Abstract

In recent years, the growth of the WWW has rapidly expanded and it’s not always easy to retrieve information. Search engines are very useful, but they treat each page as an independent document. In this paper, we propose an automatic navigation method that follows users’ requirements for Web sites. This method detects the essential pages in the site and generates an ordered list of pages that present results by applying page linkage information. Furthermore, we have designed and implemented a prototype system based on our proposed method.

Key Words: summarization, presentation, Web, context

1. Introduction

In recent years, the growth of the WWW (World Wide Web) has rapidly expanded. In particular, the number of pages at each site continues to increase and the structure of these sites is becoming more complex. Therefore, it is not always easy for users to retrieve information even if they find a site that they want to visit. Search engines are very useful in finding pages containing user-specified keywords, however, in many cases, users do not really want these pages, because they do not understand their content when browsing. The result returned by search engines is simply a list of page titles containing the specified keywords. In other words, since each page is treated as an independent document in this type of reference algorithm, the structures of the Web site is disregarded.

The structure of Web sites refers to the combination of pages and the linkage of anchors for each page. That is, it is the hypertext organization itself. The structure of a Web site usually reflects the author’s intentions. The most important purpose for designing the structure is to facilitate navigation through the pages. In other words, it is a fundamental design methodology where anchors are put in the pages to determine the manner of navigation.

Generally, users must view many pages in a Web site by browsing through them one by one. The pages often contain a great deal of description. In such situations, it is not easy to understand the structure and the contents of the site, or to specify which portions fit the user’s requirements. To deal with these problems, Cha-Cha[7] and OdiN[8] (Open Document Information Navigation) have been proposed. The Cha-Cha system analyzes the link structure of sites and produces a list of results; the results include the shortest route from the top page to the reference result pages. ODiN also adds the route from the top page, based on both link and directory structures. However, in these systems, it is not always easy to understand the site structures. Moreover, since the lists are presented in static order, user requirements are not reflected.

In order to solve these problems, we propose in this paper an automatic navigation method adhering to the user’s requirements for a Web site. This method detects the essential pages in a site and uses page linkage information to generate an ordered list of pages. We have designed and implemented a prototype system based on our proposed method. In the current version, the functionality of automatic navigation is implemented by Flash movies.

This paper is organized as follows: in Section 2, we describe how to find essential pages and generate the navigation order of pages, which is called the context path. In Section 3, we explain how to extract partial documents from each page and to present the context path. In Section 4, we describe how to present the structures of Web sites. In Section 5, we describe the prototype system, called Edward, which is based on our proposed method. In Section 6, we discuss related work and compare it with our research. Finally, we give our concluding remarks and note our future work in Section 7.
2. Automatic Navigation of Web Documents

2.1 Essential Pages in Web Sites

When users navigate pages in a Web site, they have to select and search anchor points on the page that they are currently reading in order to visit other pages. Users find it troublesome to click the anchors one by one, especially when surfing pages in sites which they never would have visited and where they do not understand the structure of the pages and links. In these situations, it would be efficient if users could automatically navigate only the essential pages, without needing to know the structure of the site.

The set of essential pages in a Web site must consist of an abstract of the contents. However, if the set of essential pages is randomly shown, users may not adequately understand the content of the site. That is, the linkage relationship (position) of pages shows the semantics of the pages. Therefore, when a page is presented to users, the linkage relationship of the page must be shown in the order of the page navigation manner. We define context path as an ordered list of pages, arranged in page navigation manner. Page navigation manner means turning over pages, as in a slide show. In the following sections, we will explain the selection method of essential pages when a Web site is given.

2.2 Selection of Essential Pages

Many kinds of content are included in a Web site. It is, therefore, very difficult to determine the essential pages in a site; however, some techniques may be adapted to this problem. One technique is to apply the results of search engines; these results are the answers to a keyword search in a specific site. For example, if the name of a sports player is given as a keyword, the engine returns pages on which the name is included. The pages may consist of the bibliography of the player, game results, speeches for show business, and so on.

Another technique is automatic topic extraction. Although several methods have been proposed, in our research any of the methods are adaptable because we need only to detect the essential pages in a site. Moreover, since topic extraction is not within the scope of this research, we have omitted a description of it.

2.3 Common Ancestor Page

In the previous section, we mentioned that it is not enough to show essential pages in random order. To show each essential page in order, we have to decide a starting point. Therefore, we calculate a shared ancestry point for all essential pages. We call this starting point the common ancestor page. The common ancestor page is found by the following procedure:

Step 1 If there are target pages in which the common ancestor page is not found, Step 2 is performed. Otherwise, Step 2 is not performed and the procedure is finished.

Step 2 Pages linked to each target page are taken out.

Step 3 Paths are deleted when the pages obtained in Step 2 are pages which have already been followed. When the pages obtained have already been followed, except for themselves or those visited simultaneously, these pages are considered to be a common ancestor page. Go to Step 1.

In Fig.1, the common ancestor page of $Q, M$, and $U$ is $E$.

2.4 Navigation Path

In order to extract pages related to the common ancestor page and each target page, paths from the common ancestor page to each target page are calculated. This path is called navigation path.

A navigation path is calculated by the following procedure: (1) Take out the pages linked to each target page, and compute the reciprocal of out links for each page. (2) Choose a path, including the link, where the calculated value is the largest. This procedure is repeated to the common ancestor page.

For example, to calculate the navigation path from $E$ to $M$ in Fig.1 take out pages $H$ and $I$ linked to the target page $M$, and calculate the reciprocal of out links of each page. In this case, since the value of $H$ is $1/3$ and the value of $I$ is $1/2$, $I$ is chosen. Consequently, the navigation path from $E$ to $M$ is fixed $E \rightarrow I \rightarrow M$ (Fig.2).

If navigation paths are found for all target pages, the path set showing the target page in a certain domain can be found. A navigation path of Fig.1 is shown in Fig.3.
2.5 Context Path

Context path is the set of the navigation paths given in a presentation order. We generate context paths based on the similarity between pages. The degree of similarity between pages is calculated using tf-idf (term frequency · inverse document frequency). A calculated degree of similarity is given to each path, and a depth priority search is performed using this degree. The page order is determined by this operation. The left side of 4 shows the calculated degrees of similarity, and the right side shows the generated page order. The composition of the site is taken into consideration in this order of the page, and the context of the page is formed. In this case, the context path is set as $E \Rightarrow I \Rightarrow N \Rightarrow U \Rightarrow M \Rightarrow G \Rightarrow Q$.

3. Presentation of the Context Path

3.1 Extraction of the Main Portion in Web Pages

In the previous section, we described the extraction method of essential pages from a Web site and the generation method of a context path. Users may understand the abstract of a site by the essential pages; however, the pages will still contain both meaningful and useless parts. In other words, it is hard for users to pay attention to the main meaningful part of a page when showing the entire contents of each page. We propose a method to extract the main portion of Web pages as well as a method to link the main portions of each page.

Generally, tags of HTML documents are classified as follows: structural tags, visualizing tags, embedding tags and so on. Structural tags are classified into two types: heading tags and other tags. When a page and a keyword are given, the main portion of the page is the neighbor of the position where the keyword exists in the page. The neighbor can be calculated as the maximum portion, as divided by the nearest heading tags vertically, both up and down. If there are no heading tags in the page, the maximum portion is divided by the other nearest structural tags.

3.2 Presentation path

Now we describe how to present the elements for context paths. The path elements are the titles, anchors and partial documents of a page. The path is called presentation path. In other words, the presentation path is a skimmed-down version of the context path, and it deletes the useless parts of each page. The algorithm for the generation of the presentation path is shown below.

Let a partial document be called content $C_i$ ($i = 1, \ldots, l$ is the number of the partial documents in a site). We define the four functions used in the algorithm as follows:

- $\text{generate-path}(C_i)$ adds the title of common ancestor page, $\text{anchor}_1 \ldots \text{anchor}_j$ ($j$ is the number of structural tags)
path \(-1\), \(C_t\) to the path, and return the path.
- check-turning-point\((C_s,C_t)\) returns the page including the turning point elements of the paths for \(C_s\) and \(C_t\).
- tag\((P,C)\) returns a tag which covers the content \(C\) in page \(P\).
- delete-before\((P,C)\) returns the path which deletes elements of the path before \(P\).

We define the input and output of the algorithm as follows:

**Input**: an ordered list of the contents in the target pages

**Output**: a presentation-path

Here is the algorithm:

**step 1** \(\text{PATH}_1 = \text{generate-path}(C_1)\)

**step 2** \(\text{PATH}_i = \text{generate-path}(C_i)\)

**step 3** \(P_i = \text{check-turning-point}(\text{PATH}_1,\text{PATH}_i)\)

If \(P_i \neq \text{null}\) then go to step 4
else go to step 5

**step 4** compare \((\text{tag}(P_i,C_i))\) with \((\text{tag}(P_j,C_j))\)

If same then \(\text{PATH}_i = \text{delete-before}(P_i,C_i)\) and go to step 5
else go to step 5

**step 5** \(\text{PATH}_i\) adds to \(\text{PATH}_1\) and \(i = i + 1\)

**step 6** If \(i > l\) then a presentation-path = \(\text{PATH}_1\), and exit.
else go to step 2

We should explain the purpose of step4 in this algorithm. The algorithm makes paths for each quantity of content in step2. If the paths are in order, the same scenes may be reproduced repeatedly. Therefore, we think that these paths must be merged. Therefore, step4 combines paths containing the same anchor and merges them.

The idea of merging is described as follows. If there are paths with the same combination of path elements from the title of a common ancestor page, those paths are called the same-level-path-group. Each path in the same-level-path-group is compared to the head. If a path element which is different from the same class appears, the element will be considered to be a key path. The relationship of these key paths determines whether it is shown in parallel.

(a) When the structural tag of a key path is the same:
We assume that the site designer showed the same semantic attachment for the anchors into the pages. Therefore, we thought that showing the anchor in parallel made it easier for users to understand. In order to show it in parallel, the same-level-path-group was combined and one reproduction path was generated.

(b) In other cases.
When the structural tag differs, we assume that the designer showed the difference semantically by designing those anchors as different structures, although they are on the same page. Therefore, it is reproduced as a separate path in these cases.

4. Presentation Effect for Structure Pattern

4.1 Structure Pattern of Path Navigation

Three kinds of presentation elements are included in presentation paths: title, anchor, and content. The five cases, described below, show different kinds of presentation elements connected by presentation paths s(Fig.3.1). We call these structure patterns.

1. Anchor to Anchor
An anchor points to another page. The anchor included in the page points to a different page.

2. Parallel Anchors
An anchor points to another page. The page includes several anchors.

3. Anchor to Contents
An anchor points to another page and content exists on that page.

4. Parallel Contents
An anchor points to another page and multiple contents exist on that page.
5. Content to Anchor
An anchor including content points to another page and another content exists on that page.

In the next section, Section 4.2, presentation-effect techniques for each structure pattern are shown. We also discuss visual effects, which are like a slide show of oral presentations.

4.2 Presentation by Visual Effect
The presentation method discussed here explains the kinds of slide-show effects added to each of the five structure patterns. That is, this presentation method is a technique where the kind of visual effect reflects the role of a page in a Web site. Roles would be an indexing page, a passing page, content itself, and so on. Here are the effects for each of the five structure patterns:

1. Anchor to Anchor (See Fig. 6)
A door-open metaphor is attached. After opening the previous page, the next pages include the next anchor. Users can easily imagine that they are following a path from an upper anchor to a lower anchor.

![Figure 6. Anchor to Anchor.](image)

2. Parallel Anchors (See Fig. 7)
In this case, a door-open metaphor is also attached. The next pages after opening the previous page include the next anchor aligned in parallel positions. Users can easily imagine that they are following a path from an upper anchor to other pages including several anchors.

![Figure 7. Parallel Anchor.](image)

3. Anchor to Contents (See Fig. 8)
In this case, an anchor is presented. Then, content appears from the bottom of the display and enters. This is a slide-in metaphor. In many cases, anchors are menus or titles of the following pages. In order to recognize that the relationship between the anchor and the following content is tight, content is shown using animation, carrying out a slide-in metaphor from the bottom.

![Figure 8. Anchor to Contents.](image)

4. Parallel Contents (See Fig. 9)
There are several partial documents in the same page. In this case, in order to recognize that they are on the same page, after showing the previous content, the next content comes like a slide-in metaphor from the right.

![Figure 9. Parallel Contents.](image)

5. Contents with Anchor (See Fig. 10)
There are several contents which are connected by an anchor. In this case, a dissolve metaphor is attached. Previous content and the anchor are displayed and disappear gradually. Then, the next content appears slowly in the same position (in place of the previous content).

![Figure 10. Contents with Anchor.](image)

5. Prototype System

5.1 Outline
These sections describe the prototype system that is based on our proposal method. We call this system `e-dward`. In this section, we explain the outline of `e-dward`.

First, a user specifies a website and their interested keyword. Thereby, a target page, as described in Section 2, is extracted. Then, processing, as stated in Sections 2 and 3, is applied to the specified website, and a presentation path is generated. We use Perl language for this process.
Next, we mount visual effects as presentation effects. Macromedia’s Flash movies are used for mounting. The templates are prepared by Generator [2] so that the Flash movies can be automatically created for each structure pattern. In addition, Flash movies that present the specified contents are created by combining the presentation paths and templates dynamically. Here, because Flash movies are created for each structure pattern, they must be reproduced continuously. We use SMIL (Synchronized Multimedia Integration Language) to do this. By applying the \texttt{<seq>} tag of SMIL to all Flash movie files, one after another movie, each of which reproduces one path, is shown as it is reproduced.

Finally, when showing a presentation path, different background music for each navigation path is used, which signals the users that the path is switching to a different one. The processing flow for e-dward is shown in Fig. 11.

5.2 System Architecture

![Diagram of System Architecture](image)

**Figure 11. Outline of Prototype System.**

5.2.1 Context Path Generation

The URL and keywords of a browsing site are input into the navigation-path generation portion of our prototype system. Retrieval by keyword is performed for pages which use the same keyword in the same site or domain. In the current prototype version, the Google power search is used. After extraction of the target pages, according to the algorithm stated in Section 3, a set of navigation paths is extracted for each target page. For this operation, e-dward analyzes the links and makes a link list of the site. The link list extracts and summarizes the link relationships between all pages in the site. After the extraction of the navigation path, e-dward calculates the degree of similarity and generates the context path, which is an ordering of the pages.

5.2.2 Presentation-Path Generation

As stated above, link analysis is performed and a link list of the site is generated. A link list extracts and summarizes the link relationship between all pages in the site. Next, according to the algorithm stated in Section 3, the presentation path from partial documents to the common ancestor page is generated.

5.2.3 Synchronization Contents Generation

**Template Generation** Templates, which contain the variable elements for Flash movie generation, are created beforehand by Generator. Each template corresponds to its use in the presentation system. The created templates are stored in the production scenario database.

**Movie Generation** A placeholder for text is contained in the templates, and Generator receives the contents from the data source. This placeholder is replaced dynamically, and a Flash movie is created. The delivery of this data source generates a batch file for running Generator offline in a Perl script; and so, this process is carried out automatically.

**Synchronization Contents Generation** Reproduction of the generated movie is controlled by SMIL. Since the length and combination of a path changes according to the Web site or input keywords, a SMIL document is generated dynamically and simultaneously with the movie generated by the Perl script. The SMIL tag used for this is the \texttt{<seq>} tag. By surrounding the created movie file with the \texttt{<seq>} tag, the Flash movie file described by the SMIL document is reproduced continuously.

5.3 Example

Our e-dward prototype was applied to a Web site with the structure shown in Fig.12. Five pages out of the 56 pages in this site contained partial documents. After path candidates were generated from these pages to the common ancestor page, paths were merged and a presentation order, as shown in Fig.12,
was generated. “T” means a title of a site, “A” expresses an anchor and “C” expresses a partial document.

Fig.13 is an example of a presentation screen of the “Anchor to Anchor” pattern (See Fig.6). The anchor of the title screen or passage page is shown, a page breaks at the center of the screen, and the anchor portion that exists in the following page appears. A passage page here is neither a common ancestor pages nor the target pages.

Fig.14 is an example of presentation of the “Parallel Anchors” and “Anchor to Contents” (See Fig.7 and, 8). That is, this example is the presentation screen of a path when two or more anchors exist to the page to the partial document. Two or more existing anchor portions, to which the screen from the last page breaks at the center, are shown. Then, signs indicating a partial document slide in from the bottom, as is shown.

Fig.15 is the example of a presentation of “Parallel Contents” (See Fig.9). In this case, two or more partial documents exist in the same page. After the first partial document is shown, signs that the following partial document, slide in from the right.

As a result of applying e-dward to this site, the presentation screen is shown in Fig.16².

²The screen images are aligned chronologically.
(a) is the screen which displays T.
(b) is the screen of the changes from T→A1. At this time, the screen of (a) breaks from the center and the screen of A1 appears.
(c) is the change from A1→A2. A crack and the screen of A2 appear from the center, as in (b).
(d) is the screen of the changes from A2→A3 and A4, the screen of A2 opens it from the center, and A3 and A4 appear.
(e) is the screen which presents C1. At this time, A4 of (d) is slid to the left-hand side, and disappears from the screen.
(f) C1 slides in from the bottom simultaneously.
(g) A4 returns.
(h) is the screen which presents C2. As in C1, A3 slides horizontally, it disappears from the screen, and C2 comes up from the bottom simultaneously.
(i) the title screen returns, in order to show the following path.
(j) is T→A5.
(k) shows the changed screen of A5→A6. It is the same flow as (a), (b), and (c).
(l), (m), and (n) are the presentation screens of C3 and C4. Since C3 and C4 are in the same page, after C3 is shown, they slide to the left side and disappear from the screen. Simultaneously, C4 slides in on the screen from the right side.

The above operation is repeated until all of the partial documents in the site are shown.

6. Related Work

We have proposed a system which presents information effectively, after extracting only the portion relevant to the user’s interests in page groups which consist of two or more pages and have a huge amount of information. In this section, we review related work and compare it with our method.

Shinagawa[11] has proposed a technique of generating a virtual page from a Web page based on an individual profile. This technique identifies the user’s profile, which consists of thresholds which control the keyword group which express a user’s interest for their viewpoint, and the degree of details of a display. Then, a virtual view page, based on the user’s profile at the time of browsing, is dynamically generated and displayed. A logical tree showing the composition of text that is not in the document structure of HTML is extracted and used in the generation of the view page. Although this technique generates a virtual page from one Web page, Shinagawa has not examined the case where the Web page becomes the generating agent for two or more pages.

Cha-Cha[7] and OdIN[8] researchers have proposed a system that displays a reference result in relation to a circumference page, in order to support user browsing. In their approach, an “outline” or “table of contents” is created by first recording the shortest paths that combine to form a hierarchical outline of the context containing the search results. However, because the reference result is shown in list form, users have to manually browse output pages one by one.

Site Cruise Theater[3] and Tourum.com[4] are also passive browsing systems. When a certain URL is specified, Site Cruise Theater scrolls the Web page automatically and shows it in the style of a TV program. Since a series of Web information constitutes the "program" and push distribution is performed, users can passively browse a Web site. However, in Site Cruise Theater, the presentation order of pages is simple and neither the contents of a page nor the link relationship of the site is taken into consideration. Tourum.com automatically presents a sequence of two or more Web pages (i.e., a tour), one after another. This system passively shows the existence of various Web sites by reproducing a tour which users arbitrarily create, with their viewpoint, and register. The purpose of Tourum.com is to introduce the existence of various Web sites; therefore, it differs from the purpose of this research.
7. Conclusions and Future Works

In this paper, we discussed a technique of extracting only the required portions from page groups which exist in a certain Web site. We also proposed a passive presentation system called e-dward, which automatically takes into account the hyperlink structure of operation when showing the results. Furthermore, we also discussed the results of applying e-dward to Web sites.

We think that users’ information-gathering is improved by this system because it extracts and presents only the portion relevant to a user’s interest, based on their input of keywords. Our system makes it easier to browse information about a certain topic on a site. Furthermore, we believe that animation helps the user to understand the position of the information shown in the site.

References