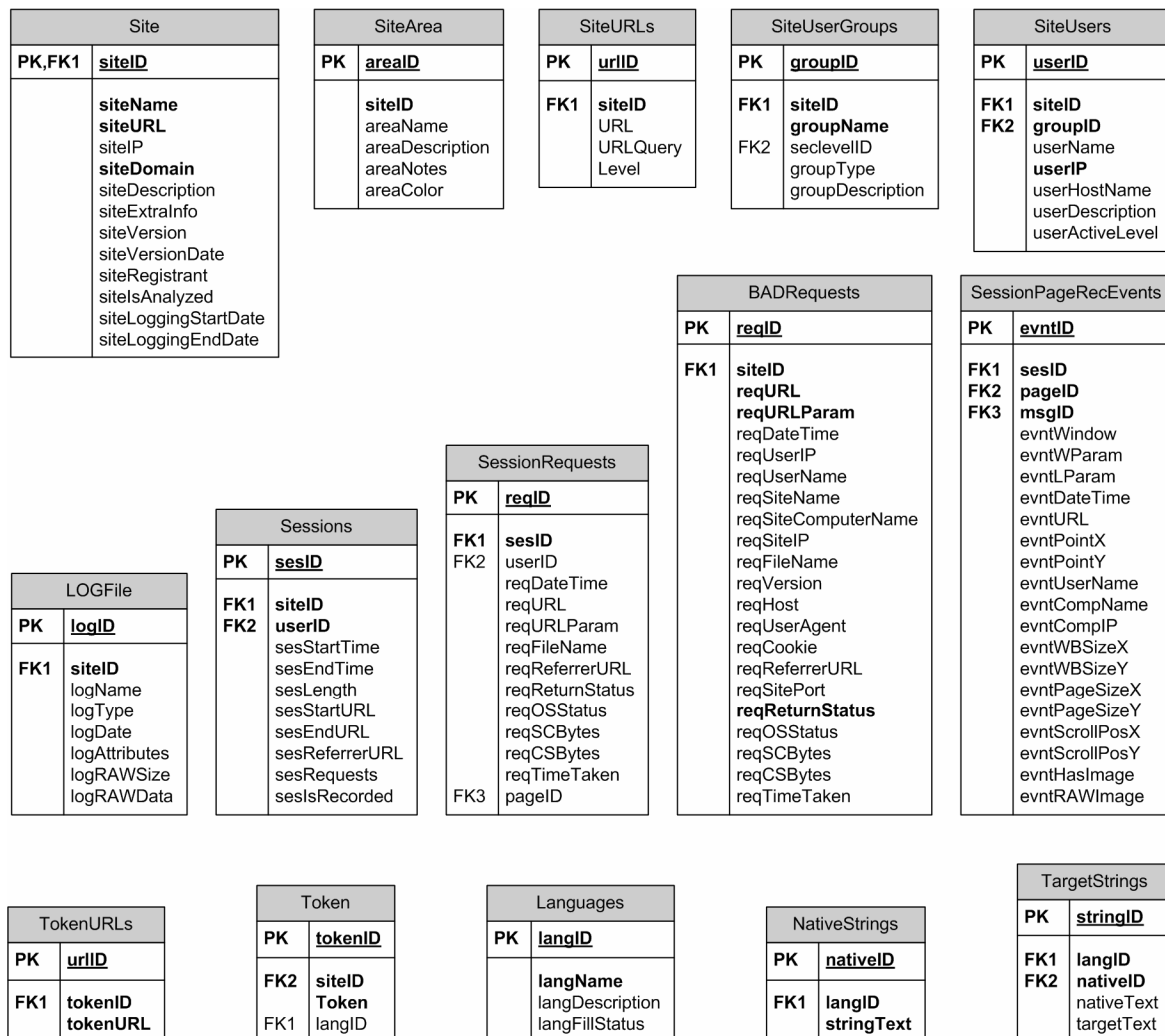


Figure 115 Website information entities

The most important website entities, as presented in Figure 115 and Figure 116, are:

- *Site* area used to store information that uniquely identify a site, like site name, URL, capture timestamp;
- *Logging* area of the site used for storing the log files of the site;
- *Site Users* and *Groups* used to identify the actual users of the site;

- *Security Levels* of the site that identifies the type of access in different areas of the site;
- *Sessions* area stores information about all site user sessions that have been identified; here the information about every session is also stored, as well as information collected from the log files or from controlled experiments;
- *Web Page* area used for uniquely identifying the pages of the site, altogether with the entire information contained by the pages (like page objects, hyperlinks, etc.);
- *Token* area stores the dictionaries of the web sites (kind of glossary).

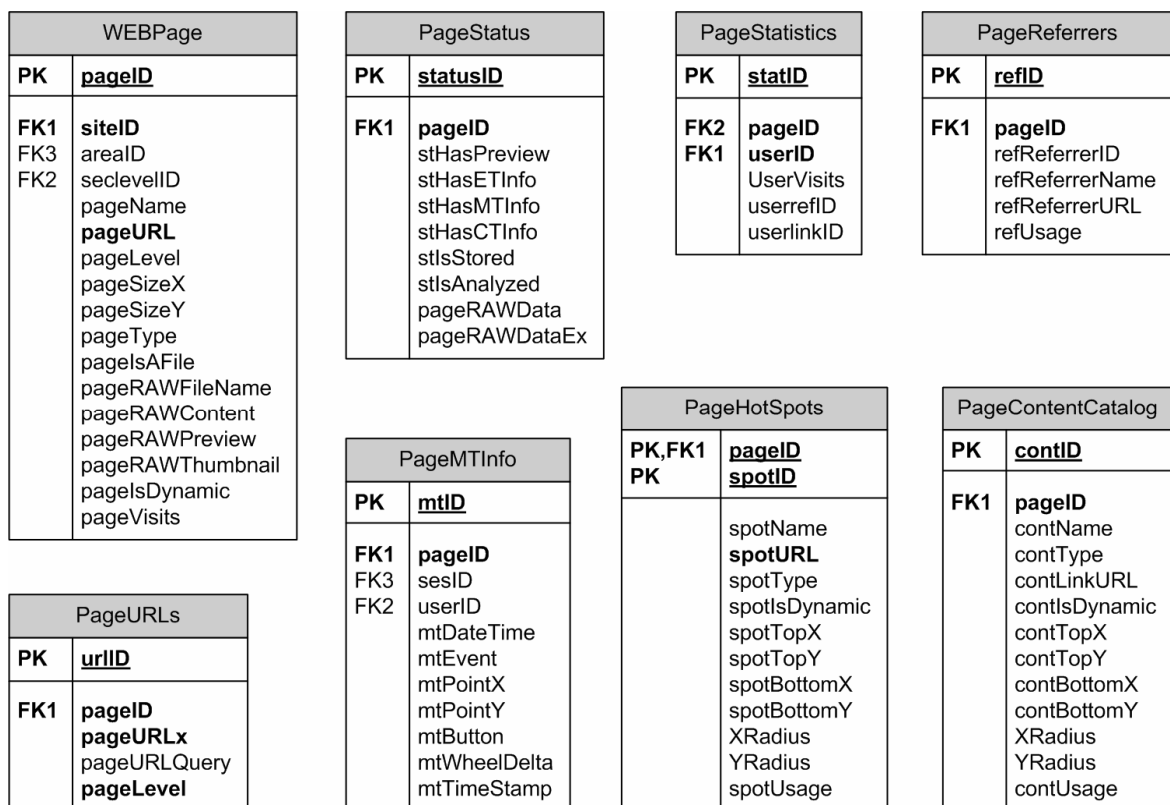


Figure 116 Webpage information entities

The *Web Page* information presented in our model is classified as follows:

- information needed to uniquely identify a page;
- the screenshot of the page for preserving the page content captured at a specific time;
- information regarding the page referrers, meaning all the possible pages from which the site users can load the page;
- information about the page content, representing all the objects contained by the page, defined as the *page content catalog*;

- information about the page hotspots, representing all the areas and de pointing URL's of the page from which the site user can change the context (to load another page or to change the page content);
- information captured from controlled experiments like eye tracking and mouse tracking information regarding the interaction user–page;
- information representing the statistical data related to page usage. It is included here the statistical information about every site user that explored the page. For instance, the entity point can be the information about all the site users that have visited the page, from where they came, and where they went. Using this approach, we are able to store information about both site users and pages, depending on the way we need to use the data.

2.2. Application framework

During the development of the application, a custom application framework has been conceptualized and implemented. As described in section Application architecture and technologies used, we refernced the framework as the *business layer* because this layer is institutional dependent and reflects the business needs of the institution, respecting its design and implementation patterns.

Application framework represents a collection of components that offer an interface between the *presentation layer* and the development platform. It is formed of a set of integrated tools for manipulating the database information, network communication, image manipulation, and a set of controls used for manipulating different types of information. All application components are using the entities of the framework, allowing an easier manipulation of the information and a better integration. In conformance with the needs of our application, we were able to identify the main entities that should be part of our framework as described in following paragraphs.

Figure 117, Figure 118, Figure 119 and Figure 120 present an overall simplified view of the network communication, application, data access and manipulation and used defined controls subsystems. We can detail some important entities as follows:

- database connection objects represented by *CADOConnection* and *CADORecordSet*;
- client-server communication entities represented by *CTCPServer*, *CTCPClient*, etc.;
- user interface entities: *CControlSplitter*, *CWndControl*, *CDlgControl*, *CMenuEx*, *CToolBarEx*, *CToolTipCtrlEx*, etc.;
- general application management entities like: *CLangManager*, *CLogManager*, *CLogicScheme*, *CWndCollection*, etc.;
- image manipulation entities like: *CDImage*, *CDBImage*, *CThImage*, *BitmapEx*, etc.;
- file logging and system logic entities like: *CFileObject*, *ClogEntry*, *CLogocState*;

- data manipulation entities represented by *CBasePtr*, *CSmartPtr*, *CUseRegistry*, *CHashTable*, etc.

All these entities were defined according to the needs of our application:

- data stored in a database requires a connection to the database and components to manipulate query results;
- distributed controlled experiments require client-server connections;
- flexible user interface requires flexible user defined controls, easy to access and to program;
- manipulating large data collections identified by unique strings requires hashing algorithms;
- manipulating data files requires a input/output interface for accessing the data;
- multilingual support requires subsystem for manipulating the language information.

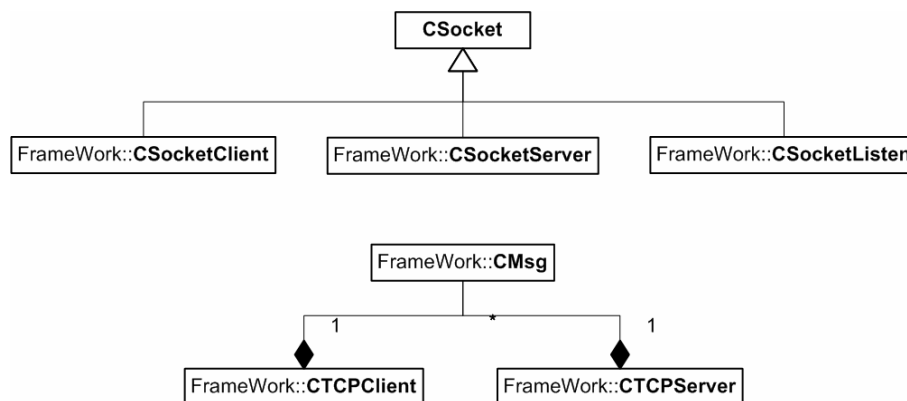


Figure 117 Framework – network communication

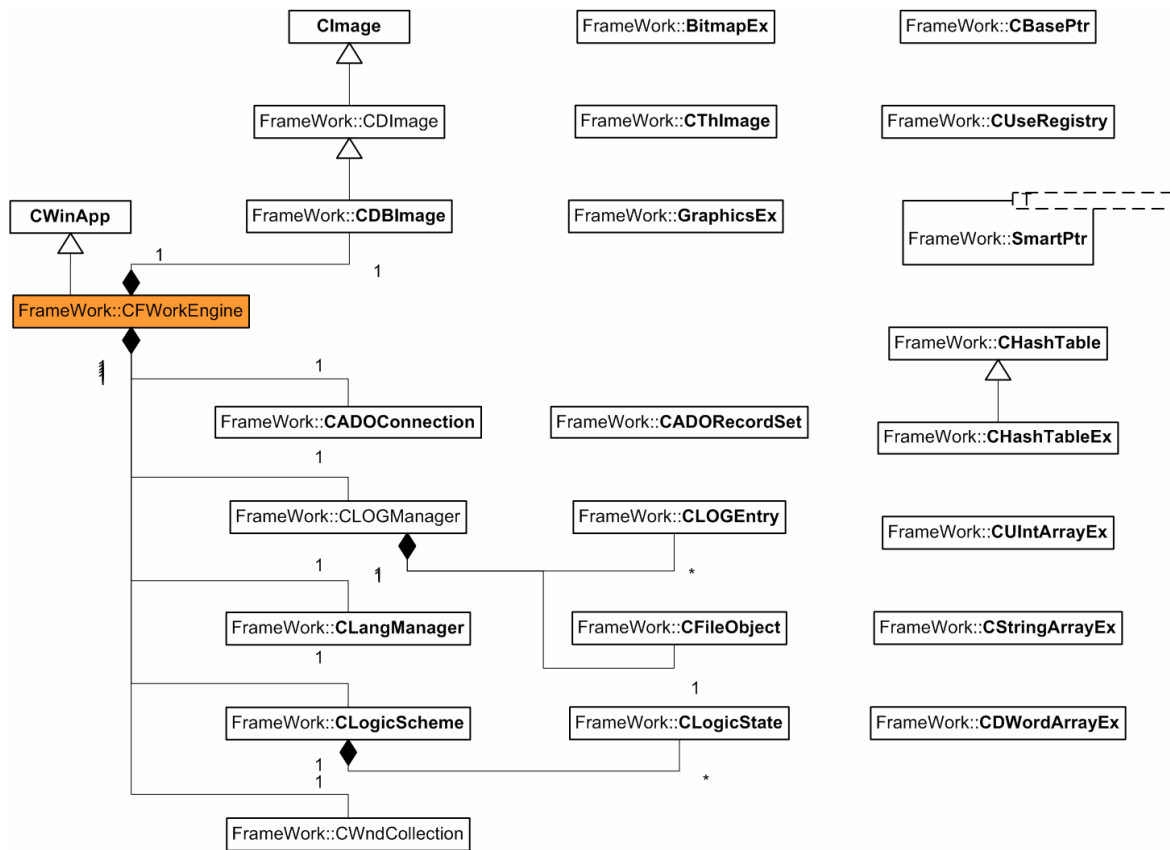


Figure 118 Framework – application subsystem, data access and image manipulation

Figure 119 presents a more detailed view of some application components.

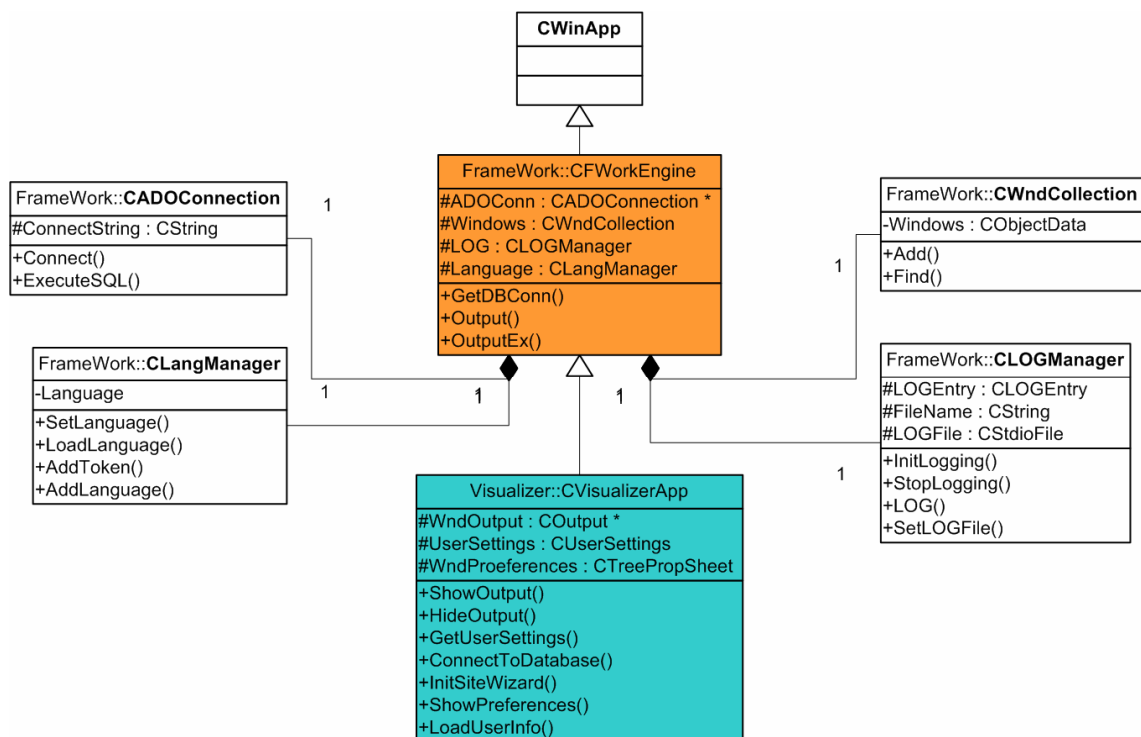


Figure 119 A more detailed view of framework integration

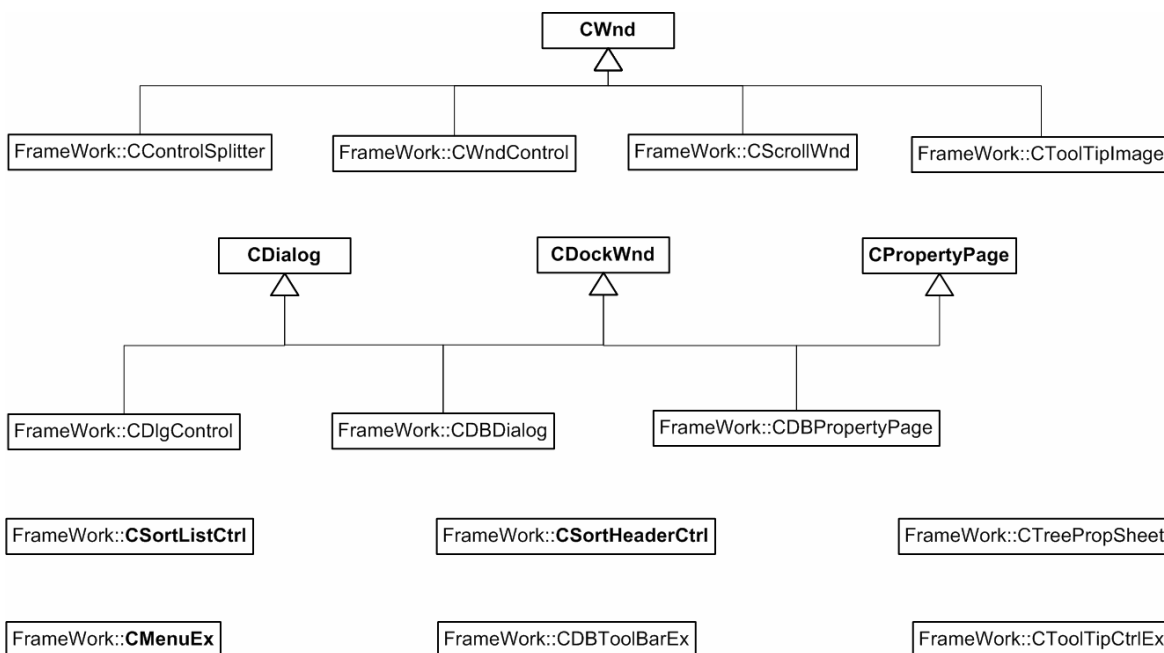


Figure 120 Framework – user defined controls

All these classes have been implemented because we needed to extend the development platform’s basic functionality with specific ones, inadequate or not present on the default implementation of the development platform. We can exemplify the following functionalities we required:

- load and save images from a memory zone or from a database;
- we required a list control with specific sorting capabilities;
- we required menus, tool-tips or toolbars with specific functionalities and aspect;
- we required window controls that implement a default database access and load/save the values of their contained controls;
- we required specific arrays data manipulation, e.g. to map a strings to values, etc.

All these functionalities were implemented and used to create the components of our application, to communicate between them and to access the database or to implement specific user interface aspects / interactions.

3. Evaluation details

3.1. Procedures and Measures

3.1.1. Procedure used with the students

A pilot evaluation was conducted with three students (that didn't participate in any of the sessions mentioned below), in order to assess the difficulty of the tasks, their duration and the clarity of the questions. As a result, some modifications had to be made both in the tasks and questions.

Four evaluation sessions were performed with the students; one (TS1) with eighteen students and another (TS2) with fourteen students. After these two test sessions, we have performed another two sessions with the same students (TS3 and TS4). Thus, each student participated in two sessions of 1h 45 m. The thirty-two students were organized, during each session, in groups of two students named User#1 and User#2; User#1 would act first as user and latter as evaluator; and User#2 would do the opposite. These roles were attributed randomly. Figure 121 shows the sequence of roles performed by the students.

Before asking the participants to perform the tasks, they were given an overall description of the application and of the *visualizations*, as well as, some details of the user interface. Then, they were debriefed about the evaluation: the tasks they were supposed to perform and observe the questionnaires they were supposed to answer, and the used scales. In the first session with each set of students, this explanation and question answering lasted approximately 45m; after this period of time, every one admitted they had understood what they were supposed to do and were willing to participate. Subsequently, participants were asked to start the test and perform or observe the tasks during 30 minutes, then change roles.

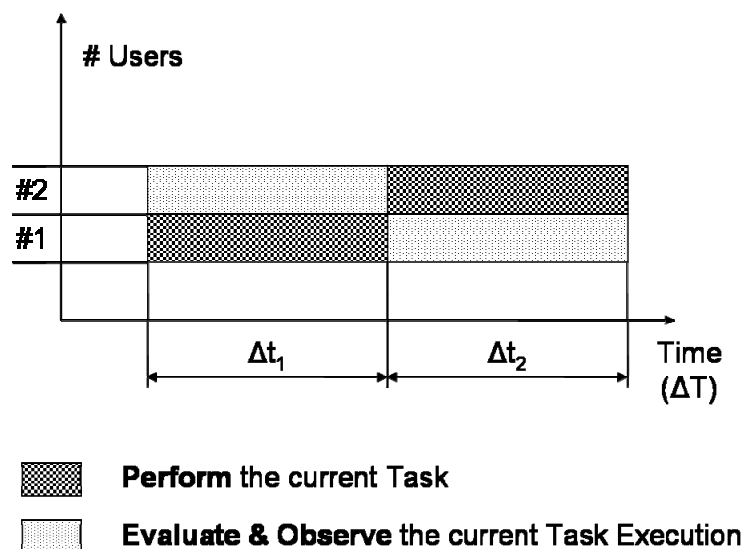


Figure 121 Roles of the students

After sessions TS1 and TS2 we got the impression that some of the students had not fully understood the tasks; thus we decided to perform a second round of sessions (TS3 and TS4) two weeks later, with the same students and following the same organization. This time we didn't explain the application, we concentrated on the explanation of the tasks and let them practice for 10 minutes, before starting the tasks. This was considered a reasonable approach since our intended users are experienced people. Consequently, in this second round of sessions the difference in experience between the two groups of users, Users #1 and #2, was expected to decrease, given that they all had already participated in one evaluation session, two weeks before.

3.1.2. Procedure used with the professionals

A different, generally less structured, approach was used for the evaluation performed with the five professionals. In an initial three hour session, with all of them, a more detailed presentation was given about the application and the *visualizations*. Some details of the user interface were also shown. Then, a period of questions and answers was allowed so that the participants could better understand the overall purpose of the application and the type of evaluation we intended to perform. After this, they have installed the prototype on their own computers and were given a period of time to use the prototype on their own.

During this session we have asked users to think how they would use our application in their everyday work and what kind of other people would also profit from the application; then we invited them to describe this through simple scenarios.

A week later, in 90 minute individual sessions, each user was asked to perform a set of tasks thinking aloud, being observed by two members of the developing team. These

tasks were basically similar to the tasks given to the students plus some others mainly meant to evaluate how they use the *visualizations*.

3.1.3. Collected meters

In the evaluation sessions performed with the students, each observer student had to register the following data concerning the performance of the user:

- time spent performing the task;
- if the task was completed;
- the answer to the question (for some of the tasks);
- user satisfaction;
- any observations the observer considered relevant.

After completing the entire set of tasks, the users were asked to fill a questionnaire giving their opinion on the *visualizations* and some interface features. Concerning each *Visualization*, the following questions were asked:

- is it familiar and intuitive?
- is it easy to use?
- is it easy to learn?
- does it give adequate feedback?
- does it use an adequate colour coding?
- does it provide adequate interaction mechanisms?

Some of these questions were also asked about the icons used to offer the functionality. All these questions (including satisfaction corresponding to each task) were answered using a qualitative scale:

| 1 | 2 | 3 | 4 | 5 | | N |

Where 1 is complete disagreement, 5 is complete agreement and N corresponds to not having an opinion or not wanting to answer. Opportunity to give suggestions or make any comments was given through the inclusion of an open question, at the end of the questionnaire.

Finally, we collected additional opinions and suggestions in informal conversation.

3.2. Exemplified evaluation tasks

Bibliography

Table 11 presents one of the tasks presented to the users during an evaluation session, while Table 12 exemplifies a final questionnaire for the users to fill in at the end of each evaluation sessions.

Table 11 Evaluation task example

Objectives	Group ____		Current Tester: _____				
	Current Evaluator: _____						
	Task NO#	Task Description	Time (mm:ss)	Answer	Task Completed? (Yes / No)	Obs.	Satisfaction 1 2 3 4 5 N
Initialize Observe Pages Interconnections and HotSpots Session Information and HotSpots	1	On SiteMAP2D, Find and Select the Page: Evolution of Organic Vision Find the number of Hotspots on the page.	-- : --	Hotspots = _____			□ □ □ □ □ □ □
	2	On SiteMAP2D, Find the Link Usage for the link between the pages: Vision - Introduction (Index) and Vision (Main Page) (Use the Circle on the Link)	-- : --	Usage = _____			□ □ □ □ □ □ □
	3	Show Possible Paths between the Pages: Pictorial Perception and Culture and Simultaneous Contrast (Use the Possible Paths Tool/Menu)	-- : --	N/A			□ □ □ □ □ □ □
	4	Find the Shortest Path in the Scheme Window SelectionExplorer2D (Only the number of Pages)	-- : --	Shortest Path = _____ (Excluding Start and Stop)			□ □ □ □ □ □ □
	5	On SiteMAP2D, Find and Select the Page Bibliografia . Which Book was most readed?	-- : --	= _____ Book			□ □ □ □ □ □ □
	6	Maximize PageExplorer2D Window Observe the Link Usage for the most visited HotSpot (Link Usage is color-coded from Dark to White , Min to Max).	-- : --	Usage = _____			□ □ □ □ □ □ □
	7	Restore PageExplorer2D Window. In SessionMAP2D Window, Select the Session: IP = 192.168.8.118 Date = 2003-02-25 12:44:25 and Requests = 31	-- : --	N/A			□ □ □ □ □ □ □
	8	Switch to Representation Function and Count all the External Jumps (Squares) between Pages	-- : --	No. Jumps = _____			□ □ □ □ □ □ □

Objectives	Group ____						
	Current Evaluator: _____		Current Tester: _____				
	Task NO#	Task Description	Time (mm:ss)	Answer	Task Completed? (Yes / No)	Obs.	Satisfaction 1 2 3 4 5 N
	9	Select All Sessions and switch to Representation Histogram . Using <i>PageExplorer2D</i> window, Find the Hotspot that corresponds to the Whitest Hotspot Area in <i>SessionMAP2D</i> .	--- : ---	Hotspot = _____			
	10	Now Use the Overlay Tool and <i>Select</i> the page Pagina Principal from <i>SiteMAP2D</i> . Find the Hotspot that corresponds to the Yellow Hotspot Area .	--- : ---	Hotspot = _____			
11	Now <i>Select All the Sessions</i> for the User with the IP = 192.168.187.209	--- : ---	N/A				
12	The visits are color-coded from Dark to White representing a scale from min to max . Which was the most visited area (Hotspot) for this user sessions?	--- : ---	Hotspot = _____				

Observations:

Table 12 Final questionnaire example

Group __		Current Tester: User _____							
NO#	Esquema de Visualização	É Familiar / Intuitivo 1 2 3 4 5 N	É Fácil de Usar 1 2 3 4 5 N	É Fácil de apreender e Memorizar 1 2 3 4 5 N	É Fácil de Compreender a Informação 1 2 3 4 5 N	Dá Feedback Adequado 1 2 3 4 5 N	Furnece Interação Adequada 1 2 3 4 5 N	Utiliza um Esquema de Cores Adequado 1 2 3 4 5 N	Opinião Geral 1 2 3 4 5 N
1	SiteMAP2D	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _
2	PageExplorer2D	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _
3	SelectionExplorer 2D	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _
4	SessionMAP2D	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _
5	Histogram	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _	_ _ _ _ _ _ _
6	ToolBars Icons	_ _ _ _ _ _ _	N/A	_ _ _ _ _ _ _	_ _ _ _ _ _ _	N/A	N/A	N/A	_ _ _ _ _ _ _

Observações:

Table 13 Example task with check items

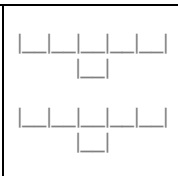
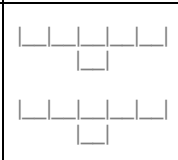
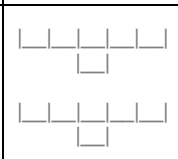
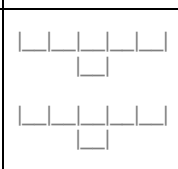
Objectives	Group ____		Current Tester: User #1					
	Current Evaluator: User #2		Current Tester: User #1					
	Task NO#	Task Description	Variable(s)	Time (mm:ss)	First Choice(s)	Approach	Observations	Satisfaction Importance 1 2 3 4 5 N
Observe Pages Interconnections and HotSpots	1	Select the Site: (Finish the Wizard) (Show the Relations Window)	<i>Interação Humano-Computador Vers 1.2.00</i>	-- : --	_____	<input type="checkbox"/> Available Sites List→Select <input type="checkbox"/> Combo Boxes→Select <input type="checkbox"/> Click on Relations inLeft Tree Other: _____ _____ _____		
	2	Select the Level:	<i>Level 2</i>	-- : --	_____	<input type="checkbox"/> Click on Scheme Header→Select <input type="checkbox"/> Ctrl + Lclick on Level Pages→Select <input type="checkbox"/> Multiple Selection Tool→Select <input type="checkbox"/> RClick + Menu: Object→Select <input type="checkbox"/> Other: _____ _____		
	3	Find and Select the Page:	<i>Programa</i>	-- : --	_____	<input type="checkbox"/> Translate→Zoom→Select <input type="checkbox"/> Zoom→Translate→Select <input type="checkbox"/> Move Cursor→HotTooltip→Select <input type="checkbox"/> Ctrl + F→Type→Mark→Select <input type="checkbox"/> Other: _____ _____		
	4	Find and Observe the HotSpot Info for HotSpot: (Maximize PageExplorer2D)	<i>Aulas Práticas</i>	-- : --	_____	<input type="checkbox"/> Translate→Zoom→Find <input type="checkbox"/> Zoom→Translate→Find <input type="checkbox"/> Move Cursor→HotTooltip→Find <input type="checkbox"/> Ctrl + F→Type→Mark→Find <input type="checkbox"/> Other: _____ _____		
	5	Show Possible Paths Between the Pages:	<i>Vision-Introduction (Index)</i> <i>Lateral Inhibition and Adaptation</i>	-- : --	_____	<input type="checkbox"/> Toolbar→LClick on Vision... →Ctrl + LClick on Lateral... <input type="checkbox"/> Toolbar→LClick on Lateral... →Ctrl + LClick on Vision... <input type="checkbox"/> Other: _____ _____		

	6	Find the Shortest Path (Only the IDs of the Pages) In the Scheme Window:	<i>SelectionExplorer2D</i>	-- : --	<input type="checkbox"/> IDs: 560→561→566→575 <input type="checkbox"/> IDs: 561→560→566→575 <input type="checkbox"/> IDs: 575→566→561→560 _____	<input type="checkbox"/> Maximize Window and Observe <input type="checkbox"/> Zoom→Translate→Observe <input type="checkbox"/> Move Cursor→Tooltip→Observe IDs <input type="checkbox"/> Other: _____ _____		
--	---	--	----------------------------	---------	---	---	--	--

Observations:

Objectives	Group __		Current Tester: User #1							
	Current Evaluator: User #2		Task NO#	Task Description	Variable(s)	Time (mm:ss)	First Choice(s)	Approach	Observations	Satisfaction Importance 1 2 3 4 5 N
	Observe Session Information and HotSpots Usage	1	Find and Select the Session:	<i>IP: 192.168.8.46</i> <i>2003-02-27</i> <i>09:32:44</i> <i>Requests = 46</i>	-- : --	_____	_____	<input type="checkbox"/> IP Sort→List Click→Select Btn <input type="checkbox"/> Requests Sortt→List Click→Select Btn <input type="checkbox"/> Date Sort→Requests Sort→Select Bt <input type="checkbox"/> Other: _____ _____		
2		Find the Most Visited Page: (Maximize the SessionMAP2D)	<i>Timeline</i> <i>Representation</i>	-- : --	_____	_____	<input type="checkbox"/> The Human Eye <input type="checkbox"/> Human Retina <input type="checkbox"/> Two Visual Systems of the ... <input type="checkbox"/> Lateral Inhibition and Adaptation <input type="checkbox"/> Other: _____ _____			

Bibliography

3	Select the Sessions:	<i>All Session</i>	-- : --	<hr/> <hr/> <hr/> <hr/>	<input type="checkbox"/> Click First→ Shift + Click Last→Select <input type="checkbox"/> Ctrl + A →Select Btn <input type="checkbox"/> Ctrl + Click on Each Session→Select <input type="checkbox"/> Other: _____		
4	Switch to Representation:	<i>Histogram</i>	-- : --	<hr/> <hr/> <hr/> <hr/>	<input type="checkbox"/> Toolbar→Histogram→Switch <input type="checkbox"/> RClick→Show→Histogram→Switch <input type="checkbox"/> Other: _____		
5	On SiteMAP2D Find and Select the Page:	<i>Página Principal</i>	-- : --	<hr/> <hr/> <hr/> <hr/>	<input type="checkbox"/> Translate→Zoom→Select <input type="checkbox"/> Zoom→Translate→Select <input type="checkbox"/> Move Cursor→HotTooltip→Select <input type="checkbox"/> Ctrl + F→Type→Mark→Select <input type="checkbox"/> Other: _____		
6	Find the Most "RED" Link on Histogram in the Scheme Window:	<i>PageExplorer2D</i>	-- : --	<hr/> <hr/> <hr/> <hr/>	<input type="checkbox"/> Contactos Link <input type="checkbox"/> Programa Link <input type="checkbox"/> Aulas Teoricas Link <input type="checkbox"/> Aulas Praticas Link <input type="checkbox"/> Other: _____		

Observations:

CONTENTS

Annexes	4
1. Website related information	4
1.1. Site contents and interconnections.....	4
1.2. Logs extracted information.....	8
1.3. Conceptual entities.....	12
1.4. Information characteristics.....	14
2. Entities and data structures	15
2.1. Website representation.....	15
2.2. Session representation.....	20
2.3. Multiple sessions.....	23
2.4. Visualization data structures.....	24
3. Taxonomical classification of visualization methods	26
4. Site Analyzer and Compiler Conceptual Models	34
5. Detailed Data Structures	36
6. SharePoint Analyzer / Designer Implementation Details	41
6.1. Application components.....	41
6.2. Used technologies.....	42
6.3. Programming Model – Key Concepts.....	42
6.4. Components.....	43
6.5. Programming Model – Xml definitions.....	47
7. Functional Description for Visualization Methods	49
Bibliography	66

LIST OF TABLES

Table 1 Visualization methods classification for structural website / session analysis	27
Table 2 Visualization methods classification for visual inspection of visual workspace coherence	31
Table 3 Xml Schema definition for Portal creation	47

LIST OF FIGURES

Figure 1 Visualization taxonomy, for each dimension – a type of classification	27
Figure 11 <i>Site Analyzer</i> conceptual model	34
Figure 12 <i>Compiler</i> conceptual model	35
Figure 13 Flowchart Control basic components	44
Figure 14 Flowchart Control data / connecting components	45
Figure 15 Flowchart Control design components	45
Figure 16 Portal manipulation and UI components.....	46
Figure 17 SharePoint manipulation component.....	46
Figure 2 3D Site Map visualization method	50
Figure 3 2D Site Map visualization method	52
Figure 4 3D Page Explorer (Links + Referrers) visualization method.....	54
Figure 5 SessionMAP 2D Timeline/Function visualization method	55
Figure 6 3D Session Map visualization method.....	57
Figure 7 Site/Page Explorer visualization method	59
Figure 8 Page Explorer visualization method	60
Figure 9 Possible Interconnection visualization method.....	62
Figure 10 Hotspot Usage Intensity visualization method.....	63

Annexes

1. Website related information

From the structural point of view, websites are classified in pages and interconnections. Using a higher level of abstraction, semantically related groups of pages belong to clusters or areas. At the page level, page elements are classified according to their interactivity (hotspots and non-hotspots) and representation type (text, graphics, etc.).

Overall website browsing experience can be analyzed based on the information extracted from the page contents and their interconnections, UI design and navigation effectiveness being considered key factors for a successful browsing experience.

1.1. Site contents and interconnections

One key goal in website usability analysis is the ability to build fully or semi automated tools able to track improper design layouts or navigational malfunctions, based on the information extracted from the contents and interconnections of websites, combined with additional usage information. Several tools have been proposed by various authors ([Faraday2000], [Card2001], [Ivory2002], etc.) to facilitate the analysis of websites. Content managers, website designers and usability specialists can use such instruments to analyze or improve the design of websites, based on the potential insight provided by the outputs of these tools.

Given the complexity of nowadays websites, fully or semi automated tools are required to process the website structure and contents, classify its pages and identify interconnections among them. For this purpose, information structures have to be identified, processed and stored in an auxiliary format that supports further processing, grouping and interrogations. One way to resolve the problem is to use tools capable to identify the structure of the website, to request and store the contents from the web servers, and to run analysis algorithms over the resulted data.

The process of identification and download of website pages is known as *web crawling* and one possible algorithm is described in [Najork2001]. The process of analysis of primary data is known as *data mining*, with a web-oriented version called *web mining*. Some possible approaches for web mining are proposed in [Spiliopoulou2000], [Niu2003], [Youssefi2003], [Zaki2001] and [Zaki2003]. Since our research field involves web mining strategies only on the data preparation phase, we focused our attention on the feedback that can be achieved using the resulted information structures and not on the web mining process itself.

Our goal is to visualize the website contents as it is provided to its final users, in a web browser window.

For representation purposes, we considered the structure of a website as a union of two sets: pages and interconnections, usually called hyperlinks:

$$(1) W = (P, H).$$

P is the set of pages

$$(2) P = \{P_1, \dots, P_n\}$$

where n = number of site pages; H is the set of all possible interconnections between all website pages

$$(3) H = \{H_1, \dots, H_m\},$$

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).

Inspired by Faraday's [Faraday2000] and Ivory's [Ivory2002] classifications for page elements, we considered the following elements as important for the classification of page contents: *Text elements*, *Link elements*, *Graphic elements*. Note that the Link elements represent the base elements for hyperlinks, being included in web pages as interactive elements that provide the means for website navigation or interaction. However, hyperlinks refer to edges in a graph representation of a website, while link elements refer to the objects contained on the page itself, usually called hotspots.

Considering T as the set of *Text elements* in a page, L as the set of *Link elements* and G as the set of *Graphical elements*, a page P_i can be specified as the union of the three sets:

$$(4) P_i = T_{i,p} \cup L_{i,q} \cup G_{i,r},$$

where p = number of Text elements on page i , q = number of Link elements on page i , r = number of Graphical elements on page i .

Using Healey's [Healey2001] and Mealha's [Mealha2004] approach and definitions as reference and definition (1), we associated attributes to each hyperlink element present on a webpage: each page $P_i = \{h_{i,1}, \dots, h_{i,n}\}$, contains n hot-spots; each hot-spot, $h_{i,n}$, of any specific page has at least $m > 1$ attributes, $A = \{A_1, \dots, A_m\}$. In conformance with this approach, 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 element:

$$(5) A_T = \{A_{T,l}, \dots, A_{T,w}\},$$

$$(6) A_L = \{A_{L,l}, \dots, A_{L,v}\},$$

$$(7) A_G = \{A_{G,l}, \dots, A_{G,w}\}.$$

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* and *Graphical* elements.

Two types of attributes were considered:

1. *structural attributes* – used define the referenced element itself;
2. *extended attributes or statistical attributes* – used to represent additional information.

Seven structural attributes that describe the contents of the web page itself were identified:

- A_1 = URI (*Unique Resource Identifier*) of the page. In some cases, the URI can be composed of the main URI of the page/site and additional query parameters. 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.).

Note that a *website level* is defined as the shortest path (number of pages visited) from the entrance of the site to the specific web page, considering a depth-first traversal of the website associated graph. Each level is defined as a set of pages, all having the same distance from the entrance of the site and considering the same depth-first traversal.

One useful measure of optimization can be defined as the *shortest path* from a specific page to a given goal (represented as two nodes of the website associated graph). and Generic algorithms like *Dijkstra* or *Belmann-Ford* can be used to determine the shortest path. User satisfaction can be achieved with the shortest path, however, several other factors might influence users navigational decisions (like page loading time, link position, etc.).

Some web pages might receive input queries as parameters the result being a complex URI composed of a unique base URI and a set of additional parameters:

$$(8) \quad A_{1,i} = A_{1,i,k} \cup \sum_{j=1}^n V_{i,j}$$

K represents one unique URI, $V_{i,j}$ represent one pair {key, value}, usually in format key=value, e.g.: <http://www.domain.com/Default.aspx?Page=Account.aspx&User=userID>.

The same formalism can be applied to each *Text*, *Link* and *Graphical elements* of the web page.

For *Text elements*, we considered a set of five structural attributes (A_T set):

- $A_{T,1}$ = 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*, we considered a set of five structural attributes (A_L set):

- $A_{L,1}$ = 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*, we considered a set of five structural attributes (A_G set):

- $A_{G,1}$ = 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.);

Note that many of these attributes are complex attributes resulted from a limited subset of elementary attributes. An example is the $A_{T,2}$ attribute, whose specific subset might consider font size, style, spacing, color, paragraph, reading order, etc., all representable as a set of pairs {key, value}.

A possible representation of complex attributes is the following:

$$(9) \quad A_w = \bigcup_{i=1}^n a_{w,i} \mid a_{w,i} \rightarrow \Delta_i, \text{ where } A_w \text{ is the complex attribute } w, a_{w,i} \text{ are elementary attributes with values in } \Delta_i, \text{ domain of elementary data types as numbers, dates, texts, etc.}$$

Link elements extend the *Text* or *Graphical* elements and might inherit some common attributes, e.g. a link element can be represented as a text or an image.

The structural attribute were accordingly extended to represent usage information extracted from the log files, captured during controlled experiments ([Drott1998], [Zamfir2004], [Fraternali2003]), or collected using visual or motion tracking and interception systems.

For the web page, the following extended attributes were considered:

- A_8 = number of times the 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 minimum time between two subsequent request;
- A_{11} = maximum amount of time the page was analyzed by the clients, using the maximum time between 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 considered for the A_T set:

- $A_{T,6}$ = the average time users spent to analyze the text element;
- $A_{T,7}$ = classification of the text element on a scale, regarding focusing and attention capture.

For *Link elements*, four extended attributes were considered for the A_L set:

- $A_{L,6}$ = the average time users spent to analyze the link element;
- $A_{L,7}$ = average time to go from one page to another using the hotspot;
- $A_{L,8}$ = number of times this link element was clicked (visited) during a specific time period;
- $A_{L,9}$ = classification of the link element on a scale, regarding focusing and attention capture.

For *Graphical elements*, two extended attributes were considered for the A_G set:

- $A_{G,6}$ = the average time users spent to analyze the graphical element;
- $A_{G,7}$ = classification of the graphical element on a scale, regarding focusing and attention capture.

Note that some of these extended attributes are complex attributes, represented accordingly.

1.2. Logs extracted information

During natural website usage, most of the application (web) servers store some information for the client requests, using one or more of several available formats: formatted text files, database files, xml files, etc. The process is called *logging* and the concept behind is that every time a website user accesses a file stored on the web server (that might reference several additional files), one or more entries are added to the site's associated log file. The file format, type and contents used to store logging information depend of the type of application server and its configuration. However, an application server must provide a reasonable set of fields required to uniquely identify the request:

who makes the request, where from, to what server, at what time, what was requested and what the results.

Considering the representation (1) of the website associated graph and a page $p \in P$ of the site, a request $r_i \in R$ (R = all requests to a server) to the page p is defined as a union of attributes that uniquely identify the request:

- T_i = timestamp (date and time) of the request;
- C_i = the client behind the request;
- D_i = the destination of the request;
- Q_i = additional querying parameters;
- B_i = client browser identification;
- S_i = client session identification;
- X_i = response parameters; and
- Z_i = the context from which the request came.

$T_i \in T$, $C_i \in C$, $D_i \in D$, $Q_i \in Q$, $B_i \in B$, $S_i \in S$, $X_i \in X$ and $Z_i \in Z$ are sets of attributes that depend of the implementation of the logging system. However, five of these sets have a particular interest for our work:

- $T_i \in T$, the timestamp = attribute(s) that uniquely identifies each subsequent request that came to the web server from the same client (usually is used the server's date and time at the moment of the request);
- $C_i \in C$, the client's identification = attribute(s) that uniquely identifies the client machine that makes the requests to the web server (usually the client's IP or DNS name is considered);
- $Q_i \in Q$, request parameters or query = attribute(s) that have a direct mapping to a subset of $x = (x_1, x_2, \dots, x_n)$ = the set of all variables (fields) the users fill on a dynamic page, $y = (y_1, y_2, \dots, y_n)$ = the set of variables that can affect personalized pages and $z = (z_1, z_2, \dots, z_n)$ = the set of variables that controls the areas the user can access. For simplicity we assume that $Q = x$;
- $S_i \in S$, session identification = attribute(s) that uniquely identify a user session, usually a *Cookie* or a *Session ID* (limited vector of characters that uniquely identifies each new connection to a web server);
- $Z_i \in Z$, context identification = attribute(s) that uniquely identify the context of the client, the referrer context (usually, the URI of the page that makes the request to the server; it is considered as referrer page).

A simplistic example of a log file format is the following:

{<Requesting host IP address or DNS name>,
 <The remote logname of the user.>,
 <User authenticated name>,
 <Date and time of the request>,
 <The request line exactly as it came from the client>,
 <HTTP status code returned>,
 <Bytes transferred>}.

Commonly, web servers use standardized formats for the logging system to store information, one of the most common standards being introduced by W3C [W3C2005], called *common log file format* (CLF). It consists of a reasonable set of fields required to uniquely identify the request. An extended version of this format do exists, identified as extended common log format (ECLF). W3C logging standard format limits the delimiter of fields to space character, every occurrence of a space or nonprintable characters being normalized.

A *user session* is defined 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:

$$(10) \quad \Delta T_k = \sum_{i=1}^n \Delta T_{j,i} = \text{the sum of all } n \text{ sessions of a specific client } j \text{ that accessed the web server (the active time of the } j \text{ client), and } \Delta t_{j,i,m} \in \Delta T_{j,i} = \text{the intervals of time between subsequent requests from a specific session } i;$$

A request $r_{j,i}$ from client j is considered in the same session i if:

$$(11) \quad \Delta t_{j,i,m+1} - \Delta t_{j,i,m} \leq \delta,$$

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.”

Usually, the logging systems register the information in a *request-oriented* format [Ruffo2004], without identifying distinct user sessions as groups of requests coming from the same user; the same file can be used for several domains and sites, turning difficult the process of identification of distinct website users. Therefore, preprocessing has to be

applied to logging information before it can be used to identify user's activity for a given website.

One important step in web-server log files processing is the identification of sessions that can be identified based on cookies, session IDs or referrer URIs. Cookies are usually stored within the request information as a distinct entity while session IDs can be represented as distinct entities or can be embedded within the request URI as cookies. In [Cooley2003] and [Ruffo2004], the identification of a session is also achieved based on cookies or session IDs.

A detailed methodology of analyzing web logs was proposed by Drott [Drott1998], based on the types of information that can be extracted from the log files.

We adopted a custom implementation of the session detection, based on the analysis of cookies, referrers URIs and client identification (user name and IP address).

A wide range of techniques have been implemented and adopted by the community to identify sessions, however, a standardization process is missing. The lack of a standard comes mainly from the variety of logging systems and the ability of website administrators to control the fields to be logged. Several web servers do not log user requests in a full standard format because of the amount of space required to store all user requests.

When several attributes that uniquely identify a session are missing from the log file, the identification of sessions turns difficult. The accuracy of the algorithm depends on the additional logged information. There are several situations when the identification of session is as inaccurate as several external factors are involved on the client's browsing experience. Let's consider the scenario when several computers are connected to the internet using the same proxy server; in this case, if no additional information is present, all requests coming from the clients behind the proxy server have the same client identification IP address. Therefore, if no additional cookie or session information is logged, identification of sessions is inaccurate. Another scenario involves one user with the same site opened in several browser windows; if the identification of the client is based only on IP address, the requests from all opened windows are considered as coming from the same session; this leads to a situation when the sessions associated to the user is inaccurately identified. In these cases, the results of visualization of a graphical representation of the session might turn the analysis confuse since the requests tend to have no common context.

The previously presented inaccurate situations can be bypassed if the logging of the client requests is performed client-side. Considering the scenario of a website controlled usability experiment, correct identification of usage scenarios can be performed if the web browser of the client is provided with an auditing tool, able to intercept and log clients' requests to the web server. The additional information obtained in this case might be critical for studying website users' behavior: contextual screenshots between subsequent

requests, client screen resolution, browser windows dimensions and positions on the screen respectively on z order, input devices used, etc.

Substantial steps have been already made in this area:

- Xml logging framework introduced in [Fraternali2003], able to log client's requests directly on the server, during natural browsing experience;
- Paganelli [Paganelli2002] introduced a logging mechanism able to collect data during usage sessions. The tool identifies session-oriented tasks as part of a overall usability tracking mechanism;
- Card [Card2001] introduced a logging tool able to log most of the client's inputs: keyboard and mouse events, client context, client vision (eye tracking system), etc.

The potential exposed by this approach (client-side logging mechanisms) is the ability to correctly identify task oriented sessions, not request oriented, as detected from the server-side log files. The additional information collected client-side is very useful for several usability analysis tasks, in [Card2001] being described several aspects of browsing behavior detected during an experiment that demanded users to search for specific information on the web.

Unlike server-side logging, client-side logging has to be used carefully; user privacy and frustration provoked by experimental conditions represent a potentially subtle problem an evaluator has to deal with.

1.3. Conceptual entities

The website information to be analyzed is collected from different sources, using several capture, filtering and storage mechanisms. The subsections 1.1. Site contents and interconnections and 1.2. Logs extracted information present the theoretical data sources and collection mechanisms for the information related to the website structure, contents and usage. These information sources can be classified accordingly as:

- a. Structural website information (website structure, contents and interconnections);
- b. Usage information (web server logs or client logs); and
- c. Statistical information obtained after processing the website structure and usage information.

The conceptual model classifies the first two types of information source as *raw data* while the third is obtain after processing and is contained only in the *Referential Digital Atlas*.

Independently of the source, three types of conceptual data structures were identified:

- I. *Elementary (primary) data structures* used to represent primary, unprocessed data like page raw content elements, website pages and interconnections identification

elements, log entries in raw format, etc. This type of data structures are represent with *elementary data types*;

- II. *Complex (preprocessed) data structures* that apply to information resulted after a preliminary processing of raw information, like: statistical information, page contents identification and usage associations, usage patterns identification, conceptual website classification, goals identification and areas classification, reference models used for statistical comparison, etc. These structures are rep[resented using related *entities*, each containing several *elementary attributes*;
- III. *Visualization data structures* are additional used to produce visual representations. They can be represented either as elementary attributes or as related *entities*.

The following entities are of interest for our system:

- I. Website – structure and evolutions in time:
 - Web pages;
 - Interconnections (hyperlinks) for all pages of the website;
 - Semantically related clusters or areas of interests classified based on content type or usage goals;
- II. Web page contents and usage history:
 - RAW and/or HTML page contents;
 - Text, Link and Graphical elements in the page;
 - Page evolution during time: incremental or complete;
- III. User and usage activity monitoring information:
 - Eye tracking – points of interaction, eye movement, timings;
 - Mouse tracking – points of interaction, events, timings;
 - Keyboard events – pressed keys, timings;
 - Screen captures – images, timings, and/or video;
 - Context – visited pages, followed paths, timings;
- IV. Session information:
 - Followed paths – the list of visited pages, using definitions (10) and (11);
 - Timings for each visited page, using definition (10);
 - Events that occurred during the session;
- V. Multiple sessions for single or multiple users:
 - List of alternative paths;

- Statistical information for followed paths (visits for each visited page, per hotspot usage, etc.);
- Minimum, average and maximum timings between subsequent page visits;

VI. Users and user groups:

- Users;
- Groups;
- Groups of users per visited area;
- Groups of users per type of explored contents;
- Cultural membership classification (language, writing style, etc.);
- Profile based classification (when role based authentication is present);
- Groups by usage behaviors – identified from the analysis of usage sessions;

Most of these entities make use of the attributes introduced to represent the website and web pages: $A \cup A_T \cup A_L \cup A_G$, discussed in section Website related information .

A detailed presentation of the conceptual entities and the data structures suitable for the purpose of the proposed system is found in annex 2 – Entities and data structures.

1.4. Information characteristics

Nowadays websites represent constantly growing and dynamic informational spaces that change their structure and content accordingly to sophisticated business needs. This aspect involves challenging issues to be considered for automated or semi-automated website analysis tools, one example being a way to capture an exact snapshot of a website, and then store its contents for further analysis and processing.

Because most of the web pages present in Internet contain HTML code, snapshots might consider each page exactly as it is presented to the client in a browser that interprets HTML. One exception occurs with dynamic or personalized pages, where the content presented to the client depends on the user credentials or several other factors like variables manually changed by the user or dynamic queries. Another exception involves non-HTML code inside web pages, some proprietary formats that cannot be analyzed or reproduced, an example being: Java Applets, ActiveX controls, Flash objects, etc.

To overcome limitations introduced by website evolution, we have to consider for analysis only the snapshots taken within the time interval that defines a version of the website: this means that any change of the contents of a predefined set of pages is considered a different site version.

For analysis purposes, the information has to be available offline, a relational database system was considered as the most suitable for our proposed system. A primary snapshot of the website (base snapshot) can be captured and stored at once, and then incremental versions can be added to the base snapshot.

In some cases, it is impossible to capture and store the personalized information for each user, or dynamically generated information that depends of user queries. To resolve this outcome, screenshots of the user’s screen are stored, to identify the exact content visualized by the user within the specific context. The capture of these screenshots can be performed during the analysis phase or implemented using an interception mechanism that runs client-side, takes screenshots for every context change and records several type of information like keys pressed, mouse movement and events, even eye-tracking information if any hardware is available, etc.

2. Entities and data structures

2.1. Website representation

Table 1 presents a conceptual representation of data structures used to represent the website, its pages and interconnections. The table presents simple and complex entities with attributes and their corresponding conceptual representation. A transpose of these structures into a real system depends on the programming language and/or the database system used for the implementation.

Note that we consider some basic types as elementary information units due to their direct transpose on most of the actual programming languages or database systems: numeric, textual, true/false, binary, image, date-time, timestamp, time, and color.

Table 1 Website data structures

Domain	Entity	Attribute	Map	Representation	Notes
Web Sites	Site				Uses definitions (1), (2) and (3)
		Name	-	Textual	
		Uri[n]	-	List of elementary textual attributes	
		Domain	-	Textual	
		Version	-	Textual	
		Analysis Date	-	Timestamp	

Domain	Entity	Attribute	Map	Representation	Notes
		Pages[p]	-	List of complex webpage entities	Uses definition (2)
		Hyperlinks[h]	-	List of complex hyperlink entities	Uses definitions (3) and (6)
Web Pages	Page				Uses definitions (2), (3), (4), (5) and (6)
		Title	-	Textual	
		Uri[n]	A_1	List of elementary textual attributes	
		Document Type	-	Numeric	Type of document: HTML, PDF, etc.
		Raw Content	A_5	Binary	Page contents - technology independent
		Dimensions[2]	A_2	Numeric	Dimensions of the page as in client's browser
		Screenshot	A_6	Image	Screenshot as presented in the browser
		Site Level	A_3	Numeric	The shortest path from the entrance of the site to the page itself
		Dynamic	A_7	Numeric	Whether the page is dynamically generated or is a static document stored on the web server, or it is a personalized page
		Objects[o]	-	List of complex object entities	All objects contained on the page
		Referrers[r]	-	List of complex	All unique pages that

Domain	Entity	Attribute	Map	Representation	Notes
				page entities	lead to the specified page
		Links[]	-	List of complex page entities	All unique pages that can be achieved from the specifies page – children
		Hotspots[h]	-	List of complex hyperlink entities	Uses definition (2) – all objects whose interaction can change the specific page context
		Areas[a]	-	List of complex area entities	List of conceptual site areas the page is included
Statistics		Visits	A_8	Numeric	Number of times this webpage was visited during a specific time period
		Frequency	A_9	Numeric	Frequency of requests for the webpage in the given time interval
		Minimum Viewing Time	A_{10}	Time	Minimum amount of time the page was visualized by the clients
		Maximum Viewing Time	A_{11}	Time	Maximum amount of time the page was visualized by the clients
		Visualized Objects[n]	A_{12}	List of complex object entities	Most visualized areas of the page (potential information generated using an eye / motion tracking or interception system)
Page Elements	Element				

Domain	Entity	Attribute	Map	Representation	Notes
		Title	-	Textual	
		Content	-	Binary	HTML or Raw contents of the page element
		Referrer Uri	$A_{L,1}$	Textual	URI of the page the element belongs to
		Position[2]	$A_{L,2}$	Numeric	Position on the page
		Size[2]	$A_{L,2}$	Numeric	Size of the occupied area
Link Elements	Hotspot	Inherits Element			Inherits Element Uses definitions (3) and (6)
		Target Uri	$A_{L,4}$	Textual	URI of the page that is referenced
		Type	$A_{L,3}$	Numeric	Textual link, image link, composed graphical link, etc.
		Dynamic	$A_{L,5}$	Numeric	Whether is dynamically generated or not, or personalized
Statistics		Average Viewing Time	$A_{L,6}$	Time	Average time users spent to visualize the link element
		Average Visit Time	$A_{L,7}$	Time	Average time to go from one page to another using the hotspot
		Visits	$A_{L,8}$	Numeric	Number of times this link element was clicked (visited) during the analysis period
		Visual Intensity	$A_{L,9}$	Numeric	Classification of the link element on a scale, regarding focusing and attention capture

Domain	Entity	Attribute	Map	Representation	Notes
Text Elements	Text	Inherits Element			Inherits Element Uses definition (5)
		Text	-	Textual	The text
		Font	$A_{T,2}$	Complex Entity	Font characteristics (size, style, spacing, color, paragraph, reading order)
		Culture Id	$A_{T,4}$	Numeric	Culture Id (language)
		Dynamic	$A_{L,5}$	Numeric	Whether is dynamically generated or not, or personalized
Statistics		Average Reading Time	$A_{T,6}$	Time	Average time users spent to read the text
		Visual Intensity	$A_{T,7}$	Numeric	Classification of the text element on a scale, regarding focusing and attention capture
Graphical Elements	Graphical	Inherits Element			Inherits Element Uses definition (7)
		Area Type	$A_{G,2}$	Numeric	Occupied area type – square, circle, ellipse, etc.
		Colors	$A_{G,2}$	Numeric	Number of colors to be represented
		Size[2]	$A_{G,2}$	Numeric	Size of the occupied area
		Motion	$A_{G,3}$	True/False	Motion presence in the represented graphical object
		Interactive	$A_{G,4}$	True/False	Whether the object is interactive or not
		Dynamic	$A_{G,5}$	Numeric	Whether is dynamically

Domain	Entity	Attribute	Map	Representation	Notes
					generated or not, or personalized
Statistics		Average Viewing Time	$A_{G,6}$	Time	Average time users spent to visualize the text
		Visual Intensity	$A_{G,7}$	Numeric	Classification of the text element on a scale, regarding focusing and attention capture
Site Areas	Area				
		Title	-	Textual	
		Description	-	Textual	
		Color	-	Color	Representation color
		Pages[p]	-	List of complex webpage entities	The pages that belongs to the specified area

2.2. Session representation

Table 2 presents the conceptual structures required to represent a usage session, as well as the entities involved and types of mouse, keyboard or eye events that can occur during website browsing.

Table 2 Session representation structures

Domain	Entity	Attribute	Map	Representation	Notes
Sessions	Session				Uses definitions (10) and (11)
		Title	-	Textual	
		User Id	-	Complex Entity	
		Pages[p]	-	List of complex webpage entities	All unique pages visited during the session

Domain	Entity	Attribute	Map	Representation	Notes
		Page Visits[n]	-	List of complex visit entities	List of all subsequent requests made by the client during the considered interval
		First Visit	-	Timestamp	
		Last Visit	-	Timestamp	
		Entry Point Uri	-	Textual	
		Referrer Uri	-	Textual	
Page Visit	Visit				
		Page	-	Complex webpage entity	Visited page
		Timestamp	-	Timestamp	
		Viewing Time	-	Time	
		Referrer Uri	-	Textual	
		Next Visit Uri	-	Textual	Uri
		Screenshot	A_6	Image	Screenshot as presented in the client's browser
		Mouse Events[m]	-	List of complex entities	All mouse events recorded during the session, if present
		Keyboard Events[k]	-	List of complex entities	Filtered keyboard events recorded during session, if present
		Eye Tracking Events[t]	-	List of complex object entities	Eye tracking events recorded during session, if present
Site User	User				
		User Id	-	Textual	
		Full Name	-	Textual	

Domain	Entity	Attribute	Map	Representation	Notes
		Group Id[g]	-	List of complex entities	The groups the user is member of
		IP[n]	-	Textual	List of all IPs used by the user during all sessions he/she participated
		Host Names[2]	-	Textual	Hosts used by the user
Mouse Events	Mouse Event				Note: new attributes were defined to represent mouse-tracking information.
		Timestamp	$A_{M,1}$	Timestamp	
		Coordinates[2]	$A_{M,2}$	Numeric	
		Event Type	$A_{M,3}$	Numeric	Coded type of mouse event: Click, Scroll, Double Click, etc.
		Button	$A_{M,4}$	Numeric	The Id of the activated button
		Keys[n]	$A_{M,5}$	List of numeric key codes	The keys pressed during the mouse event
		Scroll Delta	$A_{M,6}$	Numeric	Scroll delta (positive or negative) relative to current position
Keyboard Events	Keyboard Event				Note: new attributes were defined to represent keyboard information.
		Timestamp	$A_{K,1}$	Timestamp	
		Keys[n]	$A_{K,2}$	List of numeric key codes	The keys pressed
Eye Tracking	Eye Event				Note: new attributes were defined to

Domain	Entity	Attribute	Map	Representation	Notes
Events					represent eye-tracking information.
		Timestamp	$A_{E,1}$	Timestamp	
		Eye	$A_{E,2}$	Numeric	Left or right eye
		Coordinates[2]	$A_{E,3}$	Numeric	
		Event Type	$A_{E,4}$	Numeric	Coded type of eye event: Eye movement, Eye hold, Reading, Scanning, etc.

2.3. Multiple sessions

Table 3 presents a possible solution for the representation of multiple usage sessions. Several other attributes can be considered to code auxiliary information, like behavioral aspects (example: considering multiple sessions that describe the same final goal, one might be interested to keep tracking of the most visited pages in a set of possible paths to the same target page, or of the deviations for the selection of most used hotspots, etc.).

Table 3 Representation of multiple sessions

Domain	Entity	Attribute	Map	Representation	Notes
Multiple Sessions	<i>Multi Session</i>				
		Title	-	Textual	
		User Id[u]	-	List of complex user entities	
		Pages[p]	-	List of complex webpage entities	All unique pages visited during selected the sessions
		Page Visits[n]	-	List of complex visit entities	List of all subsequent requests made by the clients during the considered sessions
		Filter[f]	-	List of elementary attributes or complex entities	The list of filters to be applied for filtering sessions: can be users, entry points, pages,

Domain	Entity	Attribute	Map	Representation	Notes
					timestamps, etc.
		Session[s]	-	List of complex session entities	All session in the result set after applying the filters

2.4. Visualization data structures

We identified the data structures presented in **Table 4** as suitable for a preliminary visual representation of each IL and some additional data structures.

Table 4 Visualization data structures

IL	Entity	Attribute	Map	Representation	Notes
All	Object				
		Position[4]	-	Numeric	3D localization and layer order
		Size[3]	-	Numeric	Spatial sizes
		Transparency	-	Numeric	
		Connections[n]	-	List of connector entities	List of all connections for the object
All	Connector	Inherits Object			Inherits Object
		Parent	-	Object entity	Parent object
		Child	-	Object entity	Child object
		Direction	-	Numeric	The direction of the connector
		Type	-	Numeric	Type of the connector: line, arrow, etc.
All	Page	Inherits Object			Inherits Object
		Uri	A_1	Textual	
		Level	A_3	Numeric	Default site level
		Snapshot	A_6	Image	

IL	Entity	Attribute	Map	Representation	Notes
		Referrers[n]	-	List of page entities	List of all referrers for the actual page
		Children[m]	A_L	List of page entities	List of all children for the actual page
		Connectors[n+m]	-	List of connector entities	All connectors of the page for Referrers + Children
Optional		Texts[t]	A_T	List of text entities	All texts on the page
		Hyperlinks[m]	A_L	List of hyperlink entities	All hotspots of the page
		Graphics[g]	A_G	List of graphical entities	All graphical elements of the page
Reference Map	Reference Map				
		Title	-	Textual	
		Position[4]	-	Numeric	Spatial localization and layer order
		Size[3]	-	Numeric	Spatial sizes
		Objects[u]	-	List of complex objects entities	All objects contained by the representation
		Pages[p]	-	List of complex webpage entities	All unique pages visited during selected the sessions
		Page Visits[n]	-	List of complex visit entities	List of all subsequent requests made by the clients during the considered sessions

IL	Entity	Attribute	Map	Representation	Notes
		Filter[f]	-	List of elementary attributes or complex entities	The list of filters to be applied for filtering sessions: can be users, entry points, pages, timestamps, etc.
		Session[s]	-	List of complex session entities	All session in the result set after applying the filters

3. Taxonomical classification of visualization methods

We used the approximation of several aspects to classify visualization methods as introduced by [Nunes2006], a combination of the proposals of [Shneiderman1996], [Keim2001] and [Andrews2002]:

- I. Type of data;
- II. Visualization techniques; and
- III. Interaction and distortion techniques.

These three types of representations can be graphically illustrated as presented in Figure 1. A general approximation would consider one visualization method for each intersections of the three axis in a three dimensional space. Moreover, if we consider Z-axis as task oriented, we can associate a unique visualization method defined by the intersection of X and Y-axis to a unique state, defined as a task on Z-axis.

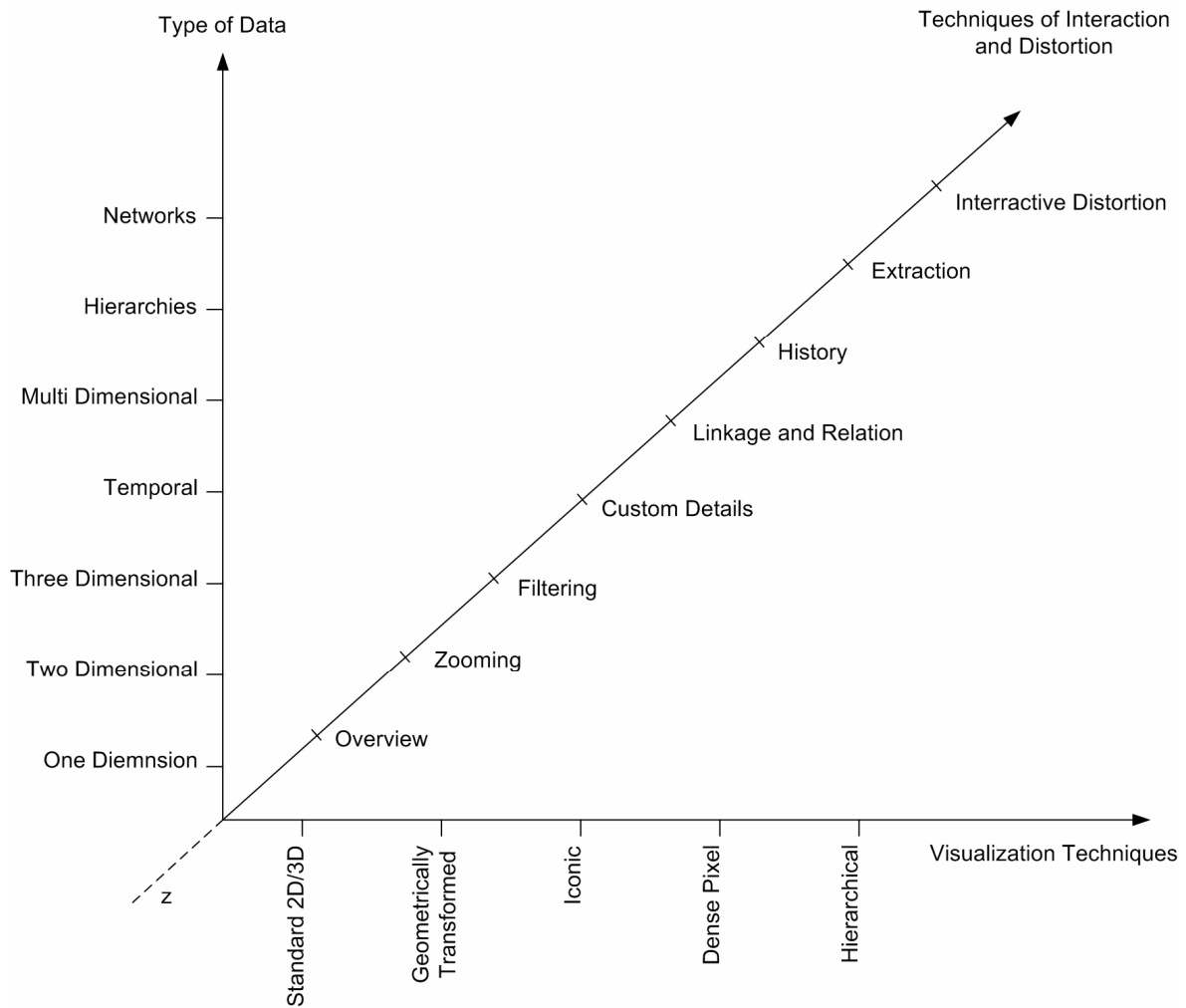


Figure 1 Visualization taxonomy, for each dimension – a type of classification [source: [Nunes2006], Figura III-1]

However, we can approximate a unique combination of these three dimensions as an information layer (IL). Several combinations of ILs can be selected to achieve a complex visual representation of the information. In addition, several techniques of interaction or distortion can be applied to such a complex representations to improve the exploration and turn the presented information more perceivable.

Table 5 Visualization methods classification for structural website / session analysis

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
Site Pages	Hierarchies	Iconic	Overview	Represents all unique site pages described using page attributes from A set. Several iconic representations are produced from $A_1 + A_2 + A_6$ attributes. The icons are represented on a scalable grid, each page being represented only once. Additional attributes can be coded to represent structural and extended attributes sets.

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
Site Structure (2D)	Hierarchies	Hierarchical / Iconic	Overview	<p>Represents all unique site pages described using attributes from A combined with A_L sets. Several iconic representations are produced from $A_1 + A_2 + A_3 + A_6$ attributes. The icons are grouped, in a two dimensional space, on columns that represent site levels (A_3).</p> <p>Two approaches can be used to represent the pages: one that represents a page as many times as it appears a link of a referrer page; the second that represents the page only once, on the lowest level of the site, and uses connections to all other referrer pages. The second approach is more suitable for a two dimensional space, being given the space occupied by each additional icon.</p> <p>The connections between pages are obtained based on $A_{L,1}$, $A_{L,2}$ and $A_{L,4}$ attributes of link elements. For each pair ($A_{L,1}$, $A_{L,4}$) is represented a connector with the direction from page specified by $A_{L,1}$ to the page specified by $A_{L,4}$.</p> <p>Additional structural and statistical information can be coded based on A_8, A_9, A_{10}, A_{11}, $A_{L,6}$, $A_{L,8}$ attributes, to represent number of visits, viewing time, paths usage, etc.</p>
Site Structure 3D (Holistic 3D view)	Hierarchies	Hierarchical	Overview	<p>Represents all unique site pages described using attributes from A combined with A_L sets. Several three dimensional representations of page objects are produced from $A_1 + A_2 + A_3 + A_6$ attributes. The 3D objects can be grouped, in a three dimensional space, on rows and columns that represent site levels (A_3). Some possible approaches are to organize each subsequent level for page links rotated to 90° / -90° or to use conical representations for each page links.</p> <p>Two approaches can be used to represent the pages: one that represents a page as many times as it appears a link of a referrer page; the second that represents the page only once, on the lowest level of the site, and uses connections to all other referrer pages. The first approach is more suitable for a three dimensional space, being given the facts that more objects can be represented on a three dimensional space and that the second approach might introduce the occlusion problems for backward connections.</p> <p>The connections between pages are obtained based on $A_{L,1}$, $A_{L,2}$ and $A_{L,4}$ attributes of link elements. For each pair ($A_{L,1}$, $A_{L,4}$) is represented a connector with the direction from page specified by $A_{L,1}$ to the page specified</p>

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
				<p>by $A_{L,4}$.</p> <p>Additional structural and statistical information can be coded based on $A_8, A_9, A_{10}, A_{11}, A_{L,6}, A_{L,8}$ attributes, to represent number of visits, viewing time, paths usage, etc.</p>
Hovering Tips	2D	Standard 2D	Custom Details	<p>Represents complementary information for most of visualization methods. The attributes sets are coded overlaid on a visualization method, usually activated by a hovering event. Several structural and extended attributes from $A \cup A_T \cup A_L \cup A_G$ can be used to represent each layer. For instance, $A_2 + A_6 + A_8$ attributes can be used to represent the Uri, screenshot and number of visits of the hovered page object; $A_{L,6}, A_{L,7}, A_{L,8}$ can be used to represent the viewing time, visit time and number of visits for the hovered link (represented as a connector between two page objects).</p>
Linkage Elements	2D	Standard 2D	Linkage and Relation	<p>Represents a subset of attributes from A combined with A_L sets, coded overlaid on a visualization method, usually activated by a hovering event over a link element or a page. In the first particular case, the relations between pages are obtained based on $A_{L,1}, A_{L,2}$ and $A_{L,4}$ attributes of link elements. For each linkage pair $(A_{L,1}, A_{L,4})$, the visual connector is represented highlighted, with the direction from page specified by $A_{L,1}$ to the page specified by $A_{L,4}$, that uses highlighted representation. The second case highlights the incoming and outgoing connections for the hovered page, as well as the page itself.</p> <p>This kind of visual feedback is particularly useful for complex representations where the level of detail does not allow one to visually distinguish the relations between objects.</p> <p>Additional structural and statistical information can be coded based on $A_8, A_9, A_{10}, A_{11}, A_{L,6}, A_{L,8}$ attributes, to represent number of visits, viewing time, paths usage, etc.</p>
Path to Goal	2D	Standard 2D	Linkage and Relation	<p>Represents all possible relations of two pages, coded on demand as a distinct visualization method that uses a subset of attributes from A combined with A_L sets.</p> <p>Two pages define the starting point and the ending point of the search scope, the goal being to represent all possible paths between these pages.</p> <p>The relations between pages are obtained based on $A_{L,1}$ and $A_{L,4}$ attributes of link elements. For each possible linkage pair $(A_{L,1}, A_{L,4})$ that is part of a possible path, the</p>

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
				<p>visual connector is represented with the direction from page specified by $A_{L,1}$ to the page specified by $A_{L,4}$.</p> <p>The starting point and the ending point are highlighted and positioned at the limits of the representation. All other connections are represented as connections between subsequent page objects that lead from the starting point to the ending point.</p> <p>Additional structural and statistical information can be coded based on $A_8, A_{L,6}, A_{L,8}$ attributes, to represent number of visits, paths usage, etc.</p>
Session History	Temporal	Hierarchical	Overview	<p>Represents all pages visited by the user during a session. Several visual representations are produced from $A_1 + A_2 + A_6$ attributes. The pages are represented on a timeline, indicating the timings between all subsequent page visits. These timings can be coded in the distance between pages and/or can be represented directly on the connectors.</p> <p>Being given the history of subsequent page visits for the session, the connections between visited pages are obtained based on $A_{L,1}, A_{L,2}$ and $A_{L,4}$ attributes of link elements, graphically coded over the visual representation of the page.</p> <p>Additional statistical information can be coded using the extended attributes sets for link, text and graphical elements.</p>
Session History 3D	Temporal	Standard 3D	Overview	<p>Represents all pages visited by the user during a session. Several visual representations are produced from $A_1 + A_2 + A_6$ attributes. The pages are represented on a timeline, indicating the timings between all subsequent page visits. These timings can be coded in the distance between pages on the Z order.</p> <p>The contexts between subsequent visits can be obtained based on $A_{L,1}, A_{L,2}$ and $A_{L,4}$ attributes of link elements, the areas of interaction being highlighted somehow.</p> <p>Additional statistical information can be coded using the extended attributes sets for link, text and graphical elements.</p>
Tree-structured traversing in time	Temporal	Hierarchical	Overview	<p>Combines the representation models for the structure of the website and for the session to obtain a temporal representation of a user session.</p> <p>The representation maps each visited page to its corresponding level on the website structure; the levels are represented as columns in a table structure. The timings between subsequent page visits are coded using distance,</p>

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
				<p>each subsequent page visit being represented as a row in the overall table.</p> <p>Additional information is coded using color and shape, to represent specific page visits as the beginning and the end of the session, or to highlight discontinuous visits: when the user jumps out of the context of the website's domain.</p> <p>Several structural and statistical attributes for page and link elements are used to transform raw data into visual features.</p> <p>This method is suitable for detecting atypical jumps inside or outside the site structure: a jump back over several levels might highlight an inconsistency on the site linkage, the user being unable to find the desired page.</p>
Page Linkage	Network	Standard 2D	Overview	<p>This representation is suitable to highlight all possible linkage information for a subset of pages of the website. Several iconic representations are produced from $A_1 + A_2 + A_3 + A_6$ attributes, for each page in the selection. The icons are represented on a circle in a two dimensional space and the linkage information is represented as oriented connectors.</p> <p>The connections between pages are obtained based on $A_{L,1}$, $A_{L,2}$ and $A_{L,4}$ attributes of link elements. For each pair $(A_{L,1}, A_{L,4})$ is represented a connector with the direction from page specified by $A_{L,1}$ to the page specified by $A_{L,4}$.</p> <p>Additional structural and statistical information can be coded based on A_8, $A_{L,8}$ attributes, to represent number of visits.</p>

Table 6 Visualization methods classification for visual inspection of visual workspace coherence

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
Page Areas	2D	Standard 2D	Overview	Represents webpage contents using the identification of page elements described by the reunion of $A_T \cup A_L \cup A_G$ attributes sets. These attributes are usually coded on top of a Reference Map identified by $A_2 + A_6$ attributes of the page itself.
Interactive Zones	2D or 3D	Hierarchical	Linkage and Relation	Represents webpage outgoing linkage information described by A_L combined with A attributes sets. These attributes are usually coded on top of a BI identified by

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
				$A_2 + A_6$ attributes of A set. $A_{L,2}, A_{L,4}, A_{L,6}, A_{L,7}, A_{L,8}, A_{L,9}$ can be coded to represent positioning, linkage and statistical information.
Page Relations	2D	Hierarchical	Linkage and Relation	Represents webpage incoming linkage information described by A_L combined with A attributes sets. These attributes are usually coded on top of several BI identified by $A_2 + A_6$ attributes of A set. $A_{L,1}, A_{L,2}, A_{L,4}$ are coded on top of BI for the pages identified by $A_{L,1}$ to represent positioning and linkage information. Additionally, $A_{L,6}, A_{L,7}, A_{L,8}, A_{L,9}$ can be coded to represent statistical information.
Hovering Tips	2D	Standard 2D	Custom Details	Represents complementary information for most of visualization methods. The attributes sets are coded overlaid on a visualization method, usually activated by a hovering event. Several structural and extended attributes from $A \cup A_T \cup A_L \cup A_G$ can be used to represent each layer, e.g. $A_2 + A_6 + A_8$ attributes can be used to represent the Uri, screenshot and number of visits of the hovered page object.
Eye-Tracking Layers	Temporal	Geometrically Transformed	Overview / Linkage and relation	Represents complementary information for webpage contents; uses eye-tracking information captured during webpage usage sessions, visually coding the events as symbolic related icons. The information is usually coded on top of a Reference Map (Base Image - BI) identified by $A_2 + A_6$ attributes of the page itself.
Mouse-Tracking Layers	Temporal	Geometrically Transformed	Overview / Linkage and relation	Represents complementary information for webpage contents; uses mouse-tracking information captured during webpage usage sessions, visually coding the mouse events as symbolic icons. The events are related by simplified cursor movement paths. The information is usually coded on top of a Reference Map (Base Image - BI) identified by $A_2 + A_6$ attributes of the page itself.
2D Links Usage	Temporal	Geometrically Transformed	Overview	Represents webpage hotspots usage history described by A_L combined with A attributes sets. These attributes are usually coded on top of one BI obtained by combining $A_2 + A_6$ attributes for all involved pages. Linkage information is represented using $A_{L,2}$ structural attributes, connecting two subsequent hotspots, for each page identified by $A_{L,1}$. Additionally, $A_{L,6}, A_{L,7}, A_{L,8}, A_{L,9}$ can be coded to represent statistical information.
3D Links Usage	Temporal	Geometrically Transformed	Overview	Represents webpage hotspots usage history described by A_L combined with A attributes sets. These attributes

Method Name	Taxonomy			Description
	ToD	VisTech	TID	
				are usually coded on top of several BI identified by $A_2 + A_6$ attributes of A set. These BI are represented in a 3D space, one on top of the other; linkage information is represented using $A_{L,2}$ structural attributes, connecting two subsequent BI, for each page identified by $A_{L,1}$. Additionally, $A_{L,6}$, $A_{L,7}$, $A_{L,8}$, $A_{L,9}$ can be coded to represent statistical information.
Interaction Workspace	2D	Standard 2D	Dense Pixel / Overview	Represents the visual workspace for the entire web site. Combines the A_2 attribute for all pages of the web site to obtain the reference map. Then overlays an information layer obtained by combining all $A_{L,2}$ structural attribute for all link elements in all pages. Areas with more hotspots give pixels denser regions.

4. Site Analyzer and Compiler Conceptual Models

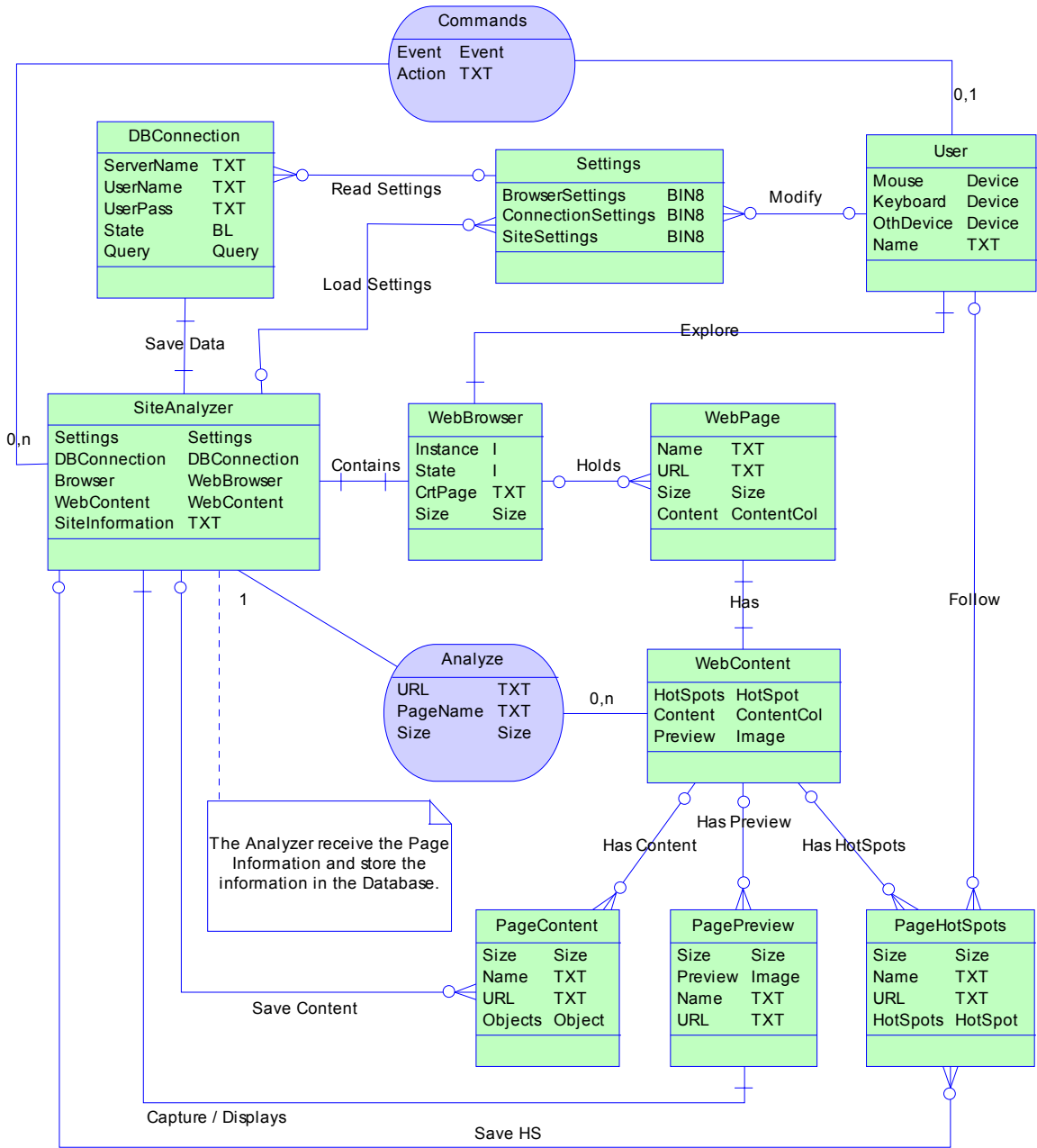


Figure 2 Site Analyzer conceptual model

Compiler

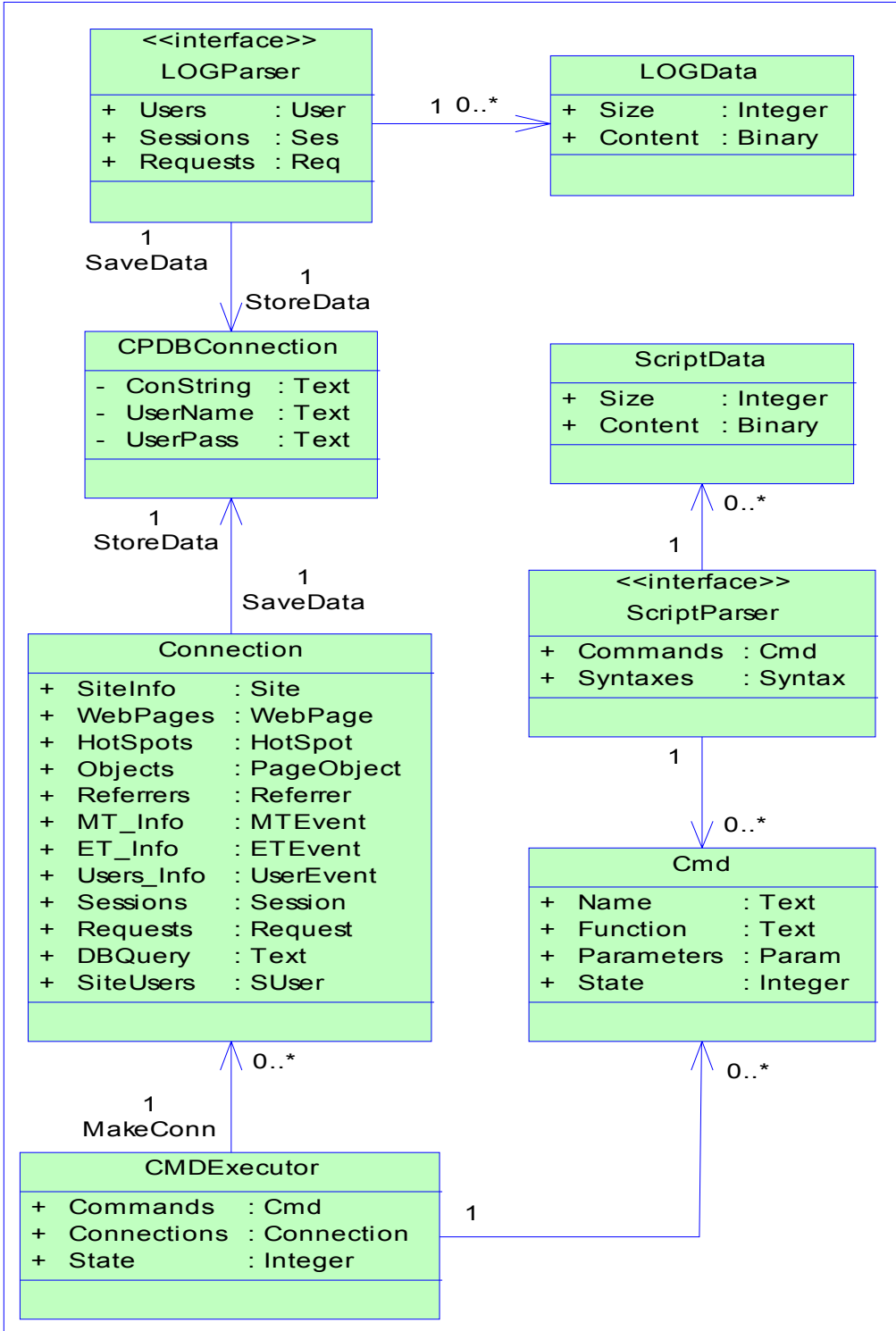


Figure 3 Compiler conceptual model

5. Detailed Data Structures

```
namespace Visualizer
{
class CWebSite;
class CWebPage;
class CHotSpot;
class CReferrer;
class CPageObject;
class CUserEvent;
class CEyeTrEvent;
class CMouseTrEvent;
class CPageStatistics;
class CPageLink;
class CSiteArea;

class CReferrer
{
public:
    // Default constructor.
    CReferrer()
    {
        m_dwReferrerID = 0;
        m_dwReferrerPageID = 0;
        m_strReferrerName = "";
        m_strReferrerURL = "";
        m_dwReferrerUsage = 0; // A Counter
        m_pRefWebPage = NULL;
    };
    // Destructor.
    virtual ~CReferrer()
    {
    };

    friend class CWebPage;

    // Attributes
public:
    DWORD m_dwReferrerID;
    DWORD m_dwReferrerPageID;
    CString m_strReferrerName;
    CString m_strReferrerURL;
    DWORD m_dwReferrerUsage; // A Counter
    CWebPage* m_pRefWebPage; // The Pointer to the Page
};

class CHotSpot
{
public:
    // Default constructor.
    CHotSpot()
    {
        m_dwSpotID = 0;
        m_strSpotName = "";
        m_strSpotURL = "";
        m_strSpotType = "";
        m_bIsDynamic = false;
        top = left = right = bottom = dXRadius = dYRadius = 0.0;
        m_dwSpotUsage = 0; // A Counter
        m_pLinkedPage = NULL;
    };
    // Destructor.
    virtual ~CHotSpot()
    {
    };

    friend class CWebPage;

    // Attributes
public:
    DWORD m_dwSpotID;
    CString m_strSpotName;
    CString m_strSpotURL;
    CString m_strSpotType;
    bool m_bIsDynamic;
    double top, left, right, bottom, dXRadius, dYRadius;
    DWORD m_dwSpotUsage; // A Counter
    CWebPage* m_pLinkedPage; // The Pointer to the Page
};

class CPageObject
```



```

{
public:
    // Default constructor.
    CPageObject()
    {
        m_dwObjectID = 0;
        m_strObjectName = "";
        m_strObjectURL = "";
        m_strObjectType = "";
        m_bIsDynamic = false;
        top = left = right = bottom = fXRadius = fYRadius = 0;
        m_dwObjectUsage = 0; // A Counter
    };
    // Destructor.
    virtual ~CPageObject()
    {
    };

    friend class CWebPage;

    // Attributes
public:
    DWORD m_dwObjectID;
    CString m_strObjectName;
    CString m_strObjectURL;
    CString m_strObjectType;
    bool m_bIsDynamic;
    float top, left, right, bottom, fXRadius, fYRadius;
    DWORD m_dwObjectUsage; // A Counter
};

class CUserEvent
{
public:
    // Default constructor.
    CUserEvent()
    {
        m_dwUEID = 0;
        m_dwUserID = 0;
        m_dwSessionID = 0;
        m_dtDateTime = COleDateTime::COleDateTime(2000,1,1,0,0,0);
        m_nPosX = m_nPosY = 0;
        m_nDevice = -1;
        m_strEvent = "";
        m_strEvtContent = "";
        m_PageSize = m_ScrollPos = CSize(0,0);
        m_tsTimeStamp = CTimeSpan::CTimeSpan(0,0,0,0); // Length of the event
    };
    // Destructor.
    virtual ~CUserEvent()
    {
    };

    friend class CWebPage;

    // Attributes
public:
    DWORD m_dwUEID;
    DWORD m_dwUserID;
    DWORD m_dwSessionID;
    COleDateTime m_dtDateTime;
    int m_nPosX, m_nPosY;
    int m_nDevice;
    CString m_strEvent;
    CString m_strEvtContent;
    CSize m_PageSize, m_ScrollPos;
    CTimeSpan m_tsTimeStamp; // Length of the event
};

class CEyeTrEvent
{
public:
    // Default constructor.
    CEyeTrEvent()
    {
        m_dwETID = 0;
        m_dwUserID = 0;
        m_dtDateTime = COleDateTime::COleDateTime(2000,1,1,0,0,0);
        m_nPosX = m_nPosY = 0;
        m_nEye = 0;
        m_strEvent = "";
        m_tsTimeStamp = CTimeSpan::CTimeSpan(0,0,0,0); // Length of the event
    };
};

```

```

        // Destructor.
        virtual ~CEyeTrEvent()
        {
        };

        friend class CWebPage;

        // Attributes
public:
    DWORD m_dwETID;
    DWORD m_dwUserID;
    COleDateTime m_dtDateTime;
    int m_nPosX, m_nPosY;
    int m_nEye;
    CString m_strEvent;
    CTimeSpan m_tsTimeStamp; // Length of the event
};

class CMouseTrEvent
{
public:
    // Default constructor.
    CMouseTrEvent()
    {
        m_dwMTID = 0;
        m_dwUserID = 0;
        m_dtDateTime = COleDateTime::COleDateTime(2000,1,1,0,0,0);
        m_nPosX = m_nPosY = 0;
        m_nButton = 0;
        m_nWheelDelta = 0;
        m_strEvent = "";
        m_tsTimeStamp = CTimeSpan::CTimeSpan(0,0,0,0); // Length of the event
    };
    // Destructor.
    virtual ~CMouseTrEvent()
    {
    };

    friend class CWebPage;

    // Attributes
public:
    DWORD m_dwMTID;
    DWORD m_dwUserID;
    COleDateTime m_dtDateTime;
    int m_nPosX, m_nPosY;
    int m_nButton;
    int m_nWheelDelta;
    CString m_strEvent;
    CTimeSpan m_tsTimeStamp; // Length of the event
};

class CPageLink
{
public:
    CPageLink()
    {
        m_dwLinkID = 0; // Link ID
        m_dwLinkUsage = 0; // A Counter
        m_strLinkURL = "";
        m_pLinkPage = NULL;
    };
    virtual ~CPageLink() {} // Destructor.

    friend class CWebPage;

    // Methods
public:
    CString GetLinkURL() { return m_strLinkURL; }

    // Attributes
public:
    DWORD m_dwLinkID; // Link ID
    CString m_strLinkURL; // The URL
    DWORD m_dwLinkUsage; // A Counter
    CWebPage* m_pLinkPage;
};

class CPageStatistics
{
public:
    CPageStatistics() // Default constructor.
    {

```

```

        m_dwStatID = 0;
        m_dwUserID = 0;
        m_dwUserVisits = 0; // A Counter
        m_dwUserReferrersID = 0;
        m_dwUserHotSpotsID = 0;
    }
    virtual ~CPageStatistics() {}; // Destructor.

    friend class CWebPage;

    // Attributes
public:
    DWORD m_dwStatID;
    DWORD m_dwUserID;
    DWORD m_dwUserVisits; // A Counter
    DWORD m_dwUserReferrersID;
    DWORD m_dwUserHotSpotsID;
};

class CSiteArea
{
public:
    // Default constructor.
    CSiteArea()
    {
        m_dwAreaID = 0;
        m_dwSiteID = 0;
        m_strAreaName = "";
        m_strAreaDescription = "";
        m_strAreaNotes = "";
    };
    // Destructor.
    virtual ~CSiteArea()
    {
    };

    friend class CWebPage;

    // Attributes
public:
    DWORD m_dwAreaID;
    DWORD m_dwSiteID;
    CString m_strAreaName;
    CString m_strAreaDescription;
    CString m_strAreaNotes;
};

// The Webpage Info
class CWebPage
{
public:
    CWebPage(); // Default constructor.
    virtual ~CWebPage(); // Destructor.

    friend class CWebSite;

    // Methods
public:
    void FreeMem();
    bool FreeExtraInfo();

    void AddChildLink(CPageLink* pLink) { if (pLink) m_aryChilds.Add(pLink); };
    void AddParentLink(CPageLink* pLink) { if (pLink) m_aryParents.Add(pLink); };
    void AddReferrer(CReferrer* pRef) { if (pRef) m_aryPageReferrers.Add(pRef); };
    void AddObject(CPageObject* pobject) { if (pobject) m_aryPageObjects.Add(pobject); };
};

    void AddHotSpot(CHotSpot* pHotSpot)
    {
        if (pHotSpot)
        {
            m_aryPageHotSpots.Add(pHotSpot);
            m_aryHotSpotsByURL.Add(pHotSpot->m_strSpotURL, pHotSpot);
        }
    };
    void AddStatistic(CPageStatistics* pStat) { if (pStat)
m_aryPageStatistics.Add(pStat); };
    void AddUserEvent(CUserEvent* pUserEv) { if (pUserEv)
m_aryPageUserEventInfo.Add(pUserEv); };
    void AddETEvent(CEyeTrEvent* pEyeTrEv) { if (pEyeTrEv)
m_aryPageEyeTrackInfo.Add(pEyeTrEv); };
    void AddMouseEvent(CMouseEvent* pMouseTrEv) { if (pMouseTrEv)
m_aryPageMouseTrackInfo.Add(pMouseTrEv); };
};

```

```

void SetParentPage(CWebPage* pPage) {m_pParentWebPage = pPage; };

DWORD GetID() { return m_dwPageID; };
void SetID(DWORD dwID) { m_dwPageID = dwID; };
DWORD GetPageID() { return m_dwPageID; };
void SetPageID(DWORD dwID) { m_dwPageID = dwID; };

bool PageHasURL(CString strPageURL);
int GetURLLevel(CString strPageURL); // Returns the Level where the URL of this page
is placed

CHotSpot* FindHotSpot(CString strSpotURL);

// Attributes
public:
DWORD m_dwPageID;
DWORD m_dwSiteID;
DWORD m_dwAreaID;
DWORD m_dwPageLevel;
CString m_strPageName;
CString m_strPageURL;
// For each extra URL add the Level where the page with that URL appears
// The Level for each URL are also contained in this array
CHashTable m_aryAllPageURL;
DWORD m_dwSecurityLevelID;
DWORD m_dwSizeX;
DWORD m_dwSizeY;
DWORD m_dwPageVisits; // A Counter
DWORD m_dwContentSize;
LPVOID m_pContentData;
CDImage* m_pScreenshotImage;
Bitmap* m_pThumbnailImage;
COLORREF m_areaColor;

bool m_bIsKindOfFile; // DOC, PDF... etc. file
CString m_strPageType; // HTML, ASP, PHP, etc., (The Extension of the file if
is a file)
bool m_bIsDynamic;
bool m_bIsStored;
bool m_bIsAnalyzed;

DWORD m_dwReferrersCount;
bool m_bHasETInfo; // Eye Track
bool m_bHasMTInfo; // Mouse Track
bool m_bHasCTInfo; // Content Catalog
bool m_bHasUEInfo; // User Events

// The Parent page used for representing the levels
CWebPage* m_pParentWebPage;

CString m_strSiteArea;

CPtrArray m_aryChilds; // class CPageLink;
CPtrArray m_aryParents; // class CPageLink;

CPtrArray m_aryPageReferrers; // class CReferrer;
CPtrArray m_aryPageHotSpots; // class CHotSpot;
CHashTable m_aryHotSpotsByURL; // class CHotSpot;
CPtrArray m_aryPageObjects; // class CPageObject;
CPtrArray m_aryPageStatistics; // class CPageStatistics
CPtrArray m_aryPageEyeTrackInfo; // class CEyeTrEvent;
CPtrArray m_aryPageMouseTrackInfo; // class CMouseTrEvent;
CPtrArray m_aryPageUserEventInfo; // class CUserEvent;

// Instance Status Info
bool m_bPageThumbnailLoaded;
bool m_bPageScreenShotLoaded;
bool m_bPageHotSpotsLoaded;
bool m_bPageObjectsLoaded;
bool m_bPageStatisticsLoaded;
bool m_bPageReferrersLoaded;
bool m_bPageEyeTrackLoaded;
bool m_bPageMouseTrackLoaded;
bool m_bPageUserEventsLoaded;

};

class CWebSite
{
public:
CWebSite(); // Default constructor.
virtual ~CWebSite(); // Destructor.

```

```

        // Attributes
public:
    DWORD m_dwSiteID;
    CString m_strSiteName;
    CString m_strSiteURL;
    CString m_strSiteIP;
    CStringArrayEx m_arySiteAllURLs;
    CStringArrayEx m_arySiteIPs;
    CString m_strSiteDomain;
    CString m_strSiteDescription;
    CString m_strSiteExtraInfo;
    CString m_strSiteRegistrant;
    CString m_strSiteVersion;
    COleDateTime m_dtVersionDate;
    bool m_bIsAnalyzed;
    bool m_bIsFullCaptured;
    DWORD m_dwSiteLevels;
    DWORD m_dwPagesCount;
    DWORD m_dwPagesCaptured;

    CHashTable m_arySitePages; // The WebSite Pages
    CPtrArray m_arySiteLevels; // The WebSite Levels

    // Used for Indexing Services
    CHashTable m_aryPagesByID; // The WebSite Pages

    // Instance Status Info
    bool m_bSiteStructureLoaded;

    CHashTable m_arySelectedPages;

// Methods
public:
    DWORD GetID() { return m_dwSiteID; };
    void SetID(DWORD dwID) { m_dwSiteID = dwID; };

    bool AddWebPage(CString strPageURL, CWebPage* pPage);
    void BuildPagesInterconnections();
    void BuildSiteLevelsInfo();

    CWebPage* SelectPage(CString strPageURL);
    CWebPage* SelectPage(DWORD dwPageID);
    CWebPage* UnSelectPage(CString strPageURL);
    CWebPage* UnSelectPage(DWORD dwPageID);
    bool SelectLevel(DWORD dwLevelID);
    bool UnSelectLevel(DWORD dwLevelID);
    void ClearSelections() { m_arySelectedPages.RemoveAll(); };
    int GetSelectionCount() { return m_arySelectedPages.GetSize(); };
    CHashTable* GetSelectedPages() { return &m_arySelectedPages; };

    int GetSitePagesCount() { return m_arySitePages.GetSize(); };
    int FindPageName(CString strPageName, int nStartAt = 0, bool bMatchCase = false,
bool bMatchWholeWord = false);
    int FindPageURL(CString strPageURL, int nStartAt = 0, bool bMatchCase = false, bool
bMatchWholeWord = false);
    int FindPageID(DWORD dwID);

    CWebPage* FindPage(CString strPageURL);
    CWebPage* FindPage(DWORD dwID);
// Methods
public:
    void FreeMem();
};
} // Namespace Visualizer

```

6. SharePoint Analyzer / Designer Implementation Details

6.1. Application components

The application is formed of three major deployable components:

- *SPSFunctions* – library that provides the access to manipulate SharePoint Portal Sites / Objects. It is mainly used on the server side, but it still provides remote capabilities;

- *SPDesignerCtrl* – UI control that provides the basic functionality of a hierarchical tree representation. It is enriched with a set of functionalities, intentionally developed for this purpose;
- *SPDesignerApp* – Windows application that represents the user interface and interaction mechanism of the main application. It also includes the main framework for Xml manipulations and SharePoint Portal creation (reverse engineer) capabilities.

6.2. Used technologies

SharePoint Portal Server 2003 and Windows SharePoint Services are built on the 1.1 version of the Microsoft .NET Framework. Therefore, because of some incompatibilities between the latest version of the development platform and SPS/WSS, the decision to develop the *SPSFunctions* library using .Net Framework V1.1 and Visual Studio .NET 2003. The other components were developed using the latest .NET development platform, running the second version of the .NET Framework, the Visual Studio .NET 2005 Team System. The version of the IDE (Integrated Development Environment) proved to be an efficient tool for RAD (Rapid Application Development), supported by a powerful revision of the .NET Framework. The new Graphical UI Design capabilities of the new platform were used to produce a flexible UI for the application.

6.3. Programming Model – Key Concepts

Since the application was designed keeping a good relation of flexibility-extensibility-modularity, meaning:

- *Flexibility* – provided by the decision of implementing the technical site/list/area/permission/meeting/webpart/webpage configurations using both xml schema and implementation, validations being used to check the compliance with the schema;
- *Extensibility* – provided by the unique feature that combines all the definitions of the portal objects in final portal xml specifications, interpreted by an extensible compiler. Each component can be extended to fulfill new requirements, the integration within the final portal specification being an automated process;
- *Modularity* – each component was designed as standalone, communication between modules being realized using xml objects, most of the component receiving a xml specification as input and providing the results as xml or action results (example the *SpXmlCompiler* component receives the portal specification xml and generated the portal structure on the SharePoint Server; *PortalXmlBuilder* receives as input the graphical representation of the Portal and produces the portal xml specification).

In order to successfully accomplish these concepts, a set of rules and programming models and design patterns were used:

- Use of asynchronous *delegates* (events) to communicate between the closed components and the outside world. This means that every time we need to

communicate the state of an action to the real world, we have to use a delegate function that passes parameters to the object's outside world (e.g. *SPDesignerControl*, *SPXmlCompiler*, *SPTreeBuider*);

- Use of *Proxy* design pattern to hide information irrelevant to the real world. Example: a component might provide a reach set of interfaces and properties, few of them being needed to be accessed to a specific set of users; for this, is implemented a proxy that provides that users only the properties and interfaces they are allowed to access(e.g. *Proxy* used to hide *SPDesignerCtrl* properties);
- Use of the *multi-threading* model rather than the *single-thread*. This option allows a better use of the hardware resources and control of simultaneously enabled taska, even if the implementation complexity is raised one level (e.g. *SPXmlCompiler*, *XmlPortalTreeBuider*, etc.);
- Extensive use of *generalization/specialization*, *polymorphism and encapsulation*, OOP design concepts facilitating the tremendously the design of a graphical designer (in our case 2/3 and fur levels of inheritance were used) (e.g. *Entity*, *SPObject*, *ShapeBase*, *ShapeRectangle*, *Connection*, etc.);
- Use of *WebServices* for communication with the server, asynchronous webservice consuming making event handling more easier to be controlled (e.g. *AreaService*, *SitesService*, *ListsService*, etc.);
- Use of *messaging techniques* and *specialized interfaces* to communicate between components (in our case a set of Interfaces were used to communicate with the Design UI Control, both for implementing tree traversals and other information exchange capabilities) (e.g. *SPDesignerCtrl*, *SPXmlCompiler*, *ShapeCollection*, etc.);
- Separated UI and Data manipulation components, in this case the UI was always a separated component while the data manipulation was implemented by *Hashed Collections* (e.g. *SPDesignerCtrl*, *ShapeCollection*, *ConnectionCollection*, etc.);
- Use of *Xml Schema* and *Xml Definitions* to represent objects properties rather that built-in properties for the objects. This will facilitate the validation o the object specification and increase its extensibility and flexibility (e.g. *SPObject*, *SPSite*, *SPList*, *SPArea*, etc.);
- Use of *Serialization* and *Deserialization* to store the object's properties, using generic streams rather than specialized ones (e.g. *GraphSerializer*, *GraphDataAttribute*, etc.);

These and many other programming models have been used to fulfill our task.

6.4. Components

The solution was developed taking into account the complexity involved by the maintenance of such an application development cycle. A three-tier conceptual organization was provided for a more flexible implementation.

A simple approach was used to present the main code components involved on the development process. Visual Studio's Class Diagram was used to draw the simplified versions of the development objects: Classes, Abstract Classes, Enumerations, Attributes, Collections, Interfaces, Forms and Delegates, as presented in Figure 4, Figure 5, Figure 6, Figure 7 and the expanded class diagram presented in Figure 8;

The purpose of this document does not allow us to make a full description of these components, instead, a simple enumeration of their roles would describe them:

- *Entity, Connection, Connector, SPObject, ShapeBase, ShapeRectangle, OvalShape, TextLabel, ParentChild* are graphical object used to represent the tree control shapes;
- *DataType, EdgeType, GraphType, NodeType, GraphDataAttributes, GraphDataCollection, GraphSerializer* are used for Xml Serialization / Deserialization;
- *ConnectionCollection, ConnectorCollection, ShapeCollection, ParentChildCollection* are basic collection for data storage purposes;
- *GraphAbstract, SPDesignerCtrl, Proxy* are used for the Tree Graphical Control;
- *IVisitor, IPrepostVisitor, DeleteVisitor, ExpanderVisitor* are synchronous event-based interfaces and classes used for BreadthFirst and DepthFirts traversals of the tree structure;
- *frmInput.cs, frmLogin.cs, SPObjectEditor.cs, SPDesignerApp.Designer.cs, SPDesignerApp.cs, SPObjectEditor.Designer.cs, Program.cs* represent the Windows Forms used by the application;
- *GenericXmlBuilder.cs, InfoVisitor.cs, ListViewColumnSorter.cs, PortalTreeColoring.cs, PortalXmlBuilder.cs, SPSPortalInfo.cs, SPXmlCompiler.cs, XmlEventLogger.cs, XmlFormatter.cs, XmlPortalTreeBuider.cs, XmlTreeBuider.cs, XmlTreeColoring.cs and XmlUtils.cs* are classed specialized in tree traversal and coloring, xml manipulation, Portal creation and reverse engineering, etc.

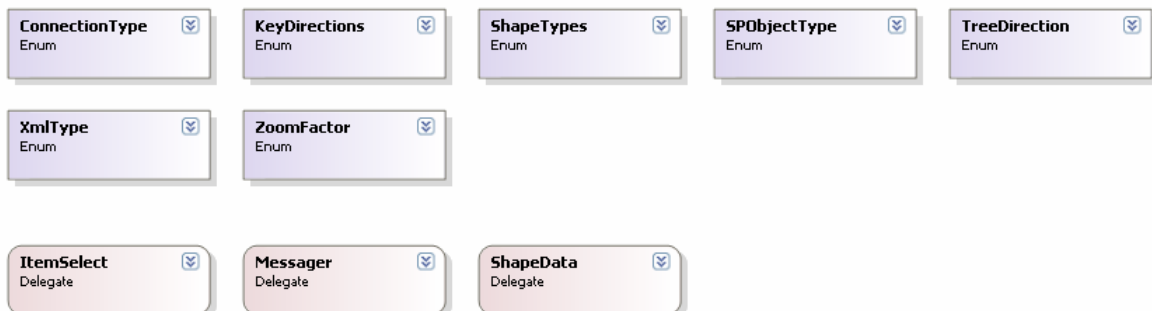


Figure 4 Flowchart Control basic components

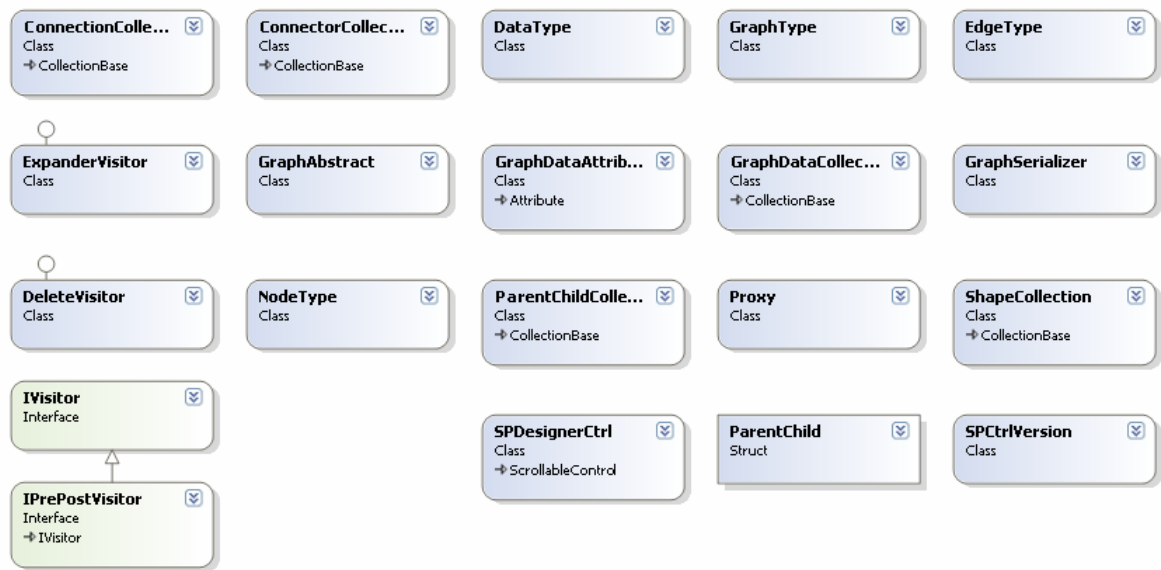


Figure 5 Flowchart Control data / connecting components

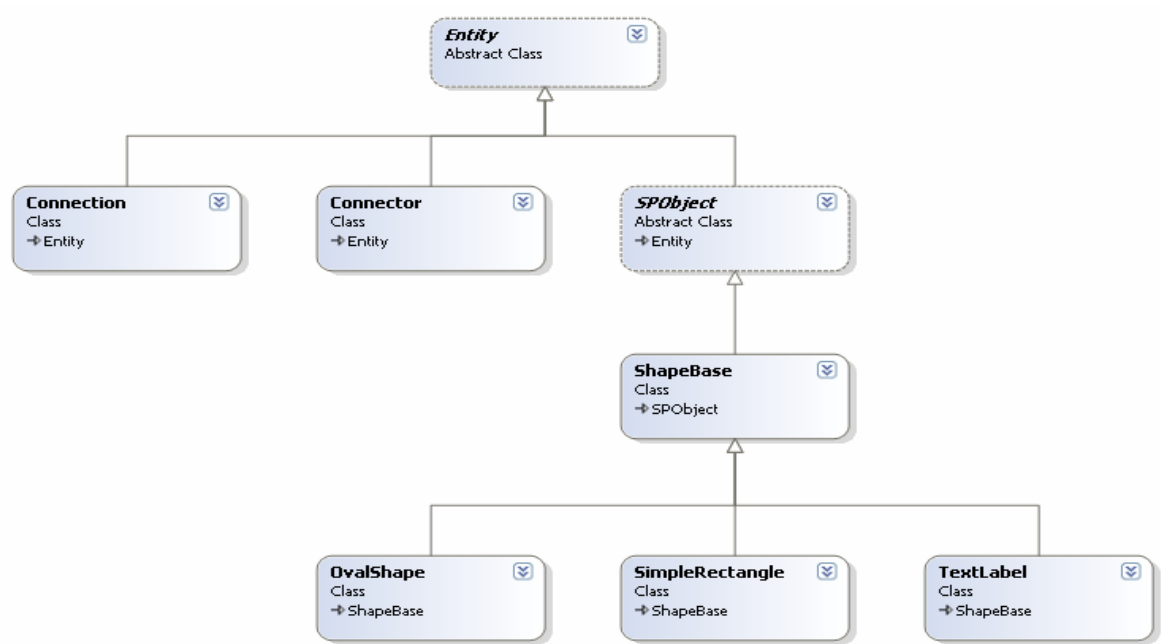


Figure 6 Flowchart Control design components



Figure 7 Portal manipulation and UI components

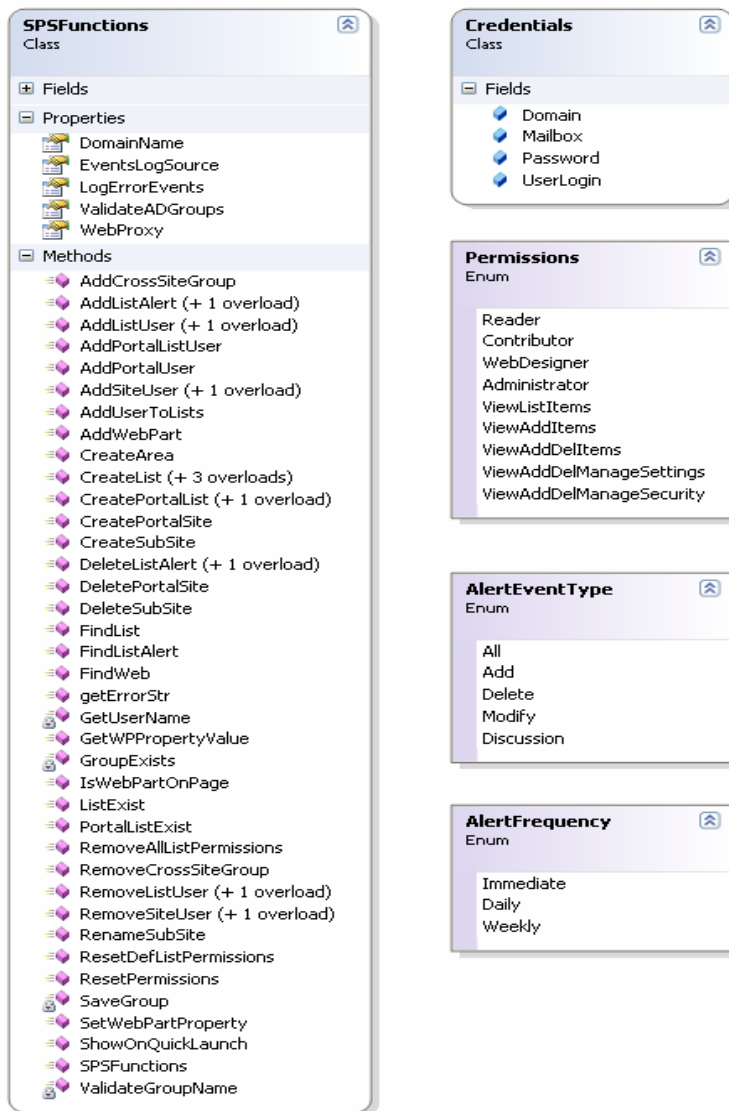


Figure 8 SharePoint manipulation component

6.5. Programming Model – Xml definitions

Being given the complexity of the definitions used on this application, a simplified version of the Portal Definition and Structure will be presented.

Table 7 presents the complete, self explained, basic specification of the portal specification xml. Can be observed that that:

- A **Portal** contains the **Permissions** specifications, **Areas**, **Lists** and **Sites**;
- Each **Area** might contain other sub-areas, **Pages** that contain **WebParts** and each webpart is specified using a set of **Properties**;
- Each **List** is specified by some properties and the **Permissions** for he list;
- Each **Site** is specified by a set of properties and might contain **Pages**, **Lists** and sub-**Sites**. Each **Page** contains **WebParts** and each **WebPart** is defined by a specific set of **Properties**.

Table 7 Xml Schema definition for Portal creation

```
<Portal>
  <!-- Portal Definition -->
  <Title>Portal DEVELOPER</Title>
  <PortalUrl>http://192.168.10.15/</PortalUrl>
  <DefPageName>default.aspx</DefPageName>
  <!-- Portal Content Goes Here... -->
  <Permissions>
    <Permission GUID="410e59ea-11d9-4e32-b9ff-424083c1149d">
      <!-- Example of all possible Permissions for a SharePoint Object -->
      <Reader>
        <User>userName</User>
      </Reader>
      <Contributor>
        <User>userName</User>
      </Contributor>
      <WebDesigner>
        <User>userName</User>
      </WebDesigner>
      <Administrator>
        <User>userName</User>
      </Administrator>
    </Permission>
  </Permissions>
  <Areas>
    <Area GUID="5d95dd88-43e6-43b8-9ec1-d7ecc0cefa7f">
      <Path>areaPathName</Path>
      <Title>areaTitle</Title>
      <Template>SPSTopic</Template>
      <Pages>
        <Page Url="default.aspx">
          <WebParts>
            <WebPart TypeName="myNameSpace.myWebPart"
filterProperty="propertyName" filterValue="filterValue">
              <Properties>
                <Property Name="propertyName">propertyValue</Property>
              </Properties>
            </WebPart>
          </WebParts>
        </Page>
      </Pages>
    </Area>
  </Areas>
</Portal>
```

```

        </Properties>
    </WebPart>
</WebParts>
</Page>
</Pages>
</Area>
</Areas>
<Lists>
  <List ID="listID" GUID="43750e17-6f4d-46a7-a981-99faa9688154">
    <Path>listPathName</Path>
    <Title>listTitle</Title>
    <Template>listTemplateName</Template>
    <QuickLaunch>showInQuickLaunch</QuickLaunch>
    <Permissions>
      <Permission>
        <!-- Example of all possible Permissions for a List -->
        <ViewItems>
          <User>userName</User>
        </ViewItems>
        <ViewAddItems>
          <User>userName</User>
        </ViewAddItems>
        <ViewAddDelItems>
          <User>userName</User>
        </ViewAddDelItems>
        <ViewAddDelManageSettings>
          <User>userName</User>
        </ViewAddDelManageSettings>
        <Administrator>
          <User>userName</User>
        </Administrator>
      </Permission>
    </Permissions>
  </List>
</Lists>
<Sites>
  <Site ID="siteID" GUID="4c65463f-22fd-4007-b626-0e09ea986819">
    <!--A SPS Site is identified by Url = http://FQDN/ManagedPath/Path-->
    <Url>siteUrl</Url>
    <!--Managed Path of the Site used by the Virtual Server-->
    <ManagedPath>Sites</ManagedPath>
    <!--Collection of Paths used to Replicate the Site Structure-->
    <Paths>
      <Path>sitePathName</Path>
    </Paths>
    <Title>siteTitle</Title>
    <Template>STS#1</Template>
    <!--Sharepoint Template-->
    <Pages>
      <Page Url="default.aspx">
        <WebParts>
          <WebPart TypeName="myNameSpace.myWebPart">
            <Properties>
              <Property Name="propertyName">propertyValue</Property>
            </Properties>
          </WebPart>

```

```
        </WebParts>
    </Page>
</Pages>
<Permissions>
    <Permission>
        <!-- Example of all possible Permissions for a Site -->
        <Reader>
            <User>userName</User>
        </Reader>
        <Contributor>
            <User>userName</User>
        </Contributor>
        <WebDesigner>
            <User>userName</User>
        </WebDesigner>
        <Administrator>
            <User>userName</User>
        </Administrator>
    </Permission>
</Permissions>

<Lists>
</Lists>

<Sites>
</Sites>
</Site>
</Sites>
</Portal>
```

7. Functional Description for Visualization Methods

1. 3D Site Map

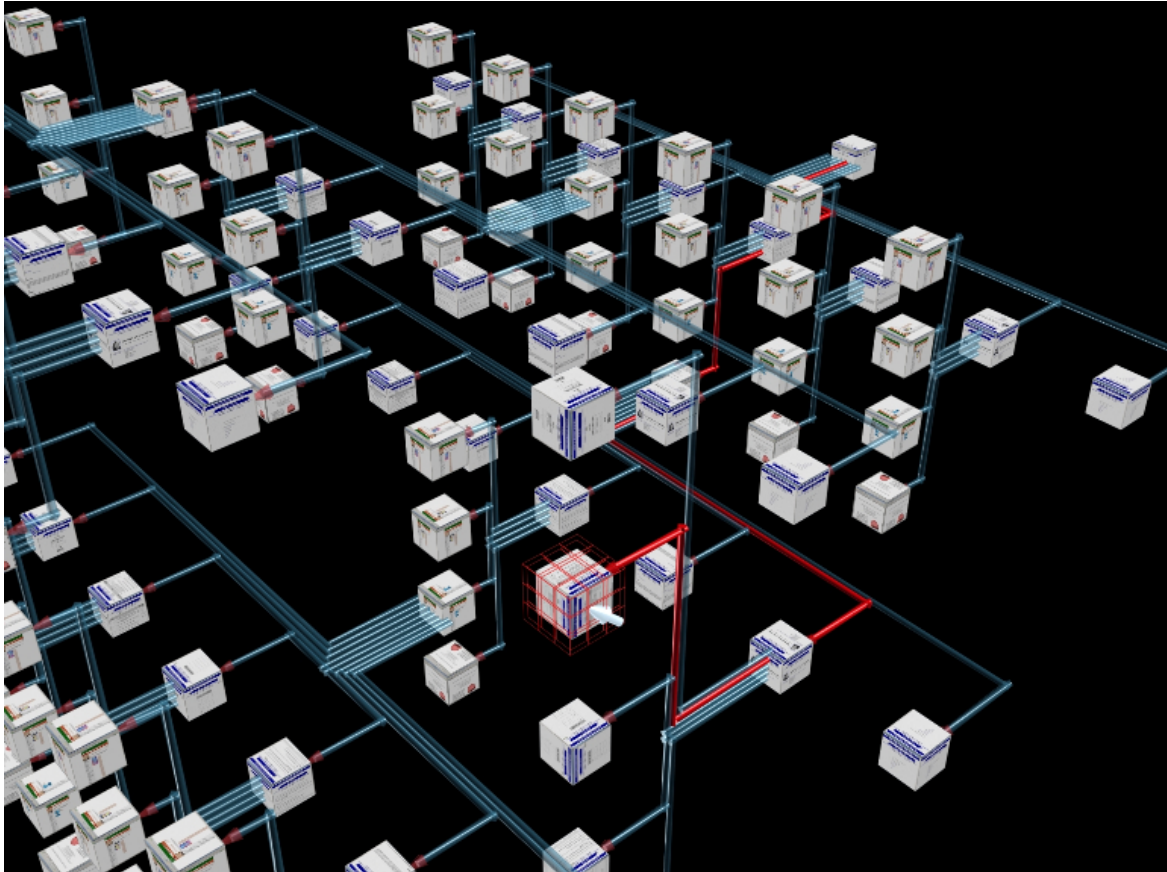


Figure 9 3D Site Map visualization method

Description

The scheme is based on the hierarchical representation, level based, having a grid distribution on levels, as it can be seen in Figure 9. By default, on this scheme, consecutive levels are represented perpendicularly one on the other. Each page is considered a Parent and the links of the page are considered Childs. Dynamic manipulations of the dispositions Parent-Child is possible to apply. In addition, different dispositions and 3D manipulations are possible to apply to each Level. Every page can be (or not) represented many times, as it appears while structuring the site in levels.

Advantages:

- representing large amount of objects without unbalancing the representation;
- easy exploration and manipulation of the representation;
- representation of multiple shapes and interconnections is easier.

Disadvantages:

- the identification of objects might be difficult (3D space);
- selection of objects might not be too visible;

Interface Functionality

- rotation;
- translation/panning;
- zooming;
- single/multiple/level selection;
- highlight levels/selection/areas;
- show/hide statistical information;
- use search for selecting pages/groups/areas;
- highlight the interconnections between selected pages;
- use filters for the represented information.

Scheme Goals

- represent/identify a page/group of pages;
- represent/identify site levels;
- highlight selections;
- show a global representation of the site.

Best Scheme for a specific Goal

- best for representing the site map for large sites;
- best for highlighting the path between selected objects.

Relations with other Schemes

- synchronization with the 2D representations of the site;
- synchronization with the representation of the selection;
- synchronization with the session 2D/3D representations;
- synchronization with the statistical representations.

2. 2D Site Map

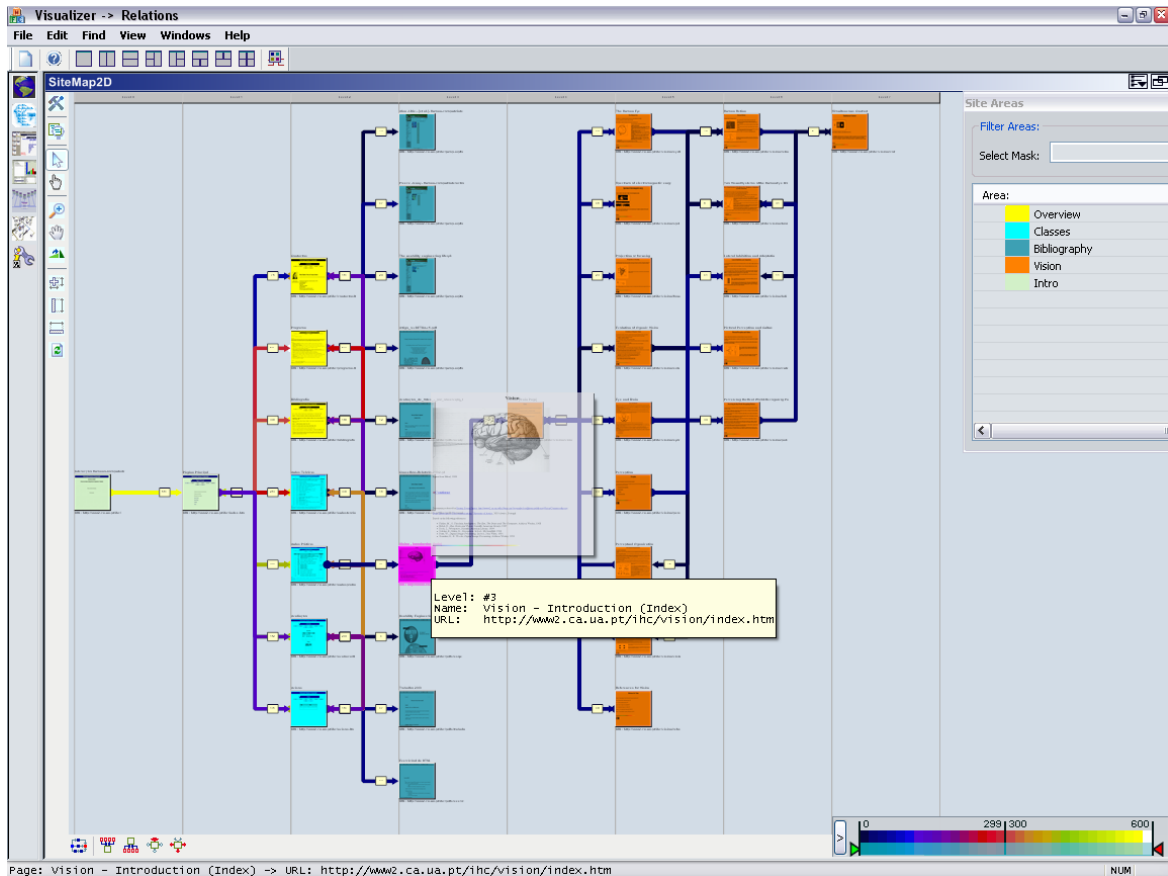


Figure 10 2D Site Map visualization method

Description

The scheme bases on the hierarchical representation. The levels are represented in columns, using a grid. Consecutive levels are represented in different columns, as it can be seen in Figure 10. The links distribution is only in one direction – from the parent page to the child links. A page is not uniquely represented.

Advantages:

- representing the objects in the same geometrical plan, easier to be observed;
- easy exploration and manipulation of levels and the relation page – child links;
- easy identification of objects and interconnections;
- selection of objects is easier and the selection is easier noticeable;

Disadvantages:

- additional data and wrong color schemes might make the representation difficult to be interpreted;
- multiple interconnections between pages might make the representation unbalanced;
- difficulties to represent large amount of data;

Interface Functionality

- translation/panning;
- zooming;
- single/multiple/level selection;
- highlight levels/selection/areas;
- use search for selecting pages/groups/areas;
- highlight the interconnections between selected pages;
- use filters for the represented information.

Scheme Goals

- represent/identify uniquely a page/group of pages;
- represent/identify site levels;
- highlight selections;

Best Scheme for a specific Goal

- best for representing the site map for small/medium sites;
- best to highlight large selections/areas.

Relations with other Schemes

- synchronization with the 3D representation of the site;
- synchronization with the representation of the selection;
- synchronization with the session 2D/3D representations;
- synchronization with the statistical representations.

3. 3D Page Explorer

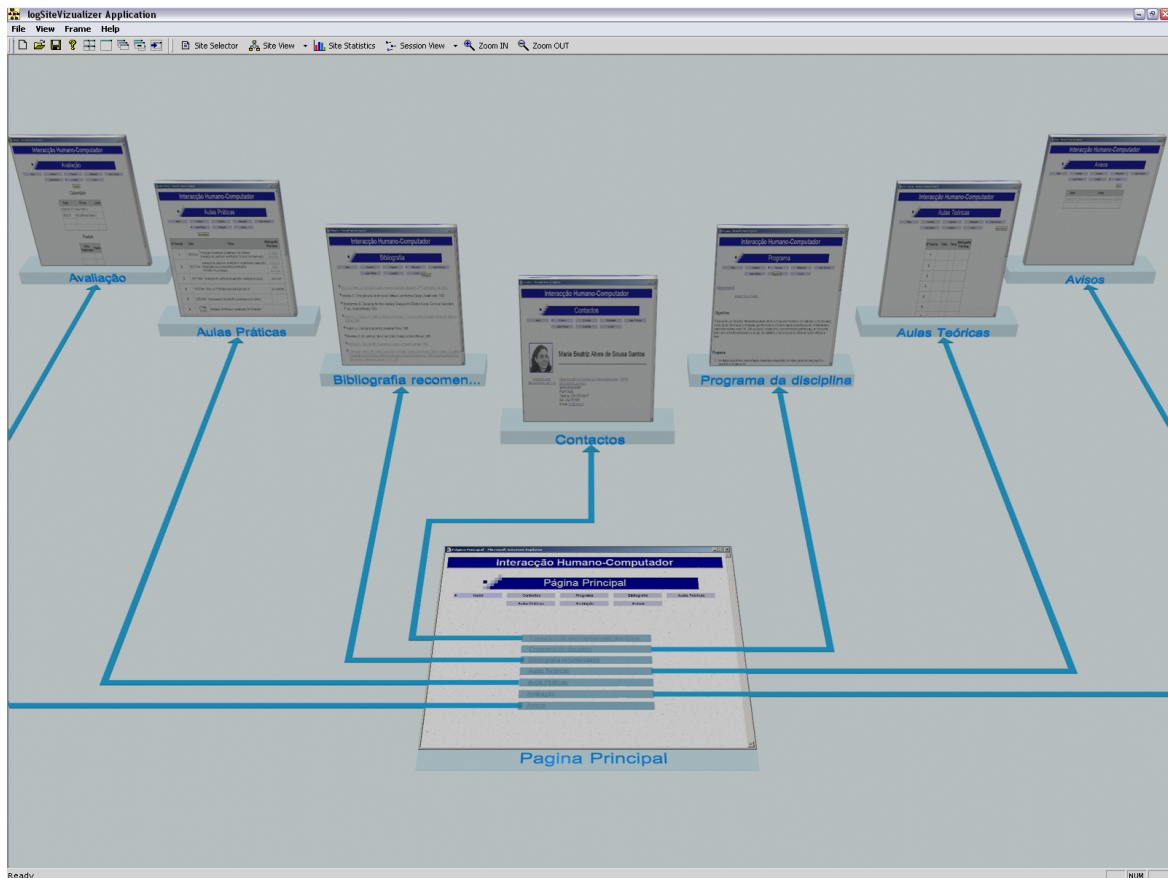


Figure 11 3D Page Explorer (Links + Referrers) visualization method

Description

The scheme bases on the Referrers->Page->PageLinks concept. On this scheme, the Page Links are represented according to the Hot Spot position on the page, as it can be seen in Figure 11. The Page Referrers can be represented in the opposite direction of the PageLinks. The links distribution is in only one direction.

Advantages:

- representing the objects in the 3D space, showing the page links/referrers directly to the physical position on the selected page;
- easy exploration/manipulation/selection of the objects;
- the representation is easier to be percept and interpreted;
- easy to represent the page statistical information;

Disadvantages:

- complex pages with many links and referrers might be difficult to be percept and interpreted;
- multiple interconnections between pages can unbalance the representation;

Interface Functionality

- translation/panning;
- zooming;
- rotation;
- single/multiple selection;
- show/hide statistical information;
- use filters for the represented information.

Scheme Goals

- represent/identify uniquely the list of referrers and page links for the selected page;
- highlight statistical information about the page;

Best Scheme for a specific Goal

- best for representing the page links/referrers;
- best for highlighting page statistical information.

Relations with other Schemes

- synchronization with the 2D/3D representation of the site;
- synchronization with the representation of the selection;
- synchronization with the session 2D/3D representations;
- synchronization with the statistical representations.

4. SessionMAP 2D Timeline/Function

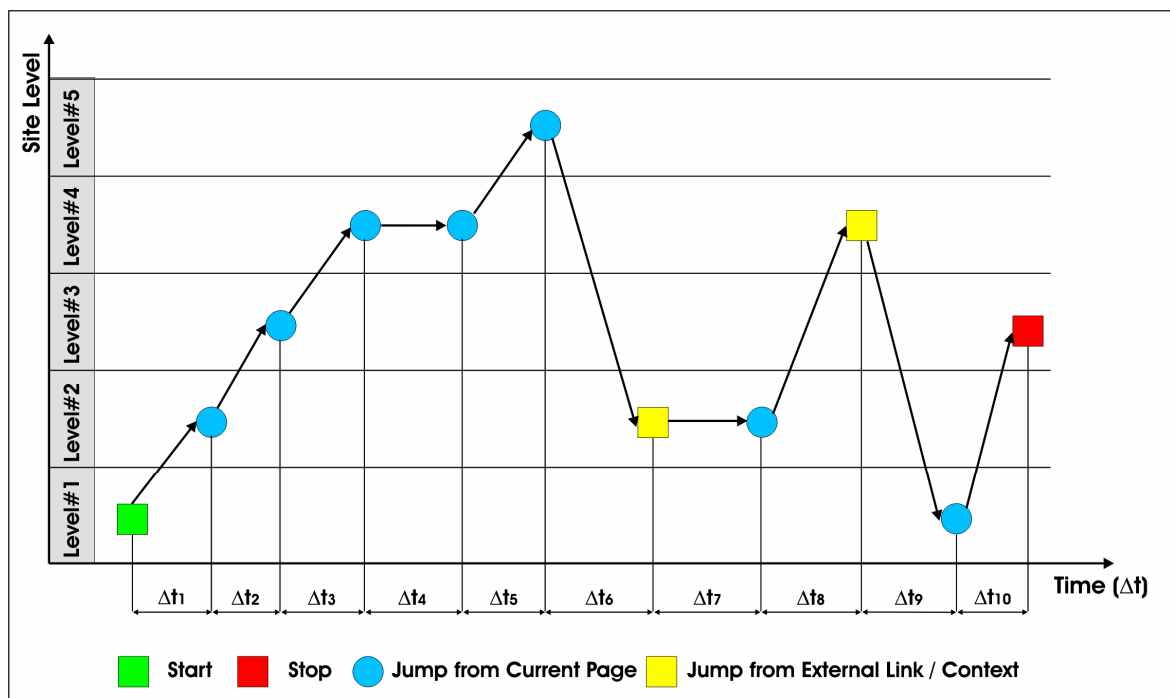


Figure 12 SessionMAP 2D Timeline/Function visualization method

Description

The scheme is based on a Time – Level representation in a Cartesian system. On this visualization method, the visited pages are represented on a horizontal timeline, according to the page level, within the steps followed by the site user, as it can be observed in Figure 12, where is presented the conceptual representation for the scheme. As it is observed here, a user session can be easily transposed into a sinusoidal function, whose fluctuations can be visually analyzed. An automated process of the function fluctuations analysis can reveal important information about the site usage and contextual information disposition on the site (page disposition). Additional information, color or/and shape coded, as contextual/external jump between the pages, also gives good feedback about the site implementation or/and usage.

Advantages:

- easy perception of the real site usage within the selected user session;
- gives good information about the site usage and also about the site interconnections;
- automated analysis process is easy to be implemented while analyzing a sinusoidal function;
- visual detection of the site organization has a good visibility;
- shape coding also helps the perception of the site usage and internal page organization;

Disadvantages:

- multiple session representations in a 2D space needs a very good exploration user interface;

Interface Functionality

- zooming;
- translation/panning;
- highlight selection/pages;
- single/multiple/level selection;
- use search for selecting pages/jumps;
- direct space manipulations of the context;
- possibility to choose the level of detail and shape coding scheme;

Scheme Goals

- represent/identify the session goals;
- represent/identify the site usage patterns;
- represent/identify the site organizational structure;
- highlight the site user behavior while exploring the site;

Best Scheme for a specific Goal

- best for representing the user jumps;
- best for identifying the session goals;
- best for automated analysis processes;
- best for representing statistical usage;
- best for understand the site contextual structure;

Relations with other Schemes

- synchronization with the 2D/3D representation of the site;
- synchronization with the representation of the selection;
- synchronization with the statistical representations.

5. 3D Session Map

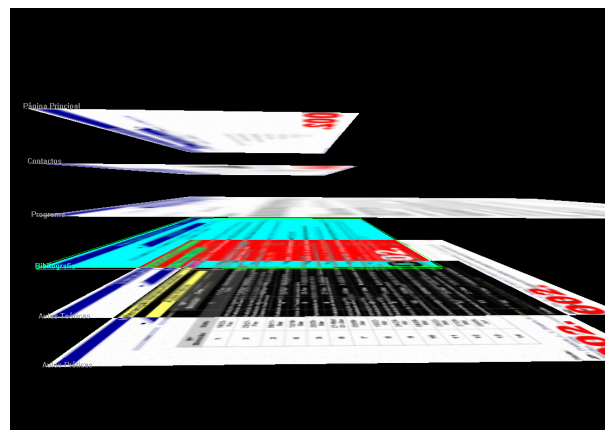
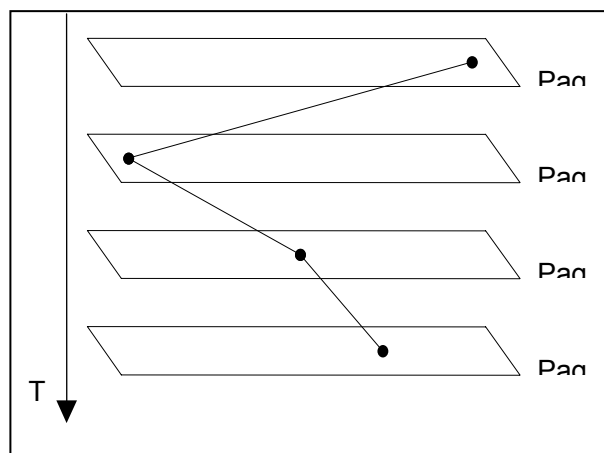


Figure 13 3D Session Map visualization method

Description

The scheme bases on the timeline based representation of the visited pages, as in Figure 13. On this scheme, the pages are represented using transparency. One page can have multiple representations, as the user selected/followed links.

Advantages:

- representing large amount of objects in a 3D space;
- easy exploration and manipulation of the representation;
- representation of multiple shapes and objects is easier, objects as hotspots, statistical information, etc.;
- the possibility to compare multiple sessions simultaneously.

Disadvantages:

- poorly designed scheme might make the identification process (for page objects) difficult;
- selection of objects might be difficult;
- representing many objects can result in the impossibility of identifying objects.

Interface Functionality

- rotation;
- translation/panning;
- zooming;
- single/multiple selection;
- rotate around selection;
- highlight selected links/selection/areas;
- show/hide statistical information;
- use search for selecting pages/groups/objects;
- highlight the selected links between selected pages;
- the possibility of using filters for the represented information

Scheme Goals

- represent/identify a page/group of pages;
- represent/identify session jumps;
- highlight selections;
- show session(s) representation

Best Scheme for a specific Goal

- best for representing multiple sessions simultaneously;
- best for highlighting the user selections

Relations with other Schemes

- synchronization with the 2D/3D representation of the site;
- synchronization with the representation of the selection;
- synchronization with the session 2D representations;
- synchronization with the statistical representations

6. 2D Site/Page Explorer

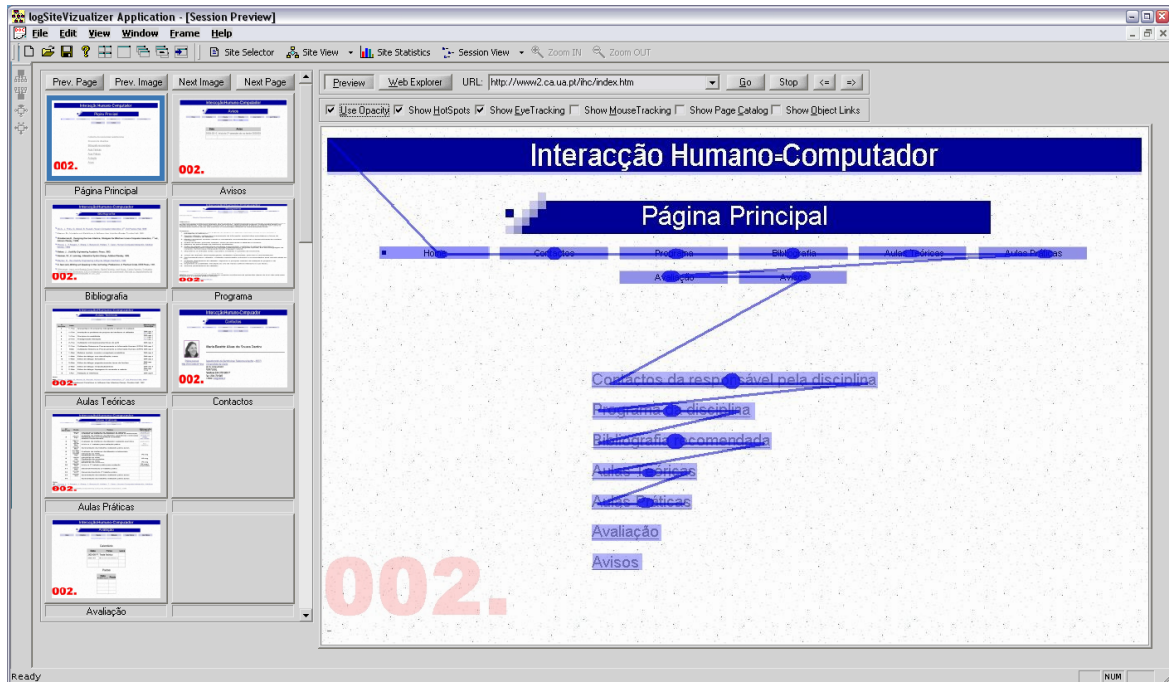


Figure 14 Site/Page Explorer visualization method

Description

The scheme bases on the thumbnail representation of pages screen preview. On this scheme, the pages are uniquely represented. The scheme is presented in Figure 14. The selected thumbnail has a real size representation on the right side of the representation. The possibility of exploring the real page is also given.

Advantages:

- easy to explore the site content by viewing small screen captures of the site pages;
- selection of objects is easier and the selection is easier to be observed;
- easy to represent the additional information for a specific page;

Disadvantages:

- the manipulation of thumbnails might be annoying with large sets of thumbnails;

Interface Functionality

- single/multiple selection;
- highlight selection/areas;
- use search for selecting pages/areas;
- use filters for the represented information;

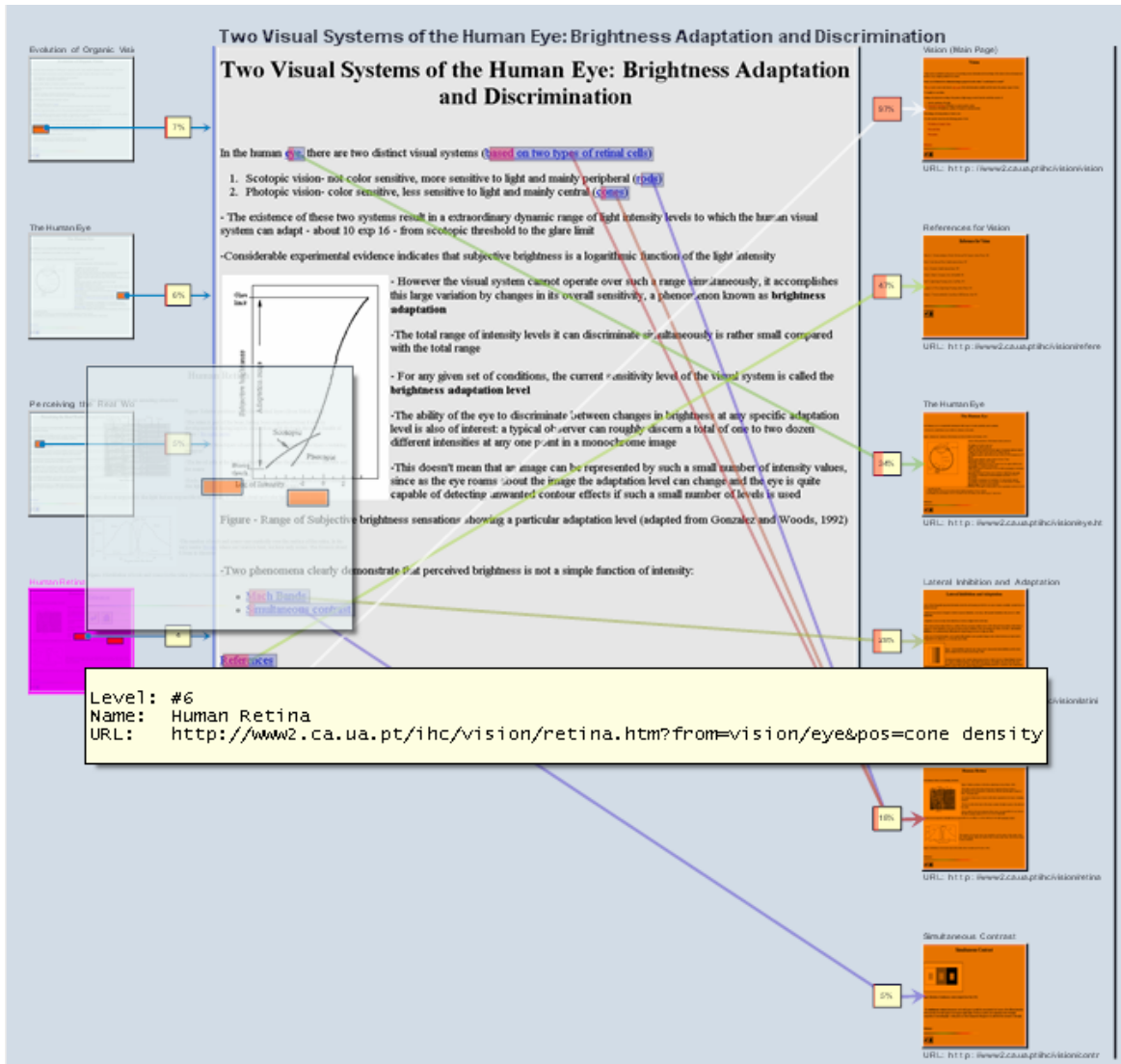


Figure 15 Page Explorer visualization method

Scheme Goals

- represent/identify the pages;
- highlight the real page preview/real page;

Best Scheme for a specific Goal

- best for representing the site/session pages for fast identification;
- best for exploring the real pages/previews;
- best for representing the additional information for pages;

Relations with other Schemes

- synchronization with the 2D/3D representation of the site;
- synchronization with the representation of the selection;
- synchronization with the session 3D representations;

- synchronization with the statistical representations.

7. Possible Interconnections

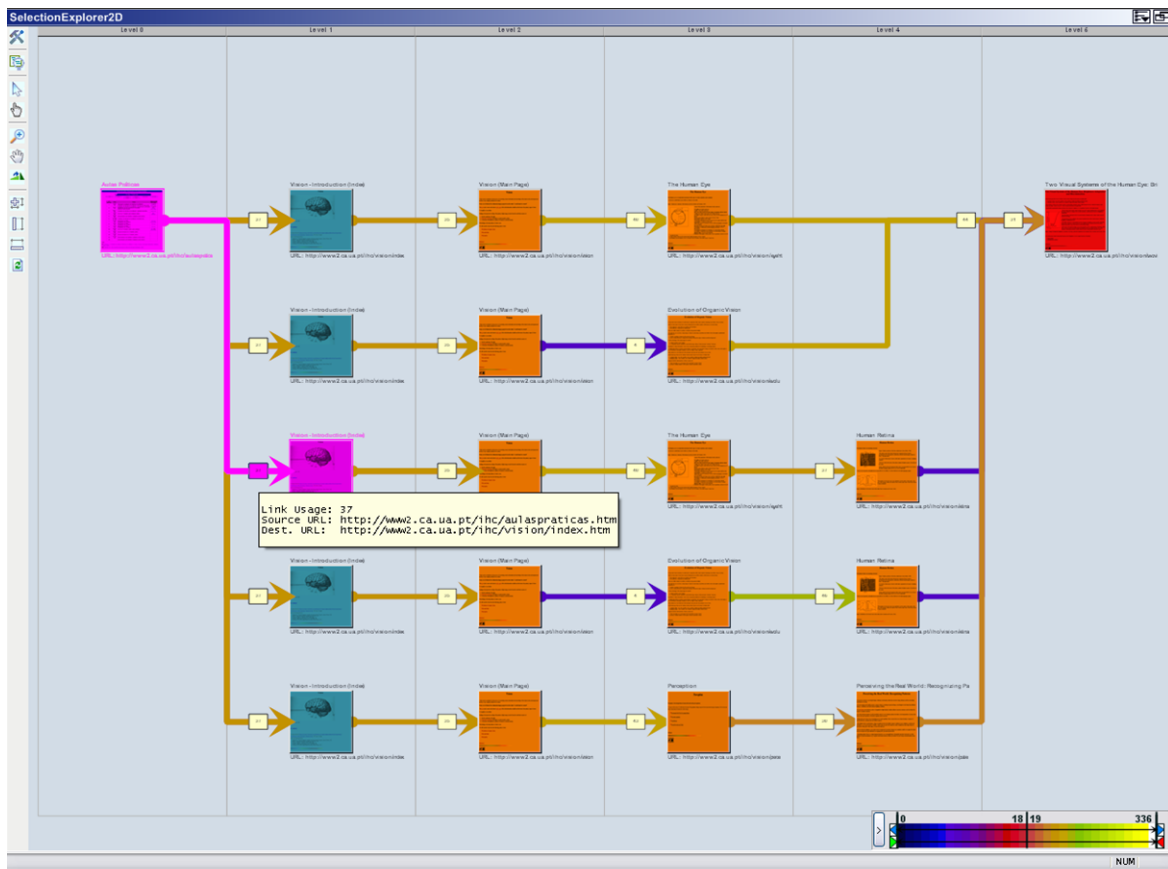


Figure 16 Possible Interconnection visualization method

Description

The scheme bases on the graph representation. It is used for representing the interconnections between the selected pages. The scheme can be observed in Figure 16. The statistical information about the links between pages can be represented using different coding schemes like color, line thickness, etc.

Advantages:

- representing all the interconnections between selected pages;
- using statistical information is easy to identify the behavior of the site users when explored the selected pages;
- additional information like link usage can be easy integrated with the scheme

Disadvantages:

- too many pages on the selection might be difficult to represent because of all the interconnections between pages;

Interface Functionality

- translation/panning;

- zooming;
- single/multiple selection;
- highlight selection/areas;
- show/hide statistical information;
- use search for selecting pages/groups/areas;
- use filters for the represented information.

Scheme Goals

- represent/identify a page/group of pages;
- represent/identify interconnections between pages;
- represent the statistical information of related pages

Best Scheme for a specific Goal

- best for representing the interconnections between selected pages;

Relations with other Schemes

- synchronization with the 2D representations of the site;
- synchronization with the representation of the selection;
- synchronization with the session 2D/3D representations;
- synchronization with the statistical representations.

8. Hotspots Usage Intensity

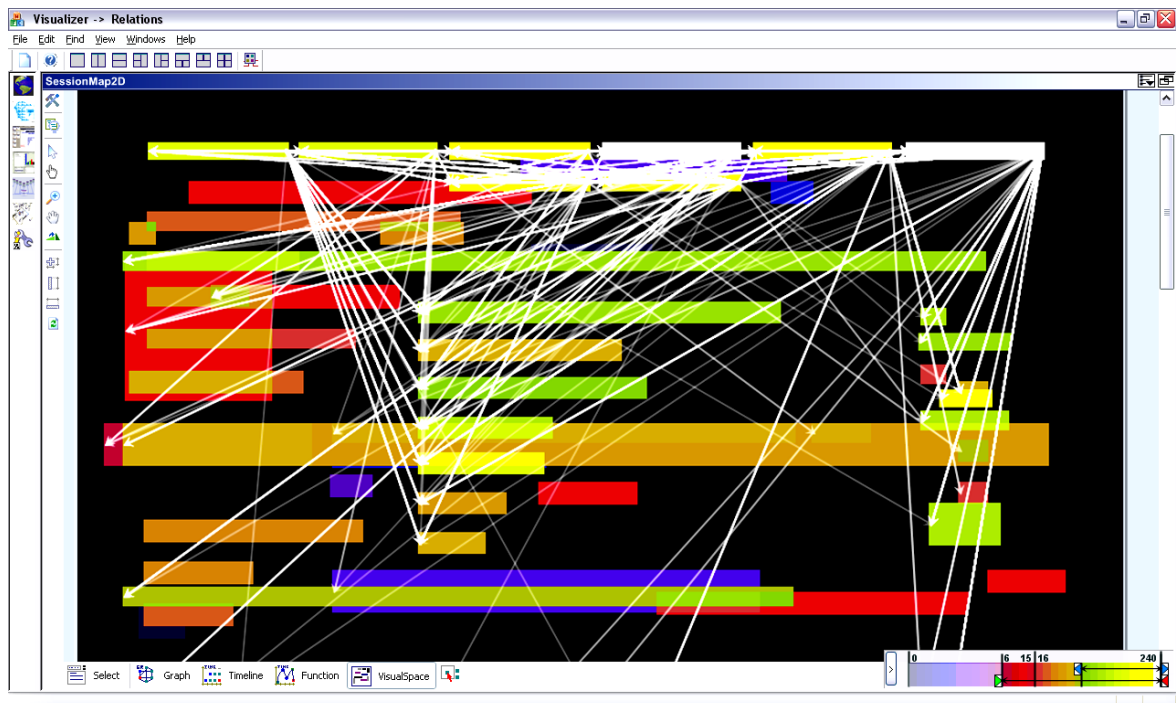


Figure 17 Hotspot Usage Intensity visualization method

Description

The scheme is based on a hotspot usage color coding representation. The scheme is observed in Figure 17. The representation highlights the most visited hotspots by analyzing the selected user sessions. The color of a region gives the usage of the hotspots located in that region. Different color-coding techniques can be used. In addition, a dynamic interaction with each region is added to the scheme.

Advantages:

- the scheme gives good information about the hottest areas of the site;
- the geometrical position of the most followed links as visual clues;
- visual feedback related to the page usage and structure;
- visual detection of the site organization has a good visibility;

Disadvantages:

- dynamic interaction with the scheme has to be present, or to be used complementary with other schemes;

Interface Functionality

- zooming;
- selecting objects;
- manipulating objects;
- color-coding selection and manipulation;

Scheme Goals

- represent/identify the hot areas;
- site pages content as visual clues;
- represent/identify the geometrical site usage;

Best Scheme for a specific Goal

- best for representing the selected links;
- best for identifying the hottest areas of the site;

Relations with other Schemes

- synchronization with the 2D/3D representation of the site;
- complementary scheme for other schemes;

Bibliography

- [Andrews1999] Andrews Keith. 1999. *Visualising Cyberspace: Information Visualisation in the Harmony Internet Browser*, In Readings in Information Visualization: Using Vision to Think, (editors) Stuart K. Card, Jock D. Mackinlay and Ben Shneiderman. San Diego, CA, Morgan Kaufman, pp 493-502.
- [Andrews2002] Andrews Keith. 2002. *Visualization Notes*. IEEE Symposium on Information Visualization (InfoVis).
- [AWStats2005] AWStats Log Analyzer. 2005.
Online at: <http://awstats.sourceforge.net>; last visit: November 2005.
- [Barlow2001] Barlow T., Neville P. *A Comparison of 2D Visualizations of Hierarchies*. IEEE Symposium on Information Visualization, InfoVis01, pp 131-138.
- [Becker1999] Becker R.A., Eick S.G. and Allan R. Wilks. 1999. *Visualizing Network Data*, In Readings in Information Visualization: Using Vision to Think, (editors) Stuart K. Card, Jock D. Mackinlay and Ben Shneiderman. San Diego, CA, Morgan Kaufman, pp 215-230.
- [Bederson2003] Bederson B., Shneiderman B. 2003. *The Craft of Information Visualization, Readings and Reflections*. Morgan Kaufman.
- [Benford1999] Benford Steve, Taylor, I., Brailsford, D., Koleva, B., Craven, M., Fraser, M., Reynard, G., and Greenhalgh, C. 1999. *Three Dimensional Visualization of the World Wide Web*. ACM Comput. Surv. N°31, pp 25.
- [Bieber1997] Bieber M., Vitali F., Ashman H., Balasubramanian V. and Oinas-Kukkonen H. 1997. *Fourth generation hypermedia: some missing links for the World Wide Web*, in International Journal Human-Computer Studies, N°47, pp 31-65.
Online at: <http://ijhcs.open.ac.uk/bieber/bieber.pdf>; last visit: July 2006.
- [Booch1998] Booch G. et al. 1998. *The Unified Modeling Language User Guide*. Reading (MA): Addison-Wesley, New York, USA.

- [Brath1999] Brath R. 1999. *Concept Demonstration Metrics for Effective Information Visualization*. IEEE Symposium on Information Visualization, InfoVis97. pp 108-111.
- [Card1999] Card S. K., Mackinlay J., Shneiderman B. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufman.
- [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). ACM Press, New York, NY, pp 498-505.
- [Ceri2000] Ceri S., P. Fraternali, and A. Bongio. 2000. *Web Modeling Language(WebML): a Modeling Language for Designing Websites*. Proceedings of WWW9 Conference, Amsterdam.
- [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). WebDB '04, vol. 67. ACM Press, New York, NY.
- [Chi1998] Chi Ed H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R., and Card, S. K. 1998. *Visualizing the evolution of web ecologies*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Los Angeles, California, United States, April 18 - 23). Editors: C. Karat, A. Lund, J. Coutaz, and J. Karat, Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, pp 400-407.
- [Chi1999] Chi Ed H. 1999. *A Framework for Information Visualization Spreadsheets*. PhD Thesis.
- [Chi2002] Chi Ed H. 2002. *Improving web usability through visualization*. IEEE Internet Computing 6, 2 (March), pp 64-71.
- Online at: <http://dx.doi.org/10.1109/4236.991445>; last visit: April 2006.

- [ClickTracks2005] ClickTracks Web Analyzer. 2005.
- Online at: <http://www.clicktracks.com/products/analyzer/>; last visit: December 2005.
- [Cockburn1997] Cockburn A, Steve J. 1997. *Design Issues for World Wide Web Navigation Visualization Tools*. Proceedings of RIAO'97: The Fifth Conference on Computer-Assisted Research of Information. McGill University, Montreal, Quebec, Canada, June.
- Online at:
<http://www.cosc.canterbury.ac.nz/andrew.cockburn/papers/riao97.pdf>.
- [Cooley2003] Cooley R. 2003. *The use of web Structures and Content to Identify Subjectively Interesting web Usage Patterns*. ACM Trans. Inter. Tech. 3, 2 (May), pp 93-116.
- [Cugini1999] Cugini J, Scholtz J. 1999. *VISVIP: 3D Visualization of Paths through websites*. In Proceedings of the 10th international Workshop on Database & Expert Systems Applications (September 01 - 03). DEXA. IEEE Computer Society, Washington, DC, pp 259.
- [Dahlback1993] Dahlback N., Jonsson A., and Ahrenberg L. 1993. *Wizard of oz studies - why and how*. In W Gray, W. E. Heey, and D. Murray, editors, Proceedings of the 1993 International Workshop on Intelligent User Interfaces. Association of Computing Machinery, Inc.
- [Deep2005] Deep Log Analyzer. 2005.
- Online at: <http://www.deep-software.com/default.asp>; last visit: December 2005.
- [Di Lucca2002] Di Lucca G. A., A. R. Fasolino, F. Pace, P. Tramontana, U. de Carlini. 2002. *WARE: a tool for the Reverse Engineering of web Applications*. csmr, p. 0241, Sixth European Conference on Software Maintenance and Reengineering.
- [Dix1998] Dix A. et al. 1998. *Human Computer Interaction*, 2nd. Prentice-Hall, London, England.
- [Dodge2003] Dodge M. 2003. *The Atlas of Cyberspace: Maps of websites*.
- Online at: http://www.cybergeography.org/atlas/web_sites.html; last

visit: December 2005.

- [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.
- [Eick2004] Eick G. Stephen. 2004. *Visualizing Online Activity*. Commun. ACM 44, 8 (Aug. 2001), 45-50.
- [Ethnio2005] Ethnio. 2005.
Online at: <http://www.boltpeters.com/ethnio/index.html>; last visit: December 2005.
- [Faraday2000] Faraday P. 2000. *Visually Critiquing web Pages*. Proceedings of HFWeb'00 (Austin, TX, June).
Online at: <http://www.tri.sbc.com/hfweb/faraday/faraday.htm>; last visit: April 2006.
- [FastStats2005] FastStats. 2005.
Online at: <http://www.mach5.com/products/analyzer/index.html>; last visit: December 2005.
- [Fraternali2003] Fraternali P, Matera M., Maurino A. 2003. *Conceptual-level log analysis for the evaluation of web application quality*. First Latin American Web Congress (LA-WEB'03), pp 46.
- [Freitas2002] Freitas C. S., Luzzaerdi P., Cava R., Winckler M., Pimenta M., Nedel L. *Evaluating Usability of Information Visualization Techniques*. Proceedings 5th Symposium on Human Factors in Computer Systems IHC2002, Fortaleza, Ceará.
- [GIS2005] Geographic Information Systems. 2005. U.S. Geographical Survey. Online at: http://erg.usgs.gov/isb/pubs/gis_poster/; last visit: May 2006.
- [Grinstein] G. Grinstein, P. Hoffman, S. Laskowski, R. Pickett. *Benchmark Development for the Evaluation of Visualization for Data Mining*. Online at: <http://home.comcast.net/~patrick.hoffman/VIZ/benchmark.pdf>; last visit: March 2004.

- [Grinstein2002] Grinstein G., Hoffman P. E. and Pickett R. M. Benchmark. 2002. *Development for the Evaluation of Visualization for Data Mining*. In Fayyad, U., Grinstein, G. G., et al. (eds.), *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann, San Francisco, pp 129-176.
- Online at: <http://home.comcast.net/~patrick.hoffman/VIZ/benchmark.pdf>; last visit; June 2006.
- [Google Analytics2005] Google Analytics. 2005.
- Online at: <http://www.google.com/analytics/index.html>; last visit: December 2005.
- [Healey2001] Healey C. G., Amant St. R., and Chang J. 2001. *Assisted Visualization of E-commerce Auction Agent*. Presented at Graphics Interface 2001 – Canadian Human-Computer Communications Society, pp. 201-208, 7-9 June.
- [Heer2002] Heer Jeffrey, Ed H. Chi. 2002. *Separating the Swarm: Categorization Methods For User Sessions On The Web*. Proceedings of CHI2002.
- [Hix1999] Hix D., J. E. Swan II, Gabbard J., McGee M., Durbin J., King T. 1999. *User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment*. In Proceedings IEEE Virtual Reality 99. pp 96-103.
- [Jacobson1999] Jacobson I., Booch G., and Rumbaugh J., 1999., *The Unified Software Development Process*. Addison-Wesley, New York, USA.
- [IFABC2006] IFABC Global Web Standards.
- Online at: <http://www.ifabc.org/standards.htm>; last visit: March 2006.
- [Ivory2002] Ivory M. Y., Hearst M. 2002. Improving website design. *IEEE Internet Computing*, v.6 n.2, p.56-63, March 2002.
- [Ivory2002] Ivory M. Y., Hearst M. 2002. *The State of the Art in Automated Usability Evaluation of User Interfaces*. *ACM Computing Surveys (CSUR)*, v.33 n.4, p.470-516, December.
- [ISAServer2004] Microsoft ISA Server. 2004. Microsoft Internet Security and Acceleration Server.

- Online at:
<http://www.microsoft.com/isaserver/evaluation/overview/default.msp>,
Last visit: December 2005.
- [IWEBTRACK2005] IWEBTRACK. 2005. Online at: <http://www.iwebtrack.com/>; last visit: December 2005.
- [Keim2001] Keim A. D. 2001. *Visual Exploration of Large Data Sets*. Communications of the ACM, vol. 44(8), pp. 38-44.
- [Kobsa2001] Kobsa A. 2001. *An Empirical Comparison of Three Commercial Information Visualization Systems*. IEEE Symposium on Information Visualization, InfoVis01. pp 123-130.
- [Larman1998] Larman C. 1998. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. Prentice-Hall.
- [LiveSTATS2005] LiveSTATS Deepmetrix. 2005.
Online at: <http://www.deepmetrix.com/>; last visit: November 2005.
- [Martin2001] Martin Johannes, Ludger Martin. 2001. *Website Maintenance With Software-Engineering Tools*. 3rd International Workshop on Web Site Evolution (WSE'01), pp 126.
- [Mealha2004] Mealha Ó, Sousa Santos B., Nunes J, Zamfir F. 2004. *Integrated Visualizations for an Information and Communication Web Log Based Management System*. Proceedings of International Conference of Information Visualization – IV04, London. July.
- [MSDN2004] MSDN. February 2004. *Enterprise Development, User Interface Design and Development, Graphics and Multimedia*. Microsoft.
- [Munzner1997] Munzner T. 1997. *H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space*. In Proceedings of the 1997 IEEE Symposium on information Visualization (infovis '97) (October 18 - 25). INFOVIS. IEEE Computer Society, Washington, DC, 2.
- [Najork2001] Najork M, Wiener I. J. 2001. *Breadth-First Search Crawling Yields High-Quality Pages*. In Proceedings of the 10th International World Wide Web Conference. Hong Kong. pp 114-118.
- [Nielsen2001] Nielsen J. 2001. *Did Poor Usability Kill E-Commerce?*. Alertbox, 19

August.

Online at: <http://www.useit.com/alertbox/20010819.html>; last visit: June 2006.

- [Nielsen1993] Nielsen J. 1993. *Usability Engineering*. Academic Press. 1993.
- [Niu2003] Niu Y., Zheng T., Chen Z., Goebel R. 2003. *WebKIV - Visualizing structure and navigation for web mining applications*. IEEE/WIC International Conference on Web Intelligence (WI'03), pp 207.
- [Nomura2002] Nomura S, Oyama S., Hayamizu T., Ishida T. 2002. *Analysis and Improvement of HITS Algorithm for Detecting Web Communities*. Proceedings of the 2002 Symposium on Applications and the Internet (SAINT'02).
- [Nunes2003] Nunes J., Zamfir F., Mealha Ó., Sousa Santos B. 2003. *Web LogVisualizer: A Tool for Communication and Information Management*. Proceedings of the 10th International Conference Human-Computer Interaction – HCI International 2003, Vol. 3 (Human-Centred Computing: Cognitive, Social and Ergonomics Aspects), Crete-Greece. 824-828, June.
- [Nunes2006] José Nunes. 2006. *Visualização de Interação em Cenários de Comunicação Humano-Computador*, Tese de Doutoramento (Versão provisória), Universidade de Aveiro.
- [North2000] North C., Schneiderman B. 200. *Snap-Together: Can Users Construct and Operate Coordinated Views?* Int. Journal Human-Computer Studies, 53, 5. 715-739.
- [OMNIWEB2004] OMNIWEB for Apple, version 5.0, Apple Inc., 2004.
- Online at: <http://www.omnigroup.com/applications/omniweb/>; last visit: July 2006.
- [Opentracker2005] Opentracker. 2005.
- Online at: <http://www.opentracker.net/index.jsp>; last visit: December 2005.
- [Paganelli2002] Paganelli L., Paternò F. 2002. *Intelligent analysis of user interactions with web applications*. In Proceedings of the 7th international Conference on intelligent User interfaces (San Francisco, California,

USA, January 13 - 16, 2002). IUI '02. ACM Press, New York, NY, 111-118.

- [Ricca2000] Ricca F., Tonella P. 2000. *We Site Analysis: Structure and Evolution*. IEEE.
- [Ricca2001] Ricca F., Tonella P. 2001. *Understanding and Restructuring websites with ReWeb*. IEEE MultiMedia 8, 2 April, 40-51.
- [Ruffo2004] Ruffo G., R. Schifanella, M. Sereno. 2004. *WALTy - a user behavior tailored tool for evaluating web application performance*. Network Computing and Applications, Third IEEE International Symposium on (NCA'04), pp. 77-86.
- [Santos2004] Sousa Santos B., Zamfir F., Ferreira C., Mealha Ó., Nunes J. 2004. *Visual Application for the Analysis of Web-Based Information Systems Usage: A Preliminary Usability Evaluation*. Proceedings of International Conference of Information Visualization – IV04, London. July.
- [Sawmill2005] Sawmill. 2005.
Online at: <http://www.sawmill.net/>; last visit: December 2005.
- [Sebrechts1999] Sebrechts M., Vasilakis J., Miller M., Cugini J., Laskowski S. 1999. *Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces*. ACM Conf. Research and Development in Information Retrieval, ACM SIGIR 99, 3-10, California.
- [Shneiderman1996] Shneiderman B. *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. Presented at IEEE Symposium on Visual Languages, Bouldes, Colorado, USA, pp. 336-343, 3-6 September.
- [Schroeder1998] Schroeder W., K. Martin, B. Lorensen. 1998. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice Hall.
- [Sitelogz2005] Sitelogz. 2005.
Online at: <http://www.sitelogz.com/sitelogz/index.php>; last visit: December 2005.
- [Spence2001] Spence R. 2001. *Information Visualization*. Addison-Wesley, 2001.

- [Spiliopoulou2000] Spiliopoulou M. 2000. *Improving the Effectiveness of a website with web Usage Mining*. In Revised Papers From the international Workshop on Web Usage Analysis and User Profiling B. M. Masand and M. Spiliopoulou, Eds. Lecture Notes In Computer Science, vol. 1836. Springer-Verlag, London, 142-162.
- [STATISTICA1999] STATISTICA for Windows, version 5.5, StatSoft Inc., 1999.
Online at: <http://www.statsoft.co.uk/>; last visit: July 2006.
- [Tauscher1997] Tauscher L. and Greenberg S. 1997. *How people revisit web pages: empirical findings and implications for the design of history systems*. Int. J. Human-Computer Studies, 47, pp 97-137.
- [Webtrends2005] Webtrends. 2005.
Online at: <http://www.webtrends.com/Products/WebTrends7.aspx>; last visit: December 2005.
- [Wiss1998] Wiss U., Carr D., Jonsson H. 1998. *Evaluating Three-Dimensional Information Visualization Designs: a case Study of Three Designs*. *Proceedings Information Visualization 98*. IEEE. pp 137-144.
- [Wusage2005] Wusage. 2005.
Online at: <http://www.boutell.com/wusage/>; last visit: December 2005.
- [Youssefi2003] Youssefi Amir H., D.Duke, M.Zaki, and E.Glinert. 2003. *Visual Web Mining*. In *Proceeding of Visual Data Mining at IEEE Intl Conference on Data Mining (ICDM)*, Florida.
- [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.
- [Zaki2001] Zaki M. J. 2001. *Spade: An efficient algorithm for mining frequent sequences*. *Machine Learning Journal*, 42:31–60.
- [Zaki2003] Zaki M. J. 2003. *Efficiently mining trees in a forest*. In *ACM SIGKDD*.
- [Ware2000] Ware Colin, 2000. *Information Visualization: Design for Perception*.

London: Morgan Kaufmann.

[W3C2005]

Logging Control In W3C HTTPD. 2005.

Online at: <http://www.w3.org/Daemon/User/Config/Logging.html>; last visit: November 2005.

[WCTD2006]

Web Characterization Terminology & Definitions Sheet - World Wide Web Consortium. Online at: <http://www.w3.org/1999/05/WCA-terms/>; last visit: March 2006.