(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103377175 A
(43)申请公布日 2013.10.30

(21)申请号 201210126487.9

(22)申请日 2012.04.26

(71)申请人 SAP 股份公司
地址 德国瓦尔多夫

(72)发明人 彭圣才 富亮 胡瑛琨 闵贤龙 张腾飞

(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 邵亚丽

(51)Int.Cl.
G06F 17/22(2006.01)

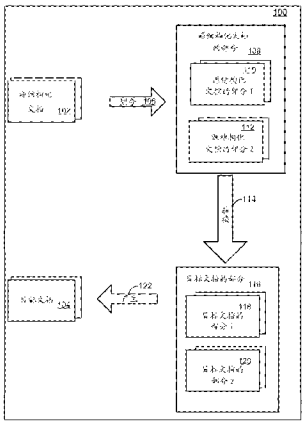
权利要求书4页 说明书13页 附图12页

(54)发明名称

基于分割的结构化文档转换

(57)摘要

描述了用于把源结构化文档转换为目标文档的系统和方法的各种实施例。接收转换源结构化文档的请求。将源结构化文档划分为多个部分。将转换规则逐一应用于源结构化文档的部分，以获得目标文档的部分。基于目标文档的所获得的部分，产生目标文档。



CN 103377175 A

1. 一种计算机实施的用于把源结构化文档转换为目标文档的方法，所述方法包含：

接收转换源结构化文档的请求；

接收对源结构化文档中的第一节点以及更多的一个或多个节点的选择，第一节点被定义为第一占位符，并且更多的一个或多个节点被定义为更多的一个或多个占位符；

基于所接收的选择，由计算机的处理器把源结构化文档划分为源结构化文档的第一部分以及更多的一个或多个部分，第一占位符指示源结构化文档的第一部分，更多的一个或多个占位符指示源结构化文档的更多的一个或多个部分；

基于转换规则，由计算机的处理器转换源结构化文档的第一部分，以获得目标文档的第一部分；

基于转换规则，由计算机的处理器转换源结构化文档的更多的一个或多个部分，以获得目标文档的更多的一个或多个部分；和

由计算机的处理器，基于所获得的目标文档的第一部分和所获得的目标文档的更多的一个或多个部分，产生目标文档。

2. 如权利要求 1 所述的计算机实施的方法，其中，划分源结构化文档包括：

基于所定义的第一占位符以及更多的一个或多个占位符，由计算机的处理器把源结构化文档划分为源结构化文档的第一部分以及更多的一个或多个部分。

3. 如权利要求 2 所述的计算机实施的方法，还包含：

在计算机的存储器中存储结构化文档的第一部分作为第一结构化文档文件，第一占位符指示第一结构化文档文件；

在计算机的存储器中存储源结构化文档的更多的一个或多个部分作为更多的一个或多个结构化文档文件，更多的一个或多个占位符指示更多的一个或多个结构化文档文件；和

把第一占位符、更多的一个或多个占位符和源结构化文档的剩余部分存储在源结构化文档索引文件中。

4. 如权利要求 3 所述的计算机实施的方法，其中，转换结构化文档的第一部分包括：

基于转换规则，由计算机的处理器转换源结构化文档索引文件，以获得中间结果文件，中间结果文件包括第一占位符、更多的一个或多个占位符，以及源结构化文档的剩余部分的转换结果；

由计算机的处理器识别中间结果文件中所包括的第一占位符；

从计算机的存储器检索被识别的第一占位符指示的第一结构化文档文件；和

基于转换规则，由计算机的处理器转换第一结构化文档文件，以获得目标文档的第一部分。

5. 如权利要求 4 所述的计算机实施的方法，其中，转换结构化文档的更多的一个或多个部分包括：

由计算机的处理器识别中间结果文件中所包括的更多的一个或多个占位符；

从计算机的存储器检索被更多的一个或多个占位符指示的更多的一个或多个结构化文档文件；和

基于转换规则，由计算机的处理器转换更多的一个或多个结构化文档文件，以获得目标文档的更多的一个或多个部分。

6. 如权利要求 1 所述的计算机实施的方法，其中，产生目标文档包括：

由计算机的处理器利用所获得的目标文档的第一部分替换源结构化文档的第一部分；和

由计算机的处理器利用所获得的目标文档的更多的一个或多个部分替换源结构化文档的更多的一个或多个部分。

7. 一种计算机实施的用于把源结构化文档转换为目标文档的方法，所述方法包含：

接收对源结构化文档中的第一节点的选择，所选择的第一节点被定义为第一占位符，且指示源结构化文档的第一部分；

接收对源结构化文档中的更多的一个或多个节点的选择，所选择的更多的一个或多个节点被定义为更多的一个或多个占位符，且指示源结构化文档的更多的一个或多个部分；

基于所定义的第一占位符和所定义的更多的一个或多个占位符，由计算机的处理器把源结构化文档划分为第一部分以及更多的一个或多个部分；

把第一占位符、更多的一个或多个占位符以及源结构化文档的剩余部分存储在源结构化文档索引文件中；

基于转换规则，由计算机的处理器转换源结构化文档索引文件，以获得中间结果文件，中间结果文件包括第一占位符、更多的一个或多个占位符，以及源结构化文档的剩余部分的转换结果；

由计算机的处理器识别中间结果文件中所包括的第一占位符；

从计算机的存储器中检索被第一占位符指示的源结构化文档的第一部分；

基于转换规则，由计算机的处理器转换源结构化文档的第一部分，以获得目标文档的第一部分；

由计算机的处理器识别中间结果文件中所包括的更多的一个或多个占位符；

从计算机的存储器中检索被更多的一个或多个占位符指示的源结构化文档的更多的一个或多个部分；

基于转换规则，由计算机的处理器转换源结构化文档的更多的一个或多个部分，以获得目标文档的更多的一个或多个部分；和

由计算机的处理器基于目标文档的第一部分、目标文档的更多的一个或多个部分以及源结构化文档的剩余部分的转换结果产生目标文档。

8. 如权利要求 7 所述的计算机实施的方法，还包含：

在计算机的存储器中存储源结构化文档的第一部分作为第一结构化文档文件，第一占位符指示第一结构化文档文件；和

在计算机的存储器中存储源结构化文档的更多的一个或多个部分作为更多的一个或多个结构化文档文件，更多的一个或多个占位符指示更多的一个或多个结构化文档文件。

9. 如权利要求 8 所述的计算机实施的方法，其中，转换源结构化文档的第一部分包括：

从计算机的存储器检索被中间结果文件中所包括的、识别的第一占位符指示的第一结构化文档文件；和

基于转换规则，由计算机的处理器转换第一结构化文档文件以获得目标文档的第一部分。

10. 如权利要求 9 所述的计算机实施的方法，其中，转换源结构化文档的更多的一个或

多个部分包括：

从计算机的存储器中，检索被中间结果文件中所包括的更多的一个或多个占位符指示的更多的一个或多个结构化文档文件；和

基于转换规则，由计算机的处理器转换更多的一个或多个结构化文档文件，以获得目标文档的更多的一个或多个部分。

11. 一种包括实际存储指令的计算机可读存储介质的制品，所述指令在被计算机执行时导致计算机：

接收转换源结构化文档的请求；

接收对源结构化文档中的第一节点以及更多的一个或多个节点的选择，第一节点被定义为第一占位符，并且更多的一个或多个节点被定义为更多的一个或多个占位符；

基于所接收的选择，把源结构化文档划分为源结构化文档的第一部分以及更多的一个或多个部分，第一占位符指示源结构化文档的第一部分，更多的一个或多个占位符指示源结构化文档的更多的一个或多个部分；

基于转换规则，转换源结构化文档的第一部分，以获得目标文档的第一部分；

基于转换规则，转换源结构化文档的更多的一个或多个部分，以获得目标文档的更多的一个或多个部分；和

基于所获得的目标文档的第一部分和所获得的目标文档的更多的一个或多个部分，产生目标文档。

12. 如权利要求 11 所述的制品，还包含指令，所述指令当被计算机执行时还导致计算机：

基于所定义的第一占位符以及更多的一个或多个占位符，把源结构化文档划分为源结构化文档的第一部分以及更多的一个或多个部分。

13. 如权利要求 12 所述的制品，还包含指令，所述指令当被计算机执行时还导致计算机：

存储结构化文档的第一部分作为第一结构化文档文件，第一占位符指示第一结构化文档文件；

存储源结构化文档的更多的一个或多个部分作为更多的一个或多个结构化文档文件，更多的一个或多个占位符指示更多的一个或多个结构化文档文件；和

把第一占位符、更多的一个或多个占位符和源结构化文档的剩余部分存储在源结构化文档索引文件中。

14. 如权利要求 13 所述的制品，还包含指令，所述指令当被计算机执行时还导致计算机：

基于转换规则，转换源结构化文档索引文件，以获得中间结果文件，中间结果文件包括第一占位符、更多的一个或多个占位符，以及源结构化文档的剩余部分的转换结果；

识别中间结果文件中所包括的第一占位符；

检索被识别的第一占位符指示的第一结构化文档文件；和

基于转换规则，转换第一结构化文档文件，以获得目标文档的第一部分。

15. 如权利要求 14 所述的制品，还包含指令，所述指令当被计算机执行时还导致计算机：

识别中间结果文件中所包括的更多的一个或多个占位符；

检索被更多的一个或多个占位符指示的更多的一个或多个结构化文档文件；和

基于转换规则，转换更多的一个或多个结构化文档文件，以获得目标文档的更多的一个或多个部分。

16. 一种用于把源结构化文档转换为目标文档的计算机系统，所述计算机系统包含：

存储器，用于存储程序代码；和

可通信地耦合到存储器的处理器，所述处理器被配置成执行程序代码以便：

接收转换源结构化文档的请求；

接收对源结构化文档中的第一节点以及更多的一个或多个节点的选择，第一节点被定义为第一占位符，并且更多的一个或多个节点被定义为更多的一个或多个占位符；

基于所接收的选择，把源结构化文档划分为源结构化文档的第一部分以及更多的一个或多个部分，第一占位符指示源结构化文档的第一部分，更多的一个或多个占位符指示源结构化文档的更多的一个或多个部分；

基于转换规则，转换源结构化文档的第一部分，以获得目标文档的第一部分；

基于转换规则，转换源结构化文档的更多的一个或多个部分，以获得目标文档的更多的一个或多个部分；和

基于所获得的目标文档的第一部分和所获得的目标文档的更多的一个或多个部分，产生目标文档。

17. 如权利要求 16 所述的系统，其中，处理器还执行程序代码以便：

基于所定义的第一占位符以及更多的一个或多个占位符，把源结构化文档划分为源结构化文档的第一部分以及更多的一个或多个部分。

18. 如权利要求 17 所述的系统，其中，处理器还执行程序代码以便：

存储结构化文档的第一部分作为第一结构化文档文件，第一占位符指示第一结构化文档文件；

存储源结构化文档的更多的一个或多个部分作为更多的一个或多个结构化文档文件，更多的一个或多个占位符指示更多的一个或多个结构化文档文件；和

把第一占位符、更多的一个或多个占位符和源结构化文档的剩余部分存储在源结构化文档索引文件中。

19. 如权利要求 18 所述的系统，其中，处理器还执行程序代码以便：

基于转换规则，转换源结构化文档索引文件，以获得中间结果文件，中间结果文件包括第一占位符、更多的一个或多个占位符，以及源结构化文档的剩余部分的转换结果；

识别中间结果文件中所包括的第一占位符；

检索被识别的第一占位符指示的第一结构化文档文件；和

基于转换规则，转换第一结构化文档文件，以获得目标文档的第一部分。

20. 如权利要求 19 所述的系统，其中，处理器还执行程序代码以便：

识别中间结果文件中所包括的更多的一个或多个占位符；

检索被更多的一个或多个占位符指示的更多的一个或多个结构化文档文件；和

基于转换规则，转换更多的一个或多个结构化文档文件，以获得目标文档的更多的一个或多个部分。

基于分割的结构化文档转换

技术领域

[0001] 本发明总体地涉及计算机系统，更具体地，涉及用于转换结构化文档的方法和系统。

背景技术

[0002] 例如SAP® Business One 的几种报告软件根据用户的要求把以结构化文档形式接收的商业数据转换为另一格式。在很多情况下，可以提供用于把结构化文档转换为另一格式的指令。例如，可扩展样式表转换语言 (extensible stylesheet transformation language，XSLT) 可以被定义成把是结构化文档的可扩展标记语言 (extensible markup language，XML) 文档翻译为另一结构化或非结构化文档形式（例如纯文本、字处理器、电子表格、数据库、pdf、HTML，等等）。

[0003] 通常，为了转换 XML 文档，XSLT 建立文档对象模型 (Document Object Model，DOM) 树，该树具有与 XML 文档的每一元素对应的节点。然后 XSLT 在所生成的 DOM 树上执行转换操作。DOM 树消耗的存储器大小和 XML 文档的大小成线性比例。因此，如果 XML 文档的大小大于可用的系统存储器，则转换过程可能抛出存储器耗尽异常。

发明内容

附图说明

[0004] 权利要求具体给出了本发明的实施例。在附图中通过举例而非限制来图示本发明，在附图中，相同的引用指示类似的元素。结合附图，从下面的详细描述可以最佳地理解本发明的实施例及其益处。

[0005] 图 1 是根据实施例示出用于把源文档转换为目标文档的方法的框图。

[0006] 图 2 是根据实施例示出用于把源结构化文档转换为目标文档的方法的详细流程图。

[0007] 图 3 是根据实施例示出源结构化文档索引文件的示范性框图。

[0008] 图 4A-4B 是根据示范性实施例示出用于转换图 3 的源结构化文档索引文件的转换文件的示范性框图。

[0009] 图 5 示出了示范性框图，根据实施例示出了在使用图 4 的转换文件转换图 3 的源结构化文档索引文件之后获得的中间结果文件。

[0010] 图 6 是根据实施例示出被图 5 的中间结果文件中的第一占位符 (placeholder) 指示的审计数据文件的第一部分的示范性框图。

[0011] 图 7 是根据实施例示出被图 5 的中间结果文件中的第二占位符指示的审计数据文件的第二部分的示范性框图。

[0012] 图 8A-8B 是根据实施例示出用于转换图 6 的审计数据文件的第一部分和图 7 的审计数据文件的第二部分的转换文件的示范性框图。

[0013] 图 9 是根据实施例示出在转换图 6 的第一文件中包括的审计数据的第一部分之后获得的目标文档的第一部分的示范性框图。

[0014] 图 10 是根据实施例示出在转换图 7 的审计数据文件的第二部分中包括的审计数据的第二部分之后获得的目标文档的第二部分的示范性框图。

[0015] 图 11 是根据实施例示出基于图 9 的目标文档的第一部分、图 10 的目标文档的第二部分和图 5 的中间结果文件获得的目标文档的示范性框图。

[0016] 图 12 是根据实施例示出其中可以实施所描述的用于基于故障 (fault) 容忍的查询执行技术的计算环境的框图。

具体实施方式

[0017] 这里描述基于分割的结构化文档转换技术的实施例。在下面的描述中，为了提供对本发明实施例的透彻理解给出了许多具体细节。但是，本领域技术人员将发现，无需这些具体细节中的一个或更多个，或者利用其他方法、部件、材料等，就可以实践本发明。在其他的实例中，为了避免模糊本发明的方面，未示出或者描述公知的结构、材料或者操作。

[0018] 贯穿本说明书，对“一个实施例”、“本实施例”和类似短语的引用意味着结合该实施例描述的特定特征、结构或者特性被包括在本发明的至少一个实施例中。因此，贯穿本说明书在各种位置出现这些短语不一定都指示相同的实施例。此外，在一个或更多个实施例中，特定特征、结构或者特性可被以任何适当方式组合。

[0019] 图 1 是根据实施例示出用于把源结构化文档 102 转换为目标文档 104 的方法的框图 100。结构化文档是一种根据例如 HTML(超文本标记语言，Hyper Text Markup Language)、XML(可扩展标记语言) 或 WSDL(网络服务定义语言，Web Service Definition Language) 的一个或更多个结构化定义语言构造的电子文档。结构化文档可以是节点的层次树的形式。每一节点可以具有名字、值以及其他相关联的信息。例如，考虑一学校的源 XML 文档：

[0020]

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="class.xsl"?>
<school>
<class>
        <student>James</student>
<student>Michael</student>
</class>
<address>
  <street> XYZ </street>
</address>
  </school>
```

[0021] 表 1

[0022] 表 1 中示出的源 XML 文档包括节点 <school>、<class>、<student>、<address> 和 <street>。节点 <student> 具有值 James 和 Michael，并且节点 <street> 具有值 XYZ。可以根据用户的要求对源结构化文档 102 执行转换操作以便获得目标文档 104。为了转换源结构化文档 102，起初对源结构化文档 102 执行划分操作 106 以便获得源结构化文档 102 的几个部分 108。可以基于源结构化文档 102 中的节点来划分源结构化文档 102。如图所示，源结构化文档 102 被划分为源结构化文档 102 的部分 1110 和 2112。在上面的例子中，表 1 中源结构化文档可被根据 <class> 节点和 <address> 节点划分以获得源结构化文档的两个部分，分别在表 2 和表 3 中示出。

[0023]

(部分 1)
<student>James</student>
<student>Michael</student>

[0024] 表 2

[0025]

(部分 2) <street>XYZ</street>

[0026] 表 3

[0027] 接着可以对源结构化文档的部分 108 执行转换操作 114 以获得目标文档 104 的部分 116。转换操作 114 可被分几个步骤执行，在每一步骤，可对源结构化文档 102 的部分 108 其中之一进行转换，以获得目标文档 104 的部分 116 其中之一。例如，可以对源结构化文档 102 的部分 1 110 执行转换操作 114，以获得目标文档 104 的部分 1 118。接着，可以对源结构化文档 102 的部分 2112 执行转换操作 114，以获得目标文档 104 的部分 2 120。在一个实施例中，定义转换规则，用于转换源结构化文档 102 的部分 108 以获得目标文档的部分 116。在上面的例子中，可以将转换规则定义为把包括在表 2 中所示的源结构化文档的部分 1 中的 <student> 节点转换为“FOUNDALEARNER”输出，并把包括在表 3 中所示的源结构化文档的部分 2 中的 <street> 节点转换为 <street> 节点的值 (XYZ) 输出。使用转换规则，可以首先转换表 2 中所示的源结构化文档的部分 1，以获得表 4 中所示的目标文档的部分 1，表 4 包括：

[0028]

(部分 1) FOUND A LEARNER
FOUND A LEARNER

[0029] 表 4

[0030] 接着，可对表 3 中所示的源结构化文档的部分 2 应用转换规则，以获得表 5 中所示的目标文档的部分 2，表 5 包括

[0031]

(部分 2)XYZ

[0032] 表 5

[0033] 最后，使用所获得的目标文档的部分 116 执行产生操作 122，以产生目标文档 104。可以通过组合目标文档 104 的获得的部分 1 118 和获得的部分 2120 来产生目标文档 104。在上面的例子中，表 4 中所示的目标文档的部分 1 和表 5 中所示的目标文档的部分 2 被组合以获得表 6 中所示的目标文档，表 6 包括 ：

[0034]

FOUND A LEARNER FOUND A LEARNER

[0035]

XYZ

[0036] 表 6

[0037] 图 2 是根据实施例示出用于把源结构化文档转换为目标文档的方法的详细流程图 200。起初，在块 202，接收转换源结构化文档的请求。可以根据例如 XML、HTML、WSDL 等的标准结构化文档语言中的任何一种构造源结构化文档。源结构化文档，例如 XML 文档，可以具有树结构，其可以具有根对象、分支对象和叶子对象。根对象、分支对象和叶子对象中的每一个是结构化文档的节点。节点可以具有对应的节点值或者与该节点相关联的任何其他信息。源结构化文档可以包括源数据，例如商业数据，或者任何其他数据。例如，图 7 中所示的表示几个公司的商业数据的源 XML 文档包括 ：

[0038]

```
-<business data>
-<companies>
-<company 1>
<company name> HAPPY COMPANY </company name>
<location> SUNNYVALE </location>
        </company 1>
        -<company 2>
<company name> WINNER INC. </company name>
            <location> NEW JERSEY </location>
          </company 2>
    </companies>
        -<customer>
            <customer name> BIG CUSTOMER </customer name>
    </customer>
</business data>
```

[0039] 表 7

[0040] 在上面的例子中，<business data>是根节点，<companies>、<company 1>、<company 2>和<customer>是分支节点，并且<company name>、<location>和<customer name>是叶子节点。节点<company name>具有值 HAPPY COMPANY 和 WINNER INC，节点<location>具有值 SUNNYVALE 和 NEW JERSEY，并且节点<CUSTOMER NAME>具有值 BIG CUSTOMER。

[0041] 接着，在块 204，选择源结构化文档中的、用于分割源结构化文档的节点。用户可以从源结构化文档选择一个或更多个节点，用户想要基于所述节点划分源结构化文档。可以提供向用户显示源结构化文档的显示工具。用户可以从显示的源结构化文档选择一个或更多个节点。在一个实施例中，系统可以仅允许用户选择结构化文档中的节点中的某些用于划分源结构化文档。例如，可仅允许用户选择根节点和分支节点，而不选择叶子节点。在上面的例子中，可将源结构化文档展示给用户。可仅允许用户选择根节点<business data>和分支节点<companies>、<company 1>、<company 2>和<customer>。假设用户选择了分支节点<companies>、<company 1>和<company 2>。

[0042] 可将所选择的节点定义为源结构化文档的占位符。占位符是指示 (refer to) 源结构化文档的一部分的节点。在一个实施例中，占位符是间接指示源结构化文档的一部分的节点。在这种情况下，占位符可以指示另一个占位符，所述另一个占位符指示源结构化文档的一部分。在上面的例子中，将所选择的节点<companies>、<company 1>和<company 2>

定义为源结构化文档的占位符。占位符 <company 1> 和占位符 <company 2> 可以分别指示源结构化文档的部分 1 和部分 2，分别在表 8 和表 9 中示出。

[0043]

```
(部分 1) <company name> HAPPY COMPANY </company name>
        <location> SUNNYVALE </location>
```

[0044] 表 8

[0045]

```
(部分 2) <company name> WINNER INC. </company name>
        <location> NEW JERSEY </location>
```

[0046] 表 9

[0047] 占位符 <companies> 间接指示源结构化文档的部分 1 和部分 2，这意味着占位符 <companies> 指示占位符 <company 1> 和占位符 <company 2>，占位符 <company 1> 和占位符 <company 2> 分别指示源结构化文档的部分 1 和部分 2。接着，在块 206，可以基于所定义的占位符指示 (refer) 的源结构化文档的部分来划分源结构化文档。划分源结构化文档可以包括：把所定义的占位符指示的源结构化文档的部分中的每一个作为单独的结构化文档文件存储。占位符可以指示存储该源结构化文档的部分的结构化文档文件。在上面的例子中，源结构化文档被划分为两个部分——被占位符 <company 1> 指示的部分 1 和被占位符 <company 2> 指示的部分 2。可以生成分别存储源结构化文档的部分 1 和部分 2 的第一结构化文档文件 (file 1.xml) 和第二结构化文档文件 (file 2.xml)。占位符 <company 1> 和占位符 <company 2> 可以分别指示第一结构化文档文件 (file 1.xml) 和第二结构化文档文件 (file 2.xml)。接着，在块 208，生成源结构化文档索引文件，该文件存储被包括在源结构化文档中的被定义的占位符以及源结构化文档的未被任一占位符指示的剩余部分。在上面的例子中，表 10 中所示的源结构化文档的未被占位符 <companies>、<company 1> 和 <company 2> 指示的剩余部分包括：

[0048]

```
-<customer>
<customer name> BIG CUSTOMER </customer name>
</customer>
```

[0049] 表 10

[0050] 所获得的源结构化文档索引文件在表 11 中示出，其包括：

[0051]

```
-<companies> (placeholder companies)
-<company 1> (placeholder company 1)
    -<company 2> (placeholder company 2)
  (remaining portion of the source structured document)
    -<customer>
        <customer name> BIG CUSTOMER </customer name>
    </customer>
```

[0052] 表 11

[0053] 在一个实施例中，可以基于例如商业数据的源数据直接生成源结构化文档索引文件。在这种情况下，可以接受用户选择以便定义指示源数据的不同部分的占位符。这种情况下的源结构化文档索引文件可以包括被定义的占位符以及源数据的未被任何占位符指示的剩余部分。接着，在块 210，对在块 208 获得的源结构化文档索引文件进行转换以获得中间结果。转换是对输入的结构化文档——要对该输入的结构化文档应用转换——进行转换以获得输出的目标文档的过程。在一个实施例中，目标文档可以是结构化或者非结构化文档形式（例如纯文本、字处理器、电子表格、数据库、pdf、HTML 等）。例如，源结构化文档可以是 XML 格式并且目标文档可以是 XML 或者 HTML 格式，或者任何其他依据用户要求的格式。可以通过转换文件执行转换操作，转换文件包括用于把源结构化文档转换为目标文档的转换规则。例如，可以使用可扩展样式表转换语言 (XSLT) 文件把 XML 源文档转换为 XML 目标文档。XSLT 转换可以由 XSLT 处理器执行，XSLT 处理器采用 XML 源文档和 XSLT 样式表作为输入，并且产生目标文档。XSLT 样式表包含转换规则的集合，其为在产生目标文档时引导处理器的指令和其他指示。XSLT 样式表可以包括与源结构化文档中的不同节点对应的转换规则。XSLT 处理器可以通过把源结构化文档中的节点与 XSLT 样式表中的转换规则匹配，并把对应的转换规则应用于该节点来执行转换。系统可以存储用于执行不同转换的几个转换文件。例如，可以将不同的转换文件存储在系统中，以便转换源结构化文档索引文件和被占位符指示的源结构化文档的部分。

[0054] 为了转换源结构化文档索引文件，可以对源结构化文档索引文件应用转换规则，以便对源结构化文档索引文件中所包括的源结构化文档的剩余部分进行转换，从而获得目标文档的剩余部分。在对源结构化文档索引文件的转换操作之后，可以获得包括目标文档的剩余部分和源结构化文档索引文件中的占位符的中间结果。在上面的例子中，用于转换源结构化文档索引文件的源结构化文档索引转换文件可以包括：

[0055]

```
<xsl: template match = “Customer”>
  <xsl: apply_template select = “Customer Name”/>
</xsl: template>
<xsl: template match = “Customer Name”>
<xsl: value of select = “.”/>
</xsl: template>
```

[0056] 表 12

[0057] 如表 12 中所示，源结构化文档索引转换文件包括转换规则 <xsl:template match>，其检查源结构化文档中的节点是否是“customer name”。在该节点是“customer name”节点的情况下，源结构化文档索引转换文件中所包括的转换规则 <xsl:value of select =“.”/> 从源结构化文档提取节点（“customer name”）的值，并把所提取的节点值放置在输出的结构化文档（在这种情况下是中间结果）中。在上面的例子中，<customer name> 节点的值，BIG CUSTOMER 被放置在在转换源结构化文档索引文件之后获得的表 13 中所示的中间结果中。表 13 中所示的把转换规则应用于源结构化文档索引文件之后获得的中间结果包括：

[0058]

```
-<companies> (placeholder companies)
-<company 1> (placeholder company 1)
    -<company 2> (placeholder company 2)
  BIG CUSTOMER  (transformation result of remaining portion of source structured document)
```

[0059] 表 13

[0060] 如所示，如表 13 中所示的中间结果包括源结构化文档的剩余部分的转换结果（目标文档的剩余部分）和在源结构化文档中定义的占位符（<companies>、<company 1> 和 <company 2>）。接着，在块 212，遍历中间结果，以识别中间结果中的占位符。在上面的例子中，遍历中间结果从而识别三个占位符 <companies>、<company 1> 和 <company 2>。接着，在块 214，检索被中间结果中所包括的占位符指示的源结构化文档的部分，用于转换这些源结构化文档的部分（块 216）。在一个实施例中，逐一检索存储被中间结果文件中的占位符指示的源结构化文档的部分的结构化文档文件，以供执行转换操作。可使用转换文件中所包括的转换规则转换源结构化文档的部分，以获得目标文档的部分。转换操作可分几个步骤执行，在每一步骤，可以将被占位符指示的源结构化文档的部分其中之一加载于系统的存储器中，以供对该源结构化文档的部分执行转换操作。可重复转换操作，直到被占位符指示的所有源结构化文档的部分都被转换以获得目标文档的部分为止。例如，假设中间结果文件包括指示源结构化文档的三个不同部分的三个占位符。可检索被中间结果中的第

一占位符指示的源结构化文档的第一部分并将其加载到存储器中。可对源结构化文档的第一部分应用转换操作，以获得目标文档的第一部分。在获得目标文档的第一部分之后，可检索被中间结果中的第二占位符指示的源结构化文档的第二部分，用于把源结构化文档的第二部分转换为目标文档的第二部分。最终，在获得目标文档的第二部分之后，可检索被中间结果文件中的第三占位符指示的源结构化文档的第三部分，用于转换源结构化文档的第三部分。

[0061] 把占位符逐一地加载到存储器中以供执行转换操作保证了，所消耗的存储器依赖于转换源结构化文档的部分的复杂度，而非基于该源结构化文档的大小。如上面所讨论的那样，可以将不同的转换文件存储在系统中，用于转换源结构化文档的不同部分。在上面的例子中，由于源结构化文档的分别被占位符 <company 1> 和 <company 2> 指示的部分 1 和部分 2 包括类似的元素，所以可以使用单个转换文件来转换源结构化文档的部分 1 和部分 2。表 14 中所示的单个转换文件可以包括：

[0062]

```
<xsl: template match = “Company”>
  <xsl: apply_template select = “Company Name”/>
</xsl: template>
<xsl: template match = “Company Name”>
<xsl: value of select = “.”/>
</xsl: template>
```

[0063] 表 14

[0064] 可将表 8 中所示的源结构化文档的第一部分加载到存储器中，并且可对源结构化文档的第一部分应用转换文件中的转换规则，以获得图 15 中所示的目标文档的第一部分，表 15 包括：

[0065]

```
HAPPY COMPANY
SUNNYVALE
```

[0066] 表 15

[0067] 在获得目标文档的第一部分之后，可将表 9 中所示的源结构化文档的第二部分加载到存储器中，并且，可将表 9 中所示的单个转换文件中的转换规则应用于源结构化文档的第二部分以获得表 16 中所示的目标文档的第二部分，其包括：

[0068]

```
WINNER INC.
NEW JERSEY
```

[0069] 表 16

[0070] 最后，在块218，基于在块216获得的目标文档的部分和在块210获得的中间结果，生成目标文档。在一个实施例中，可以通过把在块216获得的目标文档的部分与在块210获得的中间结果中的目标文档的剩余部分组合来获得目标文档。可以通过将源结构化文档的部分用目标文档的对应部分替换来生成目标文档。在上面的例子中，源结构化文档的第一部分、第二部分和剩余部分被表15中所示的目标文档的第一部分、表16中所示的目标文档的第二部分和表13中所示的中间结果中所包括的目标文档的剩余部分替换，以获得表17中所示的目标文档，其包括：

[0071]

HAPPY COMPANY
SUNNY VALE
WINNER INC.
NEW JERSEY
BIG CUSTOMER

[0072] 表17

[0073] 图3是根据实施例示出源结构化文档索引文件300的示范性框图。如上面所讨论的，可以从例如商业数据的源数据直接生成源结构化文档索引文件。基于公司的审计数据生成源结构化文档索引文件300。源结构化文档索引文件300包括具有占位符id“N_127_11”的第一占位符302和具有占位符id“N_128_6”的第二占位符304。第一占位符302和第二占位符304可以指示用户所选择的审计数据的不同部分。源结构化文档索引文件300也可以包括未被第一占位符302和第二占位符304指示的、公司审计数据的剩余部分306。源结构化文档索引文件300包括结点<company>、<companyName>、<streetAddress>、<streetname>、<city>、<transactions>、<totalDebit>、<totalCredit>、<journal>、<desc>、<jrnTp和<subledgers>。

[0074] 图4A-4B是示出根据示范性实施例的用于转换图3的源结构化文档索引文件300的转换文件400的示范性框图。转换文件400可以包括用于源结构化文档的每一节点的转换规则。例如，被包括在图3的源结构化文档索引文件300中的用于节点“companyName”的转换规则402被定义为，从图3的源结构化文档索引文件300检索节点“companyName”和节点“companyName”的值(ABC CORP)，并将其放入目标文档中（在这种情况下目标文档是中间结果文件）。类似地，转换文件400包括用于从图3的源结构化文档索引文件300检索其他节点及其对应值，并将其放入目标文档（中间结果文件）中的转换规则。

[0075] 图5示出了示范性框图，根据实施例示出了在使用图4的转换文件400转换图3的源结构化文档索引文件300之后获得的中间结果文件500。中间结果文件500包括图3的源结构化文档索引文件300中所包括的审计数据306的剩余部分的转换结果502。中间结果文件500也包括源结构化文档索引文件300中所包括的第一占位符302和第二占位符304。接着，遍历中间结果文件500以便识别中间结果文件500中的占位符——第一占位符302和第二占位符304。第一占位符302和第二占位符304可以指示审计数据的第一部分和第二部分。

[0076] 图 6 是根据实施例示出被图 5 的中间结果文件 500 中的第一占位符 302 指示的审计数据文件的第一部分 600 的示范性框图。审计数据文件的第一部分 600 可以存储被图 5 的第一占位符 302 指示的审计数据的第一部分。如所示，审计数据文件的第一部分 600 存储报告的事务处理数据，其包括数量节点 (<nr>)、描述节点 (<desc>)、周期数节点 (<period number>)、事务处理日期节点 (<trDt>) 和源 ID 节点 (<sourceID>)。

[0077] 图 7 是根据实施例示出被图 5 的中间结果文件 500 中的第二占位符 304 指示的审计数据文件的第二部分 700 的示范性框图。审计数据文件的第二部分 700 可以存储被图 5 的第二占位符 302 指示的审计数据的第二部分。如所示，审计数据文件的第二部分 700 存储审计数据中的分类帐信息，其包括分类帐类型 (<nr>)、总借方节点 (<totalDebit>) 和总贷方节点 (<totalCredit>)。

[0078] 图 8A-8B 是根据实施例示出用于转换图 6 的审计数据文件的第一部分 600 和图 7 的审计数据文件的第二部分 700 的转换文件 800 的示范性框图。转换文件 800 包括用于转换存储在图 6 的文件 600 中的审计数据的第一部分的转换规则 802。转换文件 800 也包括用于转换存储在图 7 的文件 700 中的审计数据的第二部分的转换规则 804。在一个实施例中，可以使用单独的转换文件用于转换图 6 的审计数据文件的第一部分 600 和图 7 的审计数据文件的第二部分 700。起初，可以将存储审计数据的第一部分的审计数据文件的第一部分 600 加载到系统的存储器中。接着，可以对审计数据的第一部分应用转换规则以获得目标文档的第一部分。接着，可以将存储审计数据的第二部分的图 7 的审计数据文件的第二部分 700 加载到系统的存储器中。最后，可以对审计数据的第二部分应用转换规则以获得目标文档的第二部分。转换文件包括用于图 6 的文件 600 中所包括的审计数据的第一部分中和图 7 的文件 700 中所包括的审计数据的第二部分中的每一节点的转换规则 (xsl:for each select =“text()><xsl:value of select =“.”/>。这些转换规则从图 6 的审计数据文件的第一部分 600 和图 7 的审计数据文件的第二部分 700 检索节点及对应的节点值，并将其分别放入目标文档的第一部分和第二部分中。

[0079] 图 9 是根据实施例示出在转换图 6 的审计数据文件的第一部分 600 中包括的审计数据的第一部分之后获得的目标文档的第一部分 900 的示范性框图。通过把图 8A 到 8B 的转换文件 800 中所包括的转换规则 802 应用于图 6 的文件 600 中所包括的审计数据的第一部分来获得目标文档的第一部分 900。目标文档的第一部分 900 包括在执行过转换操作之后从图 6 的审计数据文件的第一部分 600 检索的节点和节点的对应值。

[0080] 图 10 是根据实施例示出在转换图 7 的审计数据文件的第二部分 700 中包括的审计数据的第二部分之后获得的目标文档的第二部分 1000 的示范性框图。通过把图 8A 到 8B 的转换文件 800 中所包括的转换规则 804 应用于图 7 的文件 700 中所包括的审计数据的第二部分来获得目标文档的第二部分 1000。目标文档的第二部分 1000 包括在执行过转换操作之后从图 7 的审计数据文件的第二部分 700 检索的节点和节点的对应值。

[0081] 图 11 是根据实施例示出基于图 9 的目标文档的第一部分 900、图 10 的目标文档的第二部分 1000 和图 5 的中间结果文件 500 获得的目标文档 1100 的示范性框图。通过组合图 9 的目标文档的第一部分 900、图 10 的目标文档的第二部分 1000 以及图 5 的中间结果 500 中所包括的审计数据的剩余部分的转换结果 502，可以获得目标文档 1100。在一个实施例中，通过利用图 5 的中间结果中所包括的审计数据的剩余部分的转换结果 502、图 9 的目

标文档的第一部分 900 和图 10 的目标文档的第二部分 1000 分别替换图 3 的源结构化文档索引文件 300 中所包括的审计数据的剩余部分 306、第一占位符 302 和第二占位符 304，可以获得目标文档 1100。

[0082] 本发明的某些实施例可以包括被编写为一个或更多个软件部件的上述方法。这些部件，以及与每一个部件相关联的功能，可以被客户端、服务器、分布式或者对等计算机系统使用。这些部件可以用对应于一个或更多个编程语言的计算机语言书写，该一个或更多个编程语言例如函数式、说明式、过程式、面向对象式、低级语言，等等。这些部件可通过各种应用程序编程接口被链接到其他部件，然后被编译为用于服务器或者客户端的一个完整应用。或者，可以在服务器和客户端应用中实施这些部件。此外，这些部件可通过各种分布式编程协议被链接在一起。本发明的某些示范性实施例可以包括用来跨过分布式编程环境实施这些部件中的一个或更多个的远程过程调用或者网络服务。例如，逻辑层可以驻留在第一计算机系统上，第一计算机系统远离包含接口层（例如图形用户接口）的第二计算机系统。这些第一和第二计算机系统可以服务器－客户端、点对点或者某种其他结构来配置。客户端在复杂性上可以从移动和手持设备变化到薄客户端以及厚客户端或者乃至其他服务器。

[0083] 上面示出的软件部件实质上存储在计算机可读存储介质上作为指令。术语“计算机可读存储介质”应该被视为包括存储一个或更多个指令集合的单个介质或者多个介质。术语“计算机可读存储介质”应该被视为包括能够经历一组物理变化以便在物理上存储、编码或者执行供计算机系统执行的导致计算机系统执行这里描述、表达或者示出的任何方法或者过程步骤的指令集合的任何物理制品。计算机可读存储介质的例子包括但不限于：例如硬盘、软盘和磁带的磁性介质；例如 CD-ROM、DVD 和全息设备的光学介质；磁光介质；以及，被具体配置成存储和执行的硬件设备，例如专用集成电路（“ASIC”）、可编程逻辑器件 (programmable logic devices，PLD) 和 ROM 和 RAM 器件。计算机可读指令的例子包括例如由编译器产生的机器码和包含由计算机使用解译器执行的高级代码的文件。例如，可以使用 Java、C++ 或者其他面向对象编程语言和开发工具来实施本发明的实施例。本发明其他的实施例可以被实施在硬连线的电路中，代替机器可读软件指令或者或者与其组合。

[0084] 图 12 是示范性计算机系统 1200 的框图。计算机系统 1200 包括执行存储在计算机可读存储介质 1222 上的软件指令或者代码以执行上面示出的本发明的方法的处理器 1202。计算机系统 1200 包括从计算机可读介质 1222 读取指令并把指令存储在存储设备 1204 或者随机访问存储器 (RAM) 1206 中的介质读取器 1216。存储设备 1204 提供用于保持静态数据的海量空间，其中，至少可以存储某些指令以供以后执行。所存储的指令还可以被编译以生成指令的其他表达并被动态地存储在 RAM 1206 中。处理器 1202 从 RAM 1206 读取指令，并按指令执行动作。根据本发明的一个实施例，计算机系统 1200 还包括输出设备 1210（例如显示器）和输入设备 1212，输出设备 1210 向用户提供执行结果的至少一些作为输出，包括但不限于视觉信息，输入设备 1212 给用户或者另一设备提供用于输入数据和 / 或与计算机系统 1200 交互的手段。这些输出设备 1210 和输入设备 1212 中的每一个可与一个或更多个额外的外围设备结合以便进一步扩展计算机系统 1200 的能力。可以提供网络通信器 1214 以便把计算机系统 1200 连接到网络 1220，并进而连接到连接到网络 1220 的其他设备，包括例如其他的客户端、服务器、数据仓库和接口。计算机系统 1200 的模块通过

总线 1218 互连。计算机系统 1200 包括访问数据源 1224 的数据源接口 1208。可以通过以硬件或者软件实施的一个或更多个抽象层来访问数据源 1224。例如，可通过网络 1220 访问数据源 1224。在某些实施例中，可以通过例如语义层的抽象层访问数据源 1224。

[0085] 数据源是一种信息资源。数据源包括实现数据存储和检索的数据来源。数据源可以包括数据库，例如关系型、事务型、层次型、多维型（例如 OLAP）、面向对象型数据库，等等。进一步的数据源包括表格式数据（例如电子表格、定界的文本文件）、以标记语言标记的数据（例如 XML 数据）、事务数据、非结构化数据（例如文本文件、屏幕抓取）、层次数据（例如文件系统中的数据、XML 数据）、文件、多个报告，以及任何可通过已建立的协议访问的其他数据源，所述协议例如由底层软件系统（例如 ERP 系统）产生的开放数据库连接 (Open DataBase Connectivity，ODBC)，等等。数据源也可以包括其中数据不被实际存储或者是短暂的数据源，例如数据流、广播数据，等等。这些数据源可以包括相关联的数据基础、语义层、管理系统、安全系统，诸如此类。

[0086] 在上面的描述中，给出了许多具体细节以提供对本发明实施例的透彻理解。但是，本领域技术人员将发现，无需这些具体细节中的一个或更多个，或者利用其他方法、部件、技术等，就可以实践本发明。在其他的实例中，为了避免模糊本发明的方面，未示出或者描述公知的操作或者结构。

[0087] 尽管这里示出和描述的过程包括系列步骤，但是将会理解，本发明的不同实施例不受所示出的步骤次序限制，因为除了这里示出和描述的，某些步骤可以按不同的顺序出现，某些与其他步骤并发。此外，为了实施根据本发明的方法学，并非要求所有被示出的步骤。而且，将会理解，这些过程可以与这里示出和描述的装置和系统相关联地以及与未示出的其他系统相关联地实施。

[0088] 上面对本发明实施例的描述和说明，包括摘要中所描述的，并非旨在穷尽或者把本发明限制到所公开的精确形式。虽然这里为了说明性目的描述了本发明的具体实施例和例子，但是，本领域技术人员将会发现，在本发明的范围内，各种等同修改是可能的。考虑到上面的详细描述，可以对本发明做出这些修改。当然，本发明的范围要由下列权利要求确定，要依据既定的权利要求构造原则来解释所述权利要求。

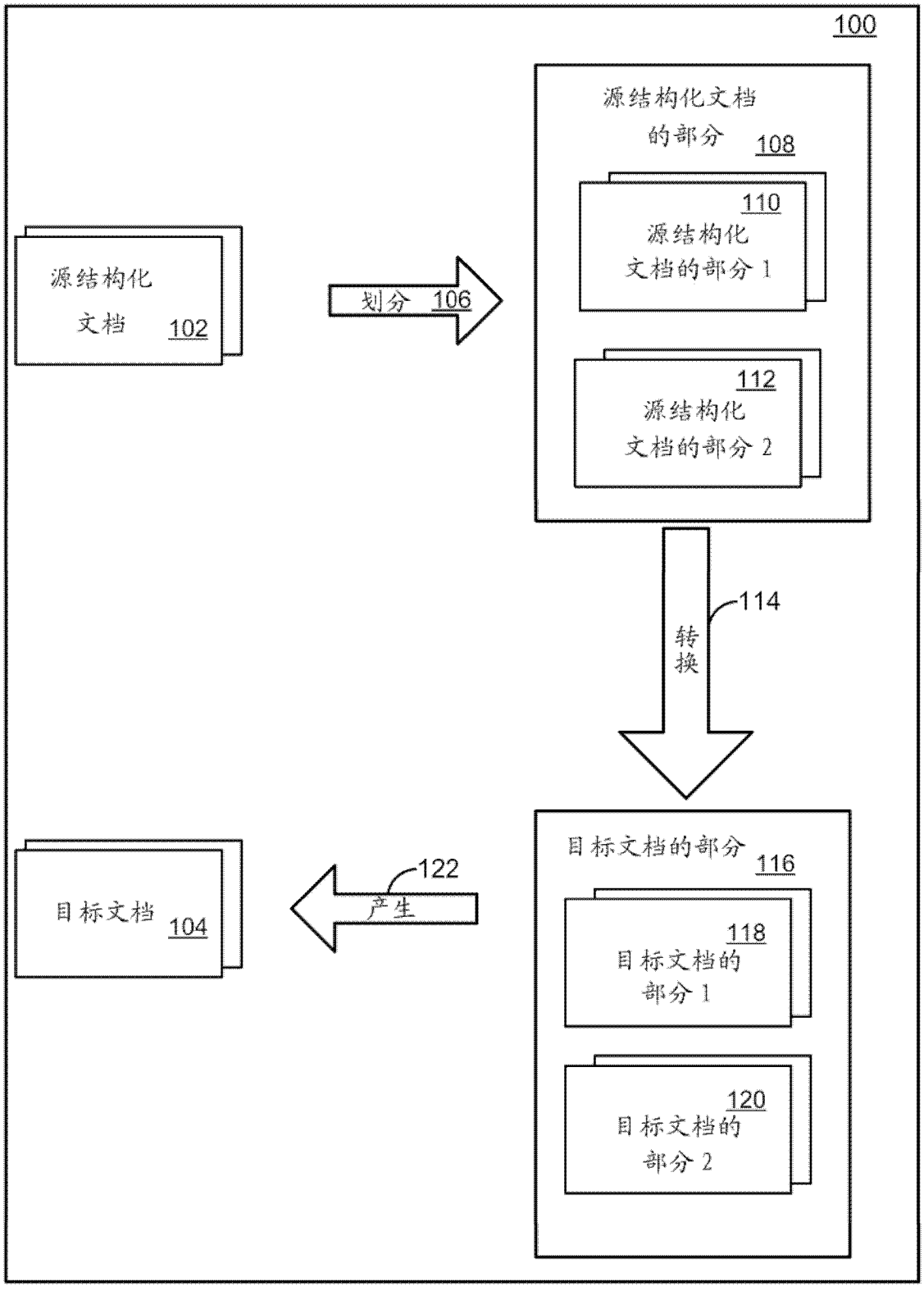

100
源结构化文档
的部分 108
110
源结构化
文档的部分 1
112
源结构化
文档的部分 2
源结构化
文档 102
划分 106
转
换
114
目标文档的部分 116
118
目标文档的
部分 1
120
目标文档的
部分 2
122
产生
目标文档 104

图 1

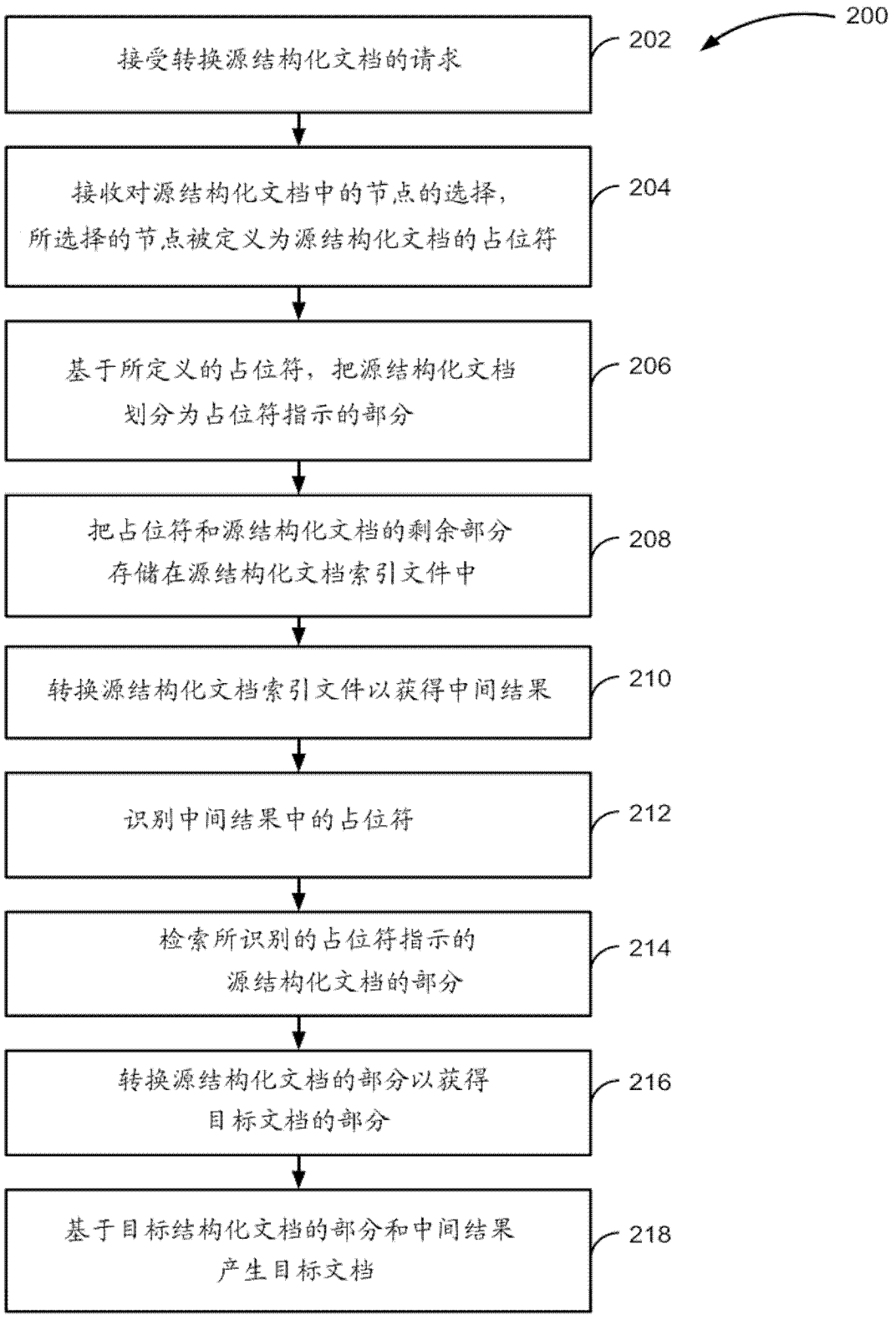

200
接受转换源结构化文档的请求
202
接收对源结构化文档中的节点的选择，所选择的节点被定义为源结构化文档的占位符
204
基于所定义的占位符，把源结构化文档划分为占位符指示的部分
206
把占位符和源结构化文档的剩余部分存储在源结构化文档索引文件中
208
转换源结构化文档索引文件以获得中间结果
210
识别中间结果中的占位符
212
检索所识别的占位符指示的源结构化文档的部分
214
转换源结构化文档的部分以获得目标文档的部分
216
基于目标结构化文档的部分和中间结果产生目标文档
218

图 2

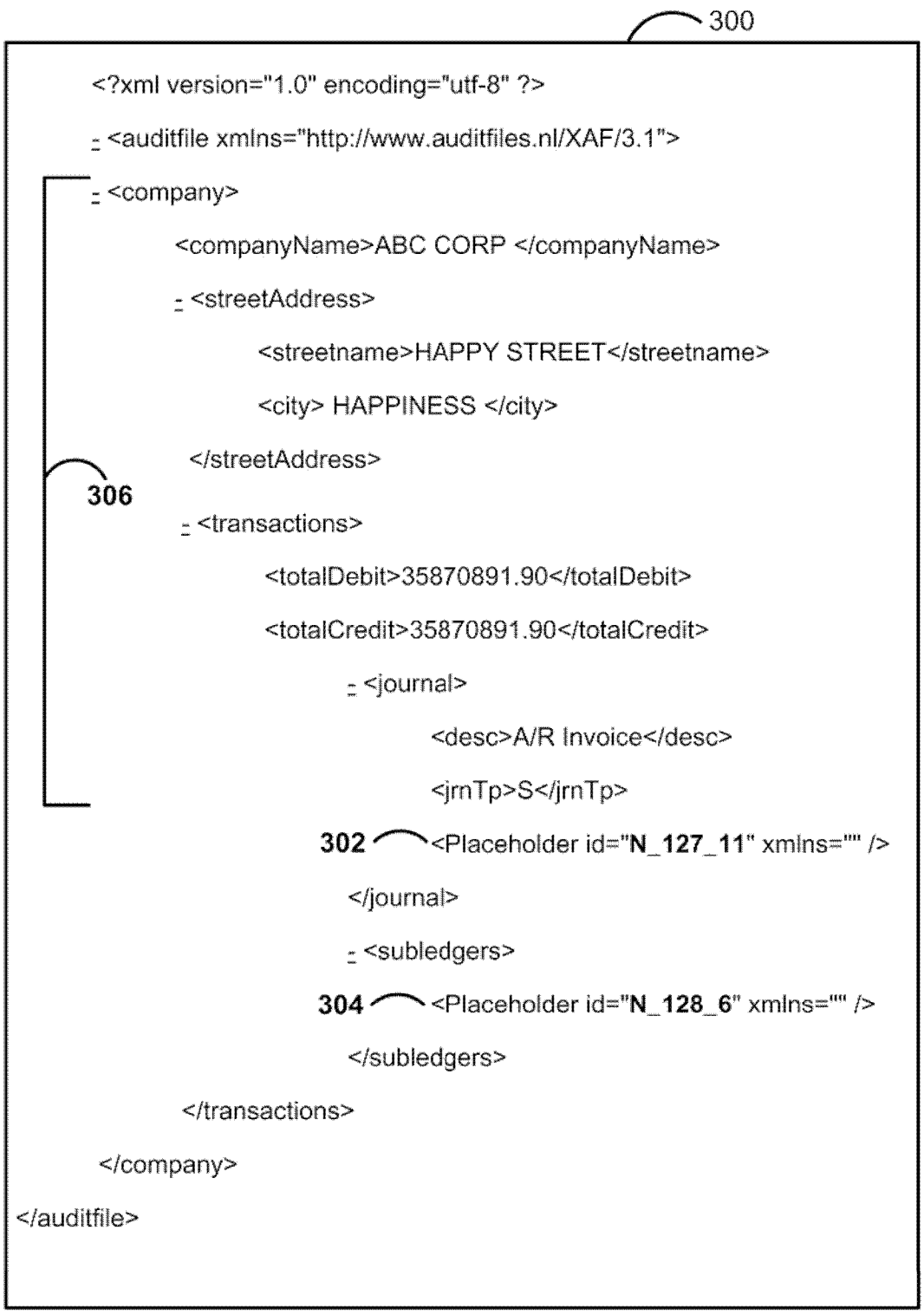

300
<?xml version="1.0" encoding="utf-8" ?>
<auditfile xmlns="http://www.auditfiles.nl/XAF/3.1">
<company>
<companyName>ABC CORP </companyName>
<streetAddress>
<streetname>HAPPY STREET</streetname>
<city> HAPPINESS </city>
</streetAddress>
306
<transactions>
<totalDebit>35870891.90</totalDebit>
<totalCredit>35870891.90</totalCredit>
<journal>
<desc>A/R Invoice</desc>
<jrnTp>S</jrnTp>
302 <Placeholder id="N_127_11" xmlns="" />
</journal>
<subledgers>
304 <Placeholder id="N_128_6" xmlns="" />
</subledgers>
</transactions>
</company>
</auditfile>

图 3

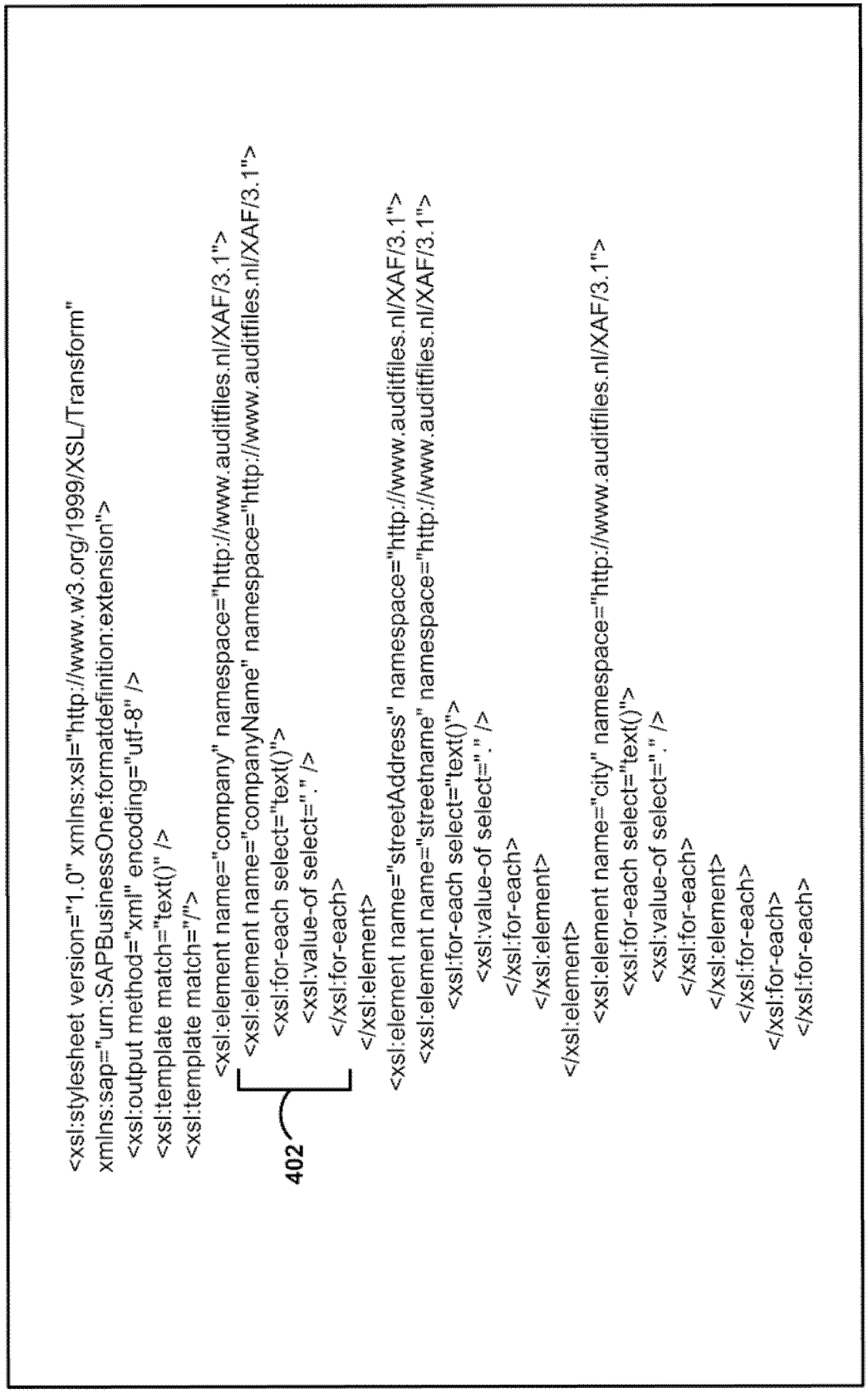

400
402
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:sap="urn:SAPBusinessOne:formatdefinition:extension">
<xsl:output method="xml" encoding="utf-8" />
<xsl:template match="text()" />
<xsl:template match="/">
<xsl:element name="company" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:element name="companyName" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
<xsl:element name="streetAddress" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:element name="streetname" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:element>
<xsl:element name="city" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>
</xsl:for-each>
</xsl:for-each>

图 4A


```
<xsl:element name="transactions" namespace="http://www.auditfiles.nl/XAF/3.1">
            <xsl:element name="totalDebit" namespace="http://www.auditfiles.nl/XAF/3.1">
                  <xsl:for-each select="text()">
                    <xsl:value-of select="." />
                  </xsl:for-each>
                </xsl:element>
              </xsl:for-each>
  <xsl:element name="totalCredit" namespace="http://www.auditfiles.nl/XAF/3.1">
                  <xsl:for-each select="text()">
                    <xsl:value-of select="." />
                  </xsl:for-each>
<xsl:element name="journal" namespace="http://www.auditfiles.nl/XAF/3.1">
    <xsl:element name="jrnTp" namespace="http://www.auditfiles.nl/XAF/3.1">
                      <xsl:for-each select="text()">
                        <xsl:value-of select="." />
            </xsl:for-each>
              </xsl:element>
            </xsl:for-each>
<xsl:element name="Placeholder">
                    <xsl:attribute name="id">
                      <xsl:value-of select="'N_127_11'" />
                    </xsl:attribute>
                  </xsl:element>
<xsl:element name="subledgers" namespace="http://www.auditfiles.nl/XAF/3.1">
              <xsl:element name="Placeholder">
                  <xsl:attribute name="id">
                    <xsl:value-of select="'N_128_6'" />
                  </xsl:attribute>
                </xsl:element>
              </xsl:for-each>
              </xsl:element>
            </xsl:for-each>
            </xsl:element>
            </xsl:for-each>
          </xsl:for-each>
        </xsl:element>
      </xsl:for-each>
    </xsl:template>
  </xsl:stylesheet>
```

图 4B

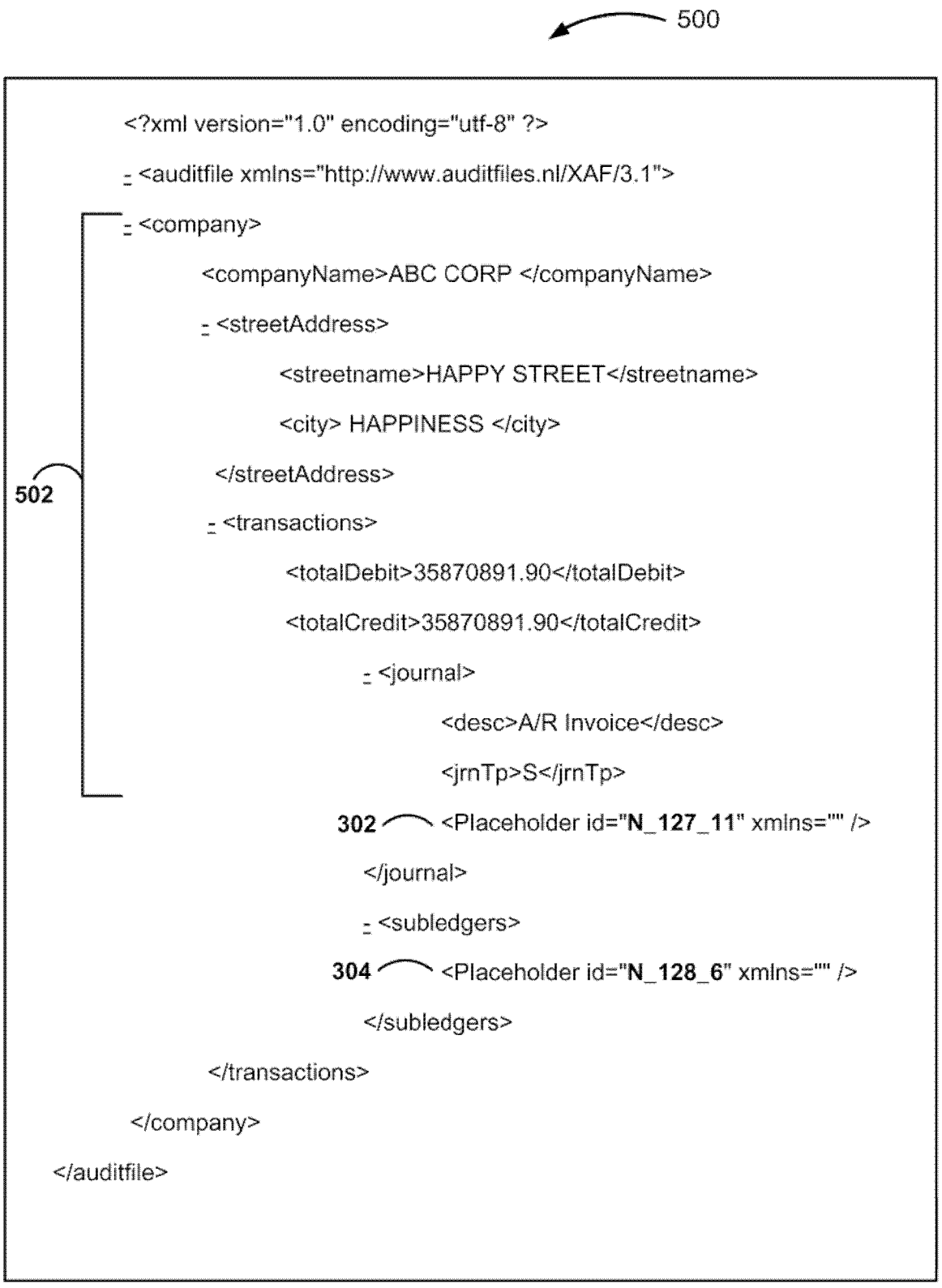

500
<?xml version="1.0" encoding="utf-8" ?>
<auditfile xmlns="http://www.auditfiles.nl/XAF/3.1">
<company>
<companyName>ABC CORP </companyName>
<streetAddress>
<streetname>HAPPY STREET</streetname>
<city> HAPPINESS </city>
</streetAddress>
502
<transactions>
<totalDebit>35870891.90</totalDebit>
<totalCredit>35870891.90</totalCredit>
<journal>
<desc>A/R Invoice</desc>
<jrnTp>S</jrnTp>
302 <Placeholder id="N_127_11" xmlns="" />
</journal>
<subledgers>
304 <Placeholder id="N_128_6" xmlns="" />
</subledgers>
</transactions>
</company>
</auditfile>

图 5

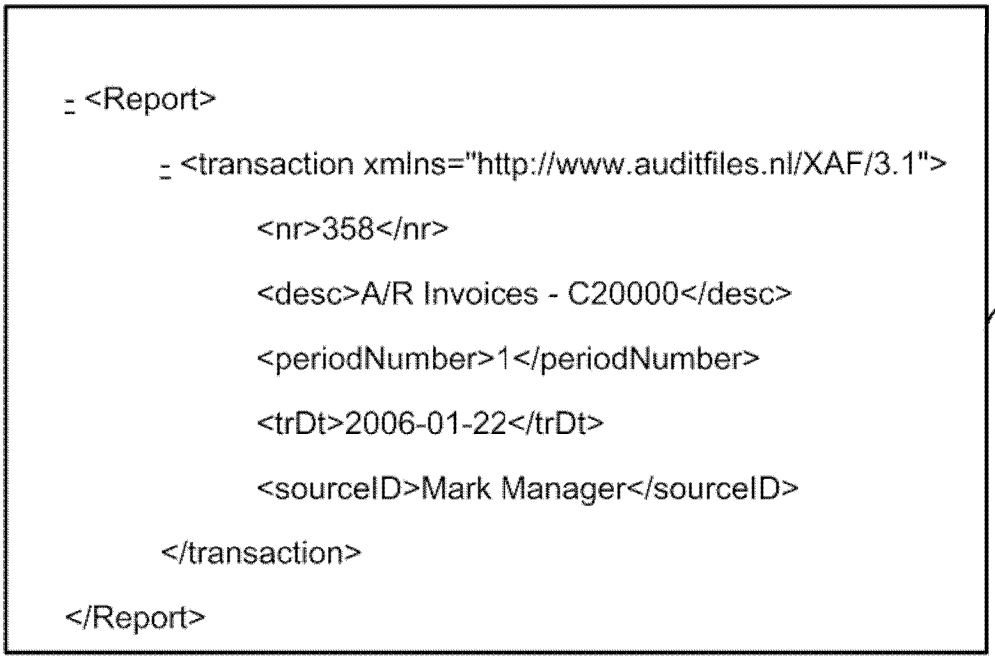

<Report>
<transaction xmlns="http://www.auditfiles.nl/XAF/3.1">
<nr>358</nr>
<desc>A/R Invoices - C20000</desc>
<periodNumber>1</periodNumber>
<trDt>2006-01-22</trDt>
<sourceID>Mark Manager</sourceID>
</transaction>
</Report>
600

图 6

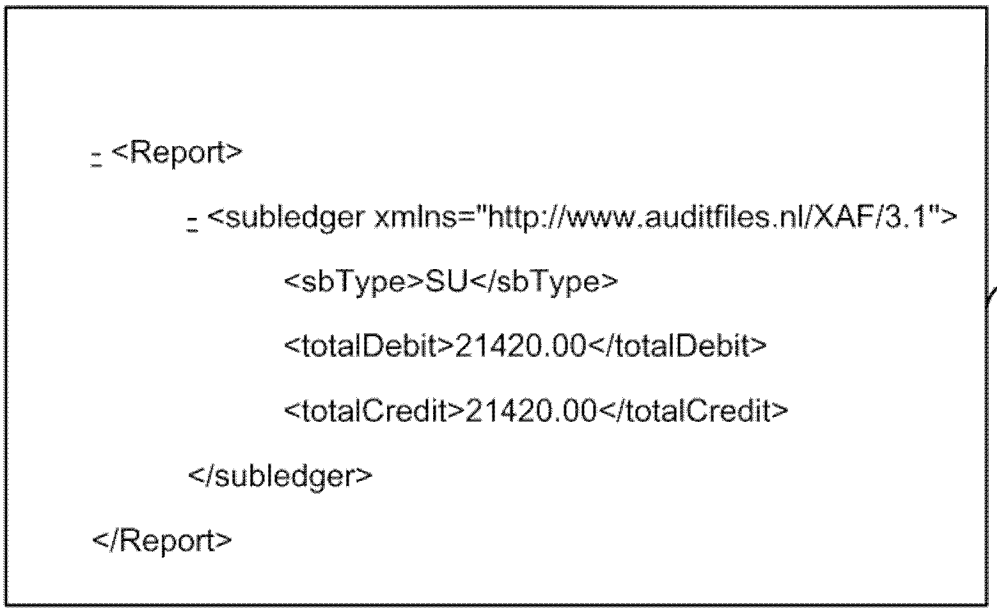

<Report>
<subledger xmlns="http://www.auditfiles.nl/XAF/3.1">
<sbType>SU</sbType>
<totalDebit>21420.00</totalDebit>
<totalCredit>21420.00</totalCredit>
</subledger>
</Report>
700

图 7

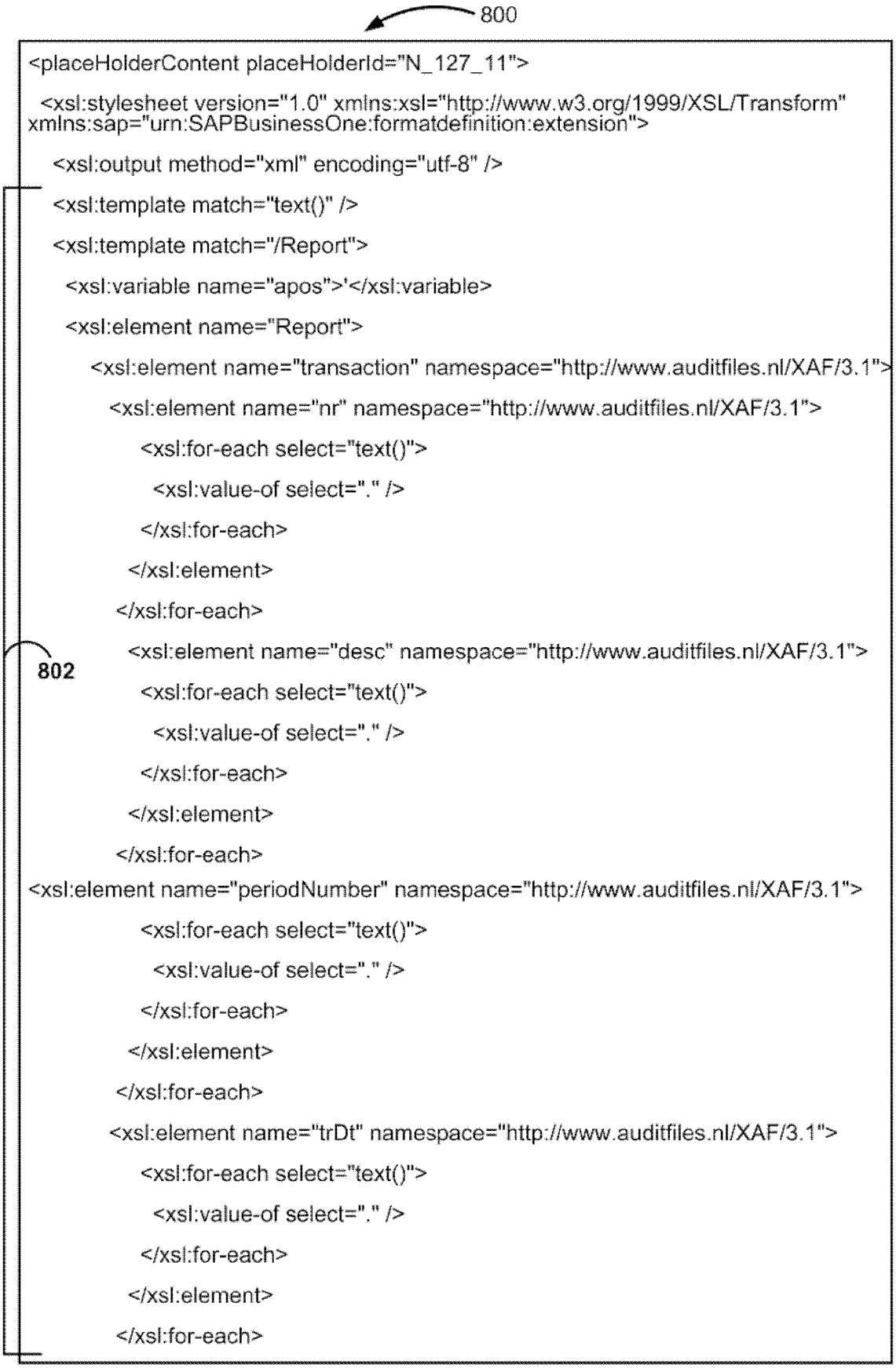

800
802
<placeHolderContent placeHolderId="N_127_11">
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:sap="urn:SAPBusinessOne:formatdefinition:extension">
<xsl:output method="xml" encoding="utf-8" />
<xsl:template match="text()" />
<xsl:template match="/Report">
<xsl:variable name="apos">'</xsl:variable>
<xsl:element name="Report">
<xsl:element name="transaction" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:element name="nr" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>
<xsl:element name="desc" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>
<xsl:element name="periodNumber" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>
<xsl:element name="trDt" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>

图 8A


800
802
804
<xsl:element name="sourceID" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>
</xsl:for-each>
</xsl:element
</xsl:for-each>
<placeHolderContent placeHolderId="N_128_6">
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:sap="urn:SAPBusinessOne:formatdefinition:extension">
<xsl:output method="xml" encoding="utf-8" />
<xsl:template match="text()" />
<xsl:template match="/Report">
<xsl:variable name="apos">'</xsl:variable>
<xsl:element name="Report">
<xsl:element name="subledger" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:element name="sbType" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>
<xsl:element name="sbDesc" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
<xsl:element name="totalDebit" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>
<xsl:element name="totalCredit" namespace="http://www.auditfiles.nl/XAF/3.1">
<xsl:for-each select="text()">
<xsl:value-of select="." />
</xsl:for-each>
</xsl:element>
</xsl:for-each>

图 8B

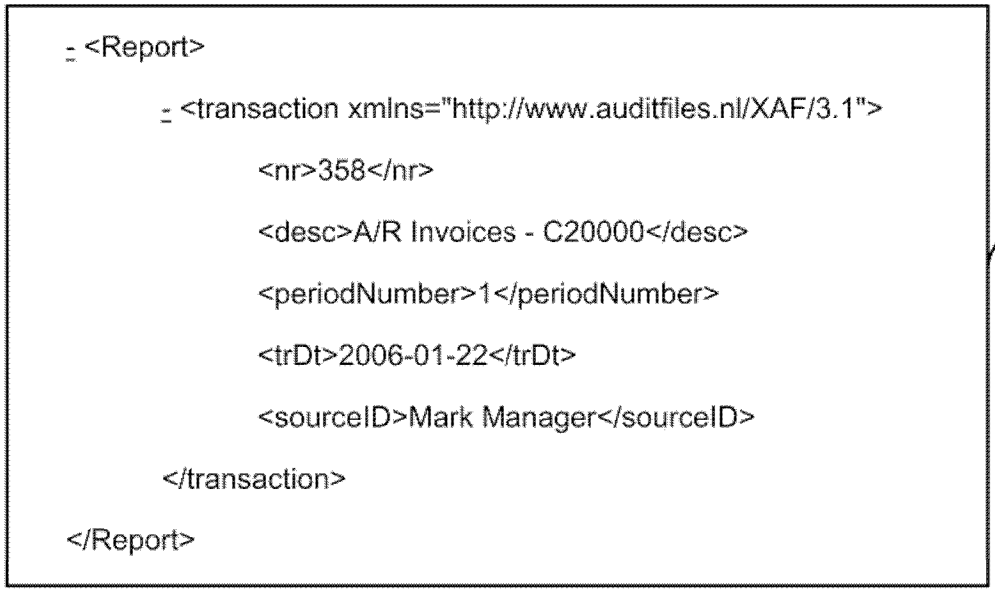

- <Report>
- <transaction xmlns="http://www.auditfiles.nl/XAF/3.1">
<nr>358</nr>
<desc>A/R Invoices - C20000</desc>
<periodNumber>1</periodNumber>
<trDt>2006-01-22</trDt>
<sourceID>Mark Manager</sourceID>
</transaction>
</Report>
900

图 9

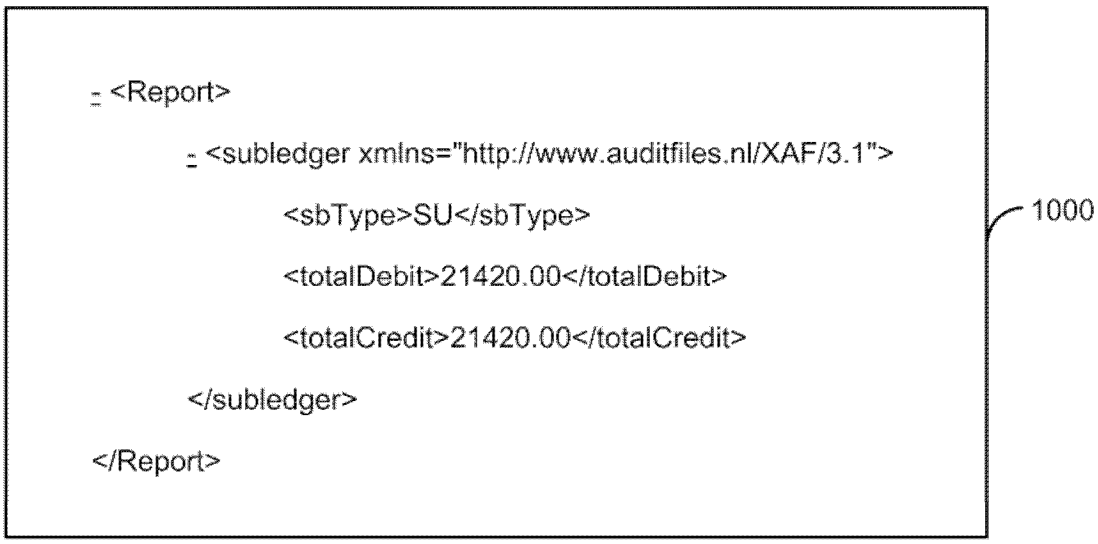

- <Report>
- <subledger xmlns="http://www.auditfiles.nl/XAF/3.1">
<sbType>SU</sbType>
<totalDebit>21420.00</totalDebit>
<totalCredit>21420.00</totalCredit>
</subledger>
</Report>
1000

图 10

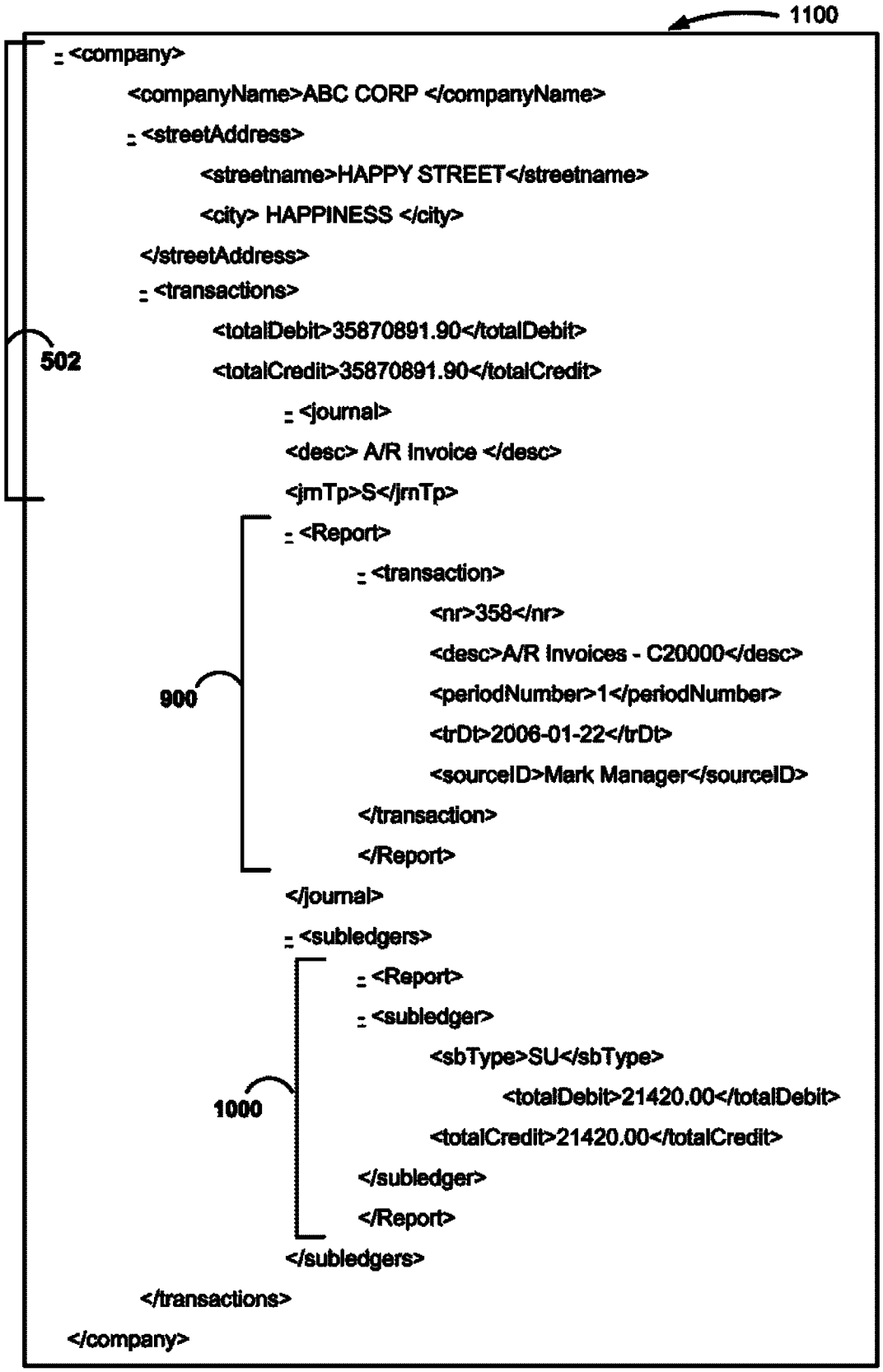

1100
<company>
<companyName>ABC CORP </companyName>
<streetAddress>
<streetname>HAPPY STREET</streetname>
<city> HAPPINESS </city>
</streetAddress>
<transactions>
<totalDebit>35870891.90</totalDebit>
<totalCredit>35870891.90</totalCredit>
502
<journal>
<desc> A/R Invoice </desc>
<jrnTp>S</jrnTp>
<Report>
<transaction>
<nr>358</nr>
<desc>A/R Invoices - C20000</desc>
900
<periodNumber>1</periodNumber>
<trDt>2006-01-22</trDt>
<sourceID>Mark Manager</sourceID>
</transaction>
</Report>
</journal>
<subledgers>
<Report>
<subledger>
<sbType>SU</sbType>
<totalDebit>21420.00</totalDebit>
1000
<totalCredit>21420.00</totalCredit>
</subledger>
</Report>
</subledgers>
</transactions>
</company>

图 11

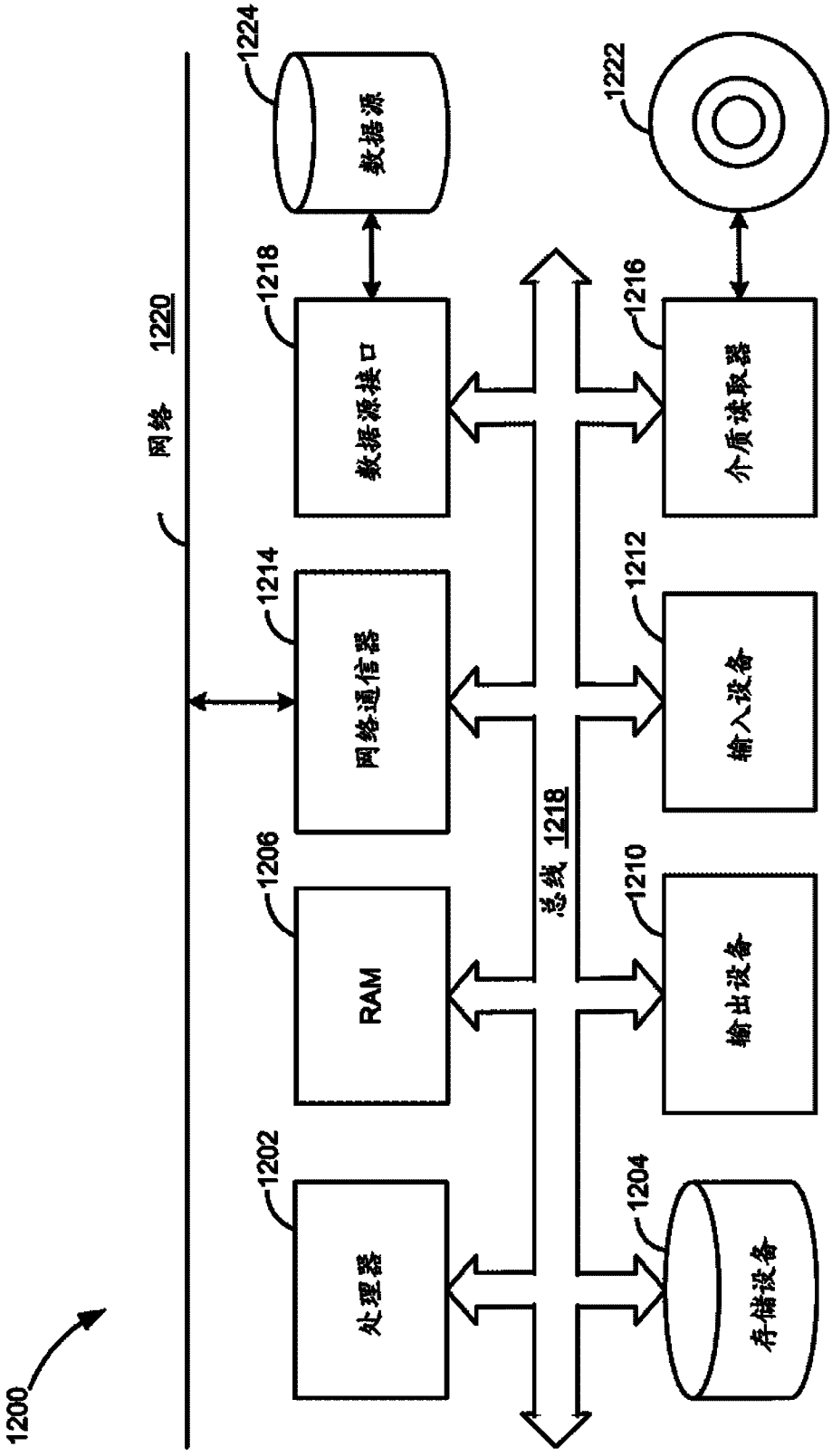

1200
网络 1220
1224
数据源
1222
1218
数据源接口
1216
介质读取器
1214
网络通信器
1212
输入设备
总线 1218
1206
RAM
1210
输出设备
1202
处理器
1204
存储设备

图 12