Web Transactions on Handhelds with Less Tears

Jalal Mahmud   Zan Sun   Saikat Mukherjee[*]   I.V. Ramakrishnan

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

ABSTRACT

Online transactions (e.g. buying a book) typically involve a number of steps spanning several pages. Conducting such transactions under constrained interaction modalities as exemplified by small screen mobile device is a strenuous, fatigue-inducing activity. But usually one needs to browse only a small fragment of a Web page to perform a transactional step (*e.g.* choosing an item from a search results list). We exploit this observation to develop an automata-based process model for online transactions that succinctly captures the transactional steps delivering only the "relevant" page fragments at each step. We realize this model by coupling techniques from content analysis of Web documents, automata learning and statistical classification. This process model and associated techniques have been incorporated into Guide-O-Mobile, a prototype system that facilitates online transactions with limited display size mobile device . Performance of Guide-O-Mobile and its user experience is reported.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Algorithms, Human Factors

Keywords

Web transaction, Content Adaption, Assisstive Device

1. INTRODUCTION

The World Wide Web has become the dominant medium for doing e-commerce. The volume of goods bought from online stores continues to grow dramatically. A Web *transaction* such as buying a CD player from an online store involves a number of user steps spanning several Web pages. As an illustration let us examine some common steps for buying a CD player from Best Buy (`http://www.bestbuy.com`). To begin with the user fills out the search form with "electronics" as the category and "CD Player" as the item. The

---

[*]Siemens Corporate Research, Princeton, NJ 08540, saikat.mukherjee@siemens.com

search result generated in response is shown in Figure 1(a). The user selects the very first item in the result list which leads to the page in Figure 1(b) containing product details. To complete the transaction the user adds this selected item to the cart which leads to the page in Figure 1(c) wherein the user selects checkout. Observe that there are two essential components to a Web transaction: (i) locating the relevant content, such as a search form or the desired item in a Web page, and (ii) performing a sequence of steps, such as filling out a search form, selecting an item from the search list and doing a checkout. For completing a transaction these steps usually span several pages.

Online transactions such as the one described above are usually performed with graphical Web browsers. The primary mode of interaction with graphical browsers is visual, an intrinsically spatial modality. Hence, users can quickly scan through the rich engaging content in Web pages scripted for e-commerce and locate the objects of interest quite easily. Moreover, the spatial organization of content in these pages helps users comprehend the sequence of steps necessary to complete a transaction.

Now consider scenarios when the interaction media has small displays (*e.g.*, mobile handhelds). Small displays offer narrow interaction bandwidths making it cumbersome and tedious to get to the pertinent content in a page. This problem is further exacerbated when such an interaction spans several pages as in an online transaction. In particular the loss of spatially organized content makes it difficult for users to comprehend the sequence of transactional steps. While content summarization can compensate somewhat for this loss, it alone is inadequate for handling the information overload that the user faces.

Thus, there is a need for developing techniques to facilitate Web transactions under constrained interaction modalities that are far less cumbersome to do than current approaches. In this paper we address this problem and describe our solution.

We capture the two aspects of a transaction, namely its operation sequence and content identification by a *process model* and an *ontology* respectively. The ontology describes the set of *semantic concepts* occurring in Web pages, which are considered essential for conducting Web transactions in a particular domain. The circled elements in Figure 1 are examples of such concepts. The process model is a deterministic finite automata (DFA) that captures the set of transactional sequences. Each state, representing an atomic operation in a transaction, is associated with a set of semantic concepts drawn from the ontology. When the model makes
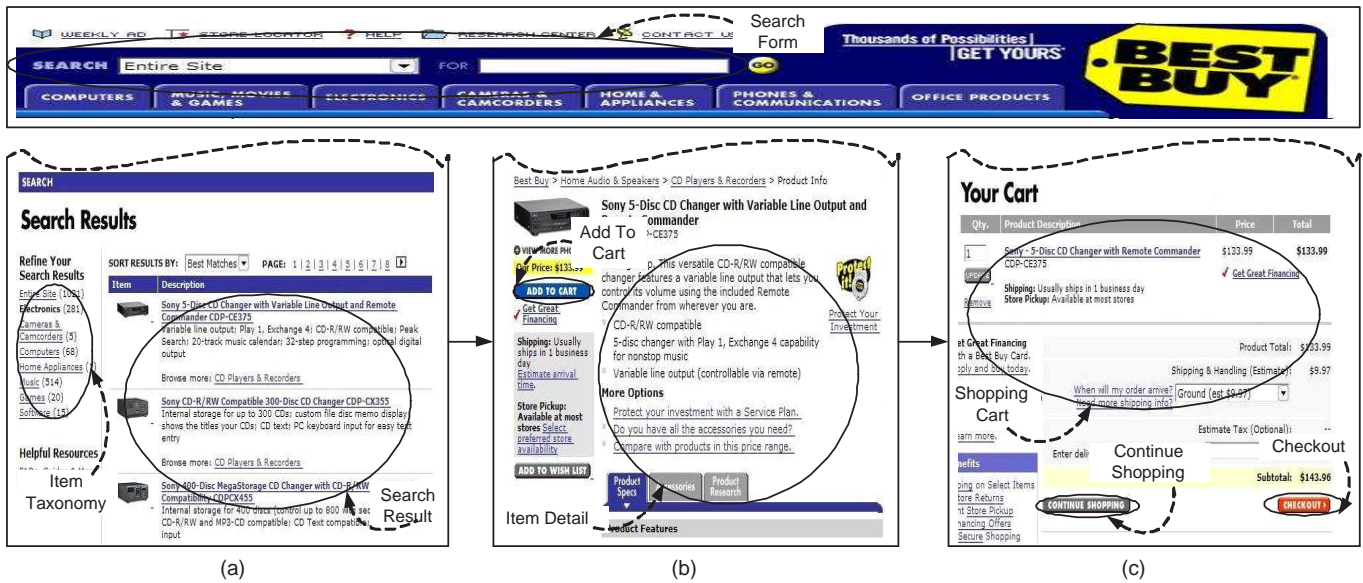
Figure 1: A Web Transaction Example

a transition to a state during the course of a transaction, a Web page is provided to the state as an input. If the concepts associated with the state are present in the page, then they alone are identified and presented to the user. For instance, if the page shown in Figure 1(a) is given as the input to a state associated with the concepts "Item Taxonomy" and "Search Result" only the two circled items in the figure will be identified and presented to the user. Since transactions are essentially interactive, we associate each concept with a user operation, *e.g.*, the *submit_searchform* operation with the "Search Form" concept. Each such operation results in a state transition and a *sequence* of operations constitutes a Web transaction. Observe that coupling content semantics with model-directed navigation can overcome the information overload problem by delivering relevant content at every step of the transaction.

Our approach to semantic concept identification in Web pages is built upon our previous work [18] where we had proposed a technique for partitioning a page's content into segments constituting concept instances. It uses a combination of structural analysis of the page and machine learning. We adopt it for the problem addressed in this paper and enhance its learning component to produce more robust statistical models of semantic concepts. It is noteworthy to point out that the use of content semantics, as opposed to syntax based techniques for identifying concepts makes it more robust to structural variations in the pages and scalable over Web sources that share similar content semantics.

We also use automata learning techniques (see [19] for a survey) to construct process models from training sequences generated from real Web transactions. The use of process models for online transactions bears similarities to the emerging Web services paradigm for conducting automated transactions. But the fundamental difference is that our technique works on *Web pages* instead of services exposed by a service provider (see Section 6 for detailed comparison).

The rest of this paper is organized as follows: In Section 2, we describe a user scenario and the architecture of Guide-

O-Mobile, a prototype system that we implemented based on the techniques detailed in this paper. In Section 3 we formalize the process model and describe its implementation using DFA learning techniques. Content analysis techniques for semantic concept identification appear in Section 4. Quantitative experimental evaluation of the Guide-O-Mobile configurations as well as qualitative user experience appear in Section 5. Related work appears in Section 6 and we conclude the paper in Section 7.

2. THE GUIDE-O-MOBILE SYSTEM

2.1 Use Scenario

Alice, a college senior, frequently uses her handheld running Guide-O-Mobile for browsing and shopping online. Recently her CD player broke down and she needs to replace it with a new one. She had seen Best Buy offering good deals recently and so she decides to buy the new CD player from Best Buy. To begin, she enters Best Buy's URL into the Guide-O-Mobile interface on her handheld. After retrieving the home page Guide-O-Mobile analyzes this page, extracts the two concepts in it, namely " Item Taxonomy" and "Search Form" (the circled items on the left and top respectively in Figure 1(a)) and shows the concept names to Alice. Alice chooses " Search Form" and in response Guide-O-Mobile displays the whole search form extracted from the original Web page. Alice chooses "Electronics" and inputs "CD Player" as she usually does with a Web page browser. Guide-O-Mobile then submits the search form filled with these two parameters that results in fetching the page containing the search results shown in Figure 1(a). Guide-O-Mobile extracts the "Search Result " concept and, after Alice chooses it, displays the brief description associated with each CD player in this list.

Alice clicks on the first item causing Guide-O-Mobile to follow the link associated with the 1st player (CDP-CE375) in the list to the page containing a detailed description of this player (Figure 1(b)). In this page Guide-O-Mobile ex-
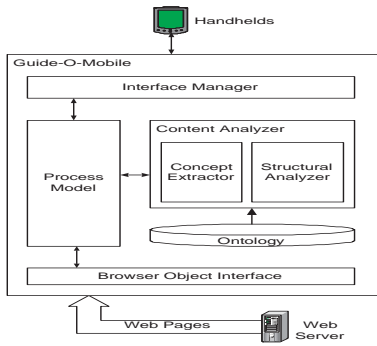
Figure 2: Guide-O-Mobile System Architecture

Concept	Operation
Shopping Cart	view_shoppingcart
Add To Cart	add_to_cart
Edit Cart	update_cart
Continue Shopping	continue_shopping
Checkout	check_out
Search Form	submit_searchform
Search Result	item_select
Item List	item_select
Item Taxonomy	select_item_category
Item Detail	show_item_detail

Table 1: Concepts in Ontology.

Figure 3: Process Model Example

tracts three concepts, namely "Search Form", "Item Detail" and "Add To Cart". Since Alice wishes to see the product details, she chooses the "Item Detail". Guide-O-Mobile displays the detailed description of the CD player she picked earlier.

At the end Alice chooses the "Add To Cart" concept and Guide-O-Mobile follows the link labeled *Add To Cart* in Figure 1(b) to the page shown in Figure 1(c). In this page Guide-O-Mobile extracts the concepts of "Search Form", "Shopping Cart", "Checkout" and "Continue Shopping". When presented with these choices Alice chooses "Checkout".

To complete the transaction Alice must provide credit card information upon checkout. Its details have been omitted as they are quite similar to the form filling step described in the first step of the scenario.

The use scenario above illustrates how a user conducts Web transactions with Guide-O-Mobile using mobile handheld device where graphical interaction with a standard browser is not feasible because of their small screen size. Instead of displaying the entire Web page to the user, Guide-O-Mobile only displays the extracted concepts of a page. Since usually these are few in number with small content, they can be browsed more easily even under small screen size constraint.

2.2 Guide-O-Mobile Architecture

The architecture of Guide-O-Mobile is shown in Figure 2. In this configuration the *Interface Manager* automatically generates a DHTML page to display the extracted concepts.

The *Content Analyzer* partitions an input Web page into homogeneous segments containing semantically related content elements (such as "Item Taxonomy", "Search Result" in Figure 1) and heterogeneous content segments (such as "Item Detail" in the figure). Using an underlying ontology of concepts present in Web pages it classifies these segments to the concepts in the ontology and labels them with their names. (Section 4 describes the classification technique.) Table 1 shows concept names in an ontology. Associated with each concept is an operation as shown in the table. When the user selects a concept in a state, the corresponding operation is invoked. The ontology also includes information for classification of page segments to concepts.

The *Browser Object Interface* fetches pages from Web servers. Special features include automatic form fill-outs and retrieval of pages pointed to by navigable links, which requires execution of javascript.

The *Process Model* orchestrates all the actions that take
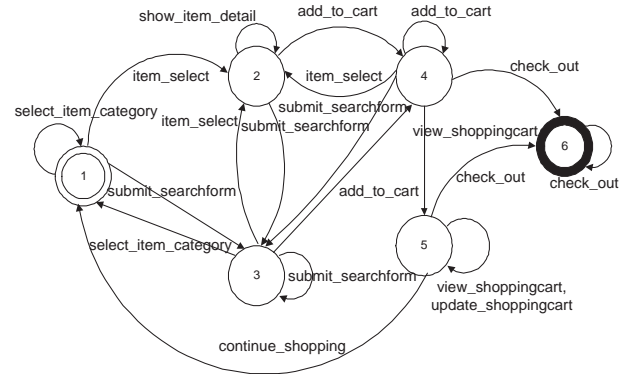
place in Guide-O-Mobile. In a state, it takes a URL as the input, calls the Browser Object Interface to fetch the page and the Content Analyzer to extract from the page the concepts that it expects. The extracted concepts are organized as a concept tree that is dispatched by the Process Model to the Interface Manager for presentation to the user. When the user selects a concept, it is sent to the Process Model as an operation on the concept. A state transition based on the operation is made and the cycle repeats.

The architecture described is in essence the runtime engine that drives the Guide-O-Mobile system. The Process Model and the Concept Extractor in the engine are learned a priori in the learning component.

3. PROCESS MODEL

Formally, our process model is defined as follows: Let $C = \{c_0, c_1, \ldots\}$ be a set of concepts, and $I(c)$ denotes the set c of concept instances. Let $Q = \{q_0, q_1, \ldots\}$ be a set of states. With every state q_i we associate a set $S_i \subseteq C$ of concepts. Let $O = \{o_0, o_1, \ldots\}$ be a set of operations. An operation o_i can take parameters. A transition δ is a function $Q \times O \rightarrow Q$, and a concept operation ρ is also a function $C \rightarrow O$. Operations label transitions, *i.e.*, if $\delta(q_i, o) = q_j$ then o is the label on this transition. An operation $o = \rho(c)$ is enabled in state q_i whenever the user selects an instance of $c \in S_i$ and when it is enabled a transition is made from q_i to state $q_j = \delta(q_i, o)$.

Technically a concept instance is the occurrence of a concept in a Web page. For example, the circled items in Figure 1 are all concept instances. But for brevity we choose not to make this distinction explicitly and use concepts and concept instances interchangeably when referring to content
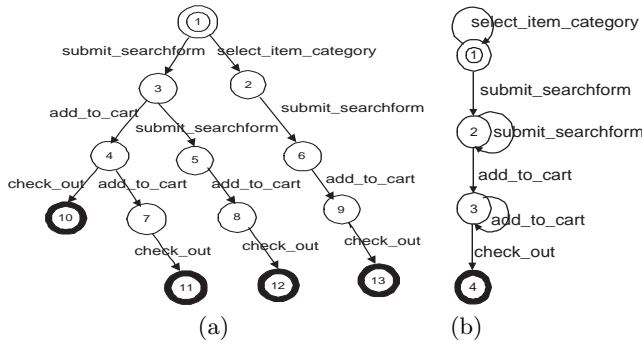
Figure 4: (a) A Prefix Automata. (b) Learned DFA

segments in Web pages.

Figure 3 illustrates a process model. The concepts associated with state q_1 are "Item Taxonomy", "Item List", and "Search Form". This means that if these concept instances are present in the Web page given to q_1 as its input, they will be extracted and presented to the user. User can select any of these concepts. When the user selects the "Search Form" concept he is required to supply the form input upon which the *submit_searchform* operation is invoked. This amounts to submitting the form with the user-supplied form input. A Web page consisting of the search results is generated and a transition is made to q_3. As discussed in the use scenario in Section 2 we have omitted the payment steps following checkout. Hence q_6 which is entered upon a *check_out* operation is deemed as the final state.

Process Model Learning

We built the process model using DFA learning techniques, a thoroughly researched topic (see Section 6 for related work). In the DFA learning problem the training set consists of two sets of example strings, one labeled positive and the other negative. Only strings in the positive set are in the DFA's language. The objective is to construct a DFA that is consistent with respect to these two sets, *i.e.*, it should accept strings in the positive set while rejecting those in the negative set. We adapted the heuristic in [20] for learning our process model, the choice being mainly dictated by its simplicity and low complexity.

The training sequences we used for learning the process model consist of strings whose elements are operations on concepts. The sequence $<$ *submit_searchform, item_select, add_to_cart, check_out* $>$ is one such example. These training sequences are (manually)labeled "completed" and "not completed". The positive example set $(S+)$ consists of sequences labeled "completed" while the negative example set $(S-)$ consists of those labeled "not completed".

We first construct a prefix tree automata as shown in 4(a) using only the examples in the positive set $S+$. In 4(a), the sequence of operations along each root-to-leaf path constitutes a string in $S+$. For this example the negative set $S-$ consists of the strings: $\{<$ *check_out* $>, <$ *submit_searchform, add_to_cart* $>, <$ *submit_searchform, check_out* $>, <$ *select_item _category, add_to_cart, check_out* $>\}$. The prefix of every string in $S+$ is associated with a unique state in the prefix tree. The prefixes are ordered and each state in the prefix tree automata is numbered by the position of its corresponding prefix string in this lexicographic order. Next

we generalize the prefix tree automata by state merging. We choose state pairs (i, j), $i < j$ as candidates for merging. The candidate pair (i, j) is merged if it results in a consistent automata. For example, merging the pair $(1,2)$ is consistent whereas $(3,4)$ is not merged as the resulting automata will accept the string $<$ *submit_searchform, check_out* $>$ in $S-$. The DFA that results upon termination of this merging process on the above example set is shown in Figure 4(b).

Since we do at most Q^2 state mergings, where Q is the cardinality of $S+$, the time complexity is polynomial. In Section 5 experimental evaluation of the performance of the learned process model is presented.

4. CONTENT ANALYSIS

Herein we describe the *content analyzer* module that extracts the relevant concepts. In a nutshell this is achieved by partitioning a Web page into segments of semantically related items and classifying them against concepts in the ontology. Below we provide an overview.

The Approach

It is based on our previous work on learning-based semantic analysis of Web content [18]. Briefly, the technique rests on three key steps: (i) inferring the logical structure of a Web page via structural analysis of its content, (ii) learning statistical models of semantic concepts using light-weight features extracted from both the content as well as its logical structure in a set of training Web pages, and (iii) applying these models on the logical structures of new Web pages to automatically identify concept instances.

- Structural Analysis: Structural analysis (see [18] for details) is based upon the observation that semantically related items in content-rich Web pages exhibit consistency in presentation style and spatial locality. Exploiting this observation, a pattern mining algorithm working bottom-up on the DOM tree of a Web page aggregates related content in subtrees. Briefly, the algorithm initially assigns *types*, reflecting similarities in structural presentation, to leaf nodes in the DOM tree and subsequently restructures the tree bottom-up using pattern mining on type sequences. The DOM tree fragment for the page in Figure 1(a) is shown in 5(a). The type of a leaf node is the concatenation of HTML tags on the root-to-leaf path and that of an internal node is composed from the types of its child nodes. In the restructured tree, known also as the *partition tree*, there are three classes of internal nodes: (i) *group* - which encapsulates repeating patterns in its immediate children type sequence, (ii) *pattern* - which captures each individual occurrence of the repeat, or (iii) *block* - when it is neither group nor pattern. Intuitively the subtree of a group node denotes homogenous content consisting of semantically related items. For example, observe how all the items in the search results list in Figure 1(a) is rooted under the group node in the partition tree. The leaf nodes of the partition tree correspond to the leaf nodes in the original DOM tree and have content associated with them. The partition tree resulting from structural analysis of the DOM in Figure 5(a) is shown Figure 5(b). The partition tree represents the logical organization of the page content.

- Feature Extraction: The statistical concept models are based upon features extracted from the content of the partition trees. Given a partition tree node p, $n_{f_i,p}$ denotes
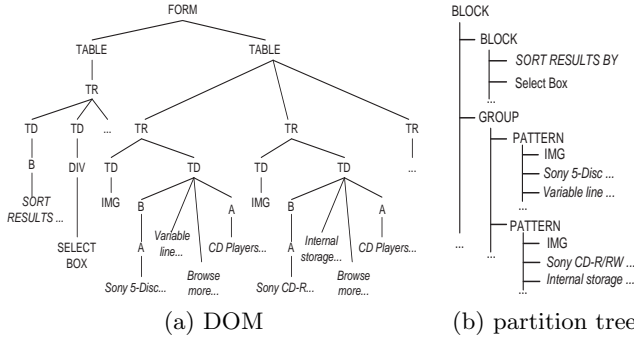
(a) DOM   (b) partition tree

Figure 5: Structural Analysis of the Page in Fig 1(a)

the frequency of occurrence of feature f_i in p. We use three different types of features in the analysis:

(i) *Word features* - These are features drawn from the text encapsulated within a partition tree node. For a leaf node in the partition tree, word features are drawn from its own text while for an internal partition tree node, the words present in all the leaves within the subtree rooted at it are aggregated. Stop words are ignored in both cases. $n_{f_i,p}$ is the number of times f_i occurs in the text of p.

(ii) *p-gram features* - These are features representing the visual presentation of content. In content-rich Web pages, it is often the case that the presentation of a semantic concept exhibits similarity across sites. For instance, in Figure 1(b), each item is presented as a link with the item name, followed by a short text description, and ending with miscellaneous text information. Similar visual presentation can also be found on other sites. A p-gram feature captures these presentation similarities. The basic p-gram features are link, text, and image found in leaf partition tree nodes. Recall that, during structural analysis, pattern nodes aggregate every individual repeat in a type sequence. Since repeats are typically associated with similar visual presentation, complex p-gram features are constructed only at pattern nodes by concatenating the p-gram features of their immediate children nodes. Internal nodes aggregate basic and possibly complex p-grams from the subtrees rooted at them. Like word features, $n_{f_i,p}$ is the number of times f_i occurs in the subtree rooted at p. For instance, in the left tree of Figure 6, the p-gram feature at the pattern node labeled "P" is $< text \cdot link >$.

(iii) *t-gram features* - While p-gram features capture the visual presentation, t-gram features represent the structure of the partition tree. Recall that internal partition tree nodes can be either group, pattern, or block while link, text, and image are the different classes of leaf nodes. The structural arrangement of these classes of nodes is also a concept characteristic and this is what is captured by t-gram features. Given a partition tree node with N nodes in its subtree, the complete structural arrangement within the node can be described in terms of a set of subtrees of k $(2 \leq k \leq N)$ nodes where each subtree is an arrangement of group, pattern, block, link, text, or image type nodes. Since enumerating all these subtrees has exponential complexity, we restrict our analysis to subtrees of 2 nodes. When $k = 2$ the t-gram is essentially a parent-child feature. For instance, in Figure 6, when $k = 2$ the t-gram feature space of the left tree is $\{< G, P >, < P, Text >, < P, Link >\}$, and the right tree
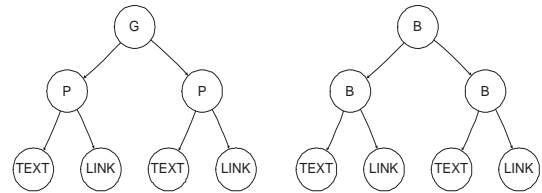
Figure 6: t-gram Features

is $\{< B, B >, < B, Text >, < B, Link >\}$, where G and B are labels of group and block nodes respectively.

- Concept Identification: A concept model consists of two components: (i) a probability distribution on the frequency of occurrence of the word, p-gram, and t-gram features, and (ii) a probability distribution on the number of nodes present in the entire subtree of a partition tree node. A collection of partition trees whose nodes are (manually) labeled as concept instances serve as training set for learning the parameters of these distributions.

A maximum likelihood approach is used to model the distribution of a feature in a concept. Given a training set of L partition tree nodes identified as instances of concept c_j, the probability of occurrence of a feature f_i in c_j is defined using Laplace smoothing as:

$$P(f_i|c_j) = \frac{\sum_{p \in L} n_{f_i,p} + 1}{\sum_{i=1}^{i=|F|} \sum_{p \in L} n_{f_i,p} + |F|}$$

where $n_{f_i,p}$ denotes the number of occurrences of f_i in partition node p and $|F|$ is the total number of unique feature including word, p-grams, and t-grams. The number of nodes within the subtree of a partition tree node for a concept c_j is modeled as a Gaussian distribution with parameters mean u_{c_j} and variance σ_{c_j} defined as:

$$\mu_{c_j} = \frac{\sum_{p \in L} |p|}{|L|}, \sigma_{c_j} = \sqrt{\frac{\sum_{p \in L}(|p| - \mu_{c_j})^2}{|L| - 1}}$$

For new partition trees, the probability $P(c_j|p)$ of a node p being an instance of concept c_j is proportional to $P(p|c_j)$ assuming an uniform distribution for $P(c_j)$. We use a modified multinomial distribution to model the likelihood $P(p|c_j)$:

$$P(p|c_j) = (\frac{\overline{N}!}{N_{f_1,p}! \cdots N_{f_{|F|},p}!}) \times \prod_{i=1}^{i=|F|} P(f_i|c_j)^{N_{f_i,p}}$$

where $\overline{N} = K \times e^{(|p| - \mu_{c_j})^2/(2\sigma_{c_j}^2)}$, with K being a normalized total feature frequency count, $|p|$ being the total number of partition tree nodes within the subtree rooted at p, and $N_{f_i,p}$ is a scaled value of $n_{f_i,p}$ such that $\sum_i N_{f_i,p} = \overline{N}$. Note that the above formulation of the likelihood takes into consideration both the *number of nodes* within p as well as the frequencies of the various features in the content encapsulated within p. This results in a tight coupling between content analysis and document structure during concept identification. The partition tree node with the maximum likelihood value is identified as the concept instance. The partition tree after concept extraction process is shown in Figure 7. Observe that the irrelevant content in the original DOM tree has been filtered.
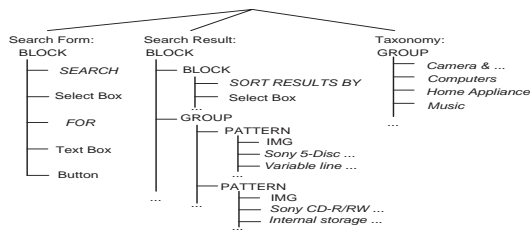
5. EVALUATION

Figure 7: Extracted Concept Tree

Figure 8: Recall for Concept Extraction.

Herein we report on the experimental evaluation of the learned process model, concept extraction and the integrated Guide-O-Mobile system. Thirty Web sites spanning the three content domains of *books*, *consumer electronics* and *office supplies* were used in the evaluation.

To create the concepts in the ontology we did an analysis of the transactional activities done by users on the Web in each of the three domains. Based on the analysis we came up with an "inclusive" set of concepts in Table 1.

5.1 Process Model

We collected 200 example transaction sequences from 30 Web sites. These were sequences whose elements are concept operations as illustrated in Figure 4. A number of CS graduate students were enlisted for this purpose. Specifically each student was told to do around 5 to 6 transactions with a Web browser and the sequences were generated by monitoring their browsing activities. They labeled a sequence as "completed" whenever they were able to complete the transaction; otherwise they labeled it as "not completed". We used 120 of these sequences spanning 15 Web sites (averaging 7 to 9 sequences per site) as the training set for learning the process model. The remaining 80 were used for testing its performance. The learned model is shown in Figure 3. The first metric that we measured was its recall/precision[1]. They were 90%/96% for the books domain, 86%/88% for the consumer electronics domain and 84%/92% for the office supplies domain. The second metric we measured was the number of transitions that remained to be completed when a true trace (completed transaction) in the test set failed to reach the final state. We observed that more than 50% of such failures ended one hop away from that state. This means that fast error recovery techniques can be designed with such a process model.

5.2 Concept Extraction

We built a statistical concept model for each of the concepts in Table 1. Recall that the five concepts in the upper half of the table are generic for all the three domains whereas those in the lower half are domain-specific. For instance the feature set of a list of books differs from that of consumer electronic items. We built one model for each concept in the upper half of the table and three - one per domain – for each concept in the lower half.

The concept model were built using the techniques in Sec-

tion 4. To build the model for each of the five generic concepts we collected 90 pages from 15 out of the 30 Web sites. For each of the domain specific concept we collected 30 Web pages from five Web sites that catered to that domain.

Note that pages containing more than one concept were shared during the building of the respective concept models. These models drive the concept extractor at runtime. We measured the recall[2] of the concept extractor for each concept in the ontology. Roughly 150 Web pages collected from all of 30 Web sites was used as the test data. Figure 8 shows the recall values for all of the 10 concepts in each of the three domains.

An examination of the Web pages used in the testing revealed that the high recall rates (above 80% for "Item Taxonomy", "Search Form", "Add To Cart", "Edit Cart", "Continue Shopping" and "Checkout") are due to the high degree of consistency of the presentation styles of these concepts across all these Web sites. The low recall figures for the "Item Detail" (about 65% averaged over the three domains) and "Shopping Cart" (about 70%) are mainly due to the high degree of variation in their features across different Web sites. A straightforward way to improve the recall of such concepts is to use more training data. However even this may not help for concepts such as "Add To Cart" that rely on keywords as the predominant feature. Quite often these are embedded in a image precluding textual analysis. It appears that in such cases local context surrounding the concept can be utilized as a feature to improve recall.

5.3 Integrated System

We conducted quantitative and qualitative evaluation of the integrated system; the former measured time taken to conduct Web transactions and the latter surveyed user experience. The evaluation was conducted separately for each domain using the corresponding domain-specific concept extractors (see Section 5.2).

Experimental Setup We used a 1.2 GHz desktop machine with 256 MB RAM as the computing platform for running the core Guide-O-Mobile system (shown within the outermost box in Figure 2). Guide-O-Mobile was architected to run in a client/server mode with the machine running the

---

[1]Recall for a process model is the ratio of the number of completed transactions accepted by the model over the total number of completed transactions. For Precision, this denominator becomes the total number of accepted transactions (completed and not completed).
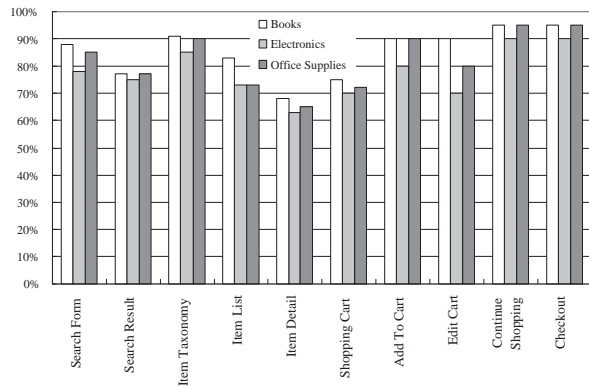
[2]Recall value for a concept is the ratio of the number of correctly labeled concept instances in Web pages over the actual number of concept instances present in them.

core Guide-O-Mobile system as the server and a 400MHz iPaq with 64MB RAM as the client. Thirty CS graduate students were used as evaluators. Prior to evaluation they were trained on how to use the system. We chose 18 Web sites (6 for each domain) to evaluate Guide-O-Mobile. We conducted roughly 5 to 6 transactions on each of them and calculated mean (μ) and standard deviation (σ) for all the measured metrics. Over 93.33% of those transactions could be completed. Evaluation of the integrated system discussed below is based on these completed transactions.

Quantitative Evaluation

Evaluators first conducted transactions with Guide-O-Mobile. Next they did the same set of transactions with the original pages loaded onto the iPaq, *i.e.* the page fetched from the Web was dispatched "as is" to the iPaq. Figure 9(a) lists the metrics measured. The "page load time" column contains the times taken to load the original pages into the iPaq. The "analysis time" is the time taken by the Guide-O-Mobile server to fetch pages from the Web, do content analysis, generate the DHTML page of the extracted concepts and upload them onto the handheld. Interaction time is measured from the moment a page is loaded into the iPaq until the user takes a navigable action like a form-fillout, clicking a link, *etc.* Figure 9(b) and (c) present graphically the overall and interaction times respectively.

Observe the significant decrease in interaction time to complete a Web transaction using Guide-O-Mobile. This reduction is mainly due to the filtering away of irrelevant content by the process model. Consequently the user avoids needless scrolling steps. Furthermore since the transaction is goal directed by the process model the user is able to complete the transaction in fewer steps. Another advantage of filtering away irrelevant content is the relatively small page load times. These times have been absorbed in the "analysis time".

Qualitative Evaluation

To gauge user experience we prepared a questionnaire for the evaluators (see Tables 2). They were required to answer them upon completing the quantitative evaluation. The questions were organized into two broad categories – system (S1 to S4) and concepts (C1 to C3) – the former to assess the overall functionality and usability of the Guide-O-Mobile system and the latter to determine the effectiveness of the semantic concepts in doing Web transactions. All of the concept questions except S1 and S2 required a yes/no response. From the responses we computed the mean percentage shown in the table.

Results:

A large percentage of evaluators felt that the concepts presented were self explanatory and contained enough information based on which they could take the right steps to make progress on their transactions (response to question C1). Some evaluators felt that notification of promotional offers, coupons, *etc.* was important and that such concepts ought to be presented (response to question C2).

Most were able to find the items they were looking for (response to question S1). However at times they were unable to complete the transaction (The "no" response to questions C3 and the unfinished transactions in S2). Analysis of such transactions revealed that in many cases the problem arose because: (a) the expected concepts in a state were not extracted; (b) the extracted concepts were mislabeled; (c)

C1	Did you find the concepts used in doing the transaction informative?	93.33%
C2	Do they capture all the useful information in the Web pages?	76.67%
C3	Did they help in accomplishing the transaction?	86.67%
S1	How often were you able to find the desired item?	96.67%
S2	How often were you able to complete the transaction for the item found?	93.33%
S3	Do you feel that the system restricted your navigation?	80%
S4	Did you find the system useful for conducting transactions?	96.67%

Table 2: Questionnaire with Response

the model could not make the correct transition. The last two problems could be addressed by training the concept extractor and the process model with more examples.

Quite a few evaluators felt that they expected more flexibility on how they can complete the transaction (response to question S3). Observe that the number of possible paths to complete a transaction is limited by the training data and hence this criticism can be addressed with more training. Overall they all felt that the system was adequate to do their tasks (response to question S4).

Evaluators also had general comments. In particular they all felt that the system requires help utilities to assist users to become familiar with the use and effect of each concept.

6. RELATED WORK

The work described in this paper has broad connections to research in Web services, semantic understanding of Web content, automata learning, non-visual Web access and browsing with mobile devices.

Web Services: This is an emerging paradigm that focuses on technologies that let service providers to export their functionalities on the Web so as to facilitate automated e-commerce. It has given rise to standardization efforts resulting in languages such as WSDL for describing services, SOAP for accessing services, UDDI for service discovery and integration, BPEL4WS for business process specification, and OWL-S as an ontology for semantic description of service metadata. Service providers are beginning to utilize them for exposing their services (see for example http://www.amazon.com/webservices) The complimentary task of annotating service descriptions with semantic metadata has been addressed in [15, 21, 25]. In contrast to these works we address a different kind of annotation problem, namely automatic annotation of different kinds of concepts that can occur in a Web page.

Web services expose very basic functionalities which by themselves are not sufficient to conduct complex transactions. For instance, Amazon's Web service exposes basic tasks such as searching for a product, adding a product into the shopping cart, *etc.* One has to compose these primitive services in order to perform complex transactions. This problem has been receiving attention lately [1, 6, 24, 27, 29]. All these works typically use process definitions and an ontology to create the composite service with varying degrees of automation. Note that our technique is based on composing operations over Web pages instead of services. A vast majority of transactions on the Web are still conducted over

Web Sites	Original Web Page in Handheld				Guide-O-Mobile			
	page load time		interaction time		analysis time		interaction time	
	μ	σ	μ	σ	μ	σ	μ	σ
Amazon	236.2	11.78	236	15.48	74.4	5.98	127	9.77
BN	293.2	31.59	194.6	34.79	73.2	7.95	101	14.23
Khazana	285.2	4.66	204.2	7.4	72.4	5.5	96.6	11.78
Blackwell	337.2	39.96	176	35.94	73.6	3.13	107.2	5.63
Angusrobertson	208.6	12.97	170.6	14.01	67.6	4.67	88.8	23.36
AbeBooks	195.6	23.04	179.8	9.83	72.6	4.93	102.6	13.09
Buy	384.2	46.82	311.6	21.17	87.4	8.47	160.4	13.24
Amazon	237.6	6.19	232.6	8.26	75.2	5.12	111.8	19.28
Bestbuy	283	24.1	251.2	16.63	83.8	5.32	130.2	22.31
CompUSA	272	11.22	227.4	12.2	84.6	5.68	115.6	7.3
eCost	224.6	26.54	287.4	42.65	78.2	11.3	142.6	14.89
outpost	326.4	45.8	179.6	17.69	65.6	6.54	134	20.19
tigerdirect	239	8.12	193.6	32.97	70	7.04	116.6	16.56
OfficeMAX	179.2	42.61	220.8	25.59	77.8	8.44	107.2	20.22
OfficeDepot	189.4	20.01	195	9.51	73.6	7.23	127.6	7.5
Walmart	300.6	54.16	220.2	23.92	95.4	12.66	123.2	17.71
Shop	383.6	48.15	227.4	10.76	89.4	11.61	99.4	16.83
QuillCorp	328.4	41.36	191.4	22.32	71.8	8.41	79.8	6.53

*All times are in seconds.

(a)
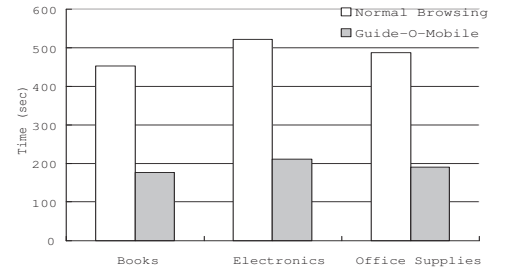
(b) Overall time
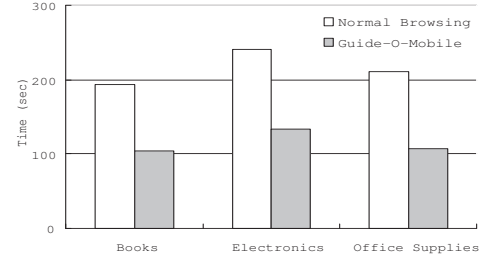
(c) Interaction time

Figure 9: Guide-O-Mobile Performance.

HTML pages. This focus on Web pages is what sets our work apart from those in Web services. Also note that our approach to Web transactions is quite flexible, in the sense that the users can define their own "personalized" transactional service instead of being confined to whatever service is exposed by the provider.

Semantic analysis of Web content: The essence of the technique underlying our content analysis module is to partition a page into segments containing "semantically" related items and classify them against concepts in the ontology. Web page partitioning techniques have been proposed for adapting content on small screen devices [4, 5, 7, 32], content caching [23], data cleaning [26, 31], and search [33]. The idea of using content similarities for semantic analysis was also recently explored in [34] in the context of Web forms. The fundamental difference between our technique and all the above works is the integration of inferring the logical structure of a page (see the partition tree in Figure 5(b)) with feature learning. This allows us to define and learn features, such as p-grams and t-grams, using the partition tree.

Concept identification in Web pages is related to the body of research on semantic understanding of Web content. Powerful ontology management systems and knowledge bases have been used for interactive annotation of Web pages [14, 16]. More automated approaches combine them with linguistic analysis [22], segmentation heuristics [10, 11], and machine learning techniques [9, 13]. Our semantic analysis technique is an extension of our previous work [18] and, in contrast to all the above, does not depend on rich domain information. Instead, our approach relies on light-weight features in a machine learning setting for concept identification. This allows users to define *personalized* semantic concepts thereby lending more flexibility to modeling Web transactions.

It should also be noted that the extensive work on wrapper learning [17] is related to concept identification. However, wrappers are syntax-based solutions and are neither scalable nor robust when compared to semantics-based techniques.

Process Model Learning: Our work on learning process models from user activity logs is related to research in mining workflow process models (see [28] for a survey). However, our current definition of a process is simpler than traditional notions of workflows. For instance, we do not use sophisticated synchronization primitives. Hence we are able to model our processes as DFAs instead of workflows and learn them from example sequences. Learning DFAs is a thoroughly researched topic (see [19] for a comprehensive survey). A classical result is that learning the smallest size DFA that is consistent with respect to a set of positive and negative training examples is NP-hard [2, 12]. This spurred a number of papers describing efficient heuristics for DFA learning(*e.g.*, [19, 20]). We have not proposed any new DFA learning algorithm for our work. Instead we adapted the simple yet effective heuristic with low time complexity described in [20].

Navigating to relevant pages in a site using the notion of "information scent" has been explored in [8]. This notion is modeled using keywords extracted from pages specific to that site. In contrast our process model is domain specific and using it a user can do online transactions on sites that share similar content semantics.

Content Adaptation: Adapting Web content for browsing with handhelds is an ongoing research activity. Initial efforts at adapting Web content for small-screen devices used WML (Wireless Markup Language) and WAP (Wireless Application Protocol) for designing and displaying Web pages [3]. These approaches relied on content providers to create WML content. Subsequent research [4, 5, 7, 30, 32] dealt with adapting HTML content onto these devices by organizing the Web page into tree-like structures and summarizing the content within these structures for efficient browsing. In our work we need to first filter away the content based on the transactional context represented by the states of the

process model. Summarization can be used to present the filtered content succinctly.

7. CONCLUSION

The use of complete Web pages for doing online transactions under constrained interaction modalities can cause severe information overload on the end user. Model-directed Web transaction can substantially reduce if not eliminate this problem. Our preliminary experimentation with Guide-O-Mobile seems to suggest that it is possible to achieve such reductions in practice. Currently Guide-O-Mobile is set up to run as a client/server application with the server doing the bulk of the processing. In the future it should be possible to port all the server steps to the handheld when improvements in handheld hardware and software technologies will be able to provide more memory and a richer set of browser operations than what are currently available. Finally integration of our framework with Web services standards is an interesting problem. Success here will let us specify the process model in BPEL4WS which in turn will enable interoperation with sites exposing Web pages as well as those exposing Web services.

8. REFERENCES

[1] S. Agarwal, S. Handschuh, and S. Staab. Surfing the service web. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[2] D. Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39(3):337–350, 1978.

[3] G. Buchanan, S. Farrant, M. Jones, H. Thimbleby, G. Marsden, and M. Pazzani. Improving mobile internet usability. In *Intl. World Wide Web Conf. (WWW)*, 2001.

[4] O. Buyukkoten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: Text summarization for web browsing on handheld devices. In *Intl. World Wide Web Conf. (WWW)*, 2001.

[5] O. Buyukkoten, H. Garcia-Molina, A. Paepcke, and T. Winograd. Power browser: Efficient web browsing for pdas. In *ACM Conf. on Human Factors in Computing Systems (CHI)*, 2000.

[6] L. Chen, N. Shadbolt, C. A. Goble, F. Tao, S. J. Cox, C. Puleston, and P. R. Smart. Towards a knowledge-based approach to semantic service composition. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[7] Y. Chen, W.-Y. Ma, and H.-J. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[8] E. Chi, P. Pirolli, K. Chen, and J. Pitkow. Using information scent to model user information needs and actions on the web. In *ACM Conf. on Human Factors in Computing Systems (CHI)*, 2001.

[9] P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[10] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan,

A. Tomkins, J. Tomlin, and J. Yien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[11] M. Dzbor, J. Domingue, and E. Motta. Magpie - towards a semantic web browser. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[12] E. M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

[13] B. Hammond, A. Sheth, and K. Kochut. Semantic enhancement engine: A modular document enhancement platform for semantic applications over heterogenous content. In V. Kashyap and L. Shklar, editors, *Real World Semantic Applications*. IOS Press, 2002.

[14] J. Heflin, J. A. Hendler, and S. Luke. SHOE: A blueprint for the semantic web. In D. Fensel, J. A. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web*, pages 29–63. MIT Press, 2003.

[15] A. Hess, E. Johnston, and N. Kushmerick. Assam: A tool for semi-automatically annotating semantic web services. In *Intl. Semantic Web Conf. (ISWC)*, 2004.

[16] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An open rdf infrastructure for shared web annotations. In *Intl. World Wide Web Conf. (WWW)*, 2001.

[17] A. Laender, B. Ribeiro-Neto, A. da Silva, and J. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2), 2002.

[18] S. Mukherjee, I. Ramakrishnan, and A. Singh. Bootstrapping semantic annotation for content-rich html documents. In *Intl. Conf. on Data Engineering (ICDE)*, 2005.

[19] K. Murphy. *Passively Learning Finite Automata*, 1995.

[20] J. Oncina and P. Garc. *Inferring regular languages in polynomial update time*. World Scientific Publishing, 1991. Pattern Recognition and Image Analysis.

[21] A. Patil, S. Oundhakar, A. Sheth, and K. Verma. Meteor-s web service annotation framework. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[22] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. Kim - semantic annotation platform. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[23] L. Ramaswamy, A. Iyengar, L. Liu, and F. Douglis. Automatic detection of fragments in dynamically generated web pages. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[24] J. Rao, P. Küngas, and M. Matskin. Logic-based web services composition: From service description to process model. In *Intl. Conference on Web Services (ICWS)*, 2004.

[25] M. Sabou, C. Wroe, C. Goble, and G. Mishne. Learning domain ontologies for web service descriptions: An experiment in bioinformatics. In *Intl. World Wide Web Conf. (WWW)*, 2005.

[26] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma. Learning block importance models for web pages. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[27] P. Traverso and M. Pistore. Automated composition of semantic web services into executable processes. In *Intl. Semantic Web Conf. (ISWC)*, 2004.

[28] W. van der Aalst and A. Weijters. Process mining: A research agenda. *Computers and Industry*, 53:231–244, 2004.

[29] A. Wombacher, P. Fankhauser, and E. J. Neuhold. Transforming bpel into annotated deterministic finite state automata for service discovery. In *Intl. Conference on Web Services (ICWS)*, 2004.

[30] C. Yang and F. L. Wang. Fractal summarization for mobile devices to access large documents on the web. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[31] L. Yi and B. Liu. Eliminating noisy information in web pages for data mining. In *ACM Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, 2003.

[32] X. Yin and W. S. Lee. Using link analysis to improve layout on mobile devices. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[33] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving pseudo-relevance feedback in web information retrieval using web page segnmentation. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[34] 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.