Facilita: Reading Assistance for Low-literacy Readers

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodrigues de Uzêda,
Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, Sandra Maria Aluísio
Institute of Mathematical and Computer Sciences, University of São Paulo
Avenida Trabalhador São Carlense, 400
CEP: 13560-970
São Carlos, SP, Brazil
{watinha at, arnaldoc at, vruzeda at grad., renata at, taspardo at, sandra at}icmc.usp.br

ABSTRACT

Texts are the media content primarily available on Web sites and applications. However, this heavy use of texts creates an accessibility barrier to those who cannot read fluently in their mother tongue due to both text length and linguistic complexity. To offer an accessible alternative to these readers, shorter and simplified versions of text content should be provided. Taking that into consideration, this paper introduces *Facilita*, an assistive technology to help lower-literacy users to understand the text content of Web applications. *Facilita* generates an accessible content from Web pages automatically, using summarization and simplification techniques. It is also important to consider interface design requirements, since *Facilita's* target audience (the functionally illiterate) is often classified as computer illiterate as well. Thus, interaction and user interface design were developed considering the limitations and skills of the functionally illiterate.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: User-centered design, Natural language, Evaluation/methodology; H.5.4 [**H.Hypertext/Hypermedia**]: User issues

General Terms

Design, Human Factors, Languages

Keywords

Web accessibility, Usability, Textual simplification, Summarization, User centered design

1. INTRODUCTION

Problems related to school effectiveness still persist in Brazil; among them are school dropout, grade repetition, as well as the average quality of education provided in Brazilian schools [13]. In this country, a large part of the population faces difficulties in activities involving reading and comprehension depending on text size and linguistic complexity; therefore, access to media that use texts as their primary way to convey information is limited.

In this context, the PorSimples (Text Simplification for Digital Inclusion and Accessibility[1]) project [1], considering the requirements of Web application accessibility, aims at developing technologies to make access to information easier for low-literacy individuals, by means of Automatic Summarization [16], Lexical Simplification, Syntactic Simplification [24], and Text Elaboration [29] (more details in Section 5.1). These resources are applied **automatically** to texts, making their language simpler and more readily understood by those with reading difficulty.

This work is one of the objectives of PorSimples and has the purpose of introducing the interface features, development, and design of ***Facilita***, a reading assistance application to facilitate access to text content in Brazilian Portuguese on Web pages and applications. This application acts as an assistive technology that allows users to read text contents on the Web, applying primarily text summarization and syntactic simplification operations. These two operations applied on a text generate contents that are also textual, but that can be more easily understood by users at the rudimentary and basic levels of literacy (detailed in Section 2).

It is also important to emphasize the interface design requirements, considering that the target audience for this application (low-literacy individuals) is often digitally illiterate as well. Thus, it is necessary to develop an interface that take its limitations into account, implementing accessibility recommendations and also design principles for low-literacy users, so that they can access the functionalities of the application.

In the scope of PorSimples, it is also included the development of *Facilita Educativo (Educational Facilita)*, which, besides performing Automatic Summarization and Syntactic Simplification operations, will carry out Text Elaboration operations that consist in explaining information mentioned in the text. *Facilita Educativo* is supposed to act as a technology to assist learning and instructional activities, but is not focused in this paper.

This article is organized as follows: the next section (Section 2) describes the features that define the target audience for the application; section 3 explains how the Web Accessibility Initiative (WAI) approaches users' reading difficulty on the Web context; section 4 describes principles to design

[1]http://caravelas.icmc.usp.br/wiki/index.php/Principal

SIGDOC'09, October 5–7, 2009, Bloomington, Indiana, USA.

applications for low-literacy users; section 5 details the requirements of *Facilita*, together with the proposed reading assistance; section 6 discusses the interface design methodology adopted in the project; and finally, section 7 contains final remarks and future work.

2. TARGET AUDIENCE LITERACY LEVEL

The Brazilian Institute of Geography and Statistics (IBGE) reports that, in 2007, 10% of the population aged 15 and over was illiterate [13]. When this concept is expanded to include the functional illiteracy rate, defined by UNESCO as the percentage of the population aged 15 and over with less than four years of schooling, illiteracy in Brazil reaches 21.7%.

To improve this type of classification, considering not only the quantitative aspects, but also the qualitative aspects of education in Brazil, the Instituto Paulo Montenegro[2] and the Ação Educativa[3] have developed a Functional Illiteracy Index (Indicador de Analfabetismo Funcional – INAF). The INAF measures literacy levels in the Brazilian adult population annually [21], interviewing and applying cognitive tests to 2,000 people representative of Brazilian citizens between 15 and 64 years old, residing in urban and rural areas in all regions of the country.

As a result of this procedure, people are classified according to their literacy skills, as follows:

- **Illiterate**: individuals who cannot perform simple tasks such as reading words and phrases, though part of them can read familiar numbers (telephone numbers, prices).
- **Rudimentary**: individuals who can find explicit information in short and familiar texts (such as an advertisement or a short letter).
- **Basic**: individuals at this level can be considered functionally literate, since they can already read and understand texts of average length, and find information even when it is necessary to make some inference.
- **Advanced**: individuals whose skills do not limit the understanding and interpreting of the usual elements of a literate society: they can read longer texts, relating their parts, comparing and interpreting information, distinguish fact from opinion, make inferences and synthesize.

According to the INAF [21], in Brazil in 2007, 7% of the individuals were classified as illiterate; 25%, as literate at the rudimentary level; 40%, as literate at the basic level; and 28%, as literate at the advanced level. Individuals classified up to the rudimentary level are considered functionally illiterate, whereas those at the basic and advanced levels are functionally literate.

It is important to observe that both the literate at the rudimentary level and those at the basic level, who total 65% of the population, face difficulties in activities involving reading and comprehension depending on text length and complexity; therefore, their access to textual media is limited. These individuals – the literate at the rudimentary and basic levels – are the target audience for *Facilita*, which has the purpose of providing text versions of a narrower range and less complex linguistic structure, suitable to the reading skills of these users.

[2]http://www.ipm.org.br/

[3]http://www.acaoeducativa.org.br/portal/index.php

3. THE WAI APPROACH

The Web Content Accessibility Guidelines (WCAG), the pre-eminent reference in Web accessibility [7, 14] , make recommendations for developers on applications for literate users at the rudimentary and basic levels with reading difficulty.

One of the WCAG 1.0 guidelines addresses language simplicity and clarity (guideline 14) [26]. This guideline is connected to the principle of application comprehensibility and suggests the use of techniques that focus on writing style (making content easier to read) and on equivalent multimedia (charts, sign language, and audio versions of the content).

WCAG 2.0, which reached the state recommended by W3C in December 2008, includes in its principle of comprehensibility an accessibility requirement related to the level of writing in Web applications [27]. This requirement (*success criterion* 3.1.5) states that texts that demand reading skills more advanced than that of individuals with lower secondary education (as a reference to the British education system – fifth to ninth grades in Brazil) should offer an alternative version of the same content or a supplementary content intended for individuals without these skills. For instance: a text summary with simplified content; illustrations, pictures, and symbols to facilitate understanding; audio versions of text, among others.

However, it is important to point out that these accessibility recommendations (guideline 14 of WCAG 1.0 and success criterion 3.1.5 of WCAG 2.0) describe features related to the content of a Web application. To meet such recommendations, it is necessary to implement the requirements manually in the application content. That is, to guarantee access to information for individuals with reading difficulties (literate at the rudimentary and basic levels), the content authors should consider the use of these resources (simplified content, illustrations, pictures, symbols, audio versions, among others) in a Web application. However, in most cases, those responsible for content preparation are not responsible for developing a Web site or application. Content authors hire Web developers to implement their Web sites or, as in the Web 2.0, users themselves disseminate information about Web applications, often without any knowledge about accessibility guidelines on the Web [8].

Another aspect to be highlighted concerns the fact that low-literacy users are often classified as computer illiterate as well, therefore requiring simplified interactions with the application, given their difficulties and little familiarity with computer systems. The mere availability of resources such as illustrations, pictures, symbols, or audio versions might be not enough to guarantee application accessibility to users at the rudimentary and basic levels of literacy. It is necessary to implement interfaces with simplified user interaction, suitable to these individuals' level of computer knowledge to facilitate access to an application.

4. RELATED WORK

Many studies have reported the development of interfaces for illiterate or functionally illiterate users (outside the Web

context). Theses studies, to ensure that low-literacy users understand the content and functionalities of an application interface, employ different means of communication to replace or complement an interface text content [28]. Among alternative means to provide an accessible content, these stand out: use of a voice interface for data input and output [23]; simplification of text content [3]; graphic representation of information and content [20]; numbers and graphic symbols [22]; and linearization of navigation structures [22].

It is important to emphasize that, although many of these applications prove efficient for their target audience, none of them generates content automatically (applications with dynamic content), and they are not even implemented using a Web architecture.

Contexts of automatic content adaptation and availability on the Web imply technological restrictions, according to some of the principles described earlier. For instance: availability of graphic representations [5] and use of numerical symbols [11] [22]. Considering the context of automatic content generation, solutions to implement the automatic retrieval of images or other types of representation that are really significant for text understanding are highly complex.

Another technological restriction relates to the use of media alternative to text, such as audio versions of text content. Although audio versions of content may be provided by Applets for static contents, the recognition, narration, or synthesis of dynamic contents is still limited on the Web platform. Technologies to develop multimodal applications, such as XHTML+Voice [2], are already addressing this problem, but their use is restricted to some browsers, and XHTML+Voice, specifically, is available only for English. The "Internet Browser with Speech Recognition and Synthesis"[4] is a project that aims at developing and improving an Internet browser with speech recognition and synthesis for Brazilian Portuguese, which will operate as a plug-in in Microsoft Internet Explorer; however, this project is still in its initial stages and continues until 2010.

5. THE FACILITA PROJECT

Facilita is a Web application that operates as an assistive technology to reading text contents available on Web pages. This reading assistance is provided by **automatically** adapting texts into shorter contents of simpler linguistic structure, so that they are more adequate to the reading skills of users at the rudimentary and basic levels of literacy. Facilitated Content is the name given to the result of the automatic text adaptation performed by *Facilita*.

Facilitated Content is also a text, however, its structure is simpler, according to what is defined in the PorSimples project, based on *Plain Language*[5], on simplification systems for the English language, such as, for example [24], and on the analysis of a corpus of texts rewritten to reach a wider audience on the Brazilian Web [1]; as to text length, it will be reduced, in view of low-literacy users' difficulty to read long texts [18].

Reading assistance is provided by the application that runs Automatic Summarization and Syntactic Simplification methods in texts from Web sites and applications. Thus, users have access to an alternative, more easily understandable version of texts available on the Web. This assistance is outlined in Figure 1.

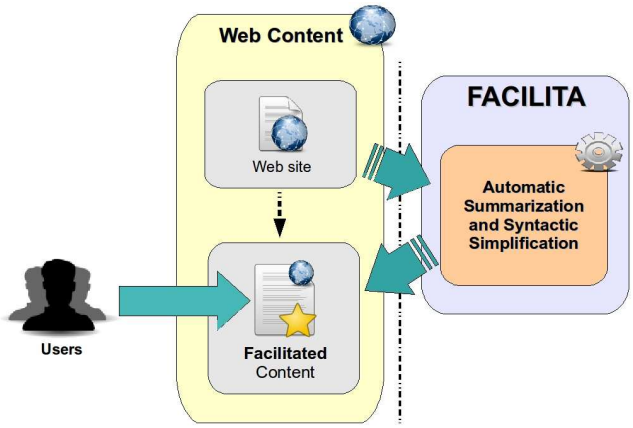

Figure 1: Outline of *Facilita's* reading assistance

Table 1: Text Adaptation techniques

Original Sentence
The senators rejected the proposal from the Information Technology Division of the Senate to change all computers in the plenary room, alleging that public opinion would not receive the disbursement willingly.
Syntactic Simplification (Text Simplification)
The senators alleged that public opinion would not receive the disbursement willingly. Then, the senators rejected the proposal from the Information Technology Division of the Senate to change all computers in the plenary room.
Lexical Simplification (Text Simplification)
The senators rejected the proposal from the Information Technology Division of the Senate to change all computers in the plenary room, alleging that public opinion would not receive the **expense** willingly.
Text Elaboration
The senators rejected the proposal from the Information Technology Division of the Senate to change all computers in the plenary room, alleging that public opinion would not receive the disbursement, or **expense**, willingly.

5.1 Facilitated Content

Young [29] mentions two different techniques for text adaptation: Text Simplification and Text Elaboration (Text Elaboration). The first can be defined as any task that reduces the lexical or syntactic complexity of a text, while trying to preserve meaning and information [15, 19]. Text Simplification can be subdivided into Syntactic Simplification, Lexical Simplification, Automatic Summarization, and other techniques. As to Text Elaboration, it aims at clarifying and explaining information and making connections explicit in a text, for example, providing definitions or synonyms for words known to only a few speakers of a language. Table 1 contains examples of Syntactic Simplification, Lexical Simplification and Text Elaboration (Automatic Summarization was not included due to space constraints).

Text Simplification enhances text readability, that is, makes

[4]http://www.bv.fapesp.br/projetos-pipe/637/navegador-internet-reconhecimento-sintese-fala/

[5]http://www.bv.fapesp.br/projetos-pipe/637/navegador-internet-reconhecimento-sintese-fala/

it easier to read. In the Syntactic Simplification approach to remove syntactic complexity, it is usual to divide long sentences and rewrite complex sentences using simpler syntactic structures, which reduce sentence length, but increase text length due to the repetition of the subject in the sentences that were divided. In Summarization-based Text Simplification, text length is reduced, and relevant information can be lost, which may hinder text comprehensibility. As to Text Elaboration, text comprehensibility can be enhanced, that is, it helps to increase easiness to understand concepts in a text. This technique always increases text length, because it inserts information and repetition to reinforce understanding and make explicit the connections between the parts of a text.

In experiments carried out during the development of the PorSimples project, reading long texts was tiring and difficult for many of the target readers, especially for those at the rudimentary level of literacy. In view of that, we first focused on Text Simplification, since this technique makes reading easier and less tiring when compared to Text Elaboration. Currently, the Text Simplification approaches used in *Facilita* involve reducing clause complexity and text summarization. Studies on Text Simplification at the lexical level are also under way. Text Elaboration is already being studied, and will consist of methods that use named entity recognition to provide simplified definitions for entities, semantic role labeling for answering WH-questions, and automatic discourse analysis for making connections explicit.

PorSimples focuses mainly on the literate at the rudimentary and basic levels. However, other groups of users can benefit from it, such as children and adults who are learning to read and write, people with cognitive disabilities that affect reading skills (dyslexia, aphasia, among others), and second-language learners. Syntactic Simplification can also be useful in systems for information extraction or information retrieval, since it makes clauses more concise and allows for making explicit the subject of subordinate and coordinate clauses when they are divided, as subject repeats in each sentence. Also, passive voice removal favors the development of search expressions.

The next sections detail the two types of Text Simplification operations used in PorSimples (Automatic Summarization and Syntactic Simplification) to generate the Facilitated Content. To the best of our knowledge there is no syntactic simplification system for the Portuguese Language.

5.2 Summarization

The summarizer developed in PorSimples is based on keyword extraction, particularly on the method introduced by [25]. The summarizer is extractive, that is, the summary generated is composed of clauses entirely taken from the original text, without alterations. Keyword extraction is the oldest summarization strategy and one of the most widely used. In this strategy, a set of existing or automatically generated keywords is used to choose candidate clauses for the summary. These clauses must contain at least one keyword.

Two extraction methods were tested: EPC-P (Keyword Extraction by Pattern) and EPC-R (Keyword Extraction by Stem Frequency). In the first, keywords are based on patterns as <Noun> or <Noun+Preposition+Noun> (in this pattern, adjectives may occur in any position). The second uses the stemming technique, finds the most frequent stems, and use them to choose the candidate clauses for the summary. There are also two variations of EPC-P and EPC-R: EPC-P2 and EPC-R2, respectively. These variations produce a ranking of clauses based on the number of keywords in each of them, and create a summary based on this ranking.

After evaluating these four extraction methods, we opted for using the EPC-R in PorSimples, since it produced better results [17]. Other summarizers were also included in the evaluation, which analyzed their precision (choice of good sentences for the summary) and their ROUGE measurement (a measurement that compares the summary generated with a manual summary written by an expert in summarization). In spite of the fact that there are summarizers that perform better than EPC-R for Brazilian Portuguese, we observed that they are not easily available or are time-consuming. After considering this fact, EPC-R proved to be the most suitable summarizer to do the task.

We carried out user evaluations in the same study. A total of 19 users at different literacy levels evaluated the summaries generated. Among readers with more than two years and less than five years of schooling, 66% considered easier to understand a summary than the original text. For readers with more than five years and less than eight years of schooling, we observed that the summary was less tiring to read, and that the levels of text comprehension and summary comprehension were equivalent. We have defined a compression rate of 70% for readers at the rudimentary level of literacy and 50% for reader at the basic level of literacy.

Table 2: Alternative simplifications for the same sentence

Original sentence
This causes more warm wind to blow from the North and increases temperature even more – evaluates Flávio Varone, of the Eighth Meteorological District.
Apposition as a separate sentence
This causes more warm wind to blow from the North and increases temperature even more – evaluates Flávio Varone. Flávio Varone is of the Eighth Meteorological District.
Additive coordinate clause as a separate sentence
This causes more warm wind to blow from the North. This increases temperature even more – evaluates Flávio Varone, of the Eighth Meteorological District.
Apposition and coordinate clause as separate sentences
This causes more warm wind to blow from the North. This increases temperature even more – evaluates Flávio Varone. Flávio Varone is of the Eighth Meteorological District.

5.3 Syntactic Simplification

Table 2 shows alternative simplifications for a given original clause. *Facilita* can generate all these simplifications. Currently, the chosen clause is that with the highest number of simplification operations, adequate for readers at the rudimentary level.

The approach based on syntactic complexity removal was inspired by the work of [24], and consists in a set of rules

for syntactic rewriting applied to the output of the parser Palavras [4] for a given text. The parser output is an XML representation of the text, which includes information on its syntactic tree. The simplification rules deal with seven syntactic phenomena [9]: apposition, passive voice, coordinate clauses, subordinate clauses, absolutive adverbial clauses, relative clauses, position of adverbials (under study), basic word order (Subject-Verb-Object – SVO). A different treatment is given to each subtype of coordinate, subordinate and relative clauses. For example, additive coordinate clauses are divided, whereas adversative coordinate clauses are divided and have their discourse markers changed.

In all, seven operations are used to treat these syntactic phenomena. More than one operation may apply for a given phenomenon and a given clause may have more than one phenomenon. These operations are:

- **Sentence splitting**: consists in finding a point where to split the original sentence (such as conjunctions, relative clause limits, or apposition limits). Two new clauses are generated with their respective subject adjusted accordingly.
- **Change of discourse markers**: consists in replacing discourse markers by more usual markers. For example, "but" replaces "however".
- **Change for the active voice**: in this operation, the order of elements in a passive voice clause is altered and the verb is adjusted accordingly. If there are sentences (coordinate, subordinate, and others) linked to the object of the operation in the passive voice, they are moved to the subject of the clause in the active voice.
- **Inversion of clause order**: this operation was primarily designed to handle subordinate clauses. Graesser et al. [12] propose, in general, to move the main clause to the beginning of the sentence, in order to help the reader processing it on their working memory. In our project, each of the subordination cases has a more appropriate order for main and subordinate clauses in order to keep a logical order of the expressed ideas.
- **Subject-Verb-Object order (SVO)**: consists in adjusting the elements of the clause, so that the resulting order is SVO. For example, the clause "It ended in a tie, the match between Brazil and Bolivia" is converted to "The match between Brazil and Bolivia ended in a tie".
- **Topicalization and de-topicalization**: consists in changing the position of adverbials, moving them to the beginning (topicalization) or to the end of the clause (de-topicalization).

There is also an operation called "Do not simplify", a special case reserved for sentences that do not suffer any simplification. Two simplification modes are predicted within the scope of PorSimples: natural Syntactic Simplification [10] (not included in *Facilita* yet) and strong Syntactic Simplification (already included in *Facilita*). The first was designed for users at the basic level of literacy, and involves a lower degree of simplification. Machine learning techniques are being studied to determine which operations will be implemented in a given clause. The second simplification mode focuses on users at the rudimentary level of literacy, who require a higher level of simplification. All phenomena will be treated in all clauses, whenever possible.

6. DESIGN APPROACH

Usually, it is difficult to specify all requirements for iterative systems at the beginning of a software development life cycle. The only way to define certain features of an application design is to implement them and test them with users [6]. As far as the development of interfaces for low-literacy users is concerned, this fact is extremely important, since a low level of literacy is often associated with computer illiteracy. Thus, considering the wide cultural gap between user and developer, it is difficult to construct a conceptual model that users accept based on non-experimental design approaches [28].

Taking that into account, in this work, we used the iterative design methodology: a design process in which we try to overcome problems inherent to incomplete requirement specification by means of evaluating several designs that improve the final product incrementally at each iteration [6].

Next, we discuss the interactions of the iterative design methodology already developed in the project.

6.1 First iteration

In the first iteration, a throwaway approach to prototypes was used to determine how the initial interaction to activate the resource available at *Facilita* would be conducted. In this stage, we held meetings with the PorSimples project team to present mockups that represent alternatives to activate *Facilita* and to define which interaction model would be used. The mockups developed are described below:

- **Text editor**: in this interaction model, the user would be asked to insert a text content in a text field of *Facilita*, which would then show a summarized and simplified version of the text content inserted in the application. We should point out that this interface requires that texts be selected from different Web sites using the copy and paste process, and requires that the user uses at least two execution instances of the browser: one for the Web site or application that is being accessed, and other for *Facilita*.
- **Search system**: works similarly to the Web application Google Translate[6]. A URL (Uniform Resource Locator) must be inserted to indicate the Web site or application that has the text content to be simplified for reading. Then a version of the same Web site indicated by that URL would be displayed, with all its text content summarized and simplified. This mockup imposes a limit for use that is linked to its interface: the computer illiterate would hardly be familiar with the concept of Web page addresses (URLs), and it would be difficult for them to attribute URLs to the search field of the interface.
- **Insertion of *Facilita* into Web sites**: *Facilita* would be a functionality inserted into the Web sites that intend to make their content accessible to low-literacy individuals. In these Web sites, there would be a link to *Facilita* and, after clicking on that link, an adapted

[6]http://www.translate.google.com/

version of the text content of the Web site would be presented. This content could be sent to *Facilita* as a parameter, or even the whole text content of the Web site could be summarized and simplified by *Facilita*, as shown in the previous mockup. However, Web developers or Web designers would have to implement or insert code excerpts into their Web sites and applications, so that their content could be sent to and adapted by *Facilita*. Thus, the use of facilitation operations would be limited to those Web sites that had implemented this functionality.

- **Browser plug-in**: we also developed a mockup that facilitates the content of a Web site or application by means of a browser plug-in. A button or link would be added to the browser, which would provide a facilitated version of the text content contained in the Web site being accessed.

Among the mockups presented, the **Browser Plug-in** was identified as one of the best approaches, since it is a resource integrated into the browser, constantly visible, always in the same position on the interface, independent of which Web site is being accessed. Given that this interaction alternative is a resource implemented directly into the browser, it can be applied to any Web site, differently from the mockup **Insertion of *Facilita* into Web sites**.

6.2 Second iteration

As soon as the way to activate the facilitation resource was determined, we started the second development cycle for *Facilita*. Differently from the first iteration, from the second stage on we used the evolutionary prototyping approach, in which the prototype is not thrown away anymore, but evolves from usability evaluations that point out the problems to be addressed in the prototype.

Prototypes differ in the amount of functionality and performance they provide in the final product [6]. Considering that, as from this iteration, prototypes were not going to be thrown away anymore, their development was carried out implementing actual functionalities and, because of that, spending more time on planning and coding.

Our decision to implement functional prototypes from this stage was also made considering the need to validate experimentally the application under development. The prototype of an interactive system is used to test its requirements through an evaluation carried out by actual users. A reliable performance of the final system's requirements can only be established if evaluation conditions are similar to those expected during actual operation [6].

Thus, in this stage, we developed a functional prototype of *Facilita* that uses Summarization with 70% of compression and Strong Simplification, both detailed in Section 5. Summarization with 50% of compression and Natural Simplification are to be included in the next prototypes.

In order that a prototype performs reading assistance, the following steps should be taken during an interaction between user and application:

- The user accesses a given Web site or application.
- The user selects the text content available in the Web site for which he/she needs assistance.

Figure 2: Browser's favorites bar with a link to *Facilita*

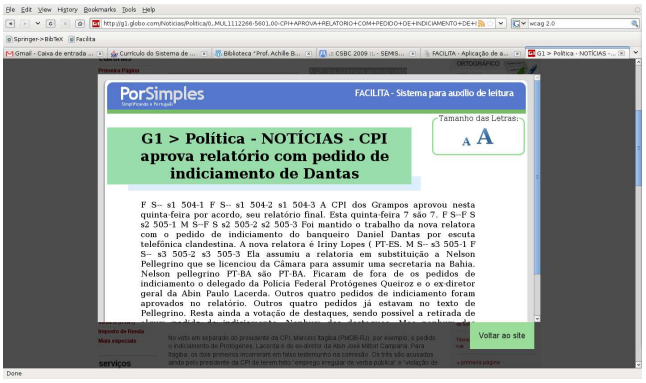

Figure 3: Display of the Facilitated Content in the first prototype

- The user clicks on a link to *Facilita* available in the browser's favorites bar (shown in Figure 2), and then the Facilitated Content will be displayed (shown in Figure 3).

This approach to user interaction is quite similar to that used in Google's Translate[7], in which the user also has to select texts available on Web sites and click on a link embedded in the browser's favorites bar. However, Google's Translate redirects the Web site that is being accessed to the application Translate, whereas *Facilita* provides the content generated on the Web site itself by means of a popup interface component.

It is important to emphasize that the popup interface component was used as a way to retain context information related to the Web site or application on which content facilitation was applied. At the same time that the Facilitated Content is displayed inside the popup element, on the background users can see the Web site or application that was being accessed previously. This popup component was implemented using HTML DIV elements, without creating a new window, since increasing the number of windows displayed by the operational system could confuse users. In the popup's bottom right corner, there is a link that closes it and takes the user back to the Web site that was being accessed.

At the top of the popup, there are the PorSimples logo and *Facilita's* name ("FACILITA – Reading assistance system") as a way to identify the function of the interface component. Another resource used to retain context information is to capture the title of the Web site being accessed at the moment when *Facilita* is activated. This title is then displayed inside the popup as a header for the Facilitated Content (HTML header element).

[7]http://translate.google.com/translate_tools

Figure 4: Display of Facilitated Content in the second prototype

We observed that Summarization with 70% of compression and Strong Text Simplification require some seconds to be completed, depending on the length of the text selected from Web sites. During the processing time necessary for Strong Simplification, a "Wait..." message is shown to the user. We also added a mechanism to increase the font size of the Facilitated Content.

A limiting factor in this way of activating is related to the selection of the text content for which one needs reading assistance. As emphasized before, *Facilita* must have the most simplified interaction possible with users. Therefore, selecting a text with a mouse may not be adequate to users with the level of computer skills expected from the target audience for this application. However, extracting text content automatically from Web pages is a complex task, and demands a longer implementation effort. Thus, we determined that text selection should be kept in the first user evaluation cycles until an alternative to extract text content is defined and implemented.

The prototype was presented to the project team members again, so that they could conduct an initial survey of usability problems with the interface. In this survey, the following problems were found:

- Information at the top of the popup component takes up a significant part of it, reducing the space to display the Facilitated Content.
- The titles of Web pages (content of the element title on the page) often contain information that is not necessarily related to the text content selected for facilitation. This being the case, displaying this content as a heading element inside the popup could confuse application users.
- The necessary delay for Automatic Summarization and Syntactic Simplification operations is longer than 10 seconds, depending on the length of the text selected for facilitation. Thus, in these cases, it is necessary to display information about the progress of text processing operations to the user.

Consequently, we adapted the prototype interface taking these problems into consideration. The resulting interface can be seen in Figure 4.

In this interface, the popup title, which was previously at the top of the component, was inserted at the bottom and its contrast with popup background color was reduced as a way of not disturbing users when they are reading. We also removed the title of the page on which *Facilita* was activated, according to the rationale identified in the initial survey of usability problems. Finally, we added a progress bar to show the progress of text processing operations performed by *Facilita*.

It is important to stress that we have not yet carried out usability tests with users to validate the prototype. However, evaluations and usability tests are planned for the next iterations of the project.

7. FINAL REMARKS

This paper discussed questions about the interaction design and development of an assistive technology for low-literacy users (functionally illiterate) to read text contents in Brazilian Portuguese on Web sites and applications.

It describes the automatic generation of an accessible content through operations for text simplification (Automatic Summarization and Syntactic Simplification), together with interface design approaches for low-literacy users, who often have also a low level of experience and skill in using computers.

As previously emphasized, we have not yet carried out design iterations with actual users to validate the prototype. Thus, evaluations and usability tests will be conducted in future work to validate the interface developed. Our future work also includes the addition of lexical simplification to *Facilita* and the development of *Facilita Educativo*, which will use Text Elaboration resources and methods.

8. ACKNOWLEDGMENTS

The authors thank to all members of the PorSimples project who directly or indirectly contributed to develop this study, and to CAPES, FAPESP, and Microsoft Research (MSR) for financial support.

9. REFERENCES

[1] S. M. Aluisio, L. Specia, T. A. Pardo, E. G. Maziero, and R. P. Fortes. Towards brazilian portuguese automatic text simplification systems. In *DocEng '08: Proceeding of the eighth ACM symposium on Document engineering*, pages 240–248, New York, NY, USA, 2008. ACM.

[2] J. Axelsson, C. Cross, J. Ferrans, G. McCobb, T. V. Raman, and L. Wilson. Xhtml+voice profile 1.2. Technical report, March 2004.

[3] S. A. Becker. A study of web usability for older adults seeking online health resources. *ACM Trans. Comput.-Hum. Interact.*, 11(4):387–406, 2004.

[4] E. Bick. The parsing system "Palavras": Automatic grammatical analysis of portuguese in a constraint grammar framework. PhD Thesis - Aarthus University, 2000.

[5] E. H. Blake. Extended abstract a field computer for animal trackers. *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 532–533, 2002.

[6] A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction (3rd Edition).*

Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

[7] A. P. Freire, R. Goularte, and R. P. de Mattos Fortes. Techniques for developing more accessible web applications: a survey towards a process classification. In *SIGDOC '07: Proceedings of the 25th annual ACM international conference on Design of communication*, pages 162–169, New York, NY, USA, 2007. ACM.

[8] A. P. Freire, C. M. Russo, and R. P. de Mattos Fortes. The perception of accessibility in web development by academy, industry and government: a survey of the brazilian scenario. *New Review of Hypermedia and Multimedia*, 14(2):149–175, December 2008.

[9] C. Gasperin, E. Maziero, L. Specia, T. A. S. Pardo, and S. M. Aluisio. Natural language processing for social inclusion: a text simplification architecture for different literacy levels. In the Proceedings of *SEMISH - XXXVI Seminário Integrado de Software e Hardware, 2009*. CSBC - XXIX Congresso da Sociedade Brasileira da Computação, pages 387-401, 2009.

[10] C. Gasperin, L. Specia, T. Pereira, and S. M. Aluísio. Learning when to simplify sentences for natural text simplification. In the Proceedings of *ENIA 2009 (VII Encontro Nacional de Inteligência Artificial)*, Bento Goncalvez, RS, Brazil. CSBC - XXIX Congresso da Sociedade Brasileira de Computação, pages 809-818, 2009.

[11] K. Ghosh, T. S. Parikh, and A. L. Chavan. Design considerations for a financial management system for rural, semi-literate users. *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 824–825, 2003.

[12] A. Graesser, D. S. McNamara, M. Louwerse and Z. Cai. Coh-Metrix: Analysis of text on cohesion and language. *Behavioral Research Methods, Instruments, and Computers*, V. 36, pp. 193-202.

[13] IBGE. Síntese dos indicadores sociais - uma análise das condições de vida da população brasileira, 2008. Available at: http://www.ibge.gov.br/home/estatistica/populacao/condicaodevida/indicadoresminimos/sinteseindicsociais2008/indic_sociais2008.pdf.

[14] B. Kelly, D. Sloan, L. Phipps, H. Petrie, and F. Hamilton. Forcing standardization or accommodating diversity?: a framework for applying the wcag in the real world. In *W4A '05: Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A)*, pages 46–54, New York, NY, USA, 2005. ACM.

[15] D. L. Mapleson. Post-grammatical processing for discourse segmentation. PhD Thesis - University of East Anglia, Norwich, 2006.

[16] D. Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA, 2000.

[17] P. Margarido, T. Pardo, G. Antonio, V. Fuentes, S. Aluísio, and R. Fortes. Automatic summarization for text simplification: Evaluating text understanding by poor readers. In the Proceedings of *VI Workshop em Tecnologia da Informação e da Linguagem Humana*, pages 310–315. In: VI Workshop em Tecnologia da Informação e da Linguagem Humana, 2008, Vila Velha., 2008.

[18] P. R. A. Margarido, T. A. S. Pardo, and S. M. Aluísio. Sumarização automática para simplificação de textos: Experimentos e lições aprendidas. In the Proceedings of *Workshop de Usabilidade, Acessibilidade e Inteligibilidade aplicadas em interfaces para analfabetos, idosos e pessoas com deficiência, 2008*, pages 1–6. In: Workshop de Usabilidade, Acessibilidade e Inteligibilidade aplicadas em interfaces para analfabetos, idosos e pessoas com deficiência, 2008, Porto Alegre, 2008.

[19] A. Max. Writing for language-impaired readers. In *Computational Linguistics and Intelligent Text Processing*, volume 3878 of *Lecture Notes in Computer Science*, pages 567–570. Springer Berlin / Heidelberg, 2006.

[20] I. Medhi, A. Prasad, and K. Toyama. Optimal audio-visual representations for illiterate users of computers. *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 873–882, 2007.

[21] I. P. Montenegro and A. Educativa. Inaf - 2007, 2007. Available at: http://epoca.globo.com/edic/499/_INAF_2007_dez07_.pdf

[22] T. Parikh, K. Ghosh, and A. Chavan. Design studies for a financial management system for micro-credit groups in rural india. *CUU '03: Proceedings of the 2003 conference on Universal usability*, pages 15–22, 2003.

[23] M. Plauché and M. Prabaker. Tamil market: a spoken dialog system for rural india. *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1619–1624, 2006.

[24] A. Siddharthan. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109, June 2006.

[25] C. F. R. Souza and M. G. V. Nunes. Avaliação de algoritmos de sumarização extrativa de textos em português. Technical Report at ICMC-USP, NILC-TR-01-09, 2001.

[26] W3C. Web content accessibility guidelines 1.0. W3C Recommendation, May 1999. Available at: http://www.w3.org/TR/WCAG10/

[27] W3C. Web content accessibility guidelines (wcag) 2.0. W3C Recommendation, December 2008. Available at: http://www.w3.org/TR/WCAG20/

[28] W. M. Watanabe and R. P. de Mattos Fortes. Revisão sistemática sobre princípios de design de aplicações web acessíveis para analfabetos funcionais. In the Proceedings of *SEMISH - XXXVI Seminário Integrado de Software e Hardware, 2009*. CSBC - XXIX Congresso da Sociedade Brasileira da Computação, pages 403-417, 2009.

[29] D. N. Young. Linguistic simplification of sl reading material: effective instructional practice. *The Modern Language Journal*, 83(3):350–366, 1999.