Mobile Access to Web Systems Using a Multi-Device Interface Design Approach

Rodrigo de Oliveira
Institute of Computing
Campinas State University (Unicamp)
Campinas, SP – Brazil

Heloísa Vieira da Rocha
Institute of Computing
Campinas State University (Unicamp)
Campinas, SP – Brazil

Abstract - *This paper uses and extends the ideas of a multi-device interface design approach from a previous work developing a framework for mobile access to web systems. It can adapt any desktop designed web system interface to common mobile devices, such as pocket PCs and smartphones. The final interface preserves the original conceptual model to avoid ambiguities on the user's mental model while using several interfaces for the same application alternatively. As a result, it also provides less maintenance cost and complexity for the interfaces generated automatically from the desktop version. We plan to evaluate a prototype for this framework sooner to check if the theory's assumption concerning a better usability over common linear transformations may be perceived.*

Keywords: Adaptive web display, conceptual model, design, interface, mobile devices.

1 Introduction

Design of interactive products has been a challenge for humans since the Antiquity while trying to accomplish even the simplest tasks. The physiological adaptations that made humans more flexible than other primates allowed the development of a wide range of abilities and an incomparable versatility in behavior. This fact gave the specie a unique capability for culture development using a vast diversity of cognitive resources, like perception, memory, learning, reasoning and creative problem solving. However, despite this innovator spirit is generally seen as the main distinct quality among the other animals, sometimes, it also turns to be the designer's curse.

For example, many alternatives have emerged since the early rubbing pieces of flint to produce sparks for fire making. As the media (flint, iron, stones, kindling, matches, electric lighters, etc.) changed, so did the behavior and appearance of all these interactive products, in spite of being the same application purpose. These cookery utensils improvements were well received by everyone, but imagine if they had to deal with all of these medias to make a simple dinner. That would certainly demand an extra effort to meet, understand the purpose and behavior of each one. The person's mental model will build internal constructions about each application conceptual model [12] to enable later manipulations for further previsions and inferences through reasoning. In fact, these applications turns to be the same (to produce fire), but there are as many different system images as the number of medias to accomplish the task.

In a computational domain, such ambiguities are also perceived in a device oriented design. As wireless interfaces for existing desktop applications have been truly encouraged due to the emerging mobile technologies, many approaches arose, suggesting the mobile design should start from the very beginning, focusing on each device's issues to get the most of them. Others try to make dynamic adaptations according to each device [3], [5]. However, recent researches found that many adapted interfaces for the same application lack in usability [6], [8]. The assumptions reside on the fact that, despite linear transformations use appropriately the smaller screen space, they tend to make bigger changes on the interface resulting in confusion while re-finding/comparing information. Although isolated usability tests on these interfaces guarantee the desired goals, they can't do it when all the interfaces are used together. This leads to misconceptions, eventual user's flaws, distrust and, hence, bad usability. The main problem is that the first application conceptual model is forgotten, giving space to a fresh and adapted one, overlooking many of the user's cognitive processes.

This observation has great importance and is the basis for the multi-device design approach proposed by Oliveira & Rocha [11]. The hypothesis defended is that one application shall not demand as many conceptual models as the final media devices to achieve its maximum usability. That is the only way to solve the user's decision making problem, allowing him to make the same and right choices to accomplish any task. To ensure the most important goal of Interaction Design is fulfilled, the approach demands similar usability indexes for the new interface, which can be achieved through usual design techniques, such as evaluations over all the design process.

Attempting to start a validation for this approach, a detailed technical framework was built to dynamically transform web system interfaces (designed for desktops) into usable mobile versions with at least three times less space, like

PDA's and smartphones. A prototype using this framework shall be implemented and a further experiment conducted sooner to evaluate its usability and advantages over linear transformations.

This paper is structured as follows: section 2 revises the multi-device interface design approach; section 3 elaborates a web system interface transformation framework using the design approach; section 4 concludes the paper, suggesting future research directions; and section 5 presents the references.

2 Conceptual Multi-Device Design

The multi-device interface design approach proposed by Oliveira & Rocha [11] states that one application should have the same conceptual model presented on the *n* interfaces available but also ensuring good usability. Here, the term conceptual model is in accordance with the definition given by Preece, Rogers & Sharp [12]: it's *a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended.*

Although some interpretations have a much narrower perspective about it, considering only the class diagram with conceptual classes, attributes and relationships between them [1], [13], the definition used in this work also concerns with the interface behavior (navigational model), look and feel (presentational model) and implementation. Each of these depends on the conceptual model development process, which is mainly iterative. And there is nothing new with this process, involving the choice of the best interaction mode, the suitable interface metaphor, the interaction paradigm in question, the most appropriate interaction styles and the movement from conceptual to physical design. The innovation resides on maintaining this conceptual model while designing other device interfaces, adapting technologies to this user mental model (also called by Winograd [14] as the user's conceptual model) to avoid ambiguities among the interfaces.

According to Oliveira & Rocha [11], this can be accomplished with a lifecycle model that considers the interaction design an endless process beyond the scope of a single product development. It should start with a potential product for some application and run forever iterative even after the final product achievement. When that happens, designers concentrate on prospective user needs ensuring new potential related products will be identified and built according to the same design model. Figure 1 presents this lifecycle model.

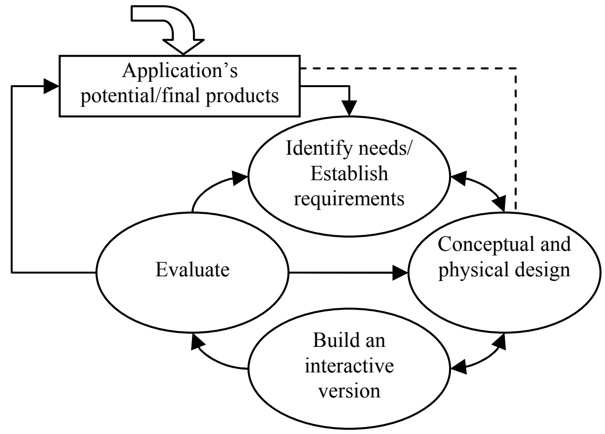

Figure 1 – Conceptual Multi-Device Design lifecycle model [11].

As a result of this interpretation for the lifecycle models, the Don Norman's framework [10] illustrating the relationship between the design of a conceptual model and a user's understanding of it has a better adaptation for the multi-device design. Figure 2 presents a sketch very close to the original model, adding only the application's multi-interface nature.

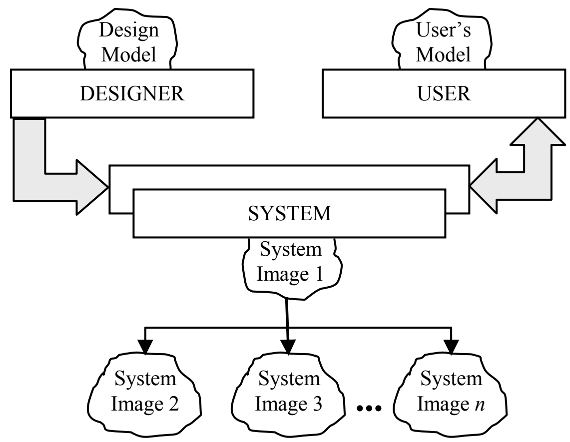

Figure 2 - Interactive components from Norman [10] adapted for multiple interfaces with the same conceptual model. Changing the conceptual model would imply adding *n* models for each interactive component.

3 A Framework for Web System Interface Transformation

Although this proposal scope has a much broader ambition for Interaction Design, the Human Computer Interaction (HCI) field was chosen to validate this approach because of some good solutions proposed recently [6], [8], [9], that can be mixed in this methodology molds to achieve its goals. Also, other issues that can justify the development and implementation of this framework using the proposed approach are:

- Less user cognitive effort when using an application through many devices, whether being a PC, a cell phone, a PDA or any other access media maintaining the bi-dimensional screen features to which this methodology can be applied conserving a good application usability;
- Great complexity and maintenance reduction for the wide range of interfaces, once they are obtained through transformations done like by an adapter. This reduces the user error rate improving both safety and efficiency of use, besides a better memorability while using the same conceptual model for each interface;
- Digital inclusion of a mobile public not able to efficiently use web applications through their portable devices;

Before trying to present the framework itself, first the supportive related work will be addressed for a softly transition between the techniques theory and application.

3.1 Supportive Related Work

There is no doubt the transition demanded from one conceptual model to *n* physical designs can be a hard work. One actual example is the mobile interfaces needed for existent desktop applications. The limited screen size of current handhelds, among other constraints, has been considered a great challenge for HCI. In particular, for the Internet, there's an urgent need for solutions to turn desktop designed web systems accessible in a usable manner for handhelds. According to Mackay, Watters & Duffy [8], these web page transformations can be divided into three categories:

- *Direct Migration* – No transformations are made to the web page. The user generally has to navigate using extensively both horizontal and vertical scrolling. Although the exactly same conceptual model is maintained, the interface design lacks in visibility and efficiency of use;
- *Linear Transformation* – The original web site is changed to a long linear list that fits within the width constraints of the small display. Used by sites like Avantgo (www.avantgo.com) and Usable Net (www.usablenet.com), it usually results on the conceptual model breaking.
- *Overview Transformation* – Provides users with an overview of the original page and, for the most part, content remains the same.

Among these categories, the latter is the only one with the closest works related to this paper proposal. Following are listed three of them:

- *Smartview* [9] – Thumbnail view of the original web page in zoom-out with illegible texts. To overcome this problem, the page is partitioned in logical regions bounded with lines (Figure 3); when selected, content is presented with good visibility inside the screen space (detailed view);

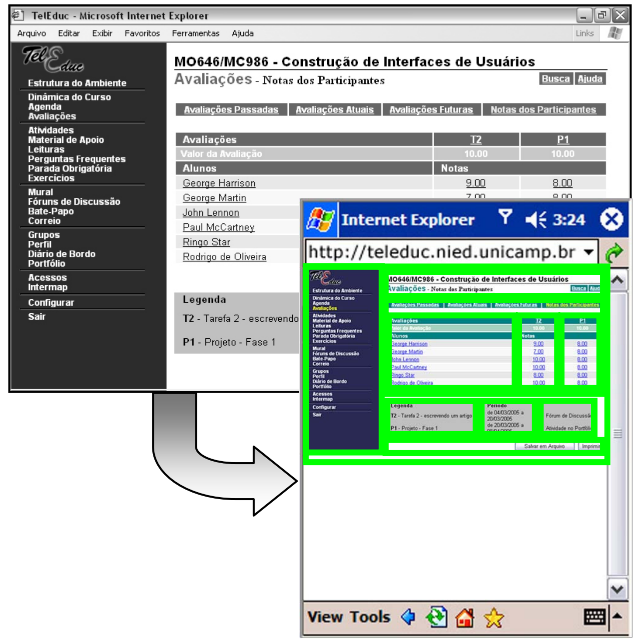

Figure 3 - Example of a Smartview transformation. Although this thumbnail view is completely illegible, when a region is selected, the corresponding detailed view with readable texts is presented adapted to the screen size.

- *Gateway* [8] – Similar to Smartview, but without the region bounds. Also, the detailed view uses a focus-plus-context technique, enlarging the selected region over the detailed view, as shown on Figure 4;

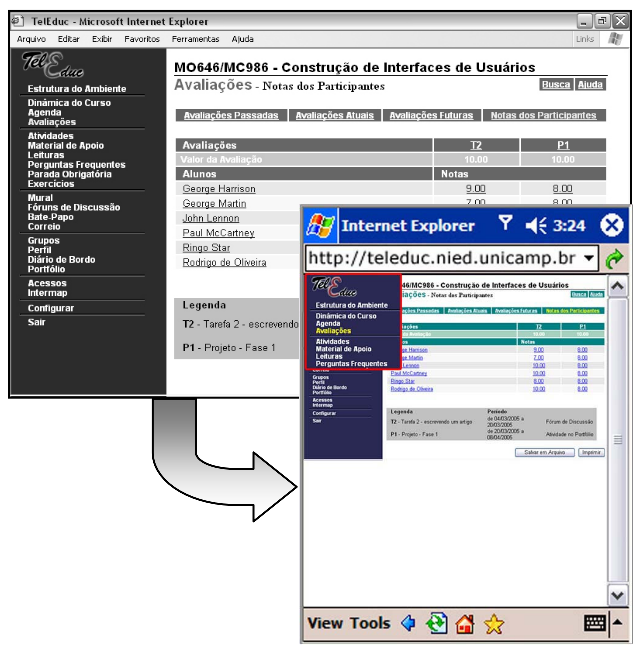

Figure 4 - Example of a Gateway transformation. No region boundaries and page zoom-in through clicks or rollovers.

- *Summary Thumbnail* [6] – Thumbnail with layout preservation and good content visibility using text reduction techniques. However, the detailed view with full text (accessed through one click to a clean area of the page) is a direct migration and has no adaptation to the screen size. Moreover, the summary is language dependent and may get undesirable results, as can be seen on Figure 5.

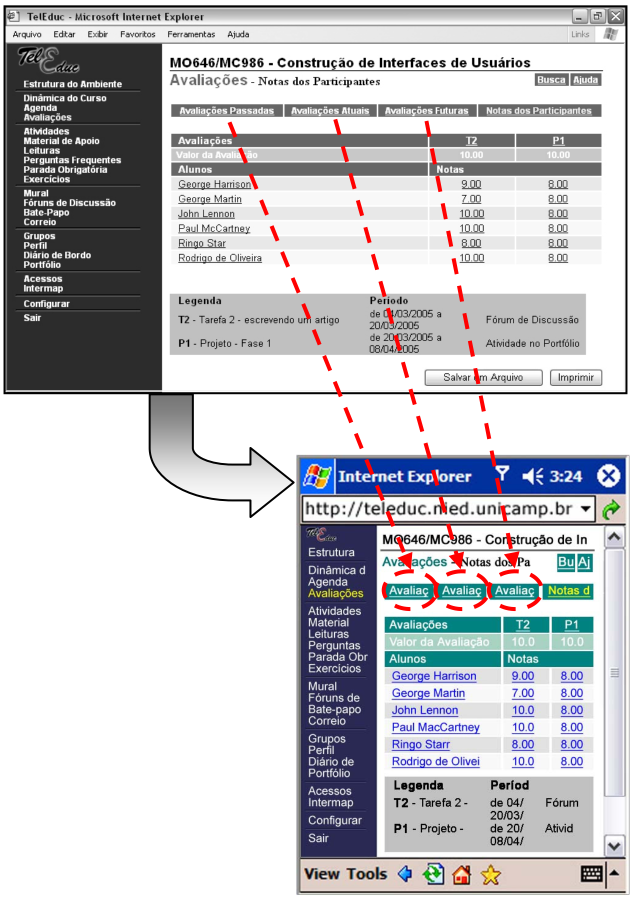

Figure 5 - Example of a Summary Thumbnail transformation. Although it achieves good visibility, the text reduction is for English only and tends to generate ambiguities (in the example, three different links with the same label *Avaliaç*).

Some alternatives for the text reduction problem identified on the latter approach could be the use of both TF/IDF (term frequency / inverse document frequency) and within-sentence clustering techniques as in the approach given by Buyukkokten et al. [2]. The keyword extraction from a body of text relies on an evaluation of each word's importance. A word within a given text is considered most important if it occurs frequently within the text, but infrequently in the larger collection. This collection may be a database containing only web pages from a specific domain (i.e. news, e-learning environments, etc.). Each word in this database has a weight (its importance) for every page calculated using the TF/IDF measure shown by Figure 6.

$$w_{ij} = tf_{ij} \times \log_2 \frac{N}{n} \text{, where}$$

w_{ij} = weight of term T_j in document D_i
tf_{ij} = frequency of term T_j in document D_i
N = number of documents in collection
n = number of documents where T_j occurs at least once

Figure 6 – TF/IDF formula [2].

Unlike it could be supposed, the summarization process isn't based on leaving the heavy (important) words and discarding the light ones. It uses the TF/IDF formula of Figure 6 just to identify relevant clusters (text fragments) that will compose the page's final content after its processing. Here's how it works: given a sentence *S*, all of its relevant words (those with TF/IDF weight above a previously chosen weight cutoff *w*) are marked. Then, *S* is divided into clusters where each of them must: (i) start and end with a significant word and (ii) contain a maximum of *d* insignificant words between two significant (*d* is the distance cutoff variable also previously chosen). Then, each cluster has its weight calculated as the sum of all of its significant terms' weight divided by the total number of terms. Figure 7 shows an example of cluster identification and summarization.

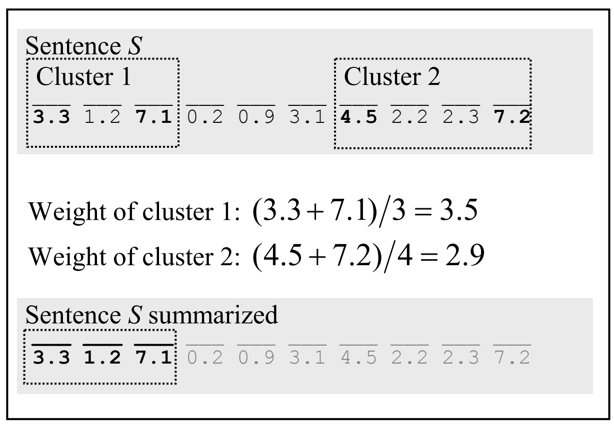

Figure 7 - Example of within-sentence cluster summarization for w = 3.2 and d = 2.

According to the example of Figure 7, the significant words of sentence *S* are those with weights higher than $w = 3.2$ (boldface weights 3.3, 7.1, 4.5 and 7.2). The within-sentence clustering phase identifies two clusters starting and ending with any of these significant words but with no more than $d = 2$ insignificant words between them. Finally, the first cluster is considered the most significant with the greatest weight (3.5 > 2.9) and constitutes the resultant summarized sentence *S*. As this summarization approach was developed for web pages manipulation on handhelds, one of its goals was clearly the efficiency of use for runtime text reduction. That's why such a lexical analysis was the best choice, which still achieved good quality results by generating dynamic text summarizations 70% in accordance with human made ones [2].

However, lexical analyses are prone to some grammatical problems, like synonymy (different words with the same meaning) and polysemy (several meanings for the same word). For example, while analyzing a text with the words "student" and "learner", the last approach would consider them as distinct words, giving each one a different weight according to their TF/IDF relation. A better approach would identify these words have the same meaning and would treat them as the same word, increasing its page occurrences in a shared way and, consequently, its importance for the given page.

To overcome these eventual grammatical problems, the Latent Semantic Analysis (LSA) [4] is a technique used to find similarities between text elements that can also be used for text reduction. With this method, both synonymy and polysemy can be resolved by automatically organizing documents into a semantic structure more appropriate for information retrieval. LSA models term-document relationships using a reduced approximation for the column and row space computed by singular value decomposition of the term by document matrix. In short words, LSA decomposes the term-document matrix into the product of three other matrices, which are used in the similarity calculation through inner product of rows and columns. Anyway, although this approach could lead to more quality results for the summarization process, there is a big charge on efficiency considering the operations performed with these matrices [7].

3.2 Constructing the Framework

In order to correctly apply the design approach given by Oliveira & Rocha [11], its lifecycle model must be followed and continue its iteration process after reaching the final product. Considering this framework addresses any kind of desktop designed web system interface, once identified the need for mobile access, the approaches presented on section 3.1 will be seen as resources to maintain the conceptual model with good usability.

In a first step, the best rival prototype to be studied is the application product itself. And, according to this design methodology, it should be almost literally copied. It doesn't mean we are against the crucial step of generating alternative designs. In fact, the final product already exists and many alternative designs were already considered in its development process. Now there is only the need to provide it multi-device access and the creative leap here is to find appropriate ways to make interface adjustments that won't result in user's mental model misconceptions.

Therefore, among the transformation techniques presented on section 3.1, Summary Thumbnail has the best usability trade-offs, ensuring good visibility and still providing almost the same conceptual model. This last one isn't exactly the same due to:

1. A simpleton summarization approach which, besides the language dependency, uses no domain orientation, resulting in text reduction of anchors and other navigational links that may rip all of its meaning and demand user's recall rather than recognition;
2. The new concepts of "thumbnail view" and "detailed view" along with their access procedures.

To solve the first problem, a summarization process based on semantic analysis like the LSA should be the best option to overcome even grammatical problems like synonymy and polysemy as discussed on section 3.1. However, for efficiency purposes on runtime web page transformations, we recommend lexical analysis like the one given by Buyukkokten et al. [2]. This alternative option should be extended using domain orientation where the database collection will have only documents from a specific domain. This means that, for e-learning applications, the database will have pages of web learning environments, like TelEduc (teleduc.nied.unicamp.br/teleduc), Moodle (moodle.org), Sakai (sakaiproject.org), among others. However, to achieve a more generic purpose for this web system interface transformation framework, the database should comprise other domains, but the text reduction process would have to still be domain oriented. In this sense, different dictionary domain files could be generated remotely, each one containing information of term occurrences in each particular domain, and used by the interface adapter according to the web page being summarized. Also, this summarization should be realized only for navigational links and access structures, as any kind of content reduction is highly questionable, besides the additional processing time waste. In this case, the approach of common news web sites can be used, reducing sentences cropping characters from right to left until it fits on the available space.

In attempting to solve the second problem, both Gateway and Smartview approaches may be combined to accomplish a better detailed view adaptation for the small screen and overcome the direct migration approach given by Summary Thumbnail. Anyway, other solutions should be encouraged to make more explicit the differences between the thumbnail and detailed views, smoothing the transition between them.

For a first iteration on the continued design lifecycle, it is expected that applying Summary Thumbnail approach with these two solutions for the problems mentioned will turn this framework truly adapted for the proposed approach, as every other interface aspect and behavior of any web application will be maintained. Clearly, further iterations will become necessary but, for a first step, Figure 8 presents an interface that should be obtained using this framework to access the e-learning environment TelEduc.

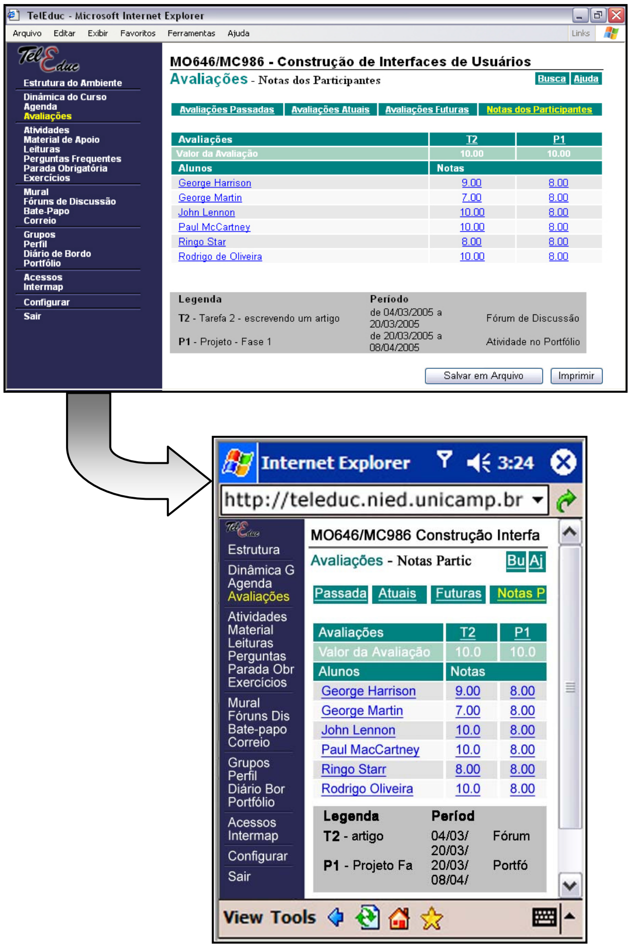

Figure 8 - Example of expected interface transformation using the Conceptual Multi-Device Design approach.

The transformation result obtained on Figure 8 reveals only the interface look and feel, but it is assumed that the transition between thumbnail and detailed views is already implemented in accordance to the solution provided. However, many other implementation issues also affect the prototype usability, like its efficiency, utility and security. Next section discusses a method to apply effectively this framework considering these attributes.

3.3 Applying the Framework

The framework proposed for mobile access to web system interfaces relies upon the Internet client/server architecture. In this paradigm, the server software generally runs on powerful computers to run business applications and the client software requests these services through any device with Internet access. Although generic client/server architectures are two-tier, many actual applications servers store data on a third machine, known as the database server. Figure 9 shows an example sketching this three-tier architecture.

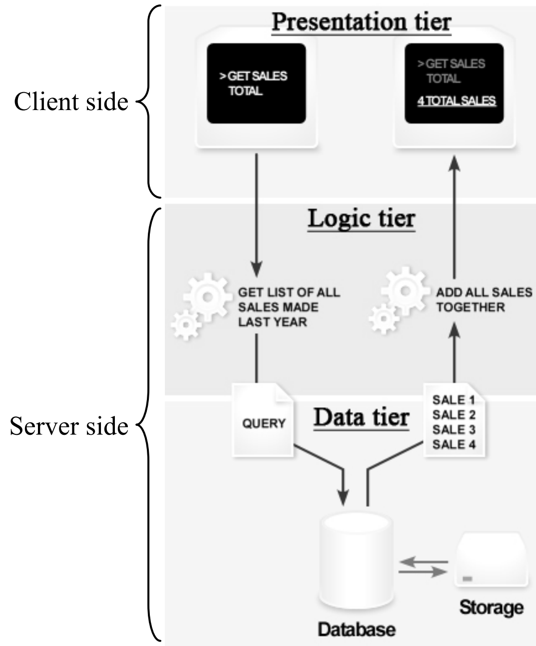

Figure 9 - Example of a sales application under three-tier client-server architecture.

As may be noticed on Figure 9, the usual two-tier client/server architecture is extended on the server side, including the repository of information called database. According to this model, the client requests a service provided by the server through HTTP (Hypertext Transfer Protocol) messages. The server's logic tier reads the message, interprets the request, accesses the data tier to search information (if needed), builds a response HTTP message with the content requested and sends it to the client. Finally, the client receives the desired content on his user agent (i.e. web browser), which is responsible for its interpretation and presentation. Usually, this content is a web page written in a simple script called HTML (Hypertext Markup Language).

Although the processing core of this client/server paradigm is generally attributed to the server side, the complexity transition to the network's edge experienced on the last years has proved this isn't a mandatory rule. In fact, we propose for the web pages transformation framework implementation that this architecture should be improved on the client side. This enhancement of the presentation tier is more suitable to the approach defended on this paper as many usability and technology issues will be fulfilled, like the following:

- *Ease of installation and personalization* – if it was the other way around, every web server should change its logic tier by installing the web system interface adapter. Once it is the mobile user's interest, one simple installation shall enable the whole web access through the handheld. Also, many personal choices must be configured and may be done easily and safely on the client side (i.e. minimum font size, image cropping for slow connections, etc.);

- *Better efficiency* – The network congestion can be decreased avoiding unnecessary further server requests. For example, when the client tries to see a detailed view of a particular region of the thumbnail, the adaptations can be done faster on its side, without having to re-send request messages to the server;
- *Portability* – Cross-platform solutions for many operational systems (i.e. Microsoft Windows, Linux, AIX, Solaris, MacOS, BSD, HP-UX, OpenVMS) and portable devices can be obtained using, for example, the XPCOM open source technology (www.mozilla.org/projects/xpcom). Besides that, there is no need to concern with server side programming languages as the client presentation tier will always deal with the resulting HTML web page delivered by the server.

Technically, this implementation can be made as a plug-in or a browser extension. Despite the actual differences between these browsers, there are many languages and technologies to perform this task. Some could even be used for handheld browsers under the GNU General Public License (GPL), like the Mozilla Minimo Project (www.mozilla.org/projects/minimo).

4 Conclusions

A framework for desktop web system adaptation to mobile devices was built in accordance to the directions provided by our previous work towards a multi-device interface design approach [11]. The framework uses page transformation and summarization techniques to maintain the original interface's conceptual model, supporting much less cost maintenance and complexity for the mobile interfaces. It also points to a better digital inclusion of mobile users with less cognitive effort while using applications through many devices. Further usability evaluations are still under progress and it is expected that such validation can certainly help to diffuse many web applications around the mobile public.

4 References

[1] Andrade, A. R.; Munson, E. V.; Pimentel, M. G. C. Engineering web applications with xml and xslt. In *Proceedings of* LA-Web 2004, IEEE CS Press, Ribeirão Preto, São Paulo, pp. 86-93.

[2] Buyukkokten, O.; Kaljuvee, O; Garcia-Molina, H.; Paepcke, A.; Winograd, T. Efficient web browsing on handheld devices using page and form summarization. In *ACM Transactions on Information Systems* (TOIS). New York, USA, v. 20, 2002. pp. 82-115.

[3] Coninx, K.; Luyten, K.; Vandervelpen, C.; Bergh, J. V. D.; Creemers, B. Dygimes: dynamically generating interfaces for mobile computing devices and embedded systems. In *Proceedings of* Mobile HCI 2003, Udine, Italy, 2003, pp. 256-270.

[4] Furnas, G. W.; Deerwester, S.; Dumais, S. T.; Landauer T. K., Harshman, R. A.; Streeter, L. A.; Lochbaum, K. E. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of* 11th International Conference on Research & Development in Information Retrieval, 1988, pp. 465-480.

[5] Hinz, M. Fiala, Z. Wehner, F. Personalization-based optimization of web interfaces for mobile devices. In *Proceedings of* Mobile HCI 2004, Glasgow, Scotland, pp. 204-215.

[6] Lam, H.; Baudisch, P. Summary thumbnails: readable overviews for small screen web browsers. In *Proceedings of* CHI 2005, Portland, OR, pp. 681-690.

[7] Macedo, A. A.; Pimentel, M. G. C.; Camacho-Guerrero, J. A. Latent semantic linking over homogeneous repositories. In *Proceedings of* the 2001 ACM Symposium on Document Engineering, Atlanta, USA.

[8] Mackay, B.; Watters, C.; Duffy, J. Web page transformation when switching devices. In *Proceedings of* Mobile HCI 2004, Glasgow, Scotland, pp. 228-239.

[9] Milic-Frayling, N.; Sommerer, R. *SmartView: enhanced document viewer for mobile devices.* MSR Technical Report, 2002.

[10] Norman, D. *The design of everyday things.* Basic Books, New York, 1988.

[11] Oliveira, R.; Rocha, H. V. Towards an approach for multi-device interface design. In *Proceedings of* Webmedia 2005, Poços de Caldas, Brazil, 2005. R. P. Fortes, Ed. WebMedia '05, vol. 125. ACM Press, New York, NY, 1-3.

[12] Preece, J.; Rogers, Y.; Sharp, H. *Interaction design: beyond human-computer interaction.* Wiley & Sons, 2002.

[13] Rossi, G.; Schwabe, D.; Lyardet, F. Web application models are more than conceptual models. In *Proceedings of* ER'99, Springer, Paris, France, 1999, pp. 239-252.

[14] Winograd, T. *Bringing design to software.* ACM Press, New York, 1996.