

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06Q 30/00 (2006.01)

G06F 17/21 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200680047362.X

[43] 公开日 2008 年 12 月 24 日

[11] 公开号 CN 101331511A

[22] 申请日 2006.11.17

[21] 申请号 200680047362.X

[30] 优先权

[32] 2005.12.16 [33] US [31] 11/303,167

[86] 国际申请 PCT/US2006/044690 2006.11.17

[87] 国际公布 WO2007/075234 英 2007.7.5

[85] 进入国家阶段日期 2008.6.16

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 S·玛什拉洙 S·高拉夫

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 陈 斌

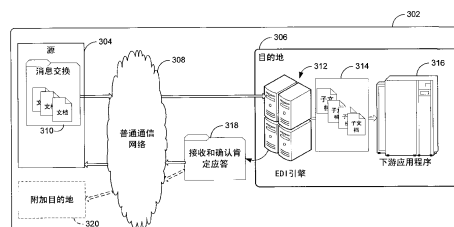
权利要求书 3 页 说明书 29 页 附图 27 页

[54] 发明名称

用于电子数据交换 (EDI) 的 XML 规范

[57] 摘要

用于变换电子数据交换 (EDI) 交易的可扩展标记语言 (XML) 规范。 批量接收 EDI 数据的集合。 该批量 EDI 数据包括多个 EDI 文档，并且这多个 EDI 文档中的每一个具有对应于一交易类型的至少一个 EDI 交易。 包括在 EDI 文档中的 EDI 交易通过根据 EDI 标准对所接收的 EDI 数据进行解码来标识。 从批量 EDI 数据中 EDI 文档生成一合并的 EDI 文档。 该合并的 EDI 文档包括根据交易类型组织的所标识的 EDI 交易。



1. 一种至少部分地由计算设备实现的、用于变换电子数据交换（EDI）交易的方法，所述方法包括：

批量接收 EDI 数据，所述批量 EDI 数据包括多个 EDI 文档，每一 EDI 文档具有对应于一交易类型的至少一个 EDI 交易；

通过根据 EDI 标准对所接收的 EDI 数据进行解码来标识包括在所述 EDI 文档中的 EDI 交易；以及

从所述批量的多个 EDI 文档中生成一合并的 EDI 文档，所述合并的 EDI 文档包括根据所述交易类型组织的所标识的 EDI 交易的 EDI 数据。

2. 如权利要求 1 所述的方法，其特征在于，所述合并的 EDI 文档是可扩展标记语言（XML）文档。

3. 如权利要求 2 所述的方法，其特征在于，还包括按照 XML 标签来组织包括在所述合并的 EDI 文档中的 EDI 交易，其中所述 XML 标签指示所述 EDI 交易的交易类型。

4. 如权利要求 1 所述的方法，其特征在于，所述合并的 EDI 文档包括一乳模式，所述乳模式表示被所述 EDI 交易参考的多个模式。

5. 如权利要求 4 所述的方法，其特征在于，还包括定义用于变换所述多个模式以便在运行时期间处理的运行时模式映射。

6. 如权利要求 1 所述的方法，其特征在于，还包括根据包括在所述 EDI 交易中的一个或多个信息将所述合并的 EDI 文档中的 EDI 交易组织为子文档，并且其中所述子文档中的 EDI 交易与所述多个 EDI 文档中的 EDI 交易相关。

7. 如权利要求 1 所述的方法，其特征在于，一个或多个计算机可读介质具有用于执行如权利要求 1 所述的方法的计算机可执行指令。

8. 一种用于在发送源和接收设备之间变换电子数据交换（EDI）交易的系统，所述系统包括：

用于批量发送 EDI 数据的发送源，所述批量 EDI 数据包括多个 EDI 文档，每一 EDI 文档具有对应于一交易类型的至少一个 EDI 交易；

用于接收所述批量 EDI 数据的接口；

用于执行计算机可执行指令的处理器，所述指令用于：

通过根据 EDI 标准对所接收的 EDI 数据进行解码来标识包括在所述多个 EDI 文档中的 EDI 交易；以及

从所述批量 EDI 数据的多个 EDI 文档中定义一合并的 EDI 文档，所述合并的 EDI 文档包括根据所述交易类型组织的所标识的 EDI 交易。

9. 如权利要求 8 所述的系统，其特征在于，所述合并的 EDI 文档是可扩展标记语言（XML）文档。

10. 如权利要求 9 所述的系统，其特征在于，所述处理器还被配置成按照 XML 标签来组织包括在所述合并的 EDI 文档中的 EDI 交易，其中所述 XML 标签指示所述 EDI 交易的交易类型。

11. 如权利要求 8 所述的系统，其特征在于，所述合并的 EDI 文档包括一乳模式，所述乳模式表示被所述 EDI 交易参考的多个模式。

12. 如权利要求 11 所述的系统，其特征在于，所述接口还接收用于变换所述多个模式以便在运行时期由所述的处理器处理的运行时模式映射。

13. 如权利要求 12 所述的系统，其特征在于，所述处理器被配置成应用所述运行时模式映射以在运行时期匹配所述多个模式。

14. 如权利要求 8 所述的系统，其特征在于，所述处理器还被配置成根据包括在所述 EDI 交易中的一个或多个信息将所述合并的 EDI 文档中的 EDI 交易组织为子文档，并且其中所述子文档中的 EDI 交易与所述多个 EDI 文档中的 EDI 交易相关。

15. 一种或多种具有用于变换电子数据交换（EDI）交易的计算机可执行组件的计算机可读介质，所述计算机可执行组件包括：

用于从一发送源批量接收 EDI 数据并将所接收的 EDI 数据发送给一接收方的接口组件，所述批量 EDI 数据包括多个 EDI 文档，每一 EDI 文档具有对应于一交易类型的至少一个 EDI 交易；

用于通过根据 EDI 标准对所接收的 EDI 数据进行解码来标识包括在所述多个 EDI 文档中的 EDI 交易的标识组件；以及

用于从所述批量 EDI 数据的多个 EDI 文档中定义一合并的 EDI 文档的变换组件，所述合并的 EDI 文档包括根据所述交易类型组织的所标识的 EDI 交易。

16. 如权利要求 15 所述的计算机可读介质，其特征在于，所述合并的 EDI 文档是可扩展标记语言（XML）文档。

17. 如权利要求 16 所述的计算机可读介质，其特征在于，还包括用于按照

XML 标签来组织包括在所述合并的 EDI 文档中的 EDI 交易的文档组件，其中所述 XML 标签指示所述 EDI 交易的交易类型。

18. 如权利要求 17 所述的计算机可读介质，其特征在于，所述文档组件还根据包括在所述 EDI 交易中的一个或多个信息将所述合并的 EDI 文档中的 EDI 交易组织为子文档，并且其中所述子文档中的 EDI 交易与所述多个 EDI 文档中的 EDI 交易相关。

19. 如权利要求 15 所述的计算机可读介质，其特征在于，所述合并的 EDI 文档包括一乳模式，所述乳模式表示被所述 EDI 交易参考的多个模式。

20. 如权利要求 15 所述的计算机可读介质，其特征在于，所述接口组件还接收用于变换所述多个模式以便在运行时期由所述的处理器处理的运行时模式映射。

用于电子数据交换（EDI）的 XML 规范

背景

商业实体经常以严格的格式通过普通通信网络发送商业交易数据以便于处理交易。例如，电子数据交换（EDI）是企业利用范围不断扩展的自动化计算系统的方式之一。

在 EDI 中，商业数据是根据诸如 ANSI X12 或 EDIFACT 等一种或多种已知且核准的标准来格式化的。例如，表示各种交易的 EDI 数据是作为一批描绘文档来发送的，并且每一描绘文档是根据严格的格式化规则来编码的，以确保接收这些文档的目的地应用程序能够成功地解析并使用该信息来进行下游的处理。

在解析和处理 EDI 消息时，现有系统发送 EDI 数据，并且在交换期间在每一描绘文档中包括格式化规则或模式。例如，表示采购定单交易的 EDI 数据包括用于采购定单交易的模式。由此，每一 EDI 交易文档既包括 EDI 数据，又包括用于该交易的特定模式。尽管这一安排或配置便于解析 EDI 数据，但是它是静态的，并且使得每一交易文档的文档大小很大。另外，所包括的模式不是可共享的。换言之，如果有两个采购定单交易文档 A 和 B，则每一采购定单交易文档需要包括一采购定单模式，即使每一文档中的模式是相同的。并且，特别地，EDI 交易是根据行或文档的数量以及发送 EDI 数据所需的带宽来收费的。由于商业实体使用 EDI 每天都发送数百万的交易，这些包括重复模式信息的大型 EDI 交易文档造成了对具有冗余模式信息的不必要的成本。

一旦接收到 EDI 交易文档，目的地应用程序通常将该 EDI 交易文档储存在存储器区域中。目的地应用程序接着向源发送指示该交易已被接收的接收肯定应答。所储存的 EDI 交易随后由应用程序确认来确定包括在该交易文档中的 EDI 数据是否符合用于该交易类型的模式的格式化规则。在此确认期间，要求源（例如，商家或客户）等待指示该交易数据符合该格式的确认肯定应答。如果确定一个或多个交易没有被正确地格式化，则需要重新发送替换的 EDI 交易文档以便处理。这一等待确认延迟进一步降低了处理 EDI 交易的效率。

概述

本发明的各实施例通过将各 EDI 交易文件变换成具有标识各种 EDI 交易类型的嵌套结构或子文档的一个 EDI 文档来克服现有系统在处理 EDI 交易时的缺点。另外,本发明的各方面通过使得模式实例在运行时处理 EDI 交易时可用来使得 EDI 文档能引用模式。有利的是,本发明的各实施例自动识别与交易类型相关联的模式,并在接收到 EDI 交易时处理该 EDI 交易。根据本发明的其它实施例,在接收到 EDI 交易时处理该 EDI 交易。

在本发明的又一实施例中,定义一单一元模式来表示多个模式。该单一元模式被提供给最终用户以修改该模式的特性。

提供本概述以便用简化的形式介绍将在以下详细描述中进一步描述的一些概念。本概述并不旨在确定所要求保护的主题的关键特征或必要特征,也不旨在用于帮助确定所要求保护的主题的范围。

其它特征将部分变得清楚,部分在以下指出。

附图简述

图 1 是示出处理 EDI 交易的一个实现的框图。

图 2A 到 2C 是示出根据本发明的一个实施例的使用电子数据交换(EDI)的交易数据的结构。

图 3 是示出根据本发明的一个实施例的用于变换 EDI 交易的系统的示例性框图。

图 4A 和 4B 是示出根据本发明的一个实施例的 EDI 交易的变换的流程图。

图 5A 是示出根据本发明的一个实施例的 EDI 交易的嵌套的框图。

图 5B 和 5C 是示出根据本发明的一个实施例的串行化 EDI 交易的框图。

图 6A 和 6B 是示出根据本发明的一个实施例的包括在合并的可扩展标记语言(XML)文档格式的 EDI 文档中的经变换的 EDI 交易的屏幕截图。

图 7A 到 7D 是示出根据本发明的一个实施例的自动标识 EDI 模式的屏幕截图。

图 8A 是示出根据本发明的一个实施例的确认 EDI 交易的流程图。

图 8B 是示出根据本发明的一个实施例的检测 EDI 交易中的错误的图示。

图 9A 和 9B 是示出根据本发明的一个实施例的 EDI 确认肯定应答结构的图示。

图 10 是示出根据本发明的一个实施例的用于修改多个 EDI 模式的单一元模式的屏幕截图。

图 10A 是示出根据本发明的一个实施例的用于使用单一元模式来修改多个 EDI 模式的方法的流程图。

图 11A 到 11D 是示出其上可储存本发明的各方面的示例性计算机可读介质的框图。

图 12 是示出其中可实现本发明的合适的计算系统环境的一个实例的框图。

附录 A 完整描述了图 10A 所示的 XML 模式。

附录 B 示出了表示采购定单模式的 XML 格式的示例性单一元模式。

在全部附图中，相应的参考标号指示相应的部分。

详细描述

图 1 是示出处理 EDI 交易的一个实现的框图。最初，如图 1 所示，源（例如，商业合伙人）102 通过普通通信网络 108 向目的地（例如，商业客户）104 发送 EDI 消息 106，该消息可包括发票 202。

源 102 经由普通通信网络 108 向目的地 104 发送包括模式和 EDI 交易数据的 EDI 消息 106。在一个实例中，EDI 消息 106 包括批量的多个 EDI 交易数据，以便节省传输或带宽成本。在另一示例中，普通通信网络 106 可以是诸如内联网等私有的专用网络，或诸如因特网等公共网络。在另一示例中，普通通信网络 108 包括一个或多个网络协议，诸如 FTP、HTTP、Kermit、Xmodem、帧延迟、EDIINT、3780 Bisync®等，以便于在合伙人之间传送 EDI 消息。

源 102 通过打开经由普通通信网络 108 与目的地 104 的连接会话（例如，安全套接字连接会话）来发起 EDI 消息 106 的传输。一旦打开了连接会话，源 102 就向目的地 104 发送 EDI 消息 106。目的地 104 上的一组 EDI 转换器系统 110 接收 EDI 消息 106，并且该 EDI 转换器系统 110 经由普通通信网络 108 向源 102 发送指示该 EDI 消息已被接收到的接收肯定应答 112。常见的是接收确认应答在源 102 关闭会话连接之前被发送会返回给源 102。

一旦接收到 EDI 消息 106，EDI 转换器系统 110 就解析并处理与 EDI 交易相关联的 EDI 数据。如本领域的技术人员已知的，对 EDI 交易的解析和/或解码涉及标识各种 EDI 标准、模式规范等的一个或多个步骤。在此期间，EDI 转换器系统 110 将经解析或解码的 EDI 数据发送给下游应用程序 114 以处理该经解析或解码的 EDI 数据。例如，下游应用程序 114 可以是处理发票的会计应用程序，或者是用于处理采购定单数据的软件。由此，下游应用程序 114 能够确认所接收到的 EDI 数

据在解析和解码之后是否符合在模式中指定的格式化规则。如果所接收到的 EDI 数据符合模式规则，则下游应用程序 114 向源 102 发送一确认肯定应答 116。另一方面，如果该接收到的 EDI 数据包括错误或是无效的，则下游应用程序 114 可以向源发送指示所接收到的 EDI 数据的错误的错误通知。

确认肯定应答 116 通常以接收肯定应答的发送之后的延迟来发送给源 102。在其它实现中，经解析的 EDI 数据被储存在一数据库或数据存储（未示出）中以等待确认。由此，源 102 经常被要求等待确认肯定应答 116 来查明该 EDI 数据能够被目的地 104 正确地处理。

图 2A 到 2C 是示出根据本发明的一个实施例的使用电子数据交换（EDI）的交易数据的结构的图示。图 2A 示出了使用 ANSI X12 格式的发票 EDI 交易文档的表示的示例。在此示例中，发票 202 包括多个段或部分（见图 2C 中的 X12 EDI 交换结构 218 的概观），诸如功能组 204 部分，它可包括发票 202 的附加信息。例如，在供应链部门中，本领域的技术人员已知功能组 204 中的信息或值与交换部分（例如，交换控制头）中的信息或值是相同的，如图 2C 所示。在另一示例中，功能组 204 中的信息或值包括标识大企业内的商业或营业单元的值或标识符。

发票 202 还包括头部分 206，它包括诸如商业客户信息等信息。在此示例中，头部分 206 包括商业客户名称“ABC 公司”以及地址“密苏里州，圣路易市，第六大街 0887 号，63101”。在一个实施例中，头部分 206 包括用于接收确认肯定应答的目的地信息，参见以下对图 8、9A 和 9B 的讨论。发票 202 还包括示出以循环 210 组织的一个或多个数据段 212 的细节表部分 208。例如，循环 210 包括一组语义上相关的数据段，并且对本领域的技术人员而言，这些段可以根据 ANSI X12.6 是有界或无界的。

根据 ANSI X12 或 EDIFACT 格式，在 EDI 交易文档中可包括其它段类型和部分以及相应的信息，而不背离本发明的范围。例如，图 2B 示出了包括在要在目的地 104 处处理的同一 EDI 消息 106 中的一个或多个交易类型。发票 214 和采购定单 216 EDI 交易文档被包括在 EDI 消息 106 中，因为发票 214 和采购定单 216 与同一客户，即“ABC 公司”相关。其它各组相关交易文档可以作为 EDI 消息 106 被包括在交换中。在一个实施例中，用于一个目的地或客户的 EDI 文档可以批量发送。

还可以理解，要求每一 EDI 交易类型都符合与该交易类型相关联的模式。例如，特别地，发票交易模式可要求关于用于商家或买家的名称的最大或最小字符长

度的限制。采购定单交易模式可要求小数点之后的最大位数。在另一示例中，用于各种交易类型的模式可以指定一特定字段是强制的，而其它字段是可任选的。

现有的实现在向诸如目的地 104 等客户发送 EDI 交易时在 EDI 交易文档中包括交易模式。尽管这些实现便于对 EDI 交易的解码，但是它们要求模式设计者在向商业合伙人发送 EDI 交易之前花费大量时间来设计或配置该模式。并且，由于合伙人之间的商业协定的修改，需要对这些模式的后续修改来重新设计该模式。

由此，本发明的实施例通过将 EDI 消息变换成具有根据交易类型来组织一个或多个 EDI 交易的嵌套结构或子文档的一个合并的 EDI 文档克服了现有实现的缺陷。EDI 文档还包括用于表示与交易类型相关联的多个模式的乳模式（uber-schema）。在另一实施例中，一运行时模式映射变换多个模式以便在目的地 104 处在运行时处理。

现在参考图 3，一框图示出了用于根据本发明的一个实施例来变换 EDI 交易的系统 302 的框图。系统 302 包括源 304，源可以是与目的地 306 或客户交易商业的商家。例如，源 304 可以是诸如向购买计算设备的公司客户销售大量商品的电子消费品零售商店等商家。在另一示例中，源 304 可以是健康护理提供者，诸如医院或药店，并且向健康护理保险公司或交换所发送 EDI 数据以便提交主张或符合健康保险便利及责任法案（HIPAA）。

在一个实施例中，源 304 和目的地 306 包括诸如图 12 中的计算机 130 等一个或多个计算设备，用于批量发送 EDI 文档。最初，源 304 发送包括多个 EDI 文档的 EDI 消息 310。每一 EDI 文档包括对应于一交易类型（例如，发票、采购定单、应付款等）的至少一个 EDI 交易。

还参考图 4A，一流程图示出了根据本发明的一个实施例的变换 EDI 交易。在源 304 打开通信网络 308 上的连接会话以与目的地 306 通信之后，源 304 向目的地 306 的 EDI 引擎 312 发送 EDI 消息 310。在一个实施例中，EDI 引擎 312 包括执行计算机可执行指令、例程或函数的一个或多个计算设备（例如，计算机 130）。在 402 处，EDI 引擎 312 接收包括多个 EDI 文档的 EDI 消息 310。在 404 处，EDI 引擎 312 标识包括在多个 EDI 文档中的 EDI 交易。使用以上的 ANSI X12 示例，EDI 引擎 312 通过标识图 2C 所示的各种数据头和数据段（例如，ISA、GS 等）来解码或解析 X12 发票，以确定交易中的 EDI 数据。在另一实施例中，EDI 引擎 312 还标识包括在多个 EDI 文档中的各种模式来确定用于该交易类型的特定格式化规则。

在 406 处，EDI 引擎 312 从该批多个 EDI 文档中生成合并的 EDI 文档 314。

在一个示例中，EDI 引擎 312 在 410 处生成合并的 EDI 文档 314 作为具有 XML 结构标记标签的 XML 文档。例如，不像其中每一交易被组织为一文档的现有的实现，本发明的各实施例将多个 EDI 文档中的 EDI 交易组织成一个 XML 文档，该 XML 文档不仅定义了个别的交易集，而且还通过捕捉 EDI 数据的所有方面，包括段、循环、字段、定界符等，定义了交换。在一个示例中，图 6A 示出了包括一个或多个 EDI 交易（诸如“PO（采购定单）”）的示例性合并的 XML 文档。

在又一实施例中，该合并的 EDI 文档 314 包括表示由 EDI 交易引用的多个模式的乳模式。例如，该乳模式被包括在 EDI 交易集中，并且被嵌入或缝合在每一 EDI 交易的功能组和信封段（envelope segment）内，使得最终用户无需为期望被包括在 EDI 消息 310 中的每一交易集创建一特定模式。作为一个示例，图 6B 示出了一屏幕截图，其示出了根据本发明的一个实施例的合并的 EDI 文档 314 中的 XML 格式的乳模式。采用这一设计，合并的 EDI 文档的交换减少了在 EDI 文档 314 中包括对应于一交易类型的一个或多个模式的需求。本发明的各实施例还减少了传输之前的模式设计和开发时间。

在另一实施例中，在图 4B 的 412 处，EDI 引擎 312 用运行时模式映射或净荷模式来变换合并的 EDI 文档。在 414 处，EDI 引擎 312 为合并的 EDI 文档 314 中的 EDI 交易创建子文档或嵌套结构（见表 1 和 2 中的附加描述）。在一个替换实施例中，该合并的 EDI 文档 314 是用净荷模式（例如，运行时模式映射）来变换的，并且还可在 416 处在结构上进行变换。或者，合并的 EDI 文档 314 可被发送到下游应用程序 316 以便在 418 处在没有结构变换的情况下进行处理。在 420 处，具有子文档或嵌套结构的合并的 EDI 文档 314 也被发送到下游应用程序 316 以供处理。

可以理解，可以使用除用于合并的 EDI 文档 314 的 XML 之外的、带有以可标识结构来定义并组织 EDI 交易的标记标签的其它格式，而不背离本发明的范围。

在另一实施例中，描述了其上可储存上述本发明的各方面的计算机可读介质 1102（在图 11A 中）。例如，接口组件 1104、标识组件 1106 以及变换组件 1108 可被包括在执行以上讨论的一个或多个操作的 EDI 引擎 312 中。

还可以理解，图 4A 所示的方法可以由源 304 来执行，使得源 304 可以在传输之前减小交换的大小。由此，合并的 EDI 文档 314 的嵌套结构或子文档减少了行数，这还可以减少当根据行数来收费时的发送 EDI 数据的成本。

例如，表 1 示出了合并的 EDI 文档中的嵌套结构中的三个 EDI 交易，以及各

自包括这三个 EDI 交易中的一个的相应的三个原始 EDI 文档。

嵌套结构中的 EDI 交易	用于下游处理的展平的 EDI 事务
交易#1 的开始	交易#1a 的开始
PO 头段	PO 头段
PO 行 1	PO 行 1
PO 进度表 1	PO 进度表 1.1
PO 进度表 2	PO 行 1 总和
PO 行 1 总和	PO 总和
PO 行 2	交易#1a 的结束
PO 进度表 2.1	
PO 行 2 总和	交易#1b 的开始
PO 总和	PO 头段
交易#1 的结束	PO 行 1
	PO 进度表 1.2
	PO 行 1 总和
	PO 总和
	交易#1b 的结束
	交易#1c 的开始
	PO 头段
	PO 行 2
	PO 进度表 2.1
	PO 行 2 总和
	PO 总和
	交易#1c 的结束

表 1：嵌套结构（左列）和三个 EDI 文档（右列）中的三个 EDI 交易

在操作中，假设诸如雇主 A 等健康护理发起者需要向诸如健康护理提供者 B 等付款人发送诸如 HIPAA 834 文档等 EDI 消息。用于这一交换的模式要求雇主 A 提供健康护理受益人/接收者（例如，雇员及其家属）的福利的细节。由此，雇主 A 通常包括发起者和付款人的详细信息。发起者和付款人的该详细信息对于所有受益

人都是共同的，并且对正在接收由雇主 A 发起的福利的每一雇员或家属重复。本发明的各实施例创建一嵌套结构，使得每一成员可连同发起者和付款人的详细信息的副本一起以一个 EDI 文档中的类似循环的逻辑来创建，而非如在现有的 EDI 实现中那样对数千个雇员和家属重复相同的发起者和付款人信息。

图 5A 是示出根据本发明的一个实施例的 EDI 交易的嵌套的框图。例如，在 502 处，在目的地（例如，目的地 306）处从源（例如，源 304）接收 EDI 消息（例如，EDI 消息 310）。在 504 处，生成一合并的 EDI 文档，其中各 EDI 交易被包括在一嵌套结构中或作为子文档来包括。在一个示例中，剥去信封/控制段（例如，ANSI X12 格式中的 ISA/GS/GE/IEA 段），并通过接收管线解析交易集（ST/SE）来对每一交易集生成多个 XML 子文档。在一个实施例中，多个 XML 子文档被存放在一消息箱中。在 506 处，目的地处的接收管线执行对传入交换的确认，并生成适当的确认肯定应答（将在图 8、9A 和 9B 中详细讨论）。在一个实施例中，接收管线还更新校验和商业总和。

如上所述，合并的 EDI 文档 314 可以由下游应用程序 316 来处理。由此，该合并的 EDI 文档被发送给一发送端口，并且在 508 处，该发送端口发送 EDI 子文档中的 EDI 交易。在一个实施例中，与发送端口相关联的发送管线串行化 XML 子文档，并传送 ‘n’ 个交换，其中在发送管线处更新段的计数。

在一个实施例中，当接收到一 EDI 交换时，确认该交换。如果没有任何确认错误，则根据其模式将每一交易集转换成 XML 格式。由此，一个 EDI 交换可包含采购定单和发票文档。采购定单可以被转换成符合采购定单模式的 XML。同样，发票将被转换成发票 XML。

图 5B 示出了 XML 格式的来自 EDI 交换的示例性采购定单。当该采购定单文档由图 5A 中的发送侧处理时，它在处理了信封段之后被转换成 EDI 格式 514。图 5C 示出了由发送端口从图 5B 中的 XML 格式产生的示例性文档。在一个实施例中，EDI 格式 514 包括两个信封段（例如，以 ISA 和 GS 开头的行）。类似地，EDI 格式 514 在该文档的末尾包括两个信封段，GE 和 IEA。如图所示，包括在 ST 和 SE 段之间的数据是用于原始交易集的数据。

在如图 5B 和 5C 所示的以上示例中，SE01 的值（见箭头 512）是“14”，并且是由发送端口动态计算的。尽管串行化一 EDI 文档，但是 EDI 引擎（例如，EDI 引擎 312）的发送侧跟踪一交易集中存在的段数。基于该值，确定 SE01 的值。

在源 304 生成合并的 EDI 文档 314 以包括来自多个 EDI 文档的 EDI 交易的情况

下，本发明的各实施例包括将所包括的 EDI 交易组织在一嵌套结构中。在另一示例中，本发明的各实施例允许从源接收合并的 EDI 文档 314 的目的地 306 为了后向兼容性或为了适应只能处理每一文档仅包含一个交易的 EDI 文档的下游应用程序 316 而从合并的 EDI 文档 314 中还原多个 EDI 文档。本发明的替换实施例允许具有嵌套结构中的 EDI 交易的合并的 EDI 文档跟踪原始的多个 EDI 文档或与这些原始文档相关。

例如，表 2 示出了将 EDI 交易从合并的 EDI 文档 314 转换成多个 EDI 文档。

A0	A1	A2	A3	A4
模式 (最少出现次数和最大出现次数)	原始 实例	拆分#1	拆分#2	拆分#3
ST (1,0) AA (1,1) BB 循环 (1, n) – 子文档断开 = 是 BB1(1, 1) BB2(0, 1) CC (1, n) DD (0, n) SE	ST AA BB1*1 BB2*1 <u>BB1*2</u> <i>BB1*3</i> <i>BB2*3</i> CC CC DD SE	ST AA BB1*1 BB2*1 CC CC DD SE	ST AA <u>BB1*2</u> CC CC DD SE	ST AA <i>BB1*3</i> <i>BB2*3</i> CC CC DD SE

表 2：合并的 EDI 文档转换

在表 2 所示的示例中，对嵌套结构中的 EDI 交易的处理通过标识预定的子文档创建断点（SubDocumentCreationBreakPoint）（例如，描述一子文档在父文档中的何处开始的“\”标记）以生成多个子文档来开始。

根据表 2，在列 A1 中所示的合并的 EDI 文档可根据在模式中的 BB 循环处定义的子文档创建断点而被拆分成三个交易：**BB1*1-BB2*1**、BB1*2 以及 *BB1*3-BB2*3*。在列 A2 中，交易集 **BB1*1-BB2*1** 被组织或拆分（由粗体文本表示）成一单独的文档，而在列 A3 中，交易 BB1*2 被组织在第二文档中（由下划线文本表示）。类似地，交易集 *BB1*3-BB2*3* 被组织成要由下游应用程序 316 处理的第三 EDI 文档（由斜体文本表示）。

通过将包括在多个 EDI 文档中的 EDI 交易变换成合并的 EDI 文档 314，本发明的各实施例使得目的地 306 或源 304 能够在变换期间有效地标识包括在每一 EDI 文档中的多个模式。另外，本发明的至少一方面使得目的地 306 在变换了合并的

EDI文档之后在连接会话仍然被打开的时间段内生成要在返回给源 304 的确认肯定应答。换言之，本发明的各方面将目的地 306 配置成在接收 EDI 交易的同时自动标识多个模式并确认 EDI 交易。

现在参考图 7A-7D，示出了根据本发明的一个实施例的自动标识 EDI 模式的一系列屏幕截图。图 7A 示出了典型的 ANSI X12 采购定单模式。模式是由与其相关联的 DocType（文档类型）来标识的。DocType 是诸如名称空间和根节点名称等配置项的组合。如图 7A 所示，该屏幕截图的左列 702 示出了一模式的分层结构。在此示例中，左列 702 示出了一模式结构。中列 704 指示了该模式的 XML 代码。右列 706 指示了包括在该模式中的特性或目标名称空间。

在一个实施例中，DocType 具有以下格式：X12 格式的“DocType = TargetNamespace ‘#’ RootNodeName”，这将在以下详细描述。可以理解，尽管在图 7A 中描述了 X12 模式，但是可以使用诸如 EDIFACT 模式等其它模式格式而不背离本发明的范围。

DocType 的根节点具有以下 X12 格式之一：“X12_{Version}_{Tsid}”。在此示例中，配置项“root node”（根节点）的值是“X12_00401_850”，如由框 708 所示。换言之，“00401”是文档的版本，并且它是在运行时处理期间确定配置或实例的一条动态信息。类似地，“850”是 Tsid，它是所处理的模式的交易标识（ID），并且是从输入实例中确定的。在此示例中，交易 ID “850”表示采购定单，如由框 710 所指示的。并且，目标名称空间由右列 706 中的框 712 来指示。

在另一示例中，为解码或标识 EDIFACT 格式的模式，EDIFACT 模式当前具有以下格式：“Efact_{Version}_{Tsid}”。换言之，所有 EDIFACT 模式都具有以“Efact”开头的根节点名称，并且 Version 和 Tsid 的定义与 X12 格式的相同。

使用图 3 作为一个示例，当目的地 306 从源 304 接收到 EDI 文档时，EDI 交易可随文档一起包括交易 ID “850”。然而，版本信息或目标名称空间信息是在运行时确定的，并且这些配置项的值可以在不同的级别下配置。由此，在应用了根据 EDI 标准（X12 或 EDIFACT）的规则以根据相应的交易类型（例如，采购定单、发票等）来解码 EDI 交易之后，EDI 引擎 312 标识经解码的 EDI 交易中的配置项。在一个实施例中，EDI 引擎 312 从一个或多个配置级别标识配置项，诸如合伙人级别和发送应用程序级别、全局级别、管线级别或默认级别。

例如，图 7B 示出了显示参与者级别配置中的配置项的屏幕截图。在此示例中，用于以上图 7A 所示的合伙人的交易 ID 850 被配置成使用如上所示的目标名称空间

和版本信息。对于所有其它文档类型，可使用默认值，因为如由框 714 所指示的，默认标志或参数被打开。在另一示例中，另一贸易合伙人可基于在商业贸易合伙人之间建立的商业协定在合伙人级别配置中设置其它特定配置项。本发明的各实施例自动标识模式时通过确定由贸易合伙人从一个或多个配置级别设置的特定值来标识配置项的值，而非静态地确定配置项的值。

在一个实施例中，由于发送者 Id 和交易 Id 的特定组合，参与者级别配置中的配置项的值可以被配置为与图 7A 中的 DocType 所示的不同的值。例如，在 X12 中，每一发送者 Id 可以表示企业内的一特定部门。由此，一个企业中的发送者 ID 可以指“硬件商品”部门，而另一发送者 ID 可以指同一企业内的“软件商品”部门。由此，本发明的各实施例识别这些不同的配置，并相应地标识模式。结果，来自一个企业的相同的采购定单可以经历不同的模式标识过程，使得例如根据配置项的值在合并的 EDI 文档 314 中以 XML 生成适当且不同的 EDI 数据。

还可以理解，一个或多个附加配置项可由特定的商业合伙人配置或设置而不背离本发明的范围。例如，一个合伙人可设置最少量的配置，而另一合伙人可在其参与者级别配置中定义详细的配置项。

现在参考图 7C，示出了具有其参与者级别配置的 EDIFACT 模式的屏幕截图。在此示例中，与 X12 模式不同，目标名称空间可以基于发送者应用程序 ID（可任选）（诸如 716 中的 UNG1.2 和 718 中的 UNG2.2）、版本 720（UNG8）、以及交易集 ID 722 的特定组合来配置。换言之，有可能对一发票 EDI 文档有多个配置，每一配置都具有唯一的应用程序 id。在此情况下，可在运行时使用匹配特定应用程序的目标名称空间。在其中没有配置发送者应用程序 ID 的情形中，发送者应用程序 ID 值可以与来自携带相同的交易 ID 的现有记录（例如，日志文件）的任何值进行匹配。在找到多个匹配的情况下，使用默认目标名称空间来确保当存在歧义时使用适当的默认值。

图 7D 是示出用于 X12 模式的全局级别配置的屏幕截图。在此示例中，在诸如目标名称空间或版本等配置项未由交易合伙人指定的情况下，使用全局级别配置中的配置项的值。在此示例中，框 724 指示对版本和目标名称空间没有配置任何值。由此，在运行时不修改配置项的值。

在其中未配置全局级别下的某些缺少的配置项的情况下，使用管线级别或运行时级别配置中的配置项的值。由此，如果在全局级别没有配置目标名称空间，则使用来自管线级别配置的值。在一个实施例中，管线级别配置中的值可由用户设置。

在另一实施例中，图 11B 示出了其上可储存本发明的各方面的计算机可读介质 1110。例如，接口组件 1112 从源批量接收 EDI 文档，其中每个有 EDI 文档具有对应于一交易类型的至少一个 EDI 交易。交易组件 1114 通过应用依照 EDI 标准（例如，X12 或 EDIFACT）的规则，根据相应的交易类型来解码 EDI 交易。配置组件 1116 对解码的 EDI 交易中的每一 EDI 交易标识一个或多个配置项中的值。模式组件 1118 基于配置项的值来确定一个或多个模式类型。

在一替换实施例中，可在运行时修改在以上各节中描述的配置项的值。由此，对于交易类型、目标名称空间、版本的值可以在 EDI 引擎 312 处理 EDI 文档（即，自动标识模式）之后修改。在这一实施例中，改变将在尚待处理的后续文档上反映。本发明的这一动态实现使得目的地 306 处的用户能够在运行时期而非在 EDI 文档从源 304 发送之前的模式设计/配置期间配置值。

在操作中，自动模式标识使得 EDI 合伙人能够流线化对 EDI 文档的处理。不像其中需要对每一合伙人并对每一文档类型配置接收连接和发送连接的现有的实现，EDI 引擎 312 启用了自动模式标识，使得配置项的值在运行时期标识和确定，从而使得 EDI 商业合伙人在处理 EDI 数据时能够更灵活。

可以回想，本发明的至少另一方面包括在接收到 EDI 数据时生成确认肯定应答，图 8A 是示出这一特征的流程图。在 802 处，从源（例如，源 304）向目的地（例如，目的地 306）发送 EDI 消息（例如，EDI 消息 310）。在 804 处，在目的地接收包括 EDI 交易的 EDI 消息。接着，在 806 处，通过确定该 EDI 消息是否预期送往正确的接收方，来确定 EDI 消息的传输是否有效。如果确定 EDI 消息的传输无效，则延缓对 EDI 消息的处理，并且在 808 处生成一交换失败肯定应答。如果确定 EDI 消息的交换是有效的，则接着在 810 处确定 EDI 交易组是否包括错误。

如果组包括错误，则延缓对 EDI 交易组的处理，并且在 812 处生成一功能失败肯定应答。例如，EDI 规范可定义可在组和交易集级别处找到的多个错误。表 3 提供了适用于 X12 EDI 交换的常见错误的列表。

代码	描述—来自 AK905 代码列表
1	功能组不支持
2	功能组版本不支持
3	缺少功能组尾

4	功能组头和尾中的组控制号不一致
5	包括的交易集的数量不匹配实际计数

表 3：功能组错误一与 GS/GE 段有关的错误

例如，EDI 引擎 312 通过标识 EDI 消息中的行/段 GS 的第六个值来确定一错误，诸如错误代码 4，“功能组头和尾中的组控制号不一致”。在图 8B 中，行 GS 532 的第六个值具有值“9”（如由框 528 所指示的）。在确认 EDi 交易时，本发明的各实施例确定在行 GE 534 的第二个值中是否也存在相同的值。如图 8B 所示，行 GE 534 的第二个值是“10”（如由框 530 所指示的）。有了这一差异，确定在该 EDI 消息中存在错误。

在另一示例中，通过标识 GS-GE 段之间的交易集来检测错误代码 5，“所包括的交易集的数量不匹配实际计数”。如图 8B 所示，有一个 GS-GE 段，而 GE 行的第一个值是“02”，指示有两个交易集。由此，该功能组有错误。

然而，如果确定组中没有错误，则接着在 814 处通过评估依照 X12 或 EDIFACT 格式的格式化规则和依照包括在 EDI 交易中的模式的规则来确定每一 EDI 交易是否有效。如果确定 EDI 交易无效，则延缓对 EDI 交易的处理，并在 16 处生成功能失败肯定应答。

例如，表 4 提供了常见的交易错误的列表。

代码	描述—来自 AK502 代码列表
1	交易集不支持
2	缺少交易集尾
3	头和尾中的交易集控制号不匹配
4	所包括的段的数量不匹配实际计数
5	一个或多个段有错误
6	缺少或无效的交易集标识符
7	缺少或无效的交易集控制号

表 4：交易集错误一与 ST 和 SE 内的数据有关的错误

使用图 8B 作为一个示例，EDI 引擎（例如，EDI 引擎 312）通过评估 ST 和 SE 之间的段（行）数来标识错误代码 4，“所包括的段的数量不匹配实际计数”。

在此示例中，该数量是“12”，而 SE 行中的第一个值是 14。由此，在该交易集中有错误，并且这一错误代码可被包括在功能失败肯定应答中。

在一个实施例中，EDI 引擎（例如，EDI 引擎 312）可参考或了解各种错误条件或 EDI 交易规则。在处理 EDI 文档时，EDI 引擎 312 确保没有一个 EDI 格式化规则被违反。在任何违反的时候，EDI 引擎 312 以交换或功能级别肯定应答的形式适当地报告。

或者，如果 EDI 交易有效，则目的地处的 EDI 引擎 312 在 818 处继续处理 EDI 交易。在 820 处，生成指示 EDI 交易有效的确认肯定应答。在一个实施例中，当接收并确认 EDI 消息、EDI 组和/或 EDI 交易时，EDI 引擎 312 可比较并生成合并的确认肯定应答。在另一实施例中，EDI 引擎与接收 EDI 消息、EDI 组和/或 EDI 交易基本同时地生成合并的确认肯定应答。

在 824 处，将所生成的确认肯定应答返回给源，源在 826 处接收该确认肯定应答。在一个实施例中，源打开用于发送 EDI 消息的连接会话并在关闭该同一连接会话之前接收确认肯定应答。由此，在文档确认期间没有数据库或数据存储访问或盘 I/O，因为该确认过程是在运行时期间或在 EDI 交易的接收期间处理的，如由图 3 的箭头 318 所示。在又一实施例中，该确认过程可以通过在运行时插入处理程序来扩展。

在一个替换实施例中，可以在接收 EDI 消息/交易的同时生成并向单独的位置（这一位置信息可以在头部分 106 中找到）发送不同的确认肯定应答类型。由此，本发明的各实施例在一个或多个阶段中（例如，在确认了交换的一方面之后）生成并发送确认肯定应答，或在单个阶段中生成并发送合并的肯定应答。在又一实施例中，这些肯定应答可以被配置成在同一或新的套接字连接会话上传送到不同的目的地，如由图 3 中的箭头 320 所指示的。

例如，假设模式或格式化规则指示针对采购定单的确认肯定应答被配置成发送给企业的客户服务部门，而发票确认肯定应答被配置为发送给同一企业的会计部门。本发明的各方面允许通过打开新的连接会话来向正确的目的地发送相应的肯定应答。图 9A 示出了用于 X12 格式化 EDI 交易的确认肯定应答，而图 9B 示出了用于 EDIFACT 格式化 EDI 交易的确认肯定应答。

在另一实施例中，图 11C 示出了其上可储存本发明的各方面的计算机可读介质。例如，接口组件 1122、肯定应答组件 1124、以及确认组件 1126 可被结合并集成到 EDI 引擎 312 中用于执行如图 8A 所描述的一个或多个步骤。

本发明的其它方面允许修改 EDI 模式而无需最终用户有 EDI 模式开发者那样多的知识。例如，假设在企业内建立一新的部门，但是对该新部门没有采用定制 EDI 模式或规则。代替请求 EDI 模式开发者为该新部门设计一特定的 EDI 模式，本发明的各实施例定义一元模式来表示所有模式，使得各模式的特性被呈现给最终用户以供修改。

图 10A 是示出根据本发明的一个实施例的用于修改多个 EDI 模式的单一元模式的屏幕截图。在窗格 1002 中，单一元模式的结构被呈现给最终用户。一旦最终用户加亮了一特性（由包围“MaxOccurs（最大出现次数）”的虚线框指示），就在窗格 1004 中加亮一相应的特性代码段，从而允许最终用户修改该特性的值。在一个实施例中，向最终用户提供实施如图 10A 所示的本发明的方面的用户界面（UI）。附录 A 完整地描述了图 10A 所示的 XML 模式。

图 10B 是示出根据本发明的一个实施例的用于使用单一元模式来修改多个 EDI 模式的方法的流程图。在 1006 处，通过对多个 EDI 模式中的数据进行解码来标识表示这多个 EDI 模式的单一结构。在一个示例中，诸如图 11D 中的数据结构 1128 等单一结构通过捕捉以下数据中的一个或多个来表示多个 EDI 模式：

1. 每一 EDI 模式包括具有名称的根元素；
2. 根元素包括可以或者是 Loop（循环）或者是 Segment（段）的重复数据块；
3. 每一 Loop 具有以下结构
 - a. Name—循环的名称
 - b. Block—数据元素集合
 - c. MinOccurs—最小出现次数
 - d. MaxOccurs—最大出现次数
4. 每一 Segment 具有各种特性
 - a. Name—段的名称
 - b. TagId—段的标签 ID
 - c. MinOccurs—最小出现次数
 - d. MaxOccurs—最大出现次数
 - e. 数据元素列表
5. 每一数据元素包括元素集合，每一元素可以或者是 Composite（复合）元素或者是 Simple（简单）元素
6. 每一 SimpleElement（简单元素）具有各种特性

- a. Name—元素的名称
- b. MinOccurs—最小出现次数
- c. MaxOccurs—最大出现次数
- d. MinLength—最小数据长度
- e. MaxLength—最大数据长度
- f. DataType—数据类型，允许的值是 A、AN、ID、R、N、Date（日期）、Time（时间）—对每一 EDI 数据类型有一个
- g. AllowedValues—允许的值集，仅在元素为类型 ID 时才适用

例如，数据结构 1128 包括第一数据字段 1130，该字段包括与多个 EDI 模式中的每一个的根元素相关联的根数据。该数据结构还包括一个或多个第二数据字段 1132，该字段包括表示多个 EDI 模式中的每一个的一个或多个数据块的数据。一个或多个第二数据字段中的数据是根据第一数据字段 1130 中的根数据来定义的。

在 1008 处，确定要包括在单一结构中的特性。该特性定义了多个 EDI 模式的特征。例如，具有特性值“purchase order”（采购定单）的根元素指示该单一结构的特征对应于诸如图 7A 所示的采购定单模式。使用具有特性值的单一结构，在 1010 处，根据所定义的特征和单一结构为用户定义单一元模式。所定义的元模式对应于多个 EDI 模式。在 1012 处，将所定义的元模式中所确定的特性提供给最终用户，使得最终用户能够如图 10A 所示修改多个 EDI 模式中的每一个的特征。

附录 B 示出了表示具有以下结构的采购定单模式的 XML 格式的示例性单一元模式：

1. PurchaseOrderDetail（采购定单细节）段；
2. 由 LineItem（行项）和 ShippingAddress（发货地址）段构成的 Loop
3. Notes（注解）段

图 12 示出了计算机 130 形式的通用计算设备的一个示例。在本发明的一个实施例中，诸如计算机 130 等计算机适用于此处所示并描述的其它附图。计算机 130 具有一个或多个处理器或处理单元 132 以及系统存储器 134。在所示的实施例中，系统总线 136 将包括系统存储器 134 的各种系统组件耦合到处理器 132。总线 136 表示任意几种总线结构中的一种或多种，包括存储器总线或存储器控制器、外围总线、加速图形端口、以及使用各种总线体系结构中的任一种的处理器或局部总线。作为示例而非局限，这类体系结构包括工业标准体系结构（ISA）总线、微通道体系结构（MCA）总线、增强型 ISA（EISA）总线、视频电子技术标准协会（VESA）

局部总线以及外围部件互连（PCI）总线（也称为小背板（Mezzanine）总线）。

计算机 130 通常具有至少某种形式的计算机可读介质。计算机可读介质可包括易失性和非易失性、可移动和不可移动介质，并且可以是可由计算机 130 访问的任何可用介质。作为示例而非局限，计算机可读介质包括计算机存储介质和通信介质。计算机存储介质包括以用于储存诸如计算机可读指令、数据结构、程序模块或其它数据等信息的任一方法或技术实现的易失性和非易失性，可移动和不可移动介质。例如，计算机存储介质包括 RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘（DVD）或其它光盘存储、磁盒、磁带、磁盘存储或其它磁存储设备、或可以用来储存所期望的信息并可由计算机 130 访问的任一其它介质。通信介质通常以诸如载波或其它传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其它数据，并包括任一信息传送介质。本领域的技术人员熟悉已调制数据信号，它以对信号中的信息进行编码的方式设置或改变其一个或多个特征。诸如有线网络或直接连线连接等有线介质，以及诸如声学、RF、红外和其它无线介质等无线介质都是通信介质的示例。上述任一的组合也应当包括在计算机可读介质的范围之内。

系统存储器 134 包括可移动和/或不可移动、易失性和/或非易失性存储器形式的计算机存储介质。在所示的实施例中，系统存储器 134 包括只读存储器（ROM）138 和随机存取存储器（RAM）140。基本输入/输出系统 142（BIOS）包括如在启动时帮助在计算机 130 内的元件之间传输信息的基本例程，它通常储存在 ROM 138 中。RAM 140 通常包含处理单元 132 立即可访问和/或当前正在操作的数据和/或程序模块。作为示例而非局限，图 12 示出了操作系统 144、应用程序 146、其它程序模块 148 和程序数据 150。

计算机 130 也可包括其它可移动/不可移动、易失性/非易失性计算机存储介质。例如，图 12 示出了对不可移动、非易失性磁介质进行读写的硬盘驱动器 154。图 12 还示出了对可移动、非易失性磁盘 158 进行读写的磁盘驱动器 156，以及对可移动、非易失性光盘 162，如 CD ROM 或其它光介质等可移动、非易失性光盘 162 进行读写的光盘驱动器 160。可以在示例性操作环境中使用的其它可移动/不可移动、易失性/非易失性计算机存储介质包括但不限于，磁带盒、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等等。硬盘驱动器 154 以及磁盘驱动器 156 和光盘驱动器 160 通常通过非易失性存储器接口，如接口 166 连接到系统总线 136。

上文讨论并在图 12 中示出的驱动器或其它大容量存储设备及其关联的计算机存储介质为计算机 130 提供了计算机可读指令、数据结构、程序模块和其它数据的存储。例如，在图 12 中，示出硬盘驱动器 154 储存操作系统 170、应用程序 172、其它程序模块 174 和程序数据 176。注意，这些组件可以与操作系统 144、应用程序 146、其它程序模块 148 和程序数据 150 相同，也可以与它们不同。这里对操作系统 170、应用程序 172、其它程序模块 174 和程序数据 176 给予不同的标号来说明至少它们是不同的副本。

用户可以通过输入设备或用户界面选择设备，如键盘 180 和定位设备 182（通常指鼠标、跟踪球或触摸垫）向计算机 130 输入命令和信息。其它输入设备（未示出）可包括话筒、操纵杆、游戏手柄、圆盘式卫星天线、扫描仪等等。这些和其它输入设备通常通过耦合至系统总线 136 的用户输入接口 184 连接至处理单元 132，但是也可以通过其它接口和总线结构连接，如并行端口、游戏端口或通用串行总线（USB）。监视器 188 或其它类型的显示设备也通过接口，如视频接口 190 连接至系统总线 136。除监视器 188 之外，计算机通常包括其它外围输出设备（未示出），如扬声器和打印机，它们可通过输出外围接口（未示出）连接。

计算机 130 可以使用到一个或多个远程计算机，如远程计算机 194 的逻辑连接在网络化环境中操作。远程计算机 194 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它常见的网络节点，并通常包括许多或所有相对于计算机 130 所描述的许多或所有元件。图 12 描述的逻辑连接包括局域网（LAN）196 和广域网（WAN）198，但也可包括其它网络。LAN 136 和/或 WAN 138 可以是有线网络、无线网络或其组合等等。这类网络环境常见于办公室、企业范围计算机网络、内联网以及全球计算机网络（例如，因特网）。

当在局域网环境中使用时，计算机 130 通过网络接口或适配器 186 连接至 LAN 196。当在广域网环境中使用时，计算机 130 通常包括调制解调器 178 或用于通过 WAN 198，如因特网建立通信的其它装置。调制解调器 178 可以是内置或外置的，它通过用户输入接口 184 或其它适当的机制连接至系统总线 136。在网络化环境中，相对于计算机 130 所描述的程序模块或其部分可储存在远程存储器存储设备（未示出）中。作为示例而非局限，图 12 示出远程应用程序 192 驻留在存储器设备上。示出的网络连接是示例性的，也可以使用在计算机之间建立通信链路的其它手段。

一般而言，计算机 130 的数据处理器借助在不同时刻储存在计算机的各种计算机可读存储介质中的指令来编程。程序和操作系统通常例如在软盘或 CD-ROM

上分发。从这些盘上，它们被安装或加载到计算机的次级存储器中。在执行时，它们被至少部分地加载到计算机的主电子存储器中。当这些和其它各种类型的计算机可读存储介质包括用于实现在下文中结合微处理器或其它数据处理器描述的步骤的指令或程序时，此处所描述的本发明的各方面包括这种介质。此外，当依照此处所描述的方法和技术来编程时，本发明的各方面包括计算机本身。

出于图示的目的，诸如操作系统等程序和其它可执行程序组件此处被示为离散的框。然而，可以认识到，这些程序和组件在各种时刻驻留在计算机的不同存储组件中，并且由计算机的数据处理器来执行。

尽管结合包括计算机 130 的示例性计算系统环境描述，但本发明的各实施例可以使用众多其它通用或专用计算系统环境或配置来操作。该计算系统环境并不旨在对本发明的任何方面的使用范围或功能提出任何局限。此外，也不应将该计算系统环境解释为对该示例性操作环境中示出的任一组件或其组合具有任何依赖或需求。

适用于本发明的各方面的众所周知的计算系统、环境和/或配置的示例包括但不限于，个人计算机、服务器计算机、手持式或膝上型设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子产品、移动电话、网络 PC、小型机、大型计算机机、包括任一上述系统或设备的分布式计算环境等等。

本发明可在诸如程序模块等由一个或多个计算机或其它设备执行的计算机可执行指令的一般上下文环境中描述。一般而言，程序模块包括但不限于，执行特定的任务或实现特定的抽象数据类型的例程、程序、对象、组件、数据结构等等。本发明的各方面也可以在其中任务由通过通信网络链接的远程处理设备来执行的分布式计算环境中实践。在分布式计算环境中，程序模块可以位于包括存储器存储设备的本地和远程计算机存储介质中。

接口可以是诸如 Java 2 平台企业版本 (J2EE)、COM 或分布式 COM (DCOM) 示例中的紧耦合的同步实现。作为替代或除此之外，接口可以是诸如 web 服务（例如，使用简单对象访问协议）中的松耦合的异步实现。一般而言，接口包括以下特征的任一组合：紧耦合的、松耦合的、同步和异步。此外，接口可符合标准协议、专有协议、或标准和专有协议的任何组合。

此处所描述的接口可以都是单个接口的一部分，或者可以被实现为单独的接口或其中的任何组合。接口可以本地或远程地执行以提供功能。此外，接口可包括比此处所示或所描述的更多或更少的功能。

在操作中，计算机 130 执行诸如在图中所示的计算机可执行指令来实现本发明的各方面。

此处所示且描述的本发明的各实施例中的操作的执行或实现顺序不是必要的，除非另外指定。即，操作可以用任何顺序来执行，除非另外指定，且本发明的各实施例可包括比此处所公开的更多或更少的操作。例如，可以构想，在另一操作之前、与此同时或之后执行或实现特定操作是在本发明的各方面的范围之内。

本发明的各实施例可以用计算机可执行指令来实现。计算机可执行指令可被组织成一个或多个计算机可执行组件或模块。本发明的各方面可以用任意数量或组织的这些组件或模块来实现。例如，本发明的各方面不限于在图中所示并在此处所描述的具体计算机可执行指令或具体组件或模块。本发明的其它实施例可包括具有比此处所示并描述的更多或更少功能的不同的计算机可执行指令或组件。

当介绍本发明或其实施例的各方面的元素时，冠词“一”、“一个”、“该”和“所述”意指存在一个或多个元素。术语“包括”、“包含”和“具有”旨在是包含性的，且意味着除所列出的元素之外还可以有其它元素。

由于可在以上构造、产品和方法中作出各种改变而不背离本发明的各方面的范围，因此包含在以上描述中并在附图中所示的所有主题都旨在被解释为说明性的而非限制的意义。

附录 A

节 1: 以 XML 格式表示 EDI 模式的元模式:

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:b="http://schemas.company.com/BizApp/2003"
xmlns="http://schema.company.com/EdiClient/MetaSCHEMA"
targetNamespace="http://schema.company.com/EdiClient/MetaSCHEMA"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="EdiSchemaRoot">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RootElementName" type="xs:string" />
        <xs:element ref="Block" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Block" type="BlockType" />
  <xs:element name="Segment">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="TagId" type="xs:string" />
        <xs:element name="MinOccurs" type="xs:integer" />
        <xs:element name="MaxOccurs" type="xs:integer" />
        <xs:element name="DataElementList">
          <xs:complexType>
            <xs:sequence>
              <xs:choice minOccurs="1" maxOccurs="unbounded">
                <xs:element name="CompositeElement">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Name" type="xs:string" />
```

```

        <xs:element maxOccurs="unbounded"
ref="SimpleElement" />
    </xs:sequence>
</xs:complexType>
</xs:element>
    <xs:element ref="SimpleElement" />
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SimpleElement">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Name" type="xs:string" />
            <xs:element name="MinOccurs" type="xs:string" />
            <xs:element name="MaxOccurs" type="xs:string" />
            <xs:element name="MinLength" type="xs:string" />
            <xs:element name="MaxLength" type="xs:string" />
            <xs:element name="DataType">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="A" />
                        <xs:enumeration value="N" />
                        <xs:enumeration value="ID" />
                        <xs:enumeration value="R" />
                        <xs:enumeration value="AN" />
                        <xs:enumeration value="Date" />
                        <xs:enumeration value="Time" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```
        </xs:simpleType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="AllowedValues" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="BlockType">
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="Loop">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" type="xs:string" />
            <xs:element ref="Block" />
            <xs:element name="MinOccurs" type="xs:int" />
            <xs:element name="MaxOccurs" type="xs:int" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="Segment" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

附录 B

节 2: 使用元模式单一结构的示例采购订单模式:

```
<ns0:EdiSchemaRoot
xmlns:ns0="http://schema.company.com/EdiClient/MetaSCHEMA">
  <RootElementName>X12_00501_850</RootElementName>
  <Block>
    <Segment>
      <Name>PurchaseOrderDetail</Name>
      <TagId>PUR</TagId>
      <MinOccurs>1</MinOccurs>
      <MaxOccurs>1</MaxOccurs>
      <DataElementList>
        <SimpleElement>
          <Name>OriginatorId</Name>
          <MinOccurs>1</MinOccurs>
          <MaxOccurs>1</MaxOccurs>
          <MinLength>4</MinLength>
          <MaxLength>10</MaxLength>
          <DataType>AN</DataType>
        </SimpleElement>
        <SimpleElement>
          <Name>FirstName</Name>
          <MinOccurs>1</MinOccurs>
          <MaxOccurs>1</MaxOccurs>
          <MinLength>1</MinLength>
          <MaxLength>10</MaxLength>
          <DataType>AN</DataType>
        </SimpleElement>
        <SimpleElement>
          <Name>LastName</Name>
          <MinOccurs>1</MinOccurs>
```

```

        <MaxOccurs>1</MaxOccurs>
        <MinLength>1</MinLength>
        <MaxLength>10</MaxLength>
        <DataType>AN</DataType>
    </SimpleElement>
</DataElementList>
</Segment>
<Loop>
    <Name>LineItemLoop</Name>
    <MinOccurs>1</MinOccurs>
    <MaxOccurs>unbounded</MaxOccurs>
    <Block>
        <Segment>
            <Name>LineItem</Name>
            <TagId>LIN</TagId>
            <MinOccurs>1</MinOccurs>
            <MaxOccurs>1</MaxOccurs>
            <DataElementList>
                <SimpleElement>
                    <Name>ItemId</Name>
                    <MinOccurs>1</MinOccurs>
                    <MaxOccurs>1</MaxOccurs>
                    <MinLength>4</MinLength>
                    <MaxLength>10</MaxLength>
                    <DataType>AN</DataType>
                </SimpleElement>
                <SimpleElement>
                    <Name>Quantity</Name>
                    <MinOccurs>1</MinOccurs>
                    <MaxOccurs>1</MaxOccurs>
                    <MinLength>1</MinLength>
                    <MaxLength>5</MaxLength>

```

```
<DataType>N</DataType>
</SimpleElement>
</DataElementList>
</Segment>
<Segment>
  <Name>ShipTo</Name>
  <TagId>SHP</TagId>
  <MinOccurs>1</MinOccurs>
  <MaxOccurs>1</MaxOccurs>
  <DataElementList>
    <SimpleElement>
      <Name>FirstName</Name>
      <MinOccurs>1</MinOccurs>
      <MaxOccurs>1</MaxOccurs>
      <MinLength>1</MinLength>
      <MaxLength>10</MaxLength>
      <DataType>AN</DataType>
    </SimpleElement>
    <SimpleElement>
      <Name>LastName</Name>
      <MinOccurs>1</MinOccurs>
      <MaxOccurs>1</MaxOccurs>
      <MinLength>1</MinLength>
      <MaxLength>10</MaxLength>
      <DataType>AN</DataType>
    </SimpleElement>
    <CompositeElement>
      <Name>Address</Name>
      <SimpleElement>
        <Name>StreetInfo</Name>
        <MinOccurs>1</MinOccurs>
        <MaxOccurs>1</MaxOccurs>
```

```

        <MinLength>1</MinLength>
    >
        <MaxLength>100</MaxLeng
th>
        <DataType>AN</DataType>
</SimpleElement>
<SimpleElement>
    <Name>City</Name>
<MinOccurs>1</MinOccurs>
    <MaxOccurs>1</MaxOccurs>
        <MinLength>1</MinLength>
    >
        <MaxLength>100</MaxLeng
th>
        <DataType>AN</DataType>
</SimpleElement>
<SimpleElement>
    <Name>State</Name>
<MinOccurs>1</MinOccurs>
    <MaxOccurs>1</MaxOccurs>
        <MinLength>2</MinLength>
    >
        <MaxLength>2</MaxLength>
    >
        <DataType>ID</DataType>
</SimpleElement>
<SimpleElement>
    <Name>Zip</Name>
<MinOccurs>1</MinOccurs>
    <MaxOccurs>1</MaxOccurs>
        <MinLength>5</MinLength>
    >

```

```

                                <MaxLength>10</MaxLengt
                                h>
                                <DataType>N</DataType>
                                </SimpleElement>
                                </CompositeElement>
                                </DataElementList>
</Segment>
</Block>
</Loop>
<Segment>
    <Name>Notes</Name>
    <TagId>NTE</TagId>
    <MinOccurs>0</MinOccurs>
    <MaxOccurs>1</MaxOccurs>
    <DataElementList>
        <SimpleElement>
            <Name>NoteLine1</Name>
            <MinOccurs>0</MinOccurs>
            <MaxOccurs>1</MaxOccurs>
            <MinLength>1</MinLength>
            <MaxLength>80</MaxLength>
            <DataType>AN</DataType>
        </SimpleElement>
        <SimpleElement>
            <Name>NoteLine2</Name>
            <MinOccurs>0</MinOccurs>
            <MaxOccurs>1</MaxOccurs>
            <MinLength>1</MinLength>
            <MaxLength>80</MaxLength>
            <DataType>AN</DataType>
        </SimpleElement>
        <SimpleElement>

```

```
<Name>NoteLine3</Name>
<MinOccurs>0</MinOccurs>
  <MaxOccurs>1</MaxOccurs>
    <MinLength>1</MinLength>
      <MaxLength>80</MaxLength>
        <DataType>AN</DataType>
      </SimpleElement>
    </DataElementList>
  </Segment>
</Block>
</ns0:EdiSchemaRoot>
```

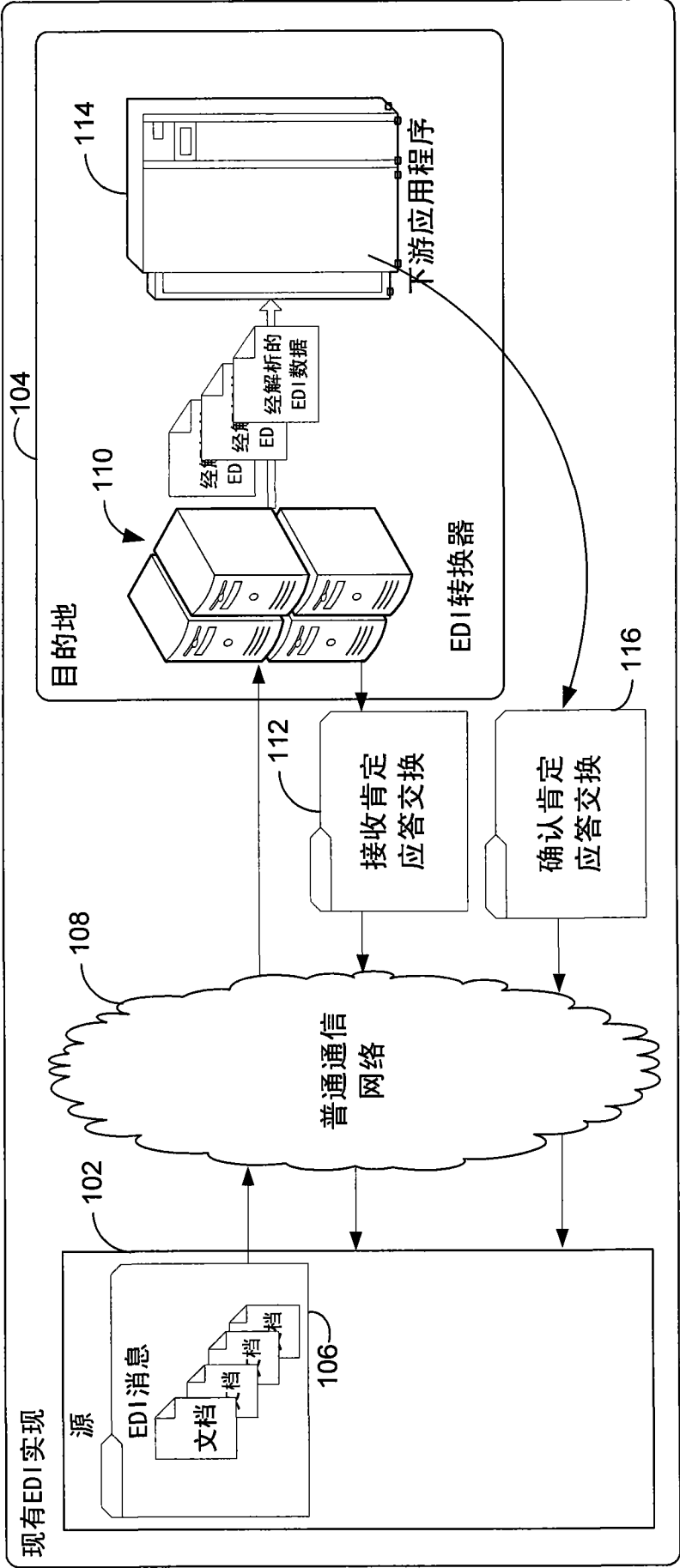


图 1

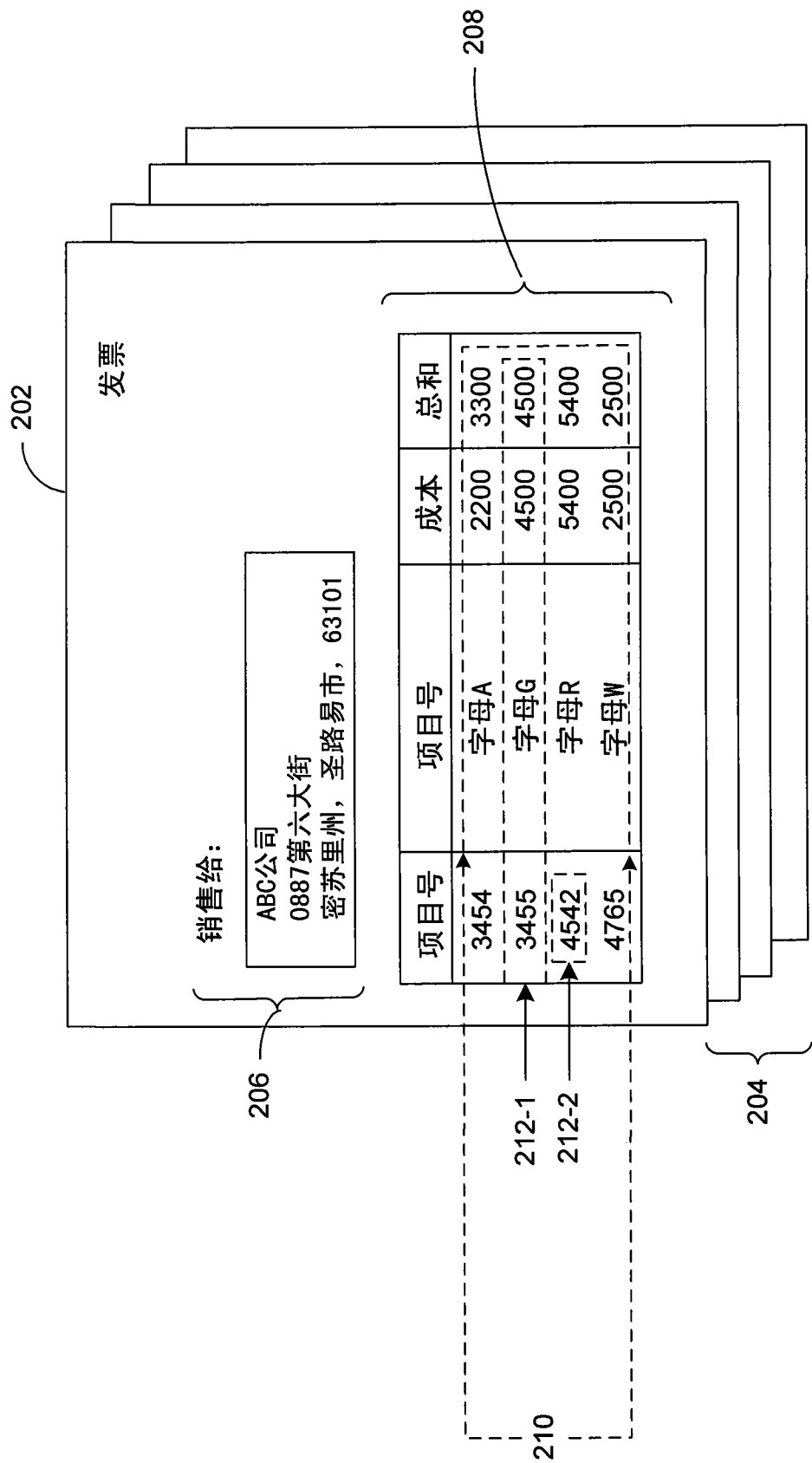


图 2A

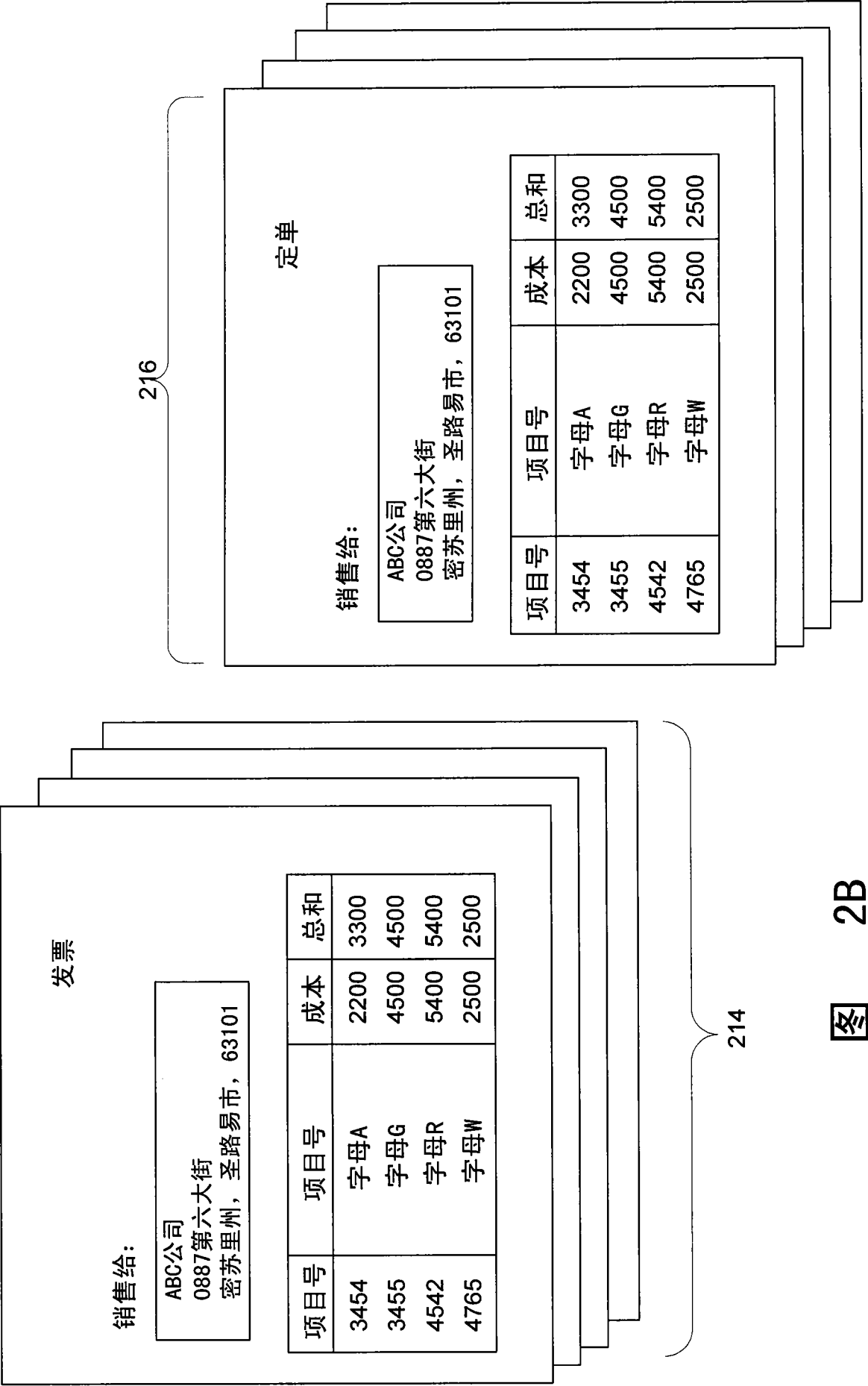


图 2B

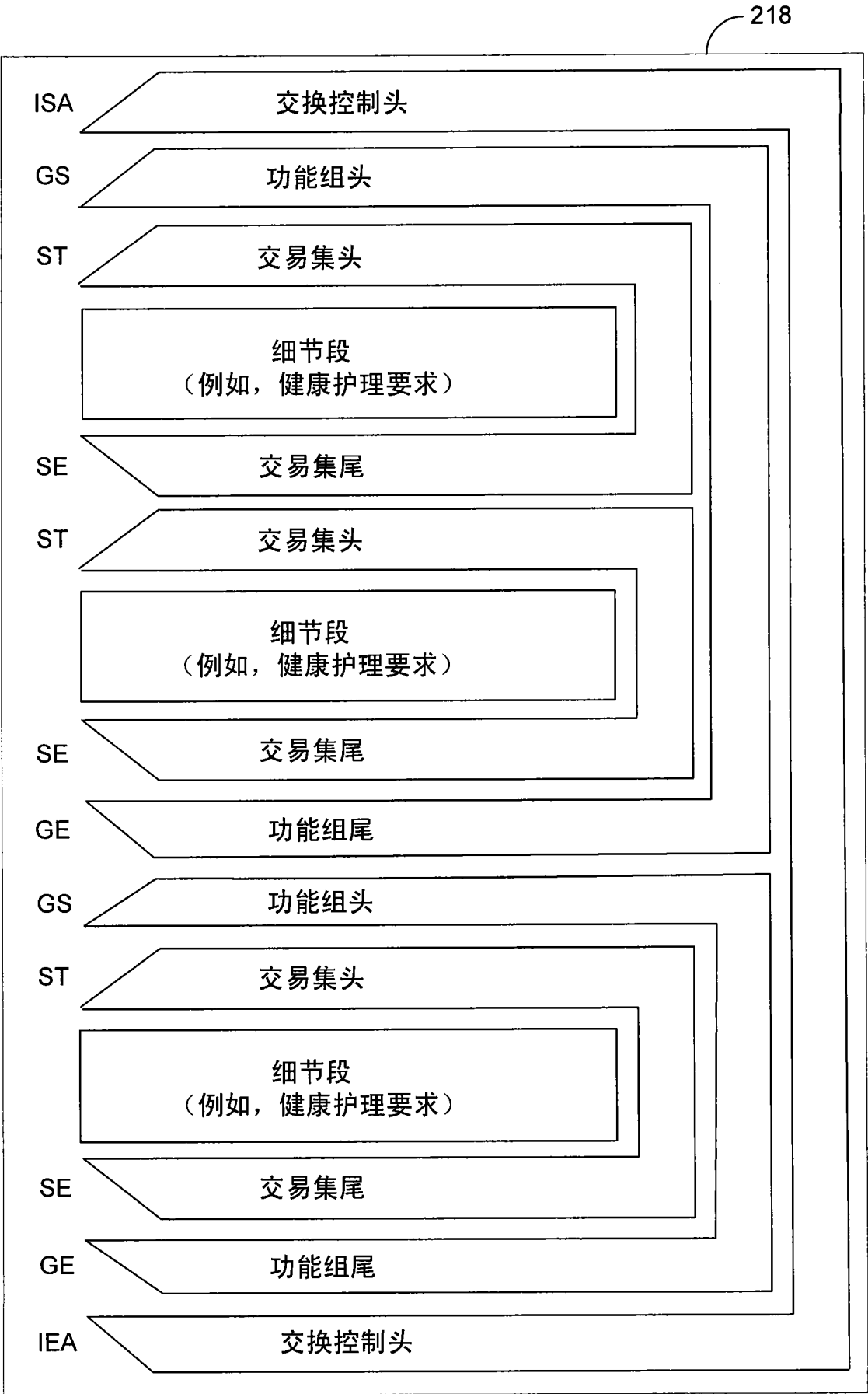


图 20

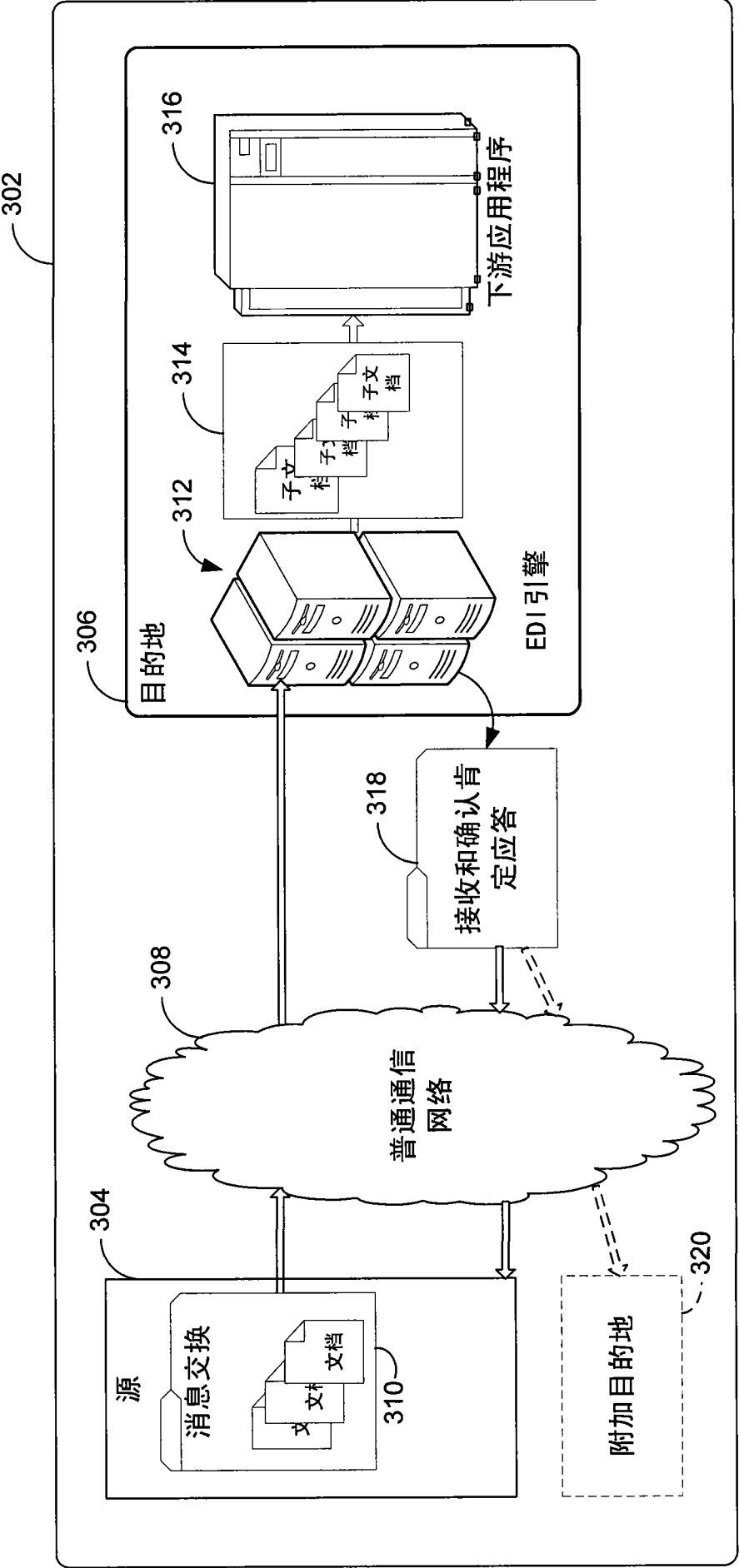


图 3

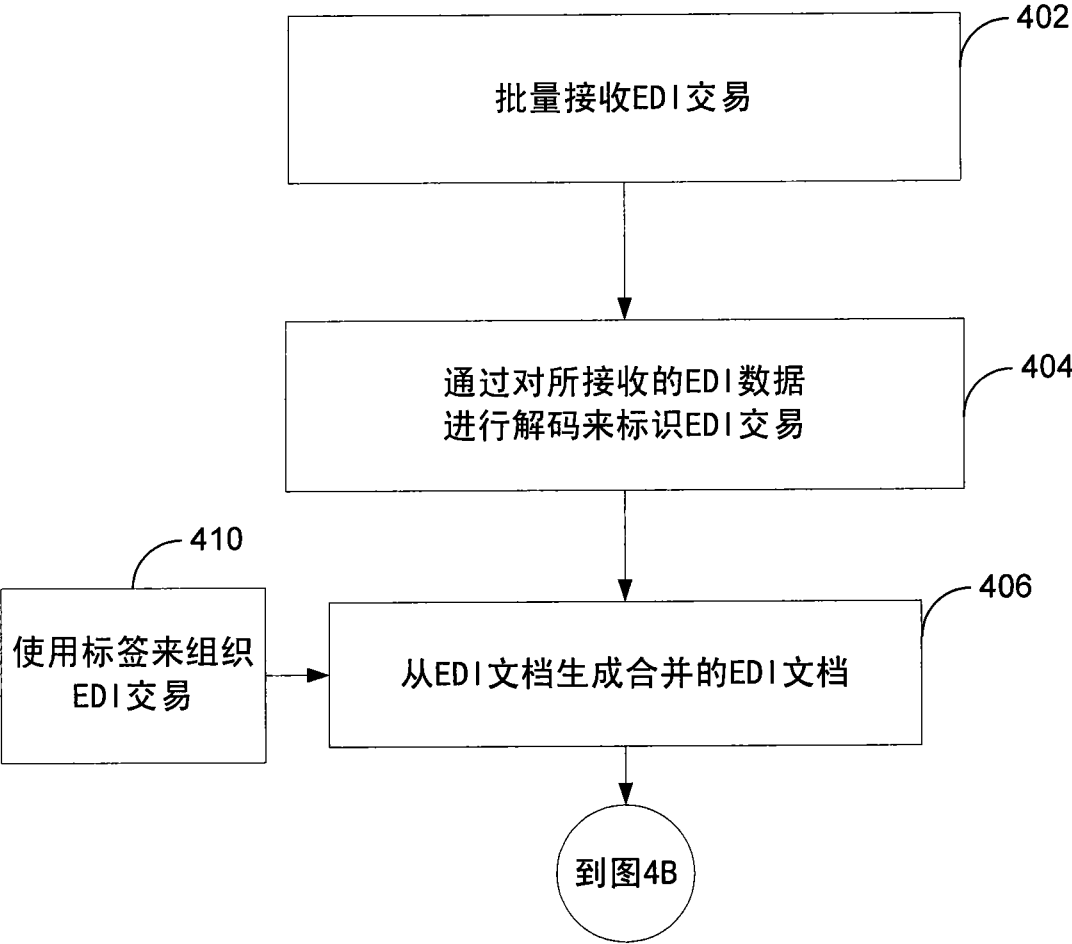


图 4A

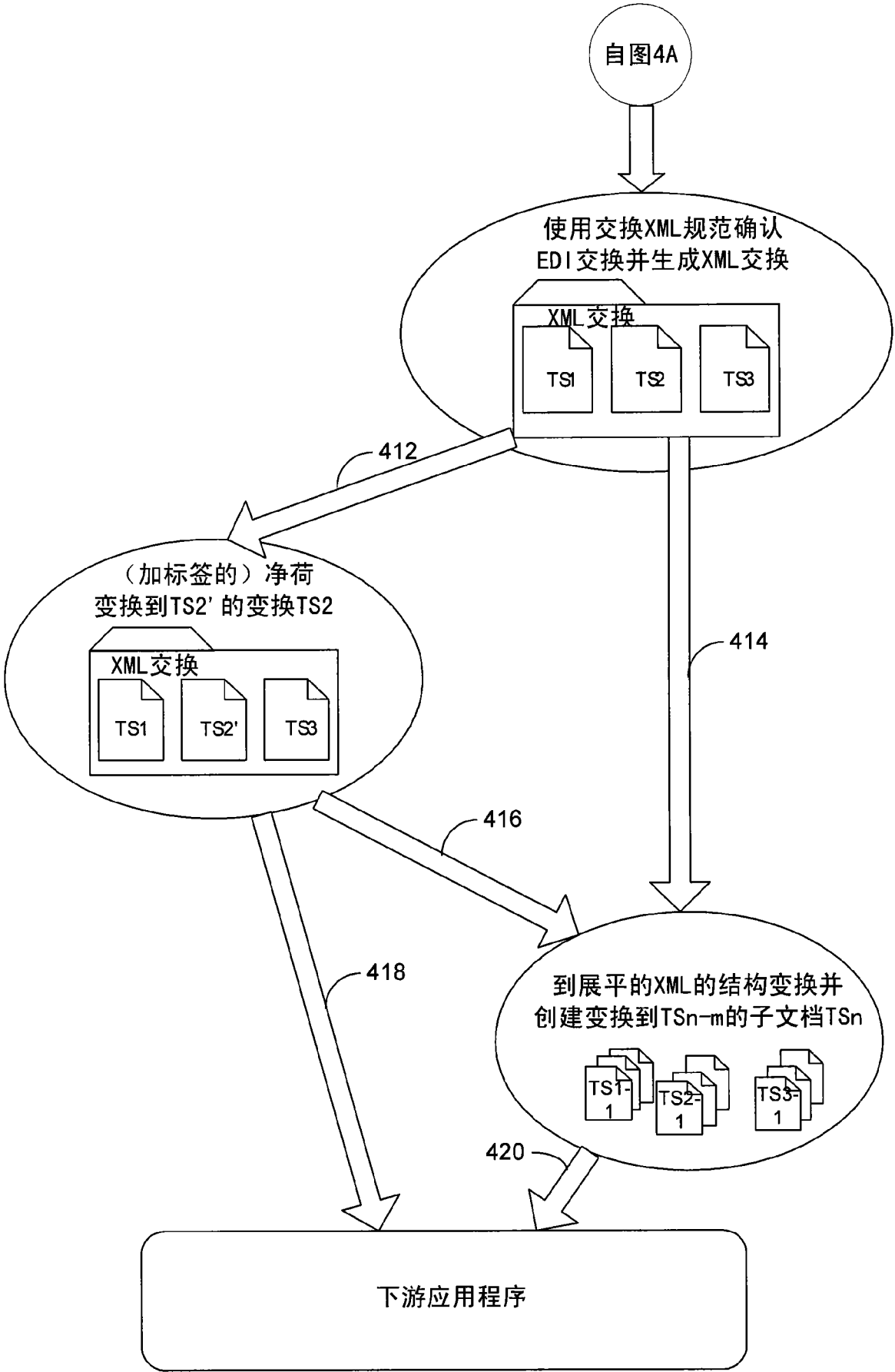


图 4B

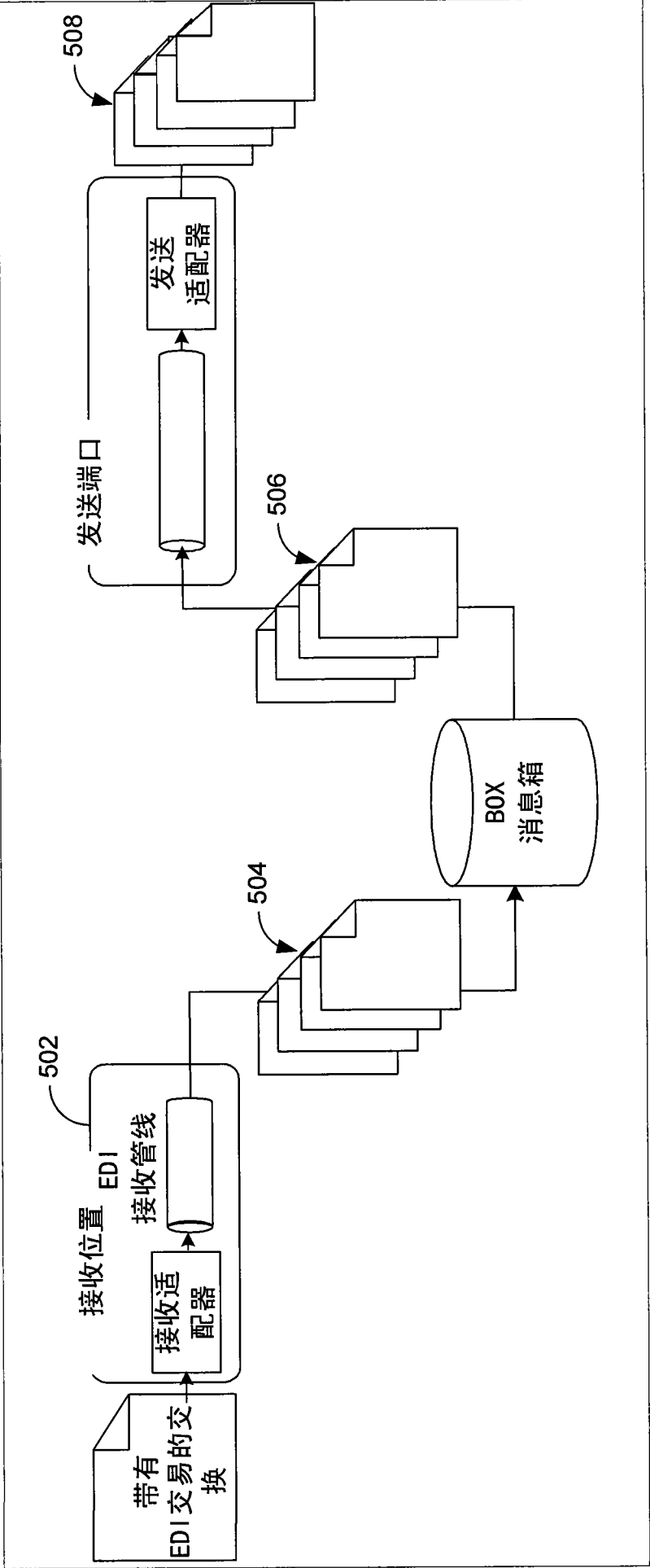


图 5A

510

```

- <ns0:X12_00401_850 xmlns:ns0="http://schemas.microsoft.com/
BizTalk/EDI/X12/2006">
- <ns1:ST xmlns:ns1="http://schemas.microsoft.com/BizTalk/EDI/X12/
2006">
  <ST01>850</ST01>
  <ST02>0001</ST02>
  </ns1:ST>
- <ns2:BEG xmlns:ns2="http://schemas.microsoft.com/BizTalk/EDI/
X12/2006">
  <BEG01>00</BEG01>
  <BEG02>NE</BEG02>
  <BEG03>3859199838</BEG03>
  <BEG05>950727</BEG05>
  <BEG08>IBM</BEG08>
  </ns2:BEG>
- <ns3:PER xmlns:ns3="http://schemas.microsoft.com/BizTalk/EDI/
X12/2006">
  <PER01>AA</PER01>
  <PER02>Andrew Carlson</PER02>
  <PER03>TE</PER03>
  <PER04>(708) 555-2930</PER04>
  </ns3:PER>
- <ns4:DTM xmlns:ns4="http://schemas.microsoft.com/BizTalk/EDI/
X12/2006">
  <DTM01>002</DTM01>
  <DTM02>970523</DTM02>
  </ns4:DTM>
- <ns5:PO1Loop1 xmlns:ns5="http://schemas.microsoft.com/BizTalk/
EDI/X12/2006">
- <ns6:PO1 xmlns:ns6="http://schemas.microsoft.com/BizTalk/EDI/
X12/2006">
  <PO101>1</PO101>
  <PO102>93</PO102>
  <PO103>BX</PO103>
  <PO104>145.39</PO104>
  <PO106>CB</PO106>
  <PO107>KOW-20392-10</PO107>
  </ns6:PO1>
- <ns7:PIDLoop1 xmlns:ns7="http://schemas.microsoft.com/BizTalk/
EDI/X12/2006">
- <ns8:PID_2 xmlns:ns8="http://schemas.microsoft.com/BizTalk/EDI/
X12/2006">
  <PID01>F</PID01>
  <PID05>Uninterruptible Power System</PID05>
  </ns8:PID_2>
  </ns7:PIDLoop1>
- <ns9:PER_3 xmlns:ns9="http://schemas.microsoft.com/BizTalk/EDI/
X12/2006">
  <PER01>AA</PER01>
  <PER02>Camilla Anderson</PER02>
  <PER03>TE</PER03>
  <PER04>(708) 555-2011</PER04>
  </ns9:PER_3>
  </ns5:PO1Loop1>
- <ns10:PO1Loop1 xmlns:ns10="http://schemas.microsoft.com/
BizTalk/EDI/X12/2006">
- <ns11:PO1 xmlns:ns11="http://schemas.microsoft.com/BizTalk/EDI/
X12/2006">
  <PO101>2</PO101>
  <PO102>25</PO102>
  <PO103>EA</PO103>
  <PO104>35.68</PO104>
  <SE01>14</SE01>
  <SE02>0001</SE02>
  </ns21:SE>
</ns0:X12_00401_850>

```

512



5B

514

ISA+01+aaaaaaaa+03+aaaaaaaa+ZZ+7654321 +ZZ+
7654321+051121+1814+*+00401+000000021+1+T+.'
GS+AE+sender+receiver+051121+1814+9+X+00401'
ST+850+0010'
BEG+00+NE+3859199838++950727+++IBM'
PER+AA+Andrew Carlson+TE+(708) 555-2930'
DTM+002+970523'
PO1+1+93+BX+145.39++CB+KOW-20392-10'
PID+F++++Uninterruptible Power System'
PER+AA+Camilla Anderson+TE+(708) 555-2011'
PO1+2+25+EA+35.68++CB+1093-4927-001'
PID+F++++High Volume Printer Stand'
PER+AA+Miranda Cappelán+TE+(708) 555-1111'
PO1+3+4+PC+2002.91++CB+ABX-2001'
PID+F++++Electronics Cabinet Package (56" High)'
CTT+3'
SE+14+0010'
GE+1+9'
IEA+1+000000021'

图 5C

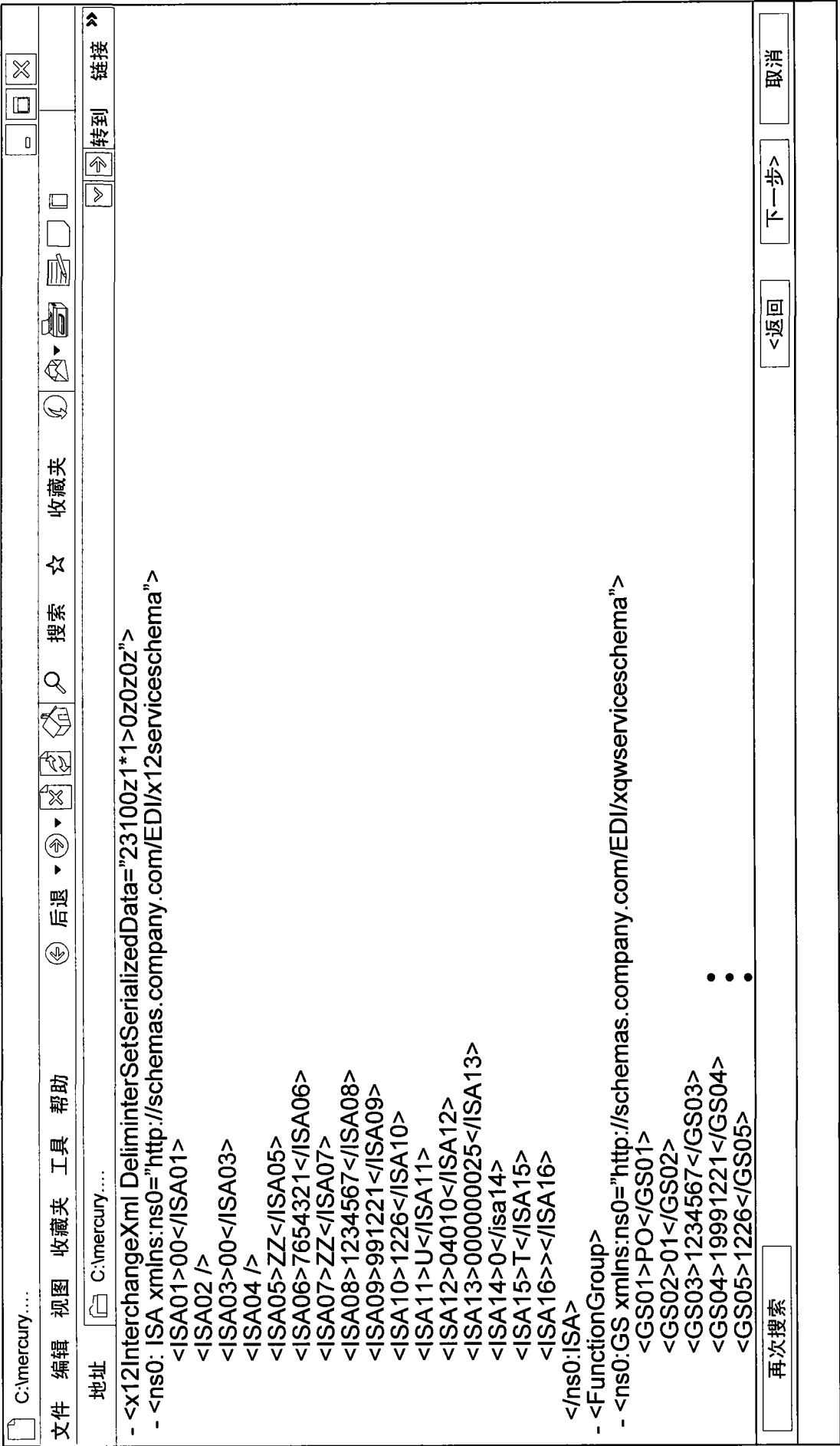


图 6A

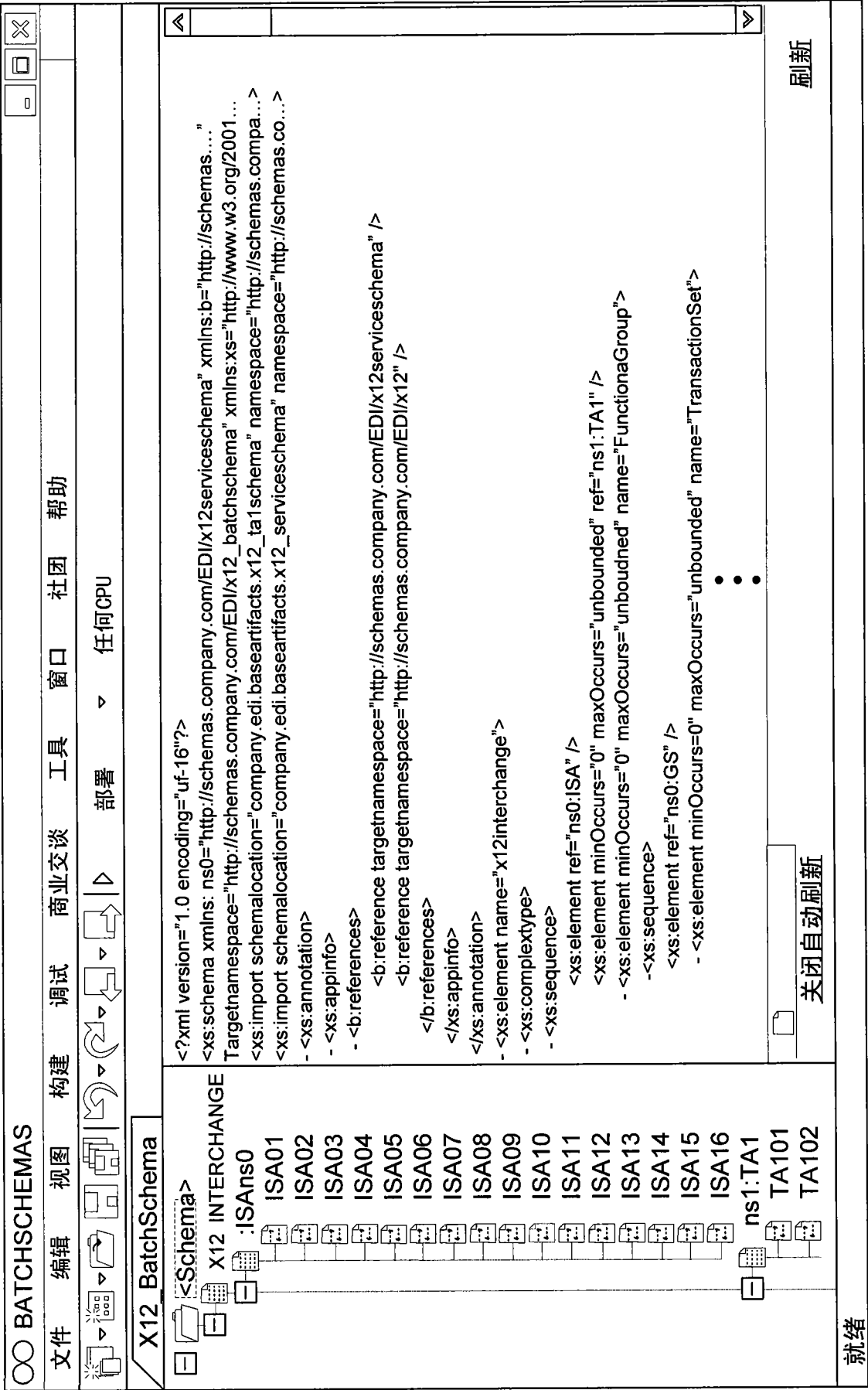


图 6B

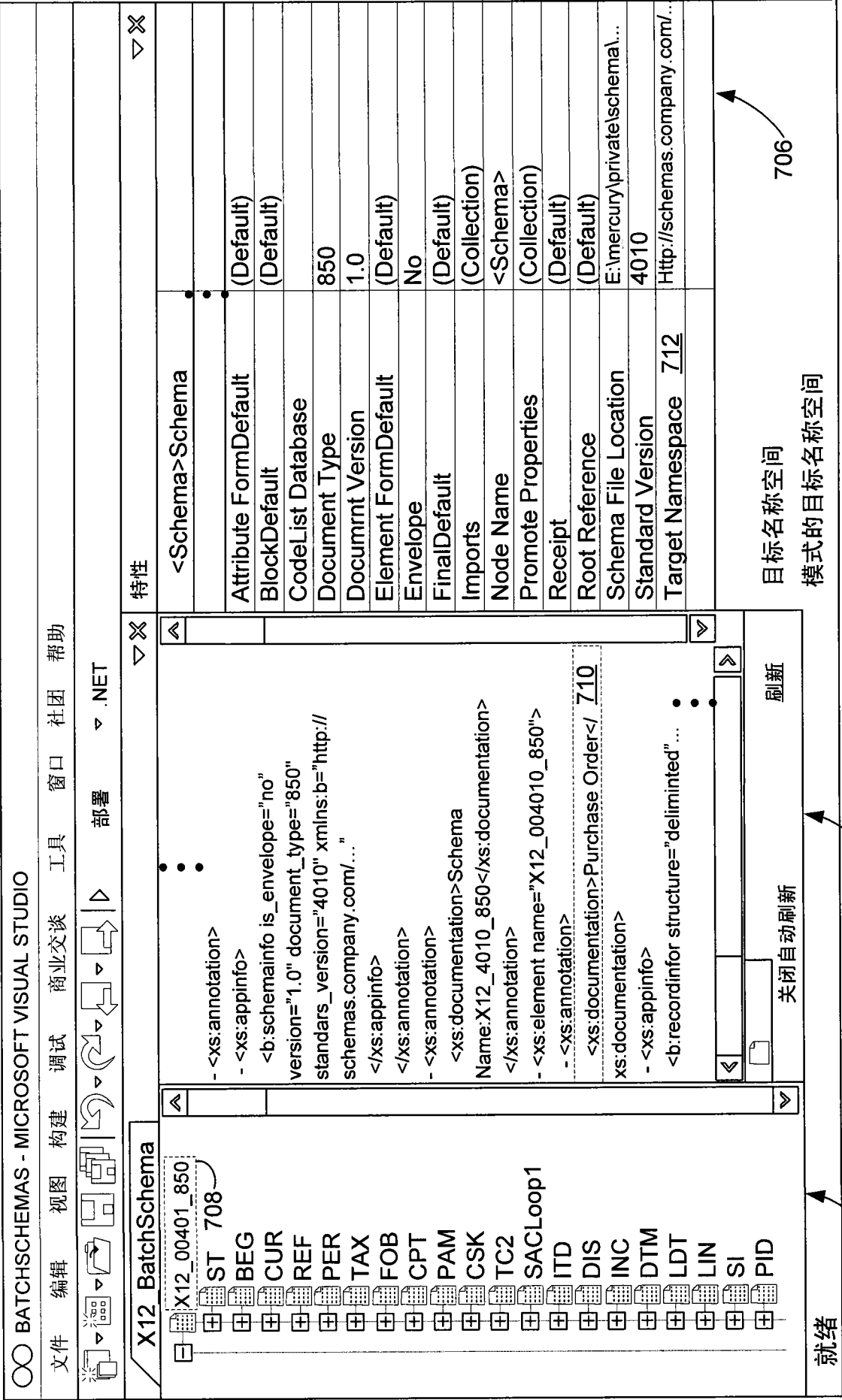


图 7A

×

合伙人特性

合伙人协定
发送者角色
X12特性
EDIFACT特性
肯定应答、确认和批处理
接收者角色
X12特性
ISA段
GS/ST段
EDIFACT特性
UNA段
UNB段
UNG/UNH段
肯定应答和确认设置
批处理设置

交换控制头
授权:
安全:
发送者标识符
☐ 使用ISA11作为重复分隔符
☐ 检查重复的控制号, ISA13 [天]
☐ 组控制头
☐ 检查交换中的重复的组号 [GS6]

默认	发送者 ID	交易 ID	目标名称...	版本
☒ 714	☐	410	http://schemas...	004
☐	☐	650	http://schemas...	004
*	☐	☐		
⏪				⏩

交易集头
☐ 检查交换中的重复的组号 [GS6]
实现约定 [ST03]:

⏪

图 7B

合伙人特性

合伙人协定
发送者角色
X12特性
EDIFACT特性
肯定应答、确认和批处理
接收者角色
X12特性
ISA段
GS/ST段
EDIFACT特性
UNA段
UNB段
UNG/UNH段
肯定应答和确认设置
批处理设置

交换头

检查重复的控制引用 [UNB5]
有效性天数
接收方引用口令 [UNB6]:
值: 限定词:
功能组 [UNG]
☐ 检查交换中的重复的组号 [GS6]

	默认	UNG 2.1	UNG 2.2	UNG8	交易	目标名称...
▷	☒	Hws	CC	d98b	☐	http://schemas...
	☐	Sft	CC	d94a	☐	http://schemas...
*	☐				☐	

716 718 720 722

交易集头[UNH]
☐ 检查组中重复的交易集控制号

图 7C

合伙人特性

公共

发送者角色

X12特性

接收者角色

X12特性

ISA段

GS/ST段

EDIFACT特性

UNA段

UNB段

UNG/UNH段

关闭

X12特性

交换头 [ISA]

授权: 限定词: 信息: FALLCOLOR

安全: COLOR

☐ 检查重复的控制号,

有效性[天] 30

组控制头

☐ 检查交换中重复的组号

724

版本/发行:

目标名称空间:

消息头

☐ 检查组中重复的交易集控制号

生成肯定应答

☒ 生成 TA1

☒ 生成 997

☐ 编辑交易集控制号

帮助

OK

取消

应用

图 7D

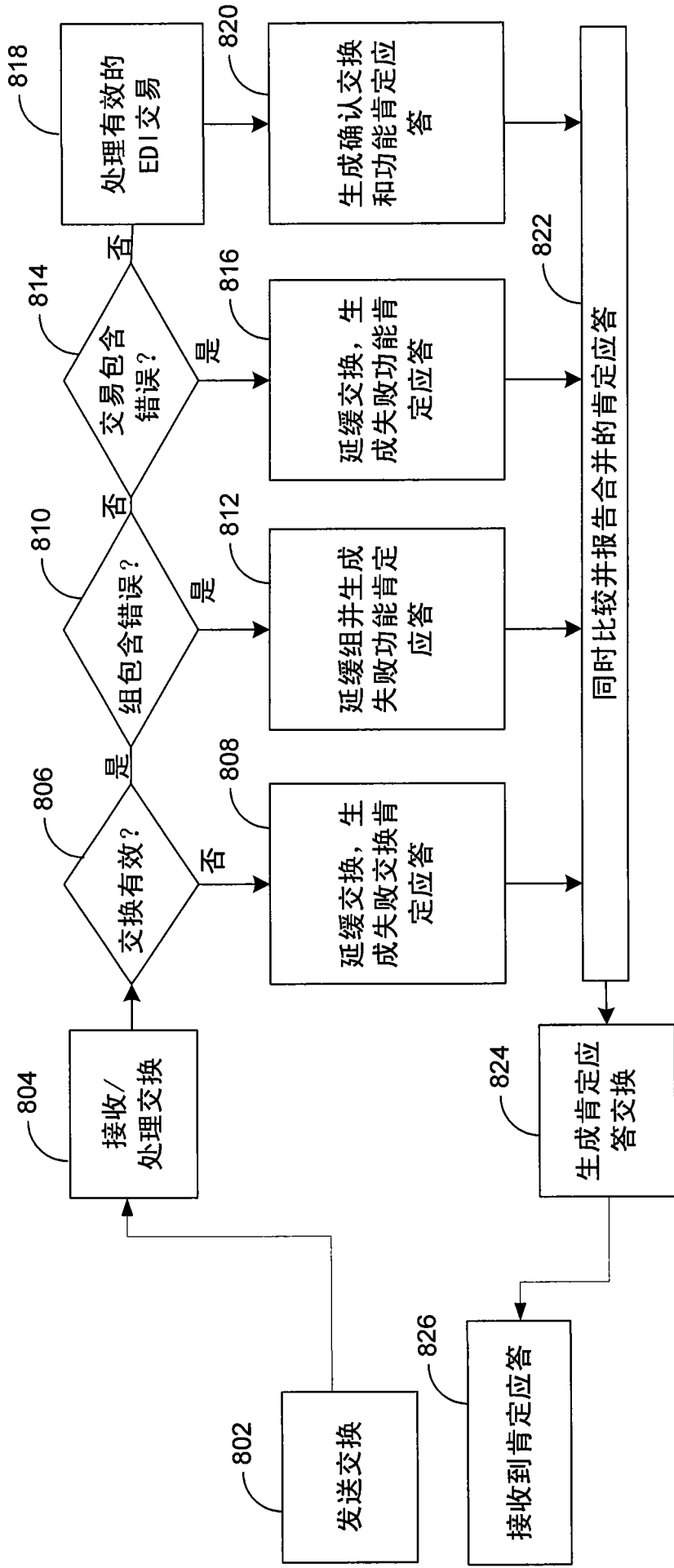


图 8A

ISA+01+aaaaaaaa+03+aaaaaaaa+ZZ+7654321 +ZZ+
 7654321+051121+1814+*+00401+000000021+1+T+.'
 532 → GS+AE+sender+receiver+051121+1814+9+X+00401'
 ST+850+0010' 828
 BEG+00+NE+3859199838++950727+++IBM'
 PER+AA+Andrew Carlson+TE+(708) 555-2930'
 DTM+002+970523'
 PO1+1+93+BX+145.39++CB+KOW-20392-10'
 PID+F++++Uninterruptible Power System'
 PER+AA+Camilla Anderson+TE+(708) 555-2011'
 PO1+2+25+EA+35.68++CB+1093-4927-001'
 PID+F++++High Volume Printer Stand'
 PER+AA+Miranda Cappelán+TE+(708) 555-1111'
 PO1+3+4+PC+2002.91++CB+ABX-2001'
 PID+F++++Electronics Cabinet Package (56" High)'
 SE+14+0010'
 534 → GE+2+ 10' 830
 IEA+1+000000021'

图 8B

位置.#	段. ID	名称	请求目的地.	最大使用	循环重复
010	ST	交易集头	M	1	999999
020	AK1	功能组响应头	M	1	
循环 ID — AK2					
030	AK2	交易集响应头	O	1	999999
循环 ID — AK2/AK3					
040	AK3	数据段注解	0	1	
050	AK4	数据段注解	0	99	
060	AK5	交易集响应尾	M	1	
070	AK9	功能组响应尾	M	1	999999
080	SE	交易集尾	M	1	

图 9A

位置#	位置	名称	S			R		
0010	UNH	消息头	M		1			
0020	UCI	交换响应	M		1			
0030	—	段组1	C		999999			
0040	UCM	消息/包响应	M		1			
0050	—	段组2	C		999			
0060	UCS	段错误指示	M		1			
0070	UCD	数据元素错误指示	C		99			
0080	—	段组3	C		999999			
0090	UCF	组响应	M		1			
0100	—	段组4	C		999999			
0110	UCM	消息/包响应	M		1			
0120	—	段组5	C		999			
0130	UCS	段错误指示	M		1			
0140	UCD	数据元素错误指示	C		99			
0150	UNT	消息尾	M		1			

图 9B

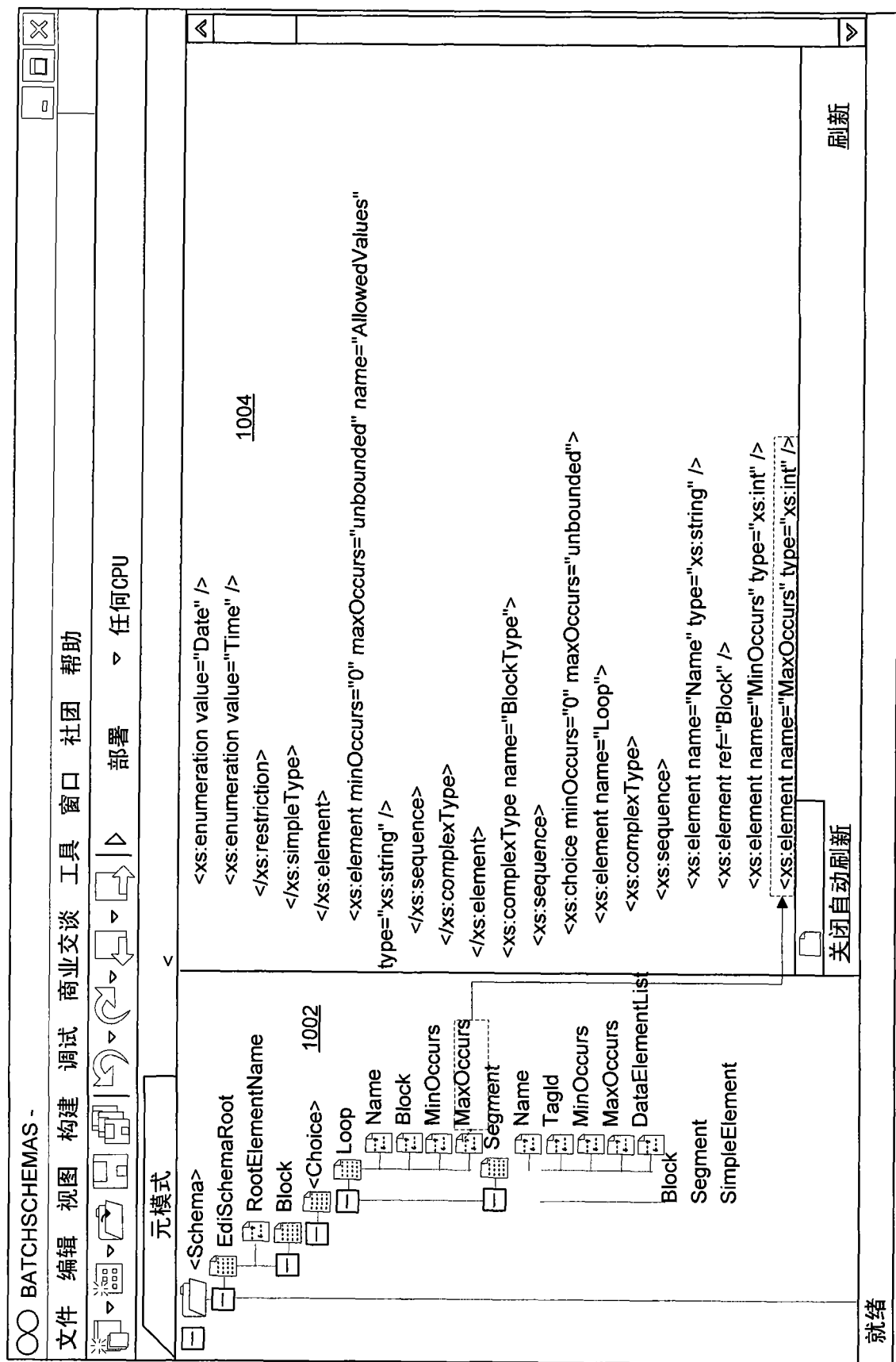


图 10A

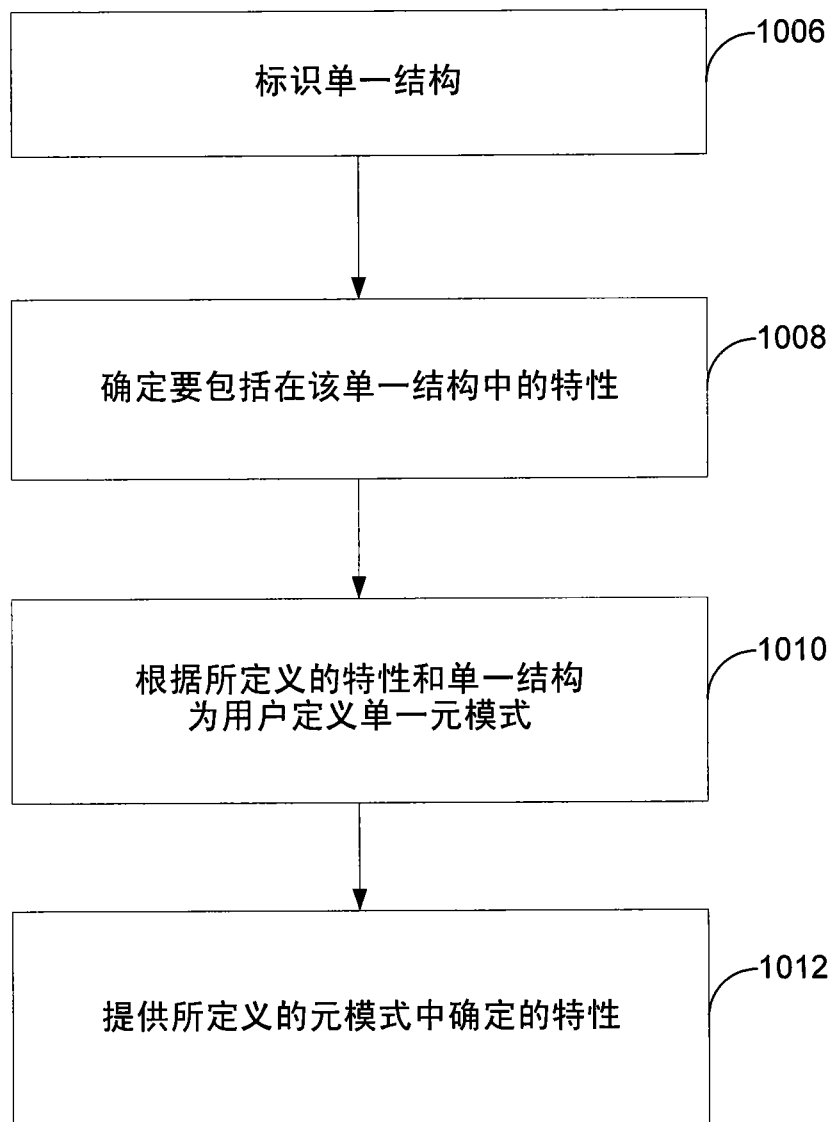


图 10B

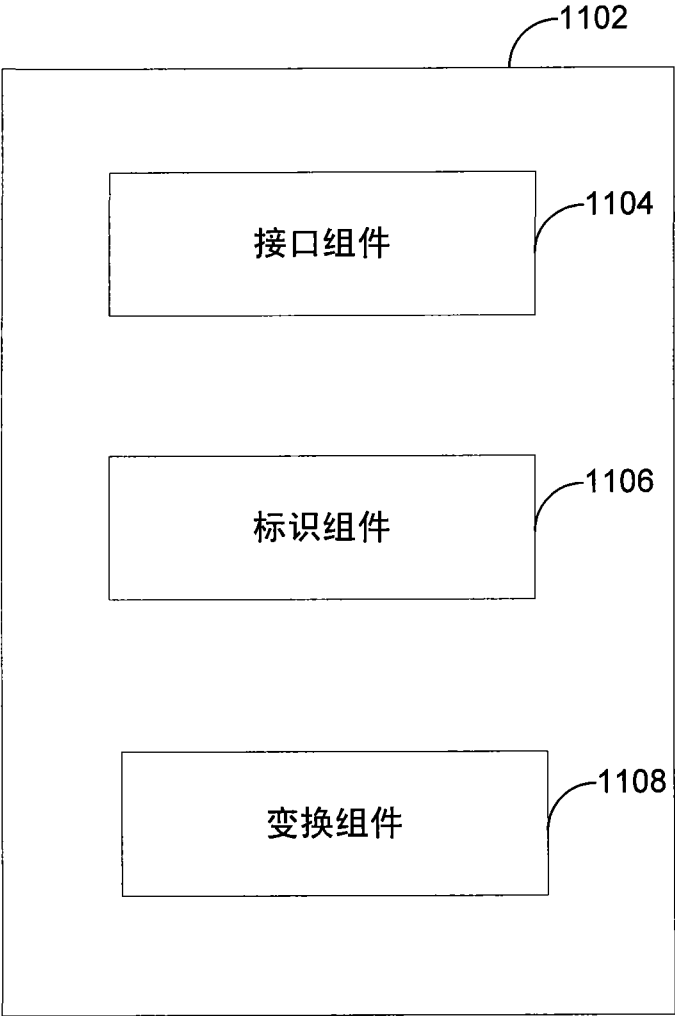


图 11A

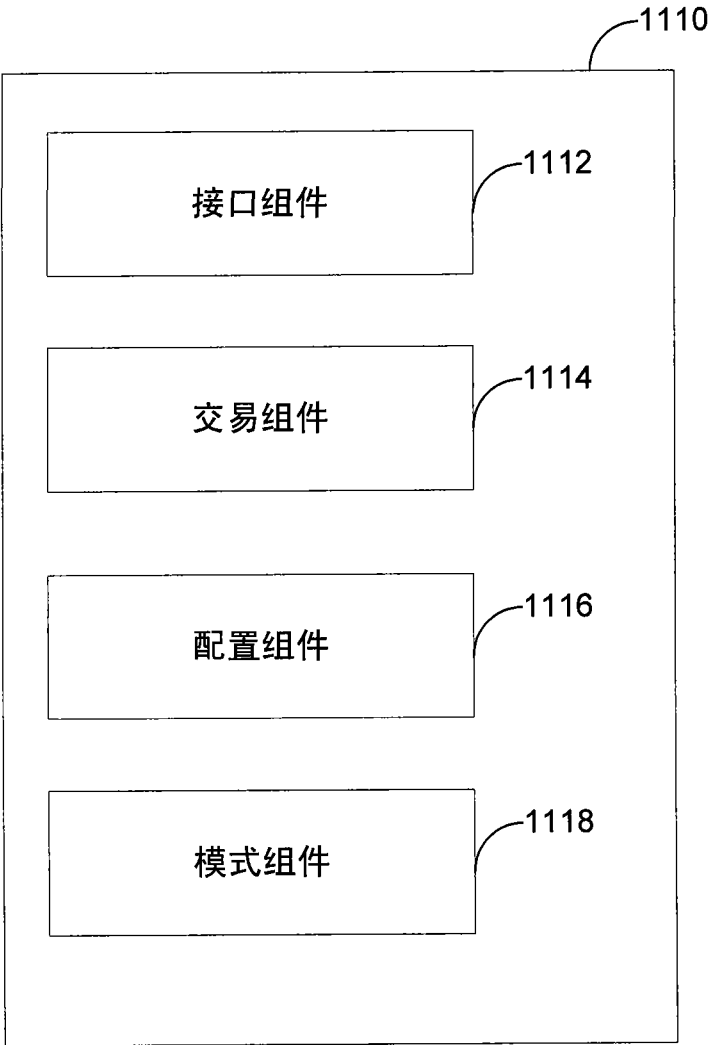


图 11B

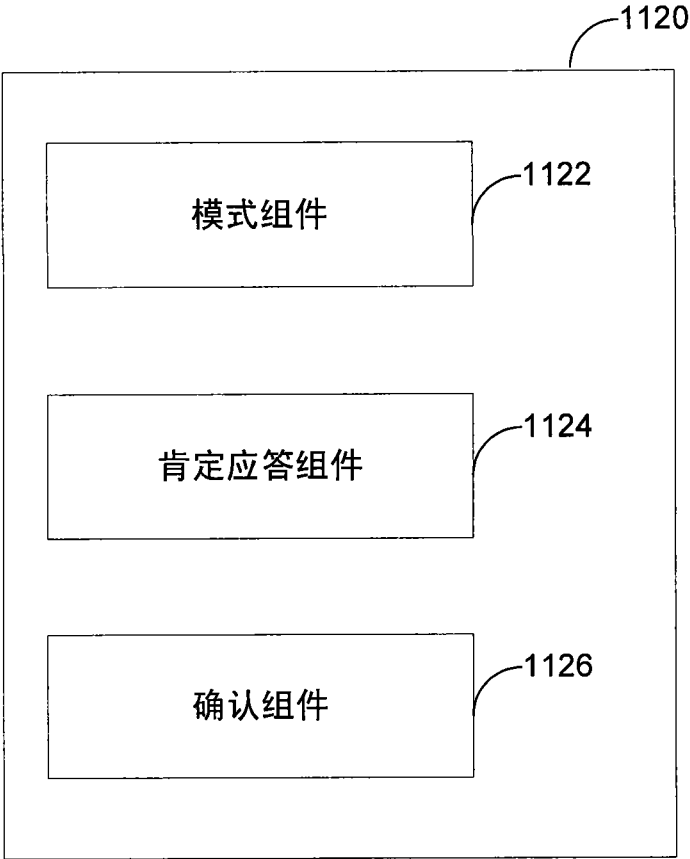


图 11C

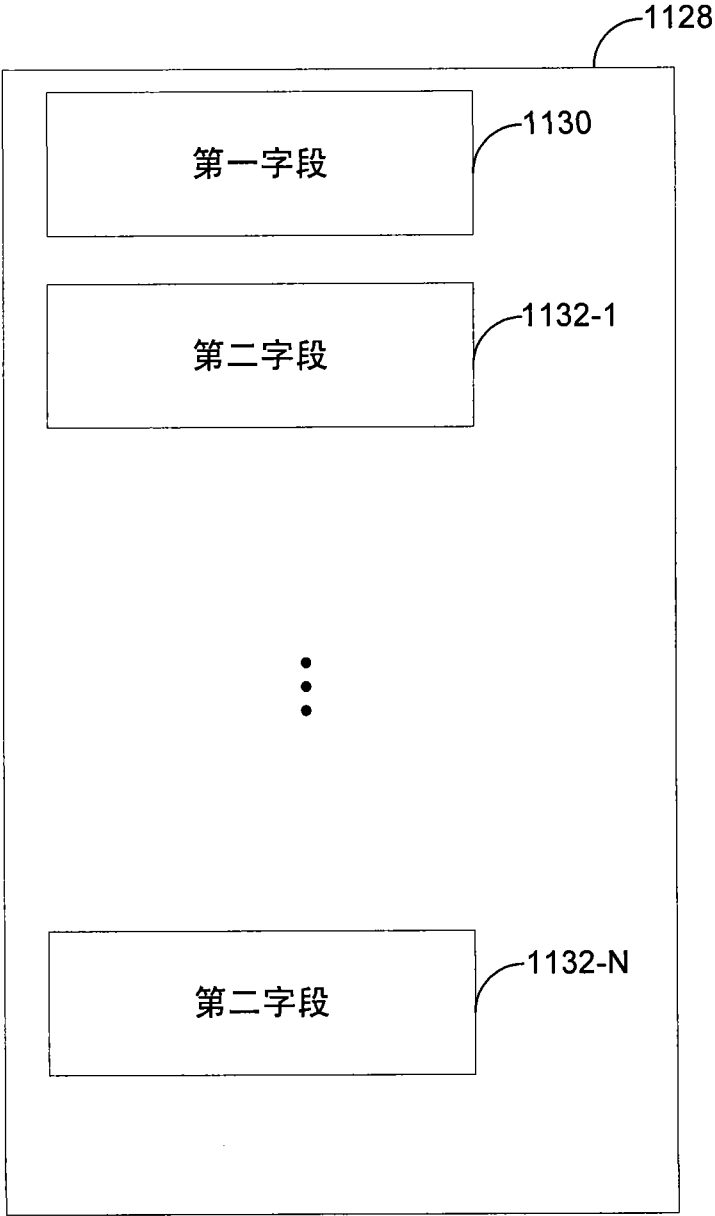


图 11D

