

## ACCESSIBILITY SUMMARIZATION & SIMPLIFICATION IN A TEMPLATE-BASED WEB TRANSCODER

BAMBANG PARMANTO, ANDI SAPTONO, REZA FERRYDIANSYAH, LIJING SONG, I WAYAN SUGIANTARA & STEPHANIE HACKETT

*University of Pittsburgh, Pittsburgh, Pennsylvania 15260, USA  
{parmanto/ans38/ref11/lis18/iws1/srhst18}@pitt.edu*

Received July 25, 2005

Revised June 14, 2006

The goal of this project is to make the Web more accessible by providing some of the features naturally available to sighted users to users with visual impairments. These features are direct access and gestalt understanding, which can emerge from simplification and summarization. Simplification is achieved by retaining sections of the webpage that are considered important while removing the clutter. The purpose of summarization is to provide the users with a preview of the webpage. Simplification and summarization are implemented as a “guide dog” that helps users navigate the entire website. Simplifying the Web and summarizing the contents require understanding the structure underlying the entire webpage document. By understanding the structure, we can decide which sections of the webpage are important and therefore to keep, and which ones are clutter and need to be skipped. The bulk of our work has been in understanding the structure of a random document with the goal of transforming its contents into a format that is more accessible to users with disabilities as well as users of mobile devices. Understanding the structure of a random webpage is a difficult problem. We propose a novel method using a combination of pagelet feature recognition and genre-based templates.

*Key words:* Web accessibility, summarization, simplification, template, cybergenre

### 1 Introduction

Using quick scanning and multimodal representation of the Web (image, text, etc), sighted users can get a gestalt comprehension of the Web almost in an instant. Based on this gestalt view of the webpage, the users are then able to decide what section of the Web to view and jump directly to that section. Visually impaired users, on the other hand, need to sequentially read the content of a webpage to form a comprehension of the page.

Visually impaired users usually rely on a screen reader to interact with computers and the Web. A screen reader is a software program that allows a blind person to read aloud text on the screen, converting the text into voice through a speech synthesizer or the computer's sound card. Screen readers generally read Web pages word by word (including clutter) from top to bottom and left to right. Figure 1 provides illustration on the sequence of reading of a webpage by a screen reader. In sequential reading and comprehension, Web clutter becomes a more serious obstacle than it is in direct access reading. In this paper, we use the term clutter to refer to content on the page that is deemed unwanted or unimportant.

Sighted users can easily ignore the obstacles and go directly to the intended section of a webpage. Visually impaired users have to go through obstacles one by one and move from one item to the next before reading the intended section.

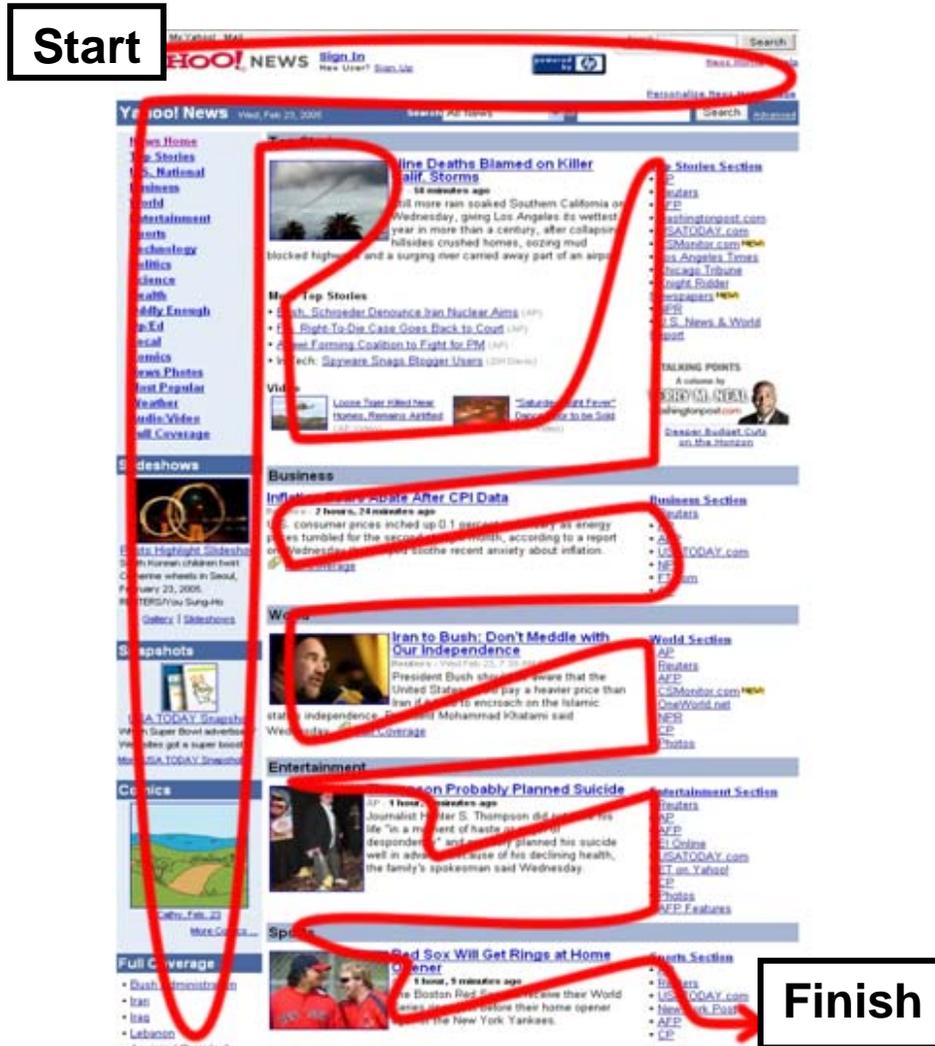


Figure 1 Screen Reader’s Sequential Reading

The goal of this project is to make the Web more accessible by providing some of the features naturally available to sighted users to users with visual impairments. The first feature is simplification. Simplification is achieved by retaining sections of the webpage that are considered important while removing the clutter and sections that are considered not important. The second feature is summarization. The purpose of summarization is to provide the users a preview of the webpage. This preview will assist visually impaired users in developing a gestalt comprehension similar to the one that is naturally attained by sighted users. Simplification and summarization are implemented in conjunction with a “guide dog” that helps users navigate the entire website.

Simplification and summarization can be used in two situations: transcoding existing webpages (HTML-based) to make them more accessible and displaying structured document for users with disabilities. Structured documents such as XML-based or database documents are easier to deal with

since the structure of the documents provides meaning about the content. This paper focuses on the situation where existing webpages need to be transcoded to make them more accessible for users with visual impairments. We implement the concepts of simplification and summarization as part of the accessibility transcoding gateway developed in our earlier project [1].

## **2 Simplification and Summarization**

The AcceSS system works by applying a page understanding algorithm to learn the overall content of the page, as well as the content of every section within the page. Once the content is understood, AcceSS provides a summary of the page to allow for a quick preview of the entire page and a summary of each section to allow for a quick preview of the section content. The summary will allow visually impaired users to navigate the entire page quickly. In addition to summarization, AcceSS also simplifies the structure of the page and the structure of every section of the page. The simplification will allow the user to navigate the page and the section smoothly. Simplification is done by removing the clutter or selecting only the most important sections of a webpage. A similar approach using page annotation is page clipping [2]. Instead of annotation, we use genre-based template and pattern matching algorithms that are potentially more extensible.

The goal of summarization is to provide visually impaired users a quick glance of the page, thus assisting them in building a gestalt model of the page. Our summarization method relies heavily on the structure and important landmarks of the page, such as abstract, title, and subtitles to build the summary of a page. Harper et al. [3] suggested that the use of context probing to assist preview may increase the mobility of visually impaired users in their virtual journeys through the Internet. The work focuses on extending the information on links in a page by adding context which describes the destination of the links. In this project, we extend the concept of context probing, not only to give information on links, but also to create a preview of the entire page in which the user interacts (summary of the page). In our previous work ([4]; [5]) we found that summarization helps users of wireless handhelds in reading large complex document such as journal articles. In that work, a similar approach using table of contents (TOC) for providing overview was proposed [6].

Simplifying the Web and summarizing the contents require understanding the structure underlying the entire webpage document. By understanding the structure, we can decide which sections of the webpage are important and therefore to keep, and which ones are clutter and need to be skipped. In order to understand the contents of a webpage and to decide which parts of the webpage are more important than others, we propose a novel method using a combination of pagelet feature recognition and genre-based templates.

There has been extensive work on understanding Web documents with the goal of adaptive content delivery [7-10], topic distillation [11], record boundary discovery [12], and of improving web information retrieval [13, 14]. Little work, however, has been done on using webpage understanding and summarization technique for improving the usability of the web for users with visual impairments. Early work on BrookesTalk [15] generates a summary and abstract of the Web document using natural language processing techniques. Our system treats a webpage as a collection of various page elements (ranging from navigational bar, stories, advertisement, etc). In this situation, using an information retrieval technique to generate a summary will return a number of unrelated sentences. Instead, we utilize the headlines and other important landmarks of the page to create an outline of the page, which functions as the summary of the page.

Our work relies on the detection of elements contained inside the page. There have been many methods suggested to do this. Takagi et al. [16] uses XPath to mark various sections. This is very efficient for site-wide transcoding, since usually there is little change in the structure of the pages within the site. We use the template matching technique because our goal is to transcode arbitrary sites. We found that utilizing XPath to pin-point a Web element creates a scalability issue in transcoding arbitrary sites. Some visual layouts can be built using different HTML codes, thus creating different DOM Tree structures. This means that across different pages, a Web element, while 'visually' being placed at the same location in the display, might have a different XPath. Another problem arises when a page is being generated dynamically. If 'noise' (such as an advertisement) is placed between Web elements in the page, it might change the XPath slightly. In this case, the XPath might point to a different node, creating a false positive condition. With the template matching technique, we have a flexible way to detect page parts (with different DOM Tree structure) while avoiding noise (like advertisements, which are inserted dynamically).

Ramakrishnan et al.[17] rely on recurring patterns to detect important sections and then find their function by using text analysis. We show a different method of detection. Ours is a two-step detection process. We first detect the landmarks of the template (for example in Figure 2 we detect the 4 main components Masthead, Left Menu, News Content, Advertisement). The second phase is detecting elements inside those landmarks, such as news titles and menu elements, which we call pagelet detection.

Our work in document understanding relies on the use of templates and genres. Classification of webpages by their genre has been an active area of research in the past several years [18]. The use of genres is to improve search results. The inclusion of genre information as part of the query can significantly improve the performance of the query [19].

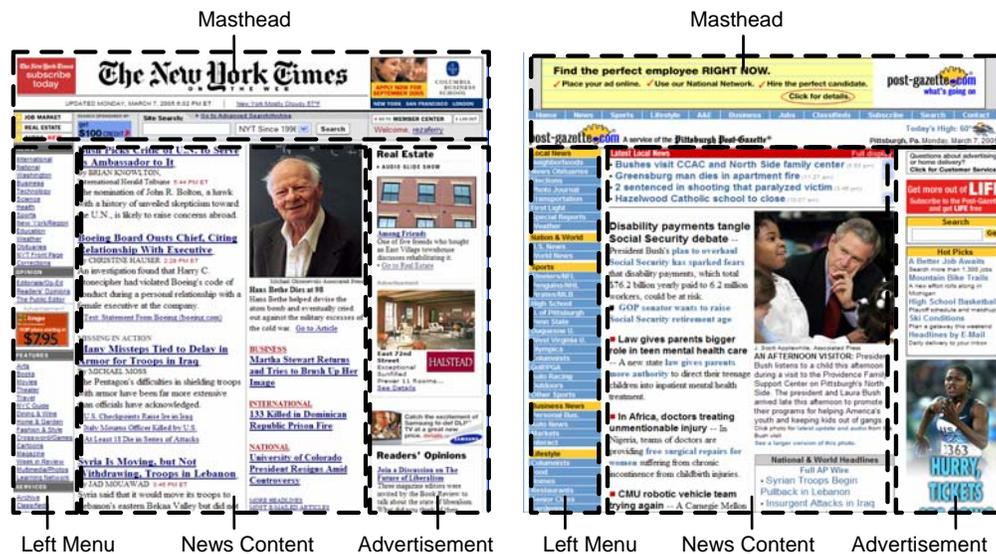


Figure 2 Templates in the News Genre: New York Times and Pittsburgh Post-Gazette

In this project, the genre is used to help understand what kind of simplification and summarization needs to be done during transcoding process to make webpages more accessible. Therefore, the focus

of our work is more on the structure and functionality of the webpages, rather than the content and the meaning of the content. However, the features and heuristics that are used in Web classification research can also be used to determine the genre in the transcoding research. There is no standard taxonomy of Web genre, no consensus yet as to the variety of genres found on the Web. There may be two reasons for the lack of consensus: genre classification is determined by the purpose of the classification, and genres, like the Web itself, continuously changes and grow. Previous work on Web classification use Web genres such as personal home page, corporate home page, and organization webpage. The way we classify the Web in this study is completely different. We divide the Web genre into news, e-commerce, informational directory, etc.

An interesting work related to this project is the Vision-based Page Segmentation (VIPS) algorithm [13] that aims to extract the semantic structure of a webpage based on its visual presentation. The semantic structure is a node in the DOM Tree structure that represents blocks. In our work, we refine the concepts of blocks by developing the concepts of landmarks (see next section).

### 3. Genre-Based Template

Humans interact with a familiar environment better than a non-familiar one. Consequently, humans prefer working in a familiar environment rather than a new, unfamiliar one. This tendency drives site builders to use a familiar site environment (template) to capture the interest of their visitors. Additionally, the use of a template in a website reduces visitors' memory workload by giving them a familiar interaction schema. This concept has been widely adopted by website development tools or document authoring tools by providing templates for several types of websites or documents. It is also no accident that many e-commerce websites follow the style of Amazon.com or news websites tend to look similar. Figure 2 shows a prototypical template for news websites: news content section usually is located in the middle, the top of the page contains the masthead, while the left part contains links to various pages and sections of the news website.

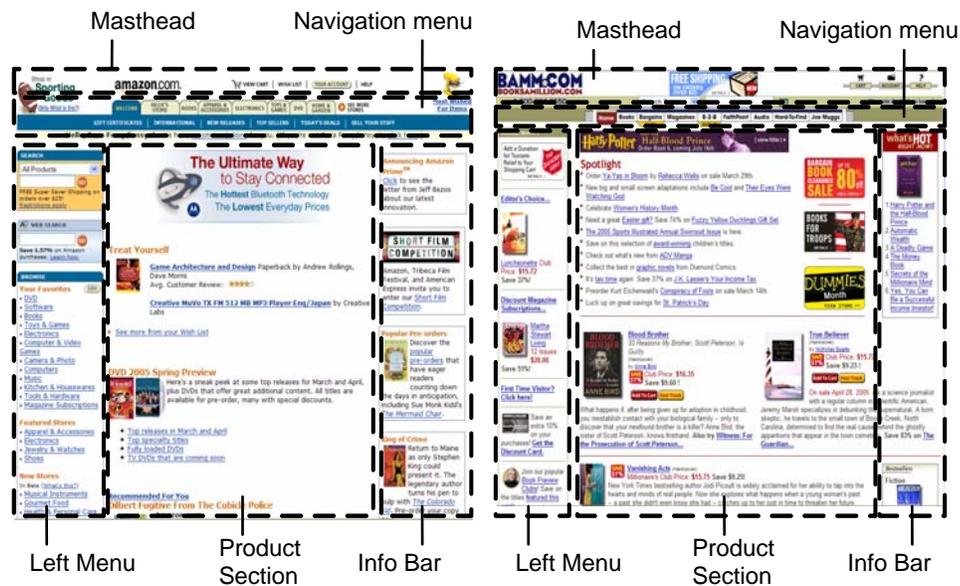


Figure 3 Templates in the E-commerce Genre: Amazon.com and Bamm.com

In addition to adopting a familiar look (that explain why news websites tend to look similar), the tendency for websites of a similar genre to have similar templates can be explained by functionality. E-commerce websites have similar functionality (list of merchandise, reviews, and a check-out transaction) and it is more efficient to organize the website in certain ways. Once a popular pioneering website is successful in developing an effective website (Amazon.com is a good example), other newer sites are compelled to follow. In figure 3, notice the navigation menu located under the masthead that has become a signature for e-commerce websites, while the middle section usually contains the list of products on sale or feature products.

In this project, we will exploit the tendency of websites from similar genres to look alike and to have similar functionalities. Based on the genre concept, we propose to build templates for each website genre. Templates contain important parts of the page that sighted users can recognize easily while traversing a webpage. We refer to these easily recognizable areas of a webpage as landmarks. The template of a genre website consists of a set of landmarks that are arranged in a certain composition. Similar to traversing in the real-world environment, landmarks in a website help users navigate that website. For example, a template may have a header landmark, a left-side vertical-navigation bar landmark, and a content landmark.

Genre-based templates are constructed from a prototypical website of each genre. For example, news.yahoo.com and CNN.com are used to construct the templates for the news genre while Amazon.com is used to construct one for the e-commerce genre. If the features within the genre are quite different from one another, more than one sub-genre can be constructed within a genre. For example, although news.yahoo.com and CNN.com are both news pages, they have substantial feature differences and therefore two sub-genres are constructed. Also, within a genre, different templates are constructed for different levels of websites. For example, within the news genre the template for the main page is quite different than the template for the full story page.

```

<root>
  <template feature_amount="5" name="news 0">
    <feature name="0" description="Page Header" base="top" direction="down"
      min_distance="0" border_type="0" />
    <feature name="1" description="Navigation Bar" base="0" direction="down"
      min_distance="30" border_type="1" />
    <feature name="2" description="Right Content" base="1" direction="down"
      min_distance="10" border_type="1" />
    <feature name="3" description="Bottom Content" base="2" direction="down"
      min_distance="100" border_type="1" />
    <feature name="4" description="Page Footer" base="bottom" direction="up"
      min_distance="10" border_type="0" />
  </template>
</root>

```

Figure 4 Example of a News Template XML Document

Templates are partly implemented as an XML document containing rules to identify landmarks. Figure 4 shows an example of an XML document that stores a template of a news genre. In this example, the

template has five landmarks: Page Header, Navigation Bar, Right Content, Bottom Content, and Page Footer. A library of templates is constructed from individual templates like this.

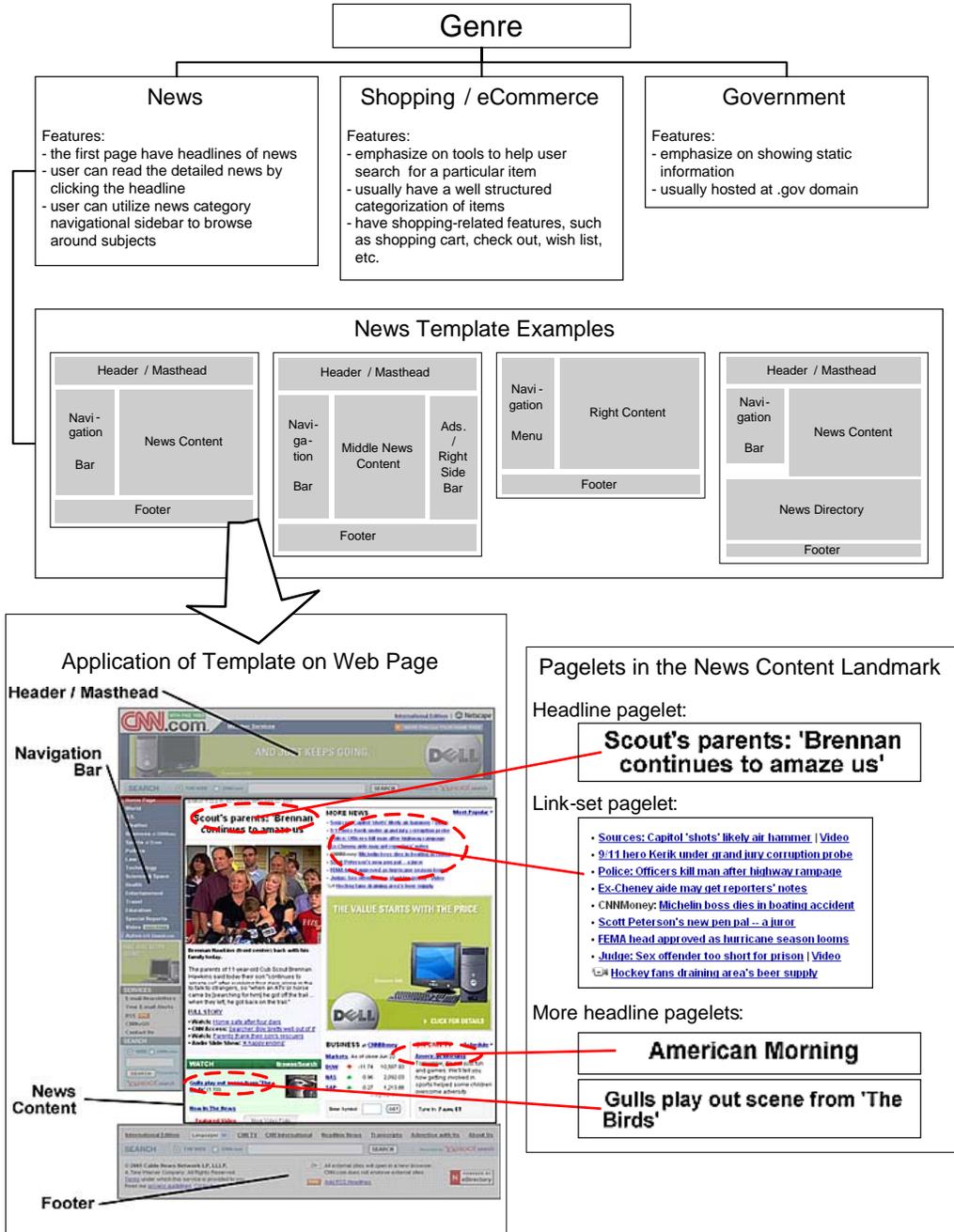


Figure 5 Hierarchy of Genre, Template, Landmark, and Pagelet

A pagelet is defined by Bar-Yossef & Rajagopalan [20] as a self-contained logical region within a page that has a well defined topic or functionality. A webpage consists of multiple pagelets which are grouped into several types based on functionality. One pagelet type may be heading, which separates one section from another. Another pagelet type may be navigation menu which takes the user to other parts of the sites. Figure 5 illustrates the hierarchy of genre, template, landmark and pagelet. In this illustration, the CNN.com homepage belongs to the News genre. Within the News genre, the CNN.com homepage matches visually with one of the templates in our library. The template that matches the homepage consists of 4 landmarks: header/masthead, navigation bar, news content and footer. Further refinement on the “news content” landmark resulted in several pagelets, such as headlines and link-set directories.

#### 4. Webpage Understanding through Template Matching and Pagelet Detecting

The bulk of our work has been in understanding the structure of a random document with the goal of transforming its contents into a format that is more accessible to users with disabilities as well as users of mobile devices [21]. Understanding the structure of a random webpage is a difficult problem. To make the problem less complicated, we enlist the help of prototypical templates as discussed in the previous section. Since the template library is organized into genres, the first step to understanding a webpage structure is to determine the genre of the page. The purpose of genre detection is not only important in the page understanding process (what type of website we are dealing with and which template it matches), but also to determine what kind of transformation would be needed.

The genre detection process is conducted in two phases: table lookup and heuristic rules. In table lookup, we list popular webpage URLs in a table and associate them with the appropriate genres. If the URL of the webpage is not listed in the table, the heuristic rules are then used. We analyze the content of the page to gather information that can be used to determine the genre of the page by using heuristic rules. The rules can range from simple text matching rules (such as 'find a sentence with the term "shopping cart" in it'), to more sophisticated rules ('if a certain body of text has a date or time associated to it, it is a news text'). These rules were utilized to look for genre-specific characteristics in the content such as the occurrence of specific keywords, length of text, length of menus, etc. For example, 'shopping cart' and links to cart-related pages indicate that the page is part of an 'e-commerce' genre. The latter rule ('if a certain body of text has a date or time associated to it, it is a news text') is one of the rules to detect 'news page' sub-genre.

The focus of this paper is on the framework and overall architecture of a template-based transcoding system. Currently, we have experimented with the websites of two genres: news and e-commerce. Evaluation of the genre and template-based detection algorithm is beyond the scope of this paper. Shepherd et al. [11] provides an excellent framework on how classification of webpages based on the genre can be conducted.

Once the genre of the webpage has been determined, the next step is to find the template that best matches the page. Since a template consists of landmarks, the matching process is carried out by comparing the landmarks of the webpage and the template. The templates are implemented as an XML document and an object, with the rules to detect landmarks implemented as the methods. These objects scan the DOM Tree of the webpage to find matching features in the structure and provide a similarity percentage between the structure and the template (for example, a template with title, left menu, right content and footer features will give a 75% similarity percentage when scanning a page which only has

title, content and footer parts). The template with the highest similarity with the webpage is selected to represent the webpage. The template represents the macrostructure of the webpage that consists of landmarks such as left menu, main content, etc. The template, however, does not have information about the detailed structure within the landmark. For example, the main content of a news webpage contains several news items. These news items are not identified in the landmark. In order to process the microstructure within a landmark, the concept of pagelet comes into play.

We create a set of HTML element patterns for detecting each pagelet. The patterns are regular expressions consisting of tokens representing HTML elements. The pattern matcher tries to match the HTML elements for each section of the page. Each landmark (section) contains different types of pagelets. For example, the info bar landmark looks for the appearance of a navigation menu, while the main content section looks for heading and news pagelets. When the pattern matches a particular HTML element, the corresponding HTML element is marked as the start of a particular pagelet. After all patterns are found, the result of the pattern matching process is refined. If an element is marked as matching two different pagelet patterns, one of the marks is deleted. The choice of the pagelet to delete depends on a priority score of the corresponding pagelet.

A pagelet type is often expressed by visual cues so that users can easily detect and use the various functions for that pagelet. For example, a heading is usually in a different font from the text it precedes. The visual information of a pagelet is encoded in the HTML code. Based on this, we propose the use of HTML element patterns to detect these pagelets.

We create a set of HTML element patterns to detect each pagelet. The patterns are regular expression patterns which consist of tokens representing HTML elements. There may be multiple patterns for each pagelet type. In our architecture, each pattern matcher executes as a different thread. Currently all patterns are created manually based on our observations of the various pagelets found in various webpages.

Table 1. Pagelet patterns examples

Pagelet Type	Patterns
Heading	SEPARATOR<B>TEXT</B>SEPARATOR
Link to news	SEPARATOR<LINK>TEXT</LINK>SEPARATOR TEXT(TEXT <LINK>TEXT</LINK>)* SEPARATOR
Navigation Menu	<TR><TD><LINK>TEXT</LINK></TD></TR> (<TR><TD><LINK>TEXT</LINK></TD></TR>)+

Table 1 shows the structure of patterns for a few types of pagelets. Each pagelet has more than one pattern. The first pattern is one of the patterns in the heading pagelet: a heading is detected when the pagelet detector finds a text encompassed by a bold tag (<b>) in a line of its own. This pattern is based on the assumption that a heading commonly occupies a separate line and has different formatting from the rest of the content. Other patterns of the heading pagelet include a similar pattern with the encompassing element changed to <div> and <span> instead of <b>. The separator token matches HTML elements which creates a new line between previous text and the next text (e.g the elements

<br>, <div>, <p>). Every pattern matcher ignores elements that are deemed insignificant for the pagelet. For example, the horizontal rule element (<hr>) is ignored by the heading pagelet pattern matcher. Therefore the pattern above will still match the following structure in the HTML: <br/><hr/><b>Main Menu</b><br/>.

The Navigation menu pagelets are visually similar, and are often created by the use of tables. The pattern for the navigation menu pagelets tries to match table cells that have only links inside them. This pattern only marks the elements as navigation menu pagelets when there is more than one row that has links. If only one row has a link inside the cell, then the row is ignored.

To evaluate the performance of our algorithms we measured the recall and precision of the algorithms in detecting the pagelets. We calculate two measures (general and specific) for both recall and precision based on whether the pagelet was detected at all, and if it had been detected, was it the same pagelet type as that found by manual inspection.

General recall (GR) is calculated as the number of a specific pagelet detected regardless of what the pagelet was detected as divided by the number of the actual specific pagelet. The general recall shows how many of the pagelets were detected regardless of what they were detected as. In other words, the general recall is the number of pagelets found by any pattern matcher run through the system. As long as a feature is detected as a pagelet, it will be processed and not classified as unimportant, even though it may not be the optimal process for that particular pagelet. Specific recall (SR) is calculated as the number of specific pagelets detected correctly compared to the number of actual specific pagelets. Specific recall calculates only pagelets detected by pattern matcher associated with that pagelet.

General precision is the ratio of all pagelet elements detected as a particular pagelet (regardless of whether they were detected correctly) compared to all elements detected as a particular pagelet. Specific precision is the ratio of all pagelet elements detected correctly compared to all elements detected as a particular pagelet. Specific precision is always lower or equal to general precision. The difference of these two numbers shows the overlap of this pattern to other pagelets. If the difference is too high, the pattern may need to be modified so that it does not encompass other pagelet types.

Table 2. Recall and precision of method

	General Precision	Specific Precision	General Recall	Specific Recall
CNN	72.46%	59.58%	80.94%	66.56%
NYTimes	65.59%	44.12%	90.38%	60.80%
Wal-Mart	88.86%	77.99%	80.35%	70.53%

We tested the pagelet detection process on thirteen webpages from three websites. The sites were chosen based on their genre consisting of one e-commerce site (walmart.com) and two news sites (nytimes.com and cnn.com). The pages chosen for each site are different from each other in structure and function. The results of the pagelet detection are shown in the Table 2.

Although the recall is adequate, this process is quite imprecise. Some of the content detected as pagelets by the detection process are not really pagelets at all. This process is subsequently refined by using the section information based on the landmarks within the template. Each landmark has a specific functionality, and therefore each landmark only contains certain types of pagelets.

The page is divided into landmarks based on the template. In each landmark, different pagelet types are searched based on the most probable pagelets for each landmark’s function. A landmark such as the “info bar” looks for the appearance of a navigation menu, while the “content” landmark looks for heading and news pagelets. As expected, the precision of the detection process increases when landmark information is used. Table 3 shows the recall and precision for this method after segmenting the page into landmarks.

Table 3. Recall and precision of method with page segmentation

	General Precision	Specific Precision	General Recall	Specific Recall
CNN	89.05%	75.18%	81.61%	68.90%
NYTimes	71.35%	64.71%	88.26%	80.05%
Wal-Mart	90.31%	84.62%	79.85%	74.81%

Once the pagelets are detected, each pagelet can be transformed based on its type. The collective transformation of all pagelets is the transformation of the whole page. An example of the results of the transformation for the site shown above is shown in Figure 6. The page is divided into 3 landmarks: in the “masthead” landmark, no pagelets are detected; in the “left menu” landmark, navigation menu and headers are detected; and in the “content” landmark, related links, headings, and news are detected. The result of the pagelet detection process is shown below the site in Figure 6.

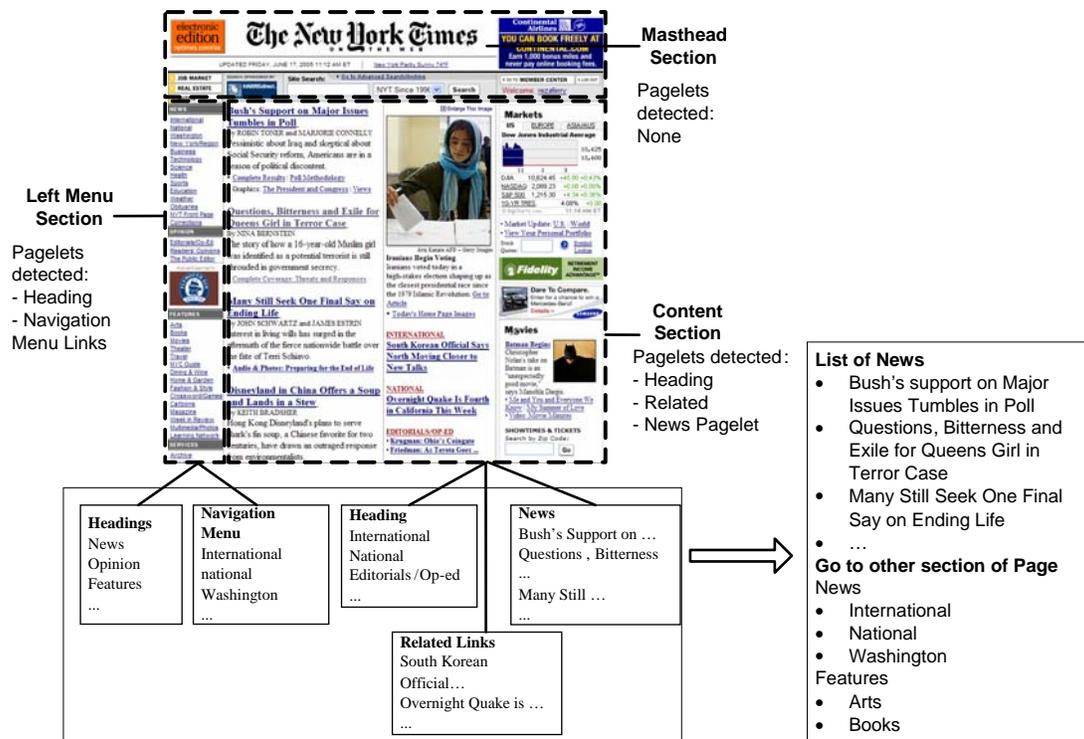


Figure 6 Pagelets Inside Landmarks in a News Template

Based on the various pagelets found, the page is transformed. The list of news headings (but not the summary of the articles) is put on the top. The left navigation links detected by these patterns are put on the bottom of the page. Other pagelets are ignored or put in a different page.

Using this method, the need for a manual annotation process such as that proposed by Takagi et al. [16] is eliminated. This method automatically detects the available pagelets in various pages. This approach is capable of dealing with pages that are dynamic or changing without any problem; it is also capable of handling sites that do not match any available annotation records.

The patterns are not specific to a single site and the same patterns can be used in various webpages to detect the pagelets. However site specific patterns can also be created if needed.

The limitation of this approach is that it still takes a significant amount of time to create and test the pagelet patterns during development. An easy pagelet may take 15 minutes to create and test while a more complex one may take a few hours. However, it is a one-time effort, as an already developed library of patterns can always be reused. The pagelet patterns created may not be the most optimal patterns for a particular pagelet. There may be better patterns with higher recall and precision. Patterns created based on learning methods may be a good idea in the future.

## 5. AcceSS Transcoding

In this project, we developed a web accessibility transcoder called AcceSS (Accessibility through Simplification and Summarization). Our system implements the template-based webpage understanding described in the previous sections and genre-based transcoding. In the genre-based transcoding, the genre of a webpage provides not only a clue as to the content of the webpage, but also the summarization strategy appropriate for the webpage. For example, a news webpage from a newswire usually contains hot stories that are intended for fast scanning or reading. Therefore, the summary for a news webpage should focus more on navigation between stories, allowing users to scan stories and jump from one story to another. Another example is a directory webpage which usually presents information in a highly structured and hierarchical fashion. The content of a directory webpage is linked to other sites, grouped by topics. Therefore, the summary of a directory webpage should reflect the hierarchy and the structure of the page, with similar links grouped together under proper topics in the hierarchy.

Two types of summarization are implemented in the genre-based transcoding: the guide dog and the page preview. The guide dog is intended to guide users with visual impairments to navigate through the webpage that they want. The content of a guide dog relies heavily on the context (genre) of the webpage. The guide dog normally consists of a short text to describe the webpage that the user interacts with (if available), links to important information or special feature pagelets of the genre (such as news, stories, etc) along with their context previews, a directory section that contains all the links available in the page, and additional interaction tools (if applicable).

The page preview is intended to give the user a snapshot of the subsequent pages that can be visited. The page preview is similar in concept to the expansion on context presented in Harper et al. [3]. The content of a page preview depends on the genre of the page. A preview of a full story from a news webpage, for example, consists of the first paragraph of the story. This is based on the assumption that news writers usually use inverse pyramid structure where the main idea and the most important information are located at the beginning of the story. Summarization may also include advanced

interaction tools such as content outline and site navigation tools. A content outline tool is an outline of pagelets contained in the webpage. It consists of pagelet titles and links to a pagelet, which enables the user to jump directly to one part of the webpage or between parts of the webpage.

The page summaries subsequently are incorporated to the original webpage in the output generation phase of the transcoding. In this phase, our system modifies the DOM structure of the original page to implement the summarization rules. At the start of the output generation phase, the system appends the original DOM structure with new section nodes such as the guide dog, the page preview and additional interaction tools. The summarization strategy determines what kind of new section nodes to be appended. The following step of the output generation page is placing pagelets into their appropriate sections according to the summarization strategy. In the news example of Figure 1, the detected navigation menu of the original page is inserted into a section called navigation; headings are copied in a section called story index; and the related links pagelet is ignored. The final process in the output generation phase is the creation of the transcoded page to be sent to the user. In this step, the system retrieves the section from the user request and adds additional information such as the title and the URL of the original page being shown. The output generation also adds links to other sections of the webpage to allow movement from one section to another.

## 6. Case Study

We use Yahoo!News as a test case to show how AcceSS works. The original Yahoo!News webpages contain many potential pagelets as well as clutter (figure 7). The webpage is very complex and difficult to navigate for the large group of sighted users (such as novice or older users). For visually impaired users, the webpage is even more daunting since it contains a huge amount of information and clutter. We use the news genre as a case study because the news genre is among the most visited sites compared to other genres [22].

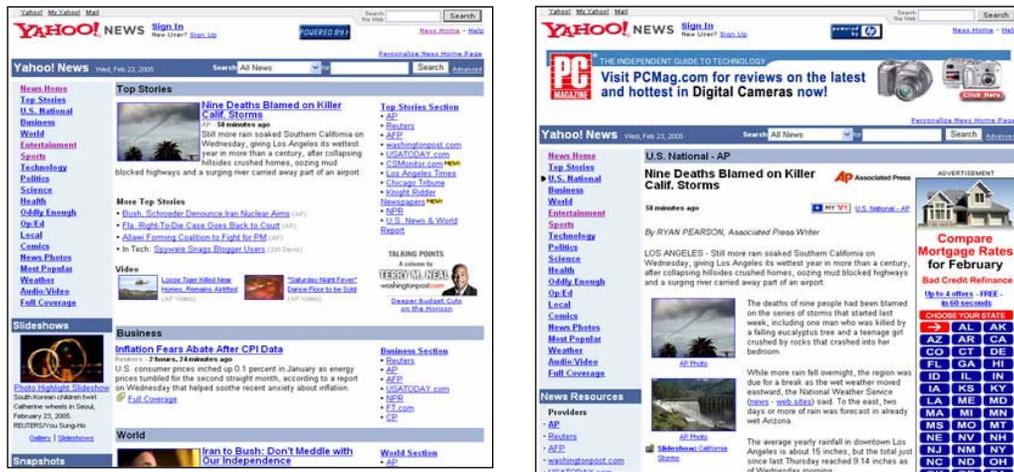


Figure 7 Yahoo!News Before Transcoding

Our transcoding system generates four important page types that visually impaired users can use to interact with this news site: guide dog, preview, full story, and site navigation page.

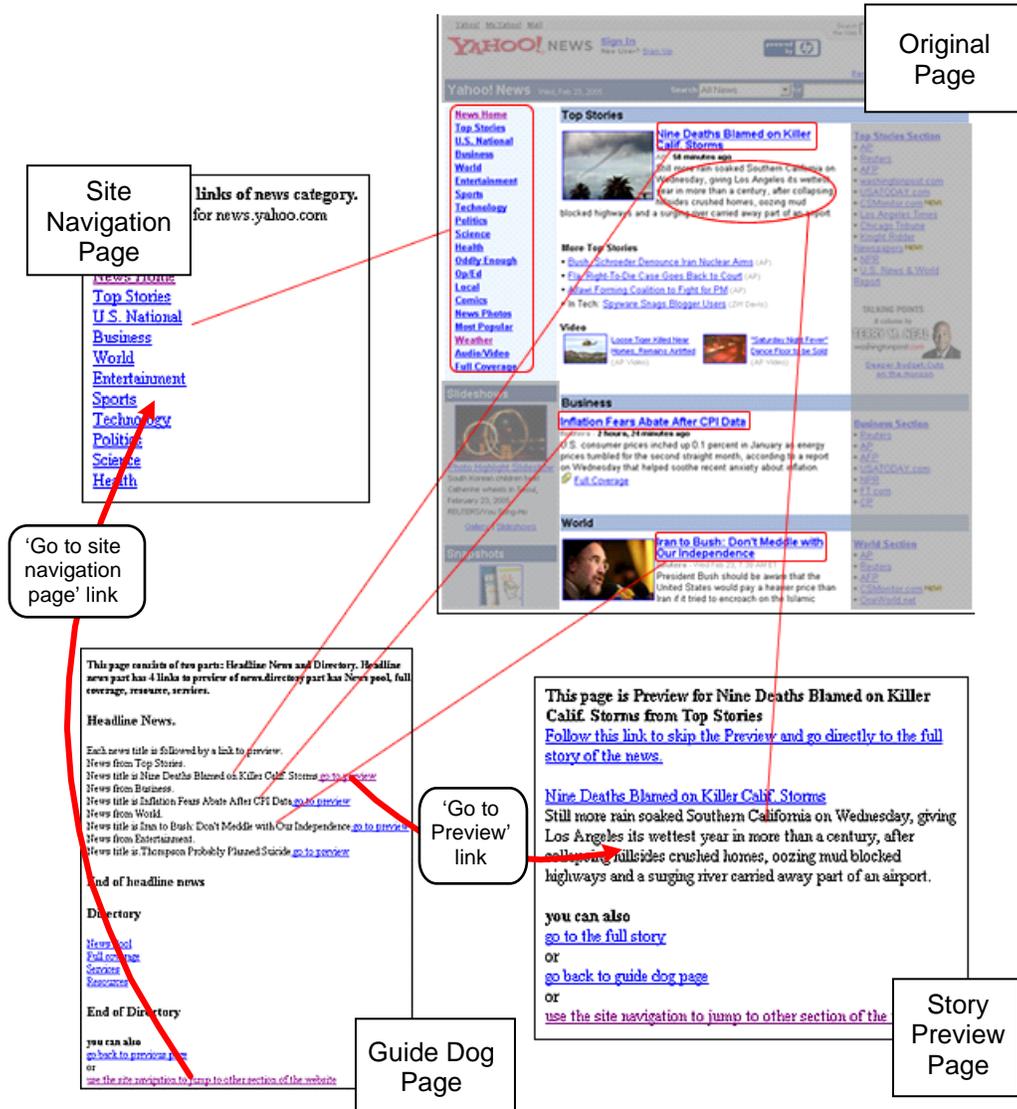


Figure 8 AcceSS Example: Transcoding Yahoo! News: Guide Dog, Site Navigation and Preview Page

Figure 8 shows the result of transcoding on a Yahoo!News Page. The original page shown on the top right part of the figure is separated into several different pages. The guide dog, which is shown at the bottom left part of figure 8, is the primary navigating page of our transcoded Yahoo!News page. The guide dog contains a list of news titles (and news title links), a directory containing links to special feature pagelets and additional news (news pool), and additional interaction tools (site navigation and go back page link). Clicking on the news title link brings the story preview page (bottom right of the figure), which the user can read before deciding whether to read the full story or skip the story altogether.

The navigation bar from the Yahoo!News page goes to a separate navigation page (top left of figure 8). This is done to reduce the clutter of the page and help visually impaired users to differentiate between the informational part of the page and the navigational part of the page. The users access this navigation link through clicking the navigation site link on the guide dog.

Our pilot usability study shows that visually impaired users have difficulty in finding the Yahoo!News full story page because of all the clutter that the user must listen to before finding the link to the story. A user is likely to follow an incorrect link, and thus get lost. With our transcoding, the user starts by reading the guide dog and immediately finds the link to the news that they are looking for. The user is able to read the full story page in a short amount of time, without having to waste time listening to the clutter. Figure 8 also shows the clutter hidden by our transcoding system to give the user quick access to the main content. Our transcoding system integrates the guide dog, preview page, and full story page to create a seamless interaction in reading the news. The hidden clutter can still be accessed through appropriate links in the guide dog page.

## **7. Usability Study**

We have conducted a usability study to evaluate whether the AcceSS approach helps in making websites more accessible and usable to users with visual impairments. The results of the usability studies have been favourable. Visually impaired users found that the AcceSS approach makes the website more accessible and usable. This section provides brief results of the usability study. A more complete report of the usability study is presented in Hackett and Parmanto [23].

During the pilot study that was conducted prior to the usability study, two pilot participants were asked to complete 3 tasks on the original Yahoo! News website and the transformed website. Both participants used JAWS for Windows, with Internet Explorer as their browser and Windows 98 as their operating system. The results of the pilot study are promising. Each participant was able to complete all three tasks using the transformed website, while only one participant was able to complete any of the tasks using the original site.

Both users preferred the transformed website over the original. Additionally, one user especially liked the lack of graphics. This user also felt that it would be better to go to the full story instead of going to the abstract/preview page first. Even when completing the task that explicitly asked the user to utilize the abstract page to obtain the answer, this user went to the full story for the answer. The second user, however, utilized the abstract page when searching for the answer to each of the three tasks on the transformed website. Both participants appeared to have a good grasp of the virtual guide dog and the site navigation tools by the end of the transformed session. Each of the users was more satisfied with their experience using the transformed website than the original.

Subsequent to the pilot study, we evaluated the usability of the template-based transcoding technology on six users with visual impairments. Although we did not ask participants to use a certain computer or screen reader, we found that all participants used the same computer environments as the one used in the pilot study (including the use of JAWS and Windows IE). The study was conducted in the participant's own home or office to provide more realistic setting. This setting also allows the participant to use the computer equipment and assistive technology with which they were most comfortable. We evaluated the usability of the Web in terms of time and success rate of the users in accomplishing the tasks. We also used subjective measurements (satisfaction, frustration, and

confidence) to measure the usability of the gateway compared to the accessing the web directly without the gateway.

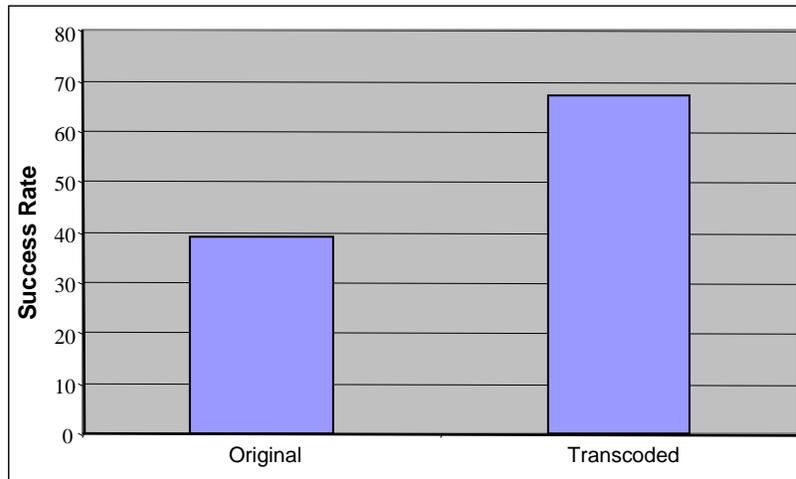


Figure 9 Comparison Between the Transcoded and the Original: Success Rate

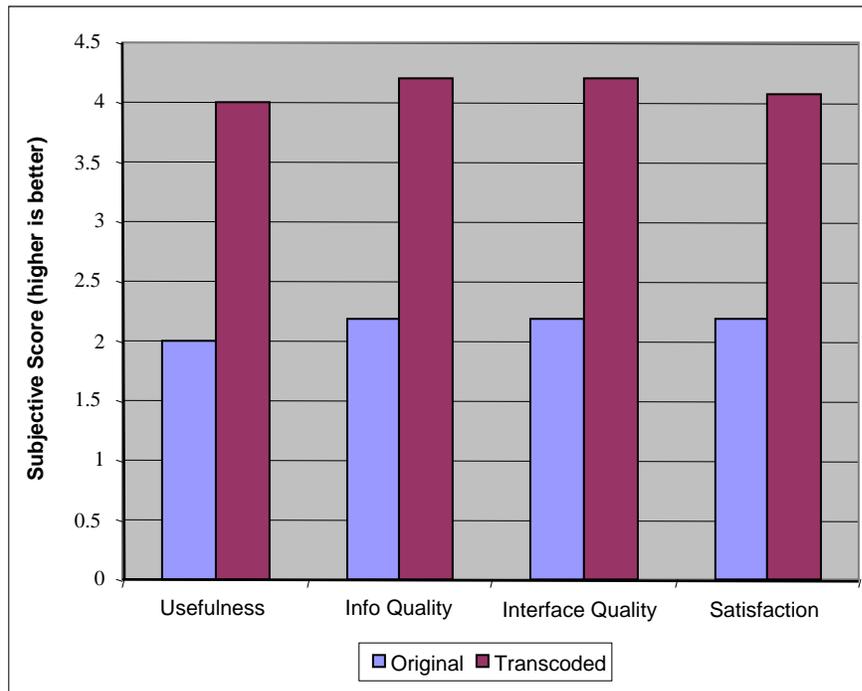


Figure 10 Comparison Between the Transcoded and the Original: Subjective Scores

Our studies show that the success rate increases when people used our gateway instead of accessing the Web directly. An increase of almost twofold was observed in the success rate of the users using the transformed website compared to the original website (67% success rate compared to 39%, as shown in Figure 9) In addition, for each task more users were able to provide the correct answer below the five minute threshold using the transformed website than the original. Time started after the task was read to the participant and finished when the user supplied their answer to the task or at five minutes, whichever occurred first. The reason for the five minute threshold on each task was to keep user fatigue to a minimum. The average time is not reported here since more than half of the participants could not finish the tasks using the original website within the five minute threshold.

On average, participants feel more satisfied, less frustrated, and more confident. Overall, all participants were more satisfied with the transformed than the original. Similar to the objective scores, the subjective scores of the transformed websites are increased nearly by twofold compared to the original websites. It is anticipated that the virtual guide dog will be beneficial to all visually-impaired Web users, regardless of experience, but may be more so to the novice and intermediate user. Users' collective satisfaction in all three of these areas was greater when using the transformed website over the original website (Figure 10).

## **8. Discussion and Future Work**

We propose a template-based transcoder for automatically transforming websites to make it more accessible and usable for users with visual impairments. Automatic transcoders have the potential for improving accessibility and usability without the prohibitively expensive process of manually retrofitting the Web. A report published by Forrester Research [24] estimated that the global 3,500 companies can expect to spend \$560 million to retrofit their websites to meet W3C Web Accessibility Initiative guidelines. Compared to the current annotation-based transcoder, the template-based transcoder can potentially reduce the cost even further by eliminating the labor-intensive annotation process. The ultimate goal of a transcoder is to increase the usability of a website for users with disabilities, not just increasing accessibility. Sites that are accessible, but not usable, will not bring improvement to the experience of users with disabilities.

The usability study on the template-based transcoder shows that the transcoding process can improve the usability of the websites. The participants found the usability of the transformed website is better than the original site as measured in the usefulness, the information quality, and the interface quality.

The focus of this paper is on the overall architecture of the template-based transcoder. Template-based page understanding potentially has a wide-range of applications beyond increasing the usability of the web page, such as implementing a context-based search and information retrieval. On the transcoding technology, we plan to use the template-based system for a site-wide transcoding system. That is, a transcoder that is used for site-specific instead of general random websites. We expect that the site-wide template-based transcoder will have a higher accuracy and a nearly automated system, features that make the transcoder's deployment practical and economically attractive.

## Acknowledgements

This work was supported in part by grant number 42-60-02013 from the National Telecommunications and Information Administration (NTIA) and grant number H133A021916 from the National Institute on Disability and Rehabilitation Research (NIDRR).

## References

1. Parmanto, B., et al. *Accessibility Transformation Gateway*. in *HICSS: Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. 2005. Big Island, Hawaii: IEEE Computer Society.
2. Hori, M., et al. *Annotation by transformation for the automatic generation*. in *Pervasive 2002*. 2002.
3. Harper, S., et al., *Middleware to expand context and preview in hypertext* in *Proceedings of the ACM SIGACCESS conference on Computers and accessibility 2004* ACM Press, p. 63-70
4. Parmanto, B., et al. *Transcoding Biomedical Information Resources for Mobile Handhelds*. in *HICSS: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. 2005. Big Island, Hawaii: IEEE Computer Society.
5. Saptono, A., *Web Transcoding Gateway Usability Study*. submitted, 2005.
6. Yesilada, Y., et al., *Screen readers cannot see - Ontology based semantic annotation for visually impaired Web travellers*. *Web Engineering, Proceedings, 2004*. **3140**: p. 445-458.
7. Buyukkokten, O., et al., *Efficient Web browsing on handheld devices using page and form summarization*. *ACM-Transactions-on-Information-Systems*. 20 (1) Jan 2002, p.82-115. il. tbls. refs., 2002.
8. Buyukkokten, O., H. Garcia-Molina, and A. Paepcke. *Accordion summarization for end-game browsing on PDAs and cellular phones*. in *Proceedings of CHI 2001. 31 March 5 April 2001 Seattle, WA, USA*. 2001: ACM, New York, NY, USA.
9. Chen, J., et al., *Function-based object model towards website adaptation*  
<http://doi.acm.org/10.1145/371920.372161> in *Proceedings of the 10th international conference on World Wide Web 2001* ACM Press: Hong Kong, Hong Kong p. 587-596
10. Kaasinen, E., et al., *Two approaches to bringing Internet services to WAP devices*  
[http://dx.doi.org/10.1016/S1389-1286\(00\)00041-4](http://dx.doi.org/10.1016/S1389-1286(00)00041-4) in *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking 2000* North-Holland Publishing Co.: Amsterdam, The Netherlands p. 231-246
11. Chakrabarti, S., *Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction*  
<http://doi.acm.org/10.1145/371920.372054> in *Proceedings of the 10th international conference on World Wide Web 2001* ACM Press: Hong Kong, Hong Kong p. 211-220
12. Embley, D.W., Y. Jiang, and Y.-K. Ng, *Record-boundary discovery in Web documents*  
<http://doi.acm.org/10.1145/304182.304223> in *Proceedings of the 1999 ACM SIGMOD international conference on Management of data 1999* ACM Press, p. 467-478

13. Cai, D., et al. *Extracting Content Structure for Web Pages based on Visual Representation*. in *the Fifth Asia Pacific Web Conference (APWeb2003)*. 2003. Xian, China.
14. Gu, X.-D., et al., *Visual Based Content Understanding towards Web Adaptation* in *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems 2002* Springer-Verlag. p. 164-173
15. Zajicek, M. and S. Hall. *Solutions for elderly visually impaired people using the Internet*. in *Proceedings of People and Computers XIV Usability or Else. Sept. 2000 Sunderland, UK*. 2000: Springer-Verlag London, London, UK.
16. Takagi, H., et al. *Site-wide annotation: reconstructing existing pages to be accessible*. in *ASSETS 2002. Proceedings of the Fifth International ACM SIGCAPH Conference on Assistive Technologies. 8 10 July 2002 Edinburgh, UK*. 2002: ACM, New York, NY, USA.
17. Ramakrishnan, I.V., A. Stent, and G. Yang, *Hearsay: enabling audio browsing on hypertext content* in *Proceedings of the 13th international conference on WWW 2004* ACM Press, p. 80-89
18. Shepherd, M., C. Watters, and A. Kennedy, *Cybergenre: Automatic Identification of Home Pages on the Web*. *Journal of Web Engineering*, 2004. 3(3 & 4): p. 236-251.
19. Dewdney, N., C. VanEss-Dykema, and R. MacMillan, *The Form is the Substance: Classification of Genre in Text.*, in *ACL 2001 Workshop on HLT and KM*. 2001.
20. Bar-Yossef, Z. and S. Rajagopalan. *Template detection via data mining and its applications*. in *Proceedings of the eleventh international conference on World Wide Web*. 2002. Honolulu, Hawaii: ACM.
21. Parmanto, B., et al., *Transcoding Biomedical Information Resources for Mobile Handhelds* in *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 6 - Volume 06 2005* IEEE Computer Society. p. 158.1
22. NTIA, *A Nation Online: How Americans Are Expanding Their Use Of The Internet*. 2002, NTIA and the Economics and Statistics Administration: Washington, DC.
23. Hackett, S. and B. Parmanto, *Usability Study of AcceSS Transcoding Gateway*. *Journal of Visual Impairment and Blindness*, 2006. (In press).
24. Souza, R., H. Manning, and M. Dorsey, *Designing Accessible Sites Now*, in *Forrester Research Techstrategy Report*. 2001, Forrester Research.