

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 17/24 (2006.01)

G06F 17/21 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200580021155.2

[43] 公开日 2007年5月30日

[11] 公开号 CN 1973285A

[22] 申请日 2005.6.24

[21] 申请号 200580021155.2

[30] 优先权

[32] 2004.6.24 [33] JP [31] 186123/2004

[86] 国际申请 PCT/JP2005/011667 2005.6.24

[87] 国际公布 WO2006/001392 日 2006.1.5

[85] 进入国家阶段日期 2006.12.25

[71] 申请人 佳思腾软件公司

地址 日本德岛县

[72] 发明人 叶俊信 青木淳平

[74] 专利代理机构 北京英赛嘉华知识产权代理有限公司

代理人 余 滕 方 挺

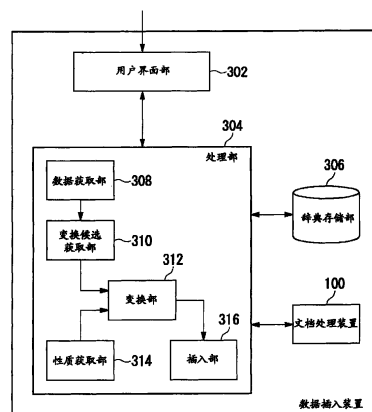
权利要求书 3 页 说明书 25 页 附图 23 页

[54] 发明名称

文档处理方法及其装置

[57] 摘要

在用户将数据插入以标记语言描述的文档中时，可以使处理简化。数据获取部(308)通过用户界面部(302)接收应插入文档中的数据。变换候选获取部(310)根据辞典存储部(306)中存储的辞典对所接收的数据进行变换。进而，将所变换的数据分别显示在屏幕中。性质获取部(314)接收处理对象文档中应插入数据位置上的处理对象词汇或处理对象架构。变换部(312)接收来自用户的从显示在屏幕上的多种数据中选择其一的指令。进而，根据处理对象词汇或处理对象架构，判断用户选择的数据是否可以插入处理对象文档中，如果可以插入，则确定该数据的插入。



1. 文档处理装置，其特征在于，包括：
第一接收部，接收应插入到以标记语言描述的文档中的数据；
第二接收部，接收描述所述文档中应插入数据位置上的所述文档的标记语言的性质；
变换部，根据由所述第二接收部接收的性质，对由所述第一接收部接收的数据进行变换；以及
插入部，将变换后的数据插入所述文档中。

2. 根据权利要求1所述的文档处理装置，其中由所述第二接收部接收的描述所述文档的标记语言的性质为属于标记语言的种类，
所述变换部变换由所述第一接收部接收的数据，以使其与由所述第二接收部接收的标记语言的种类对应。

3. 根据权利要求2所述的文档处理装置，其中在由所述第一接收部接收的数据中预先附加有预定的标签，
所述变换部修改附加在由所述第一接收部接收的数据中的标签，以使其与由所述第二接收部接收的标记语言的种类对应。

4. 根据权利要求2所述的文档处理装置，其中所述第二接收部包括：
关系接收部，接收描述所述文档的标记语言的种类与显示所述文档时应使用的标记语言种类间的关系；
位置接收部，接收以显示所述文档时应使用的标记语言描述的显示用的文档中应插入数据的位置；
位置确定部，根据由所述关系接收部接收的关系，并根据由所述位置接收部接收的显示用的文档中应插入数据的位置，确定所述文档中应插入数据的位置；以及
种类确定部，确定描述所述文档中应插入数据位置上的所述文档

的标记语言的种类。

5. 根据权利要求 2 或 3 所述的文档处理装置，其中由所述第二接收部接收的描述所述文档的标记语言的性质进一步包括包含在所述标记语言种类中的构成元素的规则，

所述变换部变换由所述第一接收部接收的数据，以使其与包含在由所述第二接收部接收的标记语言种类中的构成元素的规则对应。

6. 根据权利要求 5 所述的文档处理装置，其中在由所述第一接收部接收的数据中预先附加有预定的标签，

所述变换部修改附加在由所述第一接收部接收的数据中的标签，以使其与包含在由所述第二接收部接收的标记语言的种类中的构成元素的规则对应。

7. 根据权利要求 2 所述的文档处理装置，其中在由所述第一接收部接收的数据中预先附加有预定的标签，

所述变换部根据附加在由所述第一接收部接收的数据中的标签、以及由所述第二接收部接收的标记语言的种类，确定包含在标记语言种类中的构成元素的规则，并修改附加在由所述第一接收部接收的数据中的标签，以使其与包含在已确定的标记语言种类中的构成元素的规则对应。

8. 根据权利要求 1 至 7 任一项所述的文档处理装置，其中所述变换部根据由所述第二接收部接收的性质，在不能变换由所述第一接收部接收的数据的情况下不进行变换。

9. 文档处理方法，其特征在于，接收应插入以标记语言描述的文档中的数据、和接收描述所述文档中应插入数据位置的所述文档的标记语言的性质，根据所接收的性质，对数据进行变换然后将其插入所述文档中。

10. 计算机程序，其特征在于，所述计算机程序使计算机实现以下步骤：

接收应插入以标记语言描述的文档中的数据并将其存储到存储器；

接收描述所述文档中应插入数据位置的所述文档的标记语言的性质并将其存储到存储器；

根据在接收应插入以标记语言描述的文档中的数据并将其存储到存储器的步骤中存储的性质，对在接收所述数据并将其存储到存储器的步骤中所接收的数据进行变换；以及

将变换的数据插入所述文档中。

文档处理方法及其装置

技术领域

本发明涉及一种文档处理技术，特别是涉及一种对以标记语言描述的文档进行处理的文档处理方法及其装置。

背景技术

XML 作为适用于通过网络等与他人共享数据的格式受到人们的关注，这也促使人们开发用于创建、显示和编辑 XML 文档的应用软件（例如，请参考专利文献 1）。XML 文档由文档格式定义等定义的词汇（标签集）创建。

专利文献 1：特开 2001-290804 号公报

发明内容

发明要解决的课题

在将字符串等数据嵌入 XML 文档中的情况下，如果考虑到用户的便利性，则人们需要一种易于嵌入的数据的方法。

本发明是鉴于这种状况而做出的，其目的在于，提供一种在对以标记语言描述的文档中插入数据时可以提高用户便利性的文档处理方法及其装置。

解决课题的方案

为了解决上述课题，本发明的一个方面的文档处理装置包括：

第一接收部，接收应插入到以标记语言描述的文档中的数据；第二接收部，接收描述文档中应插入数据位置中的文档的标记语言的性质；变换部，根据由第二接收部接收的性质，对由第一接收部接收的数据进行变换；插入部，将变换的数据插入所述文档中。

“标记语言”可以是 XML 的一种方式，例如，XHTML、SVG、

MathML 等，也可以是 SGML、HTML 等。术语“标记语言的性质”是指确定描述文档的标记语言的项目，例如，标记语言的种类，以及包含在标记语言中的构成元素的规则等。

根据此实施方式，由于在将数据插入文档时，可以考虑描述该文档的标记语言的性质对数据进行变换，然后将其插入文档中，因此，即使用户没有意识到标记语言的性质也可以插入数据。

由第二接收部接收的描述文档的标记语言的性质可以是标记语言的种类。变换部可以变换由第一接收部接收的数据，以使其与由第二接收部接收的标记语言的种类对应。在由第一接收部接收的数据中可以预先附加有预定的标签，变换部可以修改附加在由第一接收部接收的数据中的标签，以使其与由第二接收部接收的标记语言的种类对应。由于可以根据标记语言的种类对附加在数据中的标签进行修改，然后将其插入文档中，因此，即使用户没有意识到标记语言的性质也可以插入数据。

第二接收部可以包括：关系接收部，接收描述文档的标记语言的种类与显示文档时应使用的标记语言种类之间的关系；位置接收部，接收由显示文档时应使用的标记语言描述的显示用的文档中应插入数据的位置；位置确定部，根据由关系接收部接收的关系，并根据由位置接收部接收的显示用的文档中应插入数据的位置，确定文档中应插入数据的位置；种类确定部，确定描述文档中应插入数据位置中的文档的标记语言的种类。由于根据描述文档的标记语言的种类和用于显示所述文档时应使用的标记语言种类之间的关系，可以确定应插入数据的文档的标记语言的种类，因此，即使该文档的标记语言的种类不明确也可以确定标记语言的种类。

由第二接收部接收的描述文档的标记语言的性质可以进一步包括包含在标记语言的种类中的构成元素的规则。变换部可以变换由第一接收部接收的数据，以使其与包含在由第二接收部接收的标记语言种类中的构成元素的规则对应。在由第一接收部接收的数据中可以预先附加有预定的标签，变换部可以修改附加在由第一接收部接收的数据中的标签，以使其与包含在第二接收部接收的标记语言的种类中的

构成元素的规则对应。

“包含在标记语言中的构成元素的规则”可以包括图表或 DTD (Document Type Definition: 文件类型定义), 也可以是所谓的 XML 文档的说明书。由于可以根据标记语言的种类对附加在数据中的标签进行修改, 然后将其插入文档中, 因此, 即使用户没有意识到标记语言的性质也可以插入数据。

在由第一接收部接收的数据中可以预先附加有预定的标签, 变换部可以从附加在由第一接收部接收的数据中的标签、和包含在由第二接收部接收的标记语言的种类中确定包含在标记语言种类中的构成元素的规则, 并可以修改附加在由第一接收部接收的数据中的标签, 以使其与包含在已确定的标记语言种类中的构成元素的规则对应。由于可以从标记语言的种类和附加在数据中的标签中确定包含在标记语言中的构成元素的规则, 因此, 即使没有接收包含在标记语言中的构成元素的规则也可以插入数据。

变换部可以根据由第二接收部接收的性质, 在不能变换由第一接收部接收的数据的情况下不进行变换。由于不进行变换, 从而可以避免错误。

本发明的另一方式为文档处理方法, 在该方法中, 接收应插入以标记语言描述的文档中的数据、和描述文档中应插入数据位置的文档的标记语言的性质, 根据所接收的性质, 对数据进行变换然后将其插入所述文档中。

另外, 作为本发明的实施方式, 以上构成元素的任意组合、以及将本发明的描述在方法、装置、系统等之间进行变换的方式也是有效的。

发明效果

根据本发明, 在对以标记语言描述的文档中插入数据时可以提高用户便利性。

附图说明

图 1 所示的是前提技术所涉及的文档处理装置的结构示意图；

图 2 所示的是由文档处理装置所编辑的 XML 文档的例子的示意图；

图 3 所示的是将图 2 所示的 XML 文档映射为以 HTML 记述的表的例子的示意图；

图 4a 所示的是用于将图 2 所示的 XML 文档映射为图 3 所示的表的定义文件的例子的示意图；

图 4b 所示的是用于将图 2 所示的 XML 文档映射为图 3 所示的表的定义文件的例子的示意图；

图 5 所示的是将图 2 所示的 XML 文档通过图 3 所示的对应关系映射为 HTML 而显示的屏幕的例子的示意图；

图 6 所示的是为使用户生成定义文件而将定义文件生成部提供给用户的图形用户界面的例子的示意图；

图 7 所示的是由定义文件生成部生成的屏幕布局的其他例子的示意图；

图 8 所示的是文档处理装置编辑的 XML 文档的编辑屏幕的一个例子的示意图；

图 9 所示的是文档处理装置编辑的 XML 文档的编辑屏幕的另一个例子的示意图；

图 10 所示的是显示图 9 所示的文档的屏幕的例子的示意图；

图 11 所示的是实施例 1 所涉及的数据插入装置的结构示意图；

图 12a - 12c 所示的是由图 11 的数据插入装置插入的数据的例子的示意图；

图 13 所示的是图 11 的数据插入装置进行数据插入处理的程序的流程图；

图 14 所示的是图 11 的数据插入装置进行数据插入处理的程序另一个例子的流程图；

图 15a - 15d 所示的是由图 11 的数据插入装置插入的数据的例子的示意图；

图 16 所示的是图 11 的数据插入装置进行数据插入处理的程序的

另一个例子的流程图；

图 17 所示的是实施例 1 所涉及的数据插入装置的结构的一个例子的示意图；

图 18a - 18d 所示的是由图 17 的数据插入装置插入的数据的例子流程图；

图 19a - 19e 所示的是由图 17 的数据插入装置插入的数据的另一个例子的示意图；

图 20 所示的是实施例 2 所涉及的数据插入装置的结构示意图；

图 21 所示的是图 20 的数据插入装置进行数据获取程序的流程图；以及

图 22a - 22d 所示的是由图 20 的数据插入装置插入的数据的例子示意图。

符号说明

20 文档处理装置	22 主控单元	24 编辑单元
30 DOM 单元	32 DOM 提供部	34 DOM 生成部
36 输出部	40 CSS 单元	42 CSS 分析部
44 CSS 提供部	46 呈现部	50 HTML 单元
52, 62 控制部	54, 64 编辑部	56, 66 显示部
60 SVG 单元	72 文档获取部	
74 命名空间 URI 获取部	76 定义文件名生成部	80 VC 单元
82 映射部	84 定义文件获取部	86 定义文件生成部
300 数据插入装置	302 用户界面部	304 处理部
306 辞典存储部	308 数据获取部	310 变换候选获取部
312 变换部	314 性质获取部	316 插入部
320 位置确定部	322 词汇获取部	

具体实施方式

首先对作为本发明的前提的技术进行说明，然后对本发明的实施例进行说明。

(前提技术)

图 1 所示的是前提技术所涉及的文档处理装置 20 的结构。文档处理装置 20 对结构化的文档进行处理, 该文档中的数据被分为具有分级结构的多个构成元素。在本前提技术中, 对处理作为结构化文档一个例子的 XML 文档的例子进行说明。文档处理装置 20 包括: 主控单元 22、编辑单元 24、DOM 单元 30、CSS 单元 40、HTML 单元 50、SVG 单元 60 以及用作变换部一个示例的 VC 单元 80。就硬件部件而言, 这些结构可由任意的计算机的 CPU 或存储器、存储器载入的程序等来实现, 但是在此描述为通过它们的协作而实现的功能模块。因此, 本领域技术人员能够理解, 这些功能模块可仅通过硬件的方式、仅通过软件的方式或通过二者相结合的方式等多种方式来实现。

主控单元 22 提供插件的载入, 和提供执行命令的框架。编辑单元 24 提供用于编辑 XML 文档的框架。文档处理装置 20 中的文档的显示和编辑功能是通过插件来实现的, 根据所处理的文档类型, 必要的插件通过主控单元 22 或编辑单元 24 载入。主控单元 22 或编辑单元 24 通过参考作为处理对象的文档的命名空间来确定 XML 文档由哪个词汇描述, 并且载入与该词汇对应的、用于显示和编辑的插件从而执行显示和编辑。例如, 在文档处理装置 20 中实现了对 HTML 文档进行显示和编辑的 HTML 单元 50, 以及对 SVG 文档进行显示和编辑的 SVG 单元 60, 对于各个词汇(标签集)将显示系统和编辑系统实现为插件, 以分别在对 HTML 文档进行编辑时载入 HTML 单元 50、在对 SVG 文档进行编辑时载入 SVG 单元 60。如以下将描述的那样, 在要对既包括 HTML 又包括 SVG 组件的复合文档进行处理时, HTML 单元 50 和 SVG 单元 60 均被载入。

通过以上结构, 由于用户能够仅选择必要的功能以安装该功能, 如果需要, 也能够在此后的阶段增加或删除适当的功能。因此, 能够有效利用记录介质的存储区域(例如储存程序的硬盘), 并能够避免在执行程序的时候存储器使用的浪费。此外, 由于这一结构有利于功能的扩展, 因此开发主体还能够以插件的形式对应于新的词汇, 因而易于开发, 而且用户也能够通过增加插件而以较低成本轻易地增加功能。

编辑单元 24 通过用户界面接收来自用户的编辑指令的事件, 将事件

通知给适当的插件等并对事件的重做 (redo) 或执行的撤消 (undo) 等处理进行控制。

DOM 单元 30 包括 DOM 提供部 32、DOM 生成部 34 以及输出部 36。DOM 单元 30 实现了与文档对象模型 (Document Object Model: DOM) 相符的功能, 在 XML 文档作为数据被处理时, 所述文档对象模型被定义以提供访问方法。DOM 提供部 32 是满足由编辑单元 24 所定义界面的 DOM 的实现。DOM 生成部 34 从 XML 文档生成 DOM 树。如以下将描述的那样, 当通过 VC 单元 80 将作为处理对象的 XML 文档映射为其它词汇时, 生成与映射源中的 XML 文档相对应的源树以及与映射目标中的 XML 文档相对应的目的树。输出部 36 例如在编辑结束时, 输出作为 XML 文档的 DOM 树。

CSS 单元 40 包括 CSS 分析部 42、CSS 提供部 44 以及呈现部 46, 并提供与 CSS 相符的显示功能。CSS 分析部 42 具有用于分析 CSS 语法的分析功能。CSS 提供部 44 是 CSS 对象的实现, 并执行对 DOM 树的 CSS 层叠处理。呈现部 46 是 CSS 的呈现引擎, 并用来显示以诸如 HTML 等词汇描述的、利用 CSS 布置的文档。

HTML 单元 50 对以 HTML 描述的文档进行显示或编辑。SVG 单元 60 对以 SVG 描述的文档进行显示或编辑。这些显示/编辑系统以插件的形式实现, 分别包括显示部 (Canvas) 56、66, 用以显示文档; 控制部 (Editlet) 52、62, 接收或发送包含有编辑指令的事件; 编辑部 (Zone) 54、64, 接收编辑指令以对 DOM 进行编辑。在控制部 52 或 62 从外部源接收到 DOM 树的编辑命令时, 编辑部 54 或 64 修改 DOM 树, 而显示部 56 或 66 更新显示。这些单元具有与被称作 MVC (Model-View-Controllers, 模型 - 视图 - 控制器) 的框架相类似的结构, 大体上, 显示部 56 及 66 与 “View” 对应, 控制部 52 及 62 与 “Controller” 对应, 编辑部 54 及 64 与 DOM 的实体 “Model” 对应。本前提技术的文档处理装置 20 中, 不仅可以以树的显示格式编辑 XML 文档, 还可以对应于每个词汇进行编辑。例如, HTML 单元 50 提供用于以类似于字符处理的方式编辑 HTML 文档的用户界面, SVG 单元 60 提供用于以类似于图像描绘工具的形式编辑 SVG 文档的用户界面。

VC 单元 80 包括映射部 82、定义文件获取部 84 以及定义文件生成部 86。通过将以某个词汇描述的文档映射为另一词汇，VC 单元 80 提供了一种框架，以通过与被映射的词汇相对应的显示和编辑插件来显示或编辑文档。在本前提技术中，该功能被称为词汇连接 (Vocabulary Connection:VC)。定义文件获取部 84 获取描述了映射定义的定义文件。该定义文件逐个节点地描述了节点间的对应 (连接)。此时，可规定各节点的元素值或属性值是否可以编辑。此外，也可描述使用了节点的元素值或属性值的运算式。这些功能将在稍后进行描述。映射部 82 参考定义文件获取部 84 已经获取的定义文件使 DOM 生成部 34 生成目的树，以管理源树与目的树之间的对应关系。定义文件生成部 86 提供图形用户界面，便于用户生成定义文件。

VC 单元 80 对源树与目的树之间的连接进行监控，当 VC 单元 80 通过由负责显示的插件提供的用户界面从用户处接收编辑指令时，它首先修改源树的相应节点。一旦 DOM 单元 30 发出表明源树已经被修改的变化事件，则 VC 单元 80 接收该变化事件，并对应于被修改的节点而修改目的树的节点，以使得目的树与源树的修改同步。对目的树进行显示/编辑的插件例如 HTML 单元 50 接收表明目的树已经被修改的变化事件，参考被修改的目的树而对显示进行更新。通过上述结构，即使是以少数用户使用的局部词汇来描述文档，也可以通过将其转换为其他的主要词汇来正确显示文档，同时提供理想的编辑环境。

下面将对利用文档处理装置 20 显示和/或编辑文档的操作进行说明。当文档处理装置 20 载入作为处理对象的文档时，DOM 生成部 34 从 XML 文档生成 DOM 树。此外，主控单元 22 或编辑单元 24 通过参考命名空间来判别描述文档的词汇。如果与词汇相对应的插件安装在文档处理装置 20 中，则该插件被载入以显示/编辑文档。在插件并未安装其中的情况下，则确认是否存在用于映射的定义文件。如果存在定义文件，则定义文件获取部 84 获取该定义文件，并根据定义生成目的树，以使得能够通过映射目标的词汇相对应的插件来显示/编辑文档。如果该文档是包含多个词汇的复合文档，如后所述，则通过与各词汇相对应的插件来显示/编辑该文档的相关部分。如果不存在定义文件，则显示文档的源或树型结构，

并在此显示屏幕中进行编辑。

图 2 示出了作为处理对象的 XML 文档的一个例子。该 XML 文档用于对学生的成绩数据进行管理。作为 XML 文档的上部节点的构成元素“成绩”下方具有设置成各个学生的多个构成元素“生徒”。构成元素“生徒”具有属性“名前”，并包括作为子元素“国語”、“数学”、“理科”以及“社会”。属性“名前”存储学生的姓名。构成元素“国語”、“数学”、“理科”和“社会”分别存储日语、数学、自然科学和社会科学的成绩。例如，姓名为“A”的学生的成绩是：日语成绩为“90”、数学成绩为“50”、自然科学成绩为“75”以及社会科学成绩为“60”。下文中，该文档中使用的词汇（标签集）被称作“成绩管理词汇”。

由于本前提技术的文档处理装置 20 不具有与成绩管理词汇的显示/编辑相对应的插件，因此，为了以源显示、树显示以外的方法来显示该文档，将使用上述的 VC 功能。也就是说，需要准备定义文件以将成绩管理词汇映射为已具有插件的另一词汇，例如 HTML 或 SVG 等。下面将对用户本身用以创建定义文件的用户界面进行说明，在此，是对已经具备了定义文件的情况进行说明。

图 3 示出了将图 2 中所示的 XML 文档映射为以 HTML 描述的表的一个例子。在图 3 所示的例子中，将成绩管理词汇描述的“生徒”节点与以 HTML 描述的表（“TABLE”节点）的行（“TR”节点）对应。各行的第一列与属性值“名前”相对应，第二列与“国語”节点的元素值相对应，第三列与“数学”节点的元素值相对应，第四列与“理科”节点的元素值相对应，而第五列与“社会”节点的元素值相对应。由此，图 2 所示的 XML 文档能以 HTML 的列表格式来显示。此外，这些属性值和元素值被指定为能够编辑，以使得用户能够通过 HTML 单元 50 的编辑功能在 HTML 形成的显示屏上对这些值进行编辑。在第六列中，指定了用来计算日语、数学、自然科学以及社会科学的成绩的加权平均的运算式，并显示每个学生成绩的平均值。以这种方式，通过在定义文件中指定运算式来完成更灵活的显示，从而提高用户在进行编辑时的便利性。另外，也可以将第六列指定为不可编辑，以使得不能单独对平均值本身进行编辑。由此，在映射定义中，能够指明可编辑或不能编辑，以避免用户可

能的错误操作。

图 4 (a) 及图 4 (b) 所示的是定义文件的一个例子, 以将图 2 所示的 XML 文档映射为图 3 所示的表。该定义文件由被定义为定义文件用的脚本语言来描述。在定义文件中描述有命令定义和显示模板。在图 4 (a) 及图 4 (b) 中, 定义了“生徒の追加”(意为“增加学生”)和“生徒の削除”(意为“删除学生”)作为命令, 并分别涉及将节点“生徒”插入源树中的操作以及将节点“生徒”从源树中删除的操作。此外, 作为模板描述了诸如“名前”和“国語”等标题显示于表的第一行中, 而节点“生徒”的内容显示于第二行及其随后的行中。在显示节点“生徒”内容的模板中, 包含“text-of”的项表示允许进行编辑, 而包含“value-of”的项表示不允许进行编辑。在这些显示了节点“生徒”内容的行中, 在第六行中描述了运算式“(src: 国語+ src: 数学 + src: 理科 + src: 社会) div 4”。这意味着显示学生成绩的平均值。

图 5 所示的是将图 2 所示的由成绩管理词汇描述的 XML 文档利用图 3 所示的对应关系映射至 HTML 以使其显示在显示屏上时, 显示屏的一个例子。在表 90 各行中从左至右显示的是各学生的姓名, 以及日语成绩、数学成绩、自然科学成绩、社会科学成绩及平均分数。用户能够在屏幕上对 XML 文档进行编辑。例如, 当第二行第三列中的值变为“70”时, 源树中与该节点相对应的元素值, 亦即学生“B”的数学成绩变为“70”。此时, 为了使目的树符合源树, VC 单元 80 修改目的树的相应部分, 从而使得 HTML 单元 50 能够根据改变的目的树来对显示进行更新。因此, 在屏幕上的表中, 学生“B”的数学成绩变为“70”, 而平均分数相应地变为“55”。

在图 5 所示的屏幕上, 例如“生徒の追加”和“生徒の削除”的命令被显示为菜单, 如图 4 所示的定义文件中所定义的那样。当用户从这些命令中选择一个命令时, 节点“生徒”增加至源树中或从源树中删除。如此, 在本前提技术的文档处理装置 20, 不仅能够对分级结构末端中的构成元素的元素值进行编辑, 而且能够对该分级结构进行编辑。具有上述树型结构的编辑功能能够以命令的形式提供给用户。此外, 例如, 增加或删除表中的行的命令也可以与增加或删除节点“生徒”的操作对应。

此外，嵌入其它词汇中的命令也可提供给用户。也可以将该表作为输入用模板，以使得对于新学生的成绩数据能够以填空的方式来增加。如上所述，通过 VC 功能，可以使用 HTML 单元 50 的显示/编辑功能，而且可以对由成绩管理词汇描述的文档进行编辑。

图 6 所示的是由定义文件生成部 86 呈现给用户的图形用户界面的一个例子，以使用户能够生成定义文件。映射源的 XML 文档在屏幕的左侧区域 91 显示为树。映射目标的 XML 文档的屏幕布局显示在屏幕的右侧区域 92 中。该屏幕布局可通过 HTML 单元 50 来编辑，用户在屏幕的右侧区域 92 中创建用于对文档进行显示的屏幕布局。然后，例如，通过诸如鼠标等指示设备将屏幕的左侧区域 91 中显示的 XML 文档的映射源的节点拖动并放置到屏幕的右侧区域 92 中显示的 HTML 屏幕布局中，以指定映射源处的节点与映射目标处的节点之间的连接。例如，当作为元素“生徒”的子元素的“数学”被放置到 HTML 屏幕上的表 90 中第一行第三列处时，“数学”节点与第三列中的“TD”节点之间建立连接。各节点均如此被指定为可编辑或者不可编辑。此外，可在显示屏中嵌入运算式。当屏幕编辑结束时，定义文件生成部 86 生成描述屏幕布局与节点之间的连接的定义文件。

虽然已经开发出了与 XHTML（可扩展超文本标记语言）、MathML（数学标记语言）以及 SVG（可缩放向量图形）等主要词汇对应的浏览器或编辑器，但是，并未开发出适于与由如图 2 中所示文档那样的自创词汇描述的文档对应的浏览器或编辑器。然而，如上所述，如果创建了用于映射为其它词汇的定义文件，则可以利用 VC 功能对由自创词汇描述的文档进行显示和/或编辑，而无需开发浏览器或编辑器。

图 7 所示的是由定义文件生成部 86 生成的屏幕布局的另一个例子。在图 7 所示的例子中，在用于显示由成绩管理词汇描述的 XML 文档的屏幕上创建有表 90 和圆图 93。圆图 93 以 SVG 描述。如后面所述，由于本前提技术的文档处理装置 20 能够对在单个 XML 文档内包含多个词汇的复合文档进行处理。因此如该例子那样，能够在同一屏幕上显示以 HTML 描述的表 90 以及以 SVG 描述的圆图 93。

图 8 所示的是由文档处理装置 20 处理的 XML 文档的编辑屏幕。在

图 8 所示的例子中,单个屏幕被分割为多个区域,而作为处理对象的 XML 文档以多种不同的显示格式显示在各个区域。该文档的源显示在区域 94 中,该文档的树结构显示在区域 95 中,而图 5 所示的以 HTML 描述的表显示在区域 96 中。文档的编辑可以在这些区域中的任意屏幕上进行,当用户在任意屏幕上进行编辑时,源树将被修改,负责各屏幕显示的各插件对应当反映上述源树修改的屏幕进行更新。具体而言,负责显示各编辑屏幕的插件的显示部被预先注册为通知源树修改的变化事件的监听器,当源树被任意插件或 VC 单元 80 修改时,显示编辑屏幕中的所有显示部接收所发出的变化事件,由此更新屏幕。此时,在插件通过 VC 功能进行显示的情况下,VC 单元 80 遵循源树的修改来修改目的树,之后,插件的显示部参考被修改的目的树来更新屏幕。

例如,在通过专用插件来实现源显示和树显示的情况下,源显示用插件和树显示用插件直接参考源树进行显示,而不利用目的树。在这种情况下,当在任意屏幕中进行编辑时,源显示用插件和树显示用插件参考被修改的源树来更新屏幕,同样,负责区域 96 的屏幕的 HTML 单元 50 通过参考目的树来更新屏幕,该目的树已根据对源树的修改而做了修改。

源显示和树显示也可以使用 VC 功能来实现。也就是说,可以通过 HTML 对源和树型结构进行布局,将 XML 文档映射为该 HTML 并通过 HTML 单元 50 来显示。在这种情况下,将生成具有源格式、树格式、表格式的树。如果在任意屏幕中进行编辑,则 VC 单元 80 对源树进行修改,并在之后分别对具有源格式、树格式、表格式的树进行修改。然后,HTML 单元 50 通过参考三个目的树来更新三个屏幕。

如此,通过在单个屏幕上以多种显示格式显示文档,可以提高用户的便利性。例如,用户可以通过源显示或树显示来掌握文档的分级结构,而且能够利用表 90 等以视觉上易于了解的格式显示和编辑文档。在上述例子中,单个屏幕被划分而同时显示多个显示格式的屏幕。但是,也可在单个屏幕上显示单个显示格式,通过用户指令来切换显示格式。在这种情况下,主控单元 22 接收来自用户的用于切换显示格式的请求,命令各自的插件进行显示切换。

图 9 所示的是由文档处理装置 20 编辑的 XML 文档的另一个例子。在图 9 所示的 XML 文档中，XHTML 文档被嵌入 SVG 文档的“foreignObject”标签中，而且，该 XHTML 文档中包含以 MathML 描述的公式。在这种情况下，编辑单元 24 参考命名空间将描绘任务分配或赋予给适当的显示系统。在图 9 所示的例子中，编辑单元 24 首先使 SVG 单元 60 描绘矩形，然后使 HTML 单元 50 描绘 XHTML 文档。而且，编辑单元 24 使图中未示出的 MathML 单元描绘公式。以这种方式，包含多个词汇的复合文档被适当地显示。图 10 示出了显示结果。

在文档编辑期间，可以根据光标（キヤリッジ：命令行）的位置来切换被显示的菜单。也就是说，当光标位于显示 SVG 文档的区域中时，显示提供 SVG 单元 60 的菜单或者由用于映射 SVG 文档的定义文件所定义的命令。当光标位于显示 XHTML 文档的区域中时，显示提供 HTML 单元 50 的菜单或者由用于映射 XHTML 文档的定义文件所定义的命令。由此，可根据编辑位置提供适当的用户界面。

如果在复合文档中不存在与词汇相符的适当插件或映射定义，则以该词汇描述的部分可显示为源或树格式。在传统实践中，在要打开其中某个文档中嵌有另一文档的复合文档时，如果其中未安装有对该嵌入文档进行显示的应用程序，则它们的内容不能被显示。但是，在本前提技术中，即使用于显示的应用程序不存在，也可以通过将由文本数据组成的 XML 文档显示为源或树格式，以确定其内容。这就是所谓作为文本的 XML 文档等文档的一个特征。

以文本的语言来描述数据的另一个有益方面在于，例如，复合文档中以某词汇描述的部分也可以参考同一文档中以其他词汇描述的部分的数据。此外，在进行 HTML 文档内的检索时，在 SVG 等的图中嵌入的字符串也可以作为检索对象。

在以某个词汇描述的文档中，也可以使用其它词汇的标签。虽然该 XML 文档通常并不有效（valid），但只要它结构良好（well-formed），就可作为有效的 XML 文档而被处理。在这种情况下，被插入的其它词汇的标签可使通过定义文件来进行映射。例如，在 XHTML 文档中，可使用诸如“重要”、“最重要”的标签，着重显示由这些标签包围的部分，也

可以按照重要性的顺序来排序而进行显示。

当文档在如图 10 所示的编辑屏幕中被用户编辑时，负责被编辑的部分的插件或 VC 单元 80 对源树进行修改。在源树中，可以为各节点注册变化事件的监听器。通常，与各个节点所属的词汇对应的插件的显示部或 VC 单元 80 被注册为监听器。当源树被修改时，DOM 提供部 32 从被修改的节点向更高层搜索。如果存在注册的监听器，则 DOM 提供部 32 向该监听器发出该变化事件。例如，在图 9 所示的文档中，如果<html>节点下方的节点被修改，那么变化事件被通报给被注册为<html>节点的监听器的 HTML 单元 50，同时，该变化事件也被通报给被注册为位于<html>节点上方的<svg>节点中的监听器的 SVG 单元 60。此时，HTML 单元 50 参考被修改的源树而更新显示。由于属于其自身的词汇的节点并未被修改，因此 SVG 单元 60 可忽视该变化事件。

根据编辑的内容，可以随着 HTML 单元 50 对显示进行的更新来改变总体布局。在这种情况下，各插件的各个显示区域的布局通过管理屏幕布局的组件，例如负责显示最高节点的插件来更新。例如，当由 HTML 单元 50 显示的区域较之以前变大时，HTML 单元 50 首先描绘其本身所负责的区域，然后确定显示区域的大小。然后，显示区域的大小被通报给管理屏幕布局的组件，以请求对布局进行更新。负责屏幕布局的组件接收通知并对各个插件的显示区域进行重新布置。因此，被编辑的部分的显示被适当更新，同时屏幕总体的布局被更新。

（实施例 1）

本发明的实施例 1 涉及数据插入装置，其将以 XML 及其他语言描述的文档（以下称为“处理对象文档”）变换成 XHTML（以下将变换成 XHTML 的文档称为“XHTML 文档”）并显示在屏幕等上，并且接收来自用户应插入到 XHTML 文档中的字符和符号等数据，将该数据插入处理对象文档中。本实施例所涉及的数据插入装置在字符的输出和复制&粘贴等中，不仅允许插入单纯的文本形式的字符串，而且允许插入附加有 XML 形式的标签的字符串和具有 XML 形式的分级结构的字符串（以下称为“插件”）等，同时，插入时可以保持处理对象文档的分级结构的完整性。数据插入装置存储创建处理对象文档的词

汇(以下称为“处理对象词汇”)的规则以及处理对象词汇中包含的构成元素的规则。具体来说,数据插入装置存储架构(スキーマ)(以下称为“处理对象架构”)。而且,根据处理对象文档中应插入数据位置的处理对象词汇以及处理对象架构,数据插入装置变换应插入的数据,将变换的数据插入处理对象文档中。

图 11 所示的是实施例 1 所涉及的数据插入装置 300 的结构。数据插入装置 300 包括用户界面部 302、处理部 304、辞典存储部 306、文档处理装置 100。此外,处理部 304 包括数据获取部 308、变换候选获取部 310、变换部 312、性质获取部 314、插入部 316。在此,数据插入装置 300 是权利要求书的范围所涵盖的文档处理装置的一个例子,文档处理装置 100 相当于前提技术中说明的文档处理装置 100。

用户界面部 302 接收来自用户的指令。在此,图中未示出的屏幕显示有变换成 XHTML 文档的处理对象文档(这里,所用的术语“文档”包括 XHTML 文档和处理对象文档,此外,本说明书所用术语“文档”是指一般意义上的而非进一步限定的文档)。对于所显示的文档,用户将规定的指令输入用户界面部 302 中。所谓预定的指令,例如是将字符和符号等数据输入文档中的指令或将记载在文档的一部分中的字符复制&粘贴到文档的其他部分中的指令等。另外,预定的指令也可以是剪切&粘贴或拖动&放置(drop),即,可以是对文档进行加工的处理。

数据获取部 308 通过用户界面部 302 接收应插入文档中的数据。在此,插入的数据也可以是上述的输入的字符和复制&粘贴的字符。

辞典存储部 306 预先存储辞典,用于变换由数据获取部 308 输入的数据。辞典存储部 306 不仅存储有用于将“平假名”变换成“汉字”的辞典(以下称为“一般辞典”),而且还存储有与 MathML 和 SVG 等对应的辞典。变换候选获取部 310 根据辞典存储部 306 中存储的辞典对数据获取部 308 中接收的数据进行变换。例如,在由数据获取部 308 接收的数据为“る一と 2”的情况下,变换候选获取部 310 使用一般辞典将其变换成文本形式的“ルート 2”,而且使用 MathML 的辞典将其变换成附加有 MathML 形式标签的“ルート 2”,即 $\langle \text{maqrt} \rangle \langle \text{mrow}$

> 2 </mrow> </msqrt> 的字符串。此外，在由数据获取部 308 接收的数据为“ $\sqrt{2}$ ”的情况下，变换候选获取部 310 使用一般辞典将其变换成文本形式的“ $\sqrt{2}$ ”，并且使用 SVG 的辞典将其变换成 SVG 形式的“circle”。而且，变换候选获取部 310 将使用多个辞典变换得到的数据分别显示在屏幕中。

性质获取部 314 接收处理对象文档中应插入数据的位置中的描述该处理对象文档的标记语言的性质，例如，处理对象词汇和处理对象架构等。在处理对象文档以一个处理对象词汇描述的情况下，性质获取部 314 原封不动地获取该处理对象词汇，但在处理对象文档以多个处理对象词汇描述的情况下，性质获取部 314 获取应插入数据位置的处理对象词汇。另外，为获取处理对象词汇，使用处理对象文档中包含的命名空间声明。此外，为获取处理对象架构，例如使用附属于处理对象文档的说明书。

变换部 312 接收来自用户的指令，以选择由变换候选获取部 310 显示在屏幕中的多种数据中的一种。而且，根据由性质获取部 314 接收的处理对象词汇或处理对象架构，变换部 312 判断用户选择的数据是否可以插入处理对象文档中，如果可以插入，则确定该数据的插入。也就是说，变换候选获取部 310 和变换部 312 对由数据获取部 308 接收的数据进行变换。例如，在由性质获取部 314 接收的处理对象词汇为 MathML 的情况下，如果附加有 MathML 形式的标签的“ $\sqrt{2}$ ”被选择，则变换部 312 将其确定为附加有 MathML 形式的标签的“ $\sqrt{2}$ ”，以与 MathML 对应。另外，也可以是文本形式的“ $\sqrt{2}$ ”。

此外，在由于性质获取部 314 接收的处理对象架构的规定，不允许插入 MathML 的 <msqrt> 标签的情况下，变换部 312 将其确定为文本形式的“ $\sqrt{2}$ ”。而且，变换部 312 将与处理对象文档对应的标签添加到已选择的数据并输出。在此，对于添加的标签的具体例子将在后面描述。另一方面，根据由性质获取部 314 接收的处理对象词汇或处理对象架构，用户不能确定选择的数据的情况下，例如当处理对象词汇为 MathML 时，用户却选择了 SVG 形式的“circle”的情况下，变换部 312 也可以不实行变换。此时，通过屏幕将此情况通知用户。

另外，变换部 312 也可以代替变换候选获取部 310，根据由性质获取部 314 接收的处理对象词汇或处理对象架构确定由变换候选获取部 310 变换的多种数据的优先顺序，并通过图中未示出的屏幕按照优先顺序显示变换的多种数据。例如，也可以首先显示与处理对象词汇或处理对象架构一致的数据，然后显示与处理对象词汇或处理对象架构兼容的数据，最后显示与处理对象词汇或处理对象架构都不兼容的数据。

插入部 316 将由变换部 312 选择的数据插入图中未示出的处理对象文档中。另外，当实行以上处理时，处理部 304 使用文档处理装置 100 中包含的预定功能，但省略其说明。例如，处理部 304 使用文档处理装置 100 中的 CSS 单元 140、HTML 单元 150、SVG 单元 160。

图 12(a) - (c) 所示的是通过数据插入装置 300 插入的数据的例子。图 12(a) 所示的是用户输入字符的屏幕。在此，从处理对象文档变换成 XHTML 的 XHTML 文档被显示在屏幕中，用户输入“ $\sqrt{2}$ ”。一旦用户通过预定的接口(interface)对字符的变换进行指示，则在变换候选窗口 400 中将附加有 MathML 形式标签的“ $\sqrt{2}$ ”显示为“1”，将文本形式且为片假名的“ $\sqrt{2}$ ”显示为“2”，将文本形式且为平假名的“ $\sqrt{2}$ ”显示为“3”。

图 12(b) 所示的是在图 12(a) 的变换候选窗口 400 中所示的变换候选中，在选择了“1”附加有 MathML 形式标签的“ $\sqrt{2}$ ”的情况下，应插入处理对象文档中的字符串。另外，图中记载了 XML 文档的一部分。在此，附加有 MathML 形式标签的“ $\sqrt{2}$ ”为从 $\langle \text{maqrt} \rangle$ 至 $\langle / \text{msqrt} \rangle$ 包围的部分，而将其夹住并附加的 $\langle \text{chapter} \rangle$ 、 $\langle \text{paragraph} \rangle$ 、 $\langle \text{section} \rangle$ 为由上述变换部 312 附加的标签，相当于与处理对象文档对应的标签。也就是说，在该例子中，假设插入位置的标签集被预先规定包含 $\langle \text{chapter} \rangle$ 、 $\langle \text{paragraph} \rangle$ 、 $\langle \text{section} \rangle$ ，变换部 312 附加这些标签，以保持处理对象文档分级结构的完整性，此外以在处理对象架构被规定的情况下遵循该架构。图 12(c) 所示的是在图 12(a) 的变换候选窗口 400 中所示的变换候选中，在选择了“2”的文本形式的“ $\sqrt{2}$ ”的情况下，应插入处理对象文档中

的字符串。在此，在文本的“ルート 2”中附加有与处理对象词汇或处理对象架构对应的标签。

图 13 所示的是由数据插入装置 300 执行的数据插入处理程序的流程图。变换部 312 接收用户从对数据获取部 308 所获取的数据进行变换时的候选中选择的数据 (S10)。性质获取部 314 确定处理对象文档中该数据的插入位置 (S12)。此外，性质获取部 314 获取处理对象词汇 (S14)。进而，性质获取部 314 获取处理对象架构 (S16)。变换部 312 判断数据是否与处理对象词汇或处理对象架构兼容。如果兼容 (S18 为 Y)，则在数据变换之后插入部 316 将其插入处理对象文档中 (S20)。另一方面，如果数据与处理对象词汇或处理对象架构不兼容 (S18 为 N)，则通过屏幕通知不变换的情况 (S22)。另外，变换部 312 也可以将其变换成可插入文本形式等的处理对象文档中的形式，然后插入部 316 自动地插入。

图 14 是数据插入装置 300 进行的数据插入处理程序的另一例子的流程图。在对流程图进行说明之前，对该处理大致进行说明。该处理理由与图 11 所示的数据插入装置 300 相同类型的数据插入装置 300 进行，但一部分处理的内容不同。在此，将复制&粘贴的处理作为对象。在由数据获取部 308 接收的数据中预先附加有预定的标签。变换部 312 根据附加的标签和由性质获取部 314 接收的处理对象架构对附加在数据中的标签进行修改。此时，当从附加在由数据获取部 308 获取的数据中的标签判断该数据为 XML 数据 (片段) 时，变换部 312 将其作为 XML 片段试着插入。另一方面，根据包含处理对象架构的标记语言的性质，在不能作为 XML 片段插入的情况下，则删除标签作为文本数据插入。变换部 312 也可以根据需要补充命名空间。例如，被插入的片段所属的命名空间与被插入位置前后的片段所属的命名空间不同的情况。在此情况下，变换部 312 也可以在附加于片段的标签中描述命名空间。例如，在附加 < section > < /section > 的标签时，也可以附加并插入 < section xmlns: =”名前空間 URI” > < /section > 的标签。

数据获取部 308 接收应插入文档中的附加有标签的数据 (S40)。性质获取部 314 确定处理对象文档中该数据的插入位置 (S42)。此外，

性质获取部 314 获取处理对象架构 (S44)。在变换部 312 将数据解释为 XML 片段的情况下 (S46 为 Y)，并且在插入部 316 可以将数据作为 XML 片段插入的情况下 (S48 为 Y)，插入部 316 将其作为 XML 片段插入 (S50)。另一方面，在数据不能解释为 XML 片段的情况下 (S46 为 N)，或者在插入部 316 不可以将数据作为 XML 片段插入的情况下 (S48 为 N)，插入部 316 将数据作为文本插入 (S52)。

图 15(a) - (d)所示的是数据插入装置 300 插入的数据的例子。这些图所示的是与图 14 中说明的处理对应的屏幕显示以及源显示。图 15(a)所示的是将处理对象文档显示为 XHTML 文档的屏幕。如图所示，第一行记载有“これがセクション 1”，第二行记载有“これがセクション 2”。图 15(b)所示的是与图 15(a)对应的 XML 中的源。图中记载了 XML 文档的一部分。如图所示，< section > 和 < /section > 的标签之间描述有上述字符。图 15(c)所示的是复制图 15(a)的第一行记载的“これがセクション 1”并粘贴到第三行时的屏幕。图 15(d)所示的是与图 15(c)对应的 XML 中的源。如果用户选择“これがセクション 1”作为复制对象，则在其前后附加 < section > 和 < /section > 的标签，并输入数据获取部 308 中。由于变换部 312 从标签的内容可以解释为 XML，因此基本无需变换就可以插入到处理对象文档中。另外，变换部 312 也可以附加预定的标签，例如 < paragraph > 和 < /paragraph >，然后将其插入处理对象文档中。

图 16 所示的是数据插入装置 300 进行的数据插入处理程序的另一例子的流程图。在对流程图进行说明之前，对该处理大致进行说明。在图 14 中，根据附加在由数据获取部 308 接收的数据中的标签和由性质获取部 314 获取的处理对象架构来确定处理对象词汇。但在图 16 中，变换部 312 根据附加在由数据获取部 308 接收的数据中的标签和由性质获取部 314 获取的处理对象词汇来确定处理对象架构。而且，变换部 312 按照与确定后的处理对象架构对应的方式来修改标签。

数据获取部 308 接收应插入文档中的附加有标签的数据 (S60)。性质获取部 314 确定处理对象文档中该数据的插入位置 (S62)。此外，性质获取部 314 获取处理对象词汇 (S64)。在变换部 312 根据附加的

标签和处理对象词汇获取处理对象架构 (S66)。在变换部 312 中如果数据与处理对象词汇或处理对象架构对应 (S68 为 Y)，则对数据进行变换，然后插入部 316 将其插入处理对象文档中 (S70)。另一方面，在变换部 312 中如果数据与处理对象词汇或处理对象架构不对应 (S68 为 N)，则插入部 316 将数据作为文本插入 (S72)。

图 17 所示的是实施例 1 所涉及的数据插入装置 300 的结构另一个例子。在对图 17 的数据插入装置 300 进行说明之前，对该装置涉及的处理进行大致说明。在此，使用描述了由定义文件获取部 184 获取的、映射定义的脚本文本 (以下称为“VCD”) 来确定处理对象词汇。用户在显示在屏幕中的 XHTML 文档中指定应插入的字符和插入位置。数据插入装置 300 根据 VCD 确定处理对象文档中应当插入字符的位置。而且，获取确定的位置中的处理对象词汇。与图 11 的数据插入装置 300 相比，图 17 的数据插入装置 300 包括位置确定部 320、词汇获取部 322。

位置确定部 320 接收来自图 1 的 VC 单元 180 的 VCD。此外，还获取由数据获取部 308 接收的 XHTML 文档中应当插入字符的位置。而且，使用 VCD 中描述的映射对应关系来确定与 XHTML 文档中应当插入字符的位置对应的处理对象文档中应当插入字符的位置。词汇获取部 322 确定处理对象文档中应当插入字符位置的处理对象词汇。后续处理由于与图 11 相同，故省略其说明。

图 18(a) - (d) 所示的是由图 17 所示的数据插入装置 300 插入的数据的例子。图 18(a) 所示的是屏幕显示。在此，示出了配置成二行二列的单元格 402。图 18(b) 所示的是与图 18(a) 对应的源显示。图 18(a) 中的四个单元格 402 由于不包含数据，因此与此对应描述有四个 `<td/>`。在此状态下，如图 18(c) 所示，用户将“七儿”输入到配置在左上方的单元格 402 中。图 17 的位置确定部 320 识别配置在左上方的单元格 402 与图 18(b) 中的一个 `<td/>` 对应。

词汇获取部 322 获取分别附加有 `<td>` 标签、`<tr>` 和 `<td>` 标签、`<table>` 和 `<tr>` 和 `<td>` 标签、`<body>` 和 `<table>` 和 `<tr>` 和 `<td>` 标签的四种变换候选，作为“七儿”的变换候选。变换部 312 在进

行变换时，根据标签的结构判断在上述四类变换候选中后两者应当被排除。原因是因为在 XHTML 的规则中不能将 <table> 进一步输入 <table> 中。因此，用户在上述四类变换候选中选择前两者之一。图 18(c) 是选择 <td> 标签的情况，此时，图 18(b) 的源显示中，<tr> <td/> <td/> </tr> 被变换成 <tr> <td> セル </td> <td/> </tr>。图 18(d) 是选择 <tr> 和 <td> 标签的情况，在图 18(b) 的源显示中附加有 <tr> <td/> <td/> </tr>。

图 19(a) - (e) 所示的是通过图 17 的数据插入装置 300 插入的数据的另一例子。在此，与图 18(a) - (d) 不同，将屏幕中未显示单元格 402 的状态作为开始状态。此为图 19(a)。图 19(b) 是用户输入“セル”而且原样地作为文本而被显示的情况。图 19(c) 为此时的源显示。另一方面，图 19(d) 是在与图 19(b) 相同也输入“セル”的情况下，变换部 312 将其解释为单元格 402 的输入时的屏幕显示。在此情况下，图 19(e) 所示的源显示被实现。如此，既可以保持处理对象文档分级结构的完整性，而且可以将数据插入到分级结构的多级中。

根据本发明的实施例，在将数据插入到处理对象文档中时，考虑处理对象词汇或处理对象架构对数据进行变换，然后，将其插入处理对象文档中，因此即使用户没有意识到处理对象词汇或处理对象架构也可以正确的插入数据。根据处理对象词汇对附加在数据中的标签进行修改，然后将其插入处理对象文档中，因此即使用户没有意识到处理对象词汇也可以正确地插入数据。根据处理对象架构对附加在数据中的标签进行修改，然后将其插入处理对象文档中，因此即使用户没有意识到处理对象架构也可以正确地插入数据。此外，在用户插入数据时，由于在没有意识到处理对象词汇或处理对象架构时可以进行处理，因此可以提高用户的便利性。根据显示处理对象文档时的标记语言的种类与处理对象词汇的关系，可以确定处理对象词汇。由于从处理对象词汇和附加在接收的数据中的标签可以确定处理对象架构，因此即使没有接收处理对象架构也可以适当地插入数据。而且，可以在不与处理对象文档的分级结构产生矛盾的情况下插入数据。

(实施例 2)

本发明的实施例 2 涉及数据插入装置，该装置将处理对象文档转换成 XHTML 的 XHTML 文档显示在屏幕等上，并接收由用户选择的 XHTML 文档的部分，将接收的部分插入 XHTML 文档其他的部分中。本发明所涉及的数据插入装置不仅可以适应复制的场合而且可以适应剪切的场合。通过复制所选择的字符存储在剪贴板中，而数据插入装置从处理对象文档的标签结构中提取配置有选择的字符的级的标签和比该标签上位的级的标签，改变它们的组合，同时生成多种的字符组合。

图 20 所示的是实施例 2 所涉及的数据插入装置 300 的结构。数据插入装置 300 包括处理部 304、存储器 334、文档处理装置 100。此外，处理部 304 包括数据获取部 308、性质获取部 314、标签生成部 330、标签附加部 332、变换部 312、插入部 316。在此，与图 11 一样，数据插入装置 300 是权利要求范围中的文档处理装置的一个例子，文档处理装置 100 相当于前提技术中所说明的文档处理装置 100。

数据获取部 308 通过用户界面部 302 接收来自用户的指令以选择处理对象文档一部分。在此，与实施例 1 一样，处理对象文档具有分级结构。此外，为复制或剪切处理对象文档的一部分而执行选择。性质获取部 314 接收包含在该文档中的标签的结构以作为处理对象文档的性质。

标签生成部 330 根据由性质获取部 314 接收的标签的结构，生成至少一个应当附加到由数据获取部 308 选择的处理对象文档的一部分上的标签。也就是说，生成至少一个应当附加到由数据获取部 308 选择的处理对象文档的一部分上的、标记语言的标签，以保持包含在处理对象文档中的标签结构的完整性。具体而言，标签生成部 330 从构成处理对象文档的分级结构中提取配置有已选择文档一部分的层的标签和比该层上位的层的标签。对于具体细节，后面将列举具体例子进行描述，而所谓上位的层的标签是配置在比配置有已选择的文档的一部分的层上位而比根元素下位的层中的标签。而且，可以按照从与下位的层对应的顺序对提取的标签进行组合，通过改变应当组合的标签的数量来生成多种标签的组合。

标签附加部 332 根据由标签生成部 330 生成的标签组合的数量，对已选择的处理对象文档的一部分进行复制，使多种标签的组合分别与复制的文档的一部分各自对应并且附加。也就是说，虽然“文档的一部分”的内容是相同的，但通过附加有不同的多个标签可生成多个“文档的一部分”。这样就相当于生成了多个片段。存储器 334 存储由标签附加部 332 输出的多个“文档的一部分”。

当变换部 312 通过用户界面部 302 接收来自用户以粘贴存储在存储器 334 中的“文档的一部分”为要点的指令时，读出存储在存储器 334 中的多个版本的“文档的一部分”。而且，变换部 312 根据用户的指令从多种“文档的一部分”中选择一种。插入部 316 根据附加在已选择的一种中的标签将其插入到处理对象文档中，以保持处理对象文档的标签的结构。另外，与图 11 一样，在执行以上处理时，处理部 304 使用包含在文档处理装置 100 中的预定的功能。

图 21 所示的是数据插入装置 300 中的数据获取程序的流程图。数据获取部 308 根据来自用户界面部 302 的指令选择处理对象文档的一部分 (S80)。性质获取部 314 获取包含在处理对象文档中的标签的结构。对于由数据获取部 308 选择的文档的一部分，标签生成部 330 提取分级结构中配置该文档的一部分的层的标签 (S82)。标签生成部 330 进一步检索上位的层的标签 (S84)。如果检索的标签与根元素的标签不匹配 (S86 为 N)，则反复执行从步骤 82 开始的处理，提取多个种类的标签。另一方面，如果检索的标签与根元素的标签匹配 (S86 为 Y)，则标签附加部 332 根据由标签生成部 330 中提取的标签的数量复制由数据获取部 308 选择的文档的一部分 (S88)。进而，改变提取的标签的组合，同时将标签附加到复制的文档的一部分中 (S90)。存储器 334 存储附加有标签的文档的一部分 (S92)。

图 22(a) - (d)所示的是由数据插入装置 300 插入的数据的一个例子。图 22(a)为处理对象文档的源显示。在此，具有将 <chapter> 作为根元素、与 <paragraph>、<section> 连续的分级结构。图 22(b)为图 22(a)的处理对象文档的屏幕显示。图 22(c)为以复制或剪切为目的，用户选择图 22(b)的屏幕显示的一部分时的屏幕显示，反白的部分相当于

用户选择的部分。图 22(d)为与图 22(c)的屏幕显示对应的处理对象文档的源显示，与图 22(c)一样，反白的部分相当于用户选择的部分。

被选择的部分与处理对象文档的分级结构中的 < section > 层对应。因此，标签生成部 330 还提取比其更上位的层且不是根元素的标签的 < paragraph >。最终，生成 < section > ... < /section > 的组合和 < paragraph > < section > ... < /section > < /paragraph > 的组合作为插入到已选择部分的前后的标签。进而，标签附加部 332 通过复制生成两份被选择的部分，对其附加两类标签组合中的任一种。

根据本发明的实施例，由于对于选自处理对象文档的一部分生成并附加标签，因此即使无需重新描述标签，也可以将其作为由标记语言描述的数据插入。此外，由于无需标签的描述，因此可以提高用户的便利性。此外，由于考虑了处理对象文档的标签的结构而生成标签，因此可以保持处理对象文档分级结构的完整性。此外，由于仅生成多个种类的标签，因此可以省略选择其中一个的动作。此外，选择的部分可以插入处理对象文档的各个部分中。

以上通过实施例对本发明进行了说明。本领域的普通技术人员应该理解，实施例仅为示例，本发明还存在对各构成元素或各处理过程进行组合的各种各样的变形实施例，这些变形实施例也包含在本发明的范围内。

在前提技术中，对处理 XML 文档的例子进行了说明，但前提技术的文档处理装置 100 同样也可以处理以其他标记语言例如 SGML、HTML 等描述的文档。

在实施例 1 中，复制或粘贴的数据为字符串，如果该字符串与插入位置中的处理对象词汇不相符，则变换部 312 确定不插入该字符。但是并不局限于此，例如，即使复制或粘贴的数据为预定的图形，变换部 312 也可以将其变换成 SVG 形式的数据，也就是说附加与 SVG 对应的标签集，并插入到处理对象文档中。此外，对于预定对象文档中的插入位置，在与应插入的数据对应的词汇没有被定义的情况下，变换部 312 也可以对与应插入的数据对应的词汇进行命名空间声明，并对词汇进行定义。根据本变形实施例，也可以将图形等插入处理对

象文档中。总之，只要不违反处理对象文档的分级结构就可以插入数据。

实施例2中，性质获取部314获取处理对象文档的标签结构作为处理对象文档的性质，但并不局限于此，例如，也可以获取处理对象词汇或处理对象架构。根据本变形实施例，可以与在处理对象文档中规定的各规则对应。总之，只要是处理对象中规定的规则就可以。

产业上的利用可能性

本发明可用作对以标记语言描述的文档进行处理的文档处理装置。

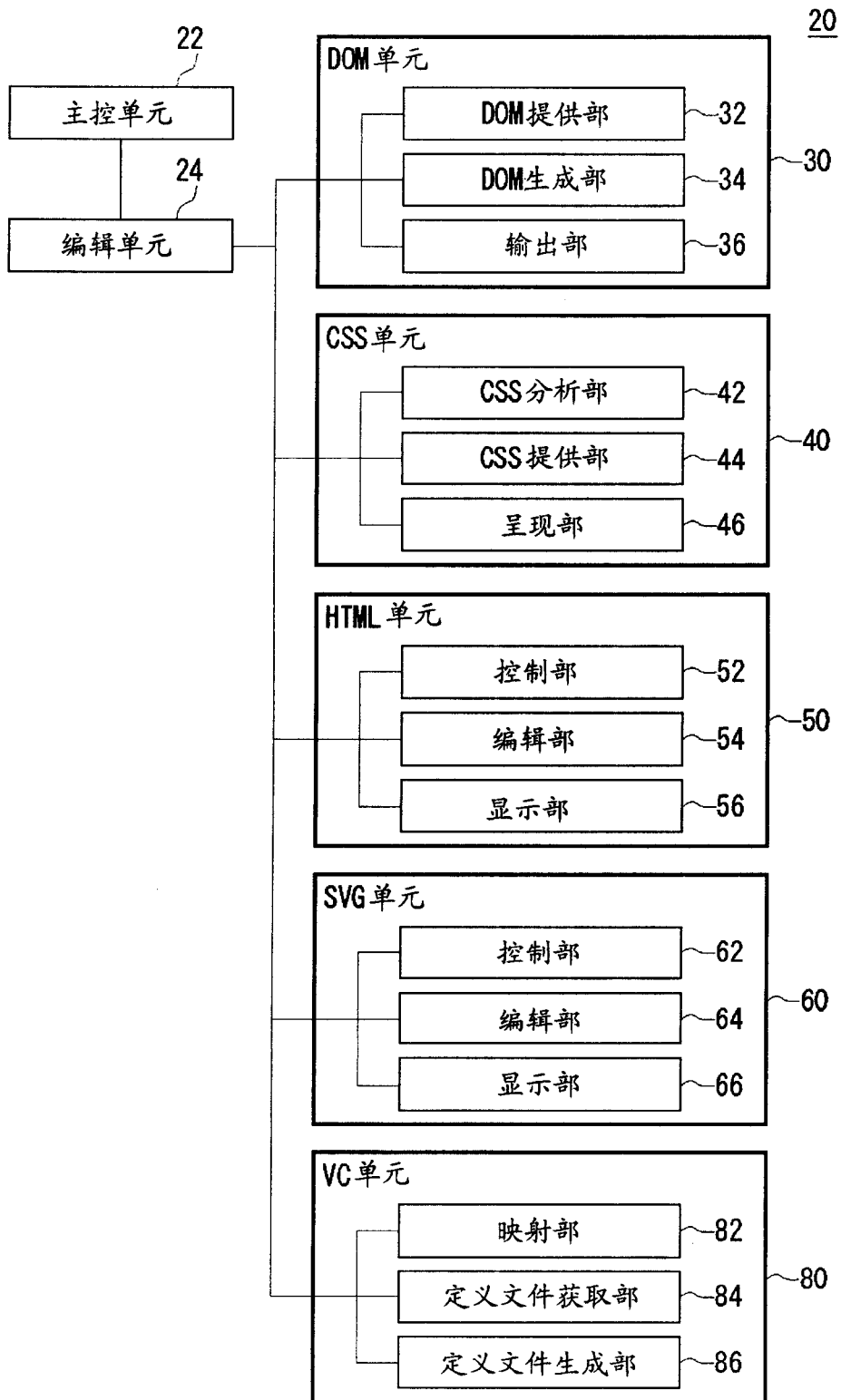


图 1

```
<?xml version="1.0" ?>
<?org.chimaira.vocabulary-connection href="records.vcd" ?>
<成績 xmlns="http://xmlns.justsystem.com/sample/records">
  <生徒 名前="A">
    <国語>90</国語>
    <数学>50</数学>
    <理科>75</理科>
    <社会>60</社会>
  </生徒>
  <生徒 名前="B">
    <国語>45</国語>
    <数学>60</数学>
    <理科>55</理科>
    <社会>50</社会>
  </生徒>
  <生徒 名前="C">
    <国語>55</国語>
    <数学>45</数学>
    <理科>95</理科>
    <社会>40</社会>
  </生徒>
  <生徒 名前="D">
    <国語>25</国語>
    <数学>35</数学>
    <理科>40</理科>
    <社会>15</社会>
  </生徒>
</成績>
```

图 2

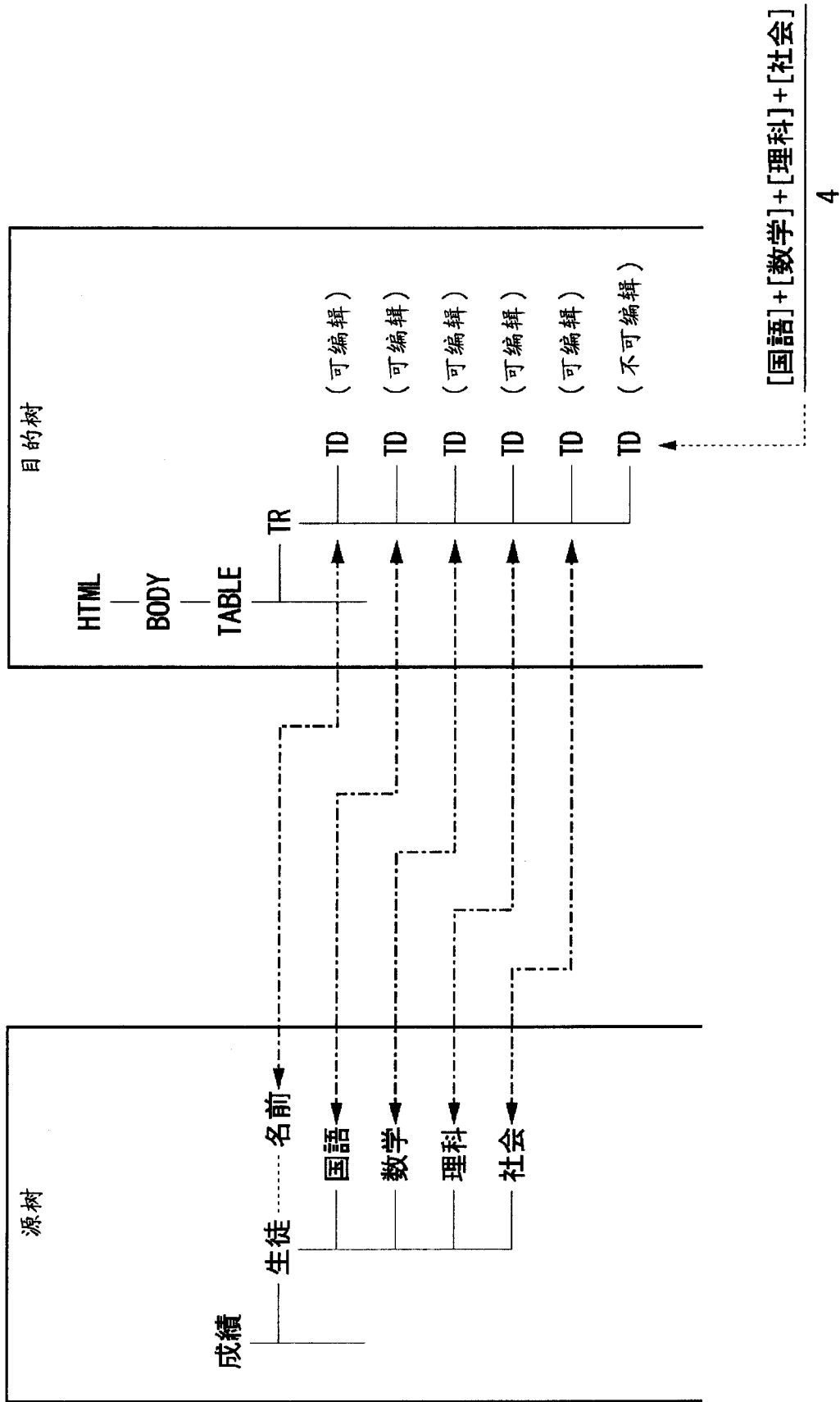


图 3

4


```

<?xml version="1.0"?>
<vc:vcd xmlns:vc="http://xmlns.chimaira.org/vcd"
  xmlns:src="http://xmlns.justsystem.com/sample/records"
  xmlns="http://www.w3.org/1999/xhtml"
  version="1.0">
  <!-- Commands -->
  <vc:command name="生徒の追加">
    <vc:insert-fragment
      target="ancestor-or-self::src:生徒"
      position="after">
      <src:生徒/>
    </vc:insert-fragment>
  </vc:command>
  <vc:command name="生徒の削除">
    <vc:delete-fragment target="ancestor-or-self::src:生徒" />
  </vc:command>

  <!-- Templates -->
  <vc:vc-template match="src:成績" name="成績表" >

    <vc:ui command="生徒の追加">
      <vc:mount-point>
        /MenuBar/成績表/生徒の追加
      </vc:mount-point>
    </vc:ui>
    <vc:ui command="生徒の削除">
      <vc:mount-point>
        /MenuBar/成績表/生徒の削除
      </vc:mount-point>
    </vc:ui>

    <html>
      <head>
        <title>成績表</title>
        <style>
          td,th {
            text-align:center;
            border-right:solid black 1px;
            border-bottom:solid black 1px;
            border-top:none 0px;
            border-left:none 0px;
          }
          table{
            border-top:solid black 2px;
            border-left:solid black 2px;
            border-right:solid black 1px;
            border-bottom:solid black 1px;
            border-spacing:0px;
          }
        </style>
      </head>
    </html>
  </vc:vc-template>

```

图 4(a)

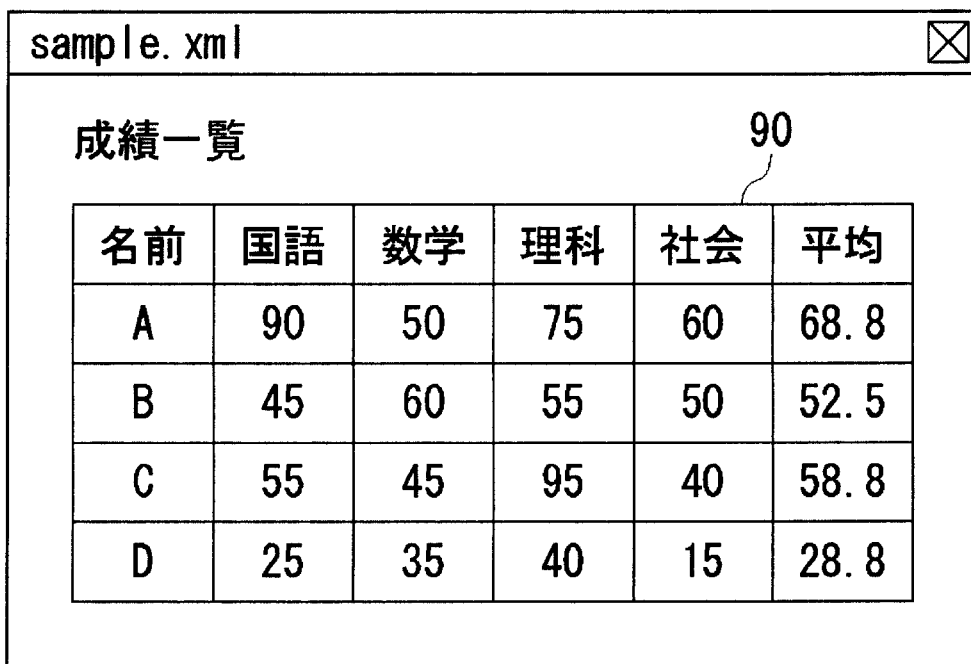
```

    tr {
      border:none;
    }
    .data {
      padding:0.2em 0.5em;
    }
  </style>
</head>
<body>
  <h1>成績一覧</h1>
  <table>
    <tr><th><div class="data">名前</div></th>
    <th></th>
    <th><div class="data">国語</div></th>
    <th><div class="data">数学</div></th>
    <th><div class="data">理科</div></th>
    <th><div class="data">社会</div></th>
    <th></th>
    <th><div class="data">平均</div></th></tr>
    <vc:apply-templates select="src:生徒" />
  </table>
</body>
</html>
</vc:vc-template>

<vc:template match="src:生徒">
  <tr>
    <td><div class="data">
      <vc:text-of select="@名前" fallback="名無し"/>
    </div></td>
    <td></td>
    <td><div class="data">
      <vc:text-of select="src:国語" fallback="0" type="vc:integer" />
    </div></td>
    <td><div class="data">
      <vc:text-of select="src:数学" fallback="0" type="vc:integer" />
    </div></td>
    <td><div class="data">
      <vc:text-of select="src:理科" fallback="0" type="vc:integer" />
    </div></td>
    <td><div class="data">
      <vc:text-of select="src:社会" fallback="0" type="vc:integer" />
    </div></td>
    <td><div class="data">
      <vc:value-of
        select="(src:国語 + src:数学 + src:理科 + src:社会) div 4" />
    </div></td>
  </tr>
</vc:template>
</vc:vcd>

```

图 4(b)



sample.xml

成績一覽

90

名前	国語	数学	理科	社会	平均
A	90	50	75	60	68.8
B	45	60	55	50	52.5
C	55	45	95	40	58.8
D	25	35	40	15	28.8

图 5

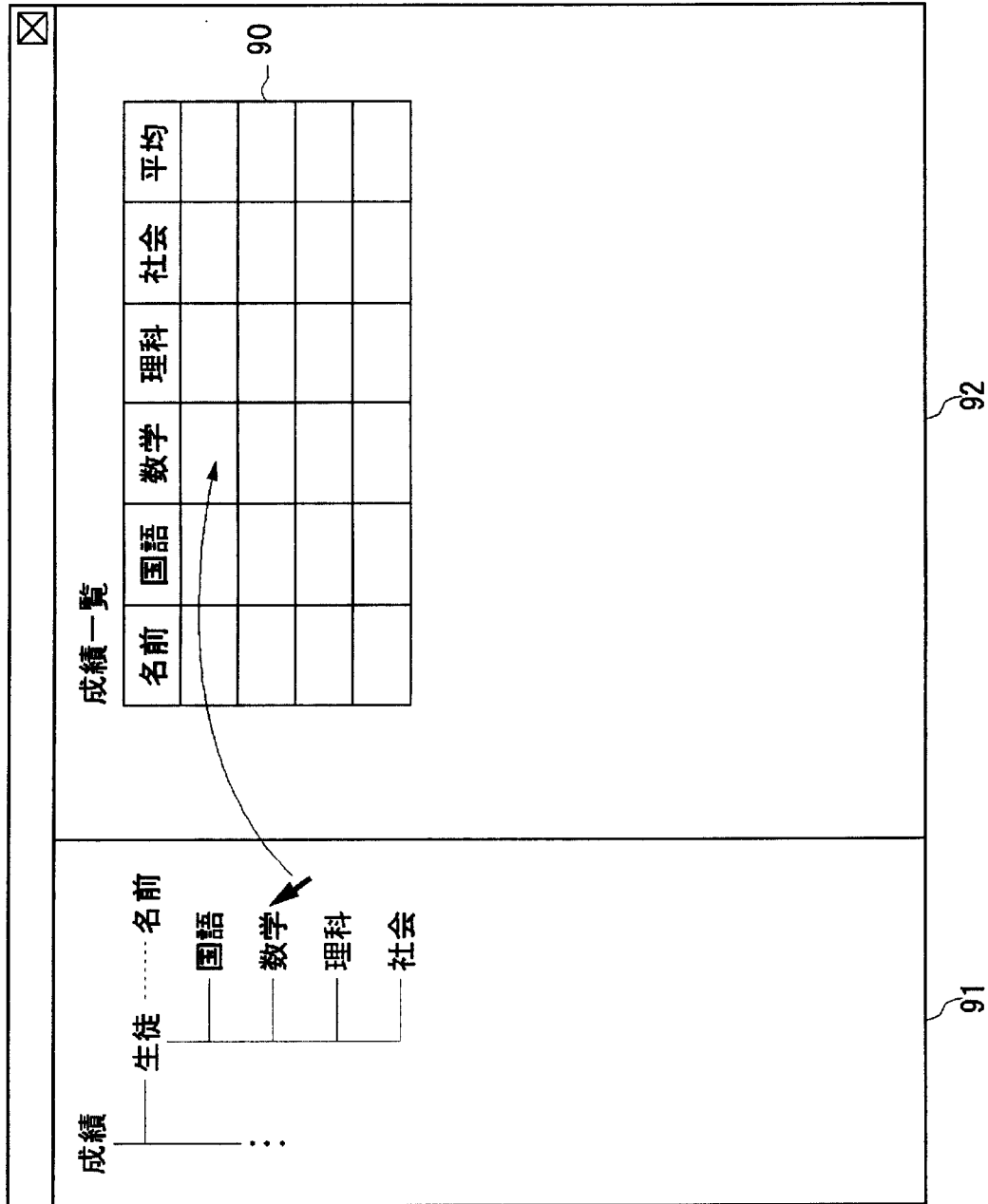


图 6

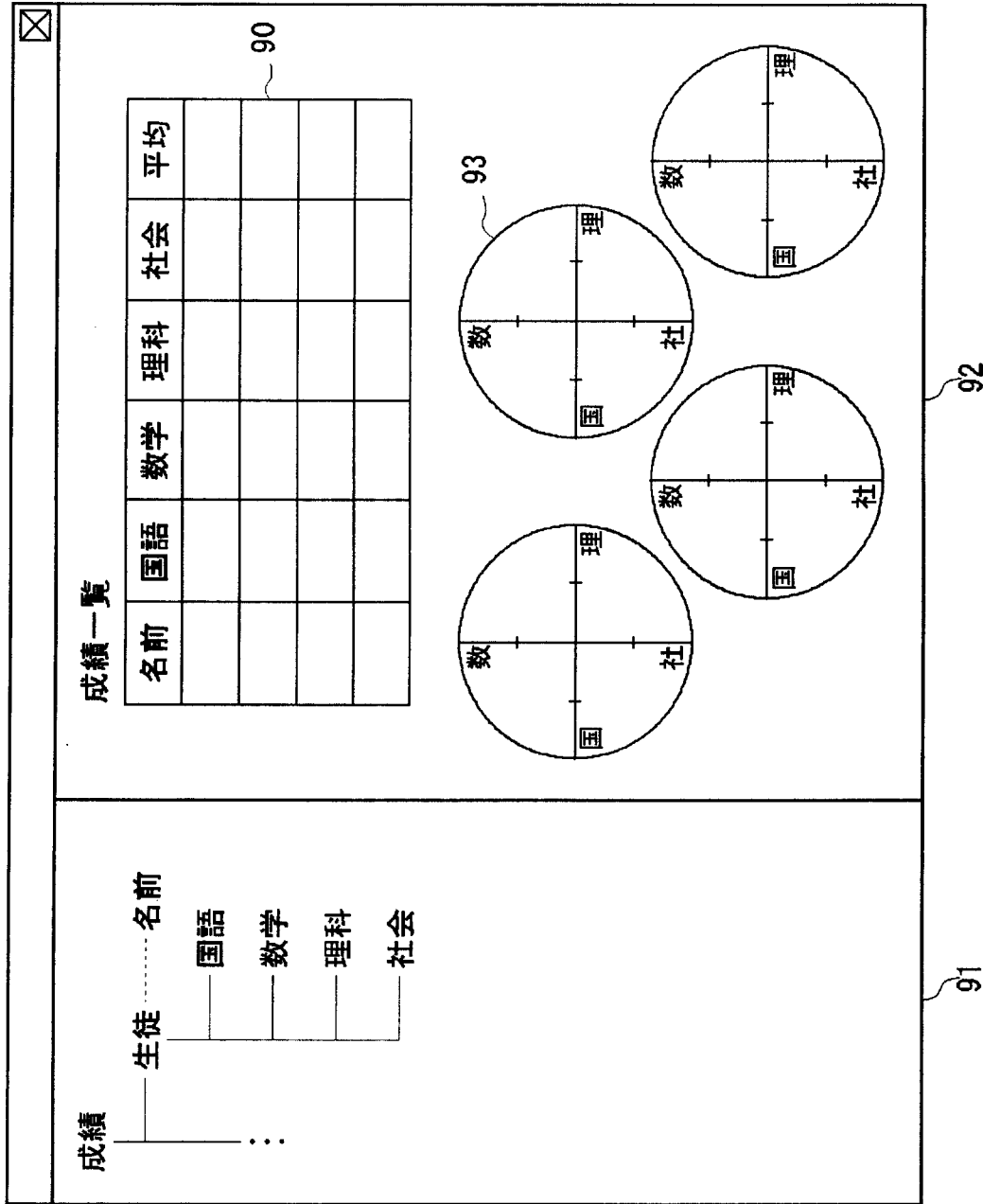


图 7

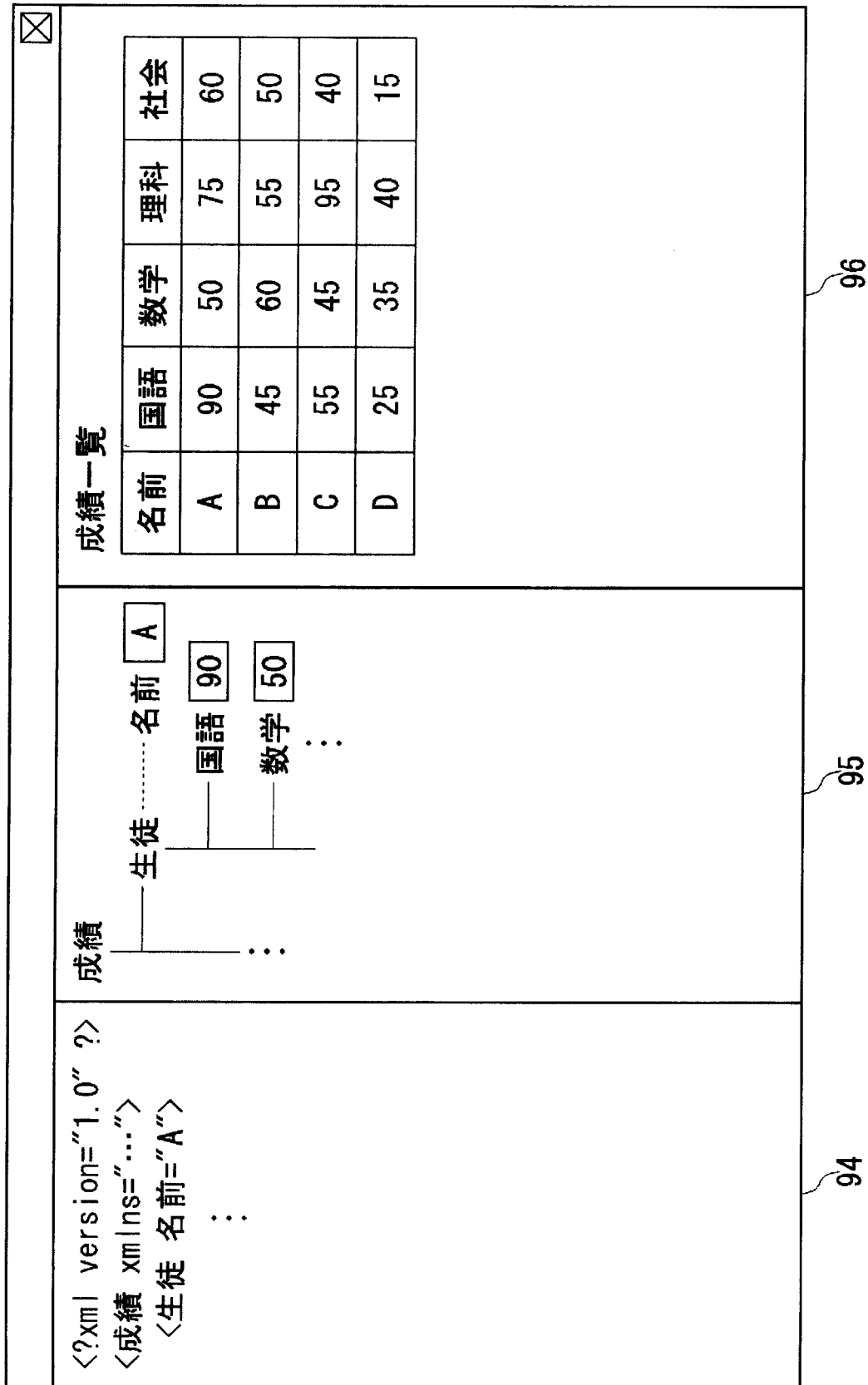


图 8

```

<?xml version="1.0" ?>
<svg xmlns="http://www.w3.org/2000/svg"
width="400" height="200"
viewBox="0 0 400 200"
>
  <rect x="-15" y="65" width="150" height="100" rx="20"
transform="rotate(-20)"
style="fill:none; stroke:purple; stroke-width:10"
/>
  <foreignObject x="190" y="10" width="200" height="200">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head><title /></head>
      <body bgcolor="#FFFFCC" text="darkgreen">
        <div style="font-size:12pt">
          SVG文書中に<math></math>を使って、
          XHTML文書を埋め込んでみました。
          数式も入れてみたりして：
          <div>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
              <mi>x</mi>
              <mo>=</mo>
              <mfrac>
                <mrow>
                  <mo>-</mo>
                  <mi>b</mi>
                  <mo>±</mo>
                  <msqrt>
                    <mrow>
                      <msup>
                        <mi>b</mi>
                        <mn>2</mn>
                      </msup>
                      <mo>-</mo>
                      <mn>4</mn>
                      <mi>a</mi>
                      <mi>c</mi>
                    </mrow>
                  </msqrt>
                </mrow>
                <mrow>
                  <mn>2</mn>
                  <mi>a</mi>
                </mrow>
              </mfrac>
            </math>
          </div><!-- math -->
        </div>
      </body>
    </html>
  </foreignObject>
</svg>

```

图 9

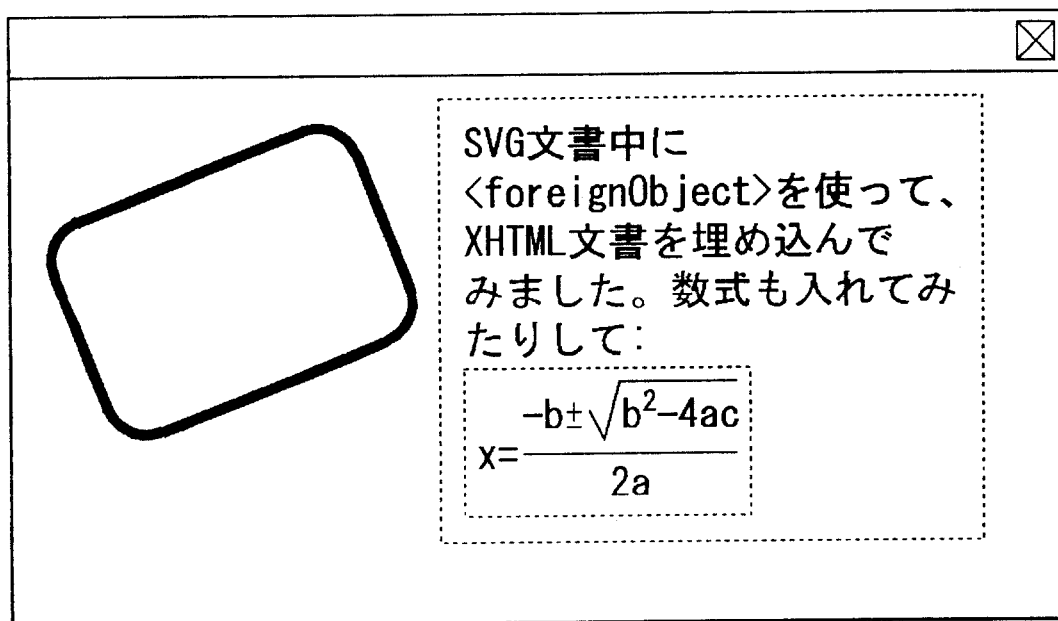
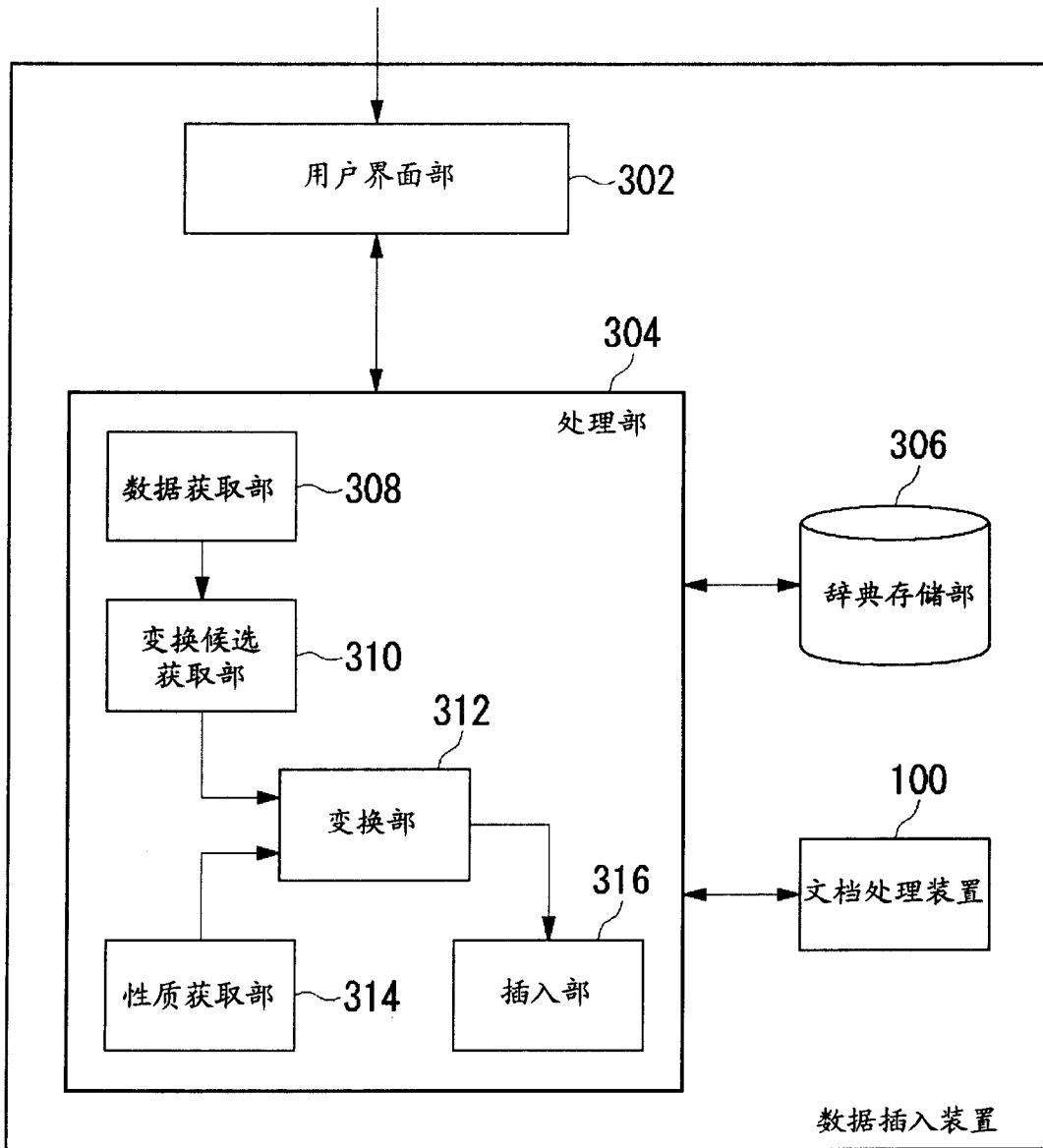


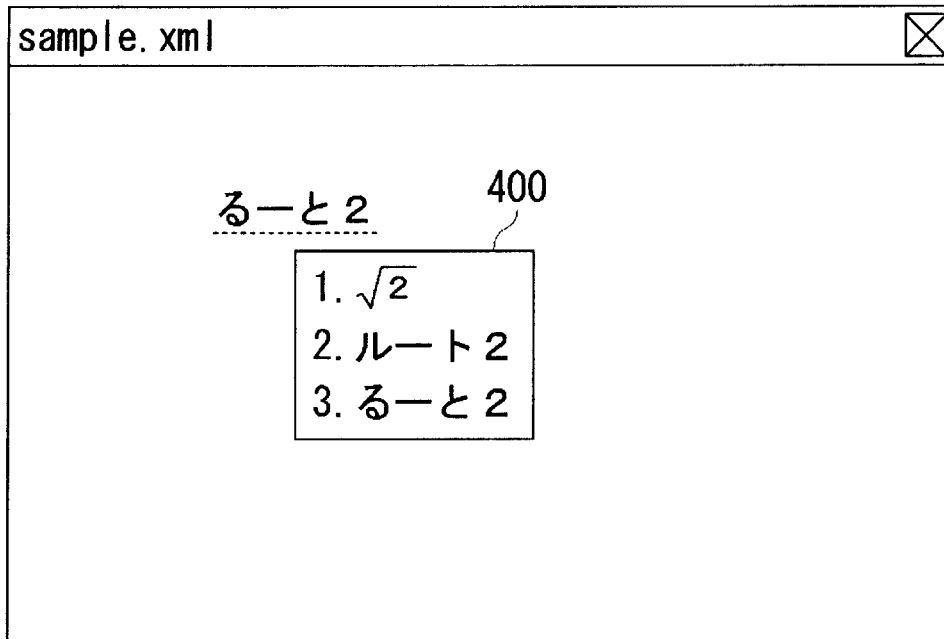
图 10



300

图 11

(a)



(b)

```

<chapter>
  <paragraph>
    <section>
      <msgrt>
        <mrow>2</mrow>
      </msgrt>
    </section>
  </paragraph>
</chapter>

```

(c)

```

<chapter>
  <paragraph>
    <section>
      ルー ト 2
    </section>
  </paragraph>
</chapter>

```

图 12

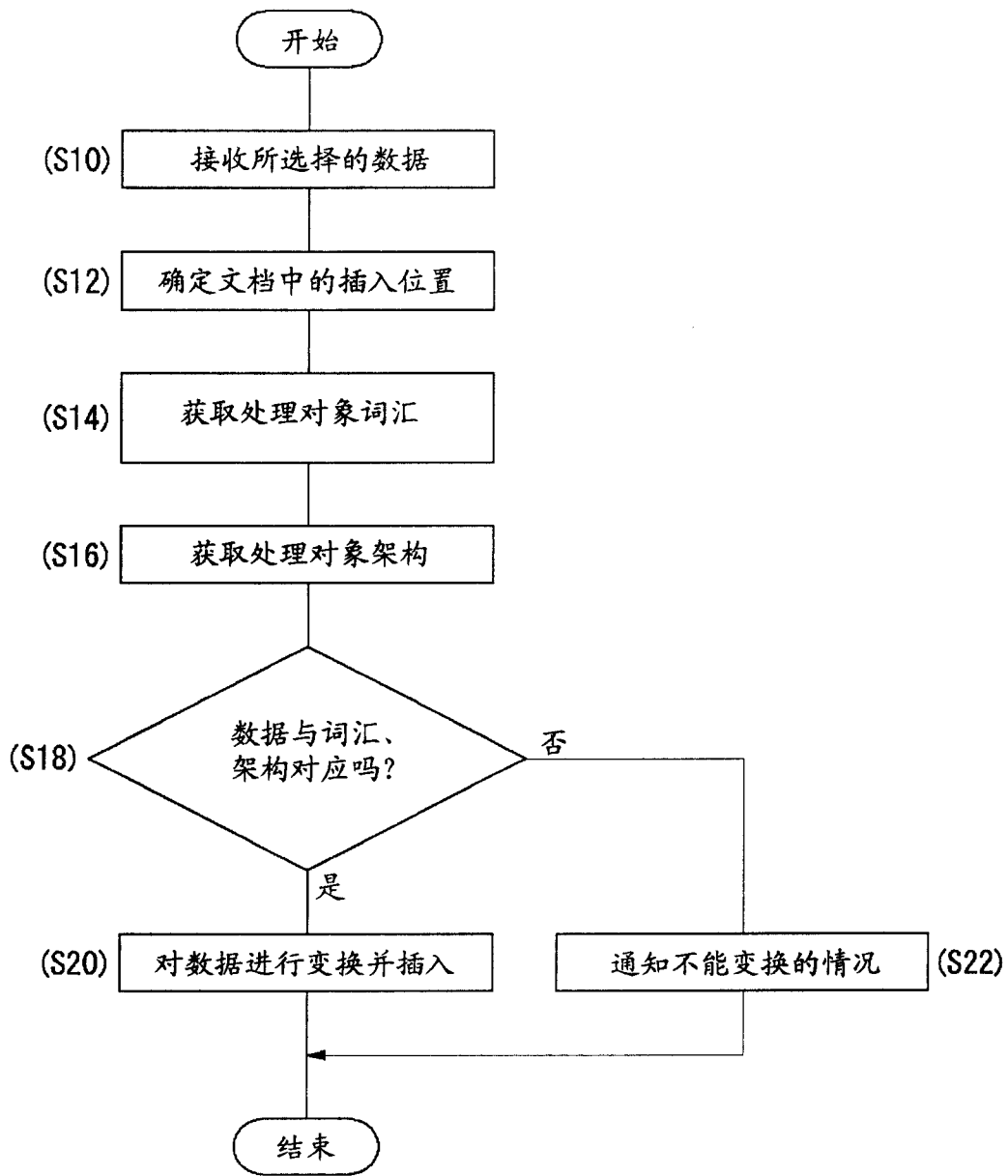


图 13

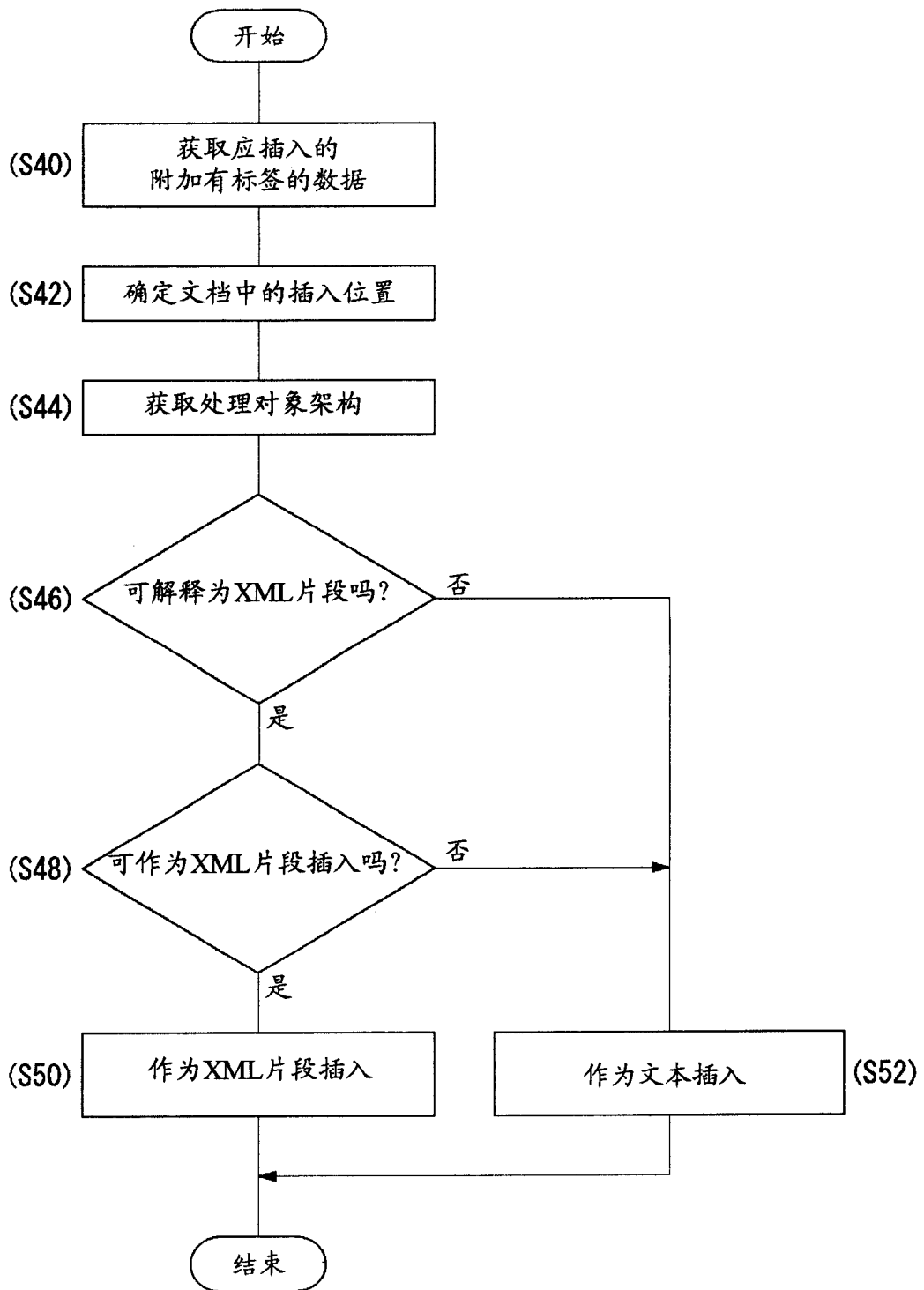


图 14

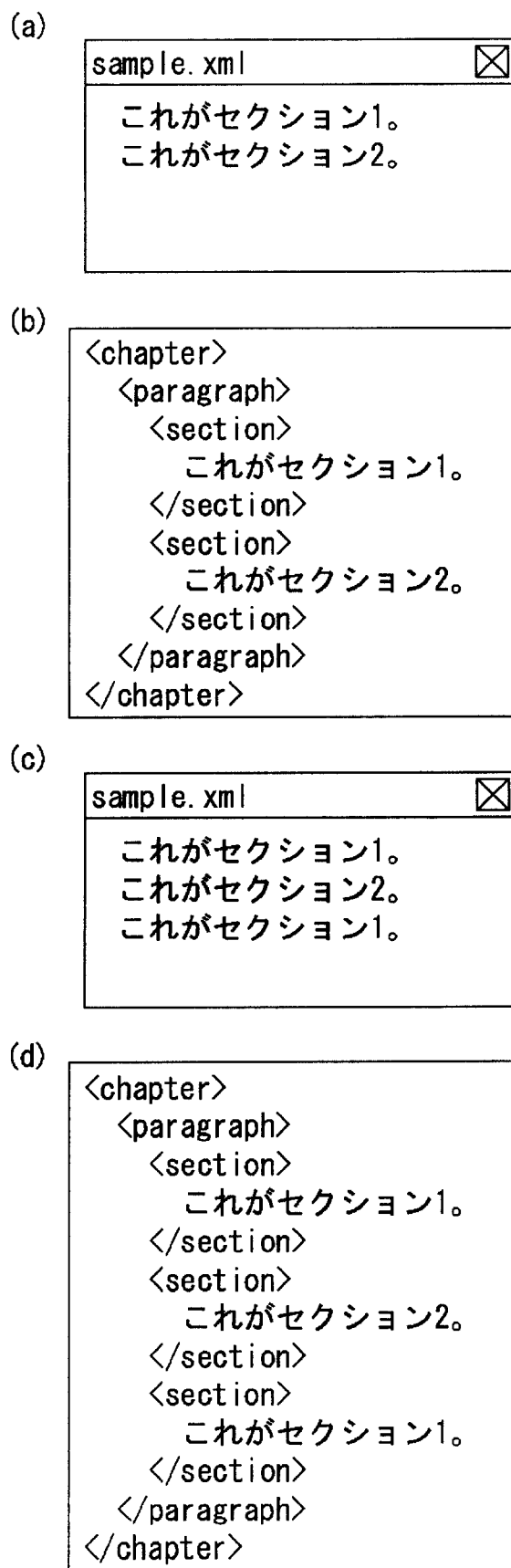


图 15

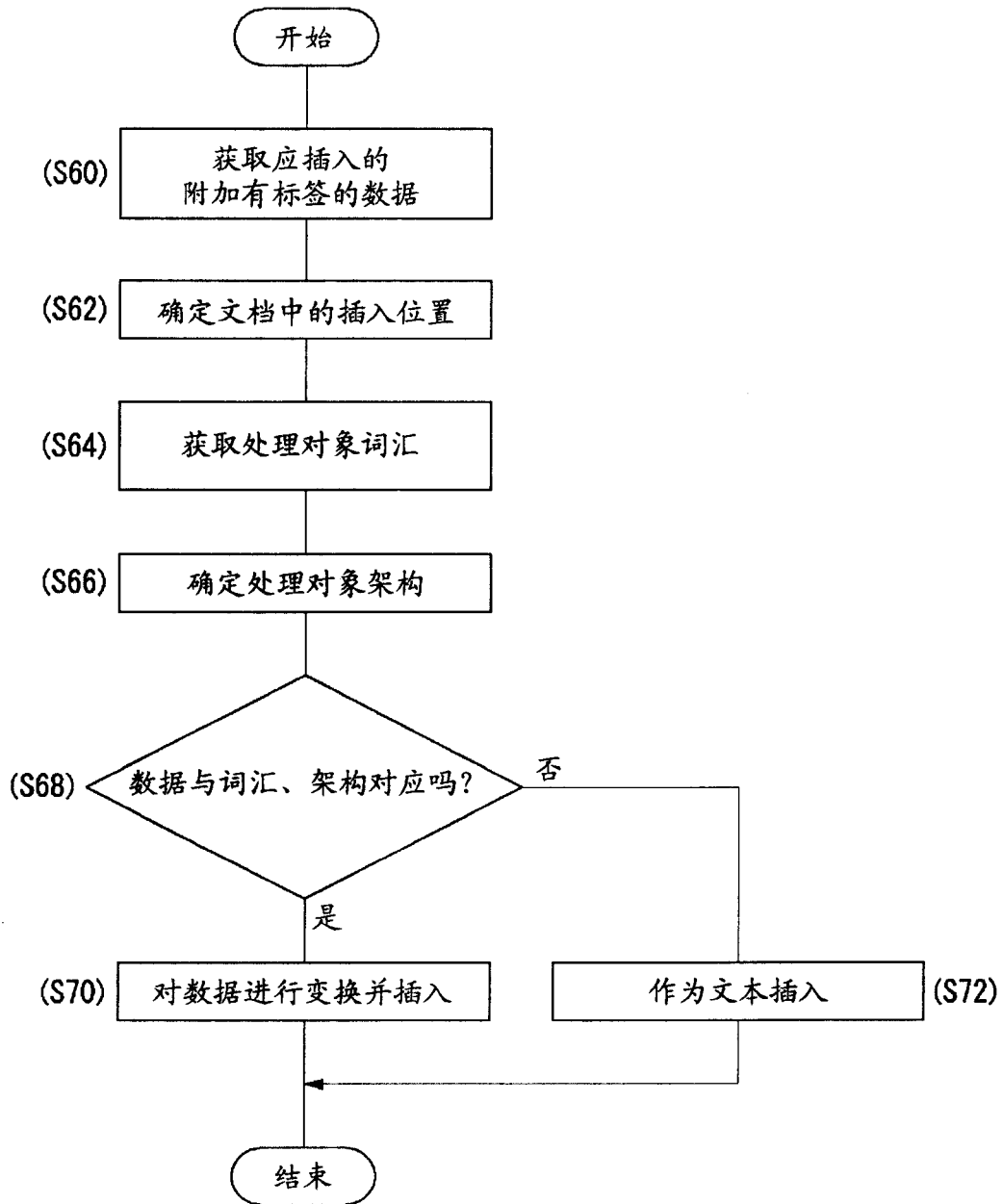
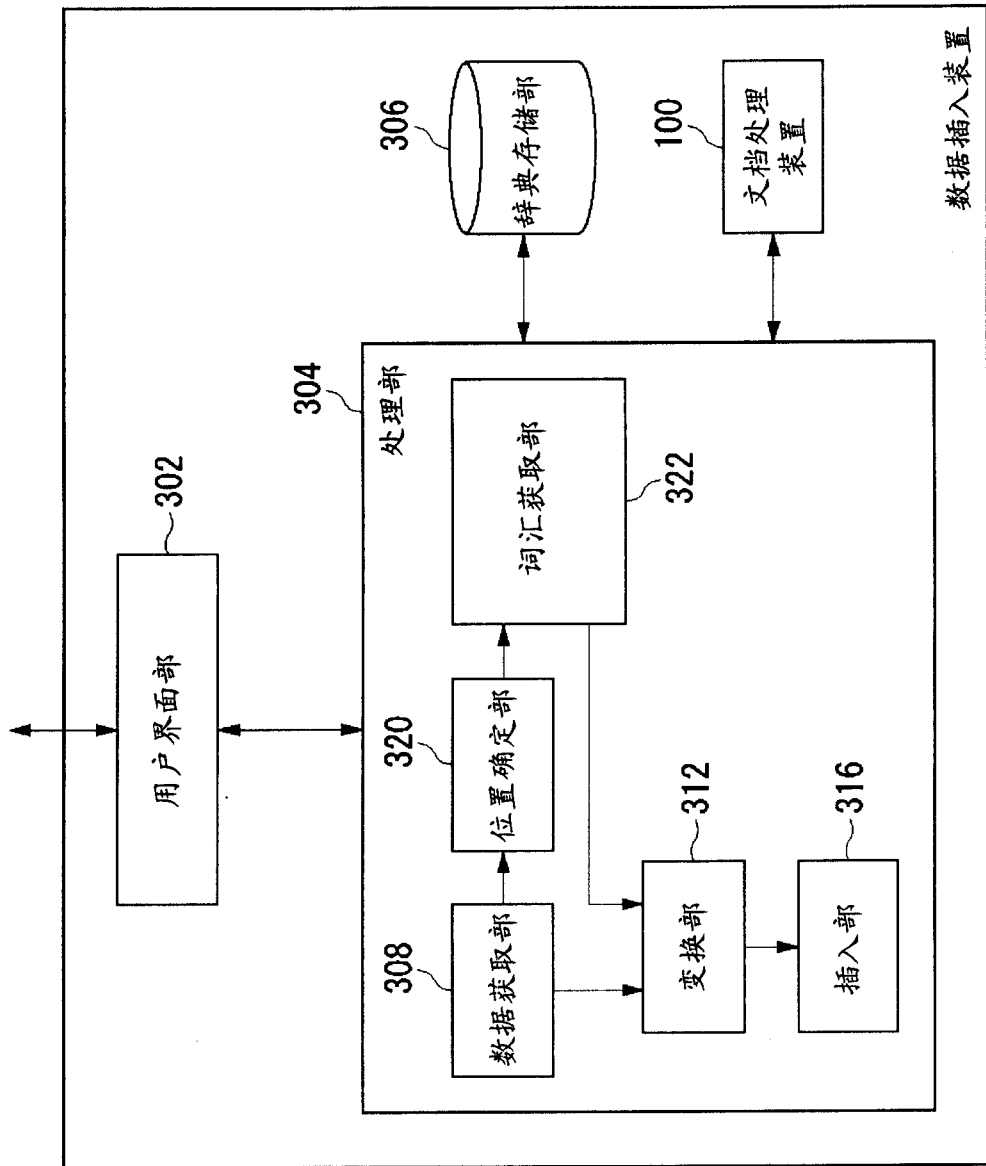


图 16



300

图 17

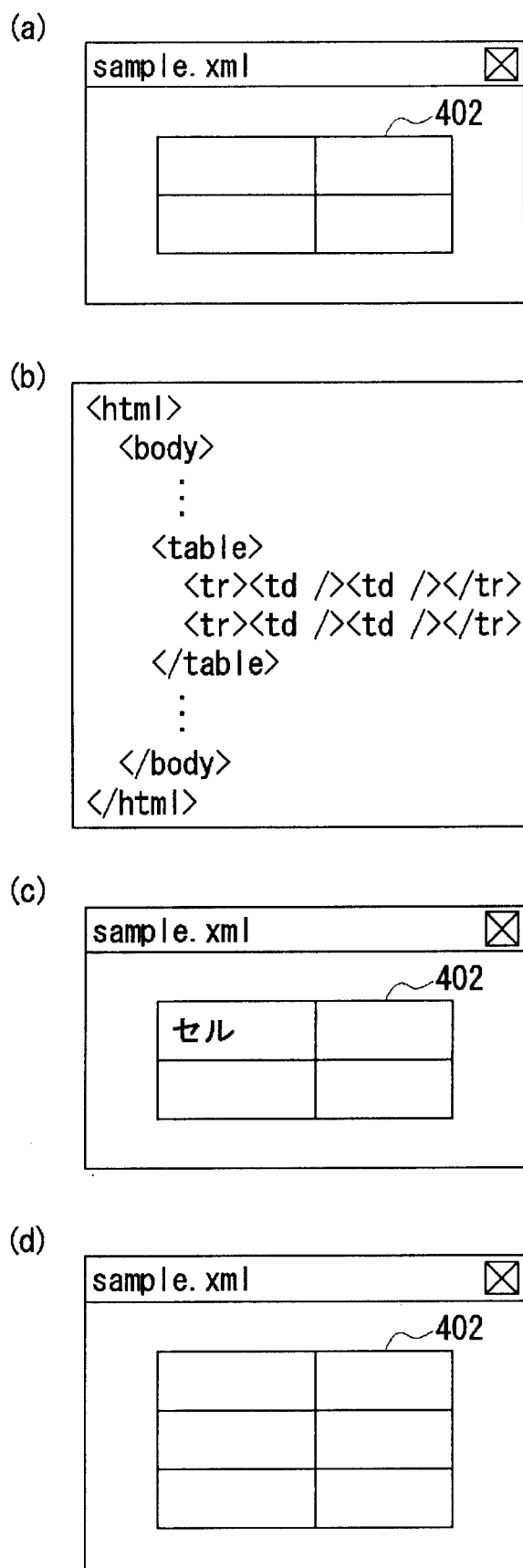


图 18

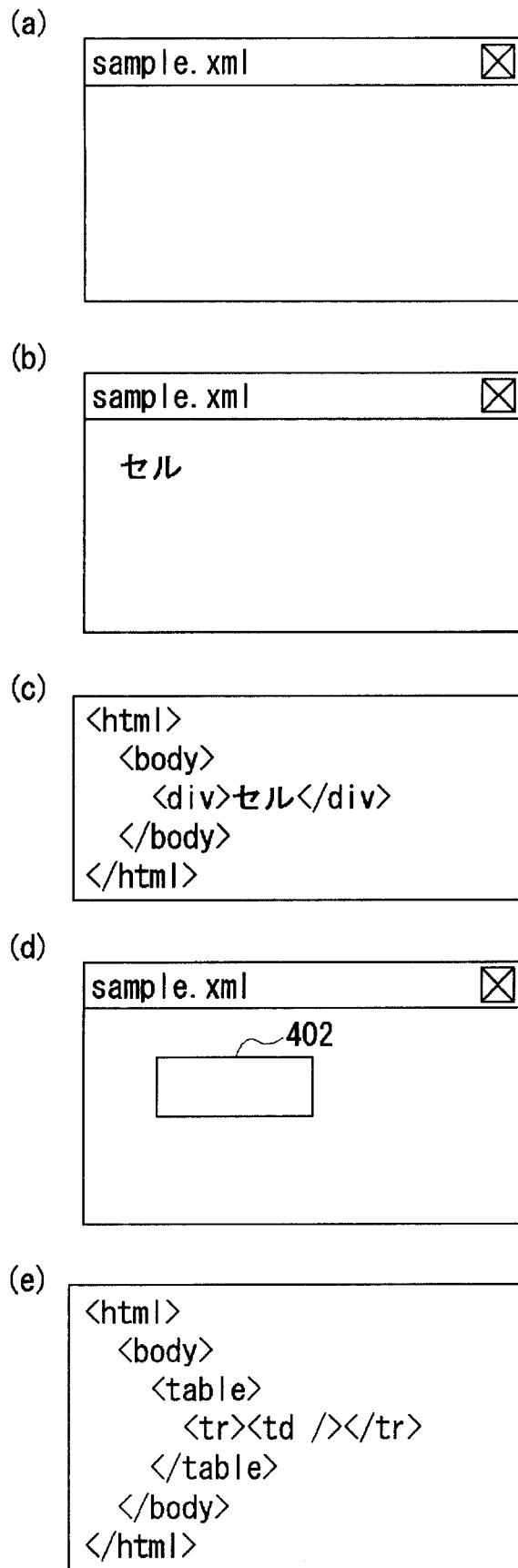
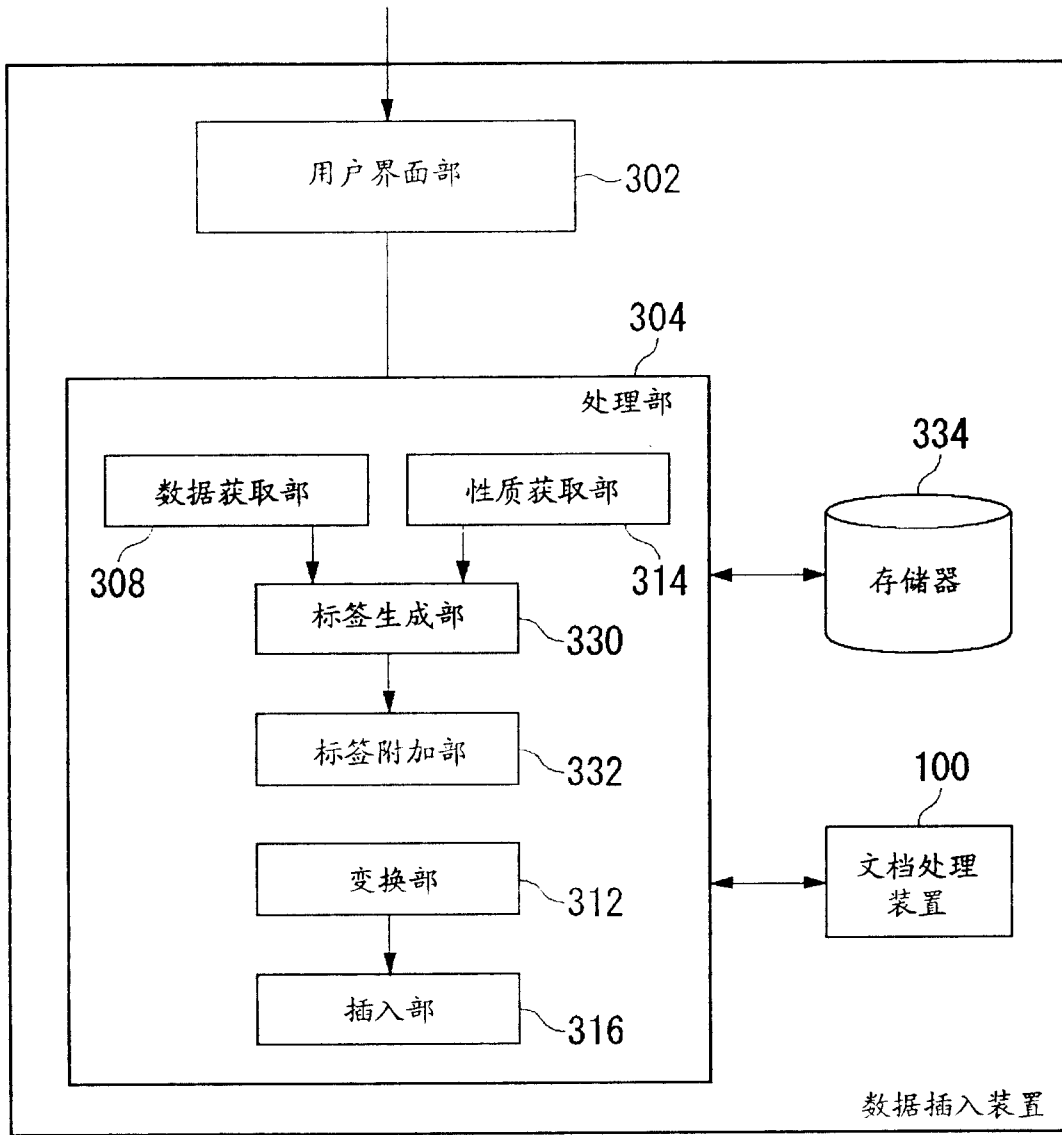


图 19



300

图 20

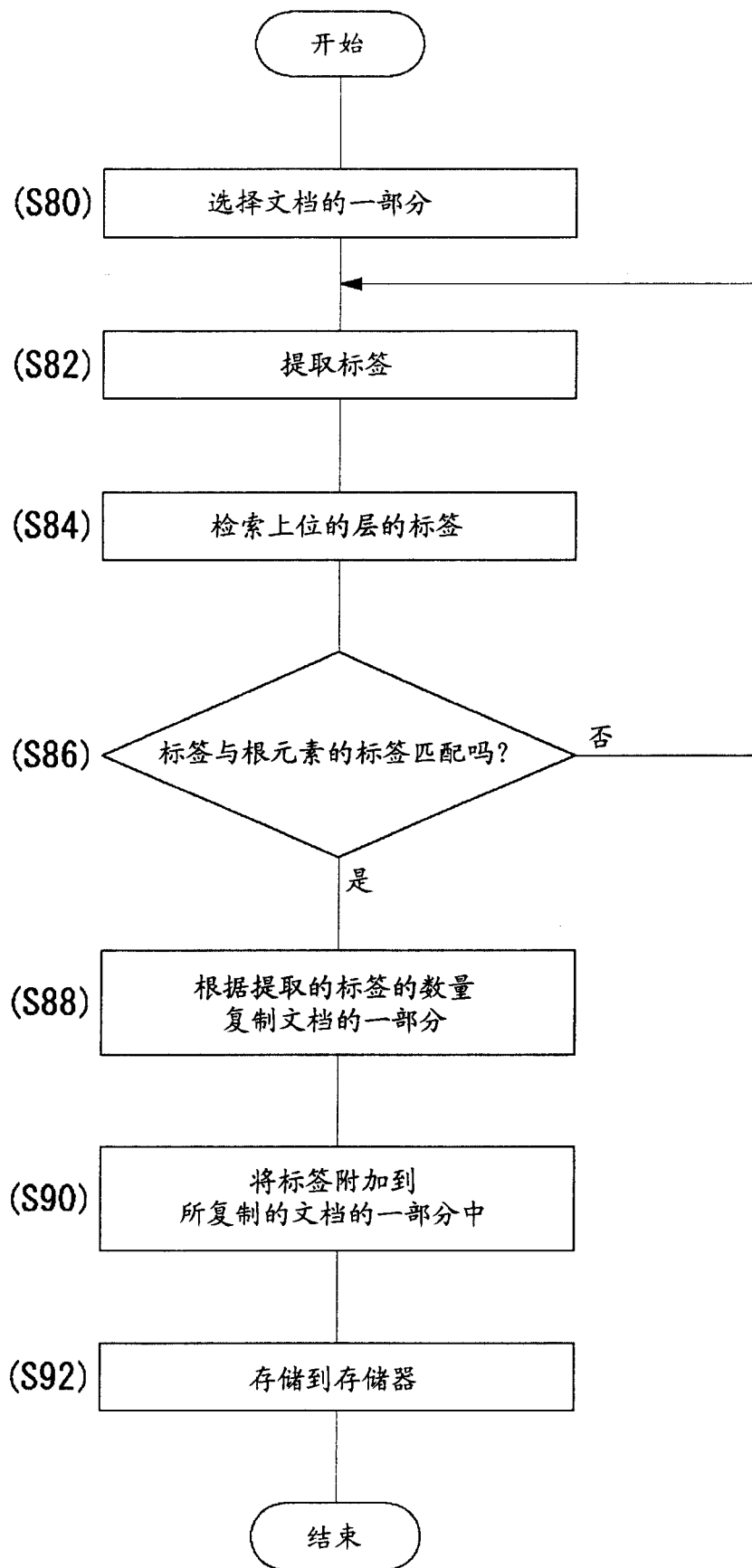


图 21

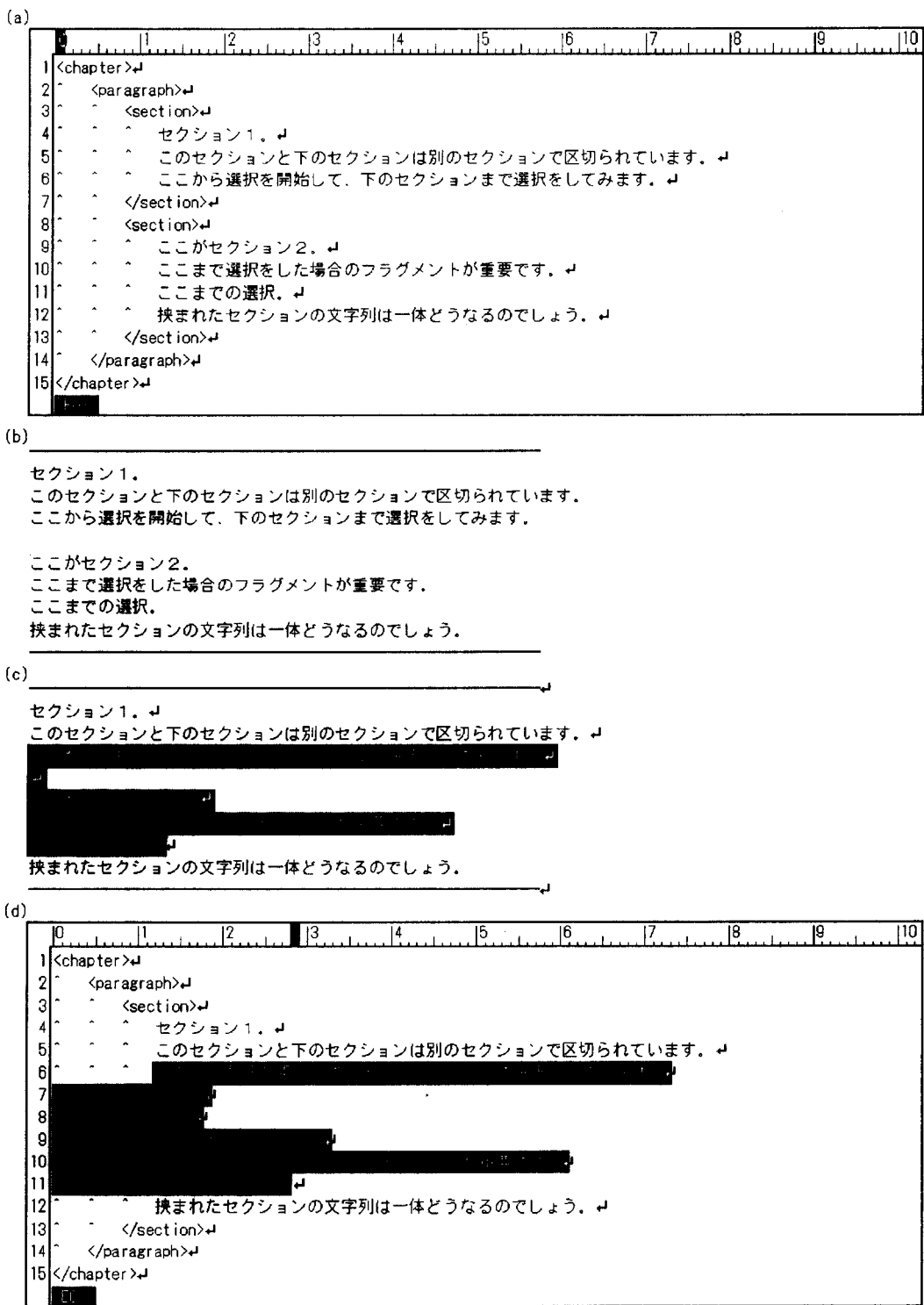


图 22