

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 9/54 (2006.01)

G06F 13/00 (2006.01)

G06F 15/00 (2006.01)



[12] 发明专利说明书

专利号 ZL 200580008344.6

[45] 授权公告日 2008年8月27日

[11] 授权公告号 CN 100414509C

[22] 申请日 2005.2.18

[21] 申请号 200580008344.6

[30] 优先权

[32] 2004.3.16 [33] JP [31] 074536/2004

[86] 国际申请 PCT/JP2005/002605 2005.2.18

[87] 国际公布 WO2005/088451 日 2005.9.22

[85] 进入国家阶段日期 2006.9.15

[73] 专利权人 佳能株式会社

地址 日本东京

[72] 发明人 藤井宪一 下野雅树 平田隆

[56] 参考文献

CN1443013A 2003.9.17

JP2004-7452A 2004.1.8

CN1205586A 1999.1.20

JP2000250735A 2000.9.14

审查员 王越

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所

代理人 曲瑞

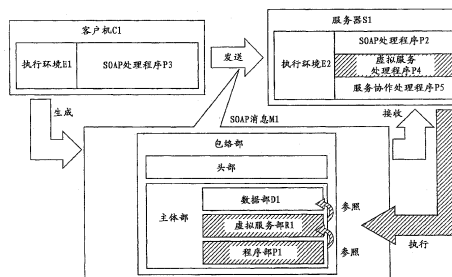
权利要求书1页 说明书16页 附图8页

[54] 发明名称

数据处理方法和数据处理装置

[57] 摘要

本发明提供一种数据处理方法、程序、装置、消息结构、消息生成方法及消息发送方法。在使用 SOAP 的消息收发中，当收发程序时，程序的描述位置部分会脱离 XML 的框架，从而在消息内会包含不可解释的部分。因此，SOAP 消息 M1 具有数据部 D1、表示处理数据部 D1 的标准的虚拟服务部 R1 和程序部 P1。服务器 S1 在执行包含在 SOAP 消息 M1 的程序部 P1 中的命令时，在 SOAP 消息 M1 的虚拟服务部 R1 表示通过执行的命令而进行的标准的情况下，按照虚拟服务部 R1 所示的标准来处理数据部 D1。



1. 一种数据处理方法，其特征在于：

接收具备数据部、标准部及程序部的消息，其中所述标准部表示处理所述数据部的标准，而所述程序部包含参考所述标准部的命令，

在执行包含在接收到的所述消息的程序部内的命令时，按照由执行的命令参考的所述标准部表示的标准，来处理包含在所述消息内的所述数据部。

2. 一种数据处理装置，其特征在于，具有：

接收单元，接收具备数据部、标准部及程序部的消息，其中所述标准部表示处理所述数据部的标准，而所述程序部包含参考所述标准部的命令；以及

处理单元，在执行包含在所接收到的所述消息的程序部内的命令时，按照由执行的命令参考的所述标准部表示的标准，来处理包含在所述消息内的所述数据部。

数据处理方法和数据处理装置

技术领域

本发明涉及数据处理方法、程序、装置、消息结构、消息生成方法及消息发送方法。

背景技术

图1图示了现有技术中的、使用客户机服务器方式下的消息的分布式处理的概要。

服务器S1、S2和客户机C1分别具有处理程序P1、P2、P3。然后，通过在客户机C1和服务器S1、S2之间收发消息M1来分布式地执行处理，所述消息M1的内部仅保存成为该处理程序的处理对象的数据D1。

当在分布式处理中以消息为基础时，所收发的消息M1只是数据。因此，即便服务器S1和S2是在互不相同的计算机语言或操作系统和服务器软件等执行环境下被构成的情况下，客户机C1和服务器S1或S2也能使用相同的消息M1。在该图中，具有执行环境E1和执行环境E2二种相互不同的执行环境的节点混合存在于客户机和服务器内。

但是，在该方式中，若使原消息内不仅包含数据还包含程序地进行发送接收，则产生了服务器侧和客户机侧的执行环境、特别是程序处理环境必须相同的问题。这是因为虽然可以在任意的编程语言、计算机环境中执行对数据的解释，但是程序的执行以依赖于该程序的描述计算机语言的执行环境为必要条件。由此，在以消息交换为基本的分布式处理技术中存在存在可以执行程序收发的情況下失去了所谓的“不同环境下的混合处理”的优点的问题。

作为有关分布式处理的已有技术,具有例如特开 2003-157178 号公报(专利文献 1)。该技术在客户机和服务器中的处理调用中,通过对参数(数据)执行 XML 变换,使得与互联网环境的协作变得容易。但是,利用该技术,只不过仅对请求的参数执行 XML 化,并不能最终解决用以执行数据和程序混合存在的对象的环境在发送侧和接收侧必须一致的问题。

作为解决这样的问题的手段,我们还考虑通过在执行环境上再叠加一个虚拟的执行环境(虚拟机·VM)来构筑共用的执行环境。但是,由于这不过使结局处的执行环境易于变为相同,而计算机语言自身的标准还使用了以往类型的描述方式,从而在现实中存在发生了如下所述的、消息中的数据和程序的描述方式不匹配的问题。

还存在作为计算机中分布式处理的构筑手法的 Web 服务技术。在 Web 服务技术中,利用作为可扩展的置标语言的 XML (eXtensible Markup Language)来描述消息。所谓 XML 是由 WWW 联盟(W3C)在 1998 年以在互联网上活用结构化文档为目标而制定的语言。关于 XML,可参照 <http://www.w3.org/XML/>下的标准。在 Web 服务技术中,将利用 XML 如此描述的消息的标准称为 SOAP (Simple Object Access Protocol)。SOAP 确定了消息的描述标准,是关于表明对什么样的数据如何进行表达的方法的规定。通过使用 Web 服务技术中的 SOAP,能够容易地实现图 1 的分布式处理。SOAP 也与 XML 一样,由 WWW 联盟 (W3C) 公开其标准,可参见 <http://www.w3.org/TR/SOAP/>下的内容。

在使用了 Web 服务技术的消息收发中,以使用用 XML 描述的 SOAP 为基础。利用了 XML 的 SOAP 还可以将消息作为结构化文档来处理。也可以将消息视为文档,与其任意部分相关地随意执行处理。

发明内容

发明所要解决的问题

如此,在使用了 SOAP 的消息收发中,将 XML 文档这种架构(粹

組み)作为消息的格式而予以维持是非常重要的。但是,由于彼此具有上述的虚拟执行环境,若在 SOAP 的结构中收发程序,则使仅仅是程序的描述位置部分脱离 XML 架构的问题显著化。

这是因为程序的描述方法本身并不特别考虑与 XML 的亲合性,有必要使用已有类型的描述方法。特别是在利用了以 SOAP 或 XML 文档流通为前提而构筑的 Web 服务技术的各节点中,这一点引发了在消息的内容中包含有不能解释的部分的状况。因此,丧失了 SOAP 的、所谓使用部分加密和签名等的 XML 进行通信传递(messaging)的优点。

在生成虚拟环境以使程序运行的方式中,由于必须假定虚拟环境可执行该计算机语言中可描述的全部程序,因此存在不能实现不会变为较大的目标,需要丰富的计算机资源的问题。即,还存在不能忽视由于需要构筑与本来的计算机环境不同的其他多重计算机环境而造成的资源溢出的问题。

若是采用覆盖到所有真正的处理的描述的语言,则也大多是采用编译方式的情况,在这种情况下,在执行时还需要进一步承担编译处理的负担。在以二进制形式发送编译完毕的对象的情况下,还会发生在执行前不能确认执行内容这样的问题。

解决问题的手段

为了解决上述问题,本发明的数据处理方法、数据处理程序、数据处理装置接收具备数据部、标准部及程序部的消息,其中所述标准部表示处理所述数据部的标准,而所述程序部包含用以执行由所述标准部所示的标准的处理的命令;在执行包含在接收到的所述消息的程序部内的命令时,决定表示通过所执行的命令而进行的处理的标准的标准部;和按照由所述消息中被确定为表示通过所述执行的命令而进行的处理的标准的标准部所示的标准,来处理所述数据部。

本发明的消息结构具有数据部、表示处理所述数据部的标准的标准部、以及程序部,其特征在于,所述程序部包含表示按照所述消息中的标准部所示的标准来处理所述数据部的命令。

发明效果

根据本发明，在消息中具备数据部、标准部和程序部，通过利用消息中的标准部来定义由程序部执行的处理的标准，能够在描述消息的架构的范围内描述程序和必要的的数据。

根据本发明，可以使用较少的计算机资源来执行程序。本发明的其他特征和优点将通过参照附图而做的以下说明变得更加明了。在附图中，对于相同或同样的结构，赋予相同的参考标记。

附图说明

图 1 是关于已有技术中的、在计算机之间的分布式处理的示意图。

图 2 是关于依据本发明实施方式的消息处理概要的示意图。

图 3 是依据本发明实施方式的、利用 PC(个人计算机)等计算机系统构成了客户机和服务器时该计算机系统的方框图。

图 4A、4B 是关于依据本发明实施方式的处理概要的流程图。

图 5 是关于依据本发明实施方式的处理细节的流程图。

图 6 是关于依据本发明实施方式的处理细节的流程图。

图 7 是关于消息结构的具体例子的示意图。

图 8 是关于依据本发明实施方式的、消息结构的具体例子的示意图。

附图标记的说明

P1 程序部

R1 虚拟服务部

D1 数据部

P2 SOAP 处理程序

P4 虚拟服务处理程序

P5 服务协作处理程序

具体实施方式

下面，将参照附图详细说明本发明的一种实施方式。尽管还一并记载了具体的处理例，但是，本发明并不限于该具体的处理例。

图2示出了本实施方式下的消息收发的概要。首先，说明 SOAP 消息的概要。SOAP 消息 M1 由包络(envelope)部、头部、以及主体(body)部构成。在包络部、头部按与已有技术相同的内容构成时，不对以 SOAP 为处理对象的已有处理系统请求变更。在本实施方式中，在主体部中，除通常按 SOAP 存储的数据部 D1 之外，还存储虚拟服务部 R1 及程序部 P1。

首先，客户机 C1 决定想在外部处理系统(服务器 S1)中执行的处理，并在 SOAP 处理程序 P3 中生成 SOAP 消息 M1。此时，不仅存储数据 D1，还存储表示希望如何处理该数据 D1 的程序 P1。此时，程序 P1 是利用服务协作描述语言来描述的，所述服务协作描述语言是与使用了 XML 的 SOAP 消息的亲合性高的计算机语言，这一点将在后面进行详细的说明。

此后，客户机 C1 将消息 M1 发送到服务器 S1。进行接收的服务器 S1 在是常规的 SOAP 消息时利用 SOAP 处理程序 P2 来执行处理，而在是内包有程序 P1 的 SOAP 消息的情况下，将处理转到虚拟服务处理程序 P4。虚拟服务处理程序 P4 通过存储在消息内的虚拟服务部 R1，将程序部 P1 和作为处理对象数据的数据 D1 连结在一起。之后，利用服务协作处理程序 P5 来执行程序 P1。之后，将执行后的结果从服务器 S1 返回到客户机 C1。

图3示出了依据本实施方式的、使用 PC(个人计算机)等计算机系统来构成客户机 C1 和服务器 S1 时该计算机系统的方框图。

在图3中，301 是对计算机系统进行控制的中央运算装置(以下记为 CPU)。

302 是随机存取存储器(以下记为 RAM)，用作 CPU 301 的主存储器，并用作执行程序的区域、该程序的执行区及数据区。

303 是记录有 CPU 301 的操作处理过程的只读存储器(以下记为

ROM)。ROM 303 包括：程序 ROM，它记录作为执行计算机系统控制的系统程序的基本软件(操作系统，OS)；以及，数据 ROM，它记录用以运行系统所必需的信息等。也可以使用后述的 HDD 309 来代替 ROM 303。图 2 中客户机 C1 的 SOAP 处理程序 P3 也记录在客户机 C1 的 ROM 303 或者 HDD 309 内。另外，服务器 S1 的 SOAP 处理程序 P2、虚拟服务处理程序 P4、服务协作处理程序 P5 也记录在服务器 S1 的 ROM 303 或者 HDD 309 内。

304 是网络接口(以下记为 NETIF)，用于诊断用以经由网络与其他计算机系统之间执行数据传输的控制和连接状况。图 2 中的客户机 C1 通过客户机 C1 的 NETIF 304 向服务器 S1 发送 SOAP 消息 M1，图 2 中的服务器 S1 通过服务器 S1 的 NETIF 304 接收 SOAP 消息 M1。图 2 中的服务器 S1 将接收到的 SOAP 消息 M1 存储在 SOAP 消息 M1 内。

305 是视频 RAM(以下记为 VRAM)，用于展开在示出了计算机系统的运行状态的 CRT 306 的画面上所显示的图像并进行其显示控制。306 是显示装置，例如是显示器(以下记为 CRT)。307 是对来自后述的外部输入装置 308 的输入信号执行控制的控制器。

308 是用于接受计算机系统的使用者对计算机执行的操作的外部输入装置，例如是键盘等。

309 是用于程序、图像信息等数据保存用的存储装置。例如是硬盘等。

310 是外部输入输出装置，用于输入输出例如软盘(注册商标)，CD-ROM 驱动器等可移动盘。用于从媒体中读出上述应用程序。以下记为 FDD。

HDD 309 内存储的应用程序和数据也可以保存在 FDD 310 内使用。300 是用于在上述各块之间执行连接的输入输出总线(地址总线、数据总线、以及控制总线)。

图 4A、图 4B 以流程图的形式示出了客户机 C1、服务器 S1 的处理步骤。图 4A 表示客户机 C1 所具有的程序的一部分，图 4B 表示

服务器 S1 所具有的程序的一部分。客户机 C1 的 CPU 301 从 ROM 303 或 HDD 309 中读出该程序，然后执行图 4A 所示的操作。服务器 S1 的 CPU 301 从 ROM 303 或 HDD 309 中读出该程序，然后执行图 4B 所示的操作。

首先，在图 4A 的步骤 S301 中，客户机 C1 决定想在外部处理系统(服务器 S1)中执行的处理。该处理例如是基于来自外部输入装置 308 的输入而决定的。

接着，利用 SOAP 处理程序 P3 生成 SOAP 消息 M1。即，首先，在步骤 S302 中，将数据 D1 保存到 SOAP 消息 M1 的主体部内。然后，在步骤 S303 中，生成表示希望如何处理该数据 D1 的程序 P1 以及虚拟服务部 R1，并将其保存到 SOAP 消息 M1 的主体部内。关于步骤 S303 的处理，后面将会有更详细的描述。虚拟服务部 R1 是表示处理数据 D1 的标准的标准部。

接着，在步骤 S304 中，客户机 C1 通过客户机 C1 的 NETIF 304，将 SOAP 消息 M1 传送到服务器 S1。

服务器 S1 在图 4B 的步骤 S305 中，利用 SOAP 处理程序 P2 来接收该消息。即，利用 SOAP 处理程序 P2 来处理通过服务器 S1 的 NETIF 304 而接收的消息 M1。

之后，在步骤 S306，判断是否是常规的 SOAP 消息，若是常规的 SOAP 消息，则在步骤 S307，利用常规的 SOAP 处理程序 P2 来执行处理。

在步骤 S306 中，若判断为内包有程序 P1，则将处理转到虚拟服务处理程序 P4。之后，在步骤 S308 中，虚拟服务处理程序 P4 根据消息 M1 内所存储的虚拟服务部 R1 的信息来取得构成虚拟服务所必需的信息，并在步骤 S309 中构筑用于将程序部 P1 和作为处理对象的数据的数据 D1 连结在一起的虚拟服务。之后，在步骤 S310 中，将该信息发送到服务协作处理程序 P5。在步骤 S311 中，服务协作处理程序 P5 执行程序 P1。之后，在步骤 S312 中，将执行后的结果从服务器 S1 返回客户机 C1。关于步骤 S309，后面将进一步详细描述其处理。

下面说明在上述实施方式中，在客户机 C1 生成 SOAP 消息 M1 时，用作程序 P1 的描述语言的服务协作描述语言。在本实施方式中，所谓服务协作描述语言是指在 Web 服务技术中供服务协作作用的协作描述语言。即，是用于在 Web 服务技术中，逐次调用提供多个服务一侧的功能、或在其处理结果中，一边执行条件分支一边执行特定的更大处理的技术。作为具体技术，可以举出被称为 BPEL4WS 或 WSCI 的技术。

所谓 BPEL4WS 是 Business Process Execution Language for Web Service (BPEL4WS) 的缩写，是作为用于使多个服务协作的复合服务描述的基准而占有一席之地服务协作描述语言。标准是由 OASIS(<http://www.oasis-open.org/>) 的 OASIS Web Services Business Process Execution Language TC 管理着。在使用 BPEL4WS 时，生成例如对 Web 服务调用的执行、数据的操作、故障(fault)的通知、或过程的结束等各种活动，并通过使这些活动连结在一起，可以生成复杂的过程。这些活动能够嵌套在定义了其执行方法(例如是顺序执行、并行执行、还是基于特定条件来执行)并使之结构化的活动中。

WSCI 是 Web Service Choreography Interface 的缩写，它同样是使复合服务协作的、用于服务描述的服务协作描述语言。标准由 WWW 联盟(W3C)管理，可参照 <http://www.w3.org/TR/wsci/>。

这些服务协作描述语言中的任何一种都具有用 XML 描述的特征，与使用了虚拟计算机环境的已有技术相比，没有将已有类型的计算机语言存储到 SOAP 消息内时的问题。即，不会产生丧失如下优点的问题：将已有的语言存储到作为 XML 文档的 SOAP 消息内时所产生的、作为 XML 的优点的、可部分处理的优点。由于即便存储了程序，也可以维持所谓的可处理整个消息的 XML 文档的状态，因此，与已有的常规 SOAP 消息的兼容性非常高，没有必要向执行例如消息传送等中间处理的节点等强行追加新的处理程序等。

但是，在使用服务协作描述语言的情况下，首先他们都以在各种 Web 服务技术中使提供服务的节点的功能协作为主要目的，存在不以

操作存储在 SOAP 消息内的数据为目的的问题。即，服务协作语言只有用于推进使用了常规 Web 服务技术的服务器(服务提供节点)的能力，而不能执行针对存储在特定 SOAP 消息内的数据的推进。因此，难以原封不动地将其用作用以描述以存储在 SOAP 消息内的数据为对象的处理的计算机语言。

因此，在本实施方式中，还为了不对服务协作描述语言的标准和处理系统施加一切修正地实现对存储在 SOPA 消息内的数据的参照，通过在消息 M1 内设置虚拟服务部 R1 的同时还在服务部 S1 内设置虚拟服务处理程序 P4 来应对。按照这种方法，没有必要扩展服务协作描述语言，以往存在的服务协作描述语言的处理系统在实际安装时没有必要支持该扩展标准。

下面说明消息 M1 内的虚拟服务部 R1、服务器 S1 中的虚拟服务处理程序 P4 在本实施方式中的具体功能和处理。如前所述，服务协作描述语言的处理对象只是 Web 服务技术中的服务，不具有参照或操作在 SOAP 消息内同时存储的数据等的功能。在本实施方式中，通过向服务协作描述语言的执行处理部(P5)提供只能应用在该 SOAP 消息处理中的虚拟 Web 服务的服务提供部，如果服务协作描述语言的处理部(P5)执行对应于常规的 Web 服务器的处理，则实际上实现了变换为参照或操作存储在该 SOAP 消息内的数据的结构。因此，在本实施方式中，能够不对服务协作描述语言的标准和处理系统施加修正，就能够实现对存在于 SOAP 消息内的数据部的参照或操作。

更具体地说，在构成图 2 中的 SOAP 消息 M1 时，将用于参照所存储的数据部 D1 的虚拟服务描述作为虚拟服务部 R1 进行存储，将程序部 P1 和数据部 D1 的关系设置为使存在于服务器 S1 内的虚拟服务处理程序 P4 能够处理。虚拟服务处理程序 P4 基于这些信息构筑虚拟的 Web 服务提供部。通过将其提供给服务协作处理程序 P5，服务协作处理程序 P5 执行与常规的服务协作处理相同的处理，而在实际上通过虚拟服务处理程序 P4 实施了对数据 D1 的操作。该虚拟服务部 R1 是表示处理数据部 D1 的标准的标准部。

图 5 是为了说明本处理、以流程图的形式表示出利用 SOAP 处理程序 P3 执行的步骤 S303(图 4A)的进一步详细化的处理的图。在步骤 S401 中, 获取在步骤 S302 中存储在 RAM 302 内的数据 D1。在步骤 S402 中, 生成基于在步骤 S301 中所决定的内容的处理程序 P1, 并将其存储在消息 M1 的主体部内。此时生成的程序 P1 按图 2 中的服务协作处理程序 P5 可处理的服务协作描述语言来描述。在步骤 S403 中, 从数据 D1 中提取出步骤 S402 中所生成的程序 P1 使用(参照或操作)的数据。之后, 在步骤 S404 中, 求出具有变更这些数据的功能的虚拟 Web 服务的标准, 并在步骤 S405 中将其做成图 2 的虚拟服务部 R1 并存储在消息 M1 的主体部内。

作为执行如此动态生成程序、并将其组入消息内的功能的已有技术, 例如有特开 2003-223376 号公报。该技术使用应用程序编程模块来构筑消息的内容, 代码生成程序由此自动生成用于处理客户机和服务器间的协议的个别程序。该技术是以自动生成将处理集中到作为消息传输部分的协议内的程序为目的, 本实施方式中步骤 S402 的程序生成是实现利用计算机语言来描述处理该消息内容的方法的技术, 在这一点上二者是不同的。

图 6 是为了说明本处理而以流程图的形式来表示利用虚拟服务处理程序 P4 处理的步骤 S309(图 4B)的更加详细的处理。虚拟服务处理程序 P4 在步骤 S308 中取得了保存在消息 M1 内的虚拟服务部 R1 的信息, 在步骤 S411 收取该信息。在步骤 S412 中, 根据虚拟服务部 R1 决定应虚拟生成的 Web 服务的提供功能的标准, 并在步骤 S413 中, 构成虚拟服务提供的实际功能并开始执行 Web 服务。由此, 可以使程序部 P1 和作为处理对象的数据的数据 D1 连结在一起。接着, 在步骤 S310 中将该信息发送到服务协作处理程序 P5。在步骤 S311 中, 服务协作处理程序 P5 执行程序 P1。

图 7 是仅仅由数据构成的消息的例子。这是在作为已有技术的 Web 服务技术中被处理的消息, 变成了基于使用了 XML 的 SOAP 而构成的消息。在 SOAP 中, 在包络部中具有头部和主体部, 进而在主

体部中，保存有名称为 value1 和 value2 的内容作为数据。

图 8 是利用本实施方式生成的消息的例子。它示出了在保持了在作为已有技术的 Web 服务技术中能够进行处理的兼容性的同时，还同时保存了数据和程序。与图 7 相同，在 SOAP 消息中，包络部中具有头部和主体部，而且在主体部中，作为数据还保存有名称为 value1 和 value2 的内容。还追加了虚拟服务部和程序部。虚拟服务部是表示处理数据部的标准的标准部。

以下，说明图 8 中的处理 SOAP 消息时的实际流程。该例子示出了接受 2 个值、进行合计运算后返回结果的简单处理。

首先，所发送的 SOAP 消息与由图 2 中在存在于客户机 C1 上的执行环境 E1 下进行操作的 SOAP 处理程序 P3 生成的 SOAP 消息 M1 相当。

首先，在图 3 的步骤 S301 中确定想在外部处理系统(服务器 S1)中执行的处理是运算 2 个数字的合计的处理。接下来，在步骤 S302 中，在 SOAP 处理程序 P3 中，与常规的 SOAP 消息生成处理相同地生成图 7 中的由要素 <env:Envelope> 构成的包络部、由要素 <env:Header> 构成的头部、由要素 <env:Body> 构成的主体部，然后生成由要素 <m:data> 构成的数据部 D1。另外，在数据部 D1 内，在要素 <m:value1> 和 <m:value2> 内保存了两个值 100 和 200。

另外，在步骤 S303 中追加了图 8 中的程序部 P1 和虚拟服务部 R1。

程序部 P1 的内容是计算上述 2 个值的合计，该程序部 P1 被追加到要素 <p:program> 中。该程序是利用作为用 XML 表达的服务协作描述语言的 BPEL4WS 来描述的。由于此时成为对象的数据是描述在该 SOAP 消息的数据部 D1 中的要素 <m:value1> 和 <m:value2> 的内容，因此，为了访问这些要素的信息而在程序部 P1 的要素 <p:partnetLinks> 内嵌入对虚拟服务“virtualService”的参照。

接下来，生成了虚拟服务部 R1。虚拟服务部 R1 由要素 <v:virtualService> 表示，进而根据内部的要素 <v:definitions> 中的

name 属性，以接受先前所述的参照的形式被赋予了“VirtualService”这样的名称。表示由虚拟服务内的访问单元返回的数据的类型的、名称为 value1 的消息被定义为要素<v:message>。另外，作为对数据部 D1 保持的数据<m:value1>和<m:value2>的访问单元的 getValue1 和 getValue2 在要素<v:portType>中被定义。getValue2 在图 8 中被省略。虚拟服务部 R1 是表示处理数据部 D1 的标准的标准部。

在图 8 中，数据部、虚拟服务部(标准部)、程序部分别由名称为 data、virtualService、program 的标签来定义。

在按照这种方式生成图 8 所示的 SOAP 消息后，在步骤 S304 中通过客户机 C1 的 NETIF 304 而被发送。

发送出的 SOAP 消息由图 2 中服务器 S1 中的 SOAP 处理程序 P2 接收。即，通过服务器 S1 的 NETIF 304 接收的消息 M1 由 SOAP 处理程序 P2 处理。所接收到的 SOAP 消息 M1 被返回到图 4B 中的步骤 S305 之后的处理。在步骤 S306 中，判定程序部 P1 是否存在于消息 M1 的主体部内。图 8 中的 SOAP 消息由于在要素<env:Body>内包含有要素<v:virtualService>和要素<p:program>，因此，被判断为是包含虚拟服务部 R1 和程序部 P1 的 SOAP 消息。

这里，控制转移到图 2 中的虚拟服务处理程序 P4。接着，在步骤 S308 中取得构筑虚拟服务所必需的信息。然后，在步骤 S309 中根据存在于图 8 的 SOAP 消息 M1 的主体部内的、作为虚拟服务部 R1 的要素<v:virtualService>的内容来决定应当构筑的虚拟服务的标准。例如，根据要素<v:definitions>的 name 属性，使虚拟服务的名称为“VirtualService”。之后，图 2 中的虚拟服务程序 P4 基于所获得的标准，开始提供虚拟服务。

然后在步骤 S310 中，虚拟服务处理程序 P4 将如此构筑的虚拟服务的信息与存储在 SOAP 消息 M1 内的程序部 P1 的信息合在一起发送到服务协作处理程序 P5。这种情况下，由于服务协作处理程序 P5 能够执行作为 Web 服务协作语言的 BPEL4WS，所以作为步骤 S311，开始逐次执行程序部 P1 内所描述的处理。此时，由存在于处

理程序部 P1 中的<p:partnetLinks>所示的服务“VirtualService”是如先前那样构筑的虚拟服务。例如，在执行存在于处理程序部 P1 内的<p:invoke partnerLink=“virtualService” operation=“getValue1”>，服务协作处理程序 P5 执行对常规服务的访问过程时，实际上通过虚拟服务处理程序 P4 接受了该请求，并根据在 SOAP 消息 M1 的数据部 D1 中所描述的<m:value1>100</m:value>，将 100 作为执行结果而返回。

即，在执行<p:invoke partnerLink=“virtualService” operation=“getValue1”>时，首先，服务协作处理程序 P5 从服务的名称(“VirtualService”)开始发出用于解决服务地址的询问，假如是关于名称“VirtualService”的询问，则虚拟处理程序 P4 应答本地主机的地址。该本地主机的地址例如其 IPv4 为 127.0.0.1。在将多个虚拟服务应答(提供)给虚拟服务处理程序 P4 的情况下，变为 IP 地址与虚拟服务标识符的组合，例如变为“http://127.0.0.1/virtualService”。即，在执行在接收到的消息 M1 中的程序部内所包含的命令<p:invoke partnerLink=“virtualService” operation=“getValue1”>时，决定表示通过执行的命令而进行的处理的标准的标准部(根据服务名称(“VirtualService”)来解决服务的地址)，但在这里，消息 M1 中的虚拟服务部 R1 被决定为表示通过执行的命令<p:invoke partnerLink=“virtualService” operation=“getValue1”>而进行的处理的标准。

“VirtualService”是用于识别表示通过该命令而进行的处理的标准的标准部的标准部标识符。“getValue1”是用于识别处理数据的标准的标准标识符，它们分别表示消息 M1 中的虚拟服务部 R1、以及在虚拟服务部 R1 中规定的数据的处理标准。“getValue1”表示从数据 D1 中取得数据标识符为 Value1 的数据。

接受该应答后，服务协作处理程序 P5 向该本地主机的地址请求执行存在于该服务(“VirtualService”)内的特定处理(“getValue1”)。该请求经由虚拟服务处理程序 P4 而被接受。虚拟服务处理程序 P4 在请

求了处理名中以“get”为前缀(例如 getValue1)的处理的情况下, 检查名称为 get 之后的值(此时为 Value1)的要素(数据)是否存在于消息 M1 的数据 D1 内. 如果存在, 则将其值(100)返回给服务协作处理程序 P5. 即, 在执行所接收的消息 M1 中的程序部 P1 内包含的命令<p:invoke partnerLink=“virtualService” operation=“getValue1”>时, 在所接收的消息中的虚拟服务部 R1(标准部)表示通过执行的命令而进行的处理的标准的情况下, 利用虚拟服务部 R1 所示的标准来处理数据部 D1. 这里, 服务的标准对于“getValue1”而言, 是从数据部 D1 中取得数据标识符是 Value1 的数据的值(100).

这样, 在针对存在于 SOAP 消息 M1 内的数据部 D1 内所保存的数据执行了处理之后, 作为步骤 S312, 返回处理结果. 由于该 SOAP 消息的处理目的是接受 2 个值后计算其合计, 所以对 100 + 200 而言返回 300.

在上述实施方式中, 作为描述保存在消息中的程序部的程序的计算机语言, 使用了作为服务协作描述语言的 BPEL4WS, 但是, 不用说它也可以是 WSCI 等其它以服务协作描述为目的的计算机语言. 此时, 该计算机语言采用以 Web 服务作为其操作对象, 并在描述中采用 XML 方式.

在上述实施方式中, 仅仅记载了单一的客户机和单一的服务器的关系, 但本发明也同样适用于多个客户机和多个服务器的关系.

本发明适用于由多个装置(例如 AV 装置、家用电器产品装置、计算机装置、接口装置等)构成的装置, 也可以适用于由一个装置构成的装置.

上面虽然详细描述了实施例, 但本发明还可以采取作为例如系统、装置、方法、程序或存储媒体(记录媒体)等的实施方式. 具体而言, 既可以适用于由多个设备构成的系统, 也可以适用于由一个设备构成的装置.

另外, 本发明包含通过将实现前述实施方式的功能的软件程序(实施方式中与图示的流程图对应的程序)直接或远程提供给系统或装

置、该系统或装置的计算机读出并执行该所提供的计算机代码来实现本发明的情况。

因此，为了通过计算机来实现本发明的功能处理，安装在该计算机内的程序代码本身也是实现本发明的方案。即，本发明还包含用于实现本发明功能处理的计算机程序本身。

在这种情况下，如果具有程序功能，则也可以是目标代码、由解释程序执行的程序、提供给 OS 的脚本数据等形式。

作为用于提供程序的记录媒体，包括例如软盘(注册商标)、硬盘、光盘、磁光盘、MO、CD-ROM、CD-R、CD-RW、磁带、非易失性存储卡、ROM、DVD (DVD-ROM、DVD-R)等。

作为其它提供程序的方法，也能够通过使用客户机计算机的浏览器而连接到互联网的主页上，并从该主页下载本发明的计算机程序本身、或者将包含压缩的自动安装功能的文件下载到硬盘等记录媒体上来进行提供。也可以通过将构成本发明的程序的程序代码分割为多个文件，从不同的主页下载各个文件来实现。即，本发明还包含针对多个用户下载用于在计算机中实现本发明的功能处理的程序文件的 WWW 服务器。

也可以通过对本发明的程序加密后将其存储到 CD-ROM 等存储媒体内，然后分发给用户，针对清除了规定条件的用户，经互联网从主页下载解密密钥信息，使用该密钥信息来执行加密程序，并将其安装到计算机内来实现。

另外，除了计算机通过执行读出的程序来实现前述实施方式的功能外，还可以由在计算机上运行的 OS 等基于该程序的指示来执行实际处理的一部分或全部，通过该处理来实现前述实施方式的功能。

另外，从记录媒体中读出的程序，在被写入插入到计算机内的功能扩展板和具备连接到计算机的功能扩展单元的存储器中之后，基于该程序的指示，由 CPU 等执行该功能扩展板和功能扩展单元等内具备的实际处理的一部分或全部，通过该处理也可以实现前述实施方式的功能。

本发明并不限于上述实施例,还考虑了各种变更和修改等。因此,本发明的技术范围是基于后附的保护范围而确定的。

优先权主张

本申请要求以于 2004 年 3 月 16 日提出日本专利申请第 2004-074536 号的优先权,在此引用其全部记载内容。

图 1

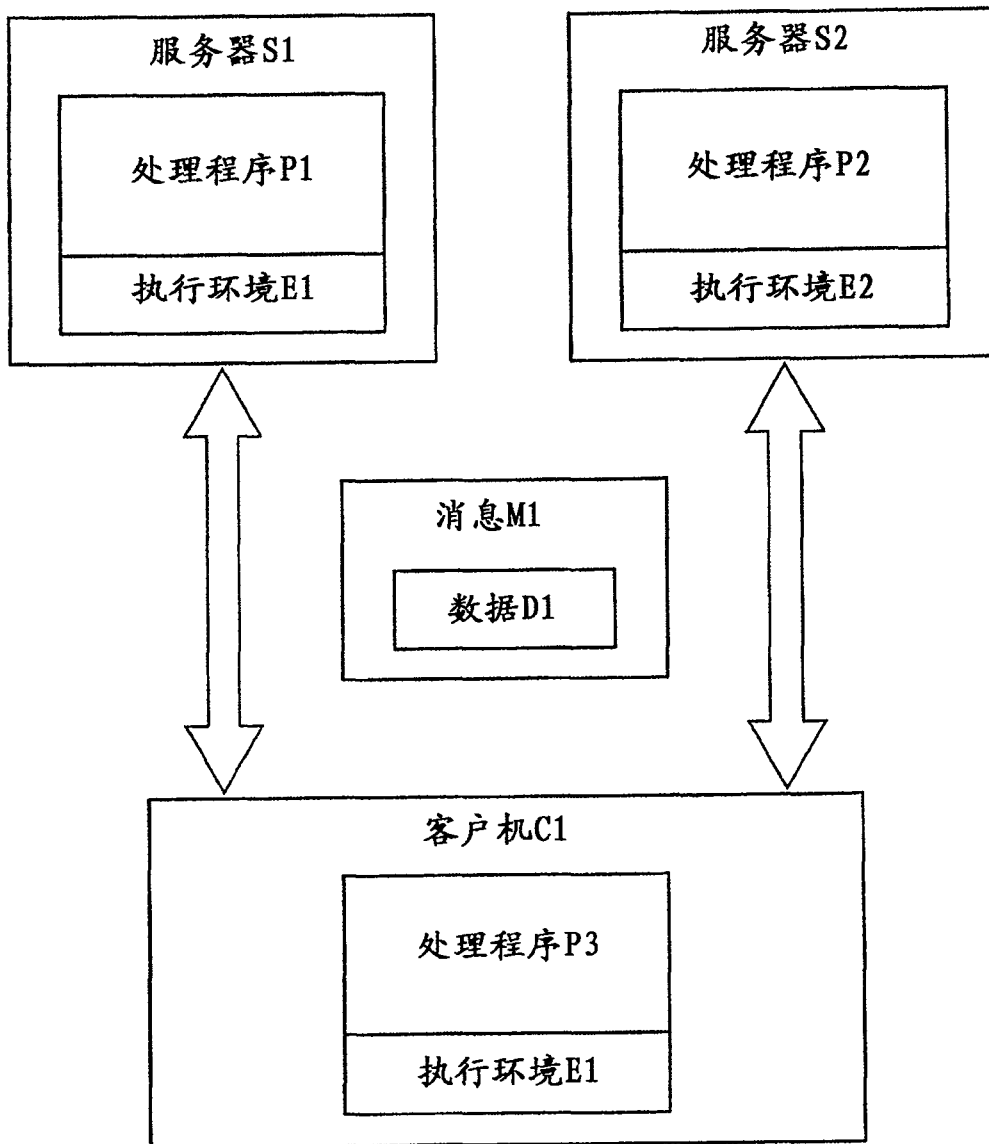


图 2

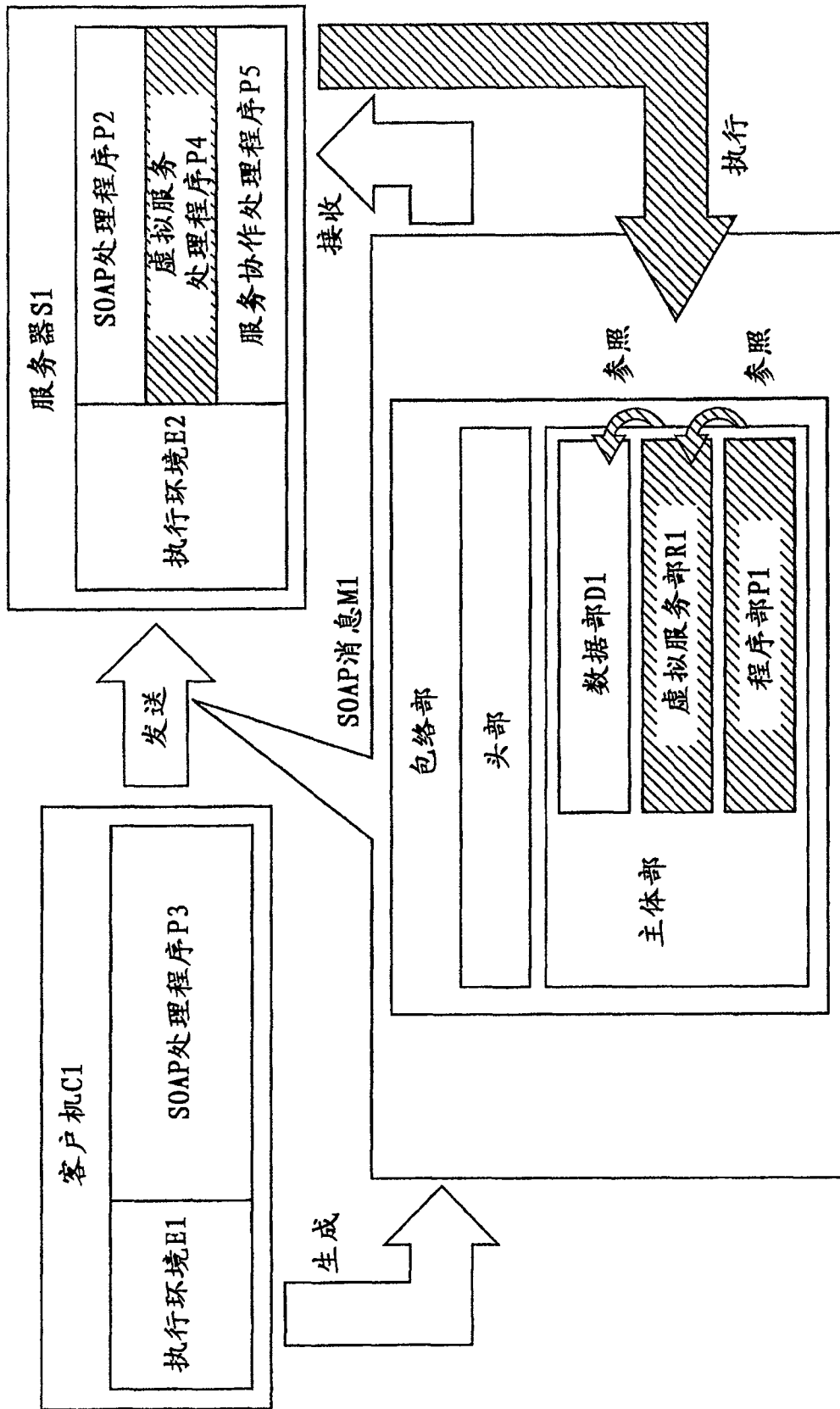


图 3

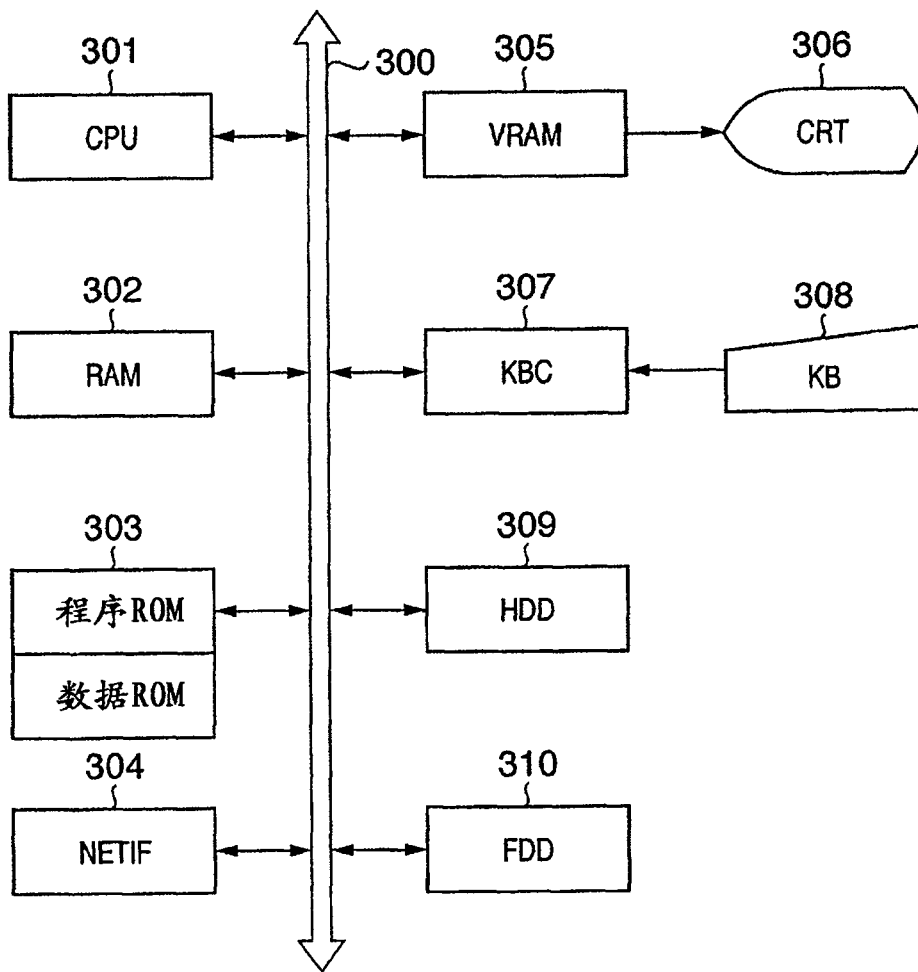


图 4A

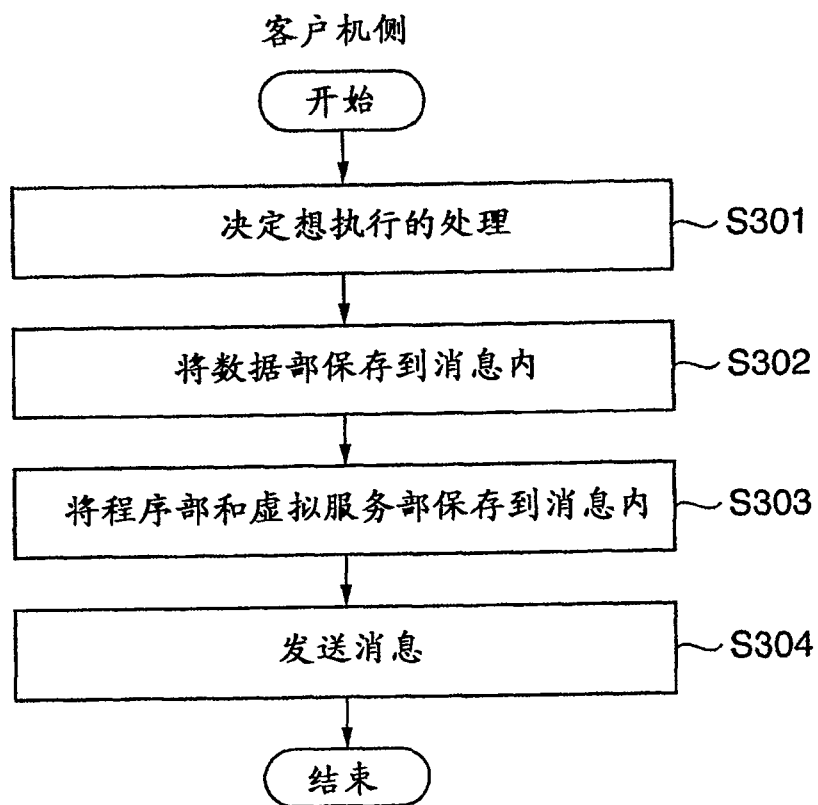


图 4B

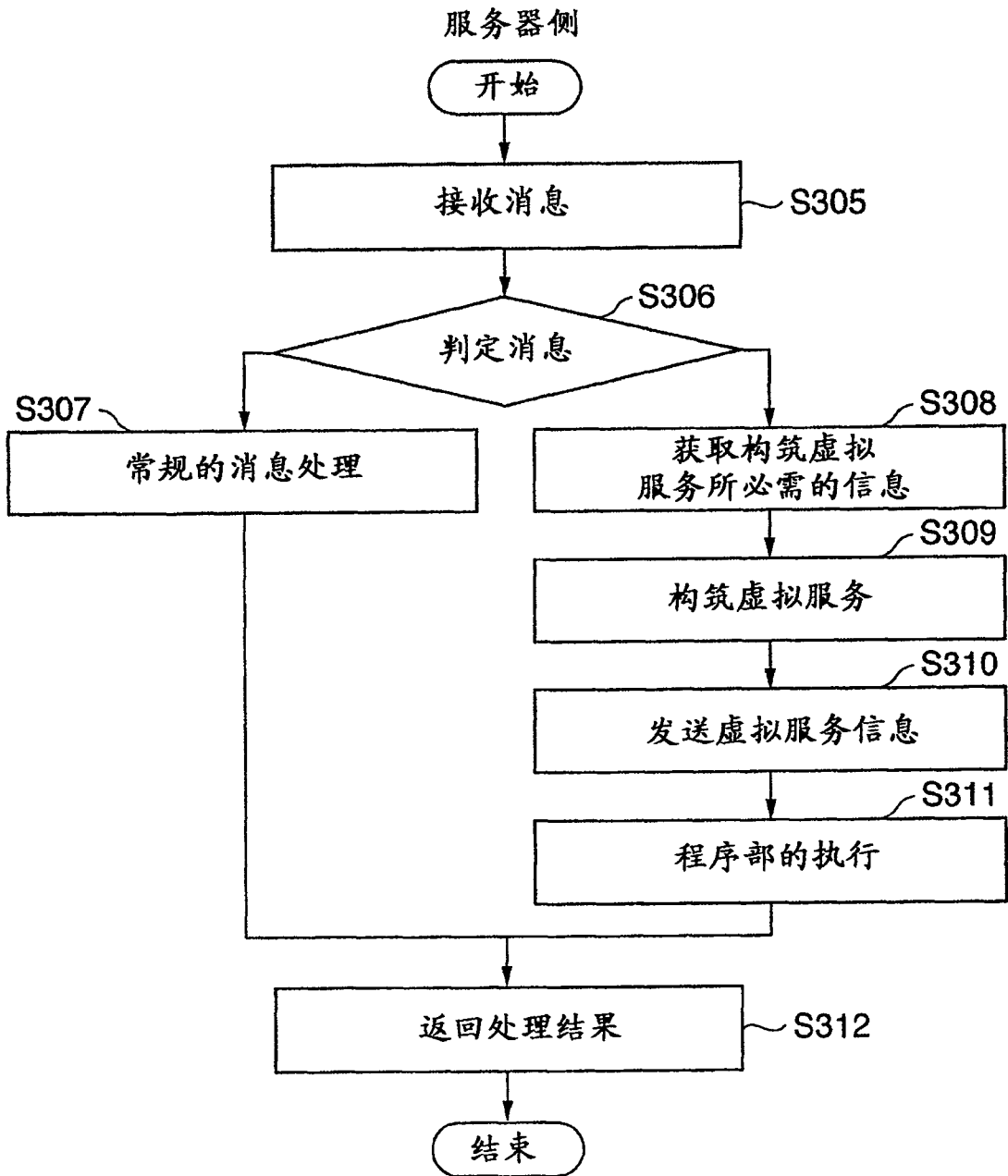


图 5

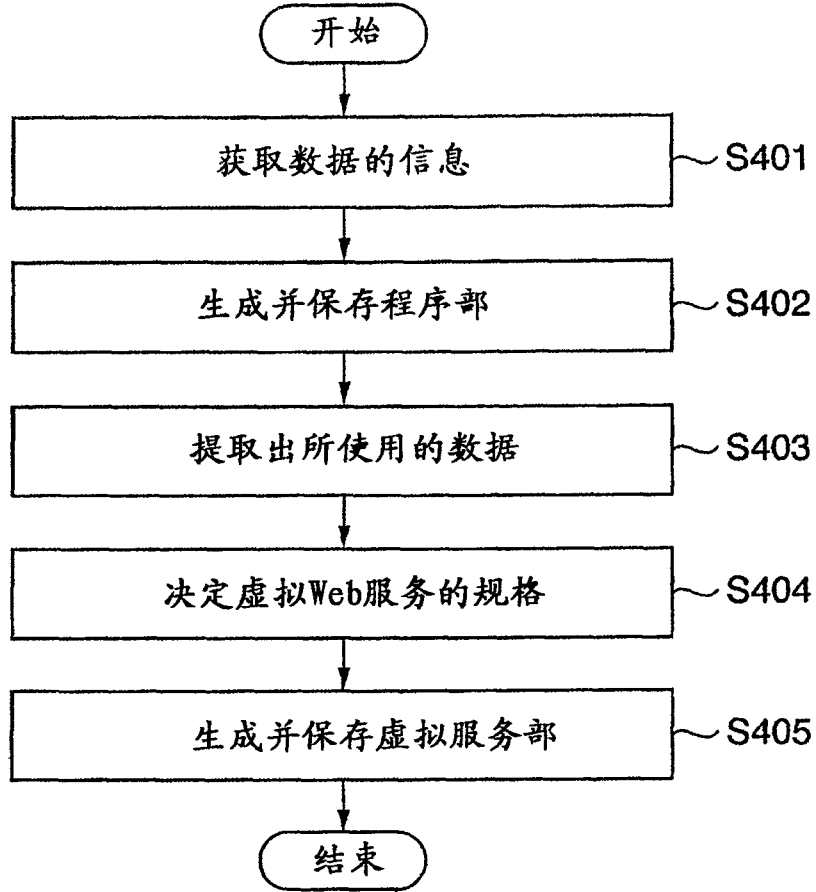


图 6

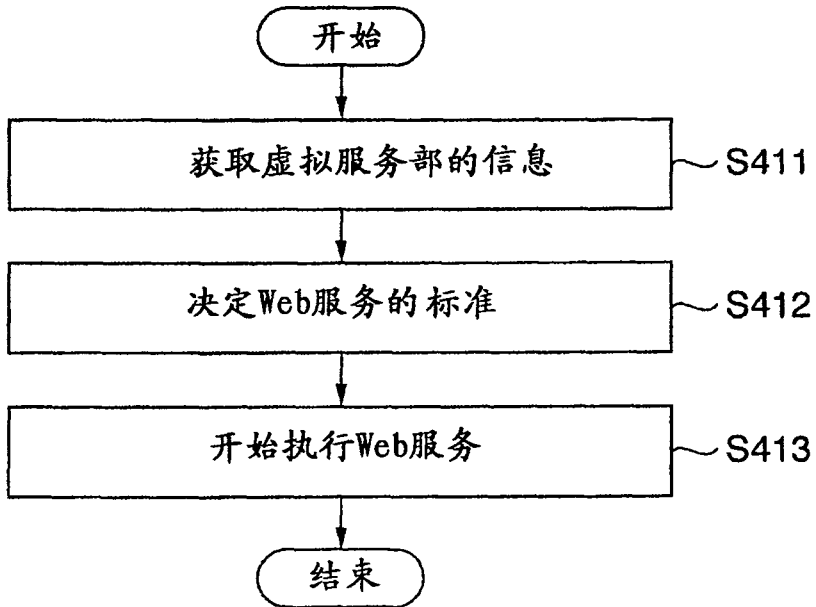


图 7

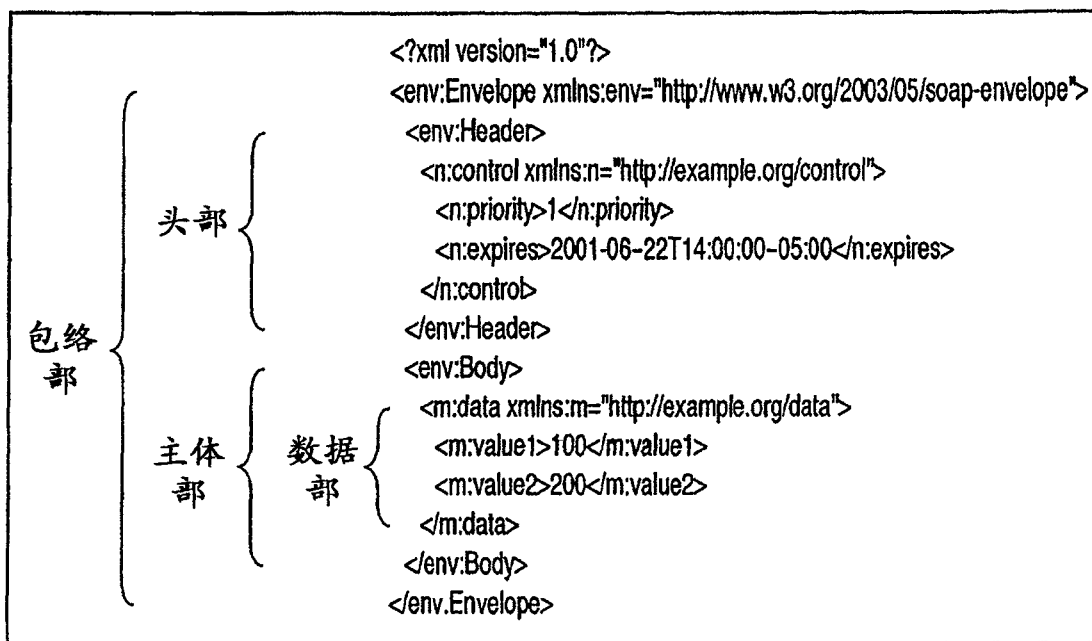


图 8

