



[12] 发明专利申请公开说明书

[21] 申请号 200410070356.9

[43] 公开日 2005 年 3 月 9 日

[11] 公开号 CN 1591425A

[22] 申请日 2004.7.29

[21] 申请号 200410070356.9

[30] 优先权

[32] 2003.7.31 [33] JP [31] 284475/2003

[71] 申请人 国际商业机器公司

地址 美国纽约

[72] 发明人 青木美穗子 福田进

[74] 专利代理机构 北京市中咨律师事务所

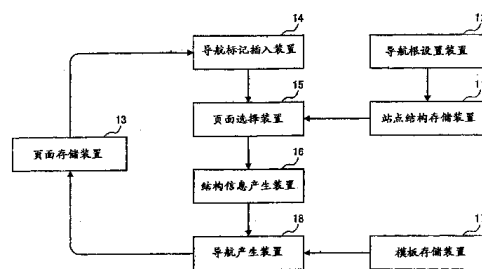
代理人 于 静 杨晓光

权利要求书 2 页 说明书 22 页 附图 25 页

[54] 发明名称 用于产生网站导航的方法

[57] 摘要

本发明公开了一种用于产生网站导航的方法。本发明的目的是产生围绕网站中的特定网页组的并且容易理解的导航。提供了用于存储关于网站树结构的信息的站点结构存储装置 11、用于将特定网页设置为导航根的导航根设置装置 12、用于存储关于网页的信息的页面存储装置 13、用于将包括等级指定的导航标记插入到网页中的导航标记插入装置 14、用于基于导航根和等级指定来选择将被包括在导航中的页面的页面选择装置 15、用于产生关于所选页面的结构信息的结构信息产生装置 16、用于存储导航模板的模板存储装置 17 和用于基于结构信息和模板来产生导航的导航产生装置 18。



1. 一种用于产生网站导航的方法，所述方法包括下面的步骤：
接收在具有网页的树结构的网站中所包括的特定网页的指定；以及
设置该特定的网页为用于产生导航的导航根，所述导航具有至少一个到局部树中的网页的链接，在所述局部树中该特定的网页被指定为其根。
2. 根据权利要求1的方法，还包括下面的步骤：
在局部树中选择将被显示在导航上的至少一个网页；以及
产生导航，所述导航包括至少一个到所选网页的链接和指示该所选网页在局部树中的位置的信息。
3. 根据权利要求2的方法，其中所述局部树是树结构的一部分，所述部分在包括所选网页的各局部树中具有最低等级的根。
4. 根据权利要求2的方法，其中在所述选择步骤中所选网页的范围由树结构中的一等级来确定。
5. 根据权利要求1的方法，其中所述树结构以XML来表示，并且通过XML标记插入导航根的信息。
6. 根据权利要求2的方法，其中通过使用模板在所述产生步骤中产生导航。
7. 一种具有机器可读程序代码的存储介质，所述机器可读程序代码用于使计算机系统实现包括下面步骤的方法：
接收在具有网页的树结构的网站中所包括的特定网页的指定；以及
设置该特定的网页为用于产生导航的导航根，所述导航具有至少一个到局部树中的网页的链接，在所述局部树中该特定的网页被指定为其根。
8. 根据权利要求7的存储介质，还包括用于使计算机系统实现下面步骤的程序代码：
在局部树中选择将被显示在导航上的至少一个网页；以及
产生导航，所述导航包括至少一个到所选网页的链接和指示局部树中该所选网页的位置的信息。

9. 根据权利要求8的存储介质,其中所述局部树是树结构的一部分,所述部分在包括所选网页的各局部树中具有最低等级的根。

10. 根据权利要求8的存储介质,其中在所述选择步骤中所选网页的范围由树结构中的一等级来确定。

11. 根据权利要求7的存储介质,其中所述树结构以XML来表示,并且通过XML标记插入导航根的信息。

12. 根据权利要求8的存储介质,其中通过使用模板在所述产生步骤中产生导航。

用于产生网站导航的方法

技术领域

本发明涉及自动产生显示在网页上的导航(navigation)的导航产生装置或者方法,并且特别涉及产生包括链接的导航的导航产生装置或者方法,所述链接是至包含在构成网站树结构的一部分的局部树中的网页的链接。在本说明书中,“导航”是为了帮助用户移到目标网页以及容易地了解当前网页在网站内的位置的目的而显示在网页上的一组信息,这组信息包括到其它网页的超链接(下文中简称为“链接”)。

背景技术

目前在因特网上存在着大量的网站。有包括用于商业用途和私人用途的各种各样的网站。特别是在商业网站中能见到许多使用与可用性有关的网站的网站。一种在网页上显示导航的方法已经被实施了,该方法已知为这种装置中的一种。详细内容请参考日本的已公布但未审查的专利申请10-149372(图2至4) (“专利文献1”)。

有些现有的网页生成软件具有产生显示在网页上的导航的功能。这些软件实现作为用于选择导航中将包括的链接的方法的第一方法和第二方法,在所述第一方法中,选择到包括在目标网站中的“所有”网页的链接,在所述第二方法中,选择到基于与目标网页的相对关系(父页面、子页面等)而被指定的那些网页的链接。

在描述了传统技术之一的专利文献1中,没有对显示在网页上的导航的建立进行描述。

如在专利文献1中所描述的导航可以通过使用现有的软件被相对容易地建立。可是却很难为大规模的网站建立令人满意的导航。

大规模的网站可能包括多组具有不同主题的网页，例如，一组关于产品 A 的网页和一组关于产品 B 的网页。尽管这些组网页被链接到了一个共同的顶层页面上。在为这样的网站建立导航时，到关于产品 A 的网页的链接和到关于产品 B 的网页的链接被包括在所有网页的导航中。非常有必要将到关于产品 B 的网页的链接从关于产品 A 的网页的导航中排除出来，以及将到关于产品 A 的网页的链接从关于产品 B 的网页的导航中排除出来。然而，第一种方法不能够使产生的导航满足这样的需求。

通过使用现有软件中的第二方法有可能暂时地满足这种需求。然而，在第二方法的使用中，又会产生下面描述的另外的问题。

即，在使用根据相对关系定义风格的模板来产生导航的情况下，这些导航以相关于它们显示所在的网页的不同风格而显示，即使它们是为到一个网页的连接而建立的。

例如，假设这样的情况，其中以风格 A 来定义到导航显示所在的网页的父页面的链接的显示；以风格 B 来定义到导航显示所在的该网页的链接或者到导航显示所在的该网页的兄弟网页的链接的显示；以风格 C 来定义到导航显示所在的网页的子页面的链接的显示，并且其中在一个网站里“1.htm”的子页面是“2.htm”；“2.htm”的子页面是“3.htm”；“3.htm”的子页面是“4.htm”；“4.htm”的子页面是“5.htm”。在这种情况下，当导航被建立在“2.htm”、“3.htm”和“4.htm”上时，在“2.htm”上的导航中到“3.htm”的链接以风格 C 来显示，在“3.htm”上的导航中到“3.htm”的链接以风格 B 来显示，在“4.htm”上的导航中到“3.htm”的链接以风格 A 来显示。即，以和链接显示所在的页面相对应的不同风格来显示链接，尽管它们连接到同一个页面“3.htm”。在这种情况下，也不能够满足用户从一个网页到另一个网页以及从另一个网页返回到一个网页时要求产生容易理解的导航的需求。

实现了本发明来解决上述技术问题，并且本发明的目的是使得能够产生围绕网站中特定的网页组并且容易理解的导航。

发明内容

被提供用来实现上述目的的本发明使用仅仅包括到一网页的链接的信息，所述网页被包括在网站中的特定的网页组中，并且使得能够识别该网页在网页组中的位置。即，根据本发明的导航产生装置具有选择至少一个包含在形成网站树结构的一部分的预定局部树中的网页的页面选择装置和产生导航的导航产生装置，所述导航包括到由页面选择装置所选择的页面的链接和指示该网页在预定局部树中的位置的信息。导航产生装置通过使用模板来为到所选网页的链接确定风格，所述风格是根据预定局部树中的网页的位置的风格，在所述模板中定义了所述根据局部树中的位置的风格。

本发明还可以作为一种信息管理装置来理解，所述信息管理装置对导航的产生进行设置。本发明的这种信息管理装置具有接收网站中包含的特定网页的指定的接收装置以及对导航的产生进行设置的设置装置，所述导航包括至少一个到包括在局部树中的一网页的链接，所述局部树以接收装置接收到的指定的所述网页作为根。

另外，本发明可以作为一种用于导航产生装置的方法来理解。本发明的这种导航产生方法包括基于存储在预定存储中的网站树结构的信息来选择至少一个包含在形成网站树结构的一部分的局部树中的网页的步骤以及向预定的存储输出导航的步骤，所述导航包括到所选择的网页的链接和指示网页在预定局部树中的位置的信息。输出导航的步骤包括通过使用模板来为到所选择的网页的链接确定风格，所述风格是根据网页在预定的局部树中的位置的风格，在所述模板中定义了所述根据局部树中的位置的风格。

另外，本发明还可以作为一种用于信息管理装置的方法来理解，所述信息管理装置对导航的产生进行设置。本发明的这种站点结构管理方法包括接收网站中包含的特定网页的指定的步骤，以及根据关于存储在预定存储中的该网站的树结构的信息来对包括至少一个到网页的链接的导航的产生进行设置的步骤，所述网页包括在以被指定的特定的网页作为根的局部树中。

本发明还可以作为计算机程序来理解，所述计算机程序使得计算机能

够实现产生导航的功能。本发明的这种计算机程序使得计算机能够实现选择至少一个包括在构成网站树结构的一部分的预定局部树中的网页的功能，以及产生导航的功能，所述导航包括到所选网页的链接和指示该网页在预定的局部树结构中的位置的信息。产生导航的功能包括通过使用模板来为到所选网页的链接确定风格，所述风格是根据该网页在预定的局部树中的位置的风格，在所述模板中定义了所述根据局部树中的位置的风格。

另外，本发明还可以作为一种用于使计算机能够实现对导航的产生进行设置的功能的计算机程序来理解。本发明的这种计算机程序使得计算机能够实现接收包括在网站中的特定网页的指定的功能，以及对导航的产生进行设置的功能，所述导航包括至少一个到包括在局部树中的网页的链接，所述局部树以指定的特定网页作为根。

本发明使得能够产生围绕网站中的特定网页并且容易理解的导航。

附图说明

图 1 是示出了本发明实施例中的导航产生装置的整个配置的框图；

图 2 是示出了该实施例中用于导航产生装置的操作过程的流程图；

图 3 是示出了该实施例的操作中导航根设置处理的流程图；

图 4 是示出了该实施例的操作中导航标记插入处理的流程图；

图 5 是示出了该实施例的操作中自动导航产生处理的流程图；

图 6 是示出了用在该实施例的例子中的站点结构的示意图；

图 7 是示出了用在该实施例的例子中的代表站点结构的站点结构信息的示意图；

图 8 是示出了用在该实施例的例子中的站点结构的示意图；

图 9 是示出了用在该实施例的例子中的代表站点结构的站点结构信息的示意图；

图 10 是示出了用在该实施例的例子中的站点结构的示意图；

图 11 是示出了用在该实施例的例子中的代表站点结构的站点结构信息的示意图；

图 12 是示出了该实施例的例子中编辑器上屏幕显示的例子的示图；

图 13 是示出了该实施例的例子中的导航标记插入对话框的例子的示图；

图 14 是示出了该实施例的例子中的插入导航标记后的页面描述的例子的示图；

图 15 是示出了该实施例的例子中编辑器上屏幕显示的例子的示图；

图 16 是示出了该实施例的例子中的模板的例子的示图；

图 17 是示出了该实施例的例子中的插入导航后的页面描述的例子的示图；

图 18 是示出了该实施例的例子中编辑器上屏幕显示的例子的示图；

图 19 是示出了该实施例的例子中的导航标记插入对话框的例子的示图；

图 20 是示出了该实施例的例子中的模板的例子的示图；

图 21 是示出了该实施例的例子中的插入导航标记后的页面描述的例子的示图；

图 22 是示出了该实施例的例子中编辑器上屏幕显示的例子的示图；

图 23 是示出了该实施例中导航设置改变对话框的示图；

图 24 是示出了该实施例中导航设置改变对话框的示图；以及

图 25 是示出了该实施例中导航设置改变对话框的示图。

符号描述

11... 站点结构存储装置

12... 导航根设置装置

13... 页面存储装置

14... 导航标记插入装置

15... 页面选择装置

16... 结构信息产生装置

17... 模板存储装置

18...导航产生装置

具体实施方式

将参考附图来详细地描述本发明的实施例。在下面的描述中，“网页”简称为“页面”。

将首先描述该实施例中导航产生装置的配置。

图1是示出了该实施例中导航产生装置的整个配置的图示。

如图1中所示，该导航产生装置具有站点结构存储装置11、导航根设置装置12、页面存储装置13、导航标记插入装置14、页面选择装置15、结构信息产生装置16、模板存储装置17和导航产生装置18。

站点结构存储装置11是用于存储关于网站的站点结构的信息(此后称为“站点结构信息”)的装置。网站通常具有树结构，该树结构具有顶层页面作为根以及末端页面作为叶子，并且在树结构中从根跟踪到一个叶子所通过的每个页面是一个节点。站点结构信息以一种适合表示树结构的方法表示出来。

在该实施例中，为包括在网站中的每个页面定义等级(该等级此后被称为“站点结构中的等级”)。更明确地，等级是以这种方式来定义的：顶层页面的等级是1，顶层页面的子页面的等级是2，以及顶层页面的该子页面的子页面的等级是3。代表等级的数字值在从根到叶子的方向上递增。

导航根设置装置12是用于将包括在存储于站点结构存储装置11的站点结构信息中的特定网页设置为导航根的装置。更明确地，导航根设置装置12包括用于接收特定网页的指定的接收装置和用于设置该接收网页为导航根的设置装置。由设置装置进行的设置不限于将页面设置为导航根的情况。由设置装置进行的设置还可以作为对导航的产生进行设置来理解，所述导航包括到包括在以特定网页作为其根的局部树中的网页的链接。

“导航根”是作为一个导航基点的页面。网站的树结构可以被分成几个较小的局部树。如果某个网页被设置为一导航中的导航根，构成以该网页作为其根的特定局部树的每个网页都包括在该导航中。

在该实施例中，等级是为包括在导航中的每个页面定义的（该等级此后将被称为“导航中的等级”）。更明确地，等级是以这样的方式来定义的：导航根的等级是1，导航根的子页面的等级是2，以及导航根的子页面的子页面的等级是3。代表等级的数字值在从导航根到叶子的方向上递增。

页面存储装置13是用于存储包括在网站中的各个页面的描述（HTML文件或者类似的东西）的装置。

导航标记插入装置14是用于在存储于页面存储装置13的页面描述中插入导航标记的装置。

页面选择装置15是用于从存储在站点结构存储装置11的站点结构信息所包括的页面中选择将被包括在导航里的页面的装置。

结构信息产生装置16是用于产生结构信息的装置，所述结构信息包括页面选择装置15所选择的页面的标签、与所选页面有关的路径、所选页面和其它页面之间的关系、等等。

模板存储装置17是用于存储模板的装置，在所述模板中定义了导航的风格。由应用程序提供的模板或者由用户原始形成的模板可以作为存储在模板存储装置17中的模板来使用。

导航产生装置18是用于根据由结构信息产生装置16产生的结构信息和存储在模板存储装置17中的模板来产生导航的装置。

普通计算机系统的硬件配置可以作为该实施例中导航产生装置的硬件配置来使用。即，如果硬件配置包括中央处理器（CPU）和主存储器，并且其中CPU和主存储器通过总线被连接到辅助存储装置，这样的硬件配置就可以满足需要了。辅助存储装置可以是硬盘、软盘、MO（磁光盘）、CD-ROM或者类似的东西。

用于实现本实施例的计算机程序被存储在辅助存储装置中。中央处理器将该计算机程序读到主存储器中，并且执行该计算机程序以实现导航根设置装置12、导航标记插入装置14、页面选择装置15、结构信息产生装置16和导航产生装置18中的每一个。可以采用类似的操作来实现控制向站点结构存储装置11、页面存储装置13和模板存储装置17中每一个的存

储区域输入和从所述存储区域中输出的功能。

站点结构存储装置 11、页面存储装置 13 和模板存储装置 17 中每一个的存储区域功能可以采用辅助存储器或者主存储器来实现。将存储在这些存储装置的每一个中的数据可以通过由例如因特网的通信线路下载下来而提供。

例如键盘和指点设备的输入设备可以连接到导航产生装置，并且例如显示器的输出设备可以连接到导航产生装置。此外，可选择地采用这样的配置，在所述配置中起输入设备和输出设备作用的终端设备是与导航产生装置分开提供的，并且通过通信线路进行这些设备和装置之间的通信。

在任何一种所述配置中，都需要用于在输入设备和输出设备上实现图 1 中未示出的用户接口的功能。这些功能例如是以这样的形式来显示网站树结构的功能，即包括在网站中的页面可以相互区分出来，以及通过这样的形式来显示关于网站的信息的功能，即可以了解包括在网站中的哪个页面被设置为导航的根。

现将描述在该实施例中导航产生装置的操作。

图 2 是示出了由用户在导航产生装置上所执行的操作的流程的流程图。

首先，用户建立站点结构并且将关于站点结构的站点结构信息存储在站点结构存储装置 11（步骤 201）中。然后用户使用导航根设置装置 12 将存储于站点结构存储装置 11 的站点结构信息中所包括的一个特定的页面设置为一个导航根（步骤 202）。

在此阶段，页面存储装置 13 维护着存储于站点结构存储装置 11 的站点结构信息中所包括的页面的页面描述，作为其存储内容。这些页面描述可以与建立站点结构同时提供或者在建立站点结构之前或之后的任选的某个时间提供。在这些准备之后，用户执行操作以建立将被显示在页面上的导航。

用户首先查看存储在页面存储装置 13 中的一个页面。即，如果数字“i”表示页面的索引，用户确定数字“i”是否等于或小于包括在站点存储结构

信息中的页面的数目（步骤 203）。

如果数字“i”不等于或小于页面的数目，那么过程结束。如果数字“i”等于或小于页面的数目，用户确定导航是否应该被插入到该页面中（步骤 204）。如果用户希望将导航插入到页面中，那么用户通过使用导航标记插入装置 14 将导航标记插入到页面中（步骤 205），并随后输入指令以自动产生导航（步骤 206）。如果页面不是用户希望插入导航的页面，或者当自动建立导航结束时，用户查看存储在页面存储装置 13 中的下一个页面，并执行与上面描述同样的处理。即，在数字 i 上加 1（步骤 207）并且过程返回到步骤 203。如果数字 i 等于或小于包括在站点结构信息中的页面的数目，重复步骤 204 到步骤 206 中的处理。

上面描述了该实施例中用户操作的流程。

将更加详细地描述在图 2 中示出的并且由用户操作开始的软件处理操作中的特定于本实施例的操作。

将首先详细地描述图 2 示出的步骤 202 中的导航根设置处理。

图 3 是示出了作为导航根设置装置 12 的操作而执行的导航根设置处理的流程图。

当给出指令来将存储在站点结构存储装置 11 的站点结构信息中一个特定页面设置为导航根时，导航根设置装置 12 首先确定将被设置的页面（步骤 301）。导航根设置装置 12 于是读出存储在站点结构存储装置 11 中的站点结构信息，并在读出的站点结构信息中添加该页面为导航根的信息，作为该页面节点的属性（步骤 302）。导航根设置装置 12 最后输出所指定的页面已经被设置为导航根的信息（步骤 303）。

于是由导航根设置装置 12 实现了导航根设置处理。

接着将详细地描述图 2 示出的步骤 205 中的导航标记插入处理。

图 4 是示出了作为导航标记插入装置 14 的操作而执行的导航标记插入处理的流程图。

导航标记插入装置 14 首先从存储于页面存储装置 13 的页面中读取并保持将被插入导航的页面。导航标记插入装置 14 还接收建立导航时所使用

的模板的选择（步骤 401），以及还接收用于选择将被包括在导航中的页面的信息的指定，例如，指示页面等级范围的开始和结束等级（步骤 402）。导航标记插入装置 14 向预先读出的页面写入导航标记，所述导航标记包括关于所选模板的信息以及用于从页面中选择的页面选择信息（步骤 403）。最后，导航标记插入装置 14 输出导航标记已经被插入的信息（步骤 404）。

于是由导航标记插入装置 14 实现了导航标记插入处理。其后，当用户请求保存页面或者明确请求自动产生导航时，进行到页面选择装置 15 的控制的转换。在该转换的时候，导航标记所插入的页面被从导航标记插入装置 14 传递到页面选择装置 15。

接着将详细地描述在图 2 示出的步骤 206 中的自动导航产生处理。

图 5 是示出了作为页面选择装置 15、结构信息产生装置 16 和导航产生装置 18 的操作而执行的自动导航产生处理的流程图。

页面选择装置 15 首先读出并保持存储于站点结构存储装置 11 中的站点结构信息，并且在读出的站点结构信息中确定导航当前所要插入的页面（此后被称为“当前页面”）的位置（步骤 501）。然后，页面选择装置 15 确定该页面是否是顶层页面（步骤 502）。如果该页面是顶层页面，那么过程进行到步骤 505。如果该页面不是顶层页面，那么页面选择装置 15 确定该页面是否是一个导航根（步骤 503）。如果该页面是导航根，那么过程进行到步骤 505。如果该页面不是导航根，那么页面选择装置 15 获取关于该页面的父页面的信息（步骤 504），并且过程返回到步骤 502。页面选择装置 15 执行同样的处理直到达到导航根或者顶层页面。

当到达导航根或者顶层页面时，页面选择装置 15 设置该页面在导航中的等级为 1（步骤 505）。

随后，页面选择装置 15 参考从导航标记插入装置 14 中传递来的页面，并且从该页面中所写的导航标记里获取页面选择信息（步骤 506）。页面选择装置 15 基于其等级已在步骤 505 中被设置为该导航中的 1 的页面的信息、该页面的页面选择信息和预先读出的站点结构信息来选择将被包括在导航中的页面（步骤 507）。例如，如果页面选择信息指示了页面的等级

范围,页面选择装置 15 就选择相应于从该基点即导航根算起的该等级范围的页面。

于是由页面选择装置 15 实现了处理,并且进行到结构信息产生装置 16 的控制的转换。在转换的时候,用于确定所选页面的识别信息、导航标记所插入的页面和站点结构信息被从页面选择装置 15 传递到结构信息产生装置 16。

结构信息产生装置 16 从由页面选择装置 15 中传递来的站点结构信息中提取与由页面选择装置 15 传递来的识别信息相对应的页面的部分,并且产生导航结构信息(步骤 508)。该结构信息包括所选页面的标签和路径,以及所选页面和其它页面之间的关系。

在完成了这些处理之后,进行到导航产生装置 18 的控制的转换。在转换的时候,结构信息和导航标记所插入的页面被从结构信息产生装置 16 传递到导航产生装置 18。

最后,导航产生装置 18 基于从结构信息产生装置 16 传递来的结构信息和存储在模板存储装置 17 中的模板来产生导航(509)。导航产生装置 18 在从结构信息产生装置 16 传递来的页面中插入导航的描述,并且将其写回到页面存储装置 13 中。

于是,实现了自动导航产生处理。

将参考其具体的例子来详细地描述该实施例的操作。

在描述该例子时,假设已经建立了例如在图 6 中示出的站点结构(步骤 201)。相应于该结构的站点结构信息是以 XML 的形式描述的,例如,如图 7 中所示。即,例如图 7 中所示的站点结构信息被存储在站点结构存储装置 11 中。

将具体地描述由导航根设置装置 12 执行的、相关于该站点结构信息来设置导航根的处理。

如图 8 中所示,在“newpage2”被设置为导航根的情形下,导航根设置装置 12 确定“newpage2”为将被设置为导航根的页面(步骤 301)。于是,如图 9 中所示,导航根设置装置 12 向站点结构信息中的“newpage2”

的描述里添加指示“newpage2”为导航根的描述。更明确地，导航根设置装置 12 在与“newpage2”对应的<page>标记中写入了“navroot = "true"”的描述。如图 8 中所示，导航根设置装置 12 还在站点编辑器上所显示的站点结构里的一个表示“newpage2”的对象中显示了一个指示“newpage2”被设置为导航根的图标（步骤 303）。

另外，如图 10 中所示，在“newpage3”也被设置为导航根的情形下，导航根设置装置 12 确定“newpage3”为将被设置为导航根的页面（步骤 301）。于是，如图 11 中所示，导航根设置装置 12 向站点结构信息中的“newpage3”的描述里添加指示“newpage3”为导航根的描述。更明确地，导航根设置装置 12 在与“newpage3”对应的<page>标记中写入了“navroot = "true"”的描述。如图 10 中所示，导航根设置装置 12 还在站点编辑器上显示的站点结构里的一个表示“newpage3”的对象中显示了一个指示“newpage3”被设置为导航根的图标（步骤 303）。

接着将具体地描述由导航标记插入装置 14 执行的、在页面中插入导航标记的处理。将参照导航标记被插入到“newpage2”中的情况进行描述。

读取对应于“newpage2”的 HTML 文件“newpage2.html”。在编辑器中产生例如图 12 中所示的屏幕显示。参考图 12，在“newpage2.html”的编辑区域没有任何描述。在某些文本描述之后可以执行随后的操作。

例如，参考图 12，从菜单条的“Insert”输入插入导航的指令以将控制移交给导航标记插入装置 14。导航标记插入装置 14 于是显示例如图 13 中所示的导航标记插入对话框。

在图 13 中所示的导航标记插入对话框中，可以通过“Site Map Type”部分来指定使用样本或者用户定义模板作为导航模板。当用户指定使用如图 13 中所述的用户定义模板时，在“Thumbnail”部分显示可选的模板。在图 13 中示出的对话框中，“sitemap_vertical.html”作为可选的模板被预览。尽管在该例子中只示出了一个模板，但是多个可选模板可能存在并且可以被预览。

当用户在所显示的模板中点击他或者她希望使用的模板时，所选模板

的位置和文件名被显示在“Specification File”部分。在该例子中，“sitemap_vertical.html”被选为模板，并且，作为该选择的结果，“/WebContent”作为位置被显示以及“sitemap_vertical.html”作为文件名被显示。

在导航标记插入对话框中，可以指定在该页面的导航中的等级的范围，作为用于选择将被包括在该导航中的页面的信息。在该例子中，指定了从“Start level”部分中所指定的“1”到“End level”部分中所指定的“3”的等级。尽管在该例子中，指定了等级的范围，作为用于选择将被包括在导航中的页面的信息，但是其它的各种指定方法也是可以设想的。另外的指定方法将在下面进行的改变导航设置的描述中被详细地描述。

当进行了上面描述的设置后点击“OK”按钮时，导航标记插入装置 14 接受关于所选模板的信息（步骤 401），并接受用于选择页面的信息（步骤 402）。如图 14 所示，导航标记插入装置 14 然后向“newpage2.html”写入导航标记（步骤 403）。更明确地，导航标记包括指示导航标记的开始“<!--siteedit:navbar...-->”、指示包括在导航中的页面等级的“targetlevel = "1-3"”、指示用作导航模板的 HTML 文件名称的“spec="sitemap_vertical.html"”以及指示导航标记的结束的“<!--/siteedit:navbar-->”。

此时，在编辑器上产生例如图 15 所示的屏幕显示。即，导航标记插入装置 14 在显示“newpage2.html”的显示区域中显示了一个指示导航标记已经被插入的图标（步骤 404）。在该图标中，左半边指示导航标记的开始，而右半边指示导航标记的结束。然而，该图表仅是在 HTML 编辑器上显示，当在浏览器上打开该 HTML 文件时并不显示该图标。

接着将具体地描述自动导航建立。

页面选择装置 15 首先通过参考存储在站点结构存储装置 11 中的站点结构信息来确定当前页面的位置（步骤 501）。在该例子中，由于导航将被插入到“newpage2”中，所以页面选择装置 15 确定“newpage2”的位置。页面选择装置 15 然后确定“newpage2”是否是顶层页面（步骤 502）。

由于在该例子中“newpage2”不是顶层页面，因而页面选择装置 12 确定“newpage2”是否是导航根（步骤 503）。由于在该例子中“newpage2”是导航根，因此“newpage2”在导航中的等级被设置为 1（步骤 505）。此时，“newpage2”的子页面“newpage4”和“newpage5”的等级被设置为导航中的 2。

随后，页面选择装置 15 从由导航标记插入装置 14 传递过来页面里所写的导航标记中获取页面选择信息（步骤 506）。更明确地，页面选择装置 15 获取了“targetlevel = “1-3””的描述。参考该描述，页面选择装置 15 决定选择导航中具有 1 到 3 等级的页面。在此例子中，由于“newpage4”和“newpage5”在导航中的等级是 2，因此这些页面被选为将要被包括在导航中的页面。不存在具有导航中的等级 3 的页面。因此，将选择的页面是“newpage2”、“newpage4”和“newpage5”。

此后，页面选择装置 15 将所选页面为“newpage2”、“newpage4”和“newpage5”的信息、插入导航标记后的页面和站点结构信息传递给结构信息产生装置 16。

随后，结构信息产生装置 16 产生导航结构信息（步骤 508）。

更明确地，结构信息产生装置 16 从由页面选择装置 15 传送来的站点结构信息中提取由“newpage2”、“newpage4”和“newpage5”构成的局部树。例如页面的标签和路径的信息也包括在构成局部树的“newpage2”、“newpage4”和“newpage5”中。因此，这组信息将被称为“项”。

由于在此例子中，建立包括到如图 13 中“Thumbnail”部分所示的列表形式的页面的链接的导航，所以结构信息产生装置 16 根据预定规则将局部树转换为列表结构的结构信息。例如，结构信息产生装置 16 建立包括“newpage2”、“newpage4”和“newpage5”的该叙述顺序的列表结构。

例如指示相应页面在导航中的等级的“level”和指示相应页面是当前页面的“self”或者指示相应页面是当前页面的直系祖先的“ancestor”的属性被加入到各个项中。

更明确地, 例如“level = 0”、“self = true”和“ancestor = false”的属性被加入到“newpage2”中; 例如“level = 1”、“self = false”和“ancestor = false”的属性被加入到“newpage3”中; 以及例如“level = 1”、“self = false”和“ancestor = false”的属性被加入到“newpage4”中。在用户接口上根的等级被定义为等级 1, 而在计算机中根的等级被定义为“level = 0”。

当结构信息产生装置 16 产生列表结构的结构信息时, 它将该结构信息传递给导航产生装置 18。

导航产生装置 18 基于由结构信息产生装置 16 传递来的结构信息和存储于模板存储装置 17 中的模板来产生导航。即, 导航产生装置 18 基于由结构信息产生装置 16 传递来的结构信息执行模板中所描述的控制结构, 并且获得导航作为其输出。

图 16 示出了模板的一个例子。在该模板中, 将被输出为导航的部分被定义为一表(<table> ~ </table>)。针对包括在结构信息中的每项来执行表的内容。在该模板中, 根据被加到诸项中的属性的结合的条件来进行情况(case)划分。在该例子中, 根据“level”和“self”的结合划分为四种情况。根据各个项所满足的条件而定义的部分被输出, 作为导航。

将更加详细地描述使用该模板的处理。

导航产生装置 18 首先输出<table>标记 1601 和<tbody>标记 1602。随后, 导航产生装置 18 通过解释标记 1603 以及在变量“item”中设置各个项来处理标记 1603 - 1606 部分。

在该部分中, 定义了根据“level”和“self”的结合的条件。即, 存在“self = true”和“level = 0”的情况、“self = false”和“level = 0”的情况、这些情况之外的“self = true”时的情况以及其它一个情况。针对各个项来执行这部分的处理, 并提取根据每个项所满足的条件而定义的<tr>元素。

更明确地, 处理的执行如下面所述。

导航产生装置 18 首先通过在变量“item”中设置结构信息的第一项“newpage2”来处理标记 1603 - 1606 部分。由于在“newpage2”中“self

= true” 且 “level = 0”，所以导航产生装置 18 输出 <tr> 元素 1604。此时，用 “newpage2” 的标签来替换 “\${item.label}”，以及用 “newpage2” 的路径来替换 “\${item.href}”。

随后，导航产生装置 18 通过在变量 “item” 中设置结构信息的第二项 “newpage4” 来处理标记 1603 - 1606 部分。由于在 “newpage4” 中 “self = true” 和 “level = 0” 都不满足，因此导航产生装置 18 输出 <tr> 元素 1605。此时，用 “newpage4” 的标签来替换 “\${item.label}”，以及用 “newpage4” 的路径来替换 “\${item.href}”。

此外，导航产生装置 18 通过在变量 “item” 中设置结构信息的第三项 “newpage5” 来处理标记 1603 - 1606 部分。由于在 “newpage5” 中 “self = true” 和 “level = 0” 都不满足，因此导航产生装置 18 输出 <tr> 元素 1605。此时，用 “newpage5” 的标签来替换 “\${item.label}”，以及用 “newpage5” 的路径来替换 “\${item.href}”。

当完成了上述处理时，结构信息中就没有其它的项了，因此导航产生装置 18 输出 </tbody> 标记 1607 和 </table> 标记 1608。

在上述的例子中，将 “self” 和 “level” 的结合作为条件来使用，也可以将包括一些其它属性例如 “ancestor” 的结合作为条件使用，以根据这些属性来提供更精细的导航。

图 17 示出了由上述处理自动产生导航之后的 HTML 源文件。由上述处理输出的导航（<table> ... </table>）被插入到导航标记开始标记 “<!--siteedit:navbar...-->” 和结束标记 “<!--/siteedit:navbar-->” 之间。

图 18 示出了此时编辑器上屏幕显示的例子。在该屏幕显示中，包括到 “newpage2”、“newpage4” 和 “newpage5” 的链接的导航显示图像被输出到指示导航标记开始标记的图标和指示结束标记的图标之间。

同样在 “newpage3” 中插入导航的情况下，进行与上述处理相同的处理来产生包括到 “newpage3”、“newpage6” 和 “newpage7” 的链接的导航。

除了上面描述的模板之外还可以使用例如图 19 所示的导航插入标记

对话框的预览显示中的模板。

图 13 所示的模板是用于通过使用列表结构的结构信息来建立导航的，而图 19 所示的模板是用于通过使用树结构的结构信息来建立导航的。

将要就在“newpage”上建立导航的情形来具体描述通过使用图 19 所示的模板自动建立导航时的操作。

页面选择装置 15 首先通过参考存储于站点结构存储装置 11 中的站点结构信息来确定当前页面的位置（步骤 501）。由于在该例子中，将把导航插入到“newpage”里，所以页面选择装置 15 确定“newpage”的位置。随后，页面选择装置 15 确定“newpage”是否是顶层页面（步骤 502）。由于在该例子中，“newpage”是顶层页面，所以“newpage”在导航中的等级被设置为 1（步骤 505）。此时，“newpage”的子页面“newpage2”和“newpage3”的在导航中的等级被设置为 2，“newpage2”的子页面“newpage4”和“newpage5”的等级以及“newpage3”的子页面“newpage6”和“newpage7”的等级被设置为导航中的 3。

随后，页面选择装置 15 从由导航标记插入装置 14 传递来的页面中所写的导航标记中获取页面选择信息（步骤 506）。更明确地，页面选择装置 15 获取了“targetlevel = “1-3””的描述。通过参考该描述，页面选择装置 15 决定选择在导航中具有 1 至 3 等级的页面。在该例子中，由于“newpage2”和“newpage3”在导航中的等级是 2，因此这些页面被选择为导航所要包括的页面。同样地，由于“newpage4”、“newpage5”、“newpage6”和“newpage7”在导航中的等级是 3，因此这些页面也被选择为导航所要包括的页面。从而，将被选择的页面是“newpage”、“newpage2”、“newpage3”、“newpage4”、“newpage5”、“newpage6”和“newpage7”。

其后，页面选择装置 15 将所选页面为“newpage”、“newpage2”、“newpage3”、“newpage4”、“newpage5”、“newpage6”和“newpage7”的信息、插入导航标记后的页面和站点结构信息传递给结构信息产生装置 16。

随后结构信息产生装置 16 产生导航结构信息（步骤 508）。

更明确地，结构信息产生装置 16 根据由页面选择装置 15 传递来的站点结构信息而导出一个由“newpage”、“newpage2”、“newpage3”、“newpage4”、“newpage5”、“newpage6”和“newpage7”构成的局部树。

由于在该例子中，建立了包括到如图 19 中“Thumbnail”部分所示的树结构形式的页面的链接的导航，因此，结构信息产生装置 16 直接设置该局部树为树结构的结构信息。

如使用图 13 中所示模板的情况一样，例如指示导航中相应页面的等级的“level”和指示相应页面是当前页面的“self”或者指示相应页面是当前页面的直系祖先的“ancestor”的属性被加入到各个项中。

当结构信息产生装置 16 产生了树结构的结构信息时，它将该结构信息传递给导航产生装置 18。

导航产生装置 18 基于由结构信息产生装置 16 传递来的结构信息和存储于模板存储装置 17 中的模板来产生导航。即，导航产生装置 18 基于由结构信息产生装置 16 传递来的结构信息来执行模板中所描述的控制结构，并且获得导航作为其输出。

图 20 示出了该模板的一个例子。该模板使得能够以如此的方式来建立具有高自由度的导航，即，将树结构中存在/不存在子节点以及子节点的数目设置到用于控制结构的情况划分的条件和将要输出的标记属性的值中。

将更加详细地描述使用该模板的处理。

导航产生装置 18 首先通过解释标记 2001 及在变量“top”中设置结构信息的第一项“newpage”来处理标记 2001 - 2021 部分。

即，导航产生装置 18 首先输出<table>标记 2002、<tbody>标记 2003、<tr>元素 2004 和<tr>标记 2005。此时，“\${top.childcount}”被作为“newpage”的子的节点（此后称为“子节点”）的数目所替换，以及“\${top.item.href}”被“newpage”的路径所替换。

随后，导航产生装置 18 通过解释标记 2006 及在变量“node”中设置

结构信息中“newpage”的第一子节点“newpage2”来处理标记 2006 - 2017 部分。

即，导航产生装置 18 首先输出<td>标记 2007 和<a>元素 2008。此时，“\${node.item.label}”被“newpage2”的标签所替换，以及“\${node.item.href}”被“newpage2”的路径所替换。

随后，由于存在“newpage2”的子节点，所以导航产生装置 18 解释标记 2009 并且处理标记 2009 - 2015 部分。

即，导航产生装置 18 首先输出标记 2010。

随后，导航产生装置 18 解释标记 2011，在变量“node2”中设置结构信息中“newpage2”的第一子节点“newpage4”，并输出元素 2012。此时，“\${node2.item.label}”被“newpage4”的标签所替换，以及“\${node2.item.href}”被“newpage4”的路径所替换。同样地，导航产生装置 18 在变量“node2”中设置结构信息中“newpage2”的第二子节点“newpage5”，并输出元素 2012。此时，“\${node2.item.label}”被“newpage5”的标签所替换，以及“\${node2.item.href}”被“newpage5”的路径所替换。

当完成该处理时，就没有“newpage2”的其它子节点了，因此导航产生装置 18 输出标记 2014 和</td>标记 2016。

随后，导航产生装置 18 以同样的方式在变量“node”中设置“newpage”的第二子节点“newpage3”并处理标记 2006 - 2017 部分。

当完成该处理时，就没有“newpage”的其它子节点了，因此导航产生装置 18 输出</tr>标记 2018、</tbody>标记 2019 和</table>标记 2020。

图 21 示出了通过上述处理自动产生导航后的 HTML 源文件。由上述处理输出的导航（<table> ... </table>）被插入到导航标记开始标记“<!--siteedit:navbar...-->”和结束标记“<!--/siteedit:navbar-->”之间。

图 22 示出了此时编辑器上屏幕显示的例子。在该屏幕显示中，导航显示图像被输出到指示导航标记开始标记的图标和指示结束标记的图标之间。

由于结构信息产生装置 16 可以被安排用于在仅仅预备了使用列表结构的结构信息的模板的系统中只产生列表结构的结构信息，并且由于结构信息产生装置 16 可以被安排用于在仅仅预备了使用树结构的结构信息的模板的系统中只产生树结构的结构信息，所以在上述的描述中，没有提及关于确定结构信息产生装置 16 是产生列表结构的结构信息还是产生树结构的结构信息的方法。

然而，其中具有使用列表结构的结构信息的模板和使用树结构的结构信息的模板两者的系统是可以设想的。在这种情况下，可以在模板中定义关于使用列表结构的结构信息还是使用树结构的结构信息的属性，并且结构信息产生装置 16 可以基于在所选模板中定义的该属性来确定产生两种结构信息中的一种。或者，结构信息产生装置 16 可以不管所选模板的类型而产生列表结构的结构信息和树结构的结构信息两者。

在该实施例中，还提供了通过使用如图 13 或 19 所示的导航标记插入对话框来改变建立的导航设置的功能。导航设置可以在如图 5 所示的自动导航建立之前或者之后进行改变。然而，在自动导航建立之后改变设置的情况下，重新进行自动导航建立以使得设置的改变反映在导航中。

将具体描述用于导航设置改变的处理。

图 23 示出了用于改变用作导航模板的文件名和用于添加导航所包括的页面显示次序的设置的对话框。后者的设置是通过选中“Current page is first sibling”以使得本页面显示在导航中的兄弟页面里的最高位置。当进行了该设置时，表示该设置的描述（“topsibling = \"true\"”）被写入导航标记。

如果有这样的描述，结构信息产生装置 16 就进行如下的处理，即在步骤 508 中产生结构信息的时候，本页面被显示在众兄弟页面中的最高位置。

图 24 示出了用于添加用于根据与本页面的相对关系来指定导航中的页面的设置的对话框。

更明确地，可以通过选中“Top page”来实现用于在导航中包括站点结构的顶层页面的设置。可以通过选中“Parent page”来实现用于在导航

中包括本页面的父页面的设置。可以通过选中“Ancestor pages”来实现用于在导航中包括祖先页面（父页面、祖父页面和曾祖父页面等等）的设置。可以通过选中“Current page”来实现用于在导航中包括本页面的设置。可以通过选中“Children pages”来实现用于在导航中包括本页面的子页面的设置。可以通过选中“Sibling pages”来实现用于在导航中包括本页面的兄弟页面的设置。可以通过选中“Children of top page”来实现用于在导航中包括顶层页面的子页面的设置。可以通过选中“First child page”来实现用于在导航中包括本页面的长子（第一子）页面的设置。可以通过选中“Previous page”来实现用于在导航中包括本页面的直接的在前页面（紧邻的兄页面）的设置。可以通过选中“Next page”来实现用于在导航中包括本页面的后一页面（紧邻的弟页面）的设置。

如果进行了这种设置，那么就将代表该设置的描述写入到导航标记中，并且页面选择装置 15 在结合该设置所指定的页面的同时在步骤 507 中进行选择。

图 25 是用于改变导航中将包括的页面的开始等级和结束等级的设置以及用于添加用于将导航中所包括的页面限制为直系祖先和后代的设置的对话框。后者的设置通过选中“Limit links to direct children only”来实现。表示这种设置的描述被写入导航标记中，并且页面选择装置 15 在步骤 507 中页面选择的时候执行用于排除除了直系的祖先和后代之外的页面的处理。

在图 23 至 25 所示的对话框中，添加了在若干图 13 或者 19 示出的导航标记插入对话框中没有设置的设置。然而，在设置改变时进行这些设置并不是必需的。可以在通过使用图 13 或者 19 所示的导航标记插入对话框来进行导航设置时改变这些设置。

已经描述了实施例的操作。

在该实施例中，导航标记插入装置 14 从页面存储装置 13 中读出所要建立导航的页面的页面描述，并且导航标记插入装置 14、页面选择装置 15 和结构信息产生装置 16 中的每个都不将该描述写回到页面存储装置 13 中，

而是由导航产生装置 18 最后将该描述写回到页面存储装置 13 中。然而，安排也可以是这样的，即导航标记插入装置 14、页面选择装置 15 和结构信息产生装置 16 中的每个都在其处理开始时读出该描述并且在处理结束时将该描述写回到页面存储装置 13 中。

尽管在所描述的安排中，页面选择装置 15 从站点结构存储装置 11 读出站点结构信息并且直接地将该读出信息传递给结构信息产生装置 16，但是安排还可以是这样的，即结构信息产生装置 16 重新从站点结构存储装置 11 读出站点结构信息而不是接收来自页面选择装置 15 的信息。

在如上所述的实施例中，产生了包括到页面的链接和指示页面位置的信息的导航，所述页面包括在网站树结构的任一局部树中。这样，就能够产生围绕网站中的特定网页组的并且容易理解的导航了。

同样是在该实施例中，在模板中风格是根据普通局部树中的各个位置来定义的，以便为到包括在特定局部树中的页面的链接确定风格，这样使得能够使用公共的模板。

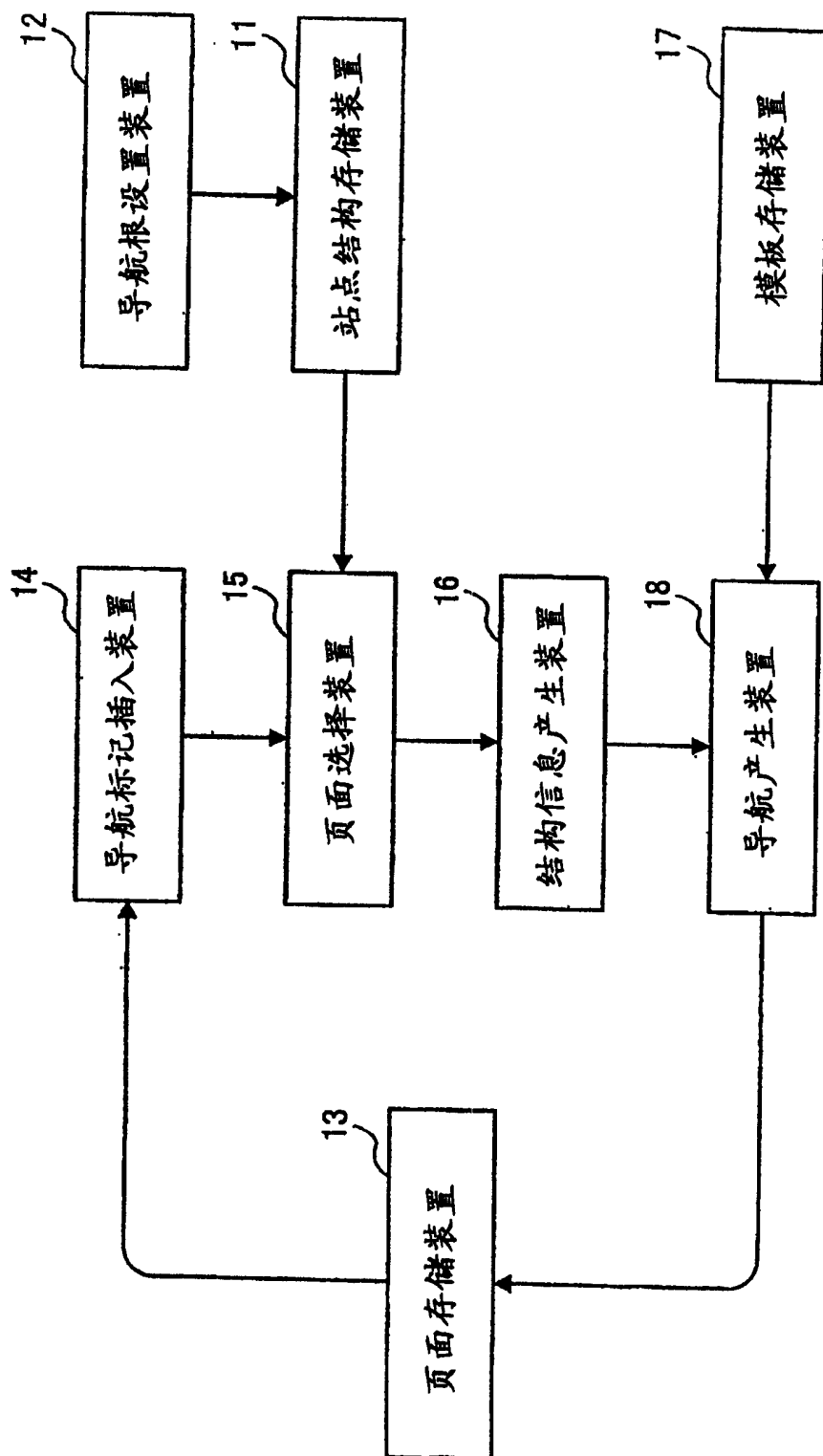


图 1

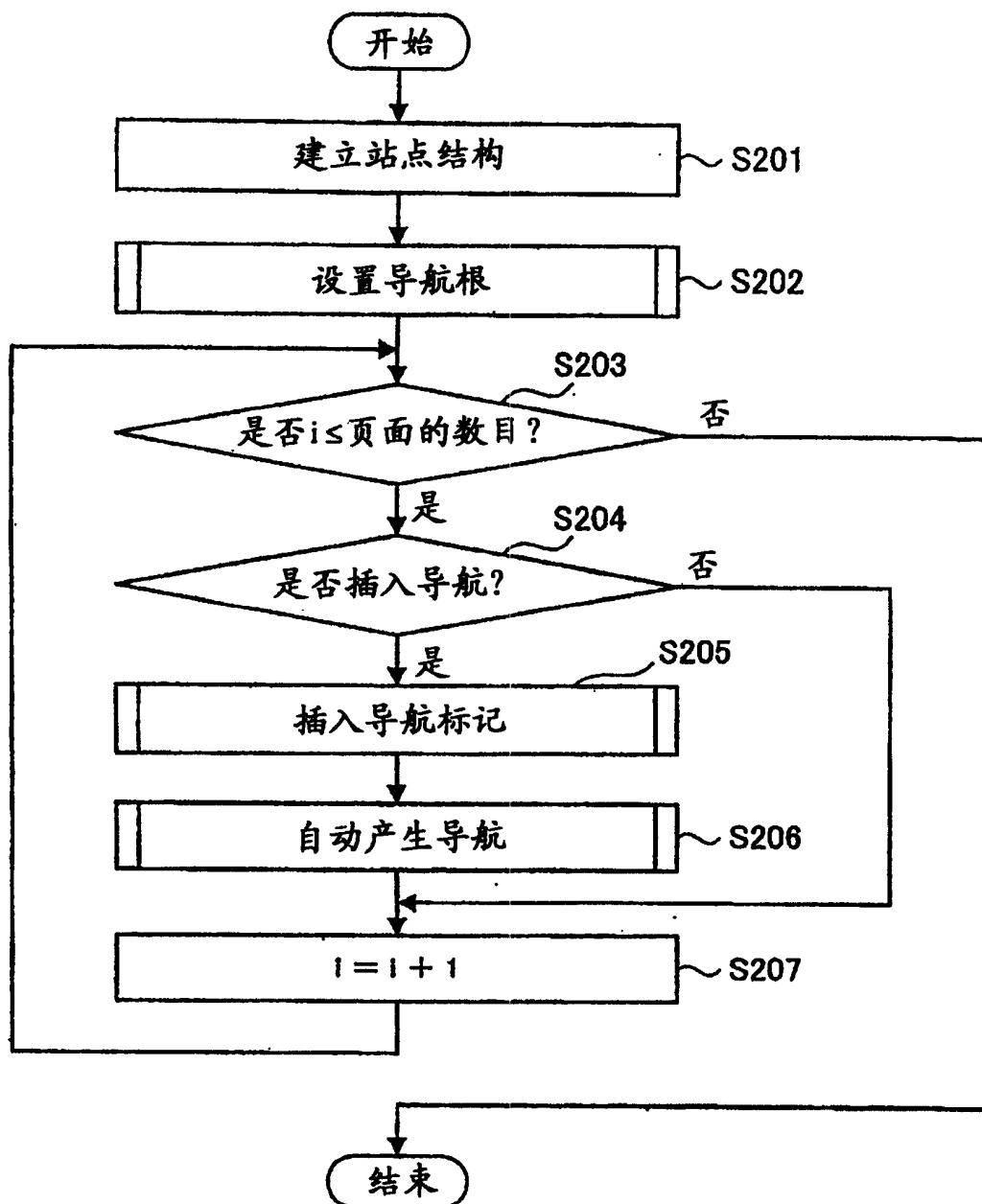


图 2

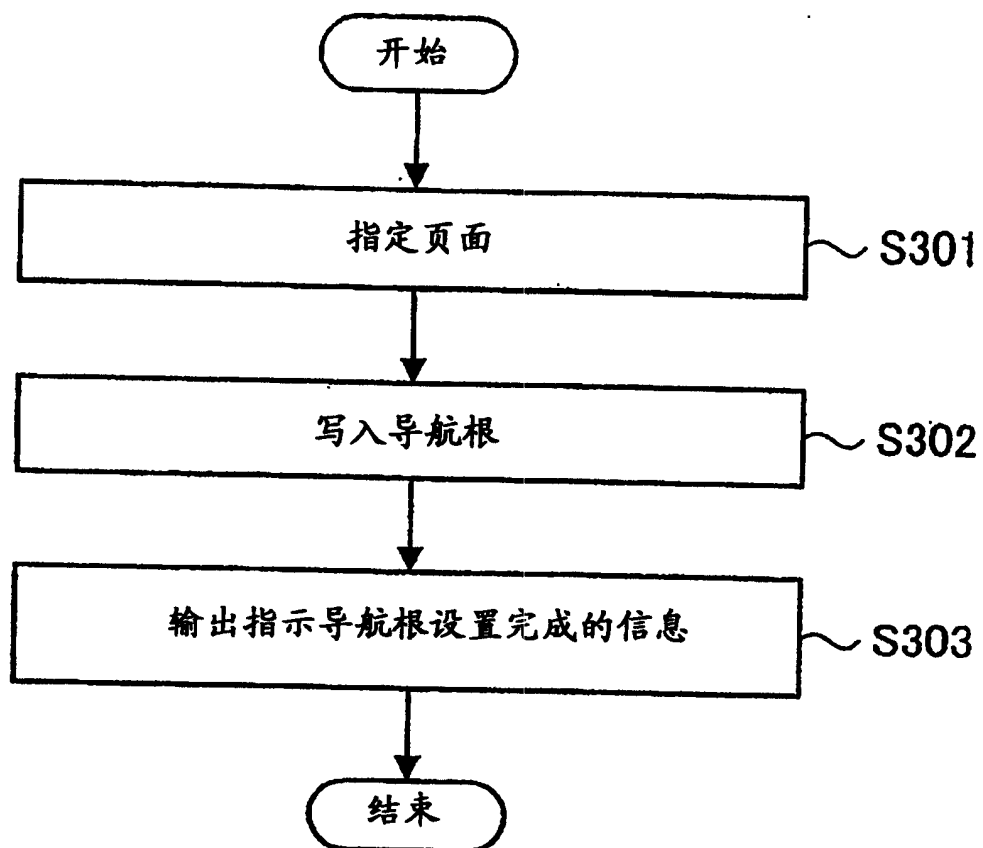


图 3

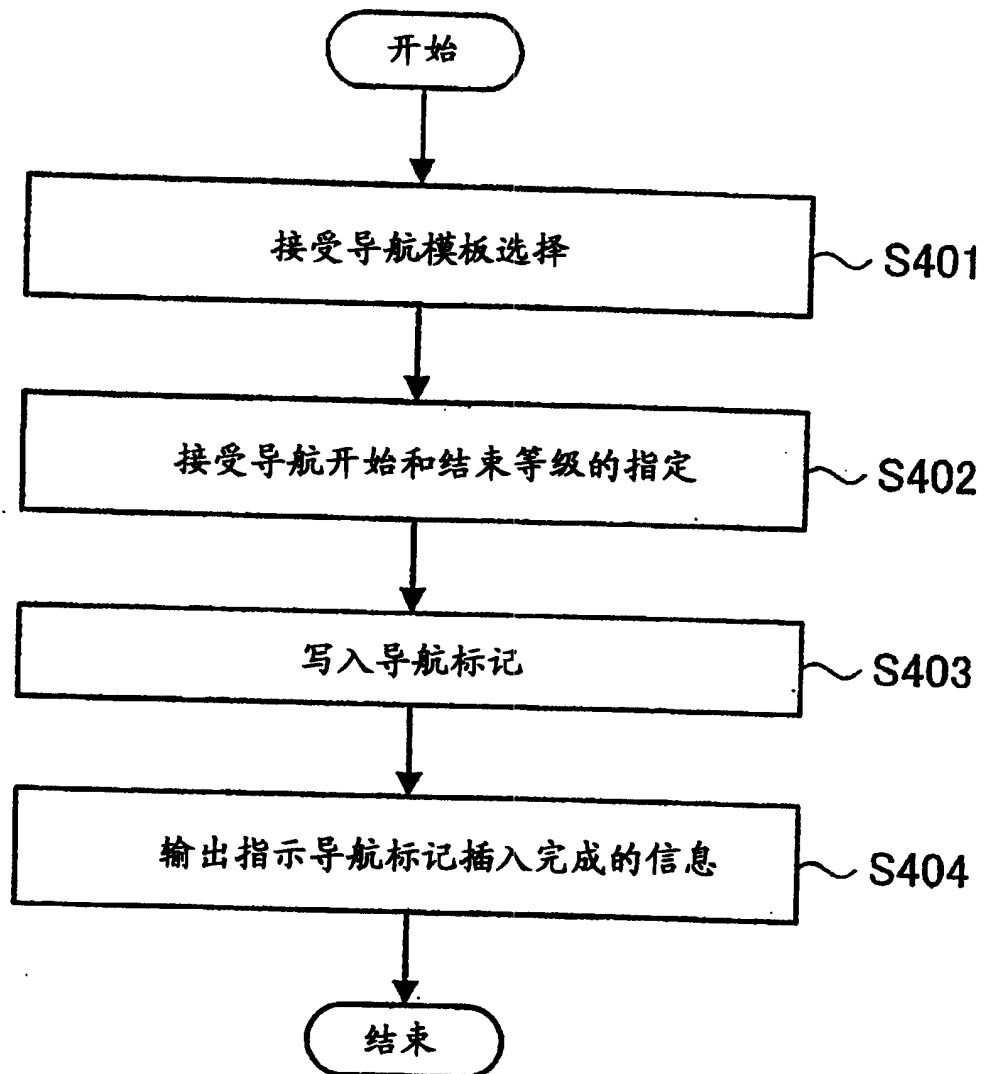


图 4

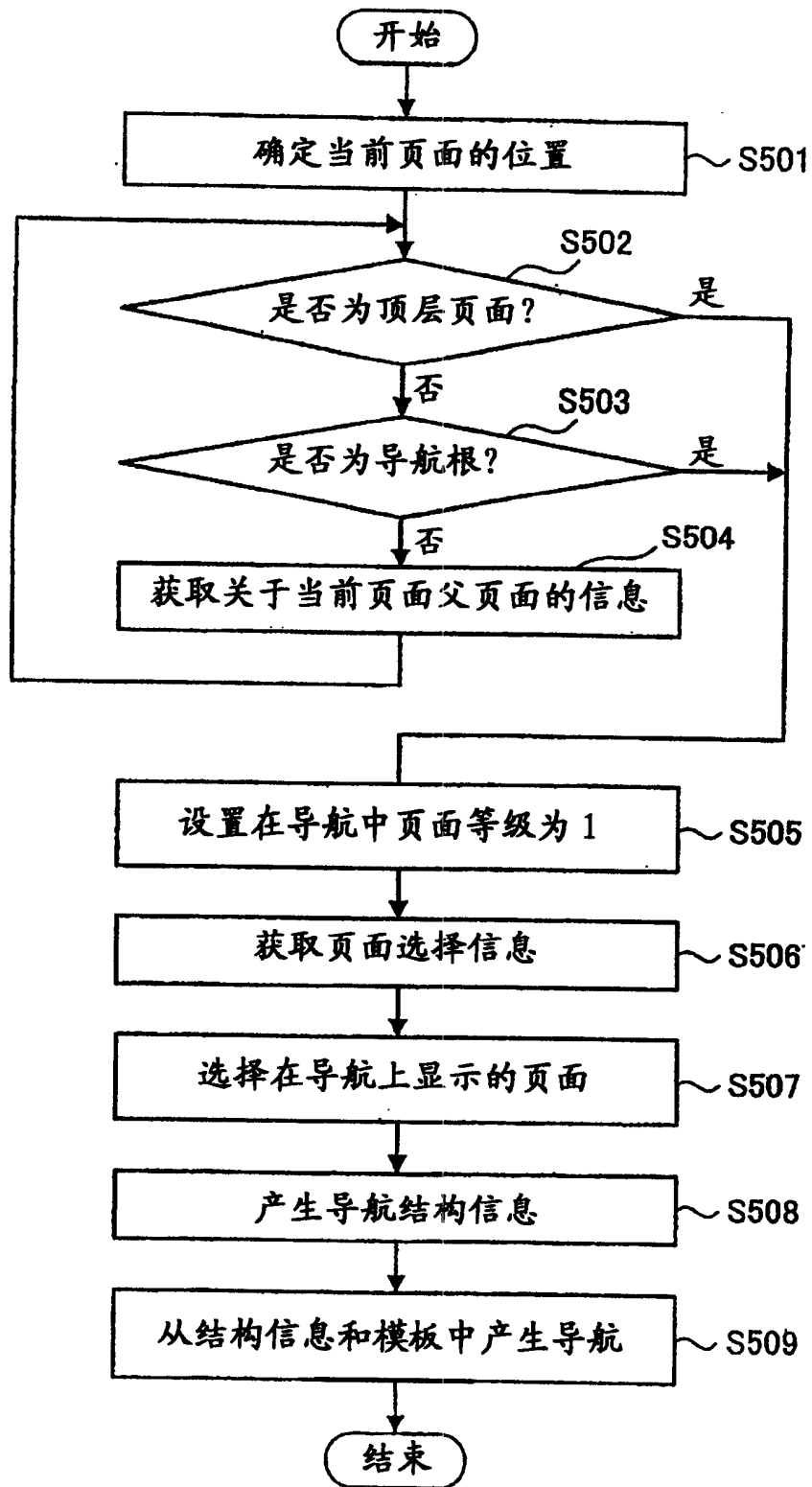


图 5

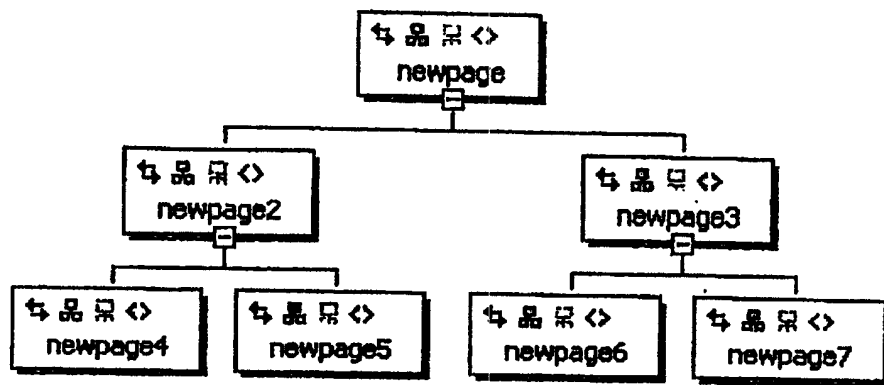


图 6

```
<? xml version="1.0" encoding="UTF -8" ?>
- < website version =" 510 ">
- < structure >
- < page src =" /WebContent/newpage.html " id =" 1 ">
  < title > newpage </ title >
- < page src =" /WebContent/newpage2.html " id =" 2 ">
  < title > newpage2 </ title >
- < page src =" /WebContent/newpage4.html " id =" 4 ">
  < title > newpage4 </ title >
  </page >
- < page src =" /WebContent/newpage5.html " id =" 5 ">
  < title > newpage5 </ title >
  </ page >
</ page >
- < page src =" /WebContent/newpage3.html " id =" 3 ">
  < title > newpage3 </ title >
- < page src =" /WebContent/newpage6.html " id =" 6 ">
  < title > newpage6 </ title >
  </ page >
- < page src =" /WebContent/newpage7.html " id =" 7 ">
  < title > newpage7 </ title >
  </ page >
</ page >
</ structure >
</ website >
```

图 7

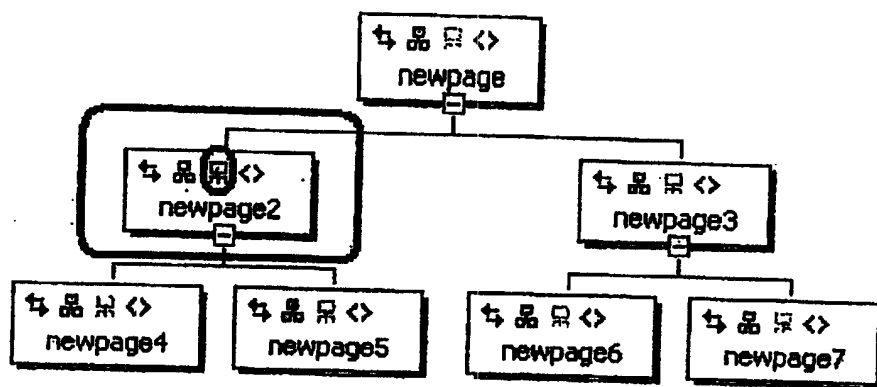


图 8

```
<? xml version="1.0" encoding="UTF-8" ?>
- < website version = " 510 " >
  - < structure >
    - < page src = "/WebContent/newpage.html" id = " 1 " >
      < title > newpage </ title >
    - < page src = "/WebContent/newpage2.html" id = " 2 " navroot = " true " >
      < title > newpage2 </ title >
    - < page src = "/WebContent/newpage4.html" id = " 4 " >
      < title > newpage4 </ title >
    </ page >
    - < page src = "/WebContent/newpage5.html" id = " 5 " >
      < title > newpage5 </ title >
    </ page >
  </ page >
  - < page src = "/WebContent/newpage3.html" id = " 3 " >
    < title > newpage3 </ title >
  - < page src = "/WebContent/newpage6.html" id = " 6 " >
    < title > newpage6 </ title >
  </ page >
  - < page src = "/WebContent/newpage7.html" id = " 7 " >
    < title > newpage7 </ title >
  </ page >
</ page >
</ structure >
</ website >
```

图 9

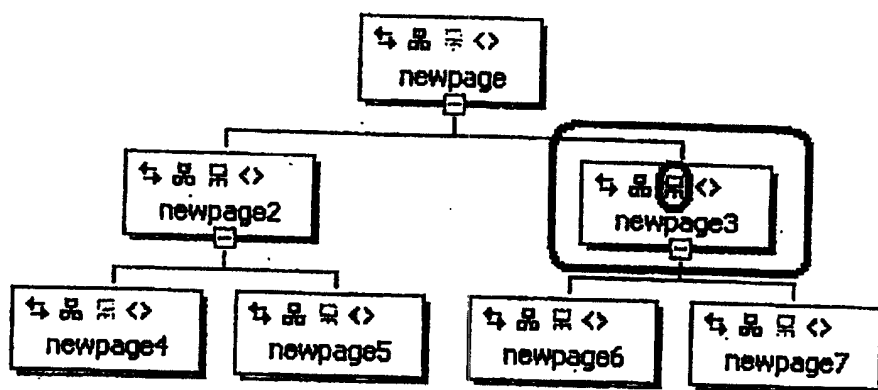


图 10

```
<? xml version="1.0" encoding="UTF -8" ?>
- < website version =" 510 ">
  - <structure>
    - <page src="/WebContent/newpage.html" id="1">
      <title> newpage </title>
    - <page src="/WebContent/newpage2.html" id="2" navroot="true">
      <title> newpage2 </title>
    - <page src="/WebContent/newpage4.html" id="4">
      <title> newpage4 </title>
    </page>
    - <page src="/WebContent/newpage5.html" id="5">
      <title> newpage5 </title>
    </page>
  </page>
  - <page src="/WebContent/newpage3.html" id="3" navroot="true">
    <title> newpage3 </title>
  - <page src="/WebContent/newpage6.html" id="6">
    <title> newpage6 </title>
  </page>
  - <page src="/WebContent/newpage7.html" id="7">
    <title> newpage7 </title>
  </page>
</page>
</structure>
</website>
```

图 11

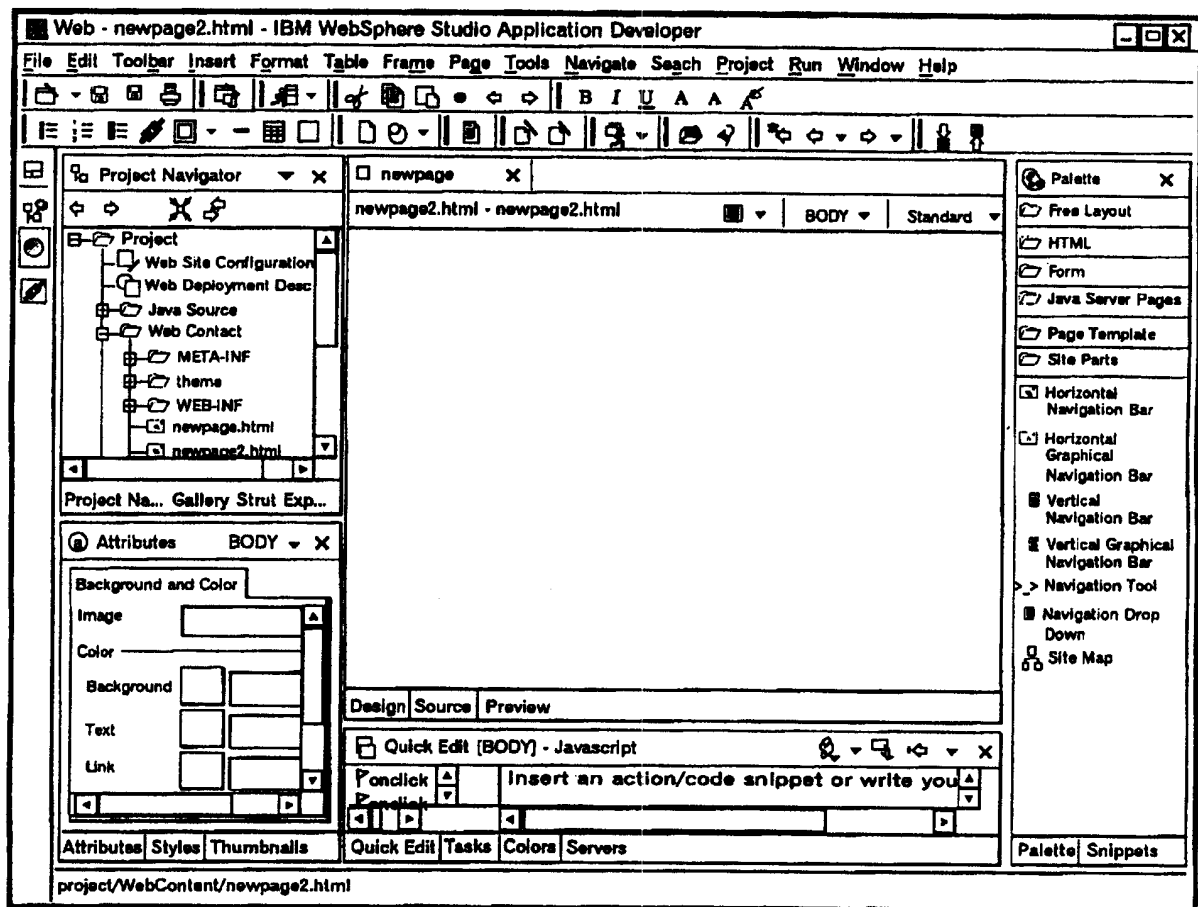
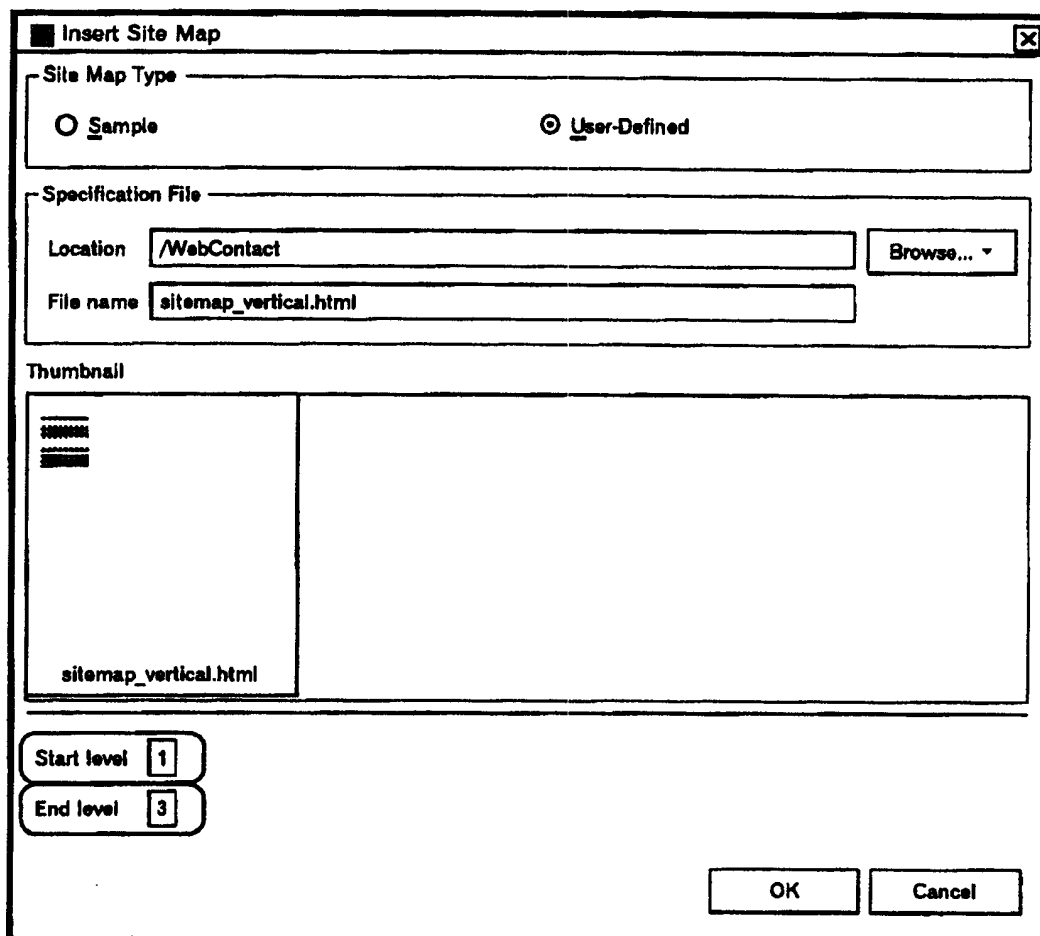


图 12



The dialog box is titled "Insert Site Map" and contains the following sections:

- Site Map Type:** Two radio buttons are present: "Sample" (unselected) and "User-Defined" (selected).
- Specification File:**
 - Location:** A text field containing "/WebContact" and a "Browse..." button.
 - File name:** A text field containing "sitemap_vertical.html".
- Thumbnail:** A large rectangular area. On the left side, there is a small thumbnail image of a site map structure, and below it, the text "sitemap_vertical.html".
- Start level:** A numeric input field with the value "1".
- End level:** A numeric input field with the value "3".
- Buttons:** "OK" and "Cancel" buttons are located at the bottom right.

图 13

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=SHIFT_JIS">
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet" type="text/css">
<TITLE>newpage2.html</TITLE>
</HEAD>
<BODY>
<!-- siteedit:navbar targetlevel="1-3" spec="sitemap_vertical.html" -->
<!-- /siteedit:navbar -->
</BODY>
</HTML>
```

图 14

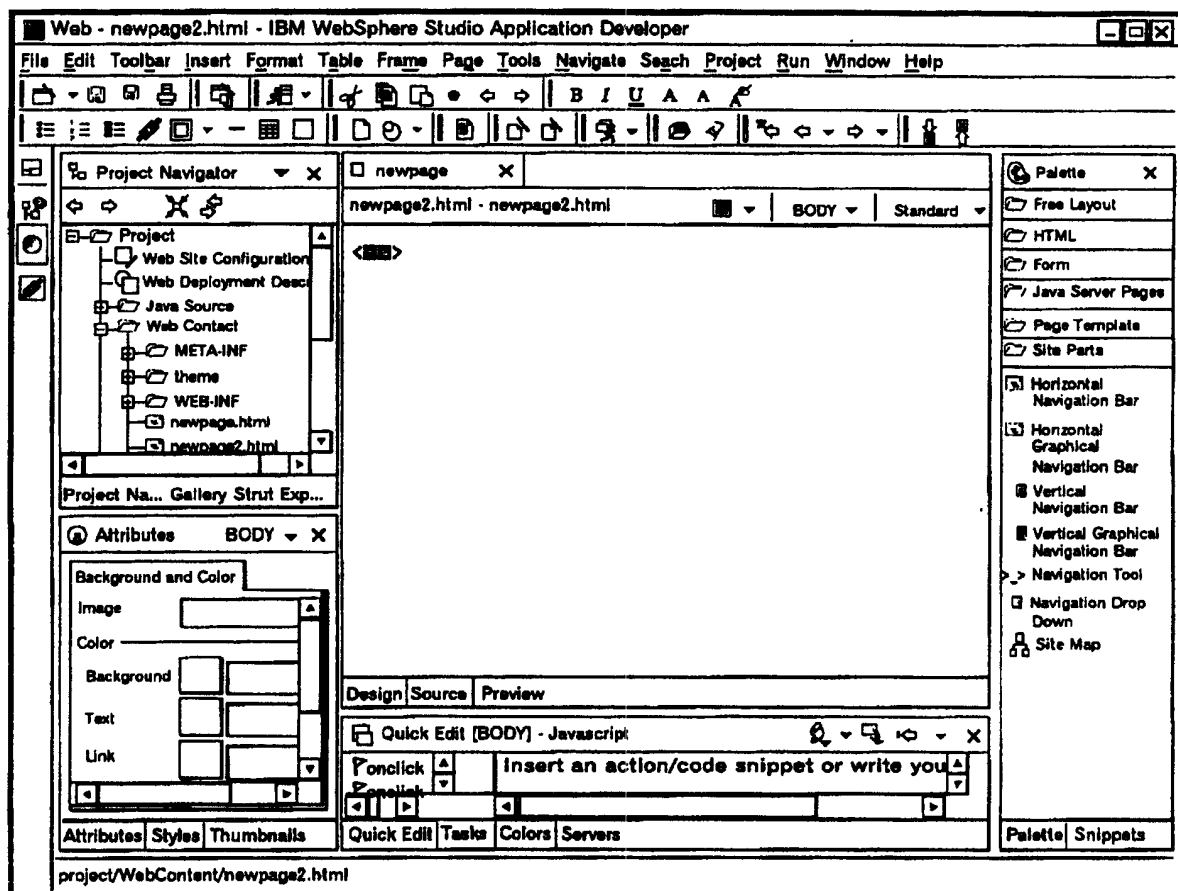


图 15

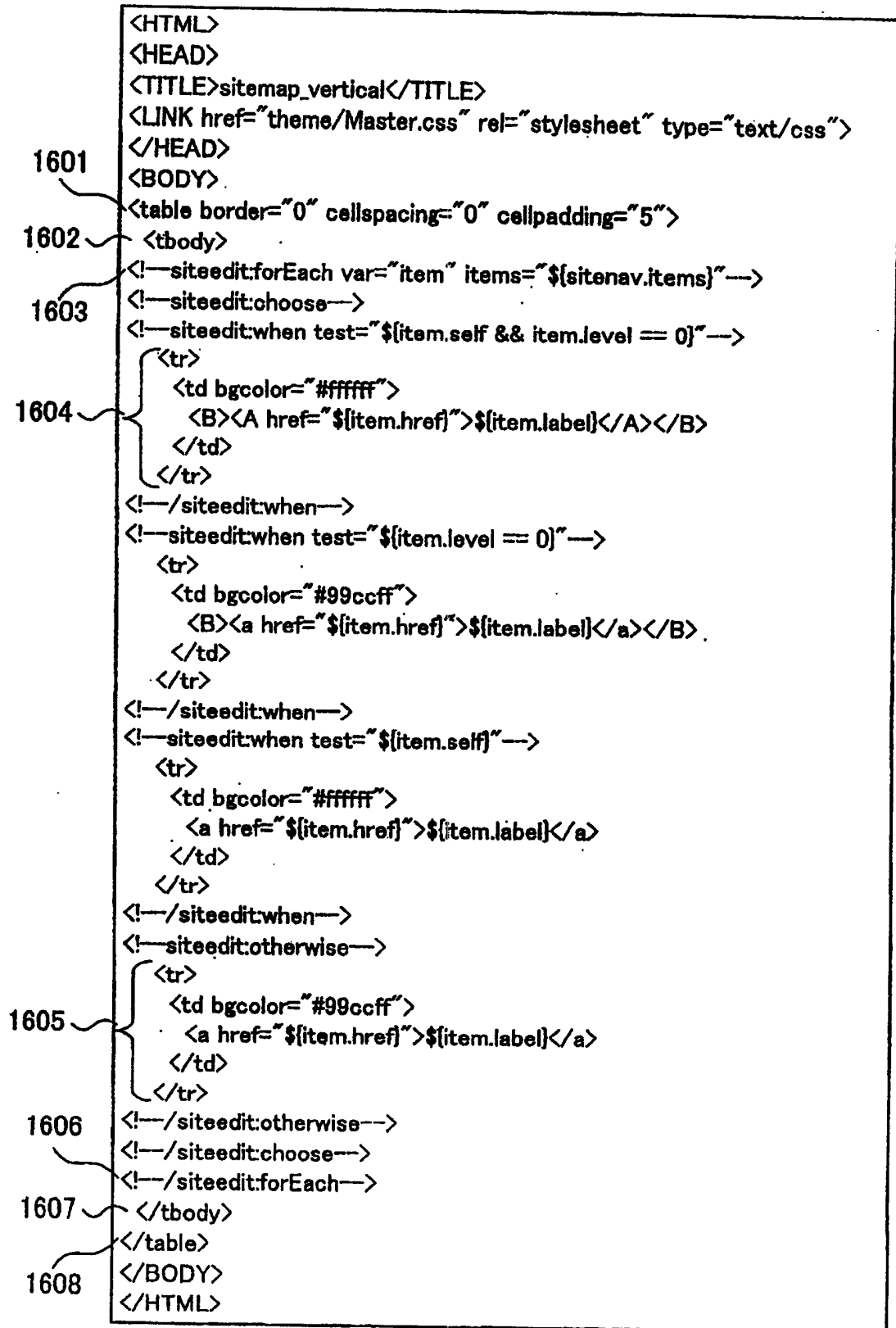


图 16

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=SHIFT_JIS">
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet" type="text/css">
<TITLE>newpage2.html</TITLE>
</HEAD>
<BODY>
<!-- siteedit:navbar targetlevel="1-3" spec="sitemap_vertical.html" -->
<table border="0" cellspacing="0" cellpadding="5">
  <tbody>
    <tr>
      <td bgcolor="#ffffff">
        <B><A href="/project/newpage2.html">newpage2</A></B>
      </td>
    </tr>
    <tr>
      <td bgcolor="#99ccff">
        <a href="/project/newpage4.html">newpage4</a>
      </td>
    </tr>
    <tr>
      <td bgcolor="#99ccff">
        <a href="/project/newpage5.html">newpage5</a>
      </td>
    </tr>
  </tbody>
</table>
<!-- /siteedit:navbar -->
</BODY>
</HTML>

```

图 17

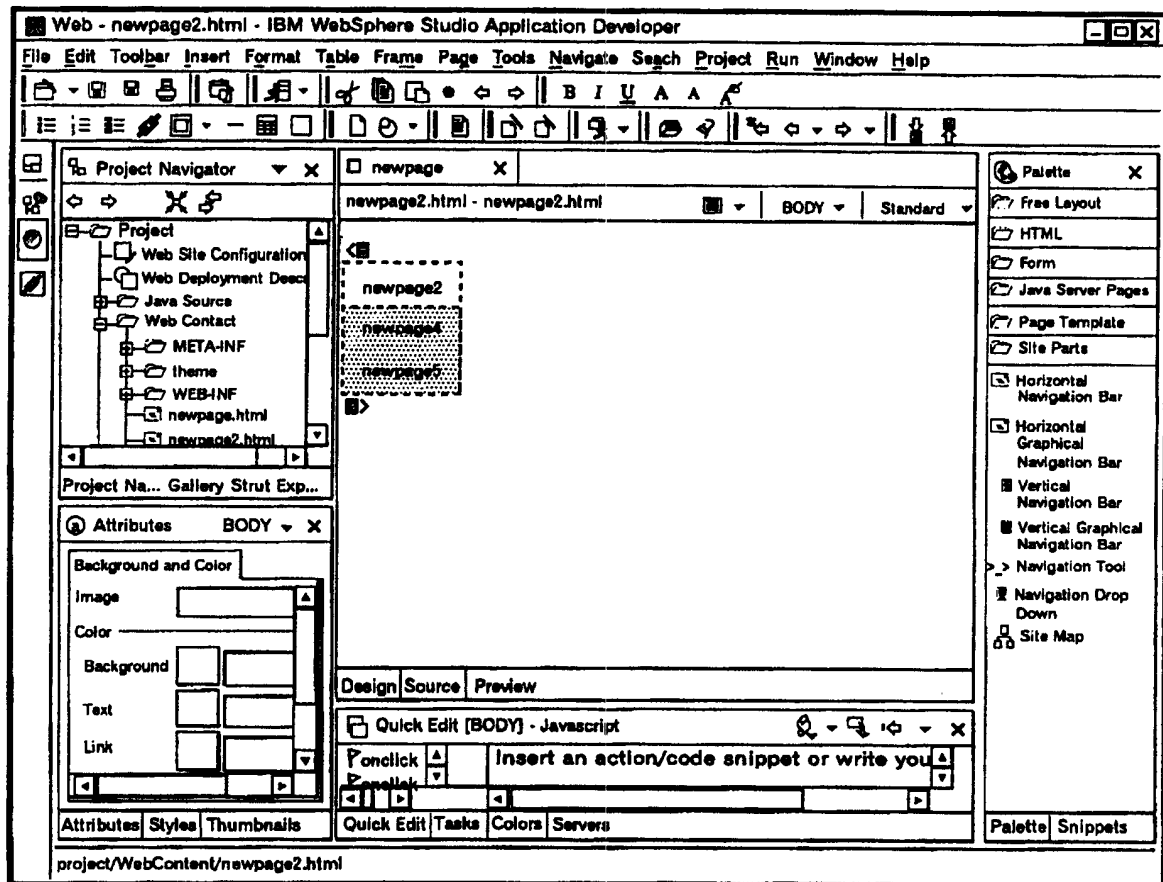


图 18

Insert Site Map

Site Map Type

☒ Sample

☐ User-Defined

Specification File

Location

Browse... ▾

File name

sitemap.html

Thumbnail

00000		
000	000	000
• 000	• 000	• 000
• 000	• 000	• 000
• 000	• 000	• 000

sitemap.html

Start level

1

End level

3

OK

Cancel

图 19

```

<HTML>
<HEAD>
<TITLE>sitemap</TITLE>
</HEAD>
<BODY>
2001 <!--siteedit:forEach var="top" items="${sitenav.tops}"-->
2002 <table border="1">
2003 <tbody>
2004 <tr>
2005 <td valign="top" align="center" colspan="${top.childcount}">
2006 <a href="${top.item.href}">${top.item.label}</a>
2007 </td>
2008 </tr>
2009 <!--siteedit:forEach var="node" items="${top.children}"-->
2010 <td valign="top">
2011 <a href="${node.item.href}">${node.item.label}</a>
2012 <!--siteedit:if test="${0 < node.childcount}"-->
2013 <ul>
2014 <!--siteedit:forEach var="node2" items="${node.children}"-->
2015 <li><a href="${node2.item.href}">${node2.item.label}</a></li>
2016 <!--/siteedit:forEach-->
2017 </ul>
2018 <!--/siteedit:if-->
2019 </td>
2020 </tr>
2021 </tbody>
</table>
</BODY>
</HTML>

```

图 20

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=SHIFT_JIS">
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet" type="text/css">
<TITLE>newpage.html</TITLE>
</HEAD>
<BODY>
<!-- siteedit:navbar targetlevel="1-3" spec="sitemap.html" -->
<table border="1">
  <tbody>
    <tr>
      <td valign="top" align="center" colspan="2">
        <a href="/project/newpage.html">newpage</a>
      </td>
    </tr>
    <tr>
      <td valign="top">
        <a href="/project/newpage2.html">newpage2</a>
        <ul>
          <li><a href="/project/newpage4.html">newpage4</a></li>
          <li><a href="/project/newpage5.html">newpage5</a></li>
        </ul>
      </td>
      <td valign="top">
        <a href="/project/newpage3.html">newpage3</a>
        <ul>
          <li><a href="/project/newpage6.html">newpage6</a></li>
          <li><a href="/project/newpage7.html">newpage7</a></li>
        </ul>
      </td>
    </tr>
  </tbody>
</table>
<!-- /siteedit:navbar -->
</BODY>
</HTML>

```

图 21

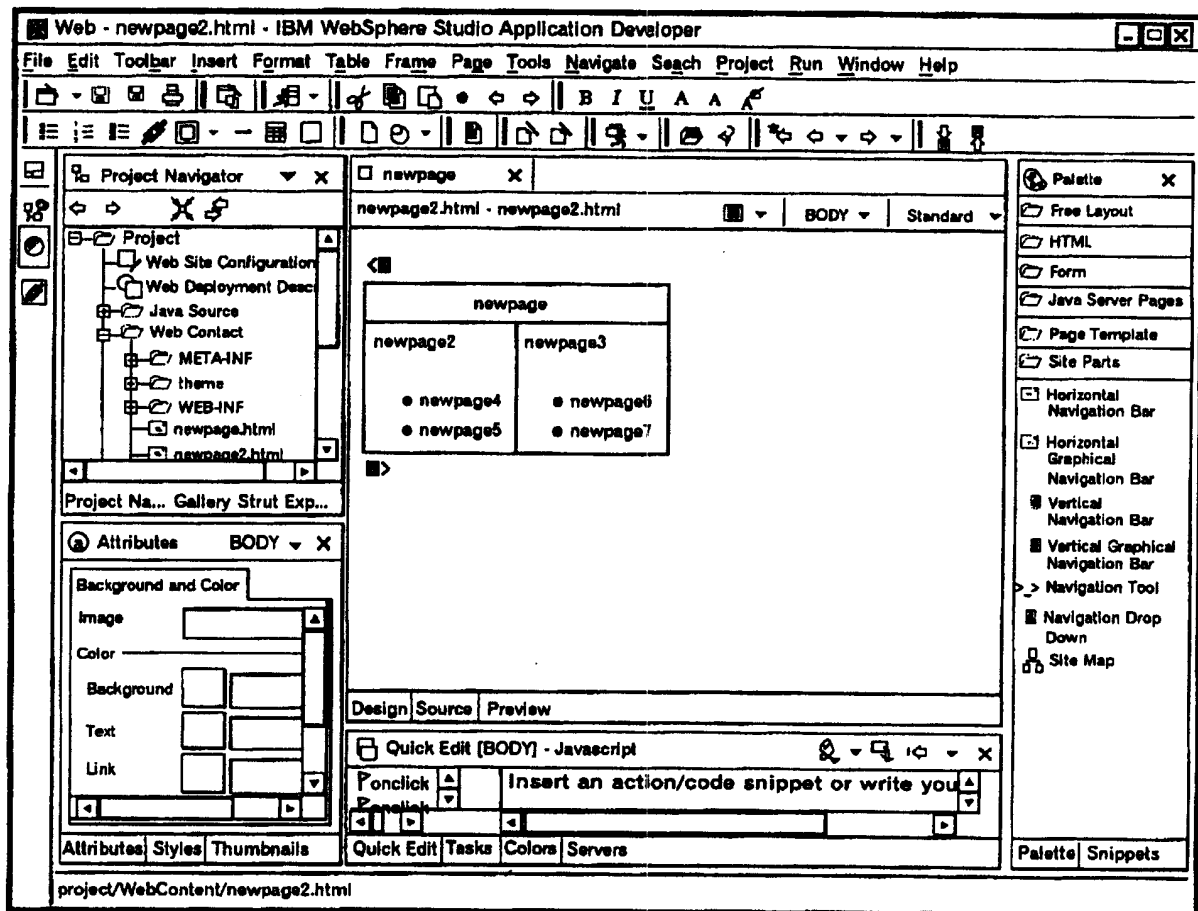


图 22

Ⓐ AttributesSITEEDIT:Navbar

SpecificationLink DestinationLink Level

Specification file: Sitemap_vertical.html

...

Edit ...

Links

☐ Current page is first sibling

Others

User value:

图 23

Attributes

SITEEDIT:Navbar ▾ ×

Specification

Link Destination

Link Level

☐ Top page

☐ Parent page

☐ Ancestor pages

☐ Current page

☐ Child pages

☐ Sibling pages

☐ Children of top page

☐ First child page

☐ Previous page

Previous page label,

☐ Next page

Next page label,

图 24

Ⓐ Attributes

SITEEDIT:Navbar ▾ ×

SpecificationLink DestinationLink Level

Start:1

End:3

☐ Limit links to direct children only

图 25