



Harlie D. Frost  
President

SBC Intellectual Property  
6500 River Place Boulevard  
Building III, 1<sup>st</sup> Floor  
Austin, TX 78730  
Phone 512 - 231-7000

January 10, 2003

Ms. Marilynne Eichinger  
President - Museum Tour  
2525 S. E. Stubb Street  
Milwaukie, OR 97222

Re: US Patent No. 5,933,841 and US Patent No. 6,442,574

Dear Ms. Eichinger:

SBC Intellectual Property is the owner of US Patent No. 5,933,841, entitled *Structured Document Browser* (the '841 Patent) and US Patent No. 6,442,574, also entitled *Structured Document Browser* (the '574 Patent). The purpose of this letter is to advise you of an opportunity to acquire a license to the '841 and '574 Patents, copies of which are enclosed for your review.

We recently observed several useful navigation features within the user interface of your [www.museumtour.com](http://www.museumtour.com) website. For example, your site includes several selectors or tabs that correspond to specific locations in your site document. These selectors seem to reside in their own frame or part of the user interface. And, as such, the selectors are not lost when a different part of the document is displayed to the user ([www.museumtour.com](http://www.museumtour.com) screen shots are enclosed for your convenience). By separating the selectors from the content, Museum Tour has truly simplified site navigation and improved the shopping experience for its users.

As you review the Structured Document Portfolio, you will notice that the above-discussed features (as well as other valuable features) appear to infringe several issued claims in both the '841 and '574 Patents. In light of Museum Tour's presumed respect for the intellectual property rights of others, we are pleased to offer you a Preferred Rate license under the Structured Document Portfolio (see enclosed Rate Schedule and Licensing Agreement). Actual licensing amounts are calculated using SBC's then current Royalty equation. SBC will gladly determine your actual licensing amount if you provide us with your 2002 Company Gross Revenue.

For your convenience and to assist your analysis of the Portfolio, I have enclosed a document that charts the claim language of an example claim (Claim 13 of the '574 Patent) to your site structure. Based on our current understanding of your company's site, several additional claims also appear relevant to the analysis (see, e.g., Claims 1, 4, 6, 7, 16 and 20, of the '574 Patent and Claims 1, 5, 15, 24, 25 and 30, of the '841 Patent).

After reviewing the patents and determining whether you prefer a prepaid license or a recurring annual license, please contact Ms. Jill Walker at 512.231.7008 to determine your actual royalty amount. If you have your 2002 Gross Revenue number, Ms. Walker will provide you with an accurate royalty number and will facilitate the execution of an appropriate license. Thank you for your prompt attention.

Sincerely,

A handwritten signature in dark ink, appearing to read "Harlie D. Frost", with a long horizontal flourish extending to the right.

Harlie D. Frost  
President

Enclosures:

www.museumtour.com screen shots

Copy of US Patent No. 5,933,841

Copy of US Patent No. 6,442,574

License Rate Schedule for Catalog/Online Sales Industry (effective 01/01/03 to 06/30/03)

Licensing Agreement (HDF.000.NE.841)

**SUBJECT TO FEDERAL RULE OF EVIDENCE 408**

**License Rate Schedule**  
**--Catalog/Online Sales Industry--**

**Preferred Rate -- Standard License**

(Effective January 1 to June 30, 2003)

<b>2002 Company Revenue (US \$)</b>	<b>Annual Royalty (US \$)</b>	<b>Fully Prepaid License (US \$)</b>
\$ 100,000	\$ 527	\$ 1,581
\$ 1,000,000	\$ 2,641	\$ 7,924
\$ 10,000,000	\$ 13,239	\$ 39,716
\$ 100,000,000	\$ 66,351	\$ 199,054
\$ 1,000,000,000	\$ 332,544	\$ 997,631
\$ 10,000,000,000	\$ 1,666,667	\$ 5,000,000

**Base Rate**

(Effective January 1 to June 30, 2003)

<b>2002 Company Revenue (US \$)</b>	<b>Annual Royalty (US \$)</b>	<b>Fully Prepaid License (US \$)</b>
\$ 100,000	\$ 5,270	\$ 15,811
\$ 1,000,000	\$ 26,415	\$ 79,245
\$ 10,000,000	\$ 132,388	\$ 397,164
\$ 100,000,000	\$ 663,512	\$ 1,990,536
\$ 1,000,000,000	\$ 3,325,437	\$ 9,976,312
\$ 10,000,000,000	\$ 16,666,667	\$ 50,000,000

- The above numbers provide a general indication of the rate structure. To determine the actual Annual Royalty and Fully Prepaid License amounts, please provide SBC with your 2002 Company Revenue numbers. The amounts in question are calculated using SBC's then current Royalty equation



- **SUBJECT TO FEDERAL RULE OF EVIDENCE 408**

United States Patent No. <u>6,442,574</u>	Museum Tourwebsite
13. A browser for navigating a document comprising a plurality of sections, the browser comprising:	A user accesses www.museumtour.comwebsite with a Web browser
A display window for displaying a document; and	<p>The user interface of www.museumtour.comwebsite includes a document frame in which selected portions of the website document are displayed</p> <p><i>(see Enclosed screen shots)</i></p>
A user interface comprising a plurality of selectors automatically configured to correspond to a respective plurality of sections of the document regardless of what section of the document is being displayed in the display window;	<p>The user interface of website www.museumtour.com includes a selector section that includes selectors for navigating through the website document. The selectors remain in the selector section regardless of the document portion displayed in the document frame</p> <p><i>(see Enclosed screen shots)</i></p>
Wherein the plurality of selectors are not part of the document displayed in the display window of the browser and continue to be displayed after one of the plurality of selectors is selected.	<p>The selectors remain in the selector section even after a user selects one to change the document portion displayed in the document frame</p> <p><i>(see Enclosed screen shots)</i></p>

# Museum Tour®

Home Arts & Music Books and Videos Construction Creative Play History Nature Puzzles and Games Science Technology

Monday January 06, 2003

Architecture

Cars, Boats and Trains

Construction

## Product Search



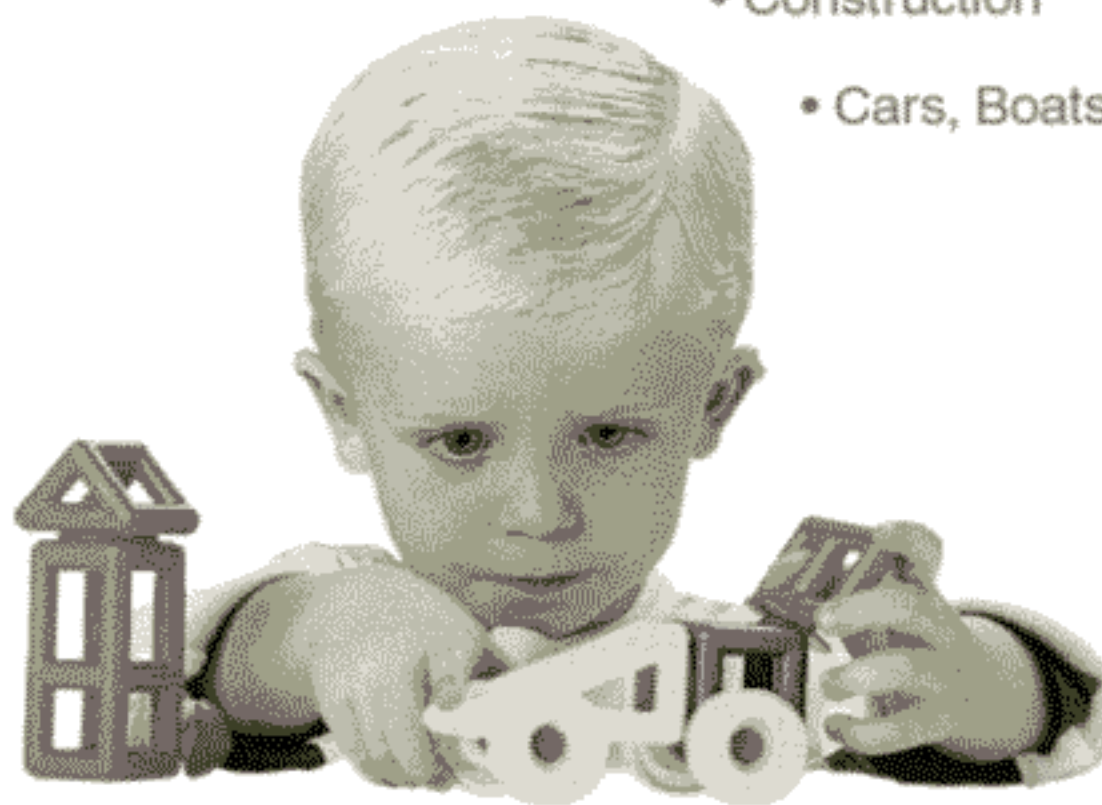
**Watch  
Lance Burton  
in action!**

Subscribe to the Museum  
Tour E-Zine!

## Site Map:

- ▶ [Home](#)
- ▶ [Arts & Music](#)
- ▶ [Books and Videos](#)
- ▼ [Construction](#)
  - ▶ [Architecture](#)
  - ▶ [Cars, Boats and Trains](#)
  - ▶ [Construction](#)
- ▶ [Creative Play](#)
- ▶ [History](#)
- ▶ [Nature](#)
- ▶ [Puzzles and Games](#)
- ▶ [Science](#)
- ▶ [Technology](#)

- [Architecture](#)
- [Construction](#)
- [Cars, Boats and Trains](#)



The first elevator was built in 1743. It was called the "Flying Chair" and worked by weights, pulleys and manpower. Unfortunately, it could only make it up one floor.

**Teachers!**  
**Click Here**  
**Classroom**  
**Science**  
**Kits**

**Closeouts**

# Museum Tour®

Home Arts & Music Books and Videos Construction Creative Play History Nature Puzzles and Games Science Technology

Monday January 06, 2003

Lance Books Calendars Learning to Read Videos

## Product Search



**Watch  
Lance Burton  
in action!**

Subscribe to the Museum  
Tour E-Zine!

### Site Map:

- ▶ [Home](#)
- ▶ [Arts & Music](#)
- ▼ [Books and Videos](#)
  - ▶ [Lance](#)
  - ▶ [Books](#)
  - ▶ [Calendars](#)
  - ▶ [Learning to Read](#)
  - ▶ [Videos](#)
- ▶ [Construction](#)
- ▶ [Creative Play](#)
- ▶ [History](#)
- ▶ [Nature](#)
- ▶ [Puzzles and Games](#)
- ▶ [Science](#)
- ▶ [Technology](#)

**Teachers!  
Click Here  
Classroom  
Science  
Kits**

**Closeouts**

## • Learning to Read



## • Books

## • Videos

## • Calendars

**"A library is the delivery room  
for the birth of ideas,  
a place where history  
comes to life."  
Norman Cousins**



## United States Patent [19]

Schumacher et al.

[11] Patent Number: 5,933,841

[45] Date of Patent: \*Aug. 3, 1999

## [54] STRUCTURED DOCUMENT BROWSER

[75] Inventors: Robert M. Schumacher, Wheaton;  
James E. Matthews, Chicago, both of  
Ill.[73] Assignee: Ameritech Corporation, Hoffman  
Estates, Ill.

[\*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: 08/649,271

[22] Filed: May 17, 1996

[51] Int. Cl.<sup>6</sup> ..... G06F 17/21

[52] U.S. Cl. .... 707/501; 707/513; 345/354

[58] Field of Search ..... 395/761, 762,  
395/773, 774, 776, 777, 335, 347, 348-349,  
354; 345/172, 335, 347, 348, 439, 354,  
350, 352; 707/500, 501, 512, 513, 514,  
515, 517

## [56] References Cited

## U.S. PATENT DOCUMENTS

4,616,336	10/1986	Robertson et al.	395/773
4,752,908	6/1988	Bouillot .	
4,829,453	5/1989	Katsuta et al.	395/773 X
5,021,989	6/1991	Fujisawa et al.	345/350
5,129,082	7/1992	Tirring et al.	
5,140,521	8/1992	Kozol et al.	
5,142,678	8/1992	MacPhail	395/761
5,146,552	9/1992	Cassola et al.	395/773
5,204,947	4/1993	Bernstein et al.	
5,278,980	1/1994	Pedersen et al.	
5,297,249	3/1994	Bernstein et al.	
5,329,111	7/1994	Sonoda et al.	
5,331,547	7/1994	Laszlo .	
5,339,091	8/1994	Yamazaki et al.	
5,339,433	8/1994	Frid-Nielsen .	
5,392,387	2/1995	Fitzpatrick et al.	

5,404,506	4/1995	Fujisawa et al.	
5,428,776	6/1995	Rothfield .	
5,432,903	7/1995	Frid-Nielsen .	
5,530,852	6/1996	Meske, Jr. et al.	395/600
5,557,722	9/1996	DeRose et al.	395/762 X
5,572,643	11/1996	Judson	395/793
5,623,679	4/1997	Rivette et al.	707/526
5,625,781	4/1997	Cline et al.	395/335
5,634,064	5/1997	Warnock et al.	395/774
5,708,826	1/1998	Ikeda et al.	707/513 X
5,774,109	6/1998	Winsky et al.	707/501 X

## OTHER PUBLICATIONS

Classified Search and Image Retrieval Student Manual for the Automated Patent System (APS), sections 1-4, 6-7, 9, May, 1991.

Fowler et al., "Visualizing and Browsing WWW Semantic Content", Emerging Technologies and App'ns in Communication, 1996 Conference, pp. 110-113, 1996.

Harger, "Introducing DSP with an Electronic Book in a Computer Classroom", IEEE Transactions on Education, v38, n2, pp. 173-179, May 1996.

Gershon, "Moving Happily through the World Wide Web", IEEE Computer Graphics and App'ns Magazine, v16 n2, pp. 72-75, Mar. 1996.

Simpson, "Mastering WordPerfect 5.1 &amp; 5.2 for Windows", pp. 510-535, 1993.

Primary Examiner—Joseph H. Feild

Attorney, Agent, or Firm—Brinks Hofer Gilson &amp; Lione

## [57] ABSTRACT

A structured document browser includes a constant user interface for displaying and viewing sections of a document that is organized according to a pre-defined structure. The structured document browser displays documents that have been marked with embedded codes that specify the structure of the document. The tags are mapped to correspond to a set of icons. When the icon is selected while browsing a document, the browser will display the section of the structure corresponding to the icon selected, while preserving the constant user interface.

30 Claims, 16 Drawing Sheets

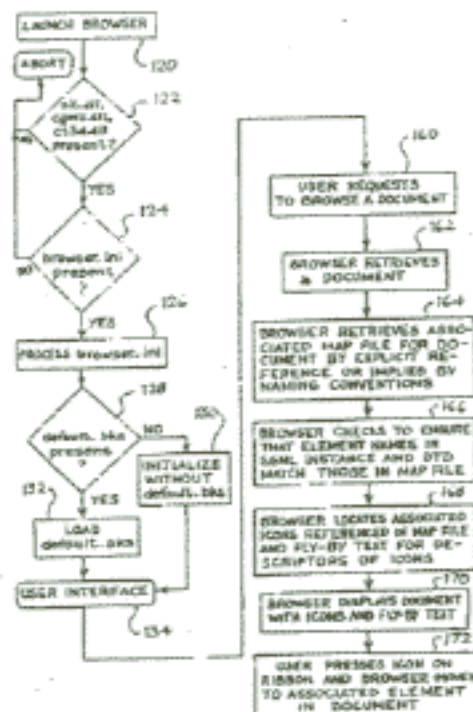
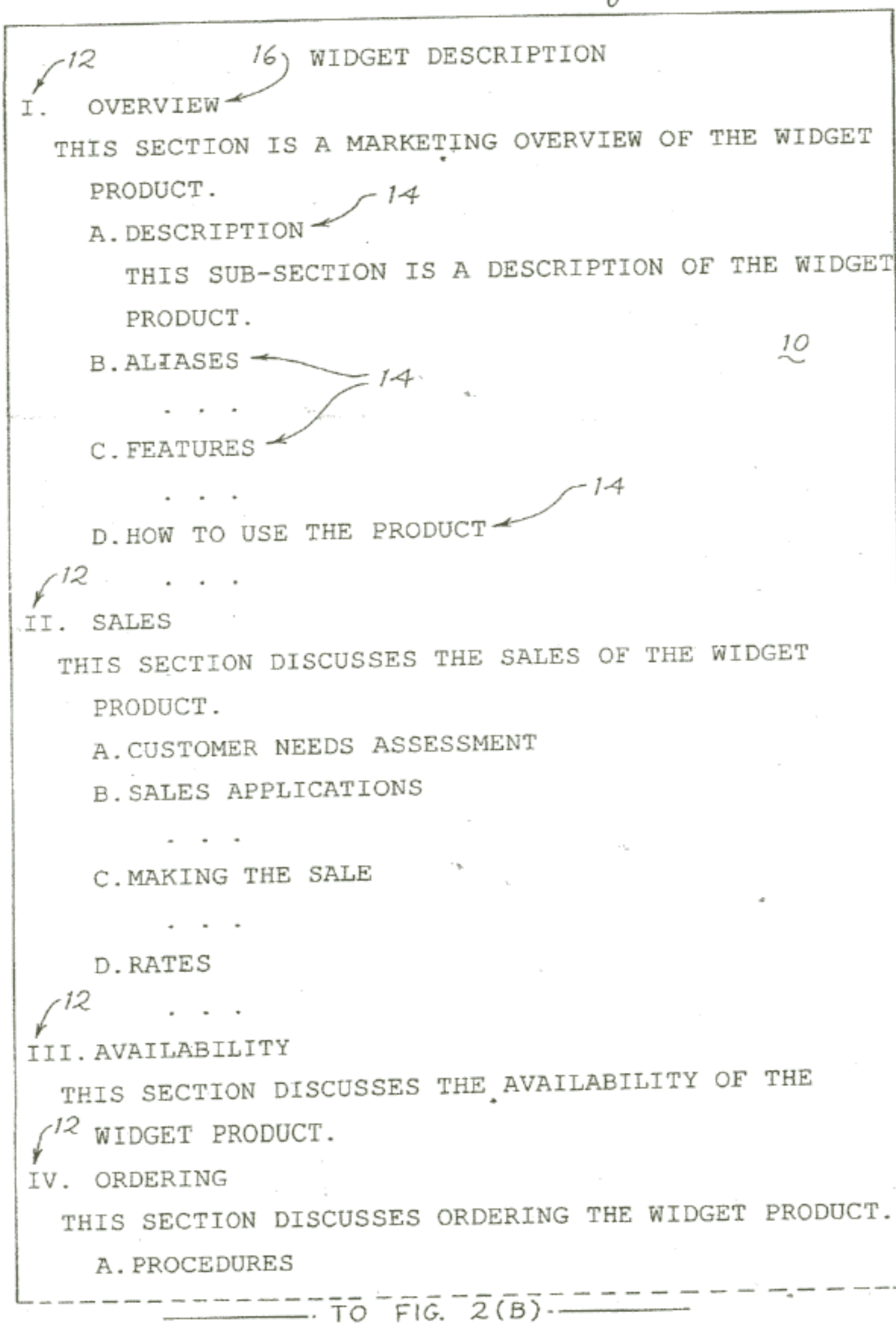


Fig. 2(A)





*Fig. 2(c)*

TO FIG. 2(B)

.....  
 C. TARIFFS & REGULATORY

.....  
 D. POLICIES

.....  
 E. TRAINING

*Fig. 3(A)*

<BRW.DOC> 22  
 <PROD.NAME> 24  
 <PRODUCTNAME>WIDGET1</PRODUCTNAME>  
 <OVER> 26 28 30 32 20  
 <OVERVIEW>OVERVIEW</OVERVIEW>  
 TEXT IN OVERVIEW SECTION.  
 </OVER> 34  
 <SALE> 36  
 <SALES>SALES</SALES>  
 TEXT IN SALES SECTION.  
 <CUSNEED> 38  
 <CUSTOMERNEEDS>CUSTOMER NEEDS  
 ASSESSMENT</CUSTOMERNEEDS>  
 TEXT IN CUSTOMER NEEDS ASSESSMENT SUB-SECTION  
 OF THE SALES SECTION.  
 </CUSNEED> 42  
 </SALE> 39

TO FIG. 3(B)

Fig. 4(A)

```
<!ELEMENT PROD.NAME - - ( PRODUCTNAME, (%SUB-SECT; |  
    TROUBL |  
    ORDER |  
    AVAIL |  
    SALE |  
    BILL |  
    SUPP |  
    OVER)* ) >  
<!ATTLIST PROD.NAME  
    ID CDATA #IMPLIED >  
<!ELEMENT PRODUCTNAME - - ( (#PCDATA | %PARA-  
    CONTENT;)* ) >  
<!ELEMENT TROUBL - - ( TROUBLESHOOTING, (%SUB-SECT; |  
    TRBL.KNOWPROB |  
    TRBL.PROBASSESS |  
    TRBL.DIAG |  
    TRBL.QUES |  
    HEAD3 |  
    HEAD2 |  
    HEAD5 |  
    HEAD4 |  
    HEAD7 |  
    HEAD6)* ) >  
<!ATTLIST TROUBL  
    ID CDATA #IMPLIED  
    ID CDATA #IMPLIED >  
<!ELEMENT TROUBLESHOOTING - - ( (#PCDATA | %PARA-  
    CONTENT;)* ) >
```

Handwritten annotations in Fig. 4(A):

- 42: Arrow pointing to the `PROD.NAME` element declaration.
- 46: Arrow pointing to the `PROD.NAME` element declaration.
- 47: Arrow pointing to the `PRODUCTNAME` element declaration.
- 49: Arrow pointing to the `PROD.NAME` element declaration.
- 44: Arrow pointing to the `PRODUCTNAME` element declaration.
- 45: Arrow pointing to the `PRODUCTNAME` element declaration.
- 48: Arrow pointing to the `TROUBLESHOOTING` element declaration.
- 40: Underlined annotation next to the `TROUBLESHOOTING` element declaration.
- 51: Arrow pointing to the `TROUBLESHOOTING` element declaration.

- TO FIG. 4(B) -

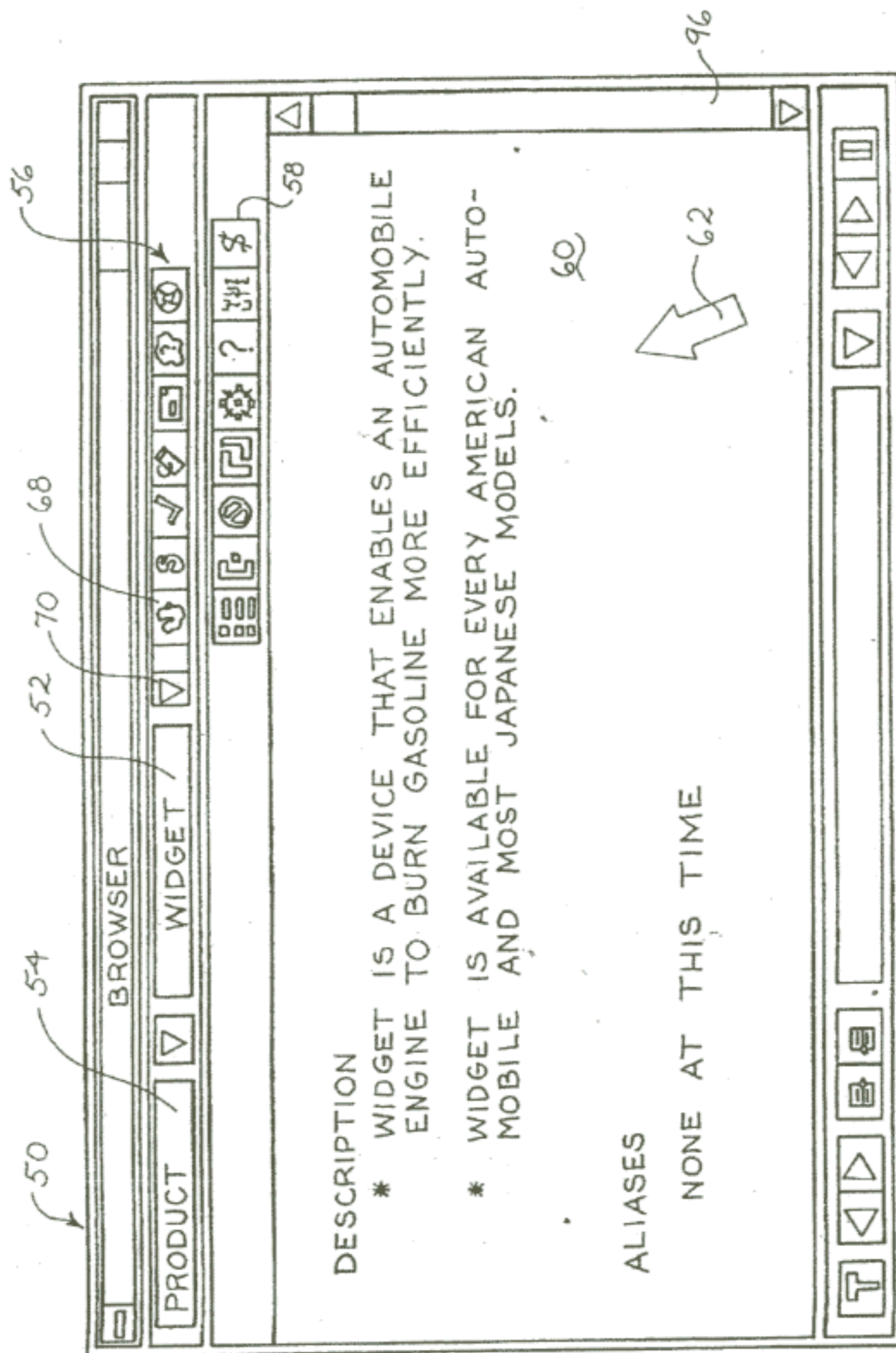
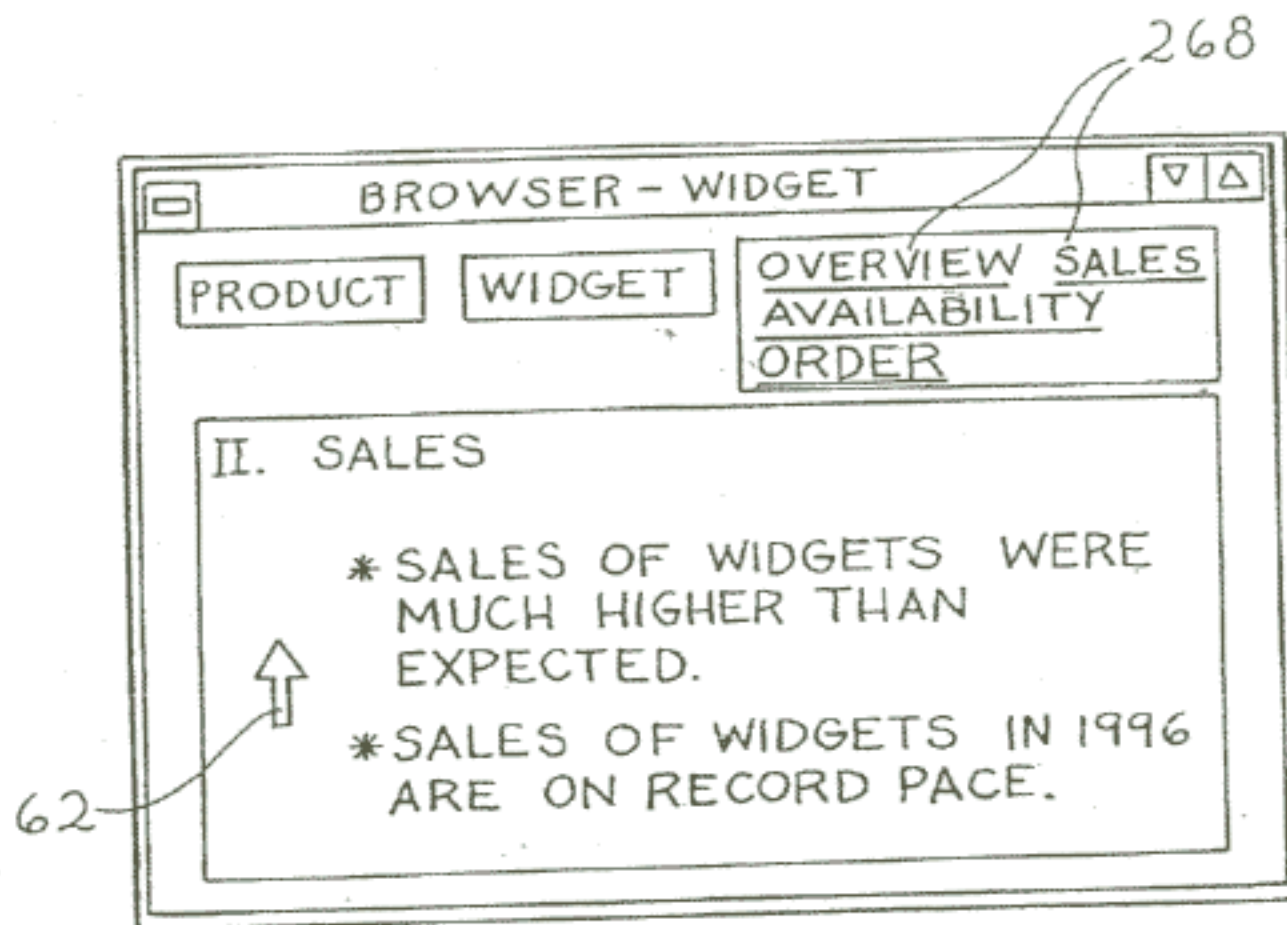
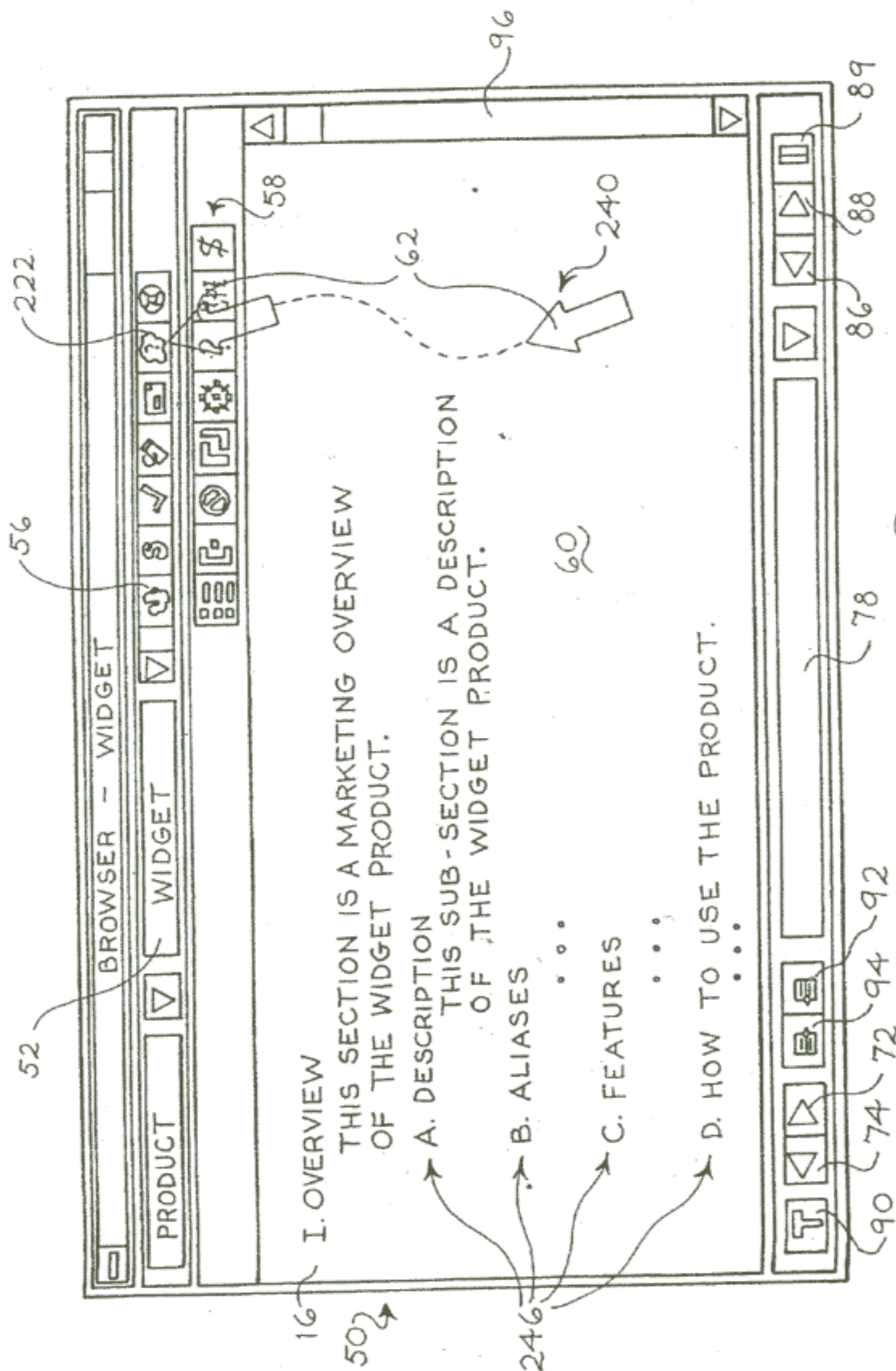


Fig. 5



[BOOK] 140  
142  
144 COLLECTION=\XYZ  
146 COLLECTIONTITLE+"PRODUCTS"  
148 BOOKNAME=WIDGET  
150 BOOKTITLE="WIDGET"  
152 MAP=PRD.MAP

*Fig. 7**Fig. 10*



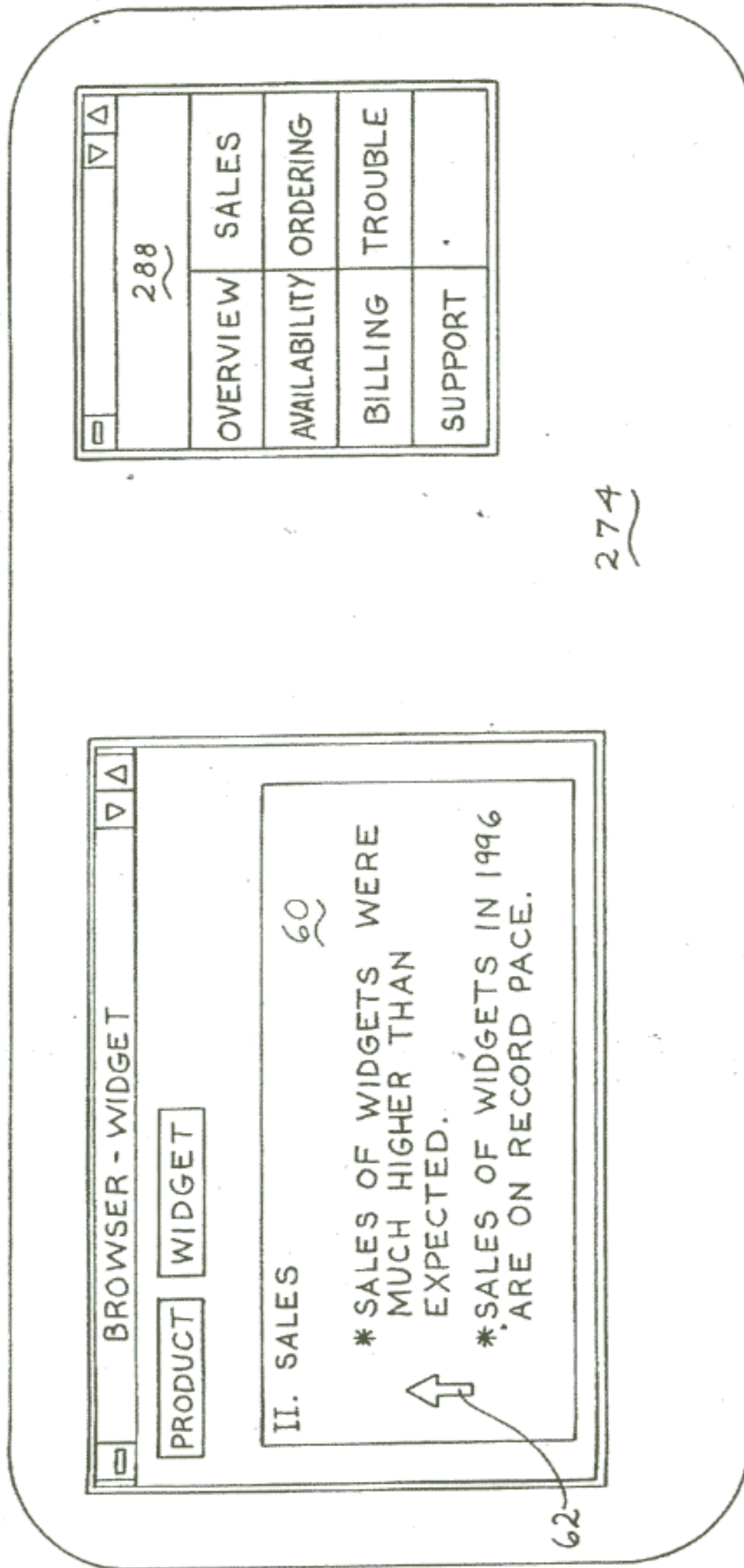


Fig. 11



Fig. 13

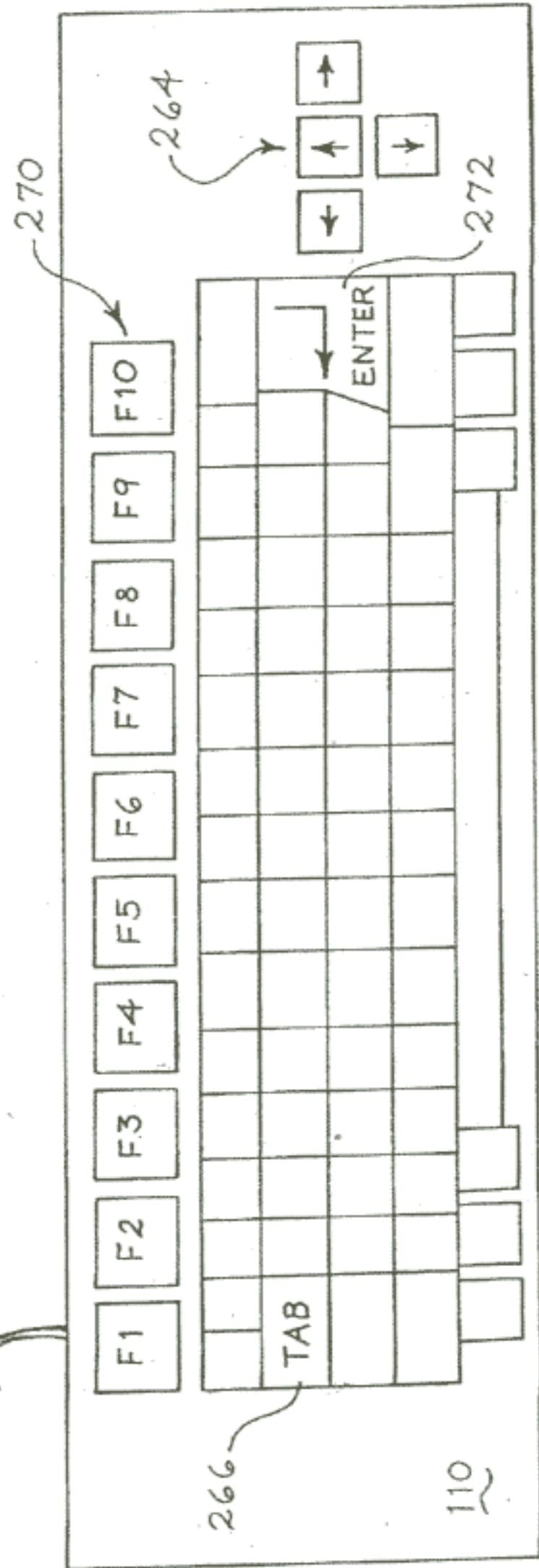
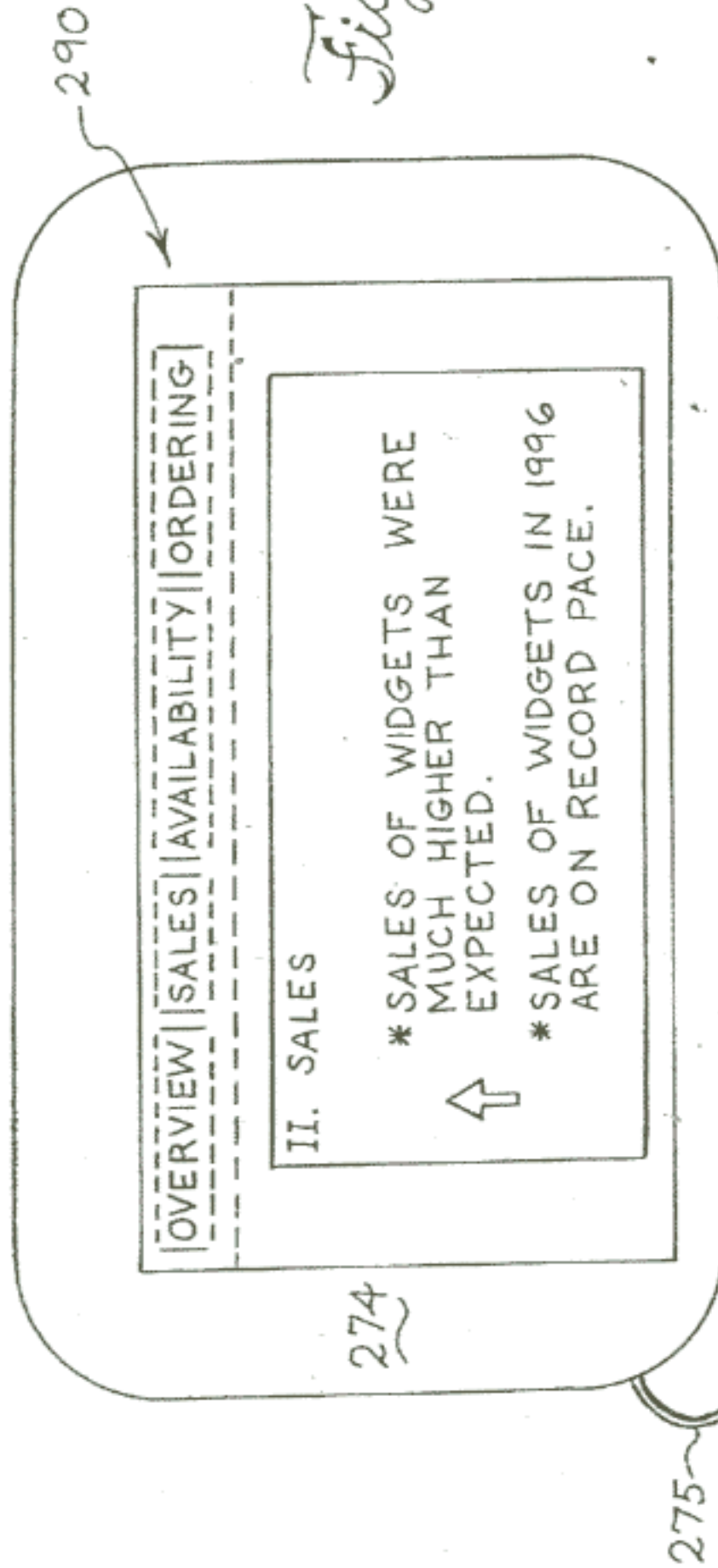




FIG. 7 illustrates an example of a bks data structure.

FIG. 8 is a diagram that shows the interaction between a button, a map file and bit map file.

FIGS. 9A & 9B illustrate the operation of the browser of FIG. 1.

FIG. 10 illustrates one example of an alternative implementation of the user interface.

FIG. 11 illustrates a second alternative implementation of the user interface.

FIG. 12 illustrates a third alternative implementation of the user interface.

FIG. 13 illustrates a fourth alternative implementation of the user interface.

#### DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

In the description that follows, reference is made to the drawings where like elements are identified by like numerals throughout.

A presently preferred embodiment of this invention includes an application program called browser.exe that has been developed using the 'c' programming language in the Windows environment. The browser.exe executable file is programmed to make function calls to three dynamic link libraries named sit.dll, cgrmzv.dll and ct13d.dll. These libraries are components of Dynatext Version 2.0, a user interface development system from Electronic Book Technologies, Inc. These Dynatext libraries provide functions that implement the SGML related functions and the graphic input/output functions. Further information regarding the Dynatext program may be obtained by contacting Electronic Book Technologies, Inc. at One Richmond Square, Providence, R.I. 02906.

The browser.exe program uses data structures in several support files that are in the same directory tree as browser.exe. These support files will be described in more detail in the description that follows.

A listing of the present version of browser.exe is attached as Appendix I of this specification. The listing is an octal representation of browser.exe. The presently preferred embodiment may be carried out by converting the octal to a binary executable file using methods that are well known in the art. After conversion, the browser.exe file may be executed from a directory that includes the dynamic link libraries and the support files described in this disclosure.

It is to be understood however, that an embodiment of the present invention may be developed for any computing environment using any suitable development system.

The browser application in a presently preferred embodiment is referred to in the following detailed description as the browser 80 as shown in FIG. 1. FIG. 1 is a block diagram describing at a high level the browser 80 in the computing environment. The components of the browser 80 in its operating environment include the browser 80 itself, an operating system 104 with GUI capabilities, storage media 108, a keyboard 110, a pointing device 100, a selecting device 102 and a monitor 112. The operating system 104 further includes an I/O system 114 and a GUI system 116.

The double headed arrows 118 denote the communication between the respective components. The communication at 118 may entail communication over a network where appropriate.

The hardware devices 108, 110, 100, 102, 112 may be implemented by choosing from among many alternatives for

each device. A pointing device 100 may be implemented using a mouse, a trackball or any other device that controls the position of a screen pointer. A selecting device 102 is typically implemented with mouse buttons or buttons that operate in conjunction with a trackball. In general, any device that may be used to affect the selection of an object at the location of the screen pointer may be used as a selecting device 102. A selecting device 102 may even include a key on the keyboard 110. The storage media 108 is understood to include random access memory (RAM), the temporary storage out of which programs are executed as well as the mass storage devices in which programs are stored. The hardware devices 108, 110, 100, 102, 112 are understood to include the software drivers necessary for their operation in the computing environment.

The browser 80 includes at least a controller 82 and a system interface 84. The system interface 84 is responsible for processing the communication between the browser 80 and the operating system 104, the I/O system 114 and the GUI system 116. The controller 82 receives and interprets requests from the system interface 84 to perform a browser function. For example, the system interface 84 receives signals from the I/O system 114 that the pointing device 100 and the selecting device 102 were used to press a button or icon, to request a display of a section of a document. The controller 82 receives the information from the system interface 84 to determine which document section to display.

In a presently preferred embodiment, the system interface 84 includes the functions provided by the Dynatext development system and any operating system or I/O system functions. The controller 82 in a preferred embodiment is the executable program browser.exe. It is to be understood that the diagram in FIG. 1 is by way of illustration and is not intended to limit the software structure chosen to carry out the invention.

The browser 80 operates with documents that have been prepared as described below. Because the browser 80 is designed to navigate documents according to their structure, the utility of the browser 80 is maximized when an organization establishes a standard structure for its key documents. A software engineering group, for example, may find it desirable to maintain a uniform structure for the software requirement specifications that the group develops. A different structure is desirable for the group's design specifications, and yet a different structure works for the group's test documentation. The group's goal for such documentation is to maintain uniformity.

Referring to FIG. 2, a marketing group might maintain product descriptions for its company's product line in documents having the predefined document structure 10. This structure is predefined to have headings 12 that provide overview information, sales information, product availability information, ordering information, billing information, troubleshooting information and product support information. The structure also has sub-headings 14 within each heading where relevant. The overview section 16 has sub-headings 14 for sections devoted to a product description, aliases, product features and instructions on how to use the product.

A specific document of the predefined structure in FIG. 2 is marked with codes for viewing with the browser 80. In a preferred embodiment, codes are used to mark the document as shown in FIG. 3. The codes shown in FIG. 3 have angle brackets around them. In a preferred embodiment, the codes are SGML elements. These codes may be replaced by elements of other markup tools in alternative embodiments.



The selectors 56, 58 are examples of document navigation tools that may be used for browsing documents in a presently preferred embodiment. More specifically, selectors 56, 58 are display regions in the user interface that are configured to perform pre-defined operations when the user places the screen pointer 62 over one of the selectors 56, 58 and then selects it with the selecting device 102. In the presently preferred embodiment, the display regions are depicted as icons that make them look like buttons with graphic images on them. The image may be designed to convey a sense of the operation to be performed if the icon is selected.

The first row of selectors 56 is configured to correspond to the first level of sections in a document. For example, the first selector 68 in the first row 56 is configured to correspond to the Overview section 16 in the document shown in FIG. 2.

Each selector is configured so that when the user places the screen pointer 62 over the selector and then selects it with the selecting device 102, the system interface 84 (described above with reference to FIG. 1) receives data indicating which selector was selected. The system interface 84 may then determine the document section associated with the selector and send the request to display that section to the controller 82 (described above with reference to FIG. 1). Alternatively, the system interface 84 may send to the controller 82 the identification of the chosen selector and let the controller 82 determine what section to display. The controller 82 will cause the document to be searched for the document section that matches the selector chosen. When the chosen section is found, the browser 80 displays in the display window 60 the section of the document structure that corresponds to the selector.

Once the desired section is in the display window 60, the user may navigate within the section by selecting one of the second row of selectors 58. Each time a selector from the first row of selectors 56 is selected, the second row of selectors 58 is configured to correspond to the sub-sections 14 within the section 12 being displayed (as shown in FIG. 2). A document structure having sub-sections within its sub-sections may also be accommodated so that three rows of selectors might be present in the user interface. The number of sub-sections within sections of a document may be further accommodated with rows of selectors as desired, or as limited by system constraints, such as the size of the display window 60.

The user may also navigate within the section by controlling the display window scroll bar 96 with the screen pointer 62 and the selecting device 102.

To describe the manner in which the selectors 56, 58 in the browser user interface 50 are configured, the initialization of the browser will be described in conjunction with exemplary files or data structures that are utilized during browsing operations. It is to be understood that this is only one implementation of the preferred embodiment, and that the files may be replaced, or integrated, or revised to form different data structures without departing from the scope of the invention. Furthermore, the Dynatext development tools may be replaced by other functionally equivalent tools.

In the presently preferred embodiment, documents are converted into "books" which are actually directory trees that reside in a directory called the "xyz\books" directory. The terms "book," "collection" and "library" are defined and used according to the specifications of the Dynatext development system.

In order to create the Dynatext books, the documents that have been coded as illustrated in FIG. 3 are supplied as input to a utility called DYNATAG which is a component of the Dynatext system. DYNATAG creates a DTD (as shown in FIG. 4) and an SGML instance (as shown in FIG. 3) of the

document. The SGML instance and the DTD are used as input files to MKBOOK, another Dynatext utility. The MKBOOK utility creates a binary instance of the document, a directory tree, or a "book," and some of the support files used by the browser 80. For example, the document for Widget having the structure for products descriptions defined in FIG. 2 is processed with MKBOOK to create the book "xyz\books\widget," a sub-directory of "xyz\books."

The browser 80 uses a number of data files to define how a document is found and displayed in the browser. These files and their names in the preferred embodiment are 1) the browser executable (browser.exe); 2) an initialization file (browser.ini); 3) a set of dynamic link libraries (sit.dll, cgmzv.dll, & ct13d.dll); 4) a bks file which is an ASCII file that contains information about a book, a library or collection of documents, and names in the browser menus (named \*.bks where the \* represents the name of a document); 5) a bitmap containing up to 100 regions for icons (named \*.bmp or default.bmp) and 6) an ASCII file that provides the linking of the document element names to the icons in the bmp file and for pop-up text in the executable (named \*.map or default.map).

The execution of the browser 80 will be described with reference to FIG. 6 which is a flowchart describing the steps taken to browse a document. When the user starts the structured document browser in the manner described above in reference to FIG. 1, the operating system launches the browser.exe executable file as shown at block 120. This file is located in a directory called the "xyz" directory. When browser.exe is launched it first looks for the three required dynamic link libraries (DLL's) in the same directory as shown at block 122. The DLL's, supplied by Electronic Book Technologies, contain functions related to the user interface, SGML processing and access to the books.

If the DLL's are available, the browser then checks for the browser.ini file as shown in 124. If the browser.ini file is present, the browser 80 reads its contents, as shown at block 126.

As the sample file in Table 1 shows, the browser.ini file contains objects, or data structures that include the [Files] object, the [DTEXT] object, and the [MAP] object. The [FILES] object defines an annotation file. The annotation file is a repository for feedback from users of the browser regarding the documents being reviewed. The [DTEXT] object contains file names that the executable will use to find the location of the data directory, security key, and public and private directories. The [MAP] object provides the file name of the initial map file that is to be loaded (typically, the name is "default.map"). The map file, as discussed in detail below, contains the associations between the document elements and the icons.

TABLE 1

Sample browser.ini File

```
[Files]
AnnotationFile=xyz\annot.txt
[DTEXT]
DATA_DIR=xyz\data
DTEXT_AUTH=@xyz\data/security
PUBLIC_DIR=xyz\temp\public
PRIVATE_DIR=xyz\temp\private
[MAP]
Icons=default.bmp
```

Referring back to FIG. 6, once the browser.ini file is processed at 126, the browser checks for a file called default.bks at block 128. The default.bks file is an ASCII file which provides the browser with the information required to display a standard initial document, such as a document that displays a message of the day.



screen pointer 62 may be moved from an initial position 240 to the position over the icon for the troubleshooting button 222. The user then selects the troubleshooting button 222 by enabling the selecting device 102. The browser 80, using the process illustrated in FIG. 8, then searches through the binary instance of the document for the troubleshooting section. The browser 80 displays the found section 242 in the display window 60 as shown in FIG. 9B. In addition to displaying the found section 242, the browser 80 updates the second row of selectors 58 to correspond to the sub-sections 248 in the found section 242.

The user interface 50 of the browser 80 may be enhanced by adding objects to give the user more tools with which to view the documents. The user interface 50, shown in FIG. 9A, includes a next section button 92, a previous section button 94, a go forward button 72, a go backward button 74 and string search tools 78, 86, 88, 89. These objects may be programmed into the browser 80 along with the software components that provide the indicated functions.

When the next section button 92 is selected with the combined action of the screen pointer 62 and the selecting device 102, a user views the next section in the document. For example, the next section after the Overview section 16 in FIG. 9A is the Sales section (See FIG. 2). Selecting the next section button 92 in FIG. 9A causes the browser 80 to display the Sales section. The previous section button 94 operates in the same manner as the next section button 92 except that the previous section is shown.

The go forward button 72 and go backward button 74 may be used to scroll text in the display window 60.

The user interface 50 as shown in FIG. 9A may also include string search tools 78, 86, 88. The string search entry box 78 may be used to input a text string that the user wishes to locate in the document. The next found and previous found buttons 86, 88 may be used to display the locations in the document in which the string was found. The clear search button 89 clears the text in the search entry box 78.

The feedback entry function gives the user the ability to provide feedback on a document for those who may browse the document at a later time. As shown in FIG. 9B, by selecting a feedback file in a menu, or by selecting a tools button 90, a text box 260 opens up to allow the user to enter a note. The text box 260 may be a compilation of messages to which users append notes, or the contents of the text box 260 may be saved into a separate repository of data periodically. In a preferred embodiment, the compilation of messages may be saved to a SGML-based file for support as a document that may be viewed by implementation of the browser 80. In addition, the textbox 260 may be replaced by a view of the messages in the display window 60.

It is to be understood that the appearance of the user interface 50 shown in FIGS. 9A & 9B is one example of the user interface in the present invention. The appearance and the choice of graphic objects may be varied to suit the needs of the intended users.

Referring to FIG. 10, one example of how the user interface 50 may be altered replaces the buttons 56, 58 with other objects. The buttons are merely display regions of the user interface configured to perform a function when selected with the screen pointer 62 and the selecting device 102. In the presently preferred embodiment of FIG. 9A, the selectors are represented by button icons. As shown in FIG. 10, these icons may be replaced with words or phrases 268 that are descriptive of the section that they are configured to display.

Another variation, shown in FIG. 11, uses a distributed user interface in which the buttons 288 are located in their

own window that is detached from the display window 60. FIG. 11 illustrates the separate windows 50, 288 as they might appear on a monitor screen 274.

In another variation shown in FIG. 12, keys on the keyboard 110 may be configured as functional equivalents of the screen pointer 62 to select document section selectors 56, 58. FIG. 12 illustrates a monitor screen 274 connected to a keyboard 110 via connection 275. In one approach to using the keyboard, the selectors 56, 58 may be mapped to function keys 270 on the keyboard 110. In another approach which may be combined with the first approach, the browser 80 may first highlight a selector in response to certain keys such as a TAB key 266, or an arrow key 264, and then select the highlighted selector 280 in response to another key such as the ENTER key 272.

The user interface so may also be implemented in an environment that lacks a GUI, such as a character-based system interface. In an example of such an implementation shown in FIG. 13, the selectors 56, 58 are words or phrases that have features such as a character-based border identifying them as selectors. The user then selects a selector using the keys on the keyboard as described above.

In another example of a character-based user interface 50, the selectors 56, 58 are not used at all and the entire screen is the display window. Function keys 270 on the keyboard 110 are implemented in place of the selectors. The function keys 270 may be mapped according to the labels indicated at 290.

It is to be understood that this specification is provided by way of illustration and that it is only the claims and their equivalents that define the invention.

We claim:

1. In a computer, a browser for viewing documents having embedded codes that identify parts of documents according to at least one predefined document structure, said browser comprising:

a user interface comprising a display window that displays a document to a user;

a plurality of input devices;

a first plurality of display regions that are responsive to said input devices, said display regions of said first plurality being continuously displayed as part of the user interface automatically and configured to correspond to respective parts of the predefined document structure regardless of what part of the document is in the display window; and

a controller operative to cause a selected part of the document to be displayed in the display window when a user uses one of said input devices to enable one of said display regions that corresponds to the selected part.

2. A browser as claimed in claim 1 wherein said plurality of input devices comprises:

a screen pointer that moves on the display window of the user interface and maintains a current position responsive to a user controlled pointing device; and

a selecting device that selects the current position of the screen pointer when enabled by a user;

wherein the controller causes a selected part of the document to be displayed on the display window when a user moves the screen pointer to the display region that corresponds to the selected part and enables the selecting device.

3. A browser as claimed in claim 2 wherein said plurality of input devices further comprises:



23. The method of browsing a document as claimed in claim 20, wherein the first plurality of display regions remain displayed as part of the browsing tool even after selection of a display region.

24. In a computer with graphical user interface capabilities, a method for browsing a document within the context of a predefined document structure comprising the steps of:

- creating one or more documents having the predefined document structure;
- embedding codes in the documents to identify parts of the predefined document structure;
- initializing a browsing tool having document navigation tools that include a first plurality of display regions, said display regions of said first plurality being continuously displayed as part of the browsing tool during browsing and automatically configured to correspond to respective parts of the predefined document structure regardless of what part of the document is displayed;
- displaying a document in a display window; and
- viewing parts of the document by repeating the steps of:
  - moving a screen pointer that maintains a current position on the display window responsive to a user-controlled pointing device over a selected region of the document navigation tools configured to correspond to a corresponding part of the predefined document structure;
  - maintaining said selected display region accessible regardless of what part of the document is in the display window; and
  - selecting the selected region by enabling a selecting device.

25. A method for browsing a document as claimed in claims 15, 20 or 24 wherein the steps for browsing a document further comprises the steps of:

- displaying a next part in the predefined document structure of the document by repeating the steps of:
  - moving the screen pointer over a next part region of the document navigation tools configured to display a part of the document that follows a part of the document that is currently displayed in the display window; and
  - selecting the next part region by enabling the selecting device; and
- displaying a previous part in the predefined document structure of the document by repeating the steps of:
  - moving the screen pointer over a previous part region of the document navigation tools configured to display a part of the document that follows a part of the document that is currently displayed in the display window; and
  - selecting the previous section region by enabling the selecting device.

26. The method of browsing a document as claimed in claim 15, further comprising the step of:

initializing a browsing tool having document navigation tools that include a second plurality of display regions, said display regions of said second plurality configured to correspond to respective parts of the predefined document structure in response to the part of the document that is displayed in the display window.

27. The method of browsing a document as claimed in claim 24, wherein the first plurality of display regions are automatically displayed as part of the browsing tool without prompting by the user.

28. The method of browsing a document as claimed in claim 24, wherein the first plurality of display regions remain displayed as part of the browsing tool even after selection of a display region.

29. In a computer comprising a graphical user interface, a browser having a user interface for viewing documents having embedded codes that identify parts of documents according to a predefined document structure, said browser comprising:

- a display window that displays a portion of the document to a user;
- a screen pointer that moves on the user interface and maintains a current position responsive to a user controlled pointing device;
- a selecting device that selects the current position of the screen pointer when enabled by the user;
- a document menu that lists documents that the user can select for viewing by moving the screen pointer to the document menu and enabling the selecting device;
- a document type menu that lists at least one document type having a predefined structure that can be selected to alter the document menu to list documents that conform to the predefined structure of the document type selectable by moving the screen pointer to the document type menu and enabling the selecting device;
- a first plurality of display regions on the user interface, said display regions of said first plurality accessible to the user and configured to correspond to a respective part of the predefined document structure regardless of what part the document is currently in the display window; and
- a controller operative to cause a selected part of the document to be displayed in the display window when a user moves the screen pointer to the region that corresponds to the selected part and enables the selecting device.

30. A browser as claimed in claim 29, further comprising a second plurality of display regions on the user interface, said display regions of said second plurality configured to correspond to a respective part of the predefined document structure in response to the part of the document that is displayed in the display window.

\* \* \* \* \*

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 5,933,841  
DATED : August 3, 1999  
INVENTOR(S) : Schumacher et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

In column 2, line 9, under "OTHER PUBLICATIONS", please change "v38" to --v39--.



Attest:

*Brenda Moore*  
Attesting Officer

Signed and Sealed this  
Seventeenth Day of April, 2001

*Nicholas P. Godici*

NICHOLAS P. GODICI

Acting Director of the United States Patent and Trademark Office





US006442574B1

(12) **United States Patent**  
Schumacher et al.

(10) Patent No.: **US 6,442,574 B1**  
(45) Date of Patent: **\*Aug. 27, 2002**

(54) **STRUCTURED DOCUMENT BROWSER**

5,339,091 A 8/1994 Yamazaki et al.  
5,339,433 A 8/1994 Frid-Nielsen

(75) Inventors: Robert M. Schumacher, Wheaton;  
James E. Matthews, Chicago, both of  
IL (US)

(List continued on next page.)

(73) Assignee: Ameritech Corporation, Hoffman  
Estates, IL (US)

**OTHER PUBLICATIONS**

(\*) Notice: Subject to any disclaimer, the term of this  
patent is extended or adjusted under 35  
U.S.C. 154(b) by 0 days.

This patent is subject to a terminal dis-  
claimer.

"Frames", <http://home.netscape.com/comprod>, down-  
loaded May 9, 1998.\*

"BrowserWatch", <http://browserwatch.internet.com/news/  
story/netscape29.htm>, downloaded May 9, 1998.\*

Classified Search and Image Retrieval Student Manual for  
the Automated Patent System (APS), sections 1-4, 6-7, 9;  
May 1991.

Fowler et al., "Visualizing and Browsing WWW Semantic  
Content", Emerging Technologies and App'ns in Commu-  
nication, 1996 Conference, pp. 110-113; 1996.

Harger, "Introducing DSP with an Electronic Book in a  
Computer Classroom", IEEE Transactions on Education,  
v39 n2, pp. 173-179, May 1996.

Gerson, "Moving Happily through the World Wide Web",  
IEEE Computer Graphics and App'ns Magazine, v16 n2, pp.  
72-75; Mar. 1996.

Simpson, "Mastering WordPerfect 5.1 & 5.2 for Windows",  
pp. 510-535; 1993.

Primary Examiner—Joseph H. Feild

(74) Attorney, Agent, or Firm—Brinks Hofer Gilson &  
Lione

(21) Appl. No.: 09/304,425

(22) Filed: Apr. 29, 1999

**Related U.S. Application Data**

(63) Continuation of application No. 08/649,271, filed on May  
17, 1996, now Pat. No. 5,933,841.

(51) Int. Cl.<sup>7</sup> ..... G06F 17/21

(52) U.S. Cl. .... 707/501.1; 707/515; 345/854

(58) Field of Search ..... 707/501.1, 513,  
707/514, 515; 345/760, 854

(56) **References Cited**

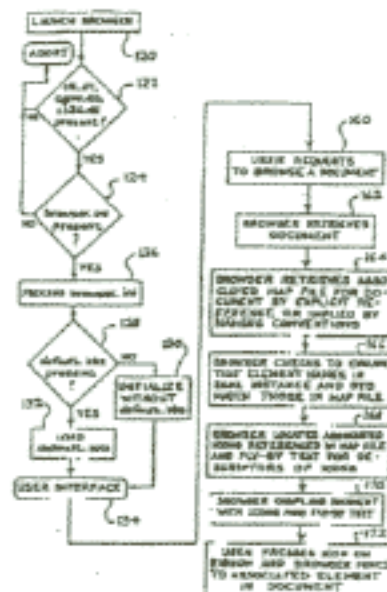
**U.S. PATENT DOCUMENTS**

4,616,336 A	10/1986	Robertson et al.
4,752,908 A	6/1988	Bouillot
4,829,453 A	5/1989	Katsuta et al.
5,021,989 A	6/1991	Fujisawa et al.
5,129,082 A	7/1992	Tirring et al.
5,140,521 A	8/1992	Kozol et al.
5,142,678 A	8/1992	Macphail
5,146,552 A	9/1992	Cassoria et al.
5,204,947 A	4/1993	Bernstein et al.
5,278,980 A	1/1994	Pedersen et al.
5,297,249 A	3/1994	Bernstein et al.
5,329,111 A	7/1994	Sonoda et al.
5,331,547 A	7/1994	Laszlo

(57) **ABSTRACT**

A structured document browser includes a constant user  
interface for displaying and viewing sections of a document  
that is organized according to a pre-defined structure. The  
structured document browser displays documents that have  
been marked with embedded codes that specify the structure  
of the document. The tags are mapped to correspond to a set  
of icons. When the icon is selected while browsing a  
document, the browser will display the section of the struc-  
ture corresponding to the icon selected, while preserving the  
constant user interface.

20 Claims, 16 Drawing Sheets



## U.S. PATENT DOCUMENTS

5,392,387 A 2/1995 Fitzpatrick et al.  
5,404,506 A 4/1995 Fujisawa et al.  
5,428,776 A 6/1995 Rothfield  
5,432,903 A 7/1995 Frid-Nielsen  
5,530,852 A 6/1996 Meske, Jr. et al.  
5,557,722 A 9/1996 Derosé et al.  
5,572,643 A 11/1996 Judson

5,623,679 A 4/1997 Rivette et al.  
5,625,781 A 4/1997 Cline et al.  
5,634,064 A 5/1997 Warnock et al.  
5,708,826 A 1/1998 Ikeda et al.  
5,774,109 A 6/1998 Winsky et al.  
5,933,841 A \* 8/1999 Schumacher et al. .... 707/501.1

\*cited by examiner

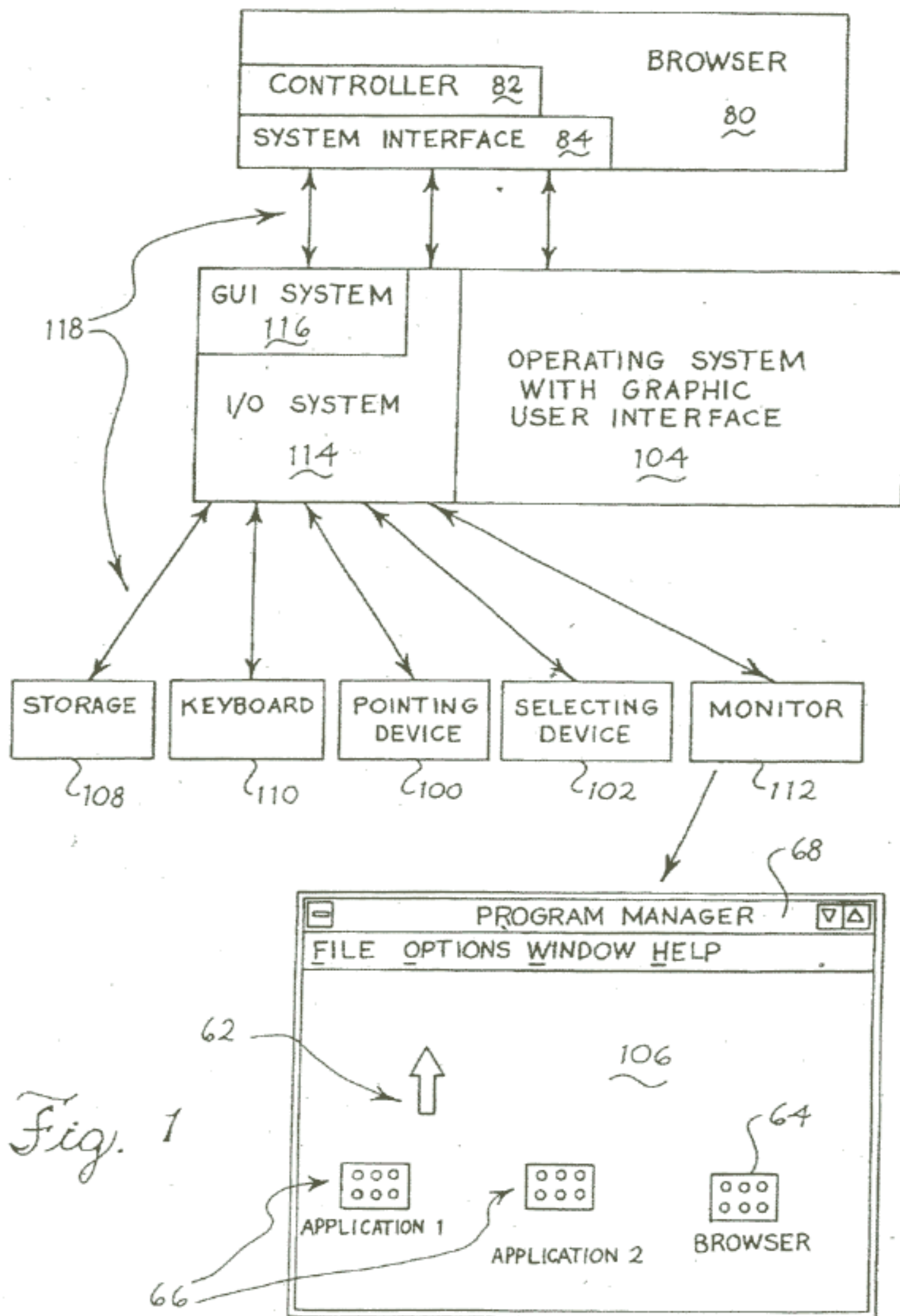
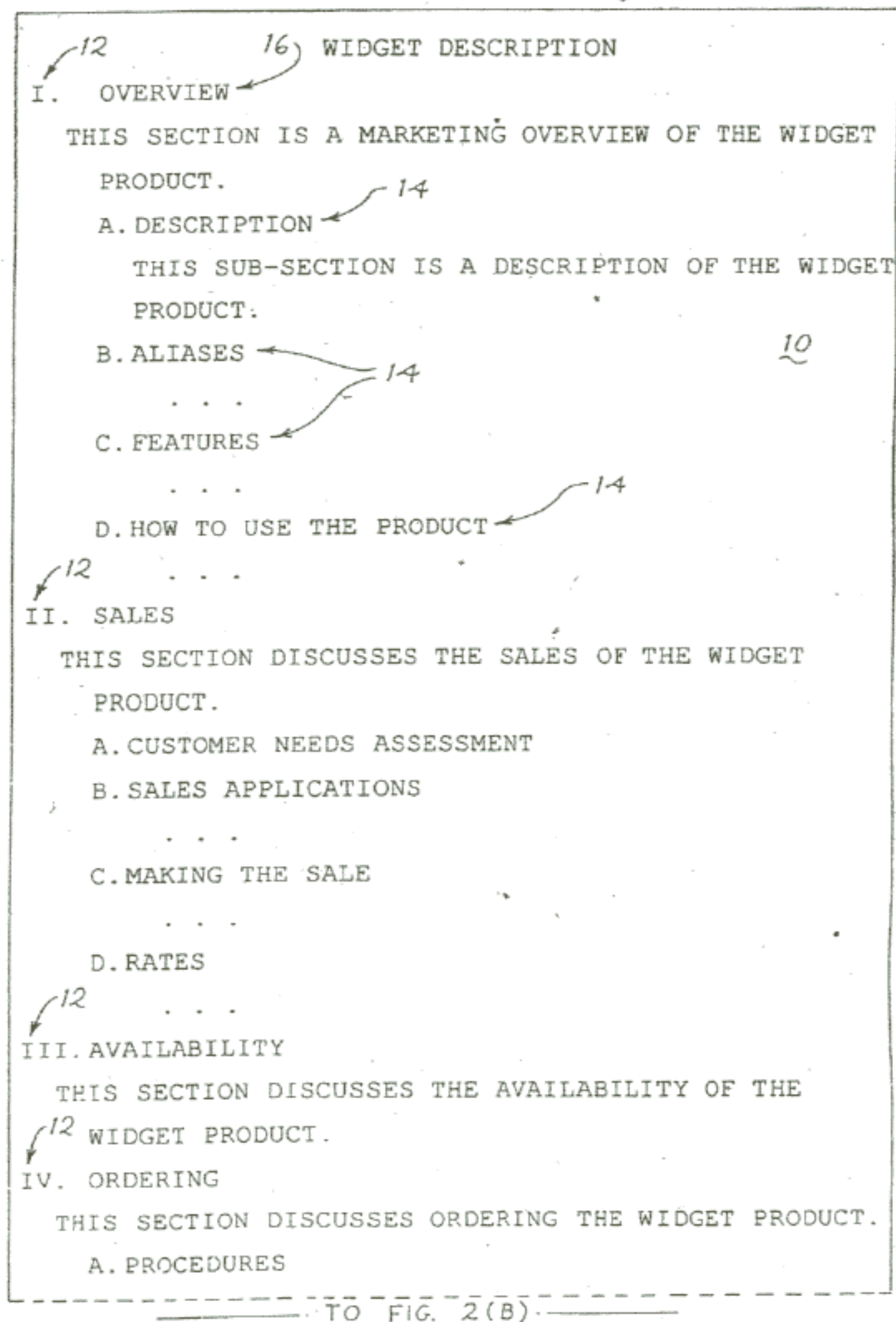




Fig. 2(A)



*Fig. 2(B)*

\_\_\_\_\_.TO FIG. 2(A).\_\_\_\_\_

. . .  
B. LISTINGS. . .  
C. DUE DATES12  
V. BILLINGTHIS SECTION DISCUSSES BILLING FOR THE WIDGET  
PRODUCT.

A. CHARGES

. . .  
B. BILLING REDUCTIONS. . .  
C. PAYMENT OPTIONS/ CONTRACTS12  
VI. TROUBLESHOOTING18  
THIS SECTION DISCUSSES TROUBLESHOOTING WIDGET  
PRODUCT PROBLEMS.

A. KNOWN PROBLEMS

. . .  
B. PROBLEM ASSESSMENT. . .  
C. DIAGNOSING CUSTOMER TROUBLE12  
VII. SUPPORT

THIS SECTION DISCUSSES WIDGET PRODUCT SUPPORT.

A. PRODUCT CONTACTS &amp; REFERRALS

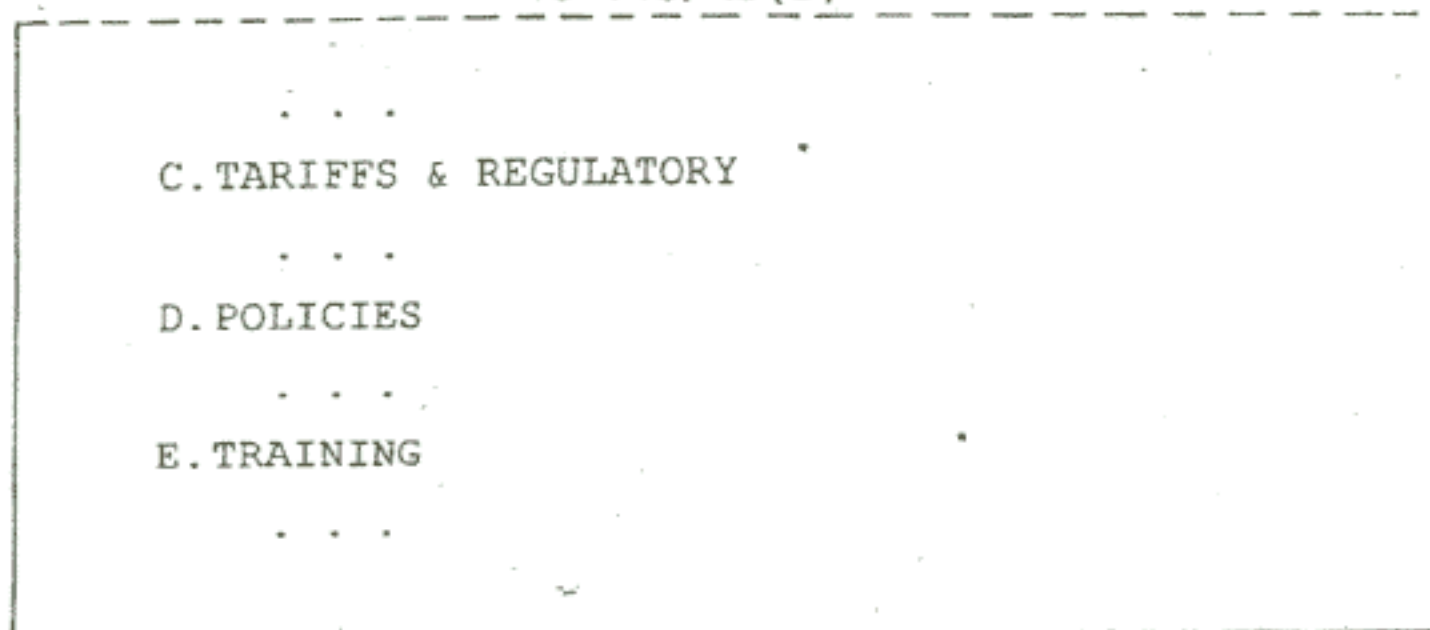
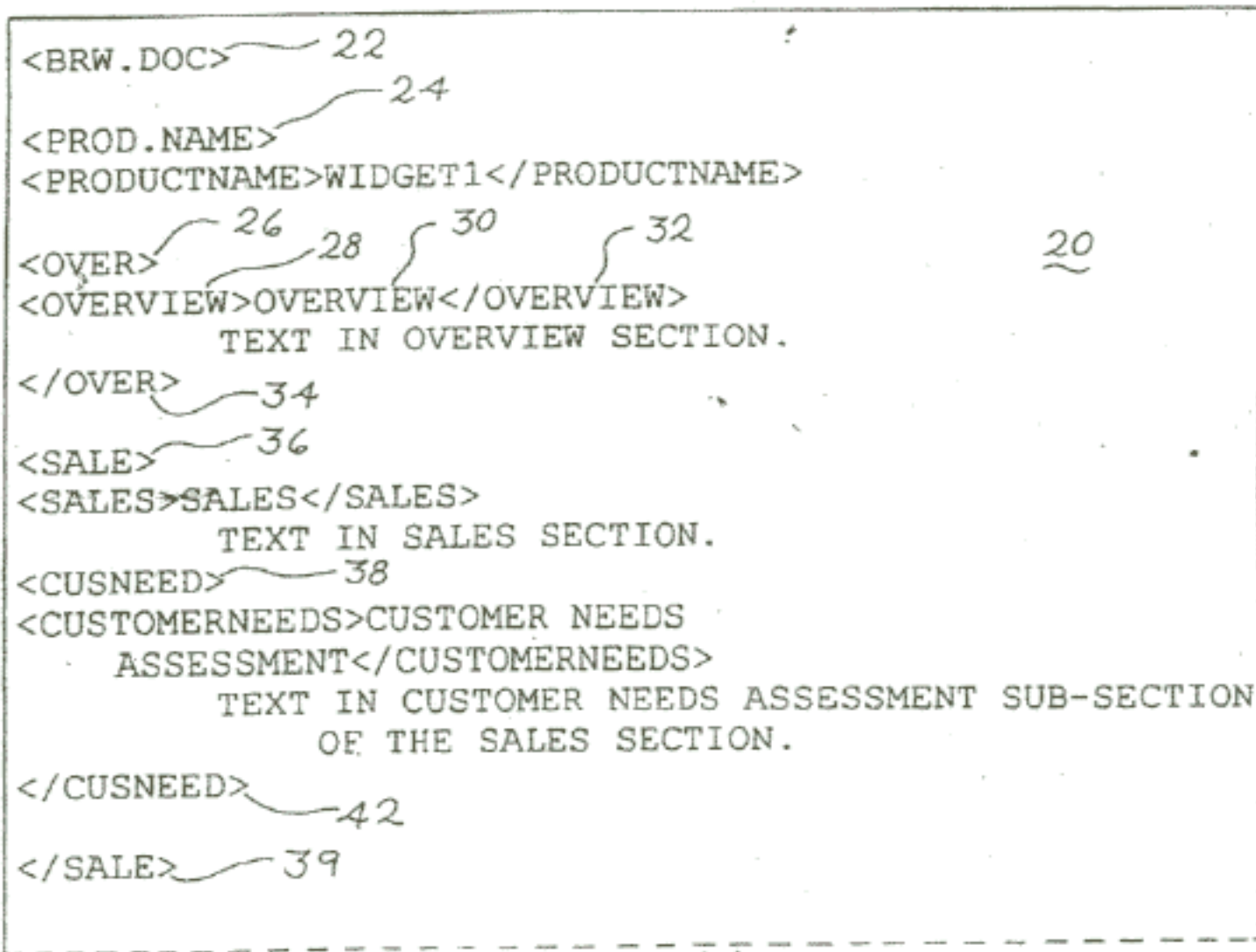
. . .  
B. RESPONSIBILITIES

\_\_\_\_\_.TO FIG. 2(C).\_\_\_\_\_



*Fig. 2(c)*

TO FIG. 2(B)

*Fig. 3(A)*

TO FIG. 3(B)

*Fig. 3(B)*

TO FIG. 3(A)

25  
<TROUBL>  
27 <TROUBLESHOOTING>TROUBLESHOOTING</TROUBLESHOOTING>  
TEXT IN TROUBLESHOOTING SECTION.  
<TRBL.KNOWPROB>  
<TRBLKNOWPROB>KNOWN PROBLEMS>/TRBLKNOWPROB>  
TEXT IN THE KNOWN PROBLEMS SUB-SECTION OF THE  
TROUBLESHOOTING SECTION.  
</TRBL.KNOWPROB>  
  
<TRBL.PROBASSESS>  
<TRBLPROBASSESS>PROBLEM ASSESSMENT</TRBLPROBASSESS>  
TEXT IN THE PROBLEM ASSESSMENT SUB-SECTION OF  
THE TROUBLESHOOTING SECTION.  
</TRBL.PROBASSESS>  
  
<TRBL.DIAG>  
<TRBLDIAG>DIAGNOSING CUSTOMER TROUBLE</TRBLDIAG>  
TEXT IN THE DIAGNOSING CUSTOMER TROUBLE SUB-  
SECTION OF THE TROUBLESHOOTING SECTION.  
</TRBL.DIAG>  
  
</PROD.NAME>  
  
</BRW.DOC>



Fig. 4(A)

```
<!ELEMENT PROD.NAME - - ( PRODUCTNAME, (%SUB-SECT; |  
    TROUBL |  
    ORDER |  
    AVAIL |  
    SALE |  
    BILL |  
    SUPP |  
    OVER)*) >  
<!ATTLIST PROD.NAME  
    ID CDATA #IMPLIED >  
<!ELEMENT PRODUCTNAME - - ( (#PCDATA | %PARA-  
    CONTENT;)* ) >  
<!ELEMENT TROUBL - - ( TROUBLESHOOTING, (%SUB-SECT; |  
    TRBL.KNOWPROB |  
    TRBL.PROBASSESS |  
    TRBL.DIAG |  
    TRBL.QUES |  
    HEAD3 |  
    HEAD2 |  
    HEAD5 |  
    HEAD4 |  
    HEAD7 |  
    HEAD6)*) >  
<!ATTLIST TROUBL  
    ID CDATA #IMPLIED  
    ID CDATA #IMPLIED >  
<!ELEMENT TROUBLESHOOTING - - ( (#PCDATA | %PARA-  
    CONTENT;)* ) >
```

Handwritten annotations in Fig. 4(A):

- 42: Arrow pointing to the vertical bar after AVAIL.
- 46: Arrow pointing to the vertical bar after SALE.
- 47: Arrow pointing to the vertical bar after TROUBL.
- 49: Arrow pointing to the vertical bar after (%SUB-SECT; |.
- 44: Arrow pointing to the vertical bar after (%PARA-.
- 45: Arrow pointing to the vertical bar after CONTENT;)\*.
- 40: Underlined, with an arrow pointing to the vertical bar after TRBL.QUES.
- 48: Arrow pointing to the vertical bar after HEAD3.
- 51: Arrow pointing to the vertical bar after ID CDATA #IMPLIED.

TO FIG. 4(B)

*Fig. 4(B)*

- TO FIG. 4(A) -

```
<!ELEMENT TRBL.KNOWPROB - - ( TRBLKNOWPROB, (%SUB-  
  SECT; |  
    HEAD3 |  
43 HEAD5 |  
    HEAD4 |  
    HEAD7 |  
    HEAD6) *) >  
<!ATTLIST TRBL.KNOWPROB  
  ID CDATA #IMPLIED  
  TYPE CDATA #IMPLIED >  
<!ELEMENT TRBLKNOWPROB - - ( (#PCDATA | %PARA-  
  CONTENT;)* ) >  
<!ELEMENT TRBL.PROBASSESS - - ( TRBLPROBASSESS,  
  (%SUB-SECT; |  
43 HEAD3 |  
    HEAD5 |  
    HEAD4 |  
    HEAD7 |  
    HEAD6) *) >  
<!ATTLIST TRBL.PROBASSESS  
  ID CDATA #IMPLIED  
  TYPE CDATA #IMPLIED >  
<!ELEMENT TRBLPROBASSESS - - ( (#PCDATA | %PARA-  
  CONTENT;)* ) >
```



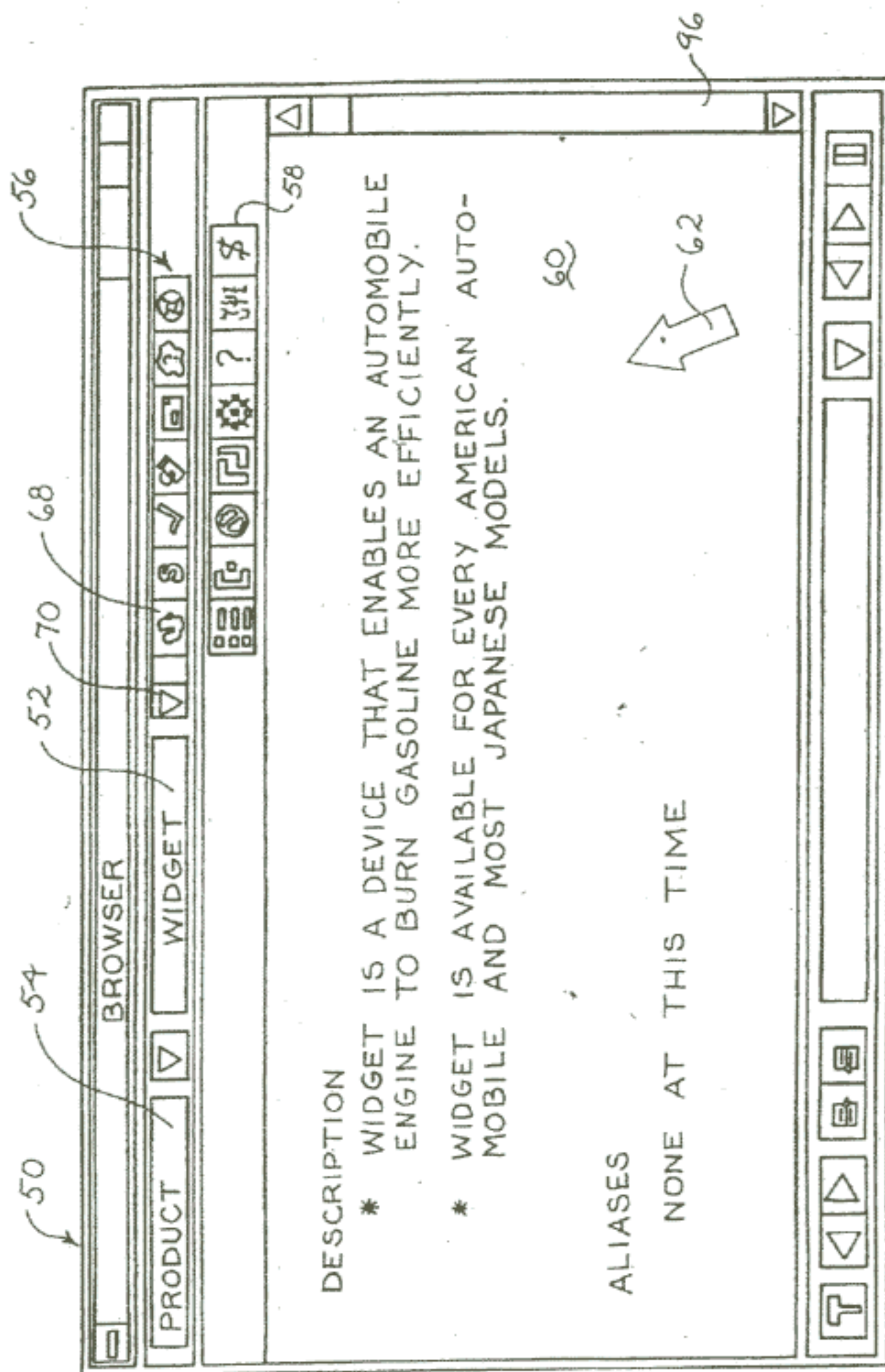
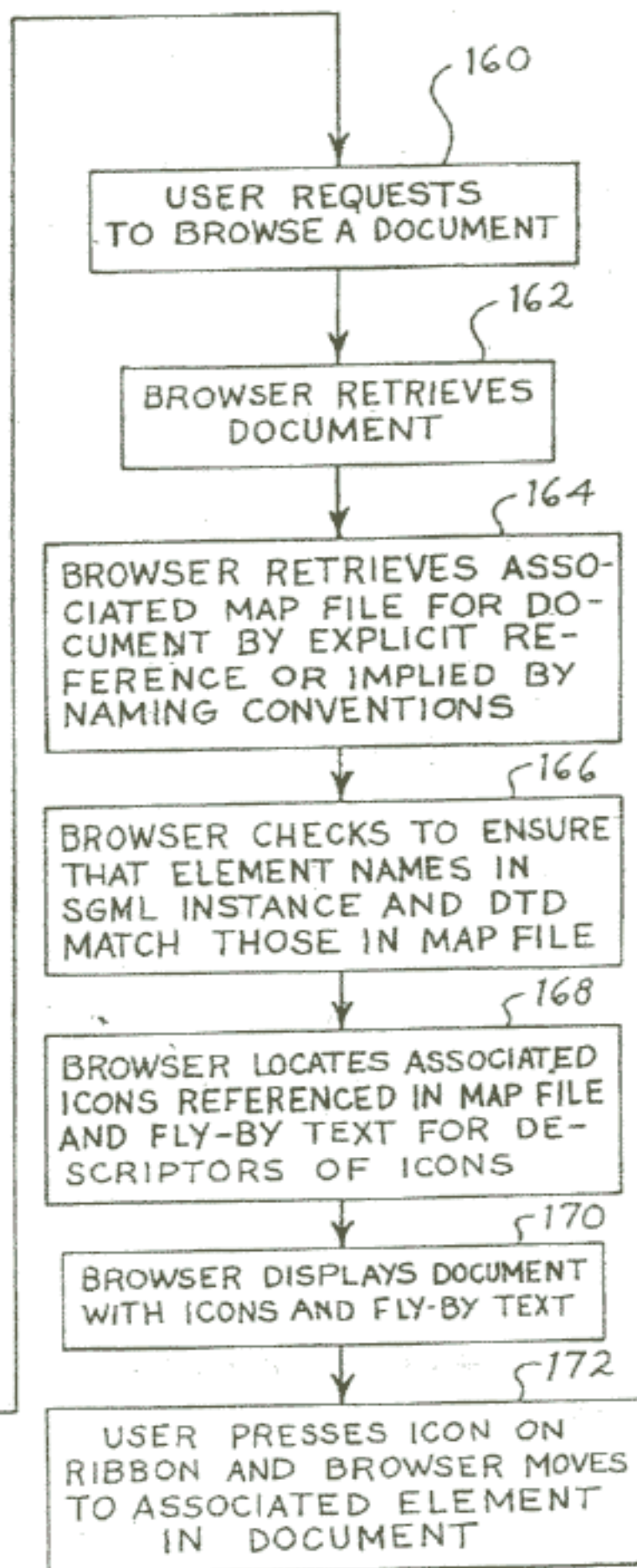
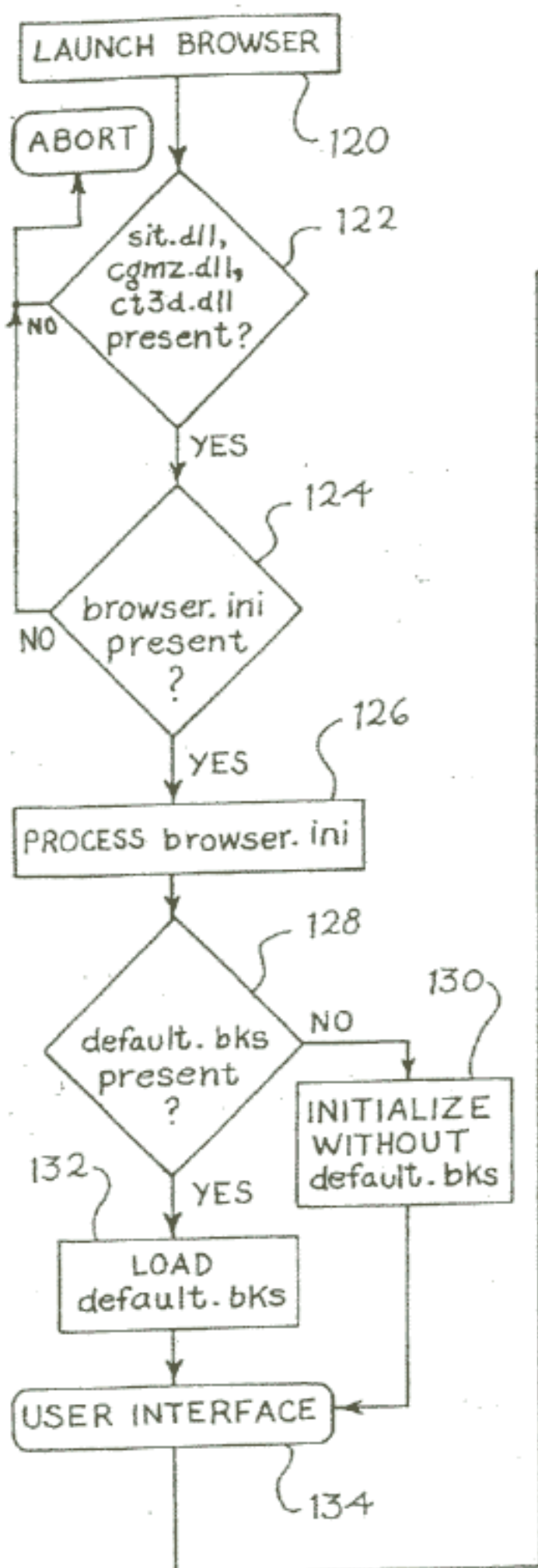
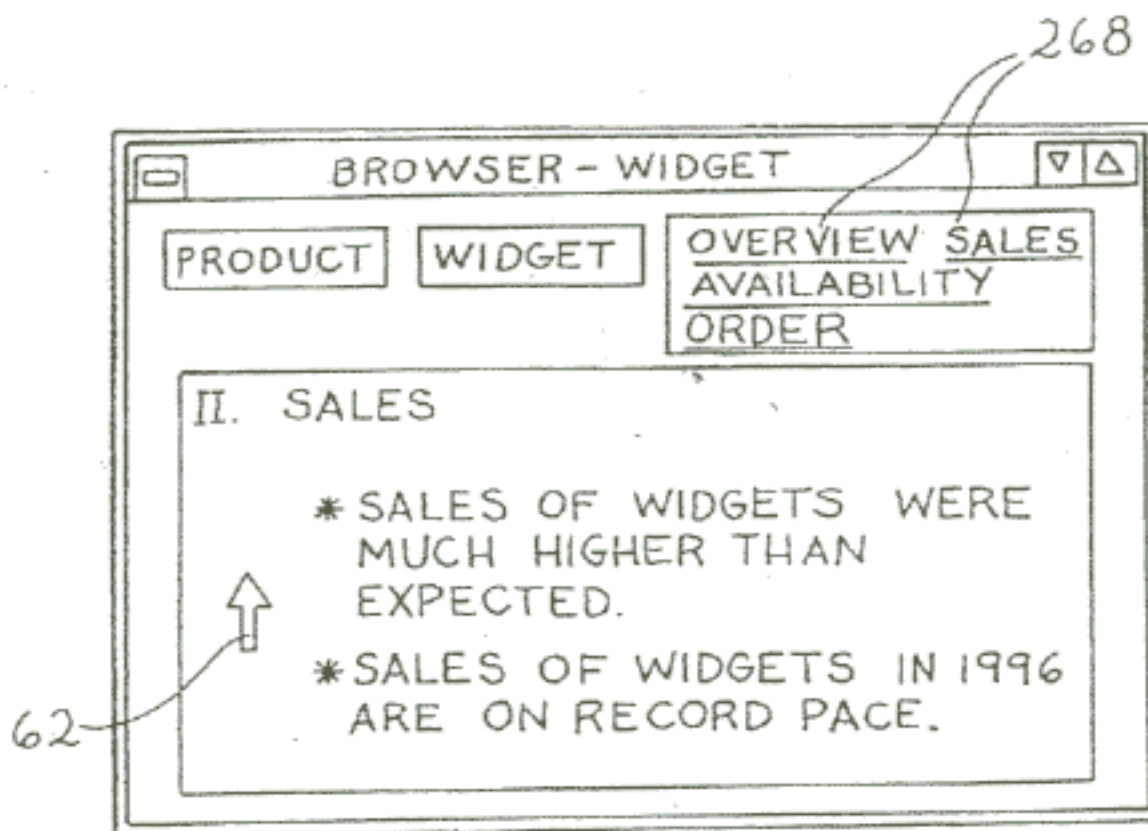


Fig. 5





[BOOK] 140  
142  
144 COLLECTION=\XYZ  
146 COLLECTIONTITLE+"PRODUCTS"  
148 BOOKNAME=WIDGET  
150 BOOKTITLE="WIDGET"  
152 MAP=PRD.MAP

*Fig. 7**Fig. 10*

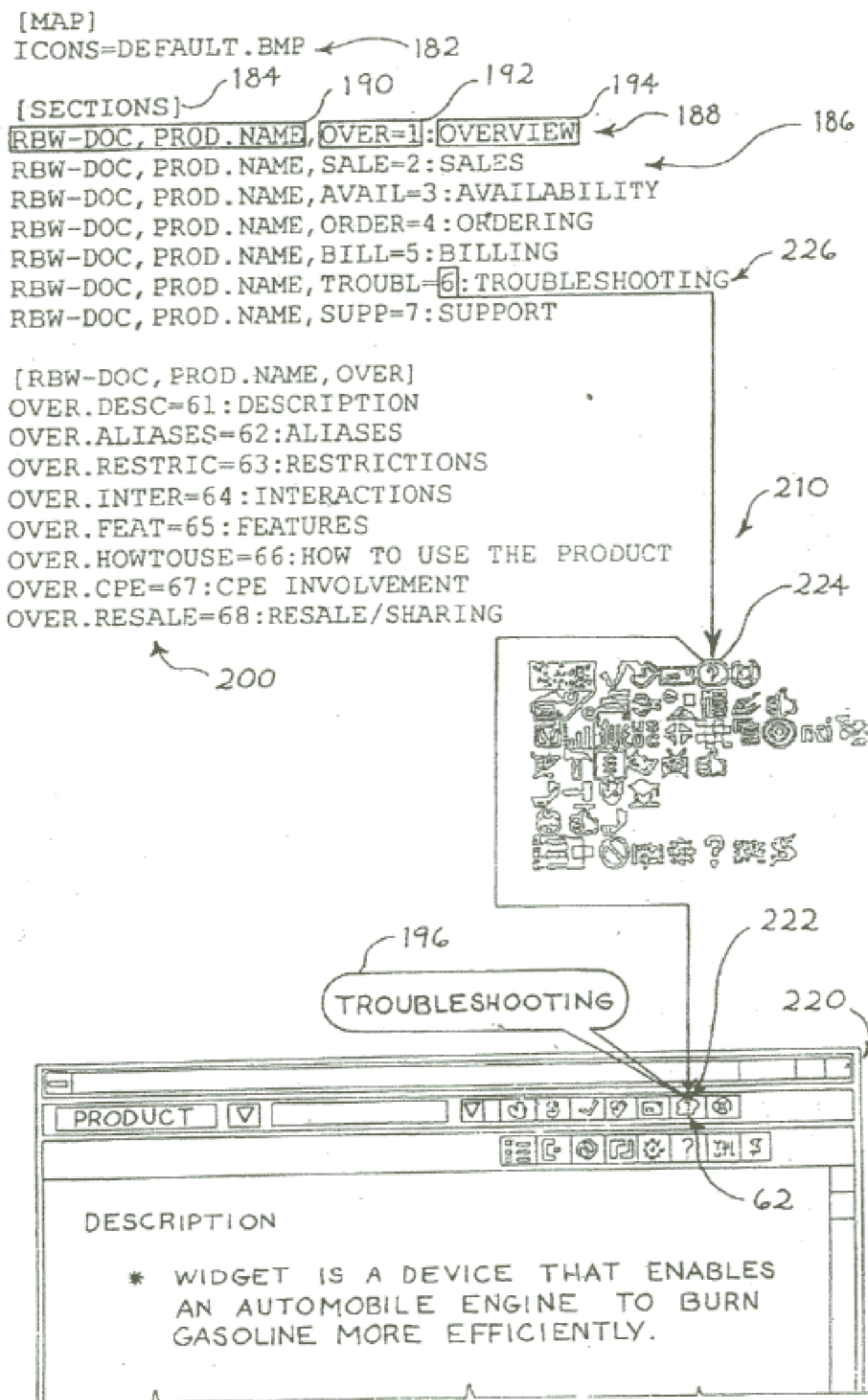


FIG. 8



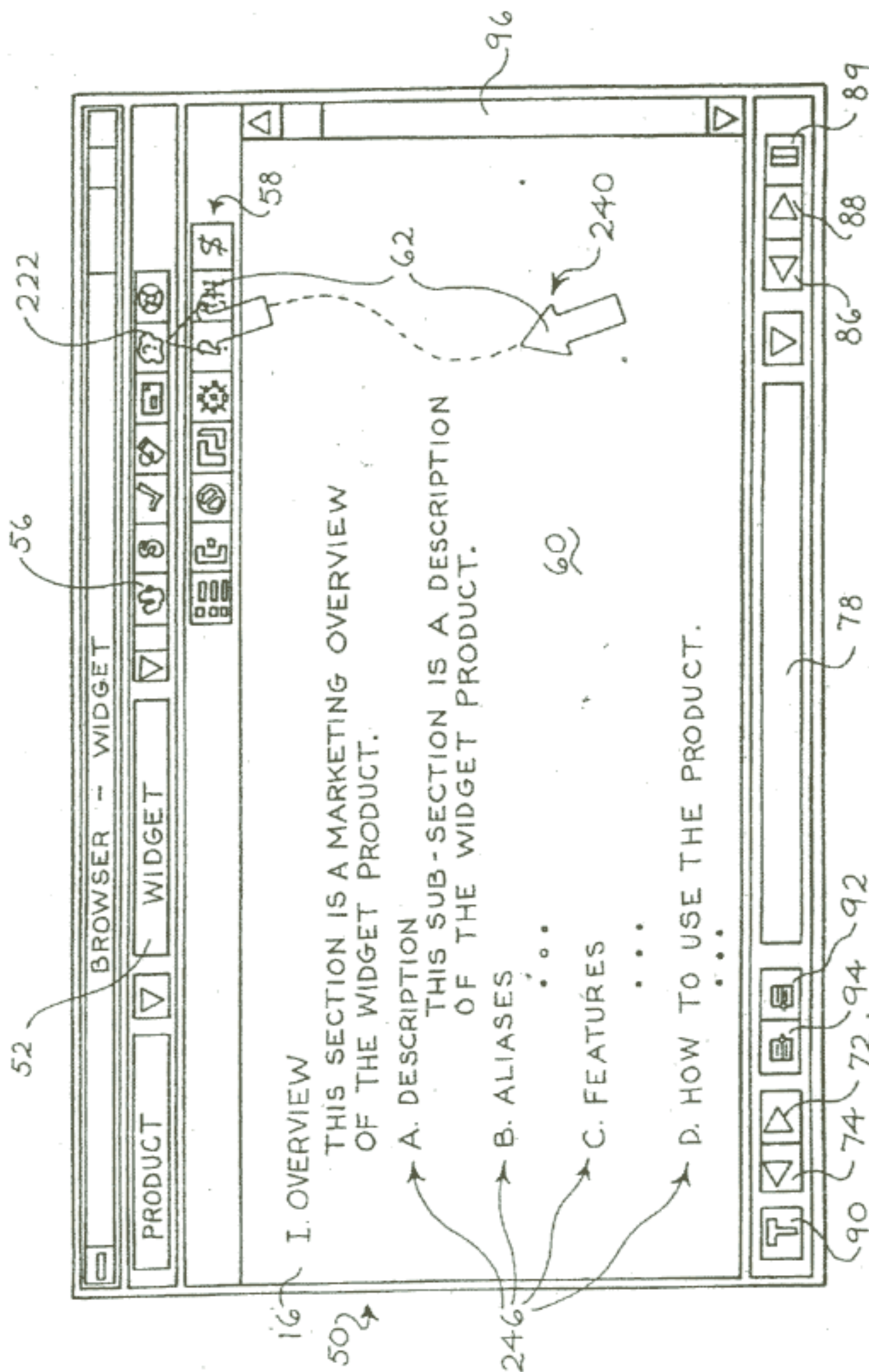


Fig. 9A