



[12] 发明专利申请公开说明书

[21] 申请号 02801119.8

[43] 公开日 2004年1月28日

[11] 公开号 CN 1471679A

[22] 申请日 2002.4.8 [21] 申请号 02801119.8

[30] 优先权

[32] 2001.4.9 [33] US [31] 60/282,609

[32] 2001.7.17 [33] US [31] 60/306,095

[86] 国际申请 PCT/US02/11247 2002.4.8

[87] 国际公布 WO02/082326 英 2002.10.17

[85] 进入国家阶段日期 2002.12.9

[71] 申请人 XML 都市公司

地址 美国加利福尼亚州

[72] 发明人 E·S·黄 H·J·金

[74] 专利代理机构 北京纪凯知识产权代理有限公司

司

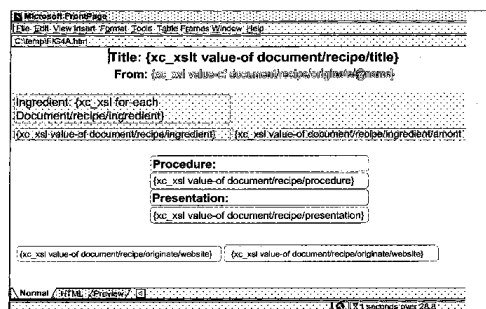
代理人 沙捷 彭益群

权利要求书 6 页 说明书 16 页 附图 22 页

[54] 发明名称 采用元标志信息的可扩展格式底稿设计

[57] 摘要

本发明公开了采用元标志信息的可扩展格式底稿设计的方法和装置。一已设计格式底稿用于将面向内容的标记语言文件转换为目标文件以支持各种演示文稿和信息交换。依据一方面，一数据处理机制以在目标文件中插入元标志说明开始，其中元标志信息属于在目标文件的动态对象和—源文件之间的信息。此带有元标志信息的目标文件然后被转换为一个或更多可扩展格式底稿。



- 1、 一种用于产生一可扩展格式底稿的方法，此方法包括：
接收第一标记语言的一目标文件；
接收至少一个第二标记语言的源文件；
确认目标文件中的动态对象；
- 5 通过元标志在每个目标文件的动态对象和至少一个源文件的元素之一之间建立联系；
用元标志替换每个动态对象；以及
参照目标文件产生可扩展格式底稿。
- 2、 如权利要求 1 所述的方法，其中所述接收第一标记语言的一
10 目标文件包括在一能以所需的方式显示目标文件的授权工具中显示目标文件。
- 3、 如权利要求 2 所述的方法，其中目标文件包括静态内容和动态对象。
- 4、 如权利要求 3 所述的方法，其中第一标记语言从一组包括超
15 文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）和无线标记语言（WML）中进行选择。
- 5、 如权利要求 5 所述的方法，其中第二标记语言从一组包括超
文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）和无线标记语言（WML）中进行选择。
- 20 6、 如权利要求 5 所述的方法还包括显示目标文件的演示文稿，其中演示文稿包括静态内容和多个动态对象。
- 7、 如权利要求 6 所述的方法，其中所述接收至少一个第二标记语言的源文件包括：
接收包括文档类型定义（DTD）的一定义文件；
25 在定义文件中的至少一个定义与动态对象之一之间建立联系；以

及

从目标文件产生一树结构，其中树包括结点，每个结点对应显示演示文稿的动态对象之一。

8、如权利要求 7 所述的方法，还包括遍历树结构以确定用于元标志的信息。

9、如权利要求 8 所述的方法，其中所述参照目标文件产生可扩展格式底稿包括产生第三标记语言的一结果文件，此结果文件包括至少一包括一说明和用于元标志信息的信息段。

10、如权利要求 9 所述的方法，其中目标文件可通过显示连同可扩展格式底稿的至少一源文件而被显示。

11、一用于产生一格式底稿的方法，此方法包括：

接收一标记语言的目标文件，此目标文件包括多个动态对象；

接收一结构树；此结构树的每个结点对应于目标文件的动态对象之一；

15 通过一元标志在目标文件的每个动态对象与至少一个源文件的一元素之间建立联系；以及

参照目标文件产生可扩展格式底稿。

12、如权利要求 11 所述的方法，其中所述接收一标记语言的目标文件包括在一能以所需的方式显示目标文件的授权工具中显示目标文件。

13、如权利要求 12 所述的方法，其中每个动态对象与出现在结构树中的一个或更多源文件中的一相应元素相关。

14、如权利要求 13 所述的方法，其中标记语言从一组包括超文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）和无线标记语言（WML）中进行选择。

15、如权利要求 14 所述的方法，其中结构树是利用从一组包括超文本标记语言（HTML），可扩展标记语言（XML），标准通用标

记语言（SGML）和无线标记语言（WML）中进行选择的标记语言而建立的。

16、如权利要求 11 所述的方法，其中所述在目标文件的每个动态对象与一元素之间建立联系包括：

- 5 在每个动态对象与结构树的结点之一建立联系；以及
 遍历结构树以获得用于元标志的相关信息，其中相关信息包括在每个动态对象和在至少一源文件的元素之间的关联。

17、如权利要求 11 所述的方法，其中所述在目标文件的每个动态对象与一元素之间建立联系包括：

- 10 将每个动态对象复制到结构树的结点之一；以及
 遍历结构树以获得用于每个动态对象的元标志的相关信息，其中相关信息包括在每个动态对象和在至少一源文件的元素之间的关联。

18、如权利要求 11 所述的方法，其中所述在目标文件的每个动态对象与一元素之间建立联系包括：

- 15 将元素经由结构树的结点之一从至少一源文件复制到每个动态对象；以及
 遍历结构树以获得用于每个动态对象的元标志的相关信息，其中相关信息包括在每个动态对象和在至少一源文件的元素之间的关联。

19、一用于产生一格式底稿的方法，此方法包括：

- 20 接收第一标记语言的目标文件，此目标文件包括多个动态对象，至少两个动态对象与第二标记语言的源文件的一元素相关；
 提供对应于一个或更多包括源文件的源文件的结构树，结构树的每个结点对应于目标文件的动态对象之一；
 通过多个元标志经由结构树将动态对象与源文件分别建立联系；
25 确认用于涉及到源文件中相同元素的至少两个动态对象的相应元标志，其中相应元标志被不同的标识符标记；以及
 参照目标文件产生可扩展格式底稿。

20、如权利要求 19 所述的方法，其中第一标记语言从一组包括

超文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）和无线标记语言（WML）中进行选择。

21、如权利要求 20 所述的方法，其中第二标记语言从一组包括超文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）和无线标记语言（WML）中进行选择。

22、如权利要求 19 所述的方法，其中所述通过多个元标志经由结构树将动态对象与源文件分别建立联系包括：

用元标志之一替换每个动态对象；以及

遍历结构树以获得用于每个元标志的相关信息，其中相关信息包括在每个动态对象和至少一源文件中的一元素之间的关联。

23、如权利要求 19 所述的方法，其中所述通过多个元标志经由结构树将动态对象与源文件分别建立联系包括：

将每个动态对象复制到结构树的结点之一；以及

遍历结构树以获得用于每个元标志的相关信息，其中相关信息包括在每个动态对象和在至少一源文件的元素之间的关联。

24、如权利要求 19 所述的方法，其中所述确认用于至少两个动态对象的相应元标志包括在用于相应元标志的每个说明中插入不同标识符。

25、如权利要求 24 所述的方法，其中不同标识符在一组包括数字、字符、文本和文字数字符号中选择。

26、一用于产生可扩展格式底稿的装置，此装置包括：

一显示屏；

用于存储作为一授权工具的代码的第一存储器；

用于存储作为软件模块的代码的第二存储器；

一处理器，与第一和第二存储器连接，当指示运行关于第一标记语言的目标文件的授权工具，使得目标文件作为输出演示文稿显示在显示屏上，其中输出演示文稿包括多个动态对象，每个对象涉及一个或更多第二标记语言的源文件中的一个元素，而且一属于源文件的结

构树然后被检索，结构树的每个结点相应于目标文件的动态对象之一；
以及

其中软件模块处理运行以执行通过一元标志在每个动态对象和一个或更多源文件中的一个元素之间建立联系的操作；

- 5 遍历结构树以获得用于元标志的相关信息；
以及参照目标文件产生可扩展格式底稿。

27、 如权利要求 26 所述的装置，其中第一标记语言从一组包括超文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）和无线标记语言（WML）中进行选择。

- 10 28、 如权利要求 27 所述的装置，其中第二标记语言从一组包括超文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）和无线标记语言（WML）中进行选择。

29、 一计算机可读介质，至少包括运行在计算设备上用于产生一可扩展格式底稿的计算机程序代码，此计算机可读介质包括：

- 15 用于接收第一标记语言的一目标文件的程序代码；
用于接收至少一个第二标记语言的源文件的程序代码；
用于确认目标文件中的动态对象的程序代码；
用于通过元标志在每个目标文件的动态对象和至少一个源文件的元素之一之间建立联系的程序代码；
20 用于用元标志替换每个动态对象的程序代码；以及
用于参照目标文件产生可扩展格式底稿的程序代码。

30、 如权利要求 29 所述的方法，其中用于接收第一标记语言的一目标文件的程序代码包括在一能以所需的方式显示目标文件的授权工具中显示目标文件的程序代码。

- 25 31、 一计算机可读介质，至少包括运行在计算设备上用于产生一可扩展格式底稿的计算机程序代码，此计算机可读介质包括：

用于接收一标记语言的目标文件的程序代码，此目标文件包括多个动态对象；

用于接收一结构树的程序代码，此结构树的每个结点对应于目标文件的动态对象之一；

用于通过一元标志在目标文件的每个动态对象与至少一个源文件的一元素之间建立联系的程序代码；以及

5 用于参照目标文件产生可扩展格式底稿的程序代码。

32、如权利要求 31 所述的计算机可读介质，其中用于接收一标记语言的目标文件的程序代码包括在一能以所需的方式显示目标文件的授权工具中显示目标文件程序代码。

33、如权利要求 32 所述的计算机可读介质，其中每个动态对象
10 与出现在结构树中的一个或更多源文件中的一相应元素相关。

34、一计算机可读介质，至少包括运行在计算设备上用于产生一可扩展格式底稿的计算机程序代码，此计算机可读介质包括：

用于接收第一标记语言的目标文件的程序代码，此目标文件包括多个动态对象，至少两个动态对象与第二标记语言的源文件的一元素
15 相关；

用于提供对应于一个或更多包括源文件的源文件的结构树的程序代码，结构树的每个结点对应于目标文件的动态对象之一；

用于通过多个元标志经由结构树将动态对象与源文件分别建立联系
20 的程序代码；

用于确认用于涉及到源文件中相同元素的至少两个动态对象的相应元标志，其中相应元标志被不同的标识符标记的程序代码；以及
用于参照目标文件产生可扩展格式底稿的程序代码。

采用元标志信息的可扩展格式底稿设计

相关申请的相互参考

本申请要求了以下美国临时申请的权益：

5 美国临时申请 NO.60/282609, 申请日 2001 年 4 月 9 日, 名称为“用于采用元标志信息的可扩展格式底稿设计的方法和装置”；

美国临时申请 NO.60/306095, 申请日 2001 年 7 月 17 日, 名称为“用于采用元标志信息和/或相关元标志信息的可扩展格式底稿设计的方法和装置”；

10 美国临时申请 NO.60/314592, 申请日 2001 年 8 月 23 日, 名称为“用于可扩展标记语言转换和采用元标志信息和/或相关元标志信息的可扩展格式底稿设计的方法和装置”；以及

美国临时申请 NO.60/349957, 申请日 2002 年 1 月 17 日, 名称为“可扩展标记语言转换和采用元标记信息和/或相关元标记信息的格式底稿设计”；

15 以上申请作为参考包括在本申请之中。本申请也与美国未决专利申请 NO.09/754969, 名称为“用于利用文件类型定义产生结构文件的方法和装置”相关, 仅供参考。

20 发明背景

技术领域

本发明涉及文件处理和电子出版系统领域, 而且尤其是, 涉及用于设计可扩展格式底稿的技术。

25 背景技术

Internet 是急剧增长的遍及全球的计算机互连通信网络。数以百万计连接的计算机共同形成了一个易于被任何互连的计算机在任何时间任何地点访问的巨大的超链接信息存储库。随着每天或每年数以百万计网页被建立并加入此巨大存储库, 非常需要能迅速并容易地将诸如

演示文稿、数据底稿或小册子的文档转换为一种能供 Internet 上其他应用或设备显示和访问的格式。

众所周知，可供万维网浏览应用（即浏览器）显示的较优格式是诸如超文本标记语言（HTML），可扩展标记语言（XML），标准通用标记语言（SGML）或无线标记语言（WML）。由万维网浏览应用撰写、编辑或管理的文件或文档通常是指结构化文件或文档。在结构化文档中，一诸如 XML 或 SGML 的面向内容的结构化文档能支持任何来自文档类型定义（DTD）或模式的标识标志，而 HTML 或 WML 的演示文稿文档只能支持一套固定的标识标志。

采用适用于特殊要求媒体的格式变换来动态转换面向内容结构化文档的命令开创了信息交换和存储的新范例。例如，网站内容以 XML 格式进行存储。如果台式电脑中的万维网浏览器（即 Netscape 或 IE）请求来自网站的信息，网站优先将 XML 内容转换为 HTML 已转换内容并随后将已转换文件发回浏览器。如果一便携设备（即 PDA 或可访问 Internet 单元电话）通过一微浏览器向相同网站请求信息，网站优先将 XML 内容转换为 WML 已转换内容并将已转换文件发回微浏览器。而且，网站也能将内部 XML 转换为另一类型的 XML 以在网站间交换信息。

前述用于各种出版和信息交换的例子需要从 XML 适当转换为各种格式。

发明内容

本单元概述本发明一些方面并简要介绍最佳实施例。简化或省略可避免使本单元目的不明确。此简化或省略并不限制本发明的范围。

本发明属于利用元标志（meta-tag）信息来设计用于将源 XML 文件转换为目标文件的可扩展格式底稿（XSL）。根据本发明的一个方面，当显示目标文件时（即在浏览器或授权工具上），输出演示文稿包括许多对象，例如图画或文句或一组单词。一些对象在某种意义上是动态的，这些对象分别与源文件中的源元素或对象链接，使得源对象的任何改变将动态地映射到目标文件中。每个插入元标志（meta-tag）规定了与源文件中相应源对象的关系。作为本发明的特征之一，放置

源标志的独特方法有赖于申请和实现来提供。

一旦在目标文件中创建了元标志，一转换模块自动产生关于目标文件的结果 XSL 文件。转换模块搜索元标志并随后产生用于目标文件的 XSL。

- 5 本发明可以方法、系统、软件产品或其他形式实现。根据一种实现方式，本发明是一种方法。此方法接收用于源 XML 的包括文档类型定义 (DTD) 的定义文件并通过剪贴板、拖放机制或 OLE 数据传送机制从定义文件中输出元标志信息至用于目标文件的授权工具。嵌入源元标志信息的目标文件进一步转换为 XSL 文件。在一种实现中，元标志信息从模式文件 (schema file) 或源 XML 文件输出。在另一实现中，源标志信息通过在目标文件的授权工具直接输入来创建。然而，XSL 文件从包括至少一表示与源文件结合关系的元标志的目标文件中创建。

- 10 前述所产生的目的、利益和优点可由本发明以下参照附图描述的实施例而获得。

附图说明

本发明的这些以及其它特征、方面以及优点将在以下说明书、所附权利要求和附图中更为清楚：

- 20 图 1A 表示依据本发明最佳实施例实现的基本系统配置；
图 1B 描述了本发明实现并达到预期结果的系统内部结构块；
图 2A 是用于接收类型文档的文档类型定义 (DTD) 的例子；
图 2 B 表示基于图 2A 的文档类型定义 (DTD) 的可扩展标记语言(XML)文件；
25 图 3A 表示可扩展格式底稿语言 (XSL) 文件，用来将如图 2B 的 XML 文件转换为 HTML 文件；
图 3B 表示图 3A 的已被转换的 HTML 文件的屏幕抓图；
图 3C 表示一 XSL 文件，将图 2B 的 XML 文件转换为一 WML 文件；
30 图 3D 表示一 XSL 文件，将图 2B 的 XML 文件转换为另一 XML 文件；

图 4A 表示一帶有插入元标志信息的目标 HTML 的屏幕抓图；

图 4B 表示一帶有插入元标志信息的目标 XML 的屏幕抓图；

图 5A 表示通过一拖放机制输出元标志信息的 XML/DTD/模式编辑器 (XDTD) 产生的目标文件的输出表示；

5 图 5B 表示利用 XDTD 编辑器创建用于 HTML 文件的 XML 文件的 DTD 以及包括所创建 DTD 的扩展视图的例子，基于所创建 DTD 而创建的 XML 元素以及带有元素数据结点和元素属性结点的 XML 树的扩展视图；

图 5C 表示相应 XML 文件；

10 图 5D 表示用于 XML 文件树结构的数据结点 (用于相应元标志)；

图 6A 表示在 FrontPage 中帶有插入元标志信息的目标 HTML 文件，尤其是，此 HTML 文件是静态文本，例如 “Title”、“From”、“Ingredient”……，和来自源 XML 文件的动态信息或对象的组合；

15 图 6B 表示在 FrontPage 中帶有插入元标志和相关元标志信息的目标 HTML 文件的一个例子；

图 6C 表示一元标志相关文件，包括分别由元标志来表示的动态对象，其中每个元标志均以 {xc_xsl for-each} 开始，以规定相关 XSL 动作和/或源 XML 标志；

20 图 7A 描述了数据处理装置方框图，在此装置中帶有元标志信息的目标文件由授权工具创建并通过转换器转换为 XSL 文件；

图 7B 描述了依据本发明一实施例产生一可扩展格式底稿的流程图；

图 8A 描述了帶有作为根元素的 <root/> 的 XML 文件；

图 8B 描述了参照图 8A 中文件相应的 HTML 文档；

25 图 8C 表示包括帶有分 XML 文本数据的元素的 HTML 文档的一个例子；

图 9 表示处理动态对象流程图，此对象涉及源文件中相同元素并可结合附图 8A-8C 理解。

30 具体实施方式

在本发明以下的详细描述中，为全面理解本发明，对大量特定细

节进行了阐述。然而，很明显对于本领域普通技术人员即使不参照这些特定细节也可实现本发明。另一方面，公知的方法、过程、组件和电路未详细描述以避免本发明不必要的不清楚的方面。详细描述大多按照过程、逻辑块、处理和其他直接或间接类似于耦合到网络的数据处理设备操作的符号表示来给出。这些处理描述和表达是本领域普通技术人员用来将他们的工作材料最有效传送给本领域其他普通技术人员的方法。

在此参照“一个实施例”或“一实施例”方法，其中结合此实施例所描述的一特殊特征、结构或特性可被包括在本发明的至少一个实施例中。出现在说明书各处的词组“在一个实施例中”并非都指相同实施例，也不是与除其他实施例之外独立或可选的实施例。而且，在描述本发明一个或更多实施例的处理流程图或图中的块命令并不固定表示任何特殊命令也不对本发明做任何限制。

参照附图，其中几个视图中相同部分采用相同附图标记。图 1A 表示依据一最佳实施例本发明实现的基本系统配置。包括诸如产品描述、功能列表和价格表信息的面向内容的文档可利用运行在计算机 100 上的授权工具来创建。更可取地，这些文档为 XML（可扩展标记语言）格式。XML（可扩展标记语言）是一种创建普通信息格式以及共享万维网、Intranet 和其他网络上格式和数据的灵活方法。因为标识符不定且自定义，所以不像 HTML，XML 是“可扩展的”。在如何创建文档结构的标准上，XML 实际上是标准通用标记语言的更简单和更易用的子集。

可在计算机设备 102 中创建用于将这些文档转换成各种表示的可扩展格式底稿（或 XSL），这些计算机设备可为下载有本发明实现的一个实施例的可运行版本的服务器或台式计算机。

在一种设置中，计算机 100 和计算机设备 102 是不分离的且执行文档转换处理最终以诸如 WML 和 HTML 的标记语言格式表示内容。在一种应用中，XML 表示并经 XSL 变换的面向对象文档可通过一专用网络 110 传送至其主机通常作为 WWW(万维网)站点的服务服务器 104。

在一种情况下，一用户利用一操作浏览应用并与数据网络 108 耦

合的台式计算机 106 来访问服务服务器 104 上的文件。在另一情况下，一用户利用一操作 WAP 浏览应用并（也许经由一无线网络）与数据网络 108 耦合的 PDA110 访问服务服务器 104 的文件。这些被请求的 XML 文件将通过适当的 XSL 文件被转换成 WML 文件并发回给 PDA

5 正如以下所要描述的，本发明并不局限于 Internet 应用。本发明可应用于用户经常脱机创建 XSL 文件以将 XML 文件转换为 HTML 或 WML 文件的单机。由于内容和演示文稿分开，用户可方便地创建另一套用于不同视觉和感觉的 XSL 文件。

图 1B 表示可实现并运行本发明的系统 118 的内部结构块。系统
10 118 对应于一客户设备（即计算机 100、102 或 106）或一服务器设备（即服务器 104）。如图 1B 所示，系统 108 包括与数据总线 120 相连的中央处理单元（CPU）122 和设备接口 124。CPU122 运行一定指令以管理所有与用于同步操作的数据总线 120 相连的设备和接口而且设备接口 124 可与诸如计算机 102 的外部设备相连，因此其中的文档被
15 接收并通过数据总线 120 存入内存或存储器。与数据总线 120 的接口包括一显示接口 126、一网络接口 128、一打印接口 130 和一软盘驱动接口 138。通常，本发明一个实施例的可运行版本通过软盘驱动接口 138、网络接口 128、设备接口 124 或其他与数据总线相连的接口被下载并存入内存（即 132、134 或 136）。

20 诸如随机存储器（RAM）的主存 132 也与数据总线 120 相连以提供给 CPU122 指令和可访问存储数据和其他指令的内存存储器 136 的 CPU122。尤其是，当运行所存储的应用程序指令时，例如本发明中已编译和链接的版本，CPU122 通过操作数据得到本发明预期结果。假如存在任何恒定指令序列，ROM（只读存储器）134 用来存储诸如用于
25 键盘 140、显示器 126 和浮点设备 142 操作的基本输入/输出操作系统的恒定指令序列。

依据一个实施例，一诸如 XML 的面向内容文档以文档类型定义 (DTD)或模式开始来定义文档元素。图 2A 描述了一个用于“接收类型”文档的 DTD200 的例子，其中一文档被分成文档元素结构。一特殊文档元素 202 可包含其他文档元素和属性。文档元素 204 的另一例子只
30 包括语法分析字符数据。图 2B 表示 XML 文件的一个例子，其中信息

用如图 2A 的文档类型定义 (DTD) 定义的语法分析数据元素表示。

一 XML 文件可采用可扩展格式底稿 (XSL) 转换成各种表示。将 XML 文件 206 转换成 HTML 文件的 XSL 文件的一个例子如图 3A 所示。尤其, 块 302 表示当 XSL 文件匹配标志 “document” 时创建顶级
5 <html>标志以及块 304 表示如何将值从源 XML 文件值分配到目标 HTML 文件中。从一浏览器应用 (即 Microsoft Internet Explorer) 所得到的 HTML 对应的屏幕抓图如图 3B 所示。

将 XML 文件 206 转换成 WML 文件的 XSL 文件的另一例子如图 3C 所示。尤其, 块 314 表示当 XSL 文件匹配根元素时创建顶级<wml>
10 标志, 块 312 表示如何将 “title” 值从源 XML 文件值分配到目标 WML 文件中, 以及块 316 表示如何将 “originate” 元素的 “name” 属性从源 XML 文件值分配到目标 WML 文件中。

除了将 XML 文件转换成各种表示, 一 XSL 文件也能将 XML 文件转换成带有不同 DTD 或模式的另一 XML 文件。此种转换的首要原因
15 是不同系统间的信息交换。图 3D 表示将 XML 文件 206 转换成另一带有一列接收标题的 XML 文件的 XSL 文件的例子。尤其, 块 320 表示当 XSL 文件匹配 “document” 元素时创建顶级<recipe_title_list>标志以及块 322 表示如何将源 XML 文件的每个 “title” 值分配到目标 XML 文件的<recipe_title>标志中去。

从以上 XSL 的例子可知 XSL 文件是一目标文件 (即以 HTML、
20 WML 或 XML 格式) 和<xsl:>标志的组合, 它将来自源 XML 文件的信息和/或通过源 XML 的 DTD 或模式文件的信息建立联系。本发明属于设计 XSL 文件的技术。依据一个实施例, 使用了一授权工具。此授权工具可为用于 HTML 的 FrontPage、用于 WML 的 Intava Gravity
25 Professional 以及用于 XML 的 XML 编辑器。具有这些 WYSIWYG 授权工具之一, 用户可设计目标文件的视觉一和一感觉或一帶有 XML 编辑器的输出演示文稿并且用户可设计表示源对象 (即文件元素) 间继承关系的目标 XML 树, 以上均在未决美国专利申请:NO.09/754969 中详细描述。然后, 用户可利用对应于目标文件的授权工具插入规定相应源信息的元标志信息。带有元标志信息的目标文件可被下载到自动
30 产生关于目标文件或演示文稿的结果 XSL 文件的 XSL 转换模块。

图 4A 表示在 FrontPage 中带有插入元标志信息的目标 HTML 文件的一个例子。例如，一原始对象“Green Chili Salsa”被一元标志“xc_xslt value-of document/recipe/title”替换。尤其，此 HTML 文件是诸如“Title:”、“From:”、“Ingredient:”的静态文本和来自源 XML 文件的动态信息（即各对象）的组合，其中动态对象用元标志表示，在此例中，包括在一说明{xc_xslt...}中。依据一个实施例，在一说明中，存在至少两项，一项用于用户规定操作，例如“value-of”，其他项用于路径信息，例如“document/recipe/title”，这样可规定相关 XSL 动作和/或源 XML 文件中的源对象。而且，元标志说明中的诸如字体大小、颜色、样式的文本也将在转换过程中被传入动态信息。图 4B 表示在转换后产生用于接收标题列表的 XSL 文件 318 的 XML 编辑器中带有插入元标志信息的目标 XML 文件的另一例子。为更易于对本发明进行详细描述，此处提供基于 HTML 的一个例子并不应作为对本发明的限制。本领域普通技术人员应能理解此处例子以及描述可很容易地应用到其他标记语言。对于一所给将被转换为 XML 的 HTML 目标文件，可通过诸如 Microsoft Internet Explorer 或文本编辑器的浏览器打开 HTML 目标文件。所示 HTML 文件不仅包括 HTML 文件内容还包括 HTML 定义用来描述当用浏览器（即 Internet Explorer）浏览时 HTML 文件视觉和感觉的配套标志。

一旦 HTML 文件被浏览器打开，用户可打开一诸如商业地址为 1617A S.Main Street,Milpitas,CA95035 的 XMLCities 公司的 XDTD 编辑器的 XML 实用程序并创建如图 5A 所示的 DTD 元素 441。在 DTD 池 440 中，用户能创建新元素、新实体、新符号、新 PI 以及新注释，它们中的一些直接与浏览器所显示的对象相关。图 5B 表示利用 XDTD 编辑器创建用于 HTML 文件的 XML 文件的 DTD 的一个例子，且包括所创建的 DTD 的扩展视图 451、基于所创建的 DTD 创建 XML 元素的一个例子 452 以及带有用于元素的数据结点和属性结点的 XML 树 453 的扩展视图。

一旦 XML 树被创建，用户可复制诸如图 5A 的一对象 461 的 HTML 数据内容的一部分并可将其作为 XML 树中标题元素的数据通过剪贴板、拖放机制或 OLE 数据转换机制插入到 XDTD 实用程序。HTML

数据的其他部分可以相同方式被插入到 XML 树中。

在 HTML 文件中必要的相应内容或对象被插入到 XDTD 编辑器的 XML 树后，用户可创建如图 5C 所示的 XML 文件。基于所给的 XML 文件和/或相应 DTD 文件，依据本发明的一个方面，创建 XSL 文件以反映当 XML 文件为源文件时 HTML 文件所表达的视觉和感觉。不像 HTML 的情况，XML 的元素名称无固有演示文稿语义。缺少格式底稿，处理器（即 Microsoft Internet Explorer）除了一串一致的字符以外也许不知如何提供文档内容。从输出演示文稿并利用 DTD 和 XML 树产生 XML 源更详细的介绍可参看美国专利申请：NO.09/754, 969。

在某个方面，XSL 提供了一综合模型和采用 XML 语法写此格式底稿的词汇。为创建一 XSL 文件，一自动 XSL 产生模块也是本发明要实现的。依据一个实施例，元标志用于将源文件（即一 XML 文件）中各元素或源对象与目标文件（即一 HTML 文件）中某些动态对象建立联系。此处的元标志或元标志信息指任何来自源文件（即 XML 文档）的信息，它包含与源文件中特殊元素相关的信息以及它的元素与源文件中其他元素之间的关系。例如，如图 5D 所示的数据结点 476 的元标志是 {xc_xslt:document/recipe/title}，其中 document/recipe/title 表示数据结点的位置和与其父结点即 title 结点及 XML 文件中其他元素的关系。相关元标志信息指任何来自诸如包含或能用于显示有关 XML 文档的元标志信息的信息的 HTML 或 XML 文件的数据。例如，如图 5D 中 471 是一相关元标志信息，因为它能用于搜索元标志信息 476，就像 XML 文件中 {xc_xslt:document/recipe/title} 的 document/recipe/title。可选地，元标志信息包括至少一个用户规定的操作，例如 <xsl:value-of>。

在本发明的一个实施例中，XML 文件可存储在计算机内存中并可利用诸如万维网共同体（W3）的文档对象模块（DOM）核的 XML 语法分析程序搜索元标志，还可使用其 API，例如 parentNode()、nextSibling()、childNodes()、getNodeName()、nodeName() 和 nodeValue()。

由于 XML 文件结点为对象分级体系，图 5D 的 476 可作为用于获得元标志信息的相关元标志信息，将 DOM 语法分析程序指针定位在 DOM 树的根元素上、利用 nodeName() 来确认并将根元素的名称以 {xc_xslt:document} 存储至单独的存储器中、利用 NodeValue() 来比较它

的结点值和元标志信息 476。假如它们不相等，利用 `firstChild()` 移动到它第一个子元素即接收元素，利用 `nodeName()` 来确认并通过将存储器位置内容与作为 `{xc_xslt:document/recipe}` 接收的新结点名称并置而将结点名称存储至与 `{xc_xslt:document}` 相同的存储器位置，并将它的 `nodeValue()` 与元标志信息 476 进行比较。递归以上方法可得到元标志信息 476，例如 `{xc_xslt:document/recipe/title}`，直到当 `nodeValue()` 等于图 5D 中的 471 时，利用相关元标志信息 471 停止递归。

同样，用户可利用反向递归方法得到用于图 5D 中的 475、472 和 473 的元标志信息。因此，用户可发现所有包含相关元标志信息的数据结点并随后利用 DOM 语法分析程序的 `parentNode()` API 在反相并置元标志信息，如 `{xc_xslt:title}` 至 `{xc_xslt:recipe/title}` 和 `{xc_xslt:recipe/title}` 至 `{xc_xslt:document/recipe/title}` 时来建立元标志信息。在本发明的另一实施例中，用户也可利用带有附加条件的相同递归方法得到元标志信息 476、472 和 473，此附加条件例如为假如 `nodeValue()` 等于相关元标志信息，则 DOM 语法分析程序指针将利用来自建立元标志信息的并置名称的 `nextSibling()` 跳过元素结点。因此，由于用于元标志信息通过关联来建立，图 5D 的 475、472、473 是相关元标志信息。

用户还可利用带有插入元标志信息和/或相关元标志信息的 WYSIWYG 授权工具来设计目标演示文稿的视觉一和一感觉。带有元标志信息和/或相关元标志信息的目标文件将下载到可自动产生与 WYSIWYG 授权工具中目标演示文稿的视觉一和一感觉相匹配的 XSL 文件的 XSL 转换器。在一个实施例中，假如目标文件包含相关元标志信息，此相关元标志信息可用于发现包含相应于目标文件相关元标志信息的 XML 文档中的元标志信息，并在 XSL 文件创建处理中用 XML 文档中的元标志信息替换目标文件的相关元标志信息。

依据本发明的一个实施例，一简单自动 XSL 转换可以以下方式实现。当需要保持往浏览器所看到的 HTML 文件的视觉一和一感觉，首先依据 XML 标准检测并编辑带有元标志信息和相关元标志信息的 HTML 文件的演示文稿格式。对 HTML 文件进行改变，例如所有标志具有开始和结束构件、空元素正确格式化，引用所有属性值以及正确

嵌套所有标志。一旦对 HTML 文件依据 XML 标准作必要改变并确认其格式，HTML 文件可通过诸如被万维网共同体（W3）定义为 DOM 核的 XML 语法分析程序被下载到计算机存储器。

一旦 HTML 文件被 DOM 语法分析程序作为一级结点对象下载到用户计算机存储器，DOM 语法分析程序的 API 首先分离 HTML 文档数据结点值核属性结点值。假如 HTML 文件包含相关元标志信息，带有/不带有相关 DTD 文件并包含匹配相关元标志信息的 XML 文件可用于确定元标志信息。一旦从 XML 文件中确认元标志信息，相关元标志信息将被所发现的元标志信息替换。

所有下载到计算机存储器的 HTML 文档的元标志信息进一步进行处理，所以假如元标志信息为 {xc_xslt:document/recipe/title}，则将被替换为 { xsl:value-of select="document/recipe/title" }。假如元标志信息是 HTML 文档属性结点值，元标志信息可被替换为 { xsl:value-of select="document/recipe/image/@img" }。在 XML 语法分析程序中的 HTML 的元标志信息和结构可进一步改变使得它能支持任何用户选择的 XSL 元素。

当 `<?xml version='1.0'encoding='ISO-8859-1'standalone='yes'>`
`<xsl:stylesheet verrsion="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">` `<xsl:template match="/">`可加到 XSL 说明之后时，XSL 的说明将通过 XML 语法分析程序 API（DOM 核）而被加到语法分析 HTML 文档的开始处，而且 `</xsl:template></xsl:stylesheet>` 将通过 XML 语法分析程序被加到语法分析 HTML 文档的结尾以将语法分析 HTML 文档变换为一有效 XSLT 文档并将 XSLT 文档作为有效 XSL 文件输出。

在本发明创建简单 XSL/XSLT 文件的另一实施例中，用户开始通过 XML Citites 的 XDTD 编辑器的输入实用程序输入一 XML 文件，并通过 WYSIWYG 授权工具的剪贴板、拖放机制或 OLE 数据转换机制插入 XML 文件的元标志信息或相关元标志信息来动态创建新 HTML 文件。

图 6A 表示在 FrontPage 中带有插入元标志信息的目标 HTML 文件的一个例子。尤其，此 HTML 文件是静态文本，例如“Title”、“From”、“Ingredient”……，和来自源 XML 文件的动态信息或对象的组合。而

且动态信息在元标志中表示,在此例中此元标志包含说明 {xc_xslt...} 且在此说明中存在诸如 “value-of document/recipe/title” 的值用来规定相关 XSL 动作和/或源 XML 标志。在另一例中, 元标志可为其他规定相关 XSL 动作和/或源 XML 标志文本字符串。而且, 文本装饰也将在转换过程中被传入动态信息。

图 6B 表示在 FrontPage 中带有插入元标志信息和相关元标志信息的目标 HTML 文件的一个例子。尤其, 此 HTML 文件是静态文本, 例如 “Title”、“From”、“Ingredient” …… , 和来自源 XML 文件的动态信息或对象的组合。而且在此例中动态信息在元标志 {xc_xsl for-each} 和间接由如图 6C 所示的元标志相关文件给出以规定相关 XSL 动作和/或源 XML 标志的相关元标志中表示。在另一例中, 相关元标志可为其他作为唯一规定 XSL 动作和/或源 XML 标志的相关元标志标识符的文本字符串。尽管利用相关元标志需要附加映射, 相关元标志可保持最后演示文稿的视觉一和—感觉, 这在诸如 FrontPage 的 WYSIWYG 授权环境中尤其重要。

存在几种方法在目标授权工具中插入元标志信息。在一实现中, 通过在授权工具中直接输入元标志说明来插入元标志。在另一实现中, 通过从 XML/DTD/模式编辑器至授权工具的剪贴板、拖放机制或 OLE 数据转换机制插入元标志。

图 7A 描述数据处理装置 700 的方框图, 其中带有元标志信息的目标文件由授权工具 704 创建并由转换器 706 转换为 XSL 文件。而且在授权工具中通过直接输入元标志说明或从 XML/DTD/模式编辑器 702 的剪贴板、拖放机制或 OLE 数据转换机制来插入元标志信息。

图 7B 表示根据本发明的一个实施例产生可扩展格式底稿的流程 图 710。在 712, 接收到一目标文件。如上所述, 目标文件是用来在设备(即台式计算机或便携设备)上显示的演示文稿。例如, 利用 FrontPage 创建的 HTML 目标文件显示在运行在台式计算机上的 Internet Explorer。随着目标文件的显示, 在 714 下载相应一个或多个源文件(即 XML 文件) 连同相应 DTD 文件以便于创建可扩展格式底稿。接着, 在 716 确定在所显示的目标文件中的动态对象并且每个动态对象均对应于源文件中的源元素或对象。取决于实现, 动态对象可由用户人工

确认或在目标文件创建时指出。

在 718，每个元标志被确定对应动态对象之一。依据一个实施例，每个动态对象被一元标志或包括在一元标志中的路径信息替换。依据另一实施例，每个动态对象被拖动到由 XML 文件和/或 DTD 文件（如图 5D）提供的 XML 树上一相应结点。通过拖动一动态对象到一结点，可建立动态对象和源文件中相应元素的联系。

在 720，用于每个元标志的标志信息被确定，例如假设 XML 树的每个节点均分别对应每个元标志的相关信息，则可通过遍历 XML 树确定。作为 722 的结果，产生对应于目标文件的格式底稿。尤其，格式底稿以另一种标记语言表示在一文件中。此文件包括每个元标志的相应说明。

在某个目标文件中，存在涉及源文件相同元素的动态对象。为避免在确认相应元标志上可能存在不清楚的情况，可采用不同的确认机制。

图 8A 表示<root/>作为根元素的 XML 文件。XML 文件包含采用 XSLT 符号的两个元素并可确认为 <xsl:value of select="root/book[1]/name/text()"/> <xsl:value of select="root/book[2]/name/text()"/>。

XML 文件的数据和属性然后被置于 HTML 文件中使得 HTML 文件可用于产生一 XSLT，其中 XML 文件的数据和属性可用于确认放置标志<xsl:value of select=""/>的位置。例如，812 作为图 8A 中<title/>元素的数据被放置。依据一个实施例，用户可利用图 8A XML 文档与图 8B HTML 文档的一字符串匹配使得 HTML 文档的“C++ Programming”可被替换为<xsl:value of select="root/book[1]/name/text">。

然而，两个唯一可确定的元素的内容可作为一 HTML 元素标志即如图 8B 所示的<p/>的数据而给出。“C++ Programming, Visual C++5” 521 作为<p/>元素的数据给出。利用 DOM，“C++ Programming, Visual C++5” 521 将被表示为<p/>元素的文本数据。通过精确的字符串匹配，由于 XML 文件包含“C++ Programming”、“Michael Kim”“ Visual C++5”以及“Ivor Horton”而不包含“C++ Programming, Visual C++5”但期望的匹配是<p><xsl:value of select="root/book[1]/name/text()"/>，

<p><xsl:value of select="root/book[1] /name/text()"> 时， “ C++ Programming, Visual C++5” 将不会发现与如图 8A 的 XML 的任何数据或属性精确匹配。

本发明的特征之一是在不同存储器中存储所有 XML 数据和属性，其中每个 XML 数据和属性包含信息或与有关于元标志信息的链接。元标志信息包含有关于父元素和父元素与根结点关系的信息，因此元标志信息可有效寻址用于在<xsl:value of select="">中选择的正确的属性值。XML 文档的数据和属性以及它们相应的元标志信息将按一顺序存储使得数据或属性可以递增或递减的顺序访问。

依据实施例，接收包含 XML 文档的数据或属性的 HTML 文档被接收。然后开始确认 HTML 文档数据并将此数据与存储在不同存储器中的 XML 数据或属性进行比较。然后对按字符串长度顺序存储在不同存储器中的数据和数据或属性进行比较和确认以确定是否最大匹配 XML 数据或属性被包括在 HTML 数据中。

图 8C 表示包含带有分 XML 文本数据元素 HTML 文档的一个例子。图 8C 显示包含图 8A 中文本数据“C++ Programming” 500 的元素 530<p/>。然而，文本数据进一步被分为两个独立文本数据这使得“C++”为作为<p/>元素 530 的子的的子而且“Programming”为<p/>元素的子。应注意到 531 和 532 均属于<p/>元素。

依据本发明的一个实施例，为将分文本数据作为 HTML 元素的相关元标志信息，利用了 531 和 532 均属于为 530 的<p/>元素的事实。本发明的软件实现获得 530 的子元素的第一文本数据 531 并将其放入一不同的存储器。然后本发明最佳实施例的软件实现获得下一文本数据 532 作为 530 下一相邻子结点并将 532 存储在 530 的下一位置。对于被给的元素不再发现有下一子元素或子元素的子之后，本发明首选的软件实现搜索用于匹配的 XML 文本数据或属性，例如并置的 531 和 532。当发现匹配时，530 的整个内容将被不久将替换为<xsl:value of select="root/book[1]/name/text()">的 500 的元标志信息替换。

随着 HTML 文档的所有元标志信息被下载到计算机存储器中，所有元标志信息将进一步处理使得假如元标志信息是 {xc_xslt:document/recipe/title}，它将被替换为 <xsl:value-of

select="document/recipe/title">。假如元标志信息是如 475 的 HTML 文档属性结点值，元标志信息可被替换为 { xsl:value-of select="document/recipe/image/@img" }。在 XML 语法分析程序中的 HTML 的元标志信息和结构可进一步改变使得它能支持任何用户选择的 XSLT 元素。

图 9 表示处理动态对象的流程图，此对象涉及源文件中相同元素并可结合附图 8A-8C 理解。在 902，一目标文件被接收。如上所述，目标文件是用来在设备（即台式计算机或便携设备）上显示的演示文稿。例如，利用 FrontPage 创建的 HTML 目标文件显示在运行在台式计算机上的 Internet Explorer。随着目标文件的显示，在 904 下载相应一个或多个源文件（即 XML 文件）连同相应 DTD 文件以便于创建可扩展格式底稿。接着，在 906 确认在所显示的目标文件中的动态对象并且每个动态对象均对应于源文件中的源元素或对象。取决于实现，动态对象可由用户人工确定或在目标文件创建时指出。

在 908，确定是否任何已确认的动态对象涉及源文件中的相同元素。假如无动态对象涉及源文件中的相同元素，则处理 900 进入 912。假如动态对象的任意两个涉及源文件中的相同元素，处理 900 进入 910，其中这些动态对象被标记不同。依据一个实施例，这些动态对象依据它们出现的顺序被数字标记，用于标记它们的标识符便于创建格式底稿。在这些动态对象被标记之后，处理 900 进入 912。

无论如何，在 912，每个元标志被确定对应动态对象之一。依据一个实施例，每个动态对象被一元标志或包括在一元标志中的路径信息替换。依据另一实施例，每个动态对象被拖动到由 XML 文件和/或 DTD 文件（如图 5D）提供的 XML 树上一相应结点。通过拖动一动态对象到一结点，可建立动态对象和源文件中相应元素的联系。

在 914 用于每个元标志的标志信息被确定，例如假设 XML 树的每个节点均分别对应每个元标志的相关信息，则可通过遍历 XML 树确定。因此，在 916 产生对应于目标文件的格式底稿。尤其，格式底稿以另一种标记语言表示在一文件中。此文件包括每个元标志的相应说明。

本发明属于利用元标志信息设计用于转换源 XML 文件为目标文

件的可扩展格式底稿（XSL）。本发明有无数利益、特征和优点。其中之一是利用了涉及目标文件中动态对象和源文件中相应元素间信息关系或联合的元标志。其他利益、特征和优点本领域普通技术人员很容易获知。

- 5 本发明已充分详细并在一定程度的特殊性上进行了描述。本领域普通技术人员在不脱离本发明权利要求所限定的精神和范围内可对本发明公开的实施例在排列组合上进行多种修改。因此，本发明的范围由所附权利要求而非前面对实施例的描述来限定。

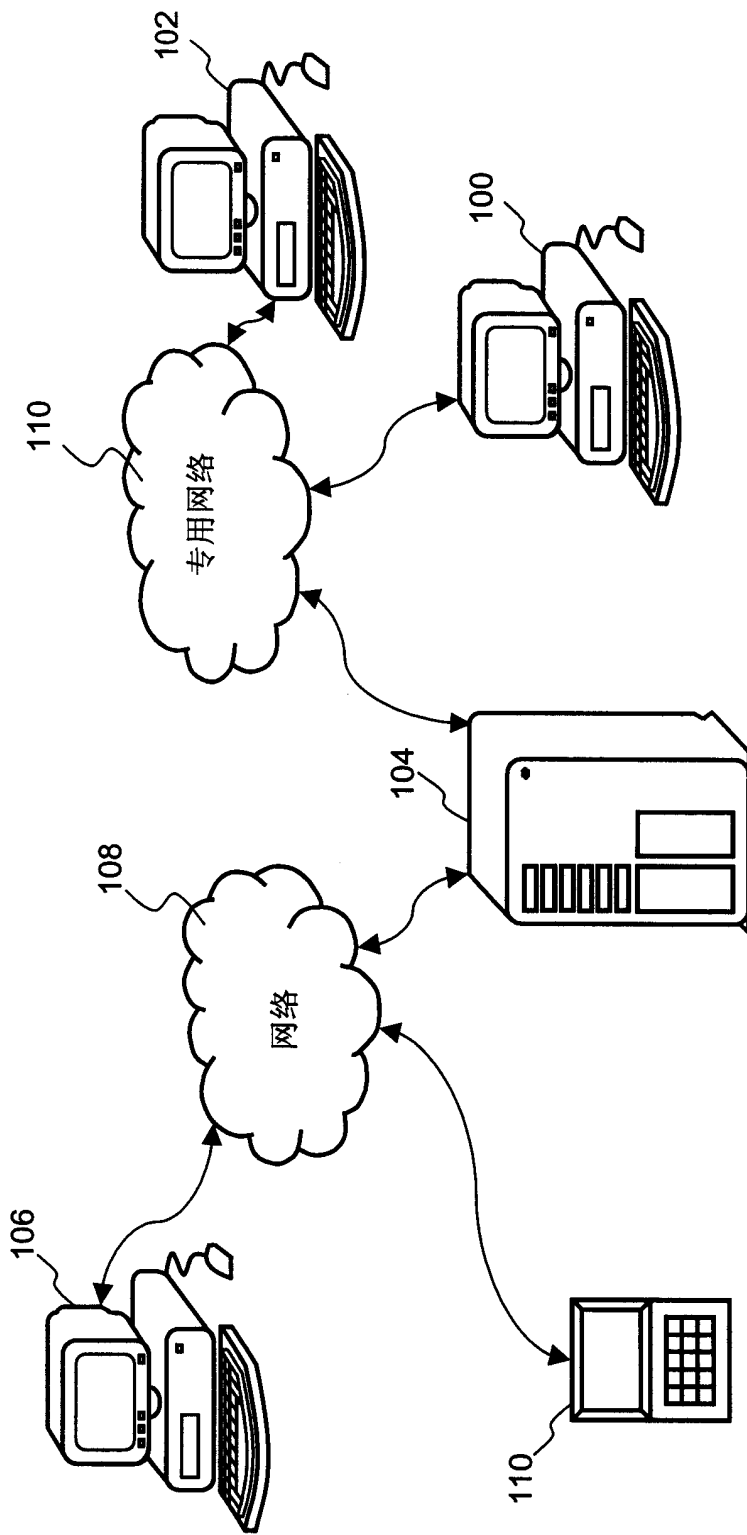


图1A

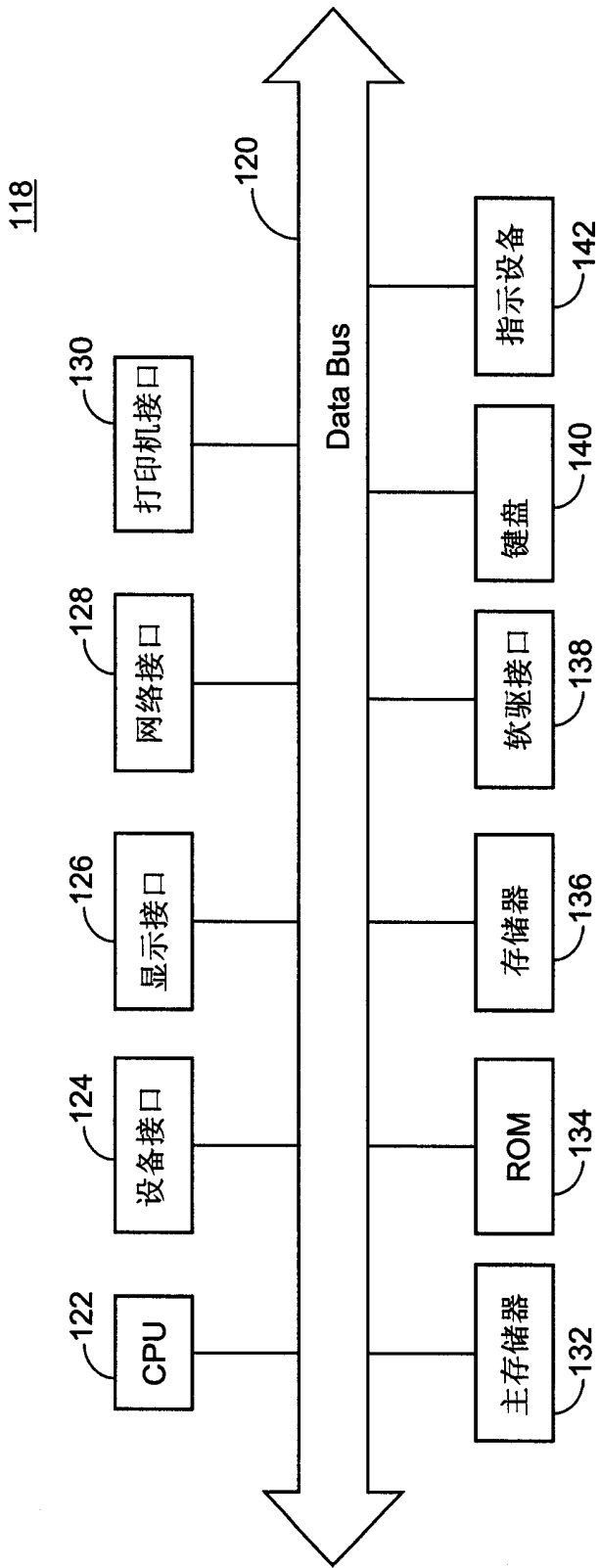


图1B

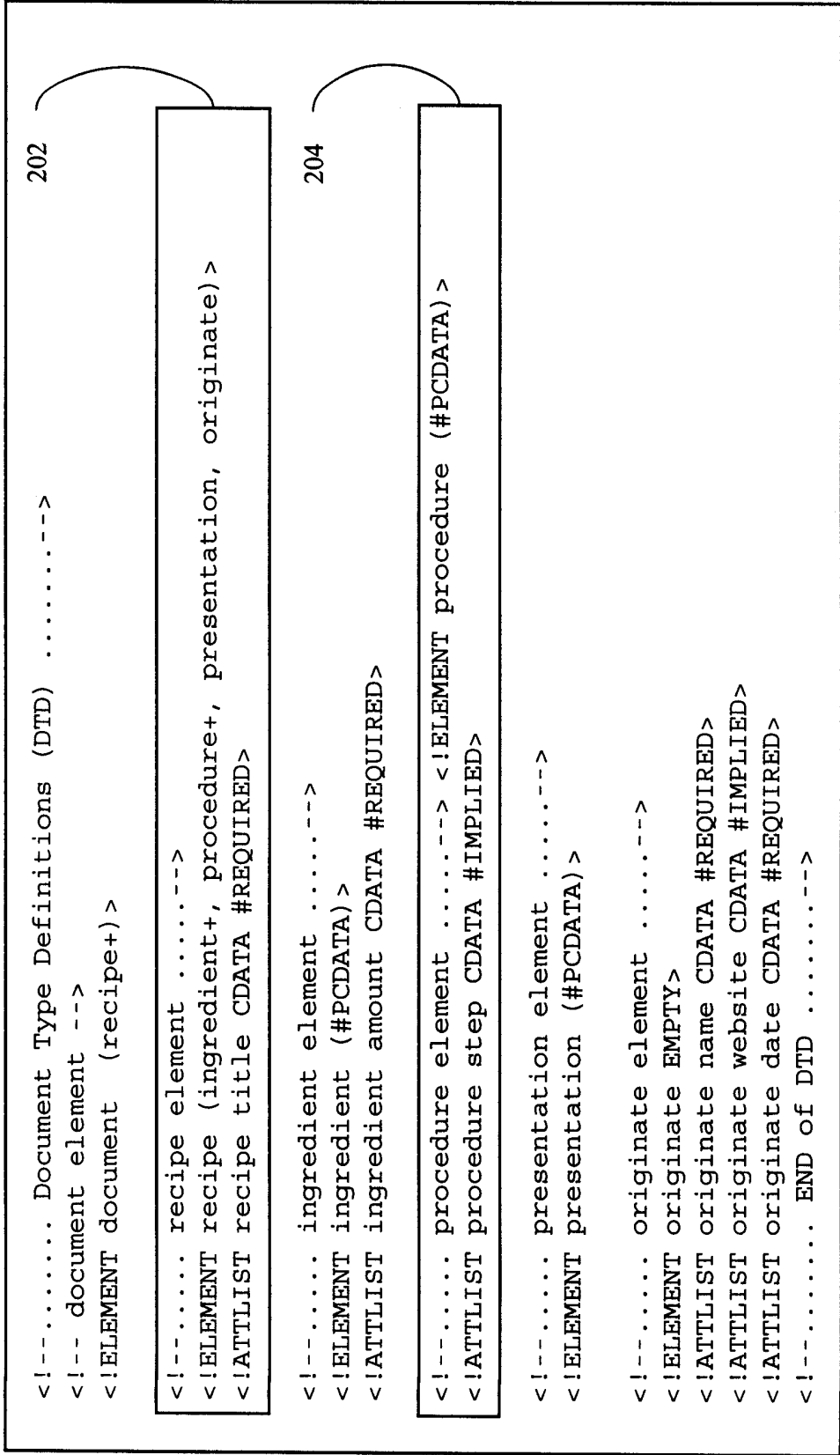


图 2A

200

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="untitled1.xsl"?>
<document>
  <recipe>
    <title>Green Chili Salsa</title>
    <ingredient>Chopped Tomatoes<amount>1 16-ounce Can</amount></ingredient>
    <ingredient>Chopped Green Chile<amount>3 Tablespoons</amount></ingredient>
    <ingredient>Diced Onions<amount>3 Tablespoons</amount></ingredient>
    <ingredient>Chopped Garlic<amount>1 Tablespoon</amount></ingredient>
    <ingredient>Salt<amount>¼ Teaspoon</amount></ingredient>
    <ingredient>Lime Juice<amount>½ Lime</amount></ingredient>
    <ingredient>Chopped Cilantro<amount>1 Tablespoon</amount></ingredient>
    <procedure>In a medium bowl, mix together all ingredients.</procedure>
    <presentation>Use as a dip, or over grilled vegetables,
    or toss with bowtie noodles for an appetizer course.</presentation>
    <originate name = "VeggieLife"><website>www.VeggieLife.com</website>
    <date>November 1997</date></originate>
  </recipe>
</document>

```

图 2B

206

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/"><xsl:apply-templates/></xsl:template>

<xsl:template match="document"><html><body><xsl:apply-templates/></body></html></xsl:template>

<xsl:template match="recipe"><h1><center><font color="teal">Title: </font><font color="red">

<xsl:value-of select="title"/>

</font></center></h1><h2><center><font color="teal">From: </font>
<font color="gray"><xsl:for-each select="originate" ><xsl:value-of select="@name" /></xsl:for-each>
</font></center></h2><p><center><table bgcolor="ffffc0"><tr><td><font color="teal" size="5">
Ingredient: </font></td></tr><xsl:for-each select="ingredient">
<tr><td><xsl:value-of select="text()" /></td>
<td><font color="blue"><xsl:value-of select="amount" /></font></td></tr>
</xsl:for-each></table></center></p><p><center><table><tr><td><font color="teal" size="5">
Procedure: </font></td></tr><xsl:for-each select="procedure"><tr><td><xsl:value-of select="text()" />
</td></tr></xsl:for-each></table></tr><td><font color="teal" size="5">Presentation: </font>
</td></tr><xsl:for-each select="presentation"><tr><td><xsl:value-of select="text()" /></td></tr>
</xsl:for-each></table></center></p><p><center><table bgcolor="c0ffff" width="80%">
<xsl:for-each select="originate"><tr><td align="left"><xsl:value-of select="website" /></td>
<td align="right"><xsl:value-of select="date" /></td></tr></xsl:for-each></table></center></p>
</xsl:template>

</xsl:stylesheet>

```

302

304

300



3A

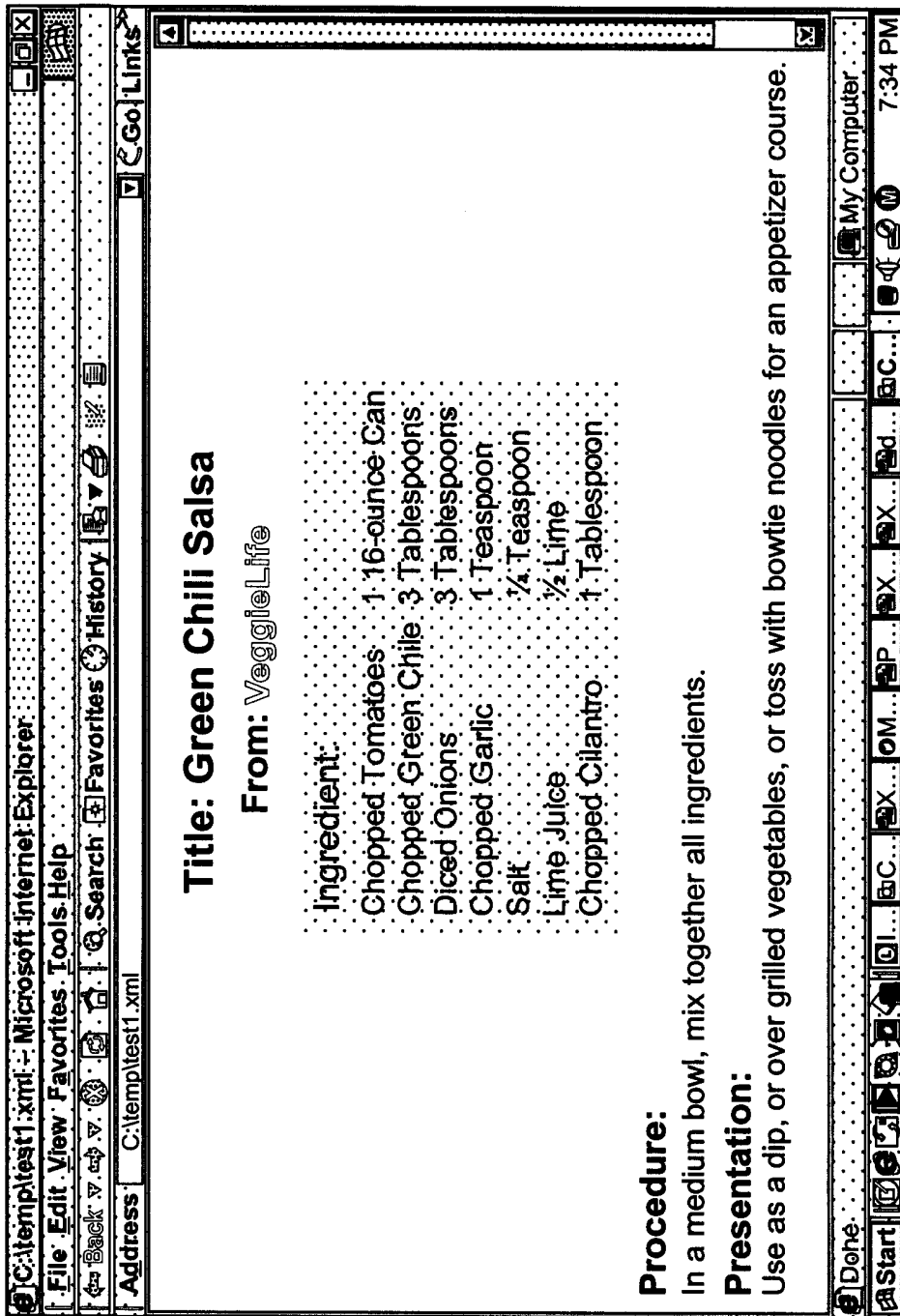


图 3B

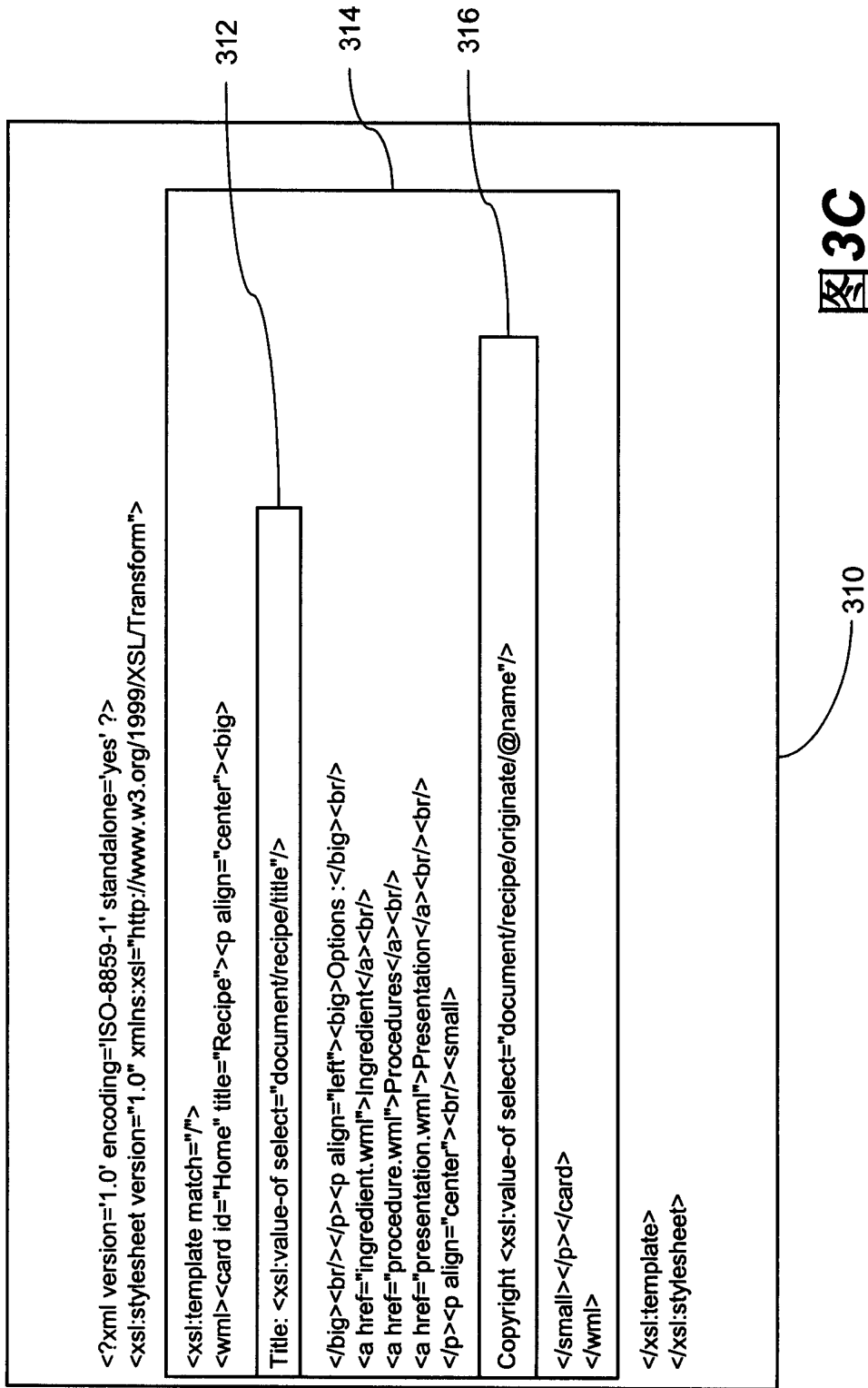
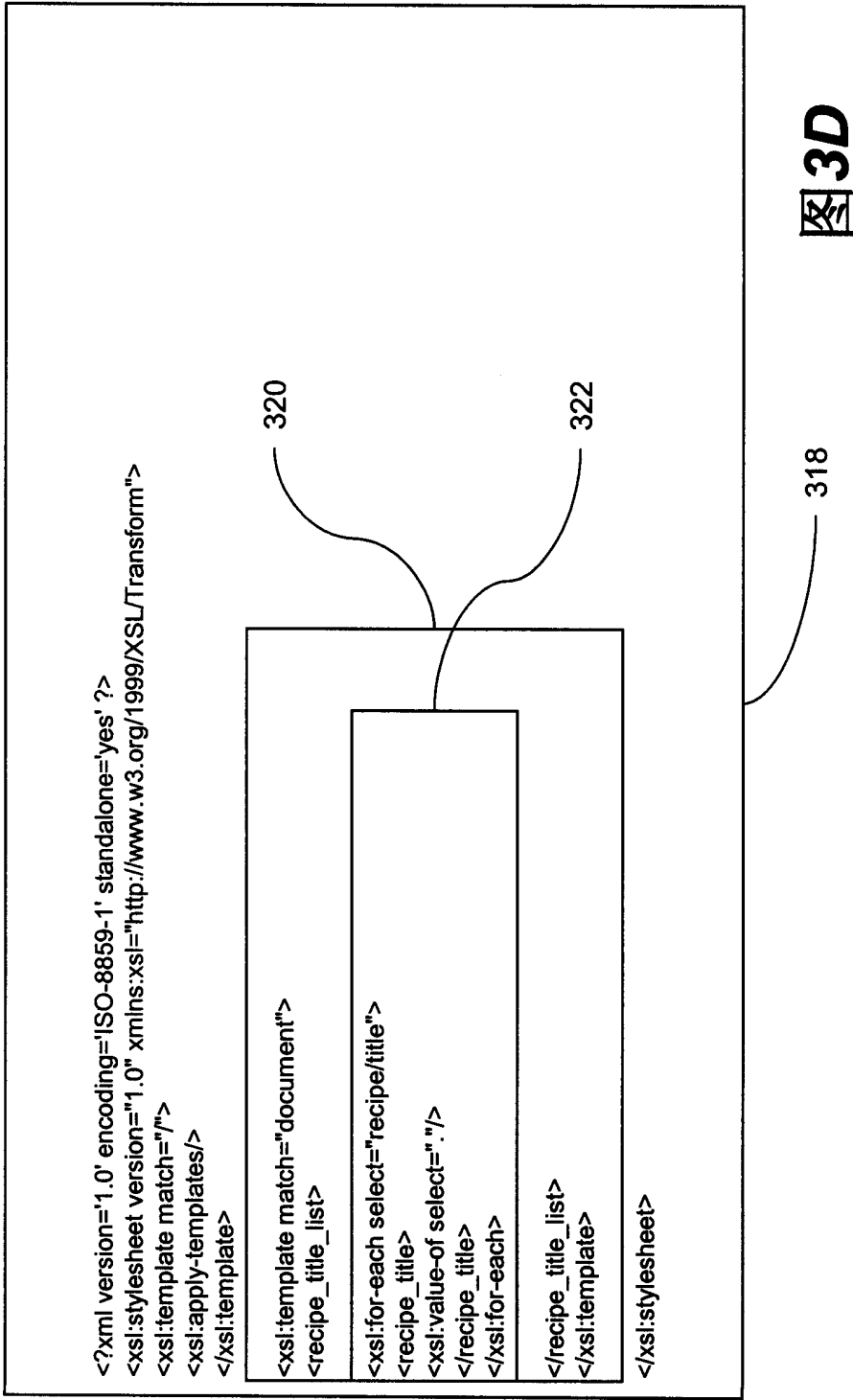


图 3C



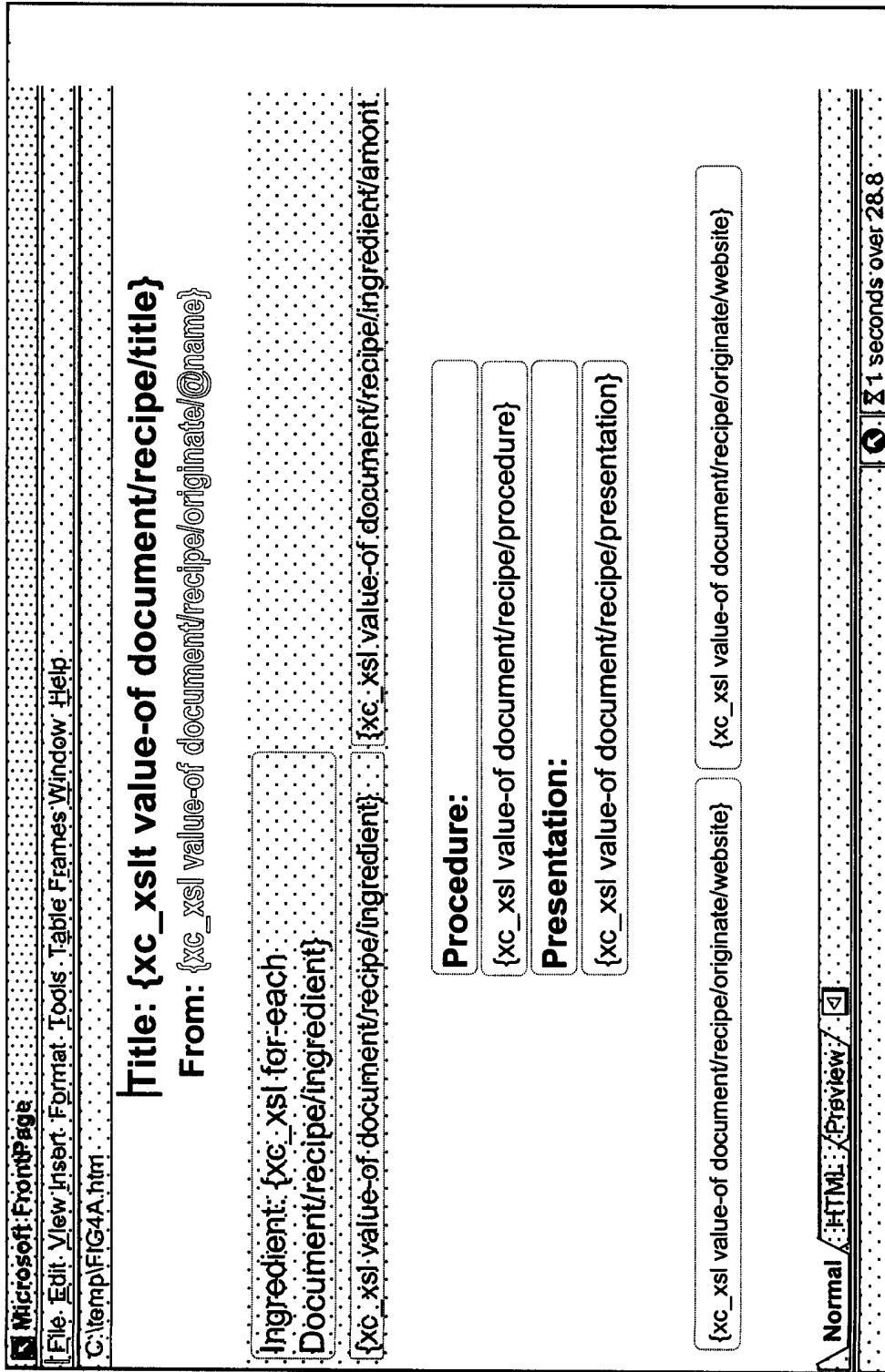


图 4A

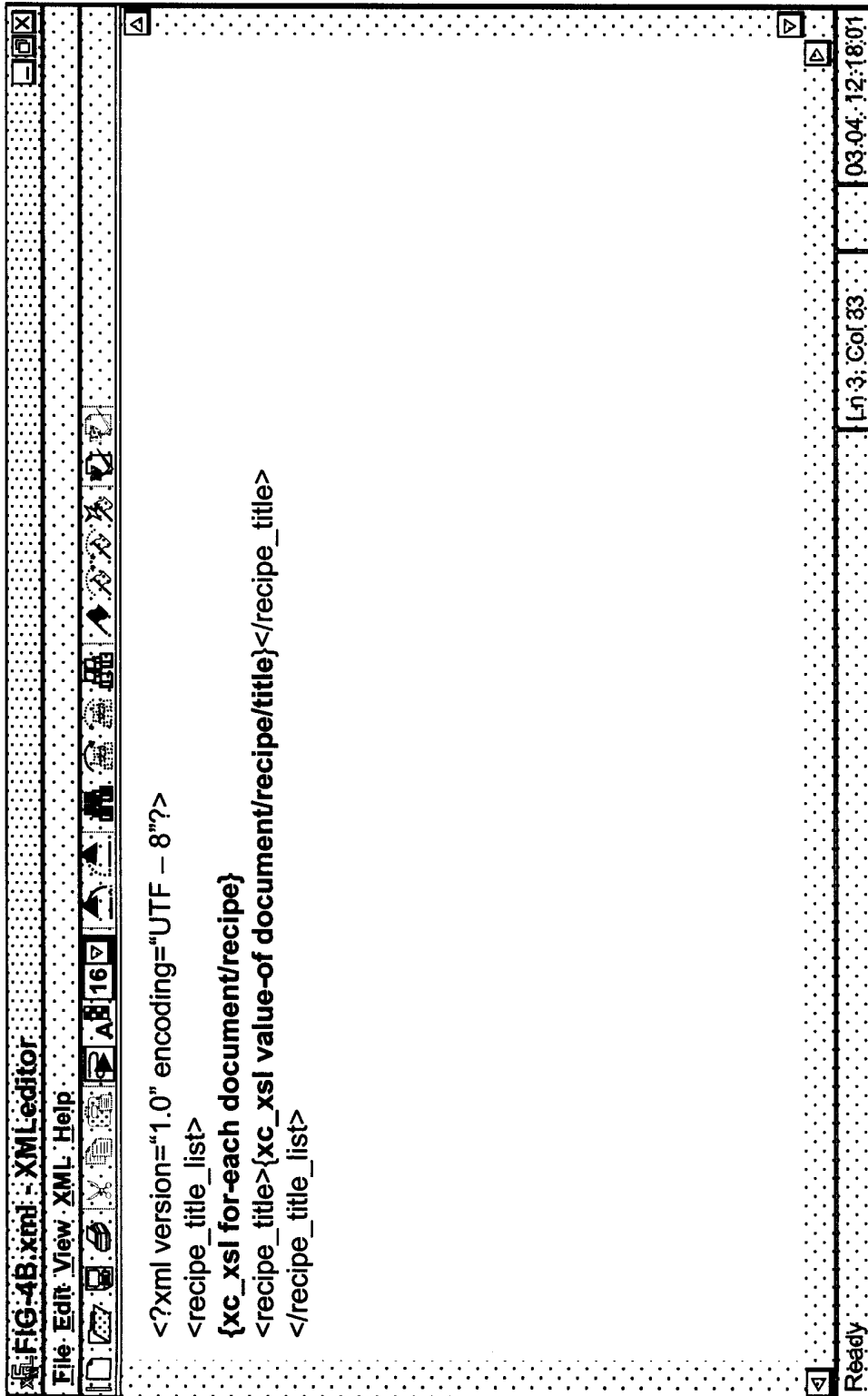
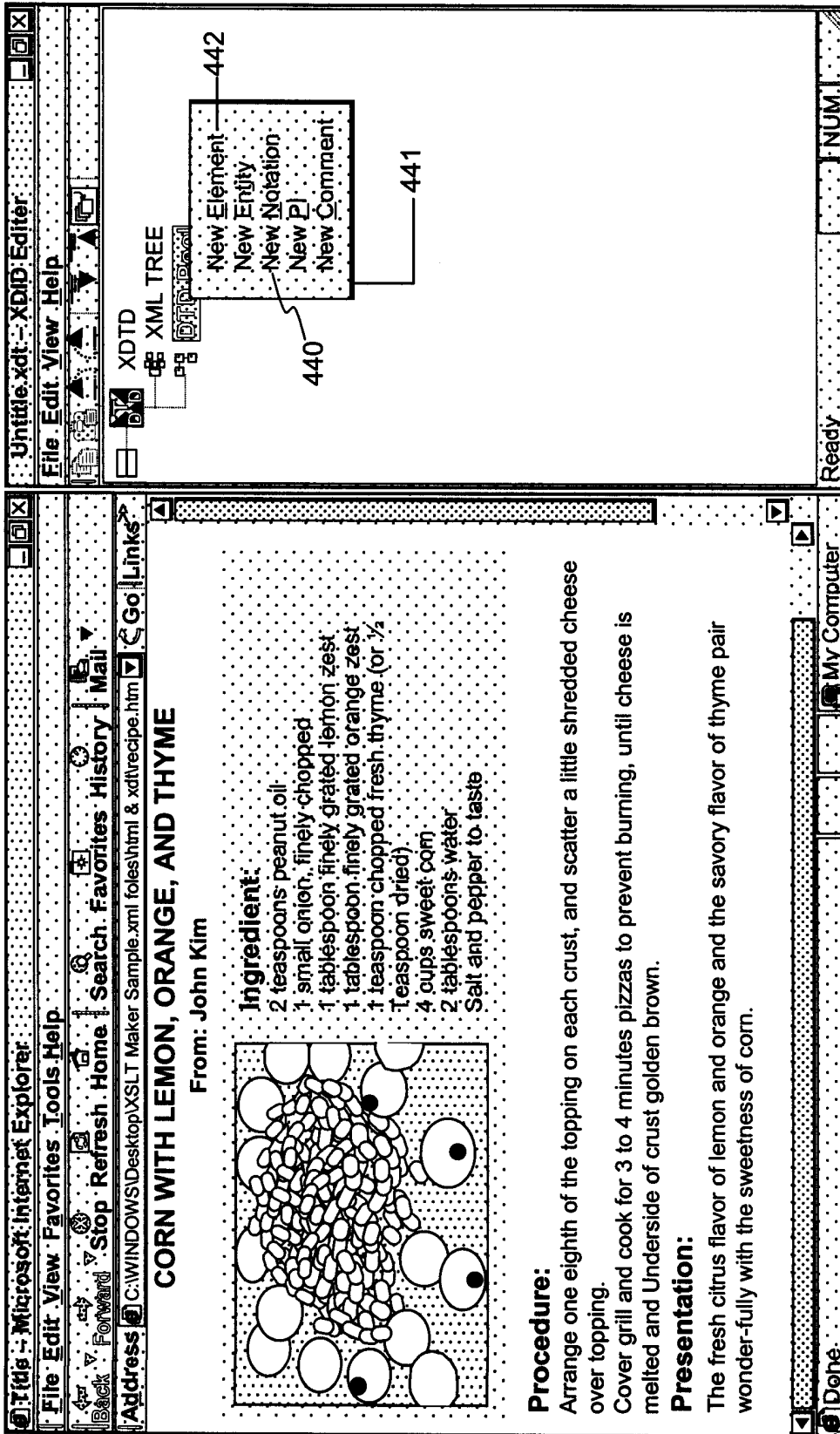


图 4B




```

recipe.xml - Notepad
File Edit View Help
<?xml version=' encoding=' ISO-8859-1' ?>
<document><recipe><title>CORN WITH LEMON, ORANGE, AND THYME</title><image
img="image\recipe.jpg"/><ingredient>2 teaspoons peanut
oil</ingredient><ingredient><ingredient>1 small onion, finely
chopped</amount></ingredient><ingredient><ingredient>1 tablespoon finely grated lemon
zest</amount></ingredient><ingredient><ingredient>1 tablespoon finely grated orange
zest</amount></ingredient><ingredient><ingredient>1 teaspoon chopped fresh thyme (or ½
teaspoon dried)</amount></ingredient><ingredient><ingredient>4 cups sweet
corn</amount></ingredient><ingredient><ingredient>2 tablespoons
water</amount></ingredient><ingredient><ingredient>Salt and pepper to
taste</amount></ingredient><procedure>Arrange one eighth of the topping on
each crust, and scatter a little shredded cheese over topping. Cover grill
and cook for 3 to 4 minutes, rotation pizzas to prevent burning, until
cheese is melted and underside of crust is golden
brown,</procedure><presentation>The fresh citrus flavor of lemon and orange,
and the savory flavor of thyme pair wonder- fully with the sweetness of
corn.</presentation><originate
name="VeggieLife"><website>www.VeggieLife.com</website><data>March
1998</data></originate></recipe></document>

```

图 5C


```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Title</title>
</head>
<body>
<h1 align="center"><center><font color="teal">Title: </font></center><font color="red">{xc_xslt
value-of document/recipe/title}</font></h1>
<h2><center><font color="teal">From: </font><font color="gray">{xc_xsl value-of
document/recipe/originate/@name}</font></center></h2>
<center>
<table bgColor="#ffffc0" width="756">
<tbody>
<tr>
<td width="383"><font color="teal" size="5">Ingredient: {xc_xsl for-each
document/recipe/ingredient}</font></td>
</tr>
<tr>
<td width="383">{xc_xsl value-of document/recipe/ingredient}</td>
<td width="359"><font color="blue">{xc_xsl value-of
document/recipe/ingredient/amont}</font></td>
</tr>
</tbody>
</table>
</center>
<p><center>
<table>
<tbody>
<tr>
<td><font color="teal" size="5">Procedure:</font></td>
</tr>
<tr>
<td>{xc_xsl value-of doucment/recipe/procedure}</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><font color="teal" size="5">Presentation:</font></td>
</tr>
<tr>
<td>{xc_xsl value-of document/recipe/presentation}</td>
</tr>
</tbody>
</table>
</center>
<p><center>
<table width="663" bgColor="#c0ffff">
<tbody>
<tr>
<td align="left" width="327">{xc_xsl value-of document/recipe/originate/
website}</td>
<td align="right" width="322">{xc_xsl value-of document/recipe/originate/
date}</td>
</tr>
</tbody>
</table>
</center>
</body>
</html>

```

图6A

```

<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Title</title>
</head>

<body>

<h1 align="center"><center><font color="teal">Title: </font></center><font color="red"><i>Green
Chili Salsa</i></font></h1>
<h2><center><font color="teal">From: </font><font color="gray"><i>VeggieLife</i></font></center></h2>
<center>
<table bgColor="#ffffc0" width="756">
<tbody>
<tr>
<td width="383"><font color="teal" size="5">Ingredient: {xc_xsl for-each}</font></td>
</tr>
<tr>
<td width="383"><i>Chopped Tomatoes</i></td>
<td width="359"><font color="blue"><i>1 16-ounce can</i></font></td>
</tr>
</tbody>
</table>
</center>
<p><center>
<table>
<tbody>
<tr>
<td><font color="teal" size="5">Procedure:</font></td>
</tr>
<tr>
<td><i>In a medium bowl, mix together all ingredients.</i></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><font color="teal" size="5">Presentation:</font></td>
</tr>
<tr>
<td><i>Use as a dip, or over grilled vegetables, or toss with bowtie noodles for an appetizer
course</i>.</td>
</tr>
</tbody>
</table>
</center>
<p><center>
<table width="663" bgColor="#c0ffff">
<tbody>
<tr>
<td align="left" width="327"><i>www.VeggieLife.com</i></td>
<td align="right" width="322"><i>November 1997</i></td>
</tr>
</tbody>
</table>
</center>


</body>

</html>

```

图 6B

Green Chili Salsa	{xc_xslt value-of document/recipe/title}
VeggieLife	{xc_xsl value-of document/recipe/originate/@name}
Chopped Tomatoes	{xc_xsl value-of document/recipe/ingredient}
1 16-ounce can	{xc_xsl_value-of document/recipe/ingredient/amount}
In a medium bowl, mix together all ingredients.	{xc_xsl value-of document/recipe/procedure}
Use as a dip, or over grilled vegetables, or toss with bowtie noodles for an appetizer course.	{xc_xsl value-of document/recipe/presentation}
www.VeggieLife.com	{xc_xsl value-of document/recipe/originate/ website}
November 1997	{xc_xsl value-of document/recipe/originate/ date}

 6C

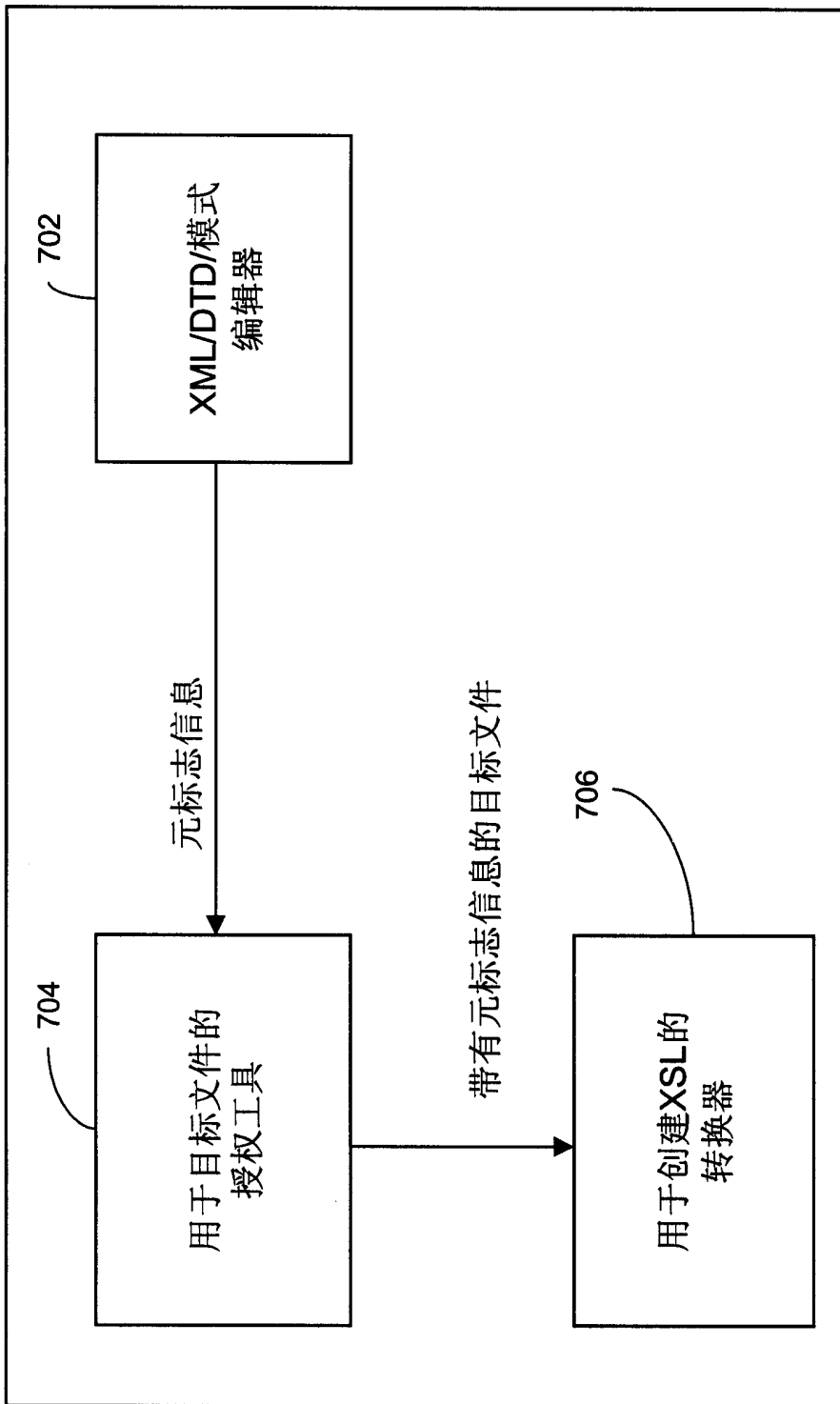


图7A

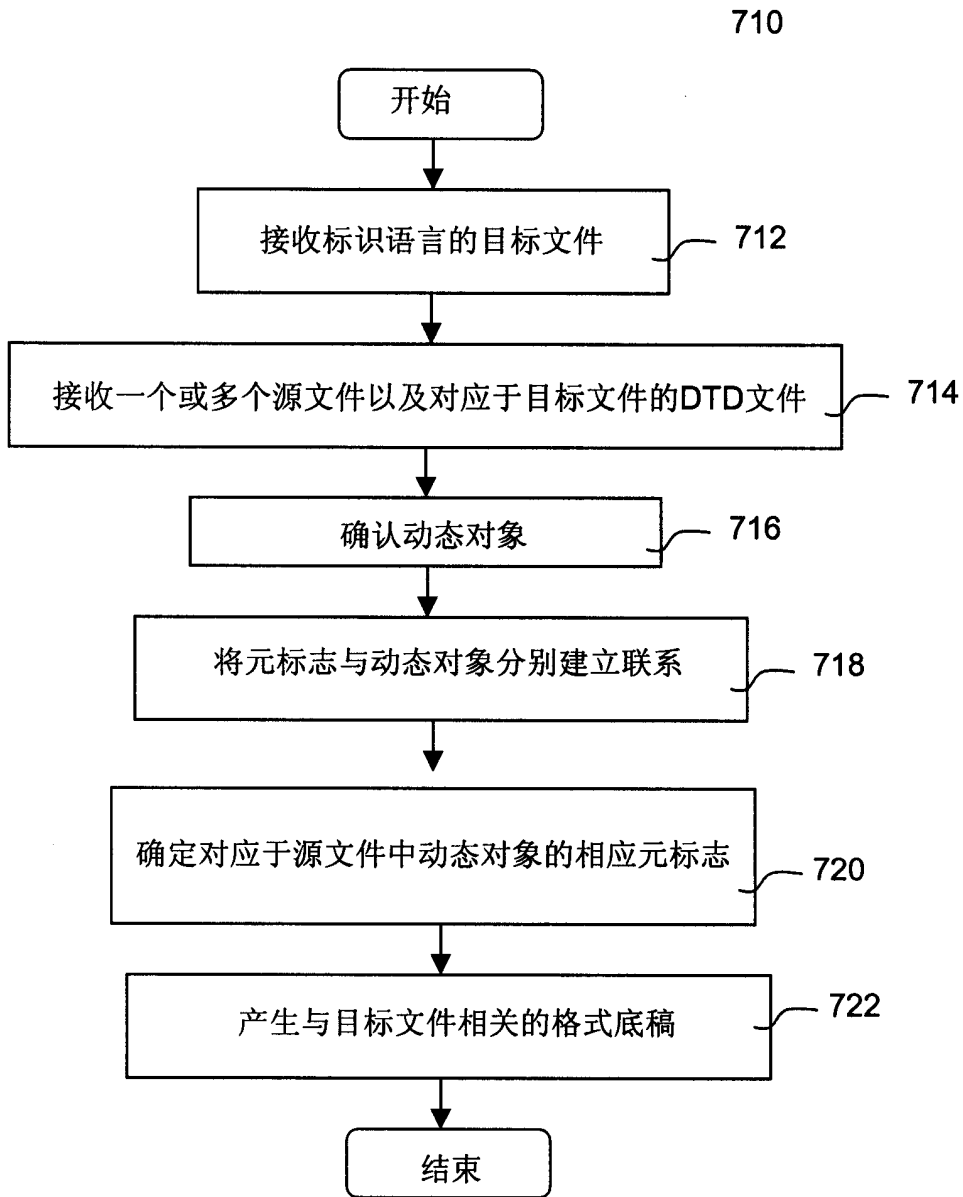


图7B

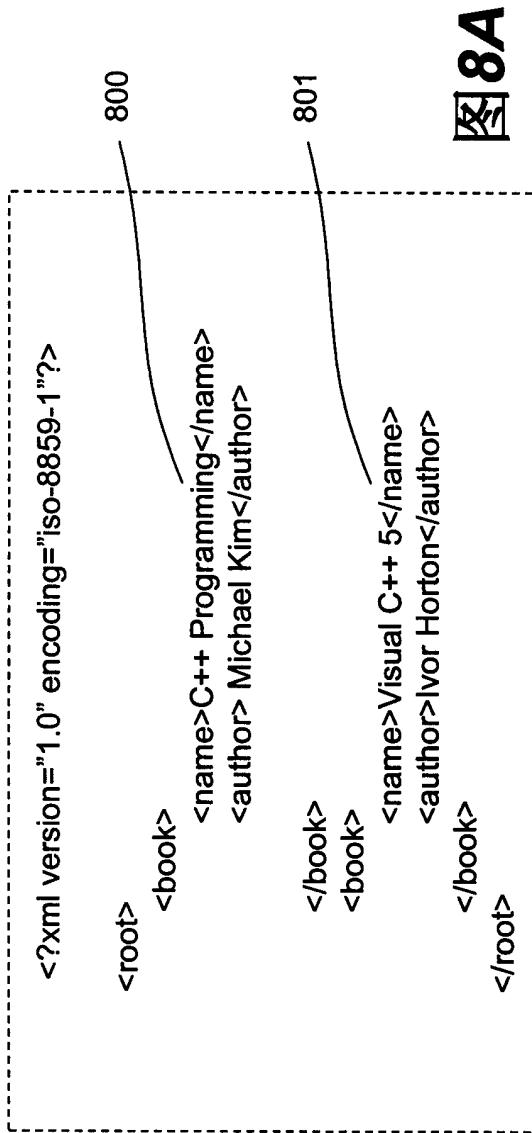


图 8A

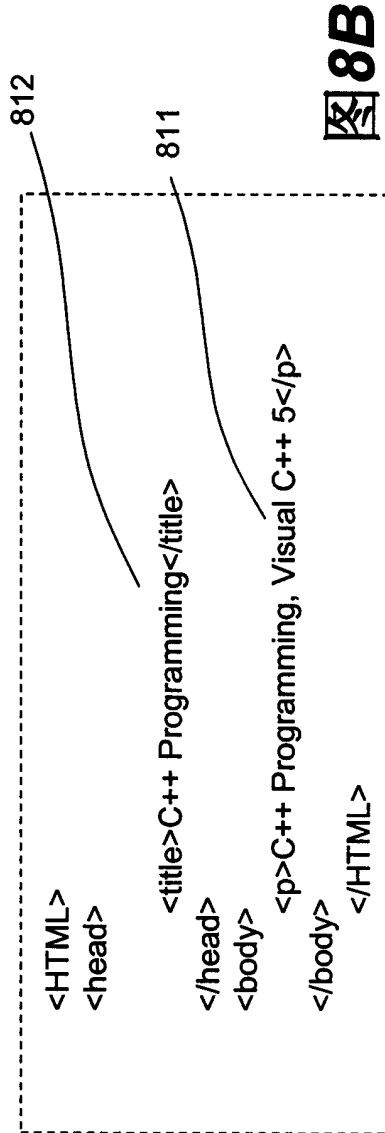


图 8B

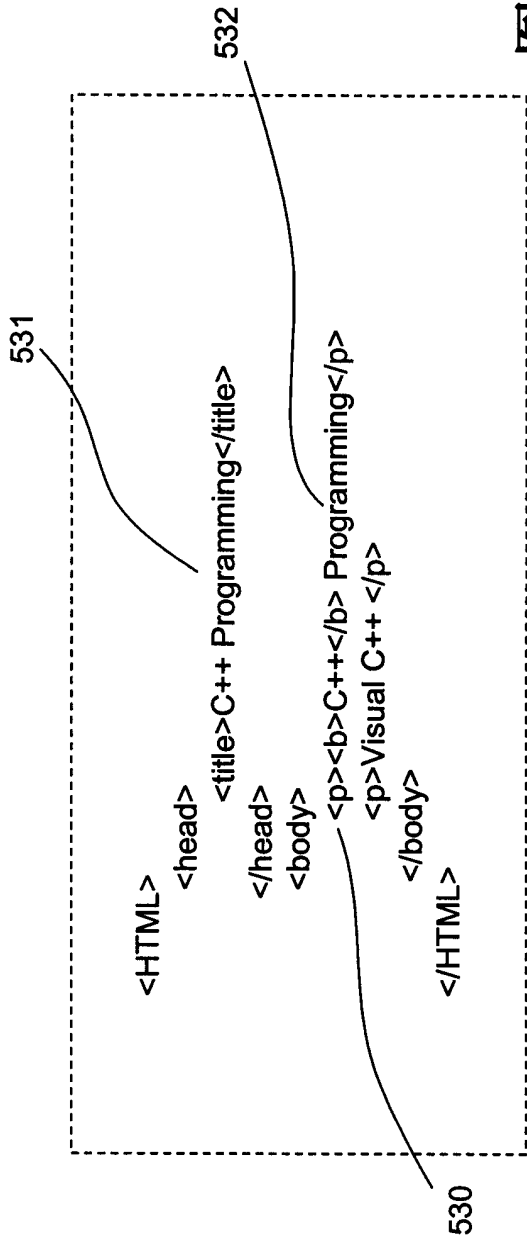


图 8C

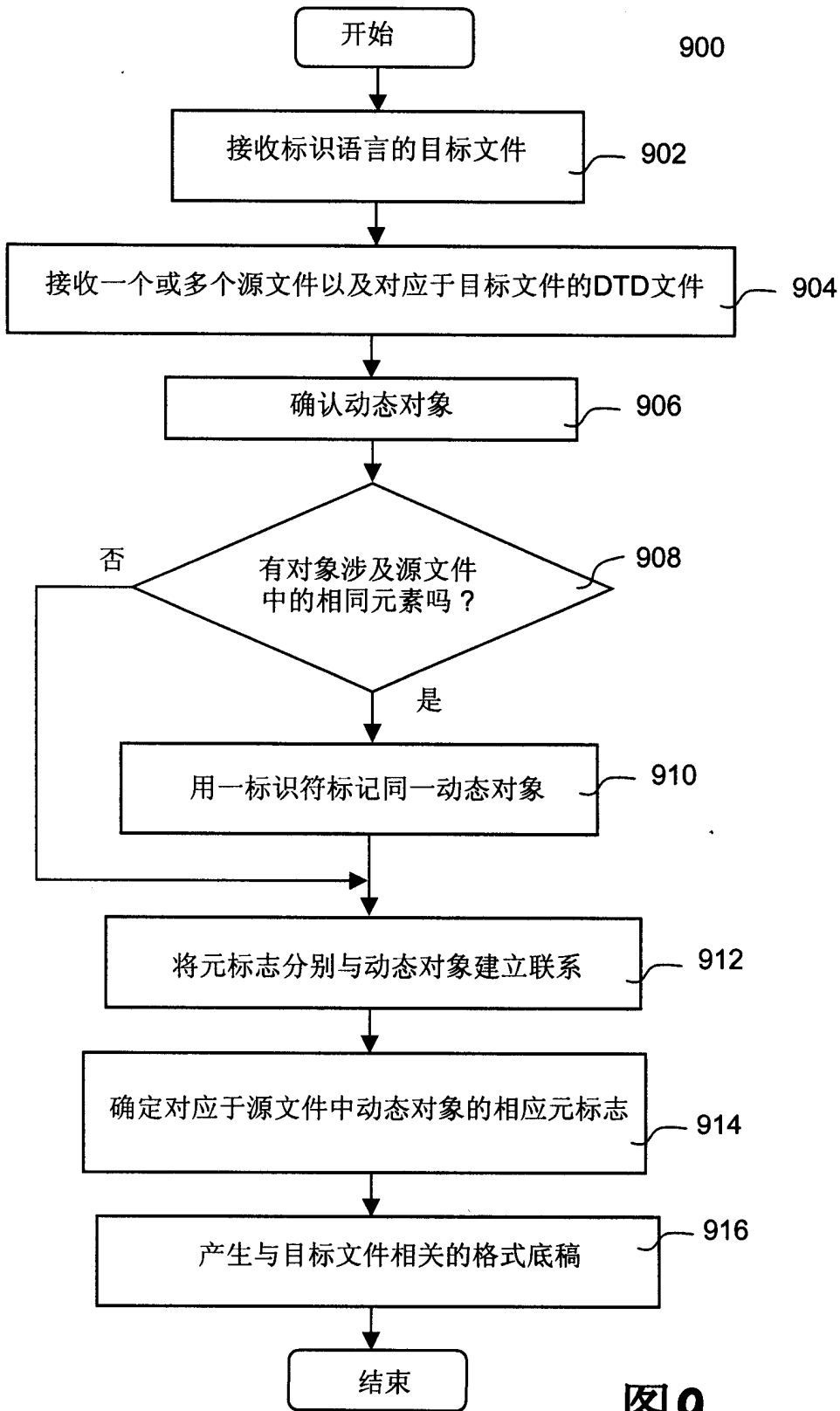


图9