

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
31 May 2001 (31.05.2001)

PCT

(10) International Publication Number
WO 01/39020 A2

- (51) International Patent Classification⁷: **G06F 17/00**
- (21) International Application Number: PCT/US00/32215
- (22) International Filing Date:
22 November 2000 (22.11.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/448,674 24 November 1999 (24.11.1999) US
- (71) Applicant: **SUN MICROSYSTEMS, INC.** [US/US];
Patent Operations Manager, M/S: UPAL01-521, 901 San
Antonio Road, Palo Alto, CA 94303 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ,
DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,
NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM,
TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF,
CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**
— *Without international search report and to be republished
upon receipt of that report.*
- (72) Inventors: **FUNK, Matthew, S.**; 3135 E. Yarrow Circle,
Superior, CO 80027 (US). **GREGORY, Marla, D.**; 6027
Cole Court, Arvada, CO 80004 (US). **PRICE, Judith, A.**;
704 Club Circle, Louisville, CO 80027 (US).
- For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.*
- (74) Agents: **BURTON, Carol, W.** et al.; Hogan & Hartson
LLP, Suite 1500, 1200 17th Street, Denver, CO 80202 (US).



WO 01/39020 A2

(54) Title: SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PUBLISHING WEB PAGE CONTENT HAVING A UNIFORM PREDETERMINED FORMAT AND FEATURES

(57) Abstract: A system, method and computer program product for publishing web page content having a uniform predetermined format and features. The web page content publisher of the present invention is an extensible, modular and flexible computer program implemented solution to combining complex standard web navigation elements with HTML, SGML, XML and/or web-based application content for the generation of web-ready pages for a variety of web developers and web sites, without regard to data structure.

SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR
PUBLISHING WEB PAGE CONTENT HAVING A UNIFORM
PREDETERMINED FORMAT AND FEATURES

BACKGROUND OF THE INVENTION

5 The present invention relates, in general, to the field of systems, methods and computer program products for publishing web page content having a predetermined format and features ("or look and feel") in a distributed computing environment such as that of a local area network ("LAN") wide area network ("WAN"), intranet or Internet. More
10 particularly, the present invention relates to an automated template publishing system which greatly facilitates the combination of individual user web content within the context of an existing template of particular utility in presenting content from multiple users with a uniform "look and feel".

15 Historically, many internal corporate webmasters and content providers maintained web pages on a page-by-page basis, often including header and side bar (collectively, the "hockey stick") navigation in the page Hypertext Markup Language ("HTML") code itself. (HTML is the set of "markup" symbols or codes inserted in a file
20 intended for display on a World Wide Web browser and informs the web browser how to display a web page's words and images for the user.) Therefore, if a link in the hockey stick changed, every page on the site then required amendment, one page at a time with predictable results for the interjection of possible errors in the HTML code and a
25 concomitant consumption of valuable programming time and effort.

 Furthermore, graphic artists were often employed at great expense to provide improved "look and feel" to the web pages in view of the fact that most publishers of web page content may not be particularly adept at providing a polished website look. However, the
30 resulting graphics could require excessively large files that that were

slow to load and perhaps not compliant with internal standards, some of which require that the HTML pages be no larger than about 50KB in size.

Still further, in a corporate setting or other situation wherein a large number of people can potentially produce web page content, a lack of uniform appearance and functionality is the likely result and even if common functionality were somehow maintained, it could be placed at any number of different locations within the HTML page or implemented in a number of different ways. As such, users were presented with a potential new and different interface for each web-based application depending on the web page content provider.

Heretofore, one known approach has included the use of a Perl language script through which the HTML content was copied verbatim into an HTML file containing web site navigational content. Among the known problems with this approach is that the script file is not extensible due to the fact that it was written as one large script file and changes cannot be made without the risk of unintended changes to the original intent of the script. Further, this approach does not provide the requisite flexibility needed as changes to the script would be difficult to make due to the implied variables inherent in the Perl language. Moreover, the script is not modular and changes to one portion of the script could lead to unexpected changes in other parts of the script. Other known problems include an acknowledged non-compliance with Year 2000 ("Y2K") readiness, its hard-coded specific intended use, lack of back up for previous live files and presupposition of a particular manner of maintaining a given web-site.

SUMMARY OF THE INVENTION

In marked contrast, the system, method and computer program product for publishing web page content of the present invention overcomes the problems in previous approaches and provides a
5 standard "look and feel" along with a standard implementation of standard functionality in the context of an automated publishing tool. This publishing tool enables users to easily incorporate standard navigational content (i.e., the "hockey stick") into their HTML pages. Particularly with respect to major web sites, which may contain tens of
10 thousands of web pages, the automated tool disclosed herein is a decided advancement over manual modification and other known existing approaches.

The present invention provides a computer program based solution for publishing web page content that combines standard web
15 navigation elements with HTML and/or web-based application content for the generation of web-ready HTML pages of particular utility for less sophisticated users, thereby enabling many more people to be able to publish accurate and consistent web page content. It advantageously provides HTML-only (for web pages) and 99% graphic-free (for web-
20 based applications) final web pages ensuring a consistent "look and feel" throughout HTML web sites and provides a standardized implementation of common functionality as well as cost savings by obviating the need to hire web page or graphics designers.

The web page content publisher disclosed herein may comprise
25 a suite of classes which takes existing HTML content and inserts configurable forms of navigation into the file. This allows users to select different configurable forms of navigation, including the option to select no navigation. This has broad, positive implications for greater adaptability and increased flexibility in terms of what the user can

provide. In a particular embodiment of the invention disclosed in detail herein in the form of a specific web page content publisher the web page content publisher may be written using distinct Java® (trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries), classes which by its very nature assures extensibility due to the fact that new classes can be defined based upon established classes and interfaces. Moreover, these classes are flexible because the implementations of operations, inputs, and outputs can be changed easily, without changing the intrinsic meaning or purpose of the original program. Further, because the classes are modular (meaning the implementations of operations are hidden), it is much less likely that changing the implementation in one class could unintentionally create errors in another class.

The system, method and computer program product of the particular embodiment of the present invention disclosed herein is also inherently Y2K compliant, having been developed using the Java Development Kit (JDK™ 1.2, a trademark or registered trademark of Sun Microsystems, Inc. in the U.S. or other countries). Furthermore, the representative web page content publisher disclosed herein does not manipulate dates, but merely passes through dates provided by the Java Development Kit, without change. For time dependent functions, a method is utilized in the java.io.File class lastModified() to return an integer which is then used in the java.util.Date(long) constructor. This allocates a java.util.Date object and initiates it to represent the specified number of milliseconds since the standard base time (known as "the epoch") date of January 1, 1970 00:00:00 GMT.

Still further, the web page content publisher of the present invention is scalable for use within specific company intranet and internet sites and may be used by a wide variety of companies, users,

industries and templates. It can be configured to back up existing files and publish new files in an optional "staging directory" and then a "go live" function can automatically move these files to the live HTML directory. The present invention also utilizes a flexible directory
5 structure which enables it to function as intended regardless of the particular data sequence and groupings imposed by a particular webmaster and may be readily configured by even intermediate users.

Particularly disclosed herein is an extensible, modular and flexible computer program-based solution which combines complex
10 standard web navigation elements with HTML and/or web-based application content for the generation of web-ready HTML pages for a variety of web developers and web sites without regard to data structure. The publisher disclosed herein may also be readily adapted to servlets and extensible markup language ("XML"), standard
15 generalized markup language ("SGML") and similar languages for making clear an interpretation of text insofar as its formatting, printing or other processing is encoded as well. (XML is a World Wide Web Consortium ("W3C") proposed recommendation for a file format to easily and cheaply distribute electronics documents on the World Wide
20 Web; SGML is an international standard for the definition of device-independent, system-independent methods of representing text in electronic form.)

Particularly disclosed herein is a method, and computer program product implementing the method, for publishing web page content in
25 accordance with a predetermined template. The method comprises the steps of creating a content file, initializing a web page content publisher for inserting the predetermined template information into the content file, requesting publication of the content file by the web page

content publisher and publishing the content file in a web viewable format in accordance with the predetermined template.

Further disclosed herein is a computer system including a plurality of central processing units interconnected by a data network with at least two of the central processing units including a data entry device and an associated visual display for displaying selected data available on the data network. The computer system comprises a predetermined data network template accessible at the first one of the central processing units for defining at least a visual appearance of the selected data available on the data network, and a network content publisher also accessible at the first one of the central processing units for receiving a content file. The network content publisher is operable for integrating the content file with the predetermined data network template to produce published data network content viewable on at least the associated visual display of a second one of the central processing units coupled to the data network in a format conforming to the predetermined data network template.

BRIEF DESCRIPTION OF THE DRAWINGS

The aforementioned and other features and objects of the present invention and the manner of attaining them will become more apparent and the invention itself will be best understood by reference to the following description of a preferred embodiment taken in conjunction with the accompanying drawings, wherein:

Fig. 1 illustrates a representational network computing system and operating environment for implementing a system, method and performing the computer implemented steps of a method in accordance with the present invention;

Fig. 2 is an exemplary conceptual scenario for a particular embodiment of a web page content publisher in accordance with the

principles of the present invention illustrative of the publishing of a content page;

Fig. 3 is a follow-on exemplary conceptual scenario for the particular embodiment of the web page content publisher of the preceding figure illustrative of the generation of a content page;

Fig. 4 is a representative object model for a particular embodiment of the web page content publisher of Figs. 2 and 3;

Fig. 5 is a more detailed object model of the page content retriever of the particular embodiment of the web content publisher of Fig. 4;

Figs. 6A and 6B are more detailed object models of the page content and published page portions respectively of the particular embodiment of the web page content publisher of Fig. 4;

Fig. 7A is an overall template object model for the particular embodiment of the web page content publisher of Fig. 4;

Figs. 7B, 7C and 7D are more detailed illustrations of the basic template, applications and web object models shown in Fig. 7A;

Fig. 8 is a representative user screen shot of the particular implementation of the web page content publisher of the preceding figures illustrating the main console thereof and showing that the content file has been created but not published and not yet been made "live";

Fig. 9 is a follow-on representative user screen shot of the main console of Fig. 8 now in a condition of the content file having been published but still needing to go "live";

Fig. 10 is a further follow-on representative user screen shot of the main console of Fig. 9 illustrating that the content file has been previously published and now been made "live";

Fig. 11 is an additional representative user screen shot of the
5 notify console of the web page content publisher of the preceding figures; and

Fig. 12 is a view of the user content as viewed from a web browser in the form of an index HTML page following being made "live".

DESCRIPTION OF A PARTICULAR EMBODIMENT

10 With reference now to Fig. 1, the present invention is illustrated and described in terms of a distributed computing environment such as an enterprise computing system using public communication channels such as the Internet. However, an important feature of the present invention is that it is readily scaled upwardly and downwardly to meet
15 the needs of a particular application. Accordingly, unless specified to the contrary, the present invention is applicable to significantly larger, more complex network environments as well as small network environments such as conventional local area network ("LAN") systems.

20 It is contemplated that the present invention will be particularly useful in environments that require a data structure that is quickly searched and where the data is suited to a hierarchical representation. Also, the system of the preferred implementation is designed to store and make available relatively compact units of data that serve to
25 configure (i.e., startup, return decisions, shutdown) devices and computer environments rather than operational or analytical data upon which the computer environment may operate at runtime. Hence, the present invention is particularly advantageously used when it stores and retrieves data that is frequently searched and retrieved, but

infrequently changed although it may also be used in conjunction with data that is frequently changed as well.

Fig. 1 shows an exemplary computing environment 100 in which the present invention may be implemented. Essentially, a number of
5 computing devices and groups of devices are interconnected through a network 101. For example, a LAN 102 and a LAN 103 are each coupled to network 101 through gateway machines 104 and 105 respectively. LANs 102 and 103 may be implemented using any available topology such as a hub and spoke topology of LAN 102 and a
10 loop topology of LAN 103. LANs 102 and 103 may implement one or more server technologies including, for example those utilizing a UNIX, Novell, Windows NT, SolarisTM operating environment (a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries) or peer-to-peer type network. Each network will
15 include distributed storage implemented in each device and typically includes some mass storage device coupled to or managed by a server computer. Network 101 comprises, for example, a public network such as the internet or another network mechanism such as a fibre channel fabric or conventional wide area network ("WAN") technologies.

20 LAN 102 includes one or more workstations such as personal computer ("PC") 106. LAN 102 also includes a server machine 107 and one or more shared devices such as printer 108. A hub or router 109 provides a physical connection between the various devices in LAN 102. Router 104 is coupled through gateway 109 to provide shared
25 access to network 101. Gateway 109 may implement any desired access and security protocols to manage access between network 101 and devices coupled to network 102. Similarly, network 103 comprises a collection of workstations 111, 112 and 113 that share a common connection to network 101 through gateway 105.

Distributed computing environment 100 further includes a wide variety of devices that have a logical connection to the network supported by a physical connection to network 101. For example, a stand alone workstation 114 may couple to network 101 through a
5 modem or other suitable physical connection. Likewise, notebook computer 115 and palmtop computer 116 may connect to network 101 using known connection technologies. It is contemplated that a wide variety of devices may join the distributed network 100 including mobile phones, remote telemetry devices, information appliances, and the like.
10 An important feature of the present invention is that it tolerates and adapts to an environment filled with heterogeneous hardware devices coupled to the network 101 from a variety of physical locations.

Each of the devices shown in Fig. 1 may include memory, mass storage, and a degree of data processing capability sufficient to
15 manage their connection to network 101. The computer program devices in accordance with the present invention are implemented in the memory of the various devices shown in Fig. 1 and enabled by the data processing capability of the devices shown in Fig. 1. In addition to local memory and storage associated with each device, it is often
20 desirable to provide one or more locations of shared storage such as disk farm 117 that provides mass storage capacity beyond what an individual device can efficiently use and manage. Selected components of the present invention may be stored in or implemented in shared mass storage such as disk farm 117.

25 The computer program product devices in accordance with the present invention include elements that operate in a server, a client or both. It is contemplated that elements may be stored remotely, delivered to a client system on demand by a server computer and executed partially and completely by the server and client. Accordingly,

the present invention is not limited by the method of distribution or packaging that a particular application involves. In other words, the present invention may be distributed as client-only software devices, server-only software devices or as system software that is distributed to
5 both client and server devices.

With reference additionally now to Fig. 2, an exemplary conceptual scenario 200 for a particular embodiment of a web page content publisher in accordance with the principles of the present invention is shown illustrative of the publishing of a content page. The
10 scenario 200 begins when a Client 220 initializes a Publisher 222 with the configuration of the site and page/template as shown in initialize(configuration) step 202. The Publisher 222 transfers the site configuration to the SiteManager 224 as in initialize(site configuration) step 204 and the SiteManager 224 maps the page content files to the
15 published page files.

At publish(content file name) step 206, the Client 220 requests of the Publisher 222 that a particular content file with a given name be published. Thereafter, at steps 208 [getPageContent(content file name)] and 210 [getPublishedFile(content file name)] the Publisher 222
20 requests the PageContent and published page file from the SiteManager 224. The Publisher 222 then queries the PageContent 226 which template should be used to publish it at getTemplate() step 212.

Following initialization of the template received from
25 PageContent 226, the Publisher 222 requests the Template Interface 228 to write the published page using the PageContent at steps 214 [initialize(page configuration)] and 216 [writePublishedFile(page content, published file)]. It should be noted that the Publisher 222 only interacts with the template through the Template Interface 228 and,

preferably, that only the PageContent 226 has knowledge of exactly which template is being used.

With reference additionally now to Fig. 3, a follow-on exemplary conceptual scenario 300 for the particular embodiment of the web page content publisher of Fig. 2 is shown illustrative of the generation of a content page. In this scenario 300, a Client 330 is the Publisher and it initializes a Template 332 with page configuration information which contains the configuration of the Template 332 at initialize(page configuration) step 302. At write PublishedFile(page content, published file) step 304, the Client 330 requests the Template 332 to be inserted into the PageContent 340 and the result to be written to the published file. Thereafter at steps 306 and 308 respectively, the Template 332 initializes the page configuration from the TemplateHead 334 and the TemplateBody 336.

At analyze(page content) step 310, the PageContent 340 is analyzed by means of the PageContentAnalyzer 338 to determine where the template elements should be inserted into the PageContent 340. At steps 312 and 314 respectively, the TemplateHead 334 and TemplateBody 336 are generated based on the page configuration.

Step 316 [getHEADStart()] begins that portion of the scenario 300 in which the insertion points in PageContent 340 are located and the template parts are inserted into the PageContent 340. It is followed by insertAfter(template HEAD, page content HEAD start) step 318. In like manner, at getBODYStart() step 320, the Template 332 interacts with the PageContentAnalyzer 338 and then with the PageContent 340 at insertAfter(template BODY top, page content BODY start) at step 322. Similarly, at getBODYEnd() step 324, the Template 332 interacts with the PageContentAnalyzer 338 and then with the PageContent 340 at insertAfter(template BODY bottom, page content BODY end) at step

326. Finally, at step 328 [writeTo(published page)], the PageContent 340 with the Template 332 elements is written to the published page.

With reference additionally now to Fig. 4, a representative object model 400 for a particular embodiment of the web page content publisher of Figs. 2 and 3 is shown. As illustrated, a command line interface is the first delivered interface to the web page content publisher class 402 of the present invention which, in the particular embodiment illustrated and described, is a representative publishing program. The class Publish is the command line implementation of the interface to the program Publish and this class can be changed to implement other interfaces such as a graphical user interface ("GUI"), servlets and the like. The web page content publisher class 402 uses both a site publisher class 404 for controlling the web page content publishing process and a utility class 406 which is used to manage the command line arguments. The utility class 406 extracts the page content and site configuration files from the command line.

An object of site publisher class 404 uses a page content retriever class 408 to create an object having the page content, which could be in a file. The page content retriever class 408, in turn, creates an object implementing the page content interface class 412. An object of site publisher class 404 also uses an object implementing the template interface class 410 to insert template elements into the page content, which, in turn uses an object implementing the page content interface class 412 created by the page content retriever class 408 as well as an object implementing the published page interface class 414, created by the site publisher class 404, which represents the resulting published page, which could be in a file.

With reference additionally now to Fig. 5, a more detailed object model 500 of the page content retriever class 408 of the particular

embodiment of the web page content publisher of Fig. 4 is shown. As depicted, an object of page content retriever implementation class 502 is utilized to retrieve standard Publisher (HTML) formatted page content files, and an object of any other generic content retriever implementation class 504 is utilized to retrieve content of some other format, providing implementation of a page content retriever interface class 506. The page content retriever interface class 506 provides a common interface for the various page content retriever implementation classes 502 and 504 for use by the page content retriever class 408 which functions to retrieve the page content files using the installed retriever.

With reference additionally now to Figs. 6A and 6B, a more detailed object model 600 of the page content and published page portions respectively of the particular embodiment of the web page content publisher of Figs. 4 and 5 are shown. An object of page content class 602, which contains the content of the page being published, implements the page content interface class 412 which is used by the page content retriever class 408 (Figs. 4 and 5). Further, an object of published page interface class 604, which directs the published page to a desired file, implements the published page interface class 414 used by the site publisher class 404 (Fig. 4).

With reference additionally now to Fig. 7A, an overall template object model 700 for the particular embodiment of the web page content publisher of Fig. 4 is shown. The template object model 700 illustrates that the template interface class 410 used by the site publisher class 404 (Fig. 4) provides an interface to multiple templates. In the particular embodiment illustrated, these templates may include a basic template class 702, an applications template class 704 and/or a web template class 706.

With reference additionally now to Figs. 7B, 7C and 7D more detailed view of the object model 700 of the basic, applications and web template classes 702, 704 and 706 respectively is shown.

Referring to Fig. 7B particularly, the basic template class 702 includes
5 a template style sheet class 708 and associated template head class 710. Similarly, a template content footer class 712 and template content header class 714 is provided.

As shown in Fig. 7C, the applications template class 704 also includes a template style sheet class 718 and associated template
10 head class 720. A template content header class 724 and template content footer class 722 are also included in a manner similar to that of the basic template class 702 (Fig. 7B). However, a template category links class 726, template image class 728 and template navigation links class 730 are also included which generate the category links and
15 navigation links for the template content header class 724 and template content footer class 722 as shown.

As shown in Fig. 7D, the web template class 706 includes a template style sheet class 738 and template head class 740. It further includes a template body class 742 in conjunction with an applications
20 template side bar navigation configuration class 744 as well as template page header 746, template header bar 748, template "greybar" navigation configuration 750, template search field 752, template web links 754 and template page footer 756 classes.

The following is an example ASCII (American Standard Code for
25 Information Interchange) content file denominated: "index.content" illustrative of the minimal user involvement required in publishing web page content utilizing the web page content publisher of the present invention. The user need only provide, if desired, a title which in this instance is: "This is my New Page" and the actual page content:


```

This is my content in fully-qualified HTML format.
<P>
This is more content.
<P>
5  This is even more content.
   as shown in the bold italicized portions following.
   <!-- NOTE: Change TITLE and META TAG values below to apply to
   your site. -->
10  <HTML>

   <HEAD>
   <TITLE>This is my New Page</TITLE>
       <META NAME="URL" CONTENT="http://my_site_URL">
15       <META NAME="DOC"
   CONTENT="hostname.domain:/full/directory/path/docname.html">
       <META NAME="LAST-MODIFIED" CONTENT="DD-MMM-YYYY">
       <META NAME="LAST-MODIFIED-TEXT" CONTENT="username@domain">
       <META NAME="OWNER" CONTENT="username@domain">
20       <META NAME="DESCRIPTION" CONTENT="My Site or Page Title">
       <META NAME="KEYWORDS" CONTENT="comma, separated, key,
   words, or phrases">
       <META NAME="SECURITY" CONTENT="Company
   Proprietary/Confidential: Internal Use Only">
25       <META NAME="EXPIRES" CONTENT="Mon, 31 Dec 2001 23:59:00
   GMT>
       <LINK REV="MADE" HREF="mailto:webmaster@mysite.domain">
   </HEAD>
30  <BODY BGCOLOR="#FFFFFF">

   <!-- ***** START CONTENT HERE ***** -->

   This is my content in fully-qualified HTML format.
35  <P>
   This is more content.
   <P>
   This is even more content.
40  <!-- ***** END CONTENT HERE ***** -->

   </BODY>
   </HTML>

```

45 With reference additionally now to Fig. 8, a representative user screen shot of the main console 800 of the particular implementation of the web page content publisher is shown in a condition of the content file "index.content" 802 having been created and now requiring publication. The designation: "NOT PUBLISHED" at reference number

804 appears instead of a date published and the designation: "NOT LIVE" as indicated by reference number 806 appears in conjunction with the heading "Date Made Live". The action indicated in conjunction with the next user action needed is "PUBLISH" as indicated by
5 reference number 808.

With reference additionally now to Fig. 9, a follow-on representative user screen shot of the main console 800 shown in Fig. 8 is illustrated now in a condition of the content file "index.content" indicated by reference number 802 being published as indicated by.
10 reference number 810 and now needing to "go live" following publication as indicated by reference number 812. In conjunction with the heading "Date Published", it can be seen that the content file "index.content" 802 has been published and the date and time of publication is indicated by reference number 810. The designation
15 "NOT LIVE" still appears as indicated by reference number 806 and the "Needed" action is indicated as "GO LIVE" at reference number 812.

With reference additionally now to Fig. 10, a further follow-on user screen shot of main console of the preceding figures is shown illustrating the user viewable main console 800 thereof with respect to
20 the previously defined user ASCII content file denominated as file: "index.content" as indicated by reference number 802. As illustrated by reference number 810, the file has been published and reference number 812 indicates that the file has now been made "live". Further, as indicated by reference number 814, no further user action is now
25 indicated as being needed.

With respect to the preceding Figs. 8-10, a site is illustrated with but a single content file. However, it should be noted that the particular implementation of the web page content publisher of the present

invention can display multiple files and directories contained within the particular site.

With reference additionally now to Fig. 11, an additional representative user screen shot of the notify console 900 of the web page content publisher of the preceding figures is shown following completion of the action of "GO LIVE" for the file "index.content" 802. In this user viewable notify console 900, the publishing of the file "index.content" 802 is indicated by reference number 902 and the fact that it subsequently "went live" is indicated by reference number 904.

Following is the actual content file "index.content" in HTML format (index.html) as published through the particular embodiment of the web page content publisher of the present invention. The title and content entered by the user is again indicated by bold italicized print and the differences between the ASCII content file and this HTML content file are readily ascertained making it apparent that a much less technically competent individual can easily publish very professional web pages without having to be a highly trained HTML programmer.

```

20  <!-- NOTE: Change TITLE and META TAG values below to apply to
    your site. -->

    <HTML>

25  <HEAD><META NAME="DATE-PUBLISHED" CONTENT="Wed Oct 13 12:32:39
    MDT 1999">
    <META NAME="TEMPLATE-VERSION" CONTENT="v1.0">
    <META NAME="TEMPLATE-BUILD-DATE" CONTENT="Mon Oct 11 18:16:58
    MDT 1999">
30  <STYLE TYPE="text/css">
    BODY { font-size: 12pt; font-family: helvetica, lucida sans;
    font-variant: normal; font-style: normal; background-color:
    #FFFFFF; color: #000000; }
    .localheader { font-family: helvetica, lucida sans; font-size:
35  11pt; text-decoration: none; color: #FFFFFF; font-weight: bold;}
    .locallinkhl { font-family: helvetica, lucida sans; font-size:
    11pt; text-decoration: none; color: #FFFC00; font-weight:
    normal; }
    .locallink { font-family: helvetica, lucida sans; font-size:
40  11pt; text-decoration: none; color: #FFFFFF; font-weight:
    normal; }
  
```

[illegible]

```

5    <A HREF="http://web.domain/search/"><FONT CLASS="web" SIZE="2"
      COLOR="#666699" FACE="helvetica, lucida
      sans"><B>Search</B></FONT></A>
      <FONT CLASS="web" SIZE="2" COLOR="#666699" FACE="helvetica,
      lucida sans"><B>&nbsp;&middot;&nbsp;</B></FONT>
      <A HREF="http://web.domain/a-z/"><FONT CLASS="web" SIZE="2"
      COLOR="#666699" FACE="helvetica, lucida
      sans"><B>A&nbsp;to&nbsp;Z</B></FONT></A>
10   </TD>
      </TR>
      <TR>
        <TD>
          <A HREF="http://web.domain/"><FONT CLASS="webLogo" SIZE="6"
            COLOR="#666699" FACE="helvetica, lucida
            sans"><B>Web</B></FONT></A>
15   </TD>
        </TR>
      </TABLE>
    </TD>
20   </TR>
  </TABLE>
  <!-- END WEB GLOBAL LINKS -->
  </TD>
25   <TD BGCOLOR="#666699" VALIGN="TOP" WIDTH="95%">
    <!-- START HEADER BAR -->
    <TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
      <TR>
        <TD>
30     <TABLE BORDER="0" CELLSPACING="0" CELLPADDING="5">
        <TR>
          <TD NOWRAP>
            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
            </TD>
          </TR>
35     <TR>
        <TD>
          &nbsp;&nbsp;&nbsp;&nbsp;&~<FONT FACE="helvetica, lucida sans" SIZE="5"
            COLOR="#FFFFFF"><B>This is my New Page</B></FONT>
40     </TD>
        </TR>
      </TABLE>
    </TD>
    </TR>
  </TABLE>
45   <!-- END HEADER BAR -->
  </TD>
  <TR>
    <TD BGCOLOR="#666699" VALIGN="TOP" WIDTH="160">
50   <BR>
    <!-- START SEARCH FIELD -->
    <FORM ACTION="http://search.domain/query.html" METHOD="GET">
    <TABLE BORDER="0" CELLSPACING="0" CELLPADDING="1" ALIGN="CENTER"
      WIDTH="160">
55   <TR>
    <TD COLSPAN="2" VALIGN="TOP">
      <FONT FACE="helvetica, lucida sans" SIZE="3"
      COLOR="#FFFFFF">Search Web</FONT>
      </TD>

```

```

    </TR>
    <TR>
    <TD>
5    <INPUT TYPE="text" NAME="qt" SIZE="12">
    </TD>
    <TD>
    <FONT SIZE="3" COLOR="#FFFFFF" FACE="helvetica, lucida
15    sans"><INPUT TYPE="submit" NAME="name" VALUE="GO"></FONT>
    </TD>
10    </TR>
    <TR>
    <TD COLSPAN="2">
    <A
    HREF="http://search.domain/query.html?col=namer+region_1+region_
15    2&qp=&qsc=&qc=&pw=100%25&ws=1&q1=a&nh=25&lk=1&rf=0"><FONT
    CLASS="locallink" COLOR="#FFFFFF" SIZE="-1">Advanced
    Search</FONT></A>
    </TD>
    </TR>
20    </TABLE>
    </FORM>
    <!-- END SEARCH FIELD -->
    <!-- START SIDEBAR NAVIGATION -->
    <!-- END SIDEBAR NAVIGATION -->
25    </TD>
    <TD VALIGN="TOP" WIDTH="95%">
    <!-- START PAGE CONTENT TABLE -->
    <TABLE WIDTH="100%" BORDER="0" CELLSPACING="0" CELLPADDING="10">
    <TR>
30    <TD VALIGN="TOP">
    <!-- START PAGE CONTENT -->

    <!-- ***** START CONTENT HERE ***** -->

35    This is my content in fully-qualified HTML format.
    <P>
    This is more content.
    <P>
    This is even more content.
40    <!-- ***** END CONTENT HERE ***** -->

    <!-- END PAGE CONTENT -->
    </TD>
45    </TR>
    </TABLE>
    <!-- END PAGE CONTENT TABLE -->
    </TD>
    </TR>
50    <TR>
    <TD BGCOLOR="#666699" VALIGN="BOTTOM">
    <!-- START WEB GLOBAL LINKS -->
    <TABLE BORDER="0" CELLSPACING="1" CELLPADDING="0">
    <TR>
55    <TD BGCOLOR="#FFFFCC">
    <TABLE BORDER="0" CELLSPACING="0" CELLPADDING="5" WIDTH="172">
    <TR>
    <TD>

```

```

<A HREF="http://web.domain/"><FONT CLASS="web" SIZE="2"
COLOR="#666699" FACE="helvetica, lucida
sans"><B>Home</B></FONT></A>
5 <FONT CLASS="web" SIZE="2" COLOR="#666699" FACE="helvetica,
lucida sans"><B>&nbsp;&middot;&nbsp;</B></FONT>
<A HREF="http://web.domain/search/"><FONT CLASS="web" SIZE="2"
COLOR="#666699" FACE="helvetica, lucida
sans"><B>Search</B></FONT></A>
10 <FONT CLASS="web" SIZE="2" COLOR="#666699" FACE="helvetica,
lucida sans"><B>&nbsp;&middot;&nbsp;</B></FONT>
<A HREF="http://web.domain/a-z/"><FONT CLASS="web" SIZE="2"
COLOR="#666699" FACE="helvetica, lucida
sans"><B><A&nbsp;&to&nbsp;&Z</B></FONT></A>
15 </TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
20 <!-- END WEB GLOBAL LINKS -->
</TD>
<TD VALIGN="BOTTOM">
</TD>
</TR>
25 </TABLE>
<!--START PAGE FOOTER -->
<BR>
<FONT FACE="helvetica, lucida sans">[ <A
HREF="http://security.domain/info_protection/index.html">Company
30 Proprietary/Confidential</A>: Internal Use
Only]&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& [Feedback to <A
HREF="http://web.domain/feedback.html">Web Team</A> ]</FONT>
<!--END PAGE FOOTER -->
35 </BODY>
</HTML>

```

With reference additionally now to Fig. 12, a view 950 of the user content (index.html) as viewed from a representative web browser 958 is shown following being made "live". The title appears in the block indicated by reference number 952 while the content appears in the block outlined with the dashed line indicated by reference number 954. The header bar across the top and the side bar extending from the header bar down the left hand portion of the screen have been automatically supplied by the web page content publisher of the present invention.

While there have been described above the principles of the present invention in conjunction with specific object models and

templates, it is to be clearly understood that the foregoing description is made only by way of example and not as a limitation to the scope of the invention. Particularly, it is recognized that the teachings of the foregoing disclosure will suggest other modifications to those persons skilled in the relevant art. Such modifications may involve other features which are already known per se and which may be used instead of or in addition to features already described herein. Although claims have been formulated in this application to particular combinations of features, it should be understood that the scope of the disclosure herein also includes any novel feature or any novel combination of features disclosed either explicitly or implicitly or any generalization or modification thereof which would be apparent to persons skilled in the relevant art, whether or not such relates to the same invention as presently claimed in any claim and whether or not it mitigates any or all of the same technical problems as confronted by the present invention. The applicants hereby reserve the right to formulate new claims to such features and/or combinations of such features during the prosecution of the present application or of any further application derived therefrom.

What is claimed is:

CLAIMS:

1. A method for publishing web page content in accordance with a predetermined template comprising:
 - creating a content file;
 - 5 initializing a web page content publisher for inserting said predetermined template information into said content file;
 - requesting publication of said content file by said web page content publisher; and
 - publishing said content file in a web viewable format in
 - 10 accordance with said predetermined template.
2. The method of claim 1 wherein said step of creating a content file is carried out by the step of:
 - manually providing markup language computer program code.
3. The method of claim 2 wherein said step of manually
- 15 providing markup language computer code is carried out by markup language computer program code selected from a group comprising, but not limited to, HTML, XML and SGML.
4. The method of claim 1 wherein said step of creating a content file is carried out by the step of:
 - 20 furnishing a publishing tool; and
 - generating markup language computer program code with said publishing tool.
5. The method of claim 4 wherein said step of generating markup language computer program code generates markup language
- 25 computer program code selected from a group comprising, but not limited to, HTML, XML and SGML.

6. The method of claim 1 wherein said step of creating a content file is carried out by the step of:
providing markup language computer program code in plain text.

7. The method of claim 6 wherein said step of providing
5 markup language computer program code is carried out by markup language computer program code selected from a group comprising, but not limited to, HTML, XML and SGML.

8. The method of claim 1 wherein said predetermined template comprises at least one web navigation element.

10 9. A computer program product comprising a computer usable medium having computer readable program code devices embodied therein for publishing web page content in accordance with a predetermined template comprising:

computer readable program code devices configured to cause a
15 computer to effect creating a content file;

computer readable program code devices configured to cause a computer to effect initializing a web page content publisher for inserting said predetermined template information into said content file;

computer readable program code devices configured to cause a
20 computer to effect requesting publication of said content file by said web page content publisher; and

computer readable program code devices configured to cause a computer to effect publishing said content file in a web viewable format in accordance with said predetermined template.

25 10. The computer program product of claim 9 wherein said computer readable program code devices configured to cause said computer to effect creating a content file is carried out by:

computer readable program code devices configured to cause said computer to allow a user to manually provide markup language computer program code.

11. The computer program product of claim 10 wherein said
5 computer readable program code devices configure to cause said computer to allow a user to manually provide markup language computer program code provides said markup language computer program code from a group comprising, but not limited to, HTML, XML and SGML.

10 12. The computer program product of claim 9 wherein said computer readable program code devices configured to cause said computer to effect creating a content file is carried out by:

computer readable program code devices configured to cause said computer to implement a publishing tool; and

15 computer readable program code devices configured to cause said computer to generate markup language computer program code with said publishing tool.

13. The computer program product of claim 12 wherein said
20 computer readable program code devices configured to cause said computer to generate markup language computer program code generates said markup language computer program code from a group comprising, but not limited to, HTML, XML and SGML.

14. The computer program product of claim 9 wherein said
25 computer readable program code devices configured to cause said computer to effect creating a content file is carried out by:

computer readable program code devices configured to cause said computer to allow for the provision of markup language computer program code in plain text.

15 15. The computer program product of claim 14 wherein said computer readable program code devices configured to cause said computer to allow for the provision of markup language computer program code in plain text is carried out by said markup language computer program code selected from a group comprising, but not limited to HTML, XML and SGML.

10 16. The computer program product of claim 9 wherein said predetermined template comprises at least one web navigation element.

15 17. A computer system including a plurality of central processing units interconnected by a data network, at least two of said central processing units including a data entry device and an associated visual display for displaying selected data available on said data network; said computer system comprising:

20 a predetermined data network template accessible at a first one of said central processing units for defining at least a visual appearance of said selected data available on said data network; and

25 a network content publisher accessible at said first one of said central processing units for receiving a content file, said network content publisher operable for integrating said content file with said predetermined data network template to produce published data network content viewable on at least said associated visual display of a second one of said central processing units coupled to said data network in a format conforming to said predetermined data network template.

18. The computer system of claim 17 wherein said predetermined data network template comprises at least one data network navigation element for enabling a user of said data entry device associated with said second one of said central processing units to access information related to links displayed in conjunction with said published data network content.

19. The computer system of claim 17 wherein said content file may be input by a user of said data entry device associated with said first one of said central processing units.

20. The computer system of claim 17 wherein said content file may be retrieved from said data network by a user of said data entry device associated with said first one of said central processing units.

21. The computer system of claim 17 wherein said content file comprises markup language computer program code.

22. The computer system of claim 21 wherein said markup language computer program code is selected from a group comprising, but not limited to, HTML, XML and SGML.

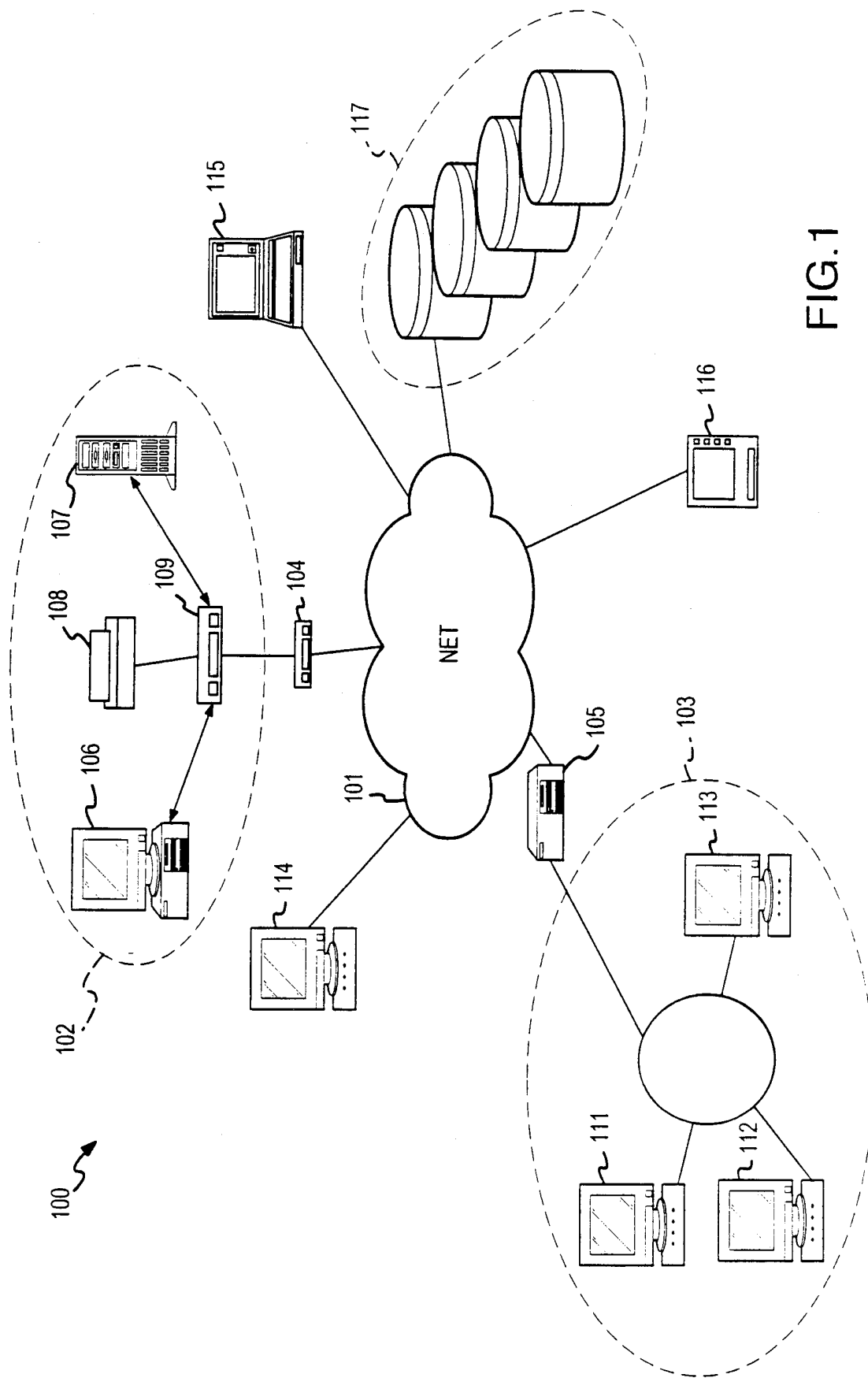


FIG.1

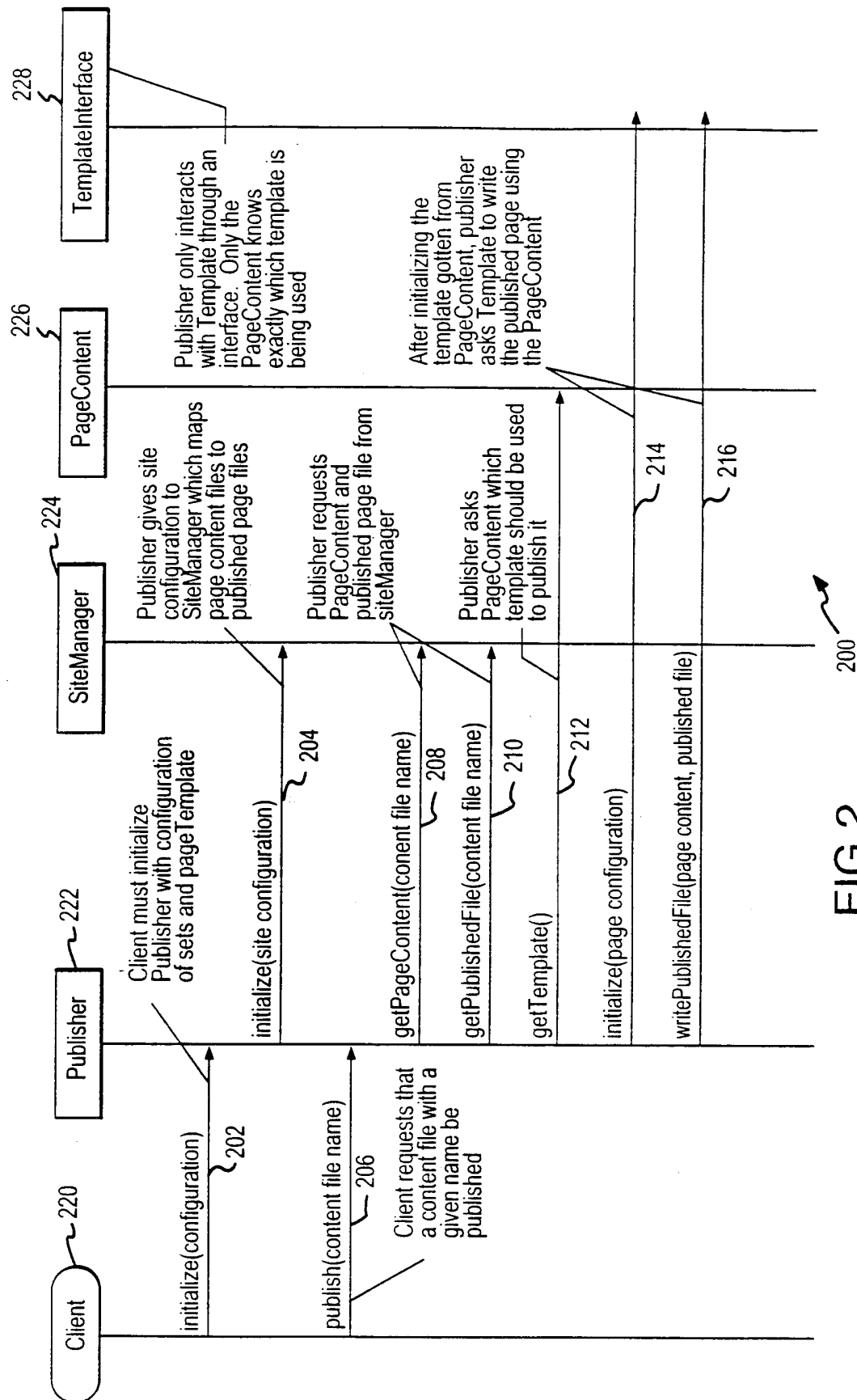
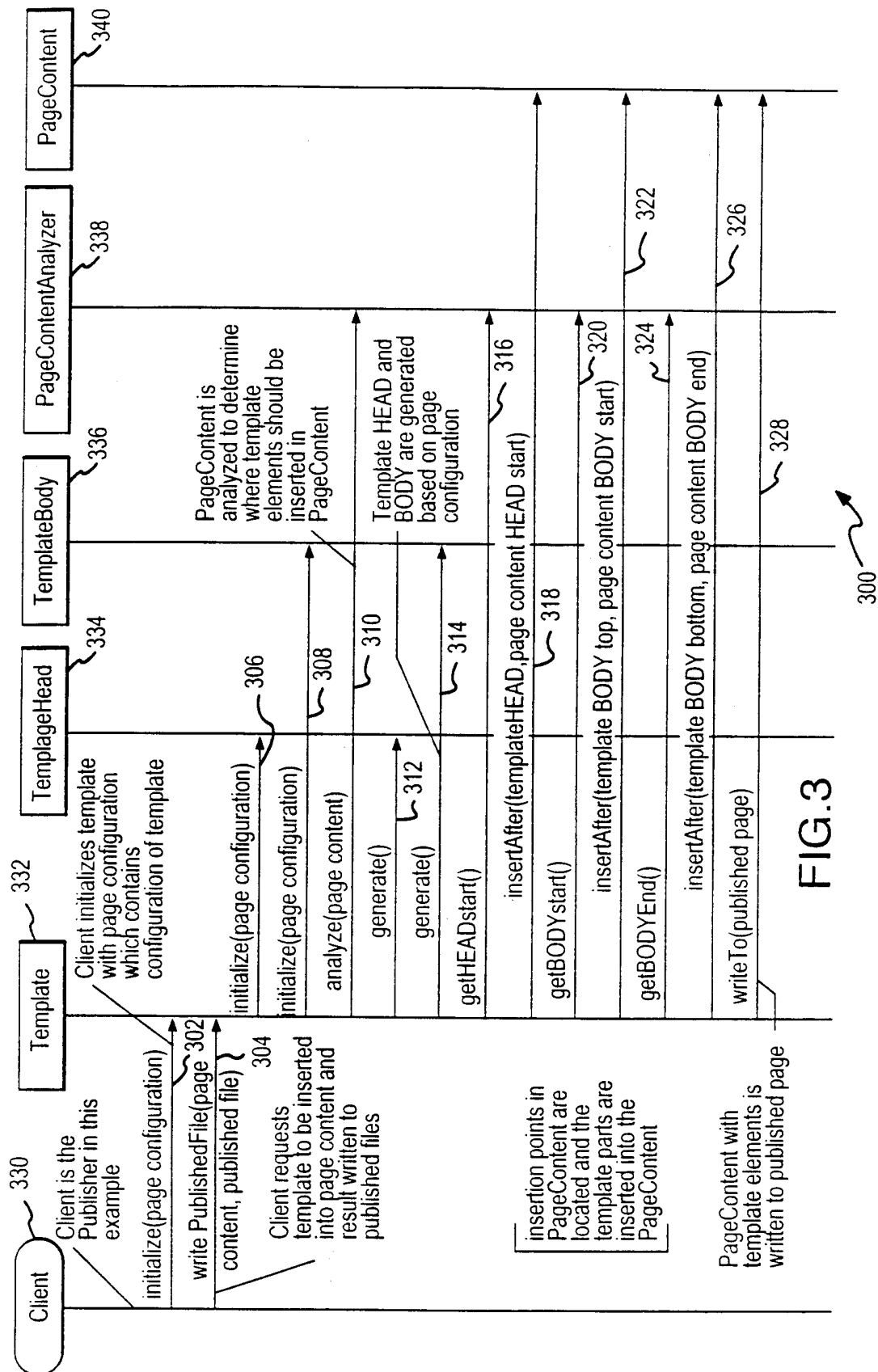
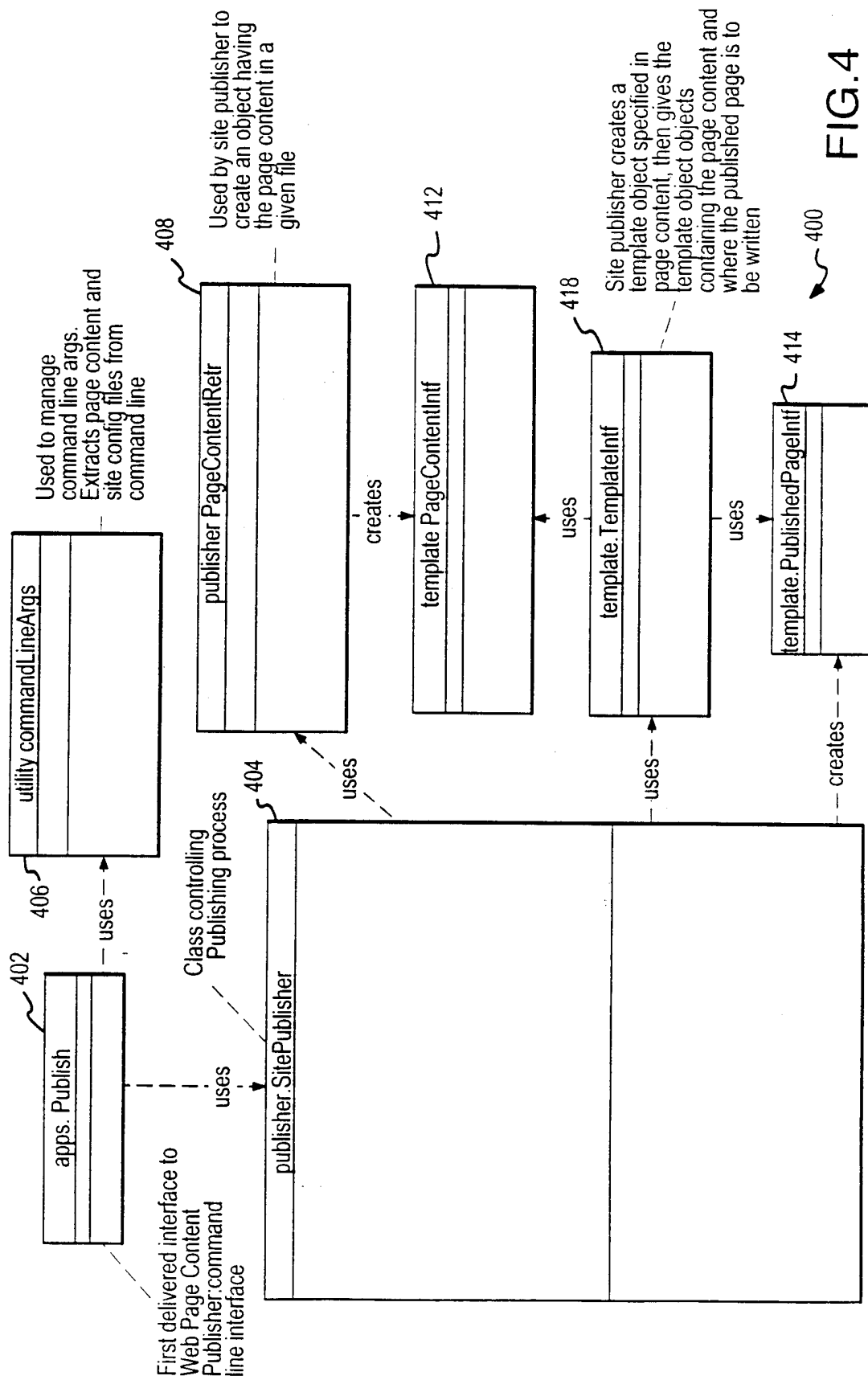


FIG.2





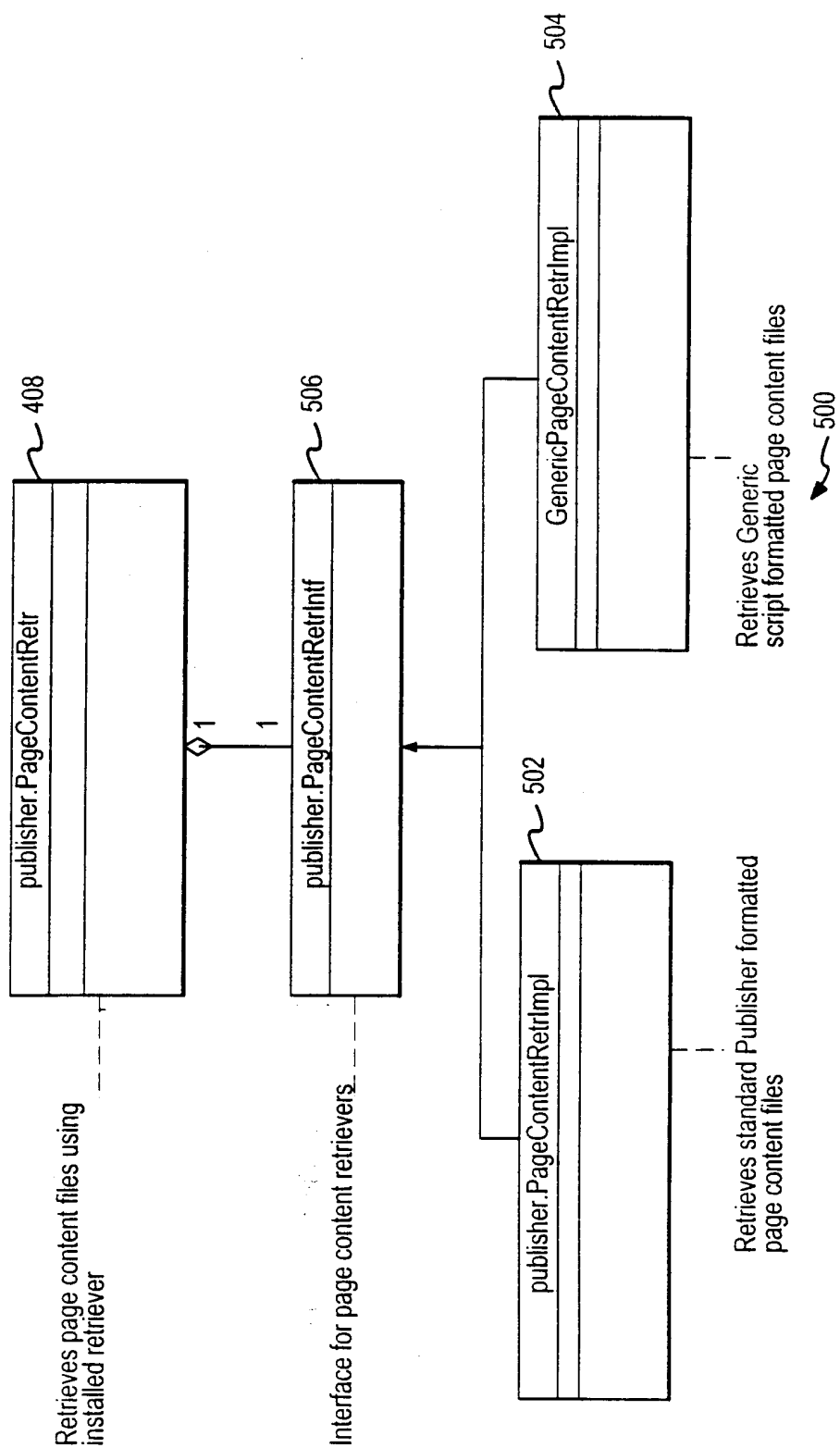


FIG.5

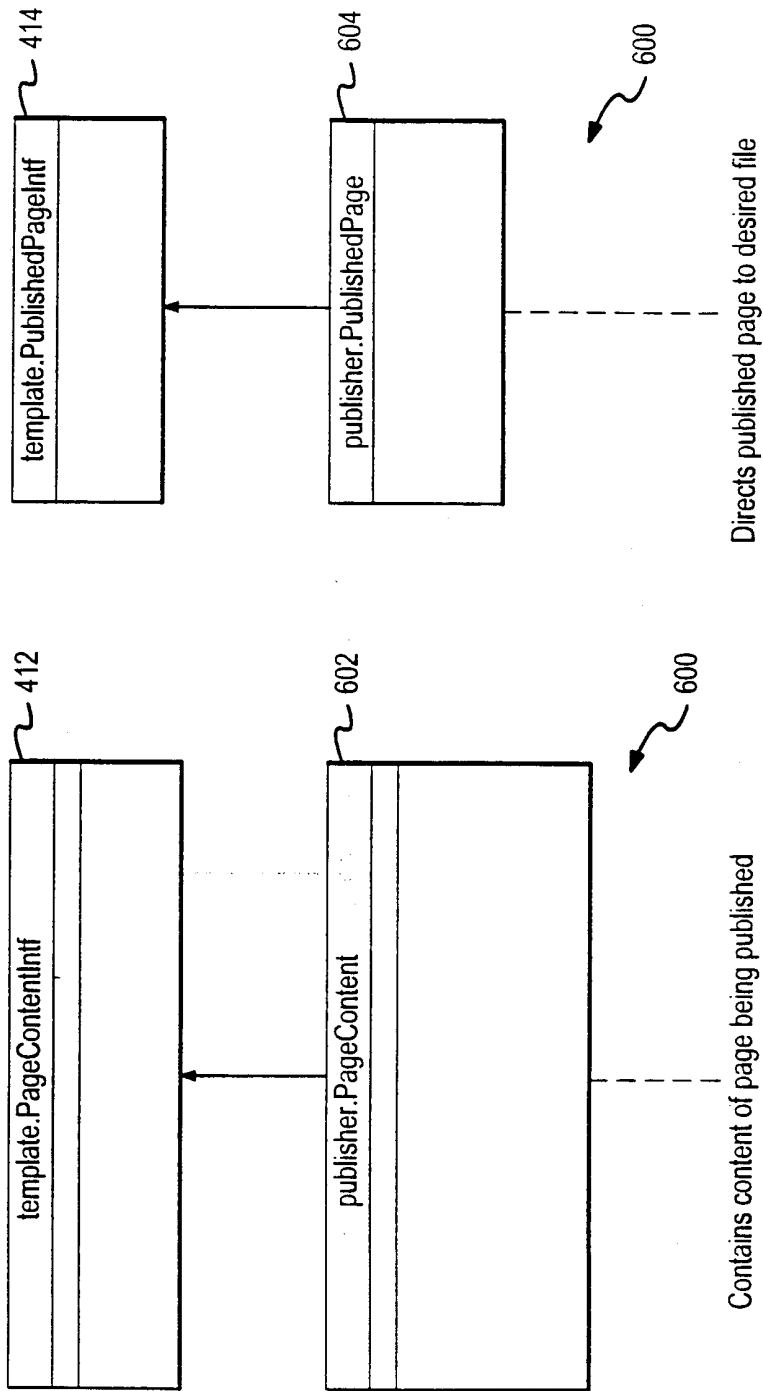


FIG. 6A

FIG. 6B

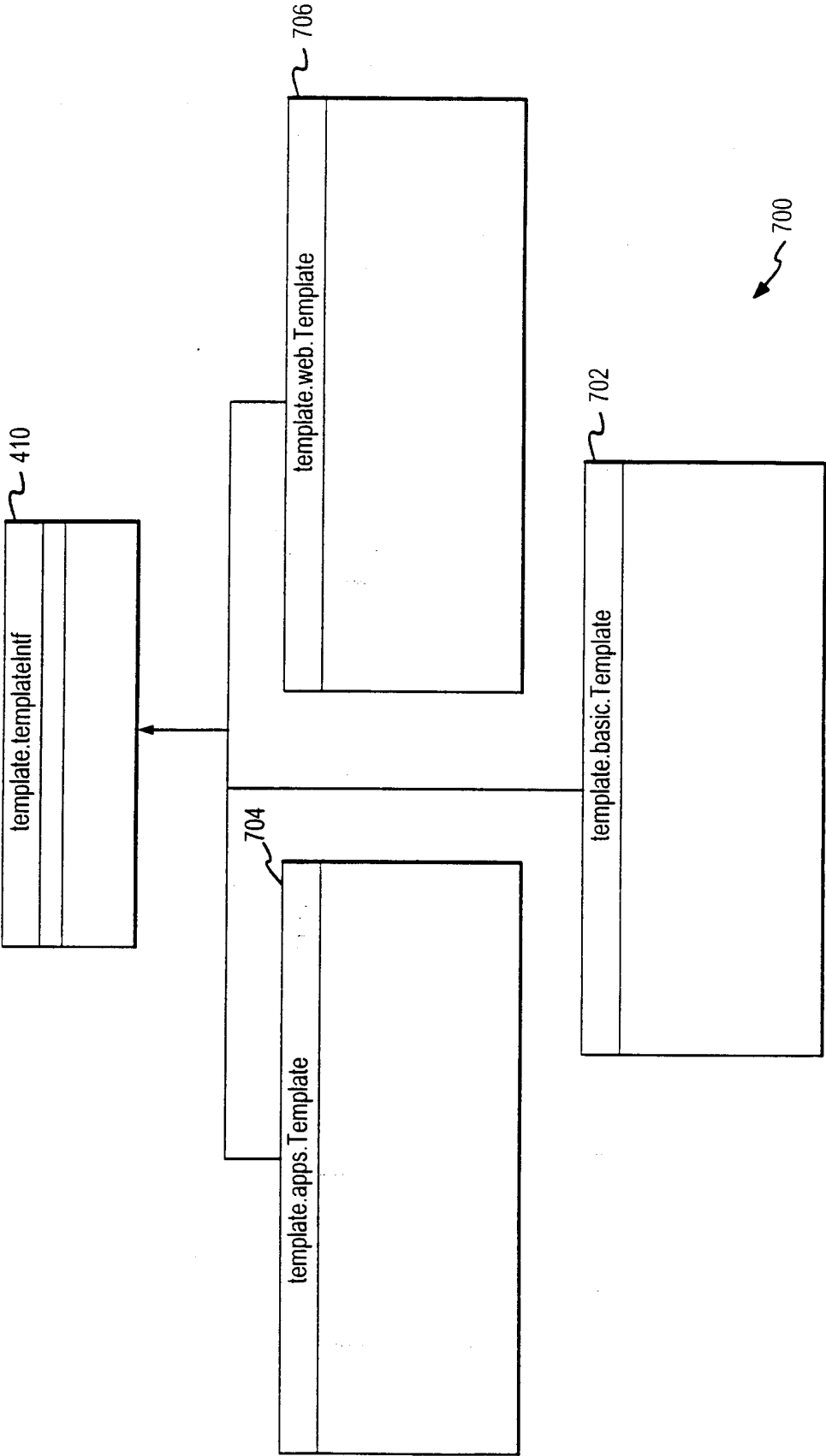


FIG.7A

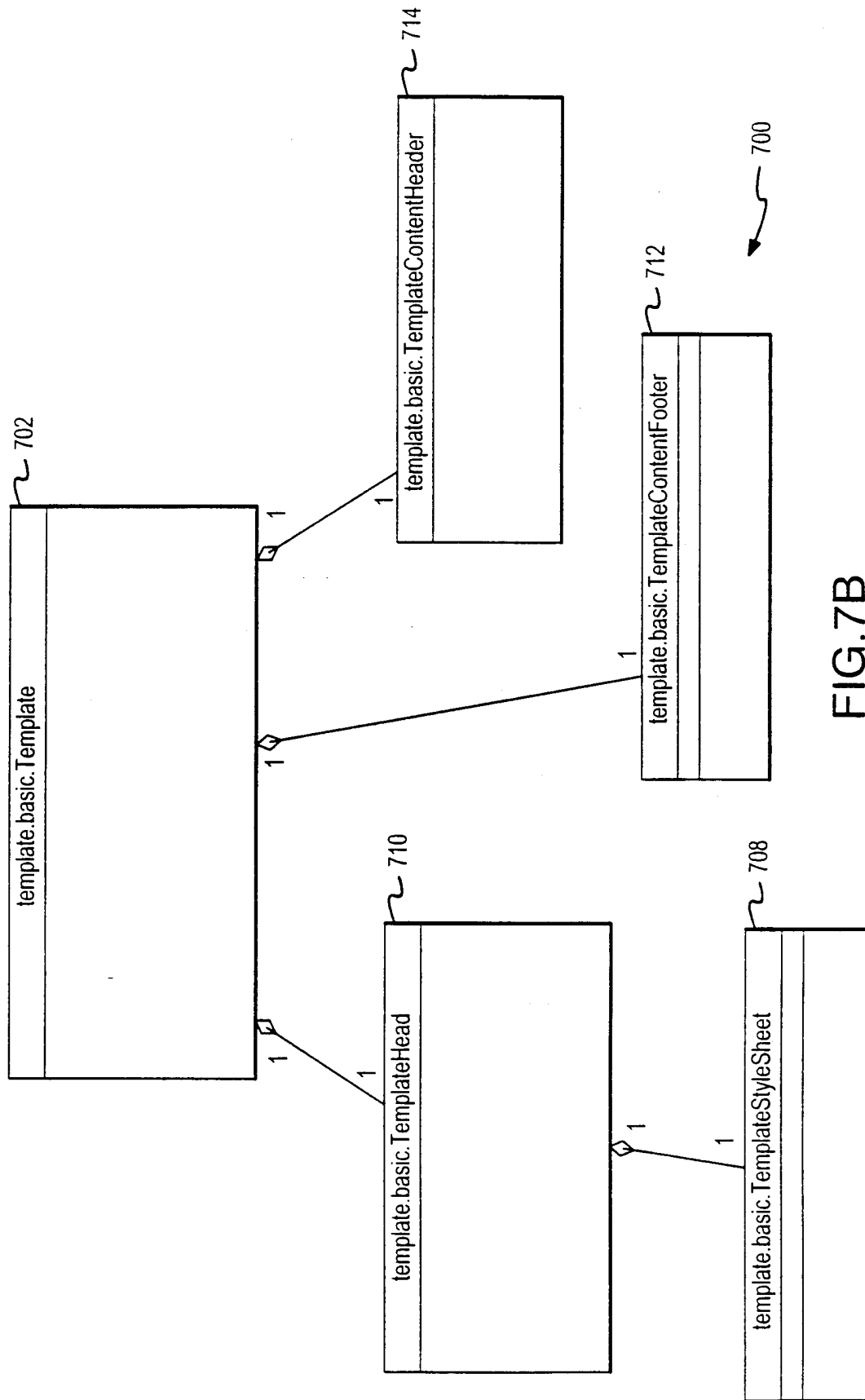


FIG.7B

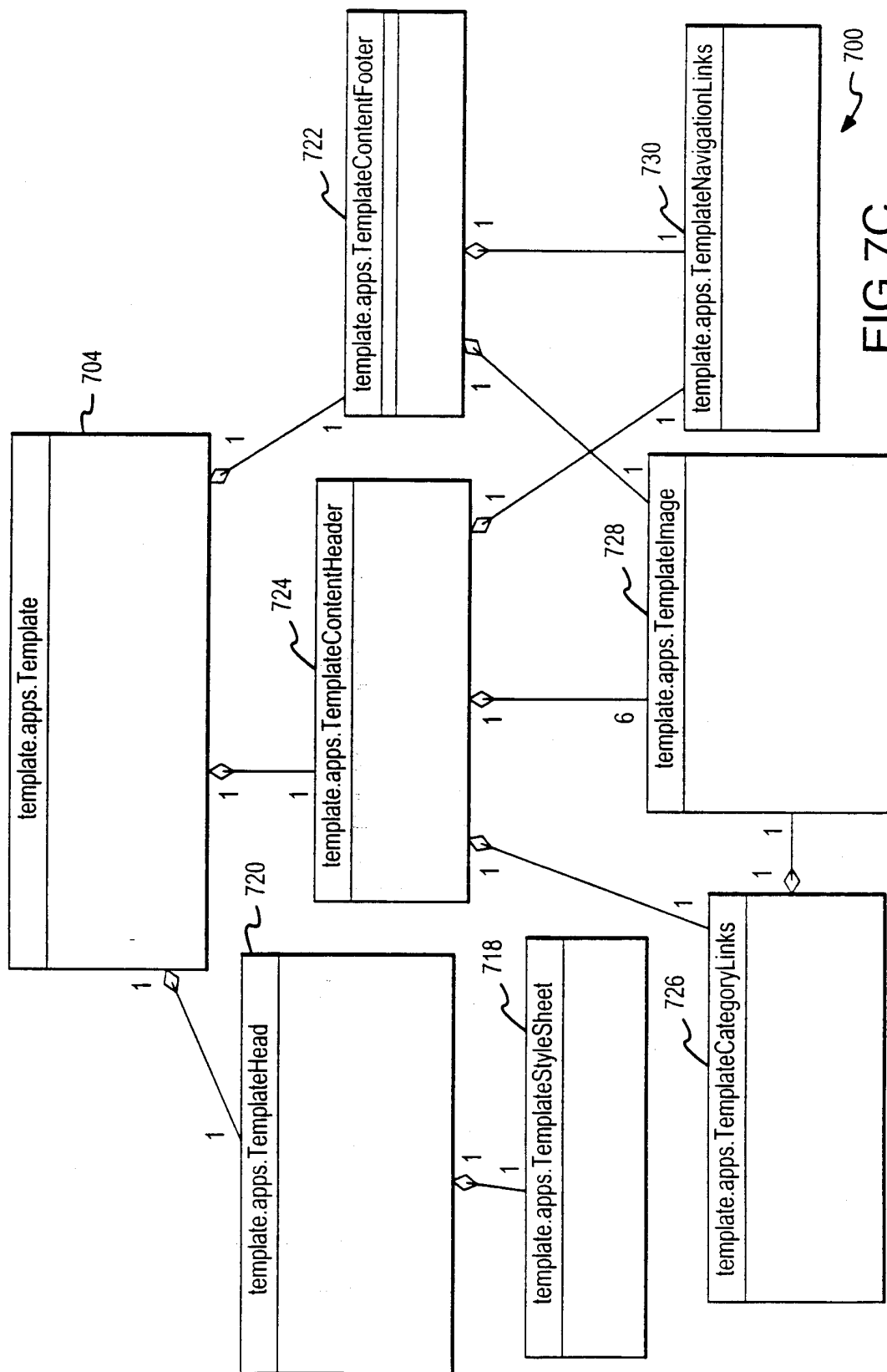
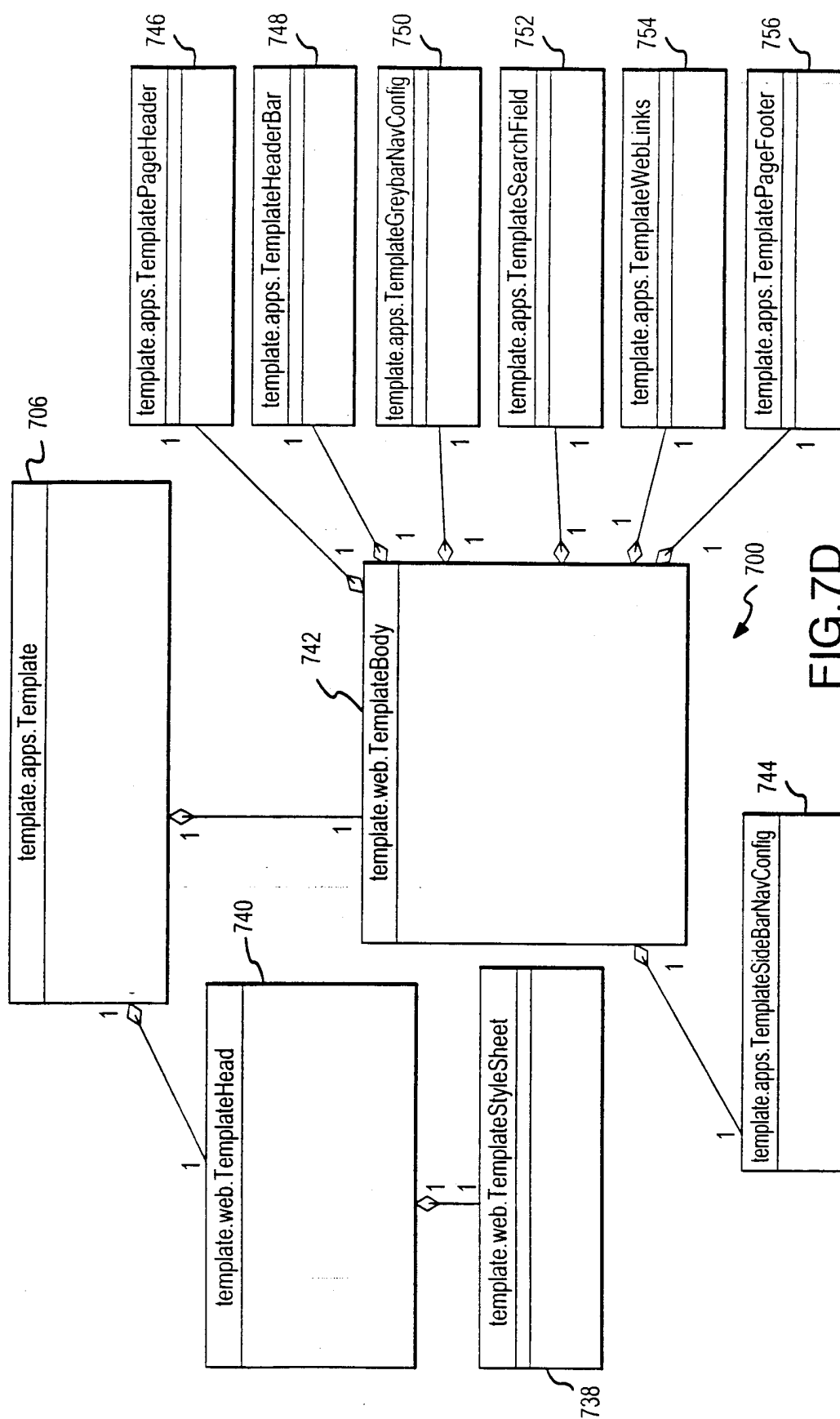


FIG.7C



Publish v1.00Content Root:/home/user/newsite

File Edit View Publish GoLive Tools Help

Directory in Site: I

Needed?	Content File	Date Content Mod	Date Published	Date Made Live
PUBLISH	index.content	1999-10-13 11:23:55	NOT PUBLISHED	NOT LIVE

808 802 804 806

800

FIG.8

Publish v1.00Content Root:/home/user/newsite

File Edit View Publish GoLive Tools Help

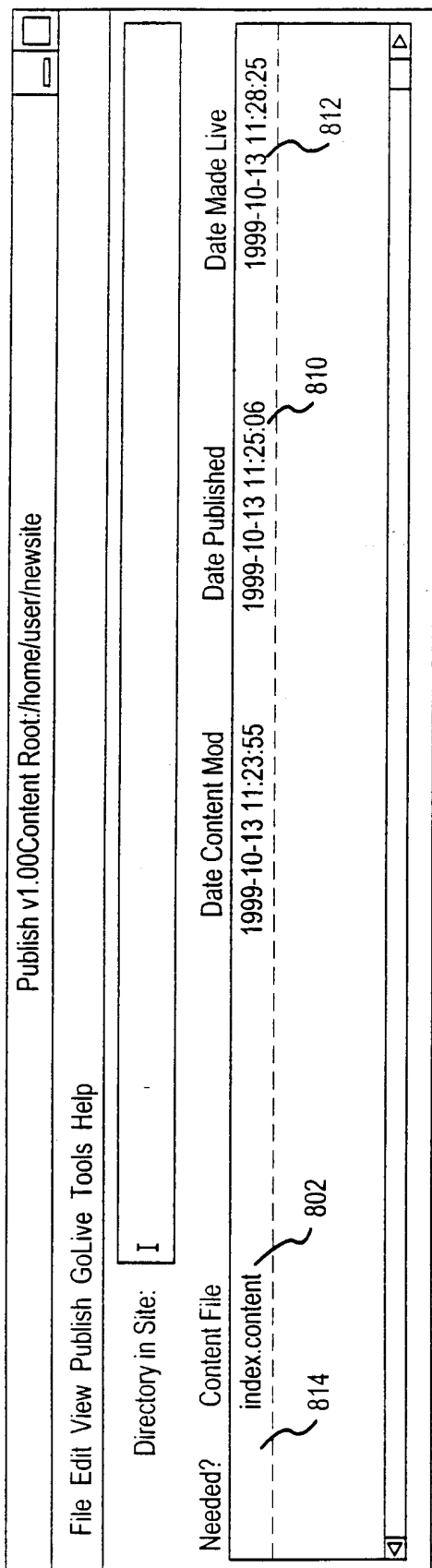
Directory in Site: I

Needed?	Content File	Date Content Mod	Date Published	Date Made Live
GO LIVE	index.content	1999-10-13 11:23:55	1999-10-13 11:25:06	NOT LIVE

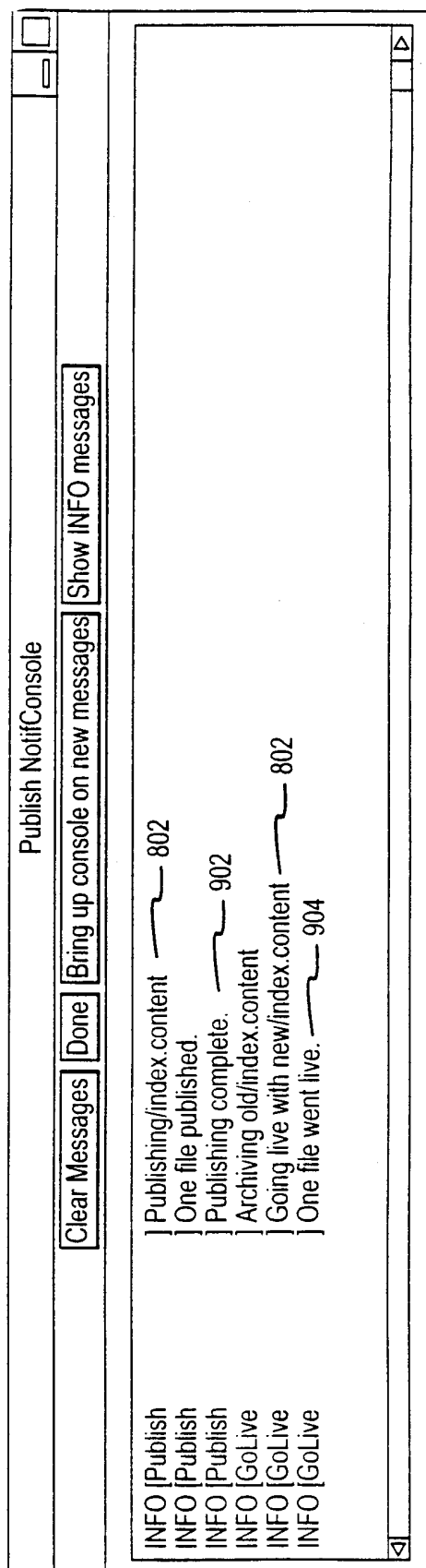
812 802 810 806

800

FIG.9



800



900

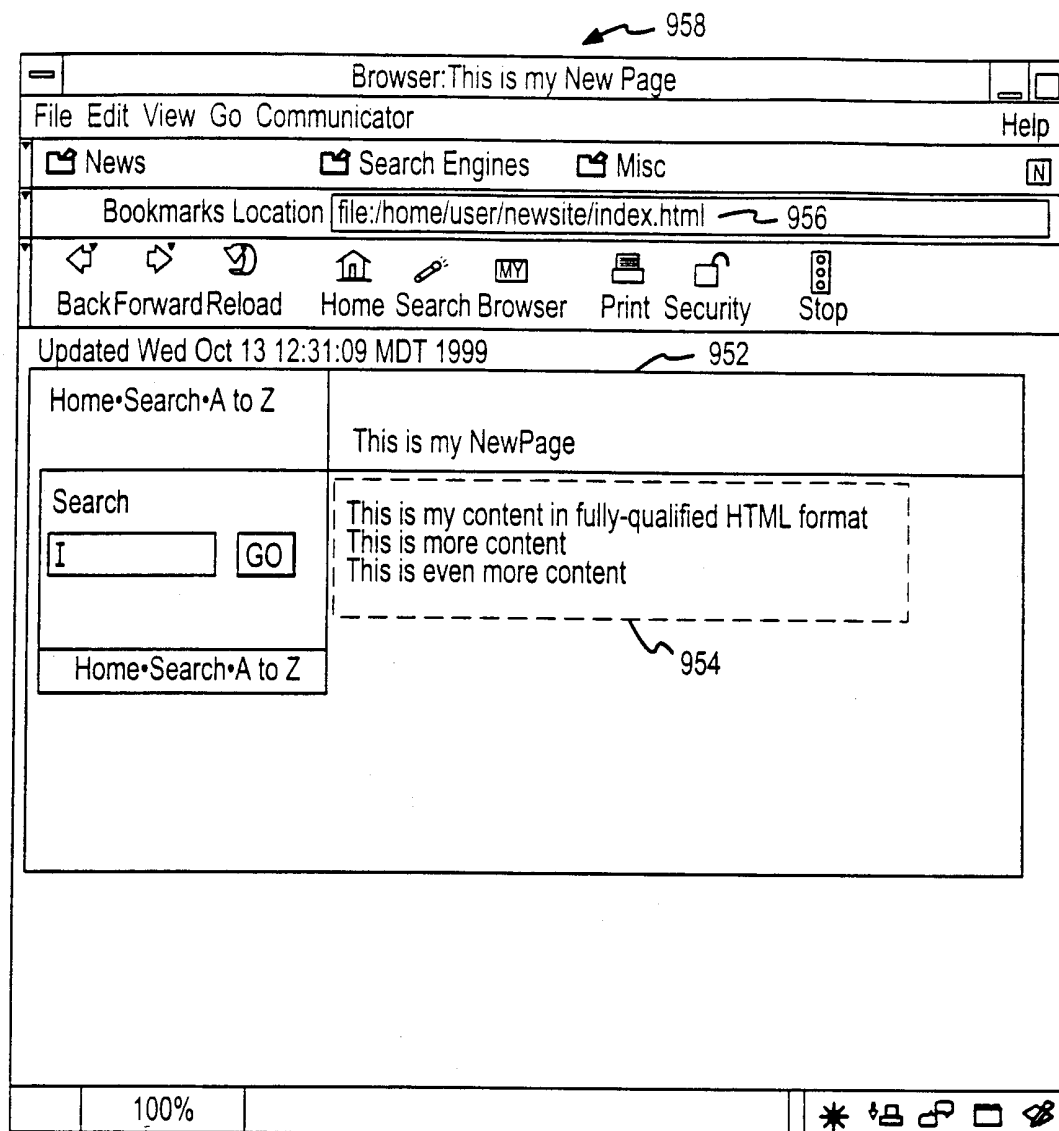


FIG. 12