Assistive Browser for Conducting Web Transactions

Jalal Mahmud, Yevgen Borodin, and I.V. Ramakrishnan

Department of Computer Science
Stony Brook University
Stony Brook, NY 11794, USA
{jmahmud, borodin, ram}@cs.sunysb.edu

ABSTRACT
People with visual impairments use screen readers to browse the Web. Sequential processing of web pages by screen readers causes information overload, making web browsing time-consuming and strenuous. These problems are further exacerbated in web transactions (*e.g.:* online shopping), which involve multiple steps spanning several web pages. In this paper we present a lightweight approach for doing Web transactions using non-visual modalities. We describe how analysis of context surrounding the link coupled with a shallow knowledge-base with patterns and keywords can help identify various concepts (*e.g.:* "add to cart", "item description", etc.) that are important in web transactions. Our preliminary results show promise that presenting such concepts to the users can reduce information overload and improve their overall browsing experience.

ACM Classification Keywords
H.3.3 Information Systems: Search and Retrieval; H.5.2 Information Interfaces and Presentation: User Interfaces

General Terms
Algorithms, Design, Human Factors, Experimentation

Author Keywords
Context, Web Transaction, Shallow Knowledge-base, Partitioning, Segments

INTRODUCTION
People suffering from vision loss have to use screen-readers, such as Freedom Scientific's JAWS and GW Micro's Windows Eyes, to access the Web. Screen-readers typically narrate Web pages sequentially, while allowing the users to move backward and forward in Web pages. Unfortunately, the aural interfaces of screen-readers do not offer much of content filtering, making browsing very time consuming and causing significant information overload. This is especially the case when doing Web transactions, (*e.g.:* shopping, registrations, bill payments, etc.) that typically span multiple pages.

Non-visual web transactions can be facilitated by giving the users a way to quickly access the most relevant concepts at any step of the transaction. In our previous work [5], we used a process model to direct Web transactions. The model was represented by a finite state automaton where each state corresponded to a web page with concepts, and the edges were the actions leading to those states. When a transaction was in a particular state, the most relevant concepts would be identified, extracted, and presented to the user.

The main difficulty of this approach was that it required considerable manual effort to collect and label data for both learning the process model and training classifiers for concept extraction. Furthermore, it worked well only in the domain for which it was trained; a new model would have to be built for different domains, *e.g.:* "banking", "bill payment", etc. Although, automata-learning tools can be used to build a new model automatically, new transaction traces would have to be collected and labeled again in order to learn both the new process model and concept classifiers. All of these made domain customization and switching between domains rather difficult.

In our subsequent research we tried to reduce the amount of training and labeling necessary to customize the transaction system for a new domain. In the process we observed that most concepts can be consistently divided into 3 broad categories: (1) those that can be detected using keywords, *e.g.:* "Add To Cart" (Table 1); (2) those that can be captured by simple patterns or rules, *e.g.:* "Item Taxonomy" (Table 2); And, (3) those that have a variation of content and presentation styles across different Web sites and cannot be detected with keywords and patterns alone, *e.g.:* "Item Detail" (Table 3). See Figure 1 for illustration.

While, in most cases, the concepts in the first two categories can be easily detected and labeled, one has to keep the state information to be able to distinguish between the concepts in the 3rd category. For example, how is "item list" different from "search results"? However, instead of maintaining the entire automaton, keeping only some information from the previous state

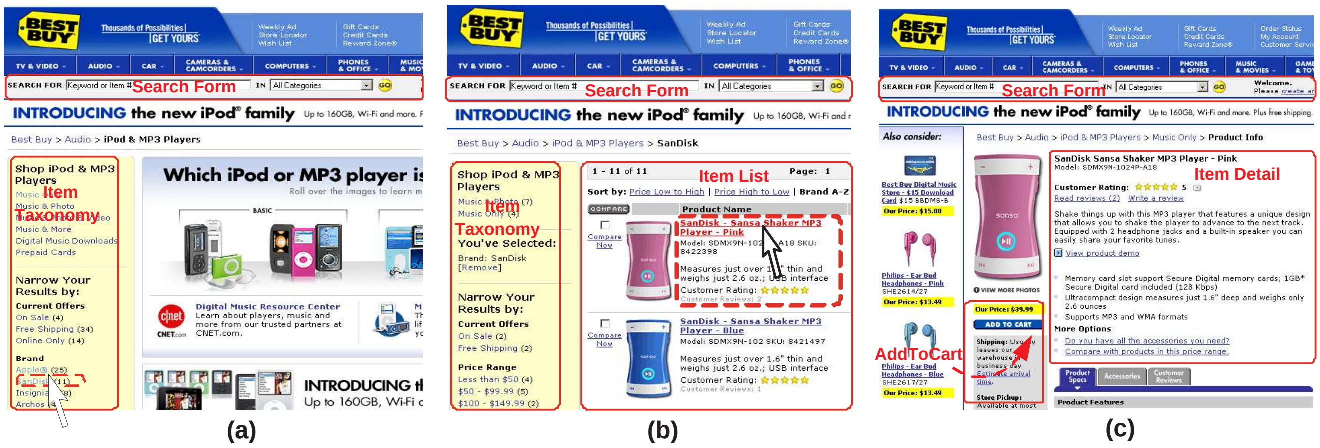

Figure 1. Relevant Concept Segment Identification by Context Analysis

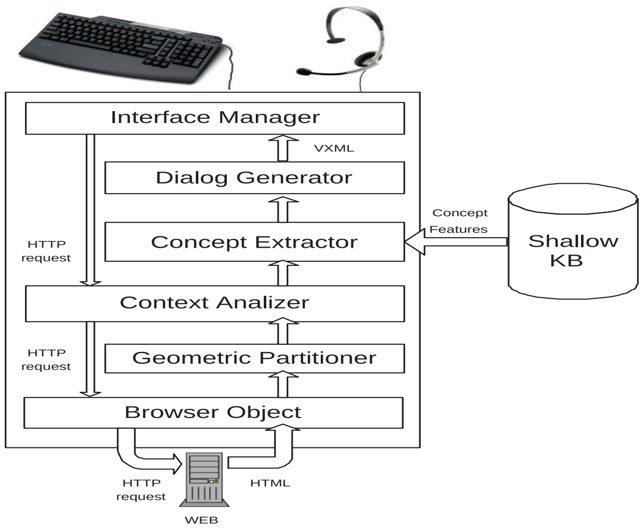

Figure 2. System Architecture

is often sufficient to obtain the same results! When a link is followed from one page to another, the context of the link can be used to identify such concept on the following page. Figure 1 shows how by using the words of the "SanDisk" link in part (a), we can detect a relevant concept in part (b) and label it as "Item List" after detecting a pattern in its presentation style; then, the context of the selected player, surrounded by the dotted line in part (b), can be used to find the "Item Detail" concept in Figure 1 (c). At each step of the transaction only the relevant concepts are presented to the users. Some of these concepts have an associated operation that the users can act on and proceed to the next step of the transaction.

In this paper we present a lightweight approach for doing Web transactions using non-visual modalities. The important aspects of this approach are that (1) it uses a shallow knowledge-base with keywords and patterns to identify the first two types of concepts; and, (2) it leverages our context-directed browsing technique [3] to identify the concepts that are otherwise difficult to detect, see Table 3. Another contribution of this work is the reduction of human effort that is required for labeling data and training classifiers in other approaches. The relevance model described in [3] uses generic features (*e.g.:* word matches) to identify the most relevant segment on following a link and does not require supervised training for each content domain. Thus, in contrast to the model-directed transactions, our current approach does not use automata learning and does not require retraining the concept extractor for new domains. In addition, the shallow KB is easy to update and maintain and can be used across multiple domains. Finally, our preliminary experiments indicate that the context-based approach for web transactions achieves equal or better performance than that of our previous process-directed approach. The remainder of the paper gives more details about the approach and describes our preliminary results and evaluation.

SYSTEM ARCHITECTURE

Our Web transactional system is based on the HearSay-CSurf[1] non-visual Web browser [3]. The architecture of the system is shown in Figure 2.

Users communicate with the system through the **Interface Manager**, which uses VoiceXML dialogs to interact with the users. The interface manager presents the segment of the page with the most relevant concept to the user. It also provides an easy way to navigate between segments containing other concepts; all concepts can be accessed through a menu or a shortcut can be used to skip from segment to segment.

The **Browser Object** module downloads Web content every time the user requests a new page to be retrieved. The module is built on top of the Mozilla Web Browser coupled with the JREX Java API wrapper.

[1] *www.cs.sunysb.edu/∼hearsay*

Concepts	Keywords in KB
Shopping Cart	cart, shopping cart, basket, total
Add To Cart	add to cart, buy, quantity, price, qty
Edit Cart	update cart, edit cart
Continue Shopping	continue, continue shopping
Checkout	check out
Search Form	search, find, go

Table 1. Simple Concepts Identified by Keywords

Geometric Partitioner uses the layout and alignment of the web page to segment it into distinct sections [3]. The algorithm uses the observation that semantically related items share similar geometric alignment and exhibit spatial locality. For example, in Figure 1 (b), all of the the listed MP3 players appear to be in the same segment.

Context Analyzer is called twice for each web page access. When the user follows a link, the module collects the context of the link, which includes the words of the link, as well as the text around the link. When a new Web page is retrieved, the module executes our algorithm [3] to locate the section of the page, which is estimated to be most relevant with respect to the context of the followed link (See Figure 1).

Shallow KB contains the features of concepts that are commonly used in online transactions, such as "Search-Form" or "Add to cart". (See Tables 1, 2).

Concept Extractor uses the KB and the context analyzer to identify semantic concepts in the web page. The most relevant segment identified by context analyzer often contains concepts, such as "Item Detail", that are difficult to identify only using the KB.

The **Dialog Generator** module generates VoiceXML dialogs for page navigation. The dialogs are then delivered to the Interface Manager, and, once the user chooses an action, the process is repeated. The interface of the system is similar to that described in [3].

TECHNICAL APPROACH

We observe that some of the concepts used in online transactions on most web sites can be identified by keywords (Table 1). For example, the section confining the "Add To Cart" concept often contains the "add to cart" trigram. Some other concepts are presented with a consistent presentation style or pattern that can be captured by regular expressions in Table 2. For example, the semantic concept "Item Taxonomy" is, in most cases, presented by a sequence of links.

Using these observations, we constructed a **shallow knowledge-base** (KB) and, after having inspected a number of web sites, we manually populated the KB with keywords, as well as their combinations, in Table 1, and simple patterns in Table 2. These features were then used to identify and label the concepts that appear on the web page.

Concept	Patterns in KB
Search Form	*editbox.submitbutton*, *combobox.editbox*
Item Taxonomy	$(\text{link})^+$, $(\text{link, text})^+$, $(\text{link, img})^+$

Table 2. Simple Concepts Identified by Patterns

Concept	Description
Search Result	List containing search result
Item List	List of similar items
Item Detail	Description of an item

Table 3. Complex Concepts Identified by Context

Once a Web page has been segmented by the geometric partitioner, the concept extractor looks for concepts in each of the page segments. It then uses the KB to match concept features (keywords and patterns) to the content of the segments and label them accordingly.

We observed that some of the concepts, such as the ones presented in Table 3, do not have simple features consistent across different Web sites. For example, in Figure 1 (b), the segment containing the list of MP3 players represents the semantic concept "Item List". Similarly, the concept "Item Detail" in Figure 1 (c) cannot be easily identified using a Shallow KB. At the same time, their presentation style may be completely different across web sites. Since there are no simple features to distinguish between these concepts, training a classifier can be very challenging as well as the the amount of effort necessary to collect and label the training data.

Instead, we chose a simpler approach that uses generic domain-independent features to find complex concepts. The presence of such concepts can be often identified by capturing contextual information of the link followed from the previous page, as described in [3]. This context can then be used to identify the most relevant segment. Other features, such as the presence of a pattern, helps determine the concept type on the next web page. Figure 1 shows three consecutive web pages with arrows indicating the followed links and dotted line boxes encompassing the context of the links. Following the link to SanDisk MP3 player in part (a) helps find the "Item List" in part (b), and following the link to one of these players in part (b), helps identify the "Item Detail" in part (c) of Figure 1.

The model that uses contextual information to identify the most relevant segment in a web page is based on generic features. It can do simple keyword matching or use a more sophisticated statistical model such as SVM (Support Vector Machine) as described in [3]. Nevertheless, this approach still remains domain independent and the model needs to be trained only once.

PERFORMANCE

We evaluated the performance of the transactional system using 12 Web sites in 3 domains: books, electronics, and office supplies. In this preliminary study we used 12 sighted evaluators. More work on the usability of the

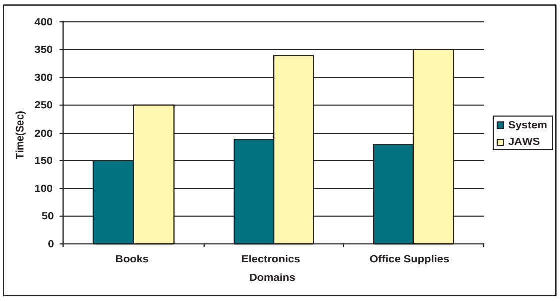

Figure 3. System Performance

system is required before it can be adequately evaluated by blind people.

Each of the evaluators was instructed to conduct 3 transactions. The screen was disabled and the evaluators had to interact with the system using headphones and a keyboard. For baseline comparison, the evaluators were also asked to conduct the same transactions with JAWS, the state-of-the-art screen reader. Evaluators were trained on how to use both systems. We calculated the mean time taken by all evaluators to complete the transactions with our system and JAWS. Figure 3 shows the comparative performance between the two systems in each of the three domains. Observe that our system compares quite favorably with the best-of-breed screen reader.

Most of the evaluators had no trouble finding the desired concepts. However, in 3 out of 36 transactions, they had to switch to the regular screen-reader mode to be able to find the information they needed. In the course of the experiments 83.3% of concepts were identified correctly. Some imperfections of the segmentation and relevant segment identification algorithms also affected the accuracy of concept extraction. However, the quantitative performance (*i.e.* accuracy and transaction completion time) of our new transaction system is comparable to that of our previous model-directed system [5]. Overall, 11 out of 12 of the evaluators felt that the system was adequate to perform transactions.

RELATED WORK

Considerable work has been done to facilitate non-visual web browsing, *e.g.:* [6, 4, 1]. However, it did not address the problem of information overload in web transactions using non-visual modalities. Since Web transactions involve web forms, research on form understanding is also relevant. Some ideas on form understanding appear in [7, 2]. Our transaction system supports filling web forms. However, more work remains to be done to improve accessibility of the forms.

In our prior research on Web transactions [5], we built a process model and trained a classifier for concept extraction using supervised learning techniques. It had inherent drawbacks, specifically, it required considerable manual effort to collect and label the training data and it was not scalable across domains, requiring a new process model to be constructed for new domains. In this paper we address these problems and describe how contextual analysis technique coupled with a shallow knowledge-base can be used to build a lightweight system to facilitate Web transactions. A shallow knowledge-base is easy to maintain and can be updated continuously as content domain changes.

CONCLUSION AND FUTURE WORK

In this paper, we presented a novel approach centered on contextual browsing technique coupled with a shallow knowledge-base to combat the information overload problem in online transactions using non-visual interaction modalities. Our preliminary results are encouraging, but work still has to be done to make the system more usable.

We identify several areas of future research. We would like to make use of linguistic analysis of web forms to present them to the users in a more comprehensive way. We are currently investigating how to construct a knowledge-base automatically and updating the knowledge-base as users perform more and more transactions.

Finally, extensive evaluations remain to be done with blind individuals who are trained to use technology at the Helen Keller Services for the Blind.

REFERENCES

1. S. Harper and N. Patel. Gist summaries for visually impaired surfers. In *Assets '05: Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, pages 90–97, 2005.
2. N. Kushmerick. Learning to invoke web forms. In *Intl. Conf. on Ontologies, Databases, and Applications of Semantics (ODBASE)*, 2003.
3. J. Mahmud, Y. Borodin, and I. V. Ramakrishnan. Csurf: A context-directed non-visual web-browser. In *WWW '07*, 2007.
4. J. T. Richards and V. L. Hanson. Web accessibility: a broader view. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 72–79, 2004.
5. Z. Sun, J. Mahmud, S. Mukherjee, and I. V. Ramakrishnan. Model-directed web transactions under constrained modalities. In *WWW '06*, pages 447–456, 2006.
6. H. Takagi, C. Asakawa, K. Fukuda, and J. Maeda. Site-wide annotation: Reconstructing existing pages to be accessible. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2002.
7. Z. Zhang, B. He, and K. C.-C. Chang. Understanding web query interfaces: Best-effort parsing with hidden syntax. In *ACM Conf. on Management of Data (SIGMOD)*, 2004.