



## [12] 发明专利申请公开说明书

[21] 申请号 200410102294.5

[43] 公开日 2005 年 7 月 27 日

[11] 公开号 CN 1645368A

[22] 申请日 2004.12.16

[74] 专利代理机构 上海专利商标事务所有限公司  
代理人 李家麟

[21] 申请号 200410102294.5

[30] 优先权

[32] 2003.12.16 [33] EP [31] 03028668.6

[71] 申请人 软件股份公司

地址 联邦德国达姆施塔特

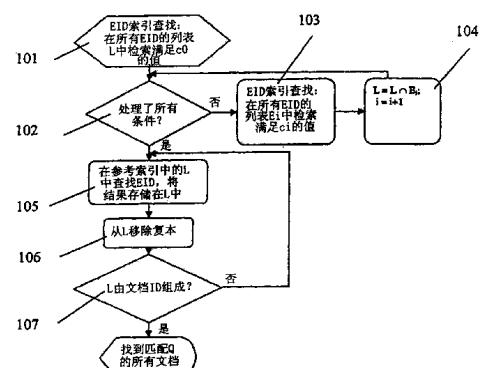
[72] 发明人 J·哈巴斯 K·H·-P·文克勤  
H·R·邵尼恩

权利要求书 2 页 说明书 5 页 附图 6 页

[54] 发明名称 用于查找数据库的方法和数据库

[57] 摘要

本发明涉及一种用于在分层结构数据库(10)的元素内进行查找的方法，其中每个元素都具有用于限定所述元素的属性的一个或多个分层结构节点，且其中所查找的元素的一个或多个节点必须实现一个或多个查找条件( $c_0, c_i, \dots$ )，该方法包括以下步骤：a. 选择多个节点并向每个所选节点分配唯一的标识符(EID)；b. 创建参考索引(20)，它允许为所选节点的每个唯一的标识符(EID)派生数据库(10)的相应元素；c. 对于一个或多个查找条件( $c_0, c_i, \dots$ )中的每一个：为实现各查找条件( $c_0, c_i, \dots$ )的节点检索起源节点的唯一标识符(EID)的组( $L, E_i$ )；d. 使这些组( $L, E_i$ )相交，以获得唯一标识符(EID)的最终组( $L$ )；以及e. 根据唯一标识符(EID)的最终组( $L$ )和参考索引(20)检索数据库(10)的所查找的元素。



1. 一种用于在分层结构数据库(10)的相同层次上的元素内进行查找的方法，其特征在于，每个元素都具有用于限定所述元素的属性的一个或多个分层结构节点，  
5 且其中所查找的元素的一个或多个节点必须实现一个或多个查找条件( $c_0, c_i, \dots$ )，该方法包括以下步骤：
- a. 选择多个节点并向每个所选节点分配唯一的标识符(EID)；
  - b. 创建参考索引(20)，它允许为所选节点的每个唯一的标识符(EID)派生数据库(10)的相应元素；
  - 10 c. 对于一个或多个查找条件( $c_0, c_i, \dots$ )中的每一个：  
为实现各查找条件的节点检索起源节点的唯一标识符(EID)的组(L, Ei)；
  - d. 使这些组(L, Ei)相交，以获得唯一标识符(EID)的最终组(L)；以及
  - e. 根据唯一标识符(EID)的最终组(L)和参考索引(20)检索所查找的元素。
2. 如权利要求1所述的方法，其特征在于，方法步骤c. 中的检索包括使用一个或多个查找索引(31, 32)，它们使起源节点的唯一标识符(EID)与可能是一个或多个查找条件( $c_0, c_i, \dots$ )的主题的节点相关联。
- 15 3. 如权利要求1或2所述的方法，其特征在于，参考索引(20)为每个唯一标识符(EID)直接指向其有关的数据库元素或者指向作为发起唯一标识符(EID)的节点起源的节点的唯一标识符(EID)。
- 20 4. 如权利要求1-3中的任一项所述的方法，其特征在于，如果在相同的分层层次上存在几个节点，则唯一标识符(EID)仅被分配给节点。
5. 如权利要求1-4中的任一项所述的方法，其特征在于，如果节点的子节点被预计是一个或多个查找条件( $c_0, c_i, \dots$ )的主题，则唯一标识符仅被分配给节点。
- 25 6. 如以上任一权利要求所述的方法，其特征在于，唯一标识符(EID)是唯一的数字。
7. 如以上任一权利要求所述的方法，其特征在于，数据库是基于XML的数据库(10)。
8. 一种具有多个元素的数据库(10)，其特征在于，包括：
- a. 多个分层结构节点，它们描述数据库(10)的元素的属性；
  - 30 b. 多个唯一标识符(EID)，它们被分配给所选的节点；
  - c. 参考索引(20)，它允许为所选节点的每个唯一标识符(EID)派生数据库(10)的相应元素；

d. 一个或多个查找索引(31, 32)，它们使起源节点的唯一标识符(EID)与可以是一个或多个查找条件( $c_0, c_i, \dots$ )的主题的节点相关联。

9. 如权利要求8所述的数据库(10)，其特征在于，参考索引(20)为每个唯一标识符(EID)直接指向其元素或者指向作为发起唯一标识符(EID)的节点起源的节点  
5 的唯一标识符(EID)。

10. 如权利要求8或9所述的数据库(10)，其特征在于，如果在数据库(10)的相同分层层次上存在几个节点，则唯一标识符(EID)仅被分配给节点。

11. 如权利要求8-10中的任一项所述的数据库(10)，其特征在于，如果所述节点的子节点被预计是一个或多个查找条件( $c_0, c_i, \dots$ )的主题，则唯一标识符仅被分  
10 配给节点。

12. 如权利要求8-11中的任一项所述的数据库(10)，其特征在于，唯一标识符(EID)是唯一的数字。

13. 如权利要求8-12中的任一项所述的数据库(10)，其特征在于，数据库(10)是基于XML的数据库(10)。

## 用于查找数据库的方法和数据库

### 5 技术领域

本发明涉及用于在分层结构数据库中进行查找的方法。此外，本发明涉及分层结构数据库，它允许其元素的方便查找。

### 现有技术

10 现今，数据库是用于管理大量数据的普通技术工具。分层结构数据库的特征是，按层次排列条目，例如可以被想象为包含根、叉、子叉等等的树形结构。例如，可以使用分层数据库管理图书馆的图书。数据库的每个元素，即每本图书，都具有几个属性，诸如标题、作者、图书中的章节等等。这些属性也可以具有子属性，例如作者的名字、他/她的地址、章节的文本等等。在分层结构数据库中，属性和子属性形成所谓的层次的节点。  
15

数据库的一个目的是方便其元素的检索。例如，图书馆的用户会需要在图书馆中根据其作者的姓和名快速找到特定的图书。为此，如现有技术中已知的，分层结构数据库常包括一个或多个索引，它们使特定节点的值与数据库的元素的标识符相关，从而允许根据被查元素的节点的条件非常快速地查找数据库的元素。这样，由于由数据库生成的索引本身产生查找结果，就不必检查数据库的每一个元素。这节省了时间和处理成本。  
20

图 1、11a 和 11b 说明了现有技术的查找技术：图 1 示出了图书 1-3 的基于 XML 的数据库的三个元素的简化实例。每个元素都包括几个属性，表示作者、姓和名、包含标题和文本的图书章节等等。图 11a 和 11b 呈现两个实例性索引表 201、202，  
25 一个用于图书馆中图书作者的名(图 11a, 201)，另一个用于作者的姓(图 11b, 202)。如果用户查找由姓 “Jones” 的作者撰写的图书，图 11b 的索引 202 将立即导向一结果，即图书 3 是所需的图书。

但是，数据库查询通常更加复杂并会包括两个或更多条件。例如，查找可以针对图 1 的数据库中作者 “Jim Miller” 的图书。使用现有技术的索引，图 11a 中对属性 “名”的索引 201 将首先被查找以产生图书 1、2 和 3 的一组。随后，查找对于属性 “姓”的图 11b 中的索引 202，以引向图书 1 和 2 的一组。满足查询的这两个条件的这两组图书的相交是图书 1 和 2。但是，图 1 中数据库元素的检查示出，

仅图书 2 是正确的结果。原因在于，图书可以由几个作者一起撰写，其中一个的名是“Jim”而另一个的姓是“Miller”。

克服该困难的一种方法是提供复合索引，例如通过使作者的组合的姓名与数据库的图书相关。但是，如果使用其它查找标准，而不是相当有限数量的作者名字，  
5 例如如果所需图书的章节标题和该章节中文本的单词是查找标准，则这种复合索引将导致海量的索引条目。这样，由于章节文本中的每个单词将与标题一起必须被组合用于复合索引的条目，明显的是，这种方法将再次导致数据库查询的很长的查找时间和较高的处理成本。

因此，本发明的问题在于提供一种用于查找数据库的方法以及一种数据库本身，它们即便对于复合查询也允许快速和有效的查找，以克服现有技术的上述缺点。  
10

### 发明内容

通过一种方法解决了该问题，该方法用于在分层结构数据库的元素内进行查找，其中每个元素都具有用于限定所述元素的属性的一个或多个分层结构节点，且  
15 其中所查找的元素的一个或多个节点必须实现一个或多个查找条件，该方法包括以下步骤：

- a. 选择多个节点并向每个所选节点分配唯一的标识符；
- b. 创建参考索引，它允许为所选节点的每个唯一的标识符派生数据库的相应元  
素；
- c. 对于一个或多个查找条件中的每一个：

为实现各查找条件的节点检索起源(ancestor)节点的唯一标识符的组；

d. 使这些组相交，以获得唯一标识符的最终组；以及

e. 根据唯一标识符的最终组和参考索引检索数据库的所查找的元素。

与其中查找索引直接关于数据库的被查找元素的现有技术相反，本发明使用唯一标识符的概念。几十在数据库的分层的不同层次处，这也允许管理具有无限数量的条件的复杂查询。此外，尽管具有超过一个条件的复杂查询的复合索引的上述使用要求基本量的预先计划，但本发明的方法提供迄今更灵活的方法来用一个或多个条件获得查找结果。  
25

步骤 c. 的检索包括使用一个或多个查找索引，它们使起源节点的唯一标识符与可能是一个或多个查找条件的对象的节点相关联。由于可以更快速地查找这种索引，就可以在相对较短的时间内获得唯一标识符的组以及作为这些组的相交的最终组。  
30

较佳地，参考索引为每个唯一标识符直接指向其有关的数据库元素或者指向作为发起唯一标识符的节点起源的节点的唯一标识符。结果，可以直接或间接地(即参考索引的循环使用)获得与方法步骤 d. 的唯一标识符相交相对应的数据库元素。

为了减少方法的总开销，较佳地，如果在相同的分层层次上存在几个节点，则唯一标识符仅被分配给节点。这不限制上述查找方法的可应用性。代替使用仅出现一次的起源节点的唯一标识符，出现超过一次的最近起源节点可用来获得查找的相同结果。结果，对于使仅出现一次的节点的唯一标识符相关联，查找示例也是不必要的，较佳地，如果所述节点的子节点被预计是一个或多个查找条件的对象，则唯一标识符仅被分配给节点。

10 在较佳实施例中，唯一标识符是唯一的数字且数据库是基于 XML 的数据库。

根据另一个方面，本发明涉及具有多个元素的数据库，它包括：

- a. 多个分层结构节点，它们描述数据库的元素的属性；
- b. 多个唯一标识符，它们被分配给所选的节点；
- c. 参考索引，它允许为所选节点的每个唯一标识符派生数据库的相应元素；

15 d. 一个或多个查找示例，它们使起源节点的唯一标识符与可能是一个或多个查找条件的主题的节点相关联。

此外，从属权项涉及数据库的较佳实施例。

## 附图概述

20 在以下详细描述中，将参考附图描述本发明的较佳实施例，其中：

图 1：用于图书 1-3 的分层数据库的实例；

图 2：图 1 的数据库节点的唯一标识符的实例性分配；

图 3：用于图 1 和 2 的数据库的唯一标识符的参考索引；

图 4 和 5：可能是一个或更多查找条件的主题的节点的查找示例的实例；

25 图 6：描述本发明的较佳实施例的流程图；

图 7：用于实现查找示例的 XML 概要的实例；

图 8：根据本发明的较佳实施例构建数据库的过程的示意性表示；

图 9：样本 XML 文档和用于说明唯一标识符的适当分配的相应 XML 概要；

图 10：XML 文档，说明何时应用语义将省去唯一标识符的分配；以及

30 图 11a； b：根据现有技术用于查找数据库的两个查找示例。

## 具体实施方式

以下，参考分层结构 XML 文档形式的数据库描述根据本发明的方法和数据库的较佳实施例。但是，可以理解，本发明不限于用于构建文档的该具体标记语言，而是可以应用于任何类型的具有树形分层结构的数据。此外，应注意，本发明可以与现有技术的查找方法和数据库组合，特别是本发明的介绍部分中所描述的技术。

5 图 1 示出了包含 3 本图书(图书 id=1, 2, 3) 的数据库 10 的已描述的简化实例。以下，参考数据库 10 中对一本图书的查询描述根据本发明的方法和数据库，这本图书例如具有含单词“standards”的标题的章节，同时该章节的文本包含单词“namespaces”。

为了执行查找，数据库 10 具备唯一的标识符，它可以是称作 EID 的唯一数字 10 的形式，用于明确地识别数据库 10 的所选节点。这在图 2 中示出。如图所示，数据库 10 的三本图书的每一个章节都具备唯一的 EID。一旦分配了 EID，就可以建立图 3 所示的参考索引 20，它将每个分配的 EID 与数据库的各元素(即本实例中各图书的文档 id)相关联。

但是，参考索引 20 不必直接指向数据库中的元素。或者，它可以使第一 EID 15 与属于一节点的第二 EID 相关联，其中上述节点是由第一 EID 标识的节点的起源。在这种情况下，可以通过循环地访问参考检索 20 找到数据库的各元素。

在进一步的预备步骤中，建立一个或多个查找索引，其使得可以是查询的查找 20 条件的主题的节点值与这些节点的起源的 EID 相关联。这种 EID 索引的实例在图 4 和 5 中以表的形式示出，其中图 4 的 EID 索引 31 使得节点“文本”中的单词与各章节的 EID(即节点文本的起源节点)相关联。按照相同的方式，图 5 的 EID 索引 32 使得节点“标题”的值与各章节的 EID 索引相关联。图 7 中示出符合 EID 索引的定义的实例性 XML 概要。

一旦已通过提供参考索引 20 和 EID 索引 31、32 准备了数据库 10，就可以进行 25 最初描述的对其文本中具有单词“namespaces”和其标题中具有单词“standards”的章节的图书复杂查询。为此所执行的步骤在图 6 的流程图中进行说明。

在步骤 101 中，调查与第一查找条件 c0 相对应的 EID 紴并检索包含满足条件 c0 的节点的 EID 的列表 L。在本实例中，条件 c0 是：单词“namespaces”出现于所查找的图书的章节的文本中。这产生包含 EID57 和 66(比较图 4)的列表 L。

在步骤 102 中，检查是否存在要满足的进一步的查找条件 ci。如果有，则方法 30 以步骤 103 继续，其中调查与查找条件 ci 相对应的 EID 紴以检索 EID 的进一步的列表 Ei。在本实例中，第二查找条件是：单词“standards”出现于章节的标题中。使用图 5 的 EID 紹，就获得包含 EID57 和 69 的列表。在步骤 104 中，使两

个列表 L 和 E<sub>i</sub> 相交，在这种情况下形成仅包含 EID57 的列表。

如果没有要满足的进一步的条件，方法进行到步骤 105，其中使用参考索引 20 将所产生的列表 L 的 EID 转换成数据库的元素。这需要在步骤 107 中判断以执行对参考索引 20 的循环访问，以便解析不直接参考数据库元素而是起源 EID 的 EID，

5 如上所述。在步骤 106 中移除所生成的数据库元素的复本。

在本实例中，必须解析单个 EID57，这通向具有文档 id2 的图书从而仅通向正确的结果，因为仅图书 2 包含章节标题中的单词“standards”和该章节的文本中的单词“namespaces”，如查询所要求的那样（比较图 1 中数据库 10 的 XML 文档）。

参考特定简单实例说明的所述方法也可以用于任意数量节点的查找条件的组合而不需要预先计划这些组合。这与组合索引的上述现有技术的方法相反。此外，条件可以涉及数据库的不同层次处的节点，这增加了执行复杂查询的灵活性。在数据库内容（即 XML 文档）修改的情况下，所述查找方法是稳定的，因为 EID 被分配给节点而不是这些节点的值，它们在文档更新时会变化。

按上述方式执行复杂查询要求准备数据库。如果结构 XML 文档用于数据元素，可以进行必要处理，如图 8 所示。存储具有要构建的 EID 索引的详细说明的 XML 概要 50 和包含实际数据的 XML 文档 51。存储过程包括创建一个或多个 EID 索引 52，创建参考索引 53 和存储文档 54 本身。一旦 XML 文档已按这种方式存储，就准备复杂查询的上述有效处理。

本发明的进一步重要方面是将 EID 分配给分层结构数据库的节点。为了减少用于处理复杂查询和存储 XML 文档的总开销，不必将 EID 分配给层次中的每个节点。优先限制可能出现一次以上的节点的 EID 分配。以下参考图 9 对此进行进一步说明。

尽管图 9 的上部呈现用于单本图书的样本 XML 文档，图 9 的第二部分显示基础 XML 概要。如图所示，图书可以具有几个作者（比较图 9 的 XML 概要中的行“<xs element ref=”author” maxOccurs=”unbounded”/>）。但是，每个作者都具有包含单个姓和单个名的单个名字。因此，有意义的 EID 分配考虑数据库结构的这种属性并仅将 EID 分配给作者而不分配给名字。

减少数据库中总开销的另一个方法是仅为可能是查找条件的主题的节点准备 EID 索引。例如，对于由图 10 中的 XML 文档表示的图书，用关于页数和章节的标题的条件的查找的相当不可能的。因此，不必创建用于使标题值与章节 EID 相关的 EID 索引和/或使 numberOfPages 值与章节的 EID 相关联的 EID 紴引。其它条件也是可以想象的，其中应用语义可用于演绎出可以省去 EID 分配的特定层次和相应 EID 紴引的创建。

```

<?xml version="1.0"?>
<book id="1">
  <author> <firstname>John </firstname> <lastname>Miller</lastname> </author>
  <author> <firstname>Jim </firstname> <lastname> Smith </lastname> </author>
  <title> XML in electronic business </title>
  <chapter number="1"> <title> electronic business </title> <text> </text> </chapter>
  <chapter number="2"> <title> XML basis </title> <text> </text> </chapter>
  <chapter number="3"> <title> web services </title> <text> </text> </chapter>
</book>
and
<?xml version="1.0"?>
<book id="2">
  <author> <firstname> Jim </firstname> <lastname> Miller </lastname> </author>
  <author> <firstname> Laura </firstname> <lastname> Smith </lastname> </author>
  <title> More about XML </title>
  <chapter number="1"> <title> XML standards </title> <text> The most basic standard
  documents are XML 1.0, XML namespaces, ... </text> </chapter>
  <chapter number="2"> <title> advanced XML </title> <text> </text> </chapter>
  <chapter number="3"> <title> the future of XML </title> <text> </text> </chapter>
</book>
and
<?xml version="1.0"?>
<book id="3">
  <author> <firstname> Mary </firstname> <lastname> Jones </lastname> </author>
  <author> <firstname> Jim </firstname> <lastname> Smith </lastname> </author>
  <title> XML in databases </title>
  <chapter number="1"> <title> XML basics </title> <text> the extensible markup language
  uses namespaces to ... </text> </chapter>
  <chapter number="2"> <title> XML standards </title> <text> ... </text> </chapter>
  <chapter number="3"> <title> XML in databases </title> <text> ... </text> </chapter>
</book>

```

图 1

1

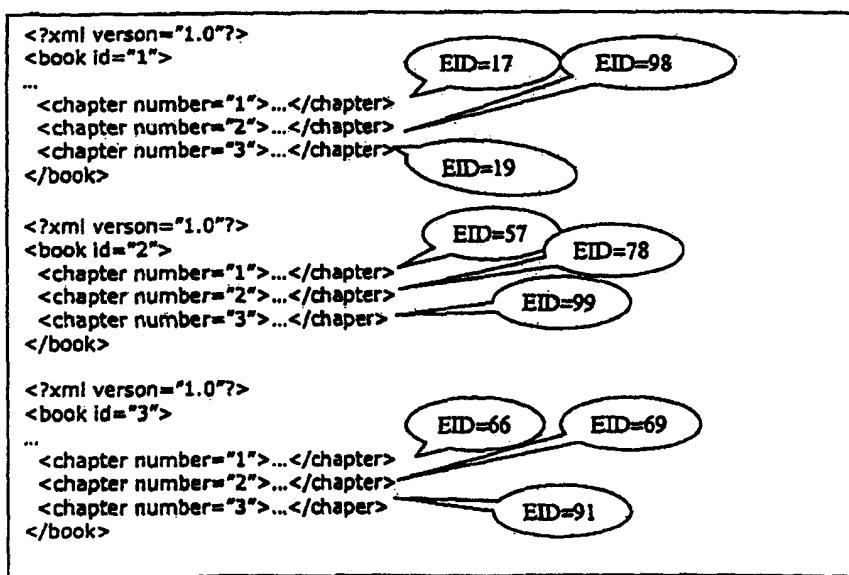


图 2

2

20

FID	指向
17	文档ID1
19	文档ID1
57	文档ID2
66	文档ID3
69	文档ID2
78	文档ID2
91	文档ID3
98	文档ID1
99	文档ID2

图 3

31

节点“文本”中的单词	
1.0	57
are	57
basic	57
documents	57
extensible	66
language	66
markup	66
most	57
namespaces	57, 66
standard	57
the	57, 66
to	66
uses	66
XML	57

图 4

32

节点“标题”中的单词	
advanced	78
basics	66
basis	98
business	17
databases	91
electronic	17
future	99
in	91
of	99
services	19
standards	57, 69
the	99
web	19
XML	57, 66, 69, 78, 91, 98, 99

图 5

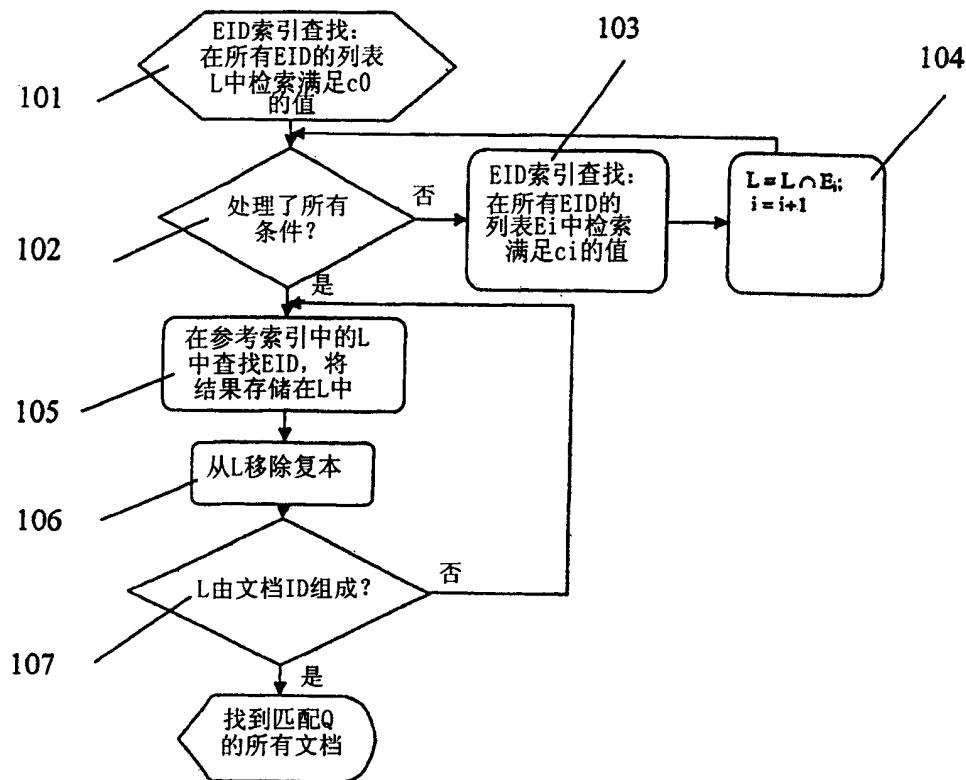


图 6

```

<xsd:element name = "title" type = "xsd:string">
  <xsd:annotation>
    <xsd:appinfo>
      <tsd:elementInfo>
        <tsd:physical>
          <tsd:native>
            <tsd:index>
              <tsd:text>
                <tsd:refers>/book/chapter</tsd:refers>
              </tsd:text>
            </tsd:index>
          </tsd:native>
        </tsd:physical>
      </tsd:elementInfo>
    </xsd:appinfo>
  </xsd:annotation>
</xsd:element>
  
```

图 7

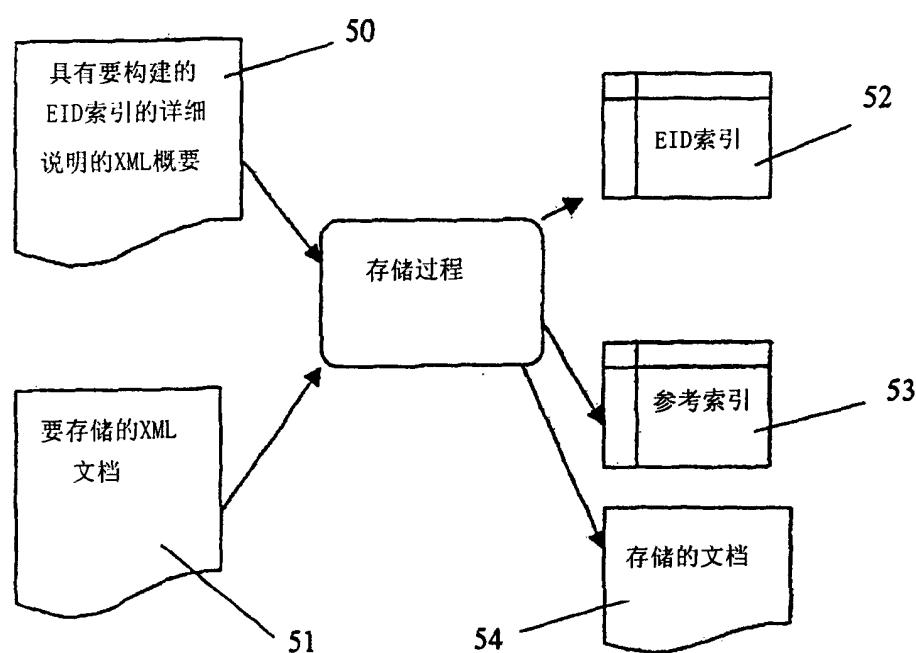


图 8

sample XML document

```
<book>
  <author>
    <name>
      <firstname>Joe</firstname>
      <lastname>Miller</lastname>
    </name>
  </author>
  <author>
    <name>
      <firstname>Jim</firstname>
      <lastname>Smith</lastname>
    </name>
  </author>
</book>
```

corresponding XML schema:

```
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xss:element name="author">
    <xss:complexType>
      <xss:sequence>
        <xss:element ref="name"/>
      </xss:sequence>
    </xss:complexType>
  </xss:element>
  <xss:element name="book">
    <xss:complexType>
      <xss:sequence>
        <xss:element ref="author" maxOccurs="unbounded"/>
      </xss:sequence>
    </xss:complexType>
  </xss:element>
  <xss:element name="firstname">
    <xss:simpleType>
      <xss:restriction base="xs:string">
        <xss:enumeration value="Jim"/>
        <xss:enumeration value="Joe"/>
      </xss:restriction>
    </xss:simpleType>
  </xss:element>
  <xss:element name="lastname">
    <xss:simpleType>
      <xss:restriction base="xs:string">
        <xss:enumeration value="Miller"/>
        <xss:enumeration value="Smith"/>
      </xss:restriction>
    </xss:simpleType>
  </xss:element>
  <xss:element name="name">
    <xss:complexType>
      <xss:sequence>
        <xss:element ref="firstname"/>
        <xss:element ref="lastname"/>
      </xss:sequence>
    </xss:complexType>
  </xss:element>
</xss:schema>
```

```
<book>
  <chapter>
    <title>Introduction</title>
    <numberOfPages>32</numberOfPages>
  </chapter>
  <chapter>
    <title>Conclusions</title>
    <numberOfPages>3</numberOfPages>
  </chapter>
</book>
```

图 10

201

Jim	1,2,3
John	1
Laura	2
Mary	3

图 11a

202

Jones	3
Miller	1,2
Smith	1,2,3

图 11b