Journal of International Information Management

Volume 10 | Issue 1 Article 3

2001

Analyzing and documenting an internet website using NoteCards

Jay M. Lightfoot
University of Northern Colorado

Analyzing and documenting an internet website using NoteCards

Jay M. Lightfoot
University of Northern Colorado

ABSTRACT

Corporate websites are an important component in the world-wide web. The traditional way of creating these websites leads to a variety of structural problems that reduce the effectiveness of the websites. These problems are difficult to locate and correct using the ad hoc analysis methods that currently exist. This paper introduces a new technique to analyze and document websites. The technique uses the NoteCards hypertext environment to build a visual model of the website. The model is semantically rich, dynamically extensible, and allows interactive update. The result of using this technique is a website that is easier to use, easier to update, and fully documented. The technique is described and demonstrated on a small website.

INTRODUCTION

The internet and the world-wide web have had a phenomenal impact on organizations and society. Individuals can search for library citations, study satellite maps, and check current sports scores in a single login session. Corporate users can view product information, order merchandise, and track shipping progress without the aid of company employees. Even the language has changed. It is common to hear conversations concerning *web addresses, ftp sites* and *homepages.* In the five years since the first web browser programs were distributed, the internet and its associated global connectivity have forever changed the way we interact and communicate with each other.

A key component of the world-wide web is the corporate website. These internet entities are the equivalent of a virtual storefront for the corporation. They provide information about the organization, advertise its products, and create customer goodwill through promotions. When customers need information about a company, they go to the website in the same way they would go to a physical store for more tangible products. In the future, dynamic client/server interaction will further enhance the importance of these web entities.

The success of an individual corporate website depends heavily upon its ease of use. End-users of the internet have little patience for ill-defined, illogical websites. When web users need information, they want to retrieve it with minimal effort. Anything less is unacceptable. Ease of use can be achieved by good website structure. Good website structure implies that the available access paths take the end-user to the information of interest in a logical, efficient manner. Without good structure, the user is left to wander through a seemingly arbitrary maze of links and dead-ends. This has the effect of frustrating the end-user and creating negative feelings toward the company--the exact opposite outcome was intended.

This paper addresses the need for good website structure by describing a technique to analyze and document existing websites. The technique uses the NoteCards[1] hypertext software environment to capture the structure of the website. The paper begins by describing the problems associated with traditional website design. It then introduces the NoteCards software tool as a solution to these problems. Next, an analysis and documentation technique using NoteCards is explained and demonstrated on a simple website example. Finally, the advantages and disadvantages of the technique are discussed and its future enhancements are covered. Using the technique will result in websites that are easier to use, easier to maintain, and fully documented. From a corporate standpoint, this is good because it will create a user-friendly site that promotes customer good will and encourages repeat visits.

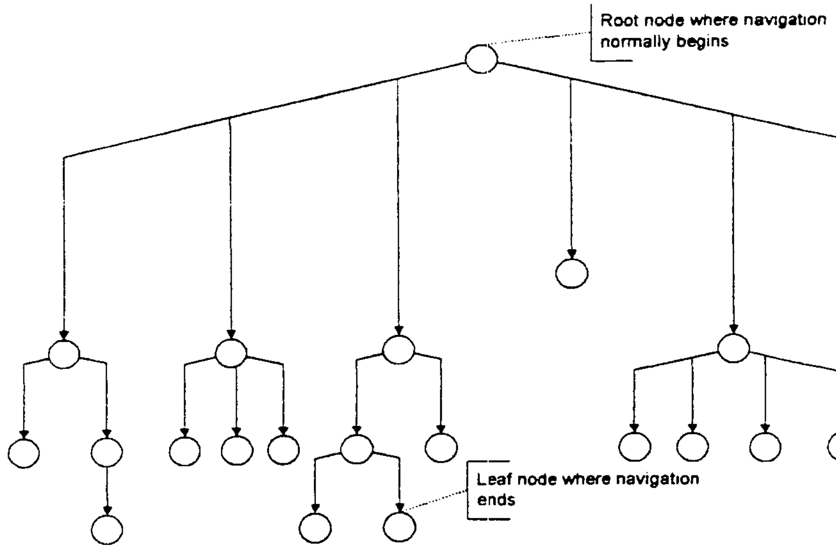
THE PROBLEMS OF TRADITIONAL WEBSITE DESIGN

Designing websites is a relatively new activity, so it is difficult to pinpoint a "traditional approach." However, some characteristics of internet website design are emerging as common practice. For instance, most websites are developed in a piecemeal fashion beginning with a single primary web page called a *homepage.* The person in charge of building the website (i.e., the *webmaster*) designs the homepage as the main entry point into the website. The homepage includes generic introductory information and links to other, more specialized, web pages within the site. These pages, in turn, lead the user to more specific information through a series of hypertext links.

The structure that results from this process is an *inverted tree* where the homepage is the *root node* and the final destination pages are the *leaf nodes.* A graphical representation of this type of website is shown in Figure 1. In this initial structure, the branches of the tree are used to categorize and classify information into logical groups. The navigation of the website is intended to proceed from the root node down through the connected branches. Since most web browsers have backtracking capabilities, the result of this structure is to allow bi-directional vertical navigation.

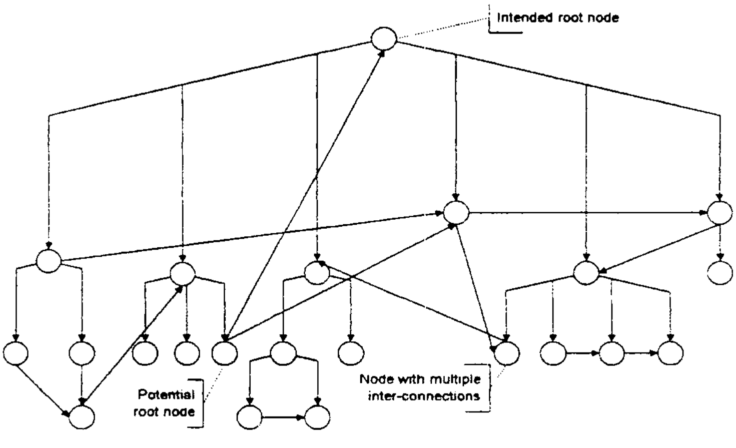
[1] NoteCards is a software package sold and supported by the Venue Corporation, San Carlos, CA.

Figure 1. Initial Inverted Tree Structure of a Website


Root node where navigation normally begins
Leaf node where navigation ends

Once the initial structure is built, the webmaster normally adds additional links between internal web pages. This creates a complex interconnected structure commonly called a *network*. In a network structure, any node can be connected to any other node by any number of links. This structure allows the greatest possible flexibility and takes advantage of the natural hypertext capabilities of the internet. A graphical representation of an interconnected network website is shown in Figure 2.

Figure 2. Interconnected Network Website Structure



In a website that is structured as a *network*, the end-suer can follow the links between any web pages that are connected. Navigation is intended to proceed from the homepage down; however, it is possible that end-suers will enter the site from an alternative node. The internal interconnections are placed between those nodes that the webmaster feels are the most likely to be needed by the user. In essence, the webmaster is pre-planning the available navigation paths for the end-user. This process requires a great deal of insight and planning. Any mistakes or oversights in this stage of website development will make the site more difficult to use.

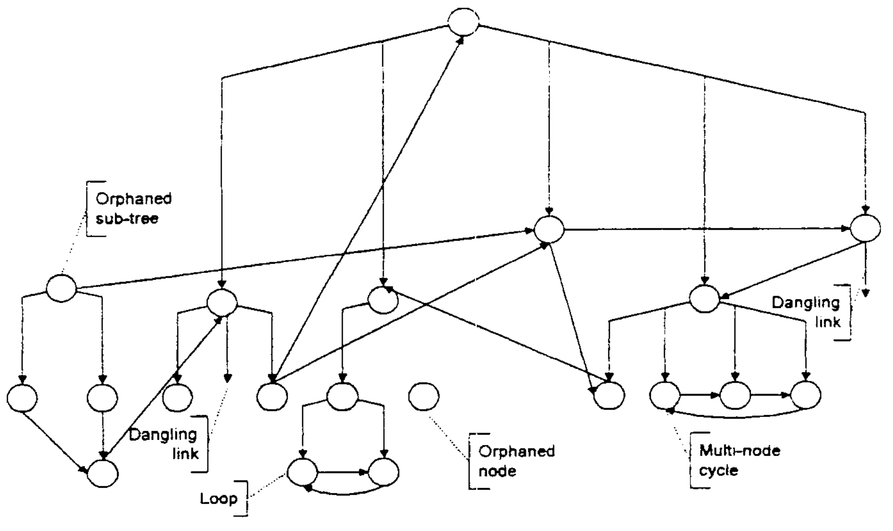
After the interconnected website is completed, it is normally put into operation. Users are allowed to access the site, navigate pages, and download information. Over time, the website must be modified and updated to remain current. This is the website equivalent of software maintenance. It is during this maintenance process that most website structural problems are created. Common website structural problems are described below and graphically illustrated in Figure 3.

- **Dangling links** -- These occur when a web page is deleted and the link(s) that point to it are left intact. Following a dangling link gives the user an error stating that the destination does not exist.
- **Orphaned nodes** -- These occur when the only incoming link to a web page is removed. This leaves a stand-alone web page that is effectively cut off from the website.
- **Orphaned sub-trees** -- These occur when the only incoming link to a sub-tree is deleted. The result is a sub-tree without any direct access paths.
- **Loops** -- Loops occur when two web pages refer to each other exclusively as the next forward link. Navigation takes the user in a tight circle.
- **Multi-node cycle** -- This problem happens when more than two web pages exclusively refer to each other as the next link

The website represented by Figure 3 exhibits all the common structural problems associated with the traditional approach to website creation. The clean organization of the original inverted tree has been lost and the carefully planned access paths no longer take the user where they need to go. These types of structural problems are normally caused by careless website maintenance. This results in a site that is difficult to use and reflects badly on the organization that created it. End-users will avoid using the site and will go elsewhere for their information needs. Unfortunately, these types of problems are the natural result of the traditional approach to website design.

In defense of the webmaster, these problems are difficult to avoid. There are few, if any, established website documentation tools and techniques available, so the webmaster is left to analyze and document the site with ad hoc methods. These methods usually consist of scattered notes and drawings intended to reflect the overall site structure. When changes are made, these notes quickly become out-of-date and are rendered useless. Unless the webmaster is extremely diligent, the hurried pace of business allows only time to perform site maintenance. Documentation and analysis are perceived as luxuries that can be put off until some future, unspecified, time (that never seems to arrive).

Figure 3. Interconnected Network Webiste Structure with Problems



A SOLUTION TO WEBSITE DESIGN PROBLEMS

The problems described above can be avoided by developing tools and techniques specifically tailored to analyze and document websites. These tools must be easy to use, graphical in nature, and on-line to facilitate quick maintenance. This section describes one technique for this purpose. It begins by introducing the NoteCards hypertext environment. NoteCards is a graphical software tool that is very well matched to the task of website analysis. Next, the capabilities of the technique are demonstrated on a simple example.

The NoteCards Software Environment

The Notecards[2] software environment is a general-purpose hypertext tools written in the InterLisp-D language. According to Halasz, Moran, and Trigg (1987),

> NoteCards provides the user with a "semantic network" of electronic notecards interconnected by typed links. This network serves as a medium in which the user can represent collections of related ideas. It also functions as a structure or organizing, storing, and retrieving information. The system provides the user with tools for displaying modifying, manipulating, and navigating through this network.

[2] In this paper, the word *notecards* (lower case) refers to the system entity capable of storing information while *NoteCards* (mixed case) refers to the software environment.

NoteCards were originally developed by researchers at the Xerox Palo Alto Research Center (PARC) to "allow users to build semantic networks of textual information" [2, p. xix]. Both hierarchical and arbitrarily complex network structures are directly supported by NoteCards. This gives the tool the flexibility to model a wide variety of domains. NoteCards has been successfully applied to many areas including argument analysis for a Ph.D. thesis, exception reporting for high level managers, brainstorming analysis of e-mail configurations, and building an index to the Lexis legal database.

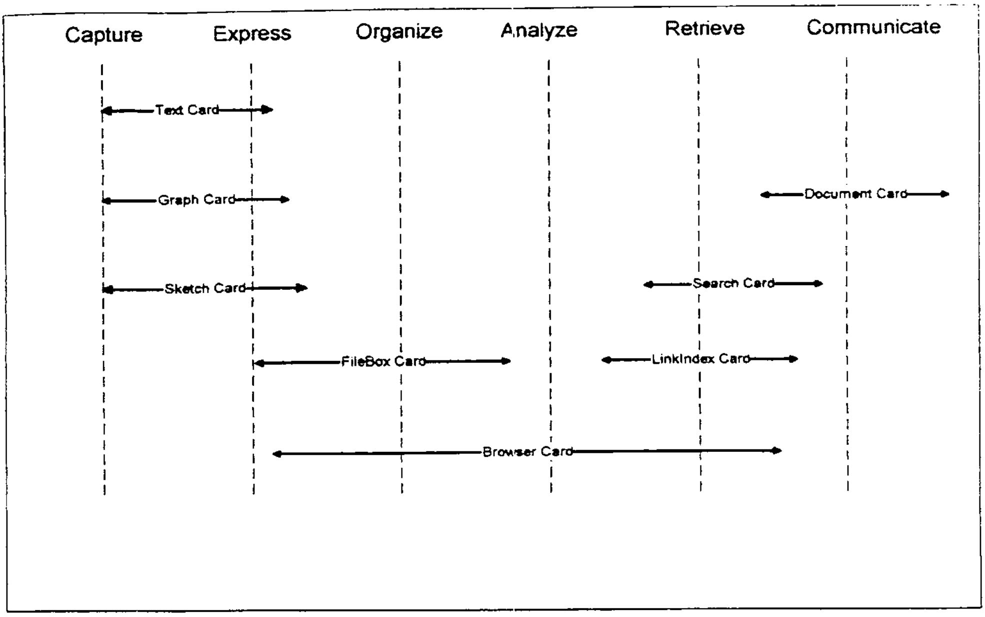
NoteCards is designed with two basic components--*notecards* and *links*. A **notecard** is a system entity capable of holding information in the form text, graphics, or a combination of the two. Each notecard can hold as much information as desired. The NoteCards system comes with eight predefined notecard types. These are described below.

- **Text cards** -- Text cards are used to hold textual material. A powerful text editor is associated with each text card. Freehand sketches and bitmap images also can be included in text cards.
- **Graph cards** -- Graph cards allow the user to create a lattice of words and phrases connected by lines. This is useful for spatial layout and high-level planning.
- **Sketch cards** -- These cards allow the user to draw and manipulate bitmap images using a sophisticated graphics editor. Images can be imported from external sources.
- **Browser cards** -- Browser cards present an interactive graphical view of cards and their links. Processing options associated with browser cards allow the user to specify the types of cards and links to be shown. Selecting a card in the browser display opens the card for editing.
- **FileBox cards** -- Fileboxes are container cards for other cards (including other filebox cards). The cards in a filebox must conform to a strict hierarchy. Normally, fileboxes are used to hold all the cards that are related to a topic. This makes them useful for categorizing and organizing information.
- **Linkindex cards** -- Linkindex cards are used to build sorted lists of the notecards that are connected by a specific link type. This allows related ideas and information to be traced.
- **Search cards** -- Search cards process notecards looking for a specified string of characters in the title portion of the cards. The result of the search is a new card with links to all cards that contain the string.
- **Document cards** -- These cards are used to collect textual information from all descendant cards of a sub-tree.

These notecard types provide a broad range of functionality. Specifically, the text, graph, and sketch cards are useful in capturing and expressing information. Filebox cards are most useful for organizing and analyzing data. Search and link index cards specialize in retrieving information, and document cards are most effective at communicating information to end-users.

Finally, browser cards provide the widest functionality by supporting organization, analysis, and retrieval operations. The functional areas covered by the 8 predefined notecard types are graphically shown in Figure 4.

Figure 4. NoteCard Types and Their Range of Use



In addition to the original eight notecard types, the user is free to extend the system by adding additional types. The user-defined card types can include a combination of original behavior and inherited behavior from existing card types. Once defined, the system treats user-defined notecard types the same as the predefined card types. The ability to dynamically extend the system is one of the characteristics that makes NoteCards a powerful, flexible modeling tool appropriate for a wide variety of problem domains.

Links are the second major component of the NoteCards system. A link is a typed connection between notecards. In this usage, *typed* means that each link instance is a named semantic object that represents a specific type of semantic relationship between the contents of the source card and the destination card. Thus, links can be used to connect related pieces of information. Any number of links can be placed between two notecards and each link can represent a different facet of the relationship between the two cards. The NoteCards system comes with three predefined link types: comment links, see links, and unspecified links. The predefined types are fairly generic and of limited use; however, user-defined links can be dynamically defined to represent special types of relationships. As with user-defined notecards, user-defined links are treated as equals with the predefined link types.

NoteCards for Website Analysis

The NoteCards environment can be used to locate and correct website structural problems. It can also be used to plan future maintenance so that structural problems are avoided. This is accomplished by using the basic tools provided by the NoteCards environment and augmenting them with a small set of user-defined link extensions. Taken together, this creates a powerful website analysis technique.

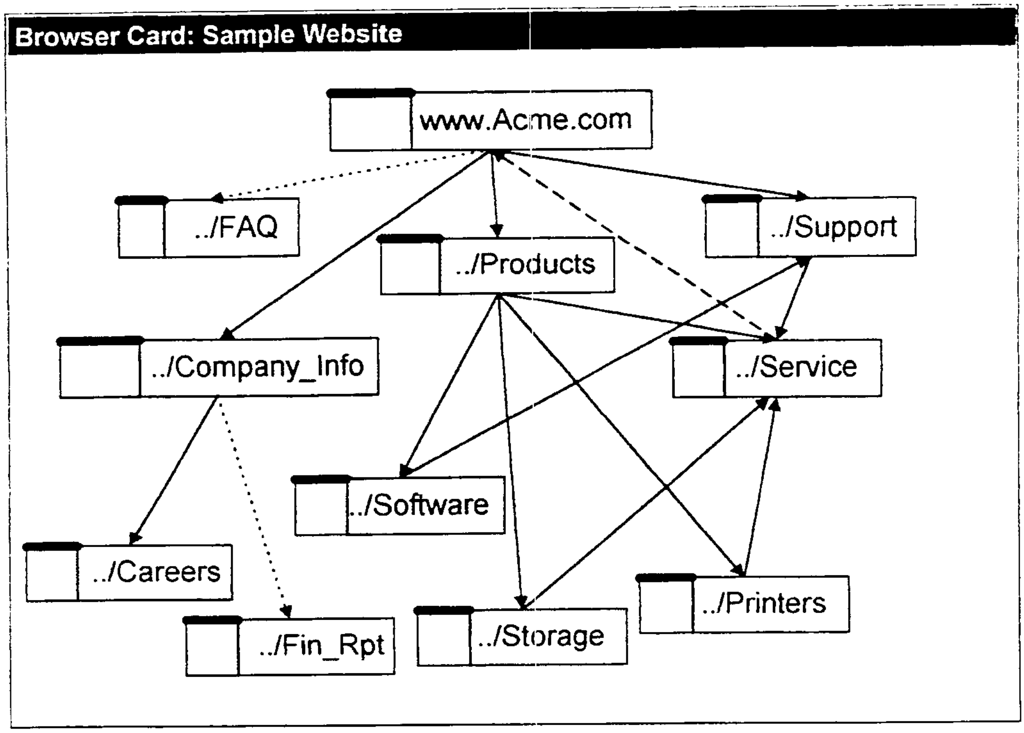
The first step in the analysis technique is to create a **browser card** that mirrors the existing site structure. Start by adding one text card for the homepage (i.e., root node). The name of this card should be a combination of descriptive text and the website address. Do not add any text contents to this, or any subsequent cards in this step. Next, add additional text cards for all the web pages linked to the homepage. The titles of these cards should also be composed of descriptive text and web address information. Continue the process until all pages of the site (or the branch currently being analyzed) are represented on the browser card.

The next step is to connect the notecards with appropriate links. A link is appropriate if it indicates the type of information that the user will find when the destination card is accessed. For example, if a web page is connected to the site homepage, then a user-defined **homepage link** should connect the two. Likewise, if a web page connects to an ftp site, an **ftp link** is appropriate. The exact set of user-defined links will vary from project to project, but the basic types for website analysis should include: **webpage link, homepage link, filebox link, ftp link, image link, audio link, video link,** and **applet link.** These types can be dynamically created (and removed) by the analyst when needed. As website technology advances, new link types can be created and incorporated into the model to reflect new types of connections. A browser card representing a simple website is shown in Figure 5.

The notecards in Figure 5 are represented as named boxes and the links as lines between the boxes. The name on the notecard indicates its contents and relative address from the homepage. The direction of the links is indicated by arrows on the link lines. The NoteCards environment is capable of showing each link type with a different line pattern. This helps the user visually comprehend the picture. For example, in Figure 5 the webpage links are shown as solid lines, ftp links as small dashed lines, and homepage links as large dashed lines.

The visual nature of the browser card is the key to its power as an analysis tool. Once the website is fully represented as a picture, the webmaster can visually detect structural problems. Dangling links, loops, multi-node cycles, and orphaned sub-trees are dramatically obvious. Orphaned nodes can be detected by a special feature of NoteCards that lists nodes with no known parents. If the complete picture is too complex to follow, the browser card can be instructed to show only those nodes connected by a specific link type or only those cards that are decedents of

Figure 5. Browser Card Containing a Simple Website Map



a specified node. This gives the webmaster a tool to locate existing problems and to certify that website maintenance will not create new problems. The on-line nature of the tool makes it easy to view and update; consequently, there is not reason for the web map provided by the browser card to become out-of-date. This web map can be included in the homepage as a static picture to help end-users to navigate the website.

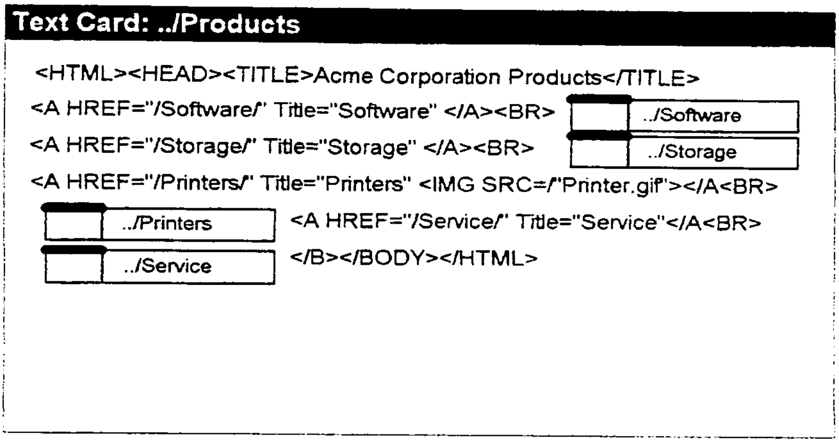
Additional analysis can be performed with linkindex and search cards. Linkindex cards trace a specific link type through the entire NoteCards data model. The result of creating a linkindex card is a list of all cards connected by the named link type. This is an easy way to make sure that a series of related web pages are connected by a continuous set of links. For example, if a group of web pages are all related by the concept of *customer support*, then a special customer support link can be created to connect the cards. When website analysis is performed, a linkindex card can be created to trace the customer support link to verify that all related cards are connected. If future maintenance accidentally deletes one of the cards in the series, the linkindex card will show a break in the hypertext chain. This proactively alerts the webmaster to the problem

Search cards are used to look for a specific text string in card titles. The result of creating a search card is a new notecard with links to all cards that contain the text string. This is useful for analysis when checking to see which web pages in a website concern a specific concept. For instance, the webmaster could create a search card to locate all notecards that contain the word "products" in the title. The list of cards shwon on the resulting search card indicates all website cards that deal with product issues. Naturally, this is only useful if card titles accurately describe the card contents.

NoteCards for Website Documentation

Websites can also be documented using the NoteCard technique. Website documentation provides a more complete record of the website and is useful when the full content of the website must be recorded. Documentation is accomplished by adding contents to the text cards located on the browser card. The contents of the text cards can include strictly the text of the webpage or the text and hypertext markup language (HTML) of the web page. Both os these can be typed into the card directly or imported from an external source. The NoteCards environment does not currently include an HTML browser, so the web page cannot be viewed as the end-user would see it. A notecard will full text contents is shown in Figure 6.

Figure 6. Expanded Text Card with Web Page Content and Links



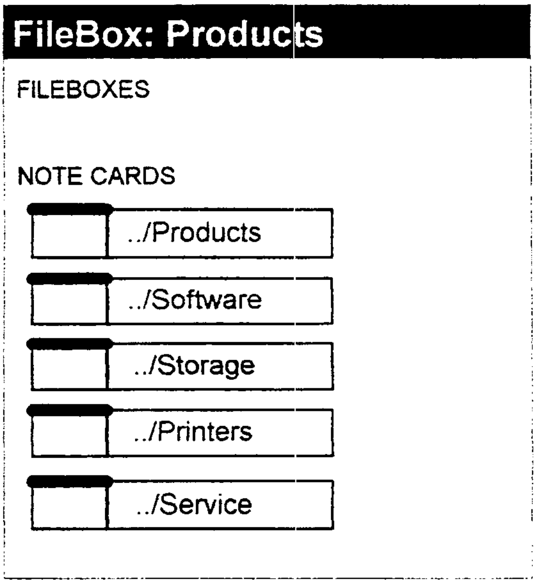
Text Card: ../Products

```
<HTML><HEAD><TITLE>Acme Corporation Products</TITLE>
<A HREF="/Software/" Title="Software" </A><BR>      ../Software
<A HREF="/Storage/" Title="Storage" </A><BR>        ../Storage
<A HREF="/Printers/" Title="Printers" <IMG SRC="/"Printer.gif"></A<BR>
../Printers     <A HREF="/Service/" Title="Service"</A<BR>
../Service      </B></BODY></HTML>
```

The text card shown in Figure 6 includes HTML code and NoteCards generated link boxes. The link boxes (shown as named rectangles) are automatically placed in the source text card when a link is created. If the webmaster double clicks on a link box, then the destination card pointed to by the link is opened for editing. In this way, the user can read the contents of a card and follow all links from the card.

Additional documentation is available using filebox and document cards. Filebox cards are containers for other notecards (including other filebox cards). This creates a hierarchical structure that is useful in organizing related ideas. From a documentation standpoint, fileboxes are used to collect all webpages related to a topic. A single notecard can be represented in any number of fileboxes; thus, web pages covering several topics can be placed in several distinct filebox collections. The webmaster can use these filebox collections to keep track of the cards that need to be modified when the website content is updated. A picture of a filebox card for the 'Products' sub-tree is shown in Figure 7.

Figure 7. FileBox Card for the 'Products' Sub-Tree



FileBox: Products

FILEBOXES

NOTE CARDS

../Products

../Software

../Storage

../Printers

../Service

Document cards collect all textual information from the descendant cards of a specified notecard or filebox. This creates a single card that represents the contents of a portion of the NoteCards data file. This is a powerful documentation tool. The webmaster can collect the contents of the full website, a sub-tree of the full website, or those notecards related to a single topic via a filebox collection. Document cards can be printed (for a hard copy record) or included in a filebox (to document the contents of the filebox). Since document cards are a special type of text card, the built-in text editor can be used to locate specific text strings for maintenance.

Taken together, the analysis and documentation capabilities of the NoteCard technique described above provide the webmaster with a powerful set of tools. These tools will help the webmaster locate structural problems that currently exist and avoid the creation of new problems when website maintenance is required. This will have the effect of creating websites with good structure that are easier to use and easier to update.

DISCUSSION

The website analysis and documentation technique described by this paper has a number of advantages over the existing ad hod and software methods.

- The structural problems associated with the traditional approach to website design are corrected. The technique can be used to analyze existing websites to locate dangling links, orphaned subtrees, loops, and multi-node cycles. Likewise, a system provided feature of NoteCards identifies orphaned nodes. Once these problems are located, the webmaster can correct them (or avoid them altogether). This results in websites that are easier to navigate.
- The technique is visually based. This makes it possible for the webmaster to comprehend the website structure quickly when maintenance is needed. If a site map is provided to the end-suer, they also benefit because the vidual image of the website layout will make information retrieval more efficient.
- The technique is on-line. All data associated with the technique is stored in a single NoteCards file. This file is easy to use and maintain. Parts of the information can even be imported directly from the web pages. Because of this, it is much easier to keep the website documentation up-to-date.
- NoteCards is a dynamically expandable environment. As website technology advances, the tools can be modified and enhanced to accommodate these changes. More importantly, the tool can be individually customized by the end-user to accommodate specific needs. Current software tools intended to aid webmasters manage websites are static and take the *one-size-fits-all* approach. The NoteCards technique is not limited in this way.

The technique also has some limitations. These are as follows:

- The technique requires the webmaster to create the original website structure by hand. This is a tedious, error-prone task that many webmasters would avoid.

- NoteCards is not currently *web-enabled.* This means that you cannot log in to the internet via NoteCards. On a related note, NoteCards is not a browser. It does not have the capability of displaying a web page as the user would see it or following hypertext links to different internet addresses.
- NoteCards is a relatively closed environment written in InterLisp-D. This means that NoteCards does not interface will with most operating systems and commercially available software. The UNIX version of NoteCards provides the best external access.

The first two disadvantages can be corrected with the addition of an internet package for the InterLisp environment. The InterLisp environment can be expanded by loading low-level routines called packages. An internet package will allow InterLisp (and consequently NoteCards) to access the world-wide web and process HTML code. Once this is in place, the first disadvantage can be addressed. Unfortunately, the problem, concerning the closed nature of NoteCards is inherent in all lisp-based products. This is not likely to be solved any time soon.

FUTURE RESEARCH

The research presented in this paper has several potential extensions. First, the analysis and documentation technique has only been applied to relatively small, simple websites. In the future, the technique will be applied to larger sites to determine how well it scales-up to more complex situations. Another future enhancement will be to add an additional property to the user-defined web links to hold information concerning how many times the link is accessed. This information can be used by the browser card to identify the low-use access paths for removal. It can also be used to recognize the high-use access paths so the webmaster can place them in a more prominent position on the web page. Information for the link property will have to be imported into NoteCards from some externally collected file.

If an internet package for InterLisp can be acquired, the technique could be enhanced to automatically build the initial website structure. This will be done by using an internet *robot.* A robot is a program that recursively follows all internet hypertext links in a website. Along the way, it can be programmed to collect information about the web pages it encounters. Most commercial web search engines use an indexed data file created by a robot application. For the purposes of this research, the robot will use NoteCards primitive functions to create text cards and links within a browser card. The application will be designed to import the contents of the web pages directly into the note cards as the structure is being built. Performing this research will turn the technique into an active hypertext visualization tool that Rhyne (1995) calls an *agent.* This enhancement will make the technique more viable by removing the difficulty associated with manually creating the initial web structure.

SUMMARY

The connectivity provided by the internet has had a profound impact in the way we so business. Information has become a fluid commodity and the corporate website has become the virtual storefront of the future. The success of corporate websites depends, to a large extent, on the internal structure and ease-of-use of the site. Poorly structured websites frustrate end-suers and generate negative feelings about the organization that created them. Well-structured websites can have the opposite effect.

The traditional way of designing and maintaining corporate websites leads to a variety of structural problems. These problems are difficult to locate and correct using the ad hoc methods that currently exist. This paper seeks to correct this situation by introducing and describing a new technique to analyze and document websites. The technique uses the NoteCards hypertext environment to build a visual model of the website. The model is semantically rich, dynamically extensible, and allows interactive update. The result of using this technique is a website that is easier to use, easier to update, and fully documented.

REFERENCES

Halasz, F. G., Moran, T. P., & Trigg, R. H. (1987, April). NoteCards in a nutshell. *Proceedings of the ACM CHI + GI '87 Conference,* Toronto, Canada.

NoteCards Users Guide. (1991, August). Release 2.0, Venue Corporation, San Carlos, CA.

Rhyne, T. (1995, July). Scientific visualization and technology transfer: An EPA case study. *IEEE Computer, 28*(7), 94-96.