



(12) 发明专利申请

(10) 申请公布号 CN 102202089 A

(43) 申请公布日 2011.09.28

(21) 申请号 201110105416.6

(22) 申请日 2011.02.15

(30) 优先权数据

10425035.2 2010.02.15 EP

12/777,015 2010.05.10 US

(71) 申请人 埃森哲环球服务有限公司

地址 爱尔兰都柏林

(72) 发明人 F·穆恩戈 A·弗兰科

E·帕特里尼

(74) 专利代理机构 北京市金杜律师事务所

11256

代理人 鄞迅 姜彦

(51) Int. Cl.

H04L 29/08(2006.01)

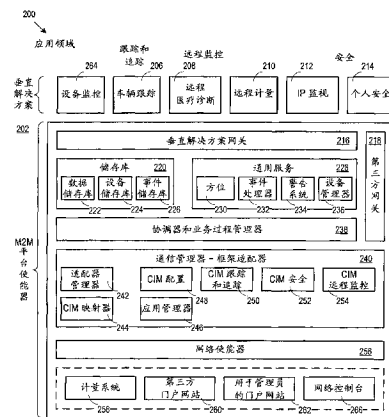
权利要求书 2 页 说明书 92 页 附图 49 页

(54) 发明名称

机器到机器架构

(57) 摘要

本发明涉及一种机器到机器架构。具体地，一种机器到机器通信平台为设备控制和解决方案托管提供灵活的系统。特别地，通信平台托管和管理与外部设备交互的第三方垂直解决方案。第三方网关向第三方提供对通信平台的访问权，使得第三方可以定义、配置、以及监控本地托管于通信平台上的自定义垂直解决方案。通信平台提供实现用于和外部设备通信的设备独立通信功能的通信管理器。因此，通信平台连接到、与之通信、以及控制几乎任何外部设备。



1. 一种机器到机器通信平台,包括:
和在所述机器到机器通信平台中托管和管理的垂直解决方案通信的垂直解决方案网关;
和所述垂直解决方案界面通信的协调层;
和所述协调层通信的第三方网关,所述第三方网关可操作以提供针对所述垂直解决方案的垂直解决方案管理的远程第三方访问;
和所述协调层通信的通信管理器,所述通信管理器为了和外部设备通信而实现设备独立通信能力;以及
所述通信管理器和所述外部设备之间的网络接口。
2. 权利要求 1 的通信平台,其中所述通信管理器包括用于所述外部设备的通信适配器。
3. 权利要求 2 的通信平台,其中所述通信适配器可操作为将从所述外部设备接收的数据转换为所述通信平台中使用的标准格式消息。
4. 权利要求 3 的通信平台,其中所述通信适配器进一步可操作为将所述标准格式消息递送给所述垂直解决方案。
5. 权利要求 1 的通信平台,进一步包括:
和所述协调层通信的事件处理器,所述事件处理器可操作为接收由所述外部设备生成的事件。
6. 权利要求 5 的通信平台,其中所述事件处理器包括事件规则储存库和事件生成器。
7. 权利要求 6 的通信平台,其中所述事件处理器进一步包括通过所述第三方网关可访问的事件管理器,其用于为第三方创建在所述通信平台中托管的自定义垂直解决方案。
8. 一种用于机器到机器通信的方法,包括:
实现可操作以和在所述机器到机器通信平台中托管和管理的垂直解决方案通信的垂直解决方案网关;
实现和所述垂直解决方案界面通信的协调层;
实现和所述协调层通信的第三方网关,以及通过所述第三方网关来提供针对所述垂直解决方案的垂直解决方案管理的远程第三方访问;
实现和所述协调层通信的通信管理器,以为了和外部设备通信而提供设备独立通信能力;以及
实现所述通信管理器和所述外部设备之间的网络接口。
9. 权利要求 8 的方法,进一步包括:在所述通信管理器中安装用于所述外部设备的通信适配器。
10. 权利要求 9 的方法,进一步包括:使用所述通信适配器将从所述外部设备接收的数据转换为所述通信平台中使用的标准格式消息。
11. 权利要求 10 的方法,进一步包括:使用所述通信适配器将所述标准格式消息递送给所述垂直解决方案。
12. 权利要求 8 的方法,进一步包括:
实现和所述协调层通信的事件处理器,所述事件处理器可操作以接收所述外部设备生成的事件。

13. 权利要求 12 的方法,进一步包括:在所述事件处理器中实现事件规则储存库和事件生成器。

14. 权利要求 13 的方法,进一步包括:在所述事件处理器中实现通过所述第三方网关可访问的事件管理器,其用于为第三方创建在所述通信平台中托管的自定义垂直解决方案。

15. 一种包括机器可读介质以及在该介质上的逻辑的产品,当其由处理器执行时使得所述处理器:

实现可操作以和在机器到机器通信平台中托管和管理的垂直解决方案通信的垂直解决方案网关;

实现和所述垂直解决方案界面通信的协调层;

实现和所述协调层通信的第三方网关,以及通过所述第三方网关来提供针对所述垂直解决方案的垂直解决方案管理的远程第三方访问;

实现和所述协调层通信的通信管理器,以为了和外部设备通信而提供设备独立通信能力;以及

实现所述通信管理器和所述外部设备之间的网络接口。

16. 权利要求 15 的产品,其中当执行时所述逻辑进一步使得所述处理器在所述通信管理器中安装用于所述外部设备的通信适配器。

17. 权利要求 16 的产品,其中当执行时所述逻辑进一步使得所述处理器使用所述通信适配器将从所述外部设备接收的数据转换为所述通信平台中使用的标准格式消息。

18. 权利要求 17 的产品,其中当执行时所述逻辑进一步使得所述处理器使用所述通信适配器将所述标准格式消息递送给所述垂直解决方案。

19. 权利要求 15 的产品,其中当执行时所述逻辑进一步使得所述处理器实现和所述协调层通信的事件处理器,所述事件处理器可操作以接收由所述外部设备生成的事件。

20. 权利要求 19 的产品,其中当执行时所述逻辑进一步使得所述处理器在所述事件处理器中实现事件规则储存库和事件生成器。

机器到机器架构

技术领域

[0001] 本申请涉及服务供给和设备管理平台。具体地,本申请涉及提供用于远程托管服务的平台,该平台使得服务可以管理和监控不同类型设备的庞大阵列,以及收集和跟踪发送到设备或来自设备的数据。

背景技术

[0002] 电信技术的快速发展已经使得各式各样的服务提供者可以远程管理和控制设备。尽管设备的远程控制和管理具有其优势,但服务提供者不断地开发需要对新类型的服务进行控制和管理的新服务,以企图在市场中保持竞争力。过去,用于管理和控制设备的平台是专有的并且一旦被开发,该平台便只能支持其最初考虑的设备 and / 或服务。如果服务提供者希望为新的设备或服务增加支持,则服务提供者需要耗费可观的资源,以更改已有的平台或开发支持新设备 / 服务的新平台。由于市场不断地针对更新的和更多类型的设备需要更多的服务和支持,这一情况变得严重。服务提供者被迫频繁更改已有的平台或开发支持新设备和服务的新平台,这是非常耗时和昂贵的。实现一个为几乎任何类型的服务和设备提供支持的灵活平台是一项艰难的技术挑战。

发明内容

[0003] 一种机器到机器通信平台为当连接到几乎任何外部设备时以灵活,开放的方式实现本地管理的垂直服务这一技术问题提供技术解决方案。通信平台不仅限于和有限类型的设备的互操作。相反,通信平台提供实现用于连接到,和其通信,以及控制几乎任何外部设备的设备独立通信功能的通信管理器。进一步地,通信平台包括规则引擎,通过该引擎,通信平台托管和管理与外部设备交互的第三方垂直解决方案。第三方网关向第三方提供对通信平台的访问权。因此,第三方可以定义,配置和监控在通信平台中本地托管的自定义垂直解决方案。

[0004] 其他系统,方法,特征和优势是,或将是,对于本领域技术人员而言基于以下附图和详细描述检查是显而易见的。所有此类附加的系统,方法,特征和优势都旨在包含在本说明书中,在发明的范围之内,并且由如下的权利要求所保护。

附图说明

[0005] 该系统可以参考如下的附图和说明得到更好的理解。附图中的组件不一定按比例绘制,强调而是被置于图解发明的原理。此外,在图中,相同的附图标记在不同的视图中表示相应的部分。

[0006] 图 1 示出了机器到机器架构 (“架构”) 的总览。

[0007] 图 2 示出了机器到机器平台的实施方式。

[0008] 图 3 示出了该架构的各种特征的总览。

[0009] 图 4 示出了通信管理器实现的设备消息收发和递送模型。

- [0010] 图 5 示出了事件处理器的高层架构。
- [0011] 图 6A 示出了设备管理组件的实现。
- [0012] 图 6B 示出了描述用于设备的 FOTA 更新循环的通用步骤的示例。
- [0013] 图 7 示出了用于设备供应功能的插入设备过程的逻辑流程图。
- [0014] 图 8 示出了用于设备供应功能的移除设备过程的逻辑流程图。
- [0015] 图 9 示出了用于服务配置功能的激活过程的逻辑流程图。
- [0016] 图 10 示出了用于服务配置功能的去激活过程的逻辑流程图。
- [0017] 图 11 示出了用于服务配置功能的更新设备配置过程的逻辑流程图。
- [0018] 图 12 示出了用于服务配置功能的交换服务 / 替代设备过程的逻辑流程图。
- [0019] 图 13 示出了用于服务配置功能的配置请求 (1) 从 SP 到设备过程的逻辑流程图。
- [0020] 图 14 示出了用于服务配置功能的配置请求 (2) 从设备到 SP 过程的逻辑流程图。
- [0021] 图 15 示出了用于服务提供者订阅功能的订阅过程的逻辑流程图。
- [0022] 图 16 示出了用于服务提供者订阅功能的取消订阅过程的逻辑流程图。
- [0023] 图 17 示出了用于应用通信功能的事件分发过程的逻辑流程图。
- [0024] 图 18 示出了用于应用通信功能的应用请求过程的逻辑流程图。
- [0025] 图 19 示出了设备管理过程的逻辑流程图。
- [0026] 图 20 示出了设备监控过程的逻辑流程图。
- [0027] 图 21 示出了数据采集表的示例。
- [0028] 图 22 示出了数据访问网络服务所用的数据访问网络服务数据模型的示例。
- [0029] 图 23 示出了设备管理网络服务数据模型的示例。
- [0030] 图 24 示出了事件处理器网络服务数据模型的示例。
- [0031] 图 25 示出了系统配置网络服务数据模型的示例。
- [0032] 图 26 示出了用于架构 100 的示例网络控制台界面 2600。
- [0033] 图 27 示出了具有示例树视图对象的网络控制台界面。
- [0034] 图 28 示出了具有树视图对象的网络控制台界面,其中该树视图对象展开以列出项目。
- [0035] 图 29 示出了具有树视图对象的网络控制台界面,其中该树视图对象展开以显示属于项目的站点。
- [0036] 图 30 示出了具有树视图对象的网络控制台界面,其中该树视图对象展开以显示特别与选定站点相关的设备。
- [0037] 图 31 示出了具有树视图对象的网络控制台界面,其中该树视图对象展开以显示针对选定设备的详细状态信息连接。
- [0038] 图 32 示出了具有数据日志下载按钮的网络控制台界面。
- [0039] 图 33 示出了订户身份模块 (SIM) 详细状态窗口。
- [0040] 图 34 示出了应用模块详细状态窗口。
- [0041] 图 35 示出了资源方位的地理显示。
- [0042] 图 36 示出了资源状态窗口的示例。
- [0043] 图 37 示出了状态显示页面的示例。
- [0044] 图 38 示出了搜索结果界面的示例。

- [0045] 图 39 示出了高级搜索界面的示例。
- [0046] 图 40 示出了定期报告界面的示例。
- [0047] 图 41 示出了增加了新定义报告的定期报告界面的示例。
- [0048] 图 42 示出了实时报告界面的示例。
- [0049] 图 43 示出了管理界面的示例。
- [0050] 图 44 示出了由设备类型选择引起的管理页面的示例。
- [0051] 图 45 示出了具有显示提供给客户的服务的树视图对象的管理页面。
- [0052] 图 46 示出了详细调度页面。
- [0053] 图 47 示出了分布式调度页面。
- [0054] 图 48 示出了警告界面的示例。
- [0055] 图 49 示出了警告配置界面。

具体实施方式

[0056] 图 1 示出了机器到机器架构 (“架构”) 100 的总览。在架构 100 中, 机器到机器平台 (“平台”) 102 通过外部网关 106 和各种各样的设备 104 通信。平台 102 也可以直接和设备 104 通信 (例如, 不通过网关通信)。平台 102 可以托管各种不同的服务, 这些服务使用 and 平台 102 通信的设备 106。服务可以包括在平台 102 中为第三方 108 实现和托管的自定义服务。第三方 108 可以通过网络 110 访问平台 102。网络 110 可以是任意类型的计算机或电信网络, 例如因特网。用户也可以通过网络 110 访问平台 102, 或者用户也可以本地访问平台 102。通过网络 110, 第三方 108 和用户 112 可以在世界范围内访问平台 102。设备 106 可以包括能和平台 102 通信的任何类型的设备 / 传感器 / 设备 / 系统, 例如蜂窝电话, 寻呼机, 汽车引擎控制单元, 轮胎压力监控系统, 医疗监控系统, 安全系统, 摄像机, 照相机, 或其他设备。

[0057] 图 2 示出了机器到机器平台 102 的实施方式 200。平台 102 主要划分为两个部分: 垂直解决方案部分 202 和机器到机器平台使能器部分 204。垂直解决方案 202 是由平台 102 托管的应用或服务。垂直解决方案 202 可以定义, 查看, 和 / 或管理用于分发采样数据的规则, 该采样数据从设备 106、基于和其相关的规则发送而来。垂直解决方案 202 提供的服务可以包括, 例如, 车辆跟踪 206, 远程医疗诊断 208, 遥测 210, IP 监控 212, 个人安全 214, 或任何其他服务提供者可以实现的服务。平台 102 为快速创建, 递送和管理新的垂直解决方案 202 提供所需的能力。平台 102 包括四个核心功能: 1) 设备管理, 2) 设备监控, 3) 数据采集和 4) 记账。

[0058] 设备管理是用于允许远程管理设备 106 的一组技术, 协议和标准。用户 112 或第三方 108 可以使用设备管理功能、利用到平台 102 的有线或无线接口来更新设备固件或操作系统, 安装应用, 或者修正错误。设备管理功能允许设备 106 的配置, 包括作为示例, 为第一次使用配置设备, 启用设备的特征, 以及禁用这些特征。设备管理功能还允许改变设备 106 的设置和参数, 将新软件和 / 或错误修正装载在设备上, 以及提供设备的固件升级和生命周期管理。设备管理功能可以通过允许第三方执行设备管理操作的网络服务接口或门户网站来提供。

[0059] 设备管理功能和储存库 220 之一 (即, 设备储存库 224) 联合工作, 其包括针对每

个特定设备 106 的可配置设置和配置 XML,用于配置设备 106。设备储存库 224 管理特定的可配置 XML 和开放移动联盟 (OMA) 配置 XML。设备储存库 224 也能够存储用于升级设备固件或软件的更新包。

[0060] 设备监控功能提供设备诊断和错误管理功能。诊断和错误管理不仅仅指硬件问题,还包括其他设备 106 的其他元件,例如 SIM 卡,设备硬件,以及应用模块。设备监控功能的设备诊断组件能够分析日志文件,设备 106 在网络上的存在以及关于设备状态的数据。错误管理组件估计评价预定规则并根据规则实现动作。可能的动作的示例为:发送短消息服务消息 (SMS) 或电子邮件,以向用户 112、第三方服务提供者 108 或其他有关的实体通知事件或告警告;触发工作流;生成关于设备 106 状态的实时报告;或其他可以由预定规则定义的动作。设备监控功能的错误管理组件使得用户 112 或第三方服务提供者 108 能动态配置规则,以及联合警告系统 234 运作的动作。警告系统 234 是平台 102 的组件,其向用户 112 或第三方服务提供者 108、通过 SMS,电子邮件,和 / 或网络警告通知警告。用户 112 或第三方 108 可以从门户网站显示报告和警告。报告可以是图表报告,仪表盘,或其他报告。由报告显示的信息的某些示例为:设备 106 在错误状态的次数 / 百分比和设备 106 在待机状态的次数 / 百分比。

[0061] 平台 102 还通过提供用于取得从设备 106 接收的数据并将数据分发到第三方服务提供者 108 或平台 102 的处理组件,或者用于在数据储存库 222 中存储数据的功能来提供数据采集功能。数据储存库 222 可以存储从设备 106 接收的详细数据。存储数据类型的某些示例是关于设备 106 状态的数据和告警。服务操作数据和 / 或图像 / 视频数据被分发给第三方 108,垂直解决方案 202 被存储在一组储存库 220 中的其特定储存库中。

[0062] 平台 102 的组件可以追踪多种类型的使用以支持记账功能。记账功能可以由流业务过程管理器来提供。记账功能也可以通过和第三方记账系统集成来提供。跟踪使用的示例包括服务的使用,发送 SMS 的数量,通知事件的数量以及生成的数据业务量。平台 102 可以和计量系统集成来为记账功能提供支持。计量系统可以是来自可通过远程方法调用 (RMI) 或简单对象访问协议 (SOAP) 调用的平台 102 独立的组件。

[0063] 机器到机器平台 102 可以包括实现在前面段落描述的功能的多个逻辑组件。垂直解决方案网关 216 和协调器和业务过程管理器 238 通信以允许在平台 102 剩余部分和垂直解决方案 202 之间的通信。

[0064] 第三方网关组件 218 提供功能以实现入站和出站请求授权,认证标准管理以及请求分发。该组件提供使得垂直解决方案 202 和第三方 108(和他们的客户)能够执行平台 102 的所有功能(例如,订阅 / 退订平台 102 上的设备,服务激活 / 去激活,设备配置,数据重试,设备状态检查,以及计量)的网络服务和 API。网络服务和 API 基于应用领域进行分类。在平台 102 上托管的每个服务可以和应用领域相关。应用领域可以和很多服务相关。将在下面进一步详细描述应用领域。

[0065] 储存库 220 可以包括用于存储用于平台 102 多个组件的数据或配置信息的多个储存库。储存库 220 可以实现为具有多个表的单个数据库,或者可以实现为用于具体功能的多个单独数据库。储存库 220 的示例包括数据储存库 222,设备储存库 224 以及事件储存库 226。数据储存库 222 管理平台 102 的资产,例如设备 106,其 SIM 卡和其配置。数据储存库 222 还支持订阅和供给功能。

[0066] 通用服务组件 228 的组件提供可以为托管多种服务而共享的功能。通用服务组件 228 可以包括位置组件 230, 事件处理器 232, 警告系统 234 和设备管理组件 236。

[0067] 协调器和业务过程管理器 (过程管理器) 238 直接或间接地和平台 102 的所有组件通信以协调平台 102 的功能。通信管理器 240 接收来自设备 106 的消息并将其转换为平台的多个组件可以理解的消息。通信管理器 240 的功能可以在外部网关 104 中实现, 设备 106 可以连接到该网关。

[0068] 网络使能器 256 可以和通信管理器 240 通信, 以使得平台 102 能够连接到电信网或平台 102 在其上和设备 106 通信的其他网络。网络使能器 256 是负责使能在平台 102 和电信网之间使用多种通信标准 (例如 SMSC, GGSN, LEA 和 Mail) 来进行通信的组件。

[0069] 平台 102 还可以和计量系统 258 集成以用于跟踪使用数据, 然而门户网站 260 和 262 使得第三方用户和管理员分别能够访问由平台 102 提供的功能。每个门户网站 260 和 262 包括两个不同的区域。管理区域允许第三方管理员管理和维持第三方项目, 设备 106 和数据。项目是设备 106 的逻辑组。门户网站 260 和 262 的设备管理区域允许第三方用户上传新的固件或增量固件, 可视化设备 106 的实际配置, 并通过直接输入值或上传新 XML 配置文件来为配置参数设置新值。

[0070] 通信管理器 240 是负责在设备 106 和平台 102 之间, 以及在外部网关 104 和平台 102 之间通信的组件。通信管理器 240 使得平台 102 能够使用独立于特定设备并特定于应用领域的通用语言和设备 106 通信。在一个实施方式中, 这样的通用语言可以是一个版本的 XML, 一种机器到机器 (M2M) XML 格式。翻译到 M2M XML 格式以及从 M2M XML 格式翻译是使用样式表 (Stylesheet) 来实现的。使用样式表, 通信管理器 240 可以将来自设备 106 接收到的消息翻译为 M2M XML 格式, 并将来自平台 102 的 M2M XML 消息翻译为接受设备 106 可以理解的消息。对于新设备的支持可以通过将新的样式表插入储存库 220 来实现。

[0071] 设备管理组件 236 是通用服务组件 228 之一。设备管理组件 236 实现以上讨论的设备管理功能。该组件负责设备管理功能, 包括实现任意所需的用于和设备 106 通信的服务器标准 (例如, 开放移动联盟设备管理, OMA DM)。

[0072] 设备监控组件 264, 作为垂直解决方案 202 之一, 实现上面描述的所有设备监控功能或者其一部分。该组件提供业务逻辑和功能, 以分析来自设备 106 的日志错误, 设备的存在以及关于设备状态的数据。错误管理逻辑可以利用事件处理器 232 并将告警发送给警告系统 234, 并实时生成关于设备状态 106 状态的报告。第三方服务提供者用户可以访问监控网络控制台 266, 以监控和控制可选数据, 将任何参数作为趋势图浏览, 以及配置, 可视化和处理告警。

[0073] 事件处理器 232 是通用服务 228 之一, 其管理设备 106 发送到平台 102 的事件 / 告警 (外部事件) 或由平台 102 生成的事件 / 告警 (内部事件)。事件处理器 232 和规则储存库 (其可以是储存库 220 之一) 联合工作, 以将触发器发送到垂直解决方案 202。警告系统 234 将警告通过 SMS, 电子邮件或网络警告通知给用户 112 和 / 或第三方 108。位置组件 230 提供地理信息系统功能, 例如浏览地图, 地理编码, 以及追踪。该组件还在监控网络控制台 266 的监控区域上提供表明项目和设备 106 方位的地图。

[0074] 计量系统 258, 门户网站 260 和 262, 以及网络控制台 266 可以在平台 102, 和平台 102 集成的独立系统, 或者独立第三方解决方案中实现。

[0075] 图 3 从功能视角 300 示出了平台 102。图 3 中表示的功能在本文档的其他角度更详细进行描述。

[0076] 图 4 示出了通信管理器 240 实现的消息收发和递送模型 400。通信管理器 240 可以支持用于在平台 102 和设备 106 之间通信的某一通信模型,以及管理任何技术的设备 106。通信管理器 240 创建能够从行为和物理视角描述设备 106 的某个抽象程度,以获取所有设备 106 独立于其潜在技术的一致视角。通信管理器 240 包括适配器管理器 242, CIM 映射器 244, 以及应用管理器 246。适配器管理器 242 管理适配器 402。一个专用于处理从设备接收的数据 404 的适配器 402 将数据转换为通用信息模块 (CIM) 设备消息 406。适配器 42 还能够将 CIM 设备消息 406 转换为设备消息 404。CIM 映射器 244 可以将 CIM 设备消息 406 变换为 CIM 服务消息 408, 反之亦然。应用管理器 246 取得 CIM 服务消息 408 并准备即将递送给对应应用的 CIM 服务消息 410。

[0077] 可以提供在平台 102 和设备 106 之间实现协议的通信模型。可以管理特定的库来为何通信模型不兼容的设备提供支持。

[0078] 在该通信模型的一个实施方式中,发送到和来自设备 106 和平台 102 的消息可以基于其目的来分类。表 1 示出了消息类型和由每个消息类型标识的呼叫者和接受者。

[0079]

表 1		
消息类型	呼叫者	接受者
配置命令 (ConfigurationCommand)	平台	设备
配置查询 (ConfigurationQuery)	平台	设备
配置查询 (ConfigurationQuery)	设备	平台
事件 (Event)	设备	平台
应用查询 (ApplicationQuery)	平台	设备
确认 (Acknowledge)	平台	设备
确认 (Acknowledge)	设备	平台

[0080] 消息类型配置命令的消息可以用于设定或更新设备 106 的配置参数值。取决于应用领域,该消息类型的消息可以有效地变成应用命令。例如在家庭自动化情况下,为了在公寓中关掉所有的灯,平台 102 可以发送针对参数“灯的状态”的配置命令和值“OFF ALL”。这一消息将关闭公寓灯的命令给到家庭自动化设备。

[0081] 消息类型配置查询的消息可由平台 102 使用来请求当前设备配置或反之亦然。

[0082] 消息类型事件的消息可在设备 106 触发条件并将有关信息发送到平台 106 时使用。

[0083] 消息类型应用查询的消息可以由平台 102 使用来向设备 106 执行数据请求。请求数据的类型或内容取决于应用领域的定义。

[0084] 消息类型确认的消息可以用于进行之前的通信。

[0085] 应用领域表示基于应用环境所有权的消息分类。换言之,应用领域标识一个消息族。以这样的方式,可以只授权向那些订阅了事件所属应用领域的第三方 108 分发来自设备 106 事件。例如,监控服务提供者应该在接收监控事件之前订阅监控应用领域。其他应用领域的例子有:汽车追踪,安全,家庭自动化,自动贩卖机管理,远程计量 (AMR/AMI, 远程监控和控制), 远程健康诊断和配置。领域配置是用于配置目的的通用域。然而,仅特定应用领域需要某些配置动作。

[0086] 协议数据单元 (PDU) 可以是通信模型的一部分,其可以用于平台 102 暴露的、通往设备 106 的接口,并且反之亦然。每个请求 / 响应消息可以包括 PDU。PDU 可以定义:

[0087] 请求:设备 106 或平台 102 所提供的、发起到接受者的通信的参数集。有如下解释的多种类型请求,以管理同步和异步通信情景。

[0088] 响应:之前同步请求的接受者所提供的参数集。

[0089] 表 2 示出了用于“请求”和“响应”的参数示例。

[0090]

表 2	
请求 (Request)	响应 (Response)
请求类型 (requestType)	响应状态 (responseStatus)
呼叫者 (caller)	会话 ID (conversationID)
时间戳 (Timestamp)	有效载荷 (Payload)
优先级 (Priority)	协议版本 (protocolVersion)
有效期 (Expires)	
接受者 (Recipient)	
会话 ID (conversationID)	
有效载荷 (Payload)	
协议版本 (protocolVersion)	

[0091] 如下是用于参数“请求类型”可能取值的示例:

[0092] 获取 (GET):平台 102 或设备 106 使用来以同步或异步模式向接受者请求资源。如果物理信道不支持双向信道,通信被强制为异步。在一些情况下,如果异步模式不适合于业务过程,可以假设不需要响应。

[0093] 发送 (POST):平台 102 或设备 106 使用来设置或更新接受者状态。

[0094] 回复 (Replay):在异步情景中使用来传达之前请求的响应。

[0095] 通知 (Notify):在呼叫者想要通知资源可用性时使用。通常,“通知”将跟随 GET 请求以便获得资源。

[0096] 通信模型的物理信道可以是 SMS 或 GPRS 上的 HTTP/HTTPS。SMS 通常由平台 102

在平台需要发起和设备 16 的通信时使用。SMS 仅支持异步通信。GPRS 上的 HTTP/HTTPS 由设备 106 使用来传达触发的事件,也还由平台 102 使用来基于之前的请求发送响应到设备。

[0097] 表 3 示出了 PDU,呼叫者接受者,消息类型 (messageType),以及物理信道之间可能的结合的示例。

[0098]

PDU	呼叫者 (Caller)	消息类型 (MessageType)	物理信道 (PhysicalChannel)	接受者 (Recipient)
异步.发送 (Async.POST)	平台	配置命令 (ConfigurationCommand)	SMS	设备
异步.获取 (Async.GET)	平台	配置查询 (ConfigurationQuery)	SMS	设备
异步.获取 (Async.GET)	平台	应用查询 (ApplicationQuery)	SMS	设备
同步.响应 (Sync.Resonse)	平台	确认 (Acknowledge)	HTTPs	设备
异步.通知 (Async.Notify)	平台	配置命令 (ConfigurationCommand)	SMS	设备
同步.回复 (Sync.Replay)	设备	确认 (Acknowledge)	HTTPs	平台
同步./异步 获取 (Sync./Async GET)	设备	配置查询 (ConfigurationQuery)	HTTPs	平台
同步.发送 (Sync.POST)	设备	事件 (Event)	HTTPs	平台
同步.回复 (Sync.Reply)	设备	事件 (Event)	HTTPs	平台

[0099]

[0100] 返回参考图 4,基于应用环境所有权,将用于递送的 CIM 服务消息 410 聚集为应用领域。对应于消息所属应用领域的 CIM 处理相应的消息。

[0101] 图 5 示出了事件处理器 232 的高级架构 500。事件处理器 232 管理由设备 106 发送到平台 102 的事件 / 告警或由平台创建的事件 / 告警。事件处理器 232 包括事件生成器 502,事件管理器 504,事件储存库 226,规则储存库 506,数据储存库 222,规则编辑器 508,以及事件管理前端 510。

[0102] 事件生成器 502 组件的任务是,通过预定义的规则和查询可用信息的选用的资源(在数据库中,来自网络服务的响应,以及其他资源),验证是否生成告警。

[0103] 事件管理器 504 的作用是管理事件的生命周期。事件储存库 226 存储所有事件,即使是那些已经关闭的。规则储存库 506 存储所有默认规则。数据储存库 222 可以用作数据处理器 232 在规则储存库 506 中包括的规则的评定过程中可参考的数据词典。这些数据可以同样地是数据库中的数据,针对平台内或外的网络服务调用的响应等。规则编辑器 508 是接口,平台 102 使其可用于定义新的规则和管理已有的规则。规则编辑器 508,通常由商业 BRMS 提供,为规则储存库 224 提供用于定义和填充的接口。通过在事件处理器 232 中由平台 102 使得接口(网络服务)可用,事件管理前端 510 给第三方服务提供者用户提供详细查看所有事件的能力和管理事件的生命周期(创建新事件,关闭事件,以及生命周期的其它部分)的可能性。例如,事件管理器前端 510 的一部分可以是对于控制间的操作者可用于查看事件细节和要求关闭系统安全已经验证的事件的图形化用户接口(GUI)。

[0104] 作为示例,事件可以包括如下的属性:

[0105] 事件标识(event id)

[0106] 事件类型

[0107] 事件状态

[0108] 创建方法(内部/外部)

[0109] 事件创建时间戳

[0110] 最后状态改变时间戳

[0111] 上级事件 id(可选)

[0112] 事件拥有者标识(组件/操作符)

[0113] 事件参数

[0114] 事件管理器 232 的主要目标是处理事件:一旦创建一个事件,确定采取什么动作以及以怎样的顺序(事件-反应 workflow)。可基于事件的属性,特别是参考事件类型和按照伴随其的参数(事件参数)具有的特定值,来采取事件反应。响应于事件执行的动作 512 可包括发送电子邮件,发送 SMS,调用网络服务,以及创建下级事件。

[0115] 图 6A 示出了设备管理组件 236 的实现 600a。设备管理组件 236 实现设备管理功能。对于设备 106,移动电话 106a 可以区别于其他设备 106 由设备管理组件 236 来管理。可以提供管理移动电话 106a 的移动管理功能来用于自己的集成解决方案,例如 NOKIA™Siemens™ 设备管理解决方案或 Afaris™ Sybase™ 设备管理解决方案。可以为除移动电话 106a 以外的设备 106 实现用于设备管理功能的自定义解决方案。然而,自定义解决方案也可以为移动电话 106a 提供管理功能。

[0116] 实现自定义解决方案的设备管理组件 236 支持空中固件(Firmware Over-The-Air)过程。该解决方案也兼容 OMA 标准。FOTA 是允许嵌入式软件在任意地点和任意时间无线更新的过程。固件包括所有使能设备操作的软件。与常规软件不同,固件不能由使用设备的终端用户定制。

[0117] 设备管理组件 236 进一步包括 OMA-DM 组件 602, OMA-FUMO 组件 604,增量生成器 606 和更新代理 608。OMA-DM602, OMA-FUMO(开放移动联盟-固件更新管理对象)604 和增量生成器 606 组件驻留在平台 102 上。OMA-DM 组件 602 是实现 OMADM 1.2 标准的组件。

OMA DM 1.2 标准是用于空中管理所有小设备的设备管理协议。OMA-FUMO 组件 604 是实现用于更新固件的 OMAFUMO 1.0 标准的组件。增量生成器 606 可以是比较两个固件版本并产生仅包括两个固件版本之间区别的增量包的软件。更新代理 608 是驻留在设备 106 的固件之中的客户端组件,并将实际更新用于固件。增量生成器 606 和更新代理 608 可以对于每个设备专有,但兼容 OMA 标准。还可以在设备 106 上提供用于和驻留在平台 102 上的设备管理组件 236 的组件通信的 OMA-DM 客户端 610。可实现设备管理组件 236 的所有组件以支持 linux 操作系统 (OS)。

[0118] FOTA 更新周期通常可伴随一个四阶段的过程:1) 生成固件更新(增量包);2) 将更新发送和提交到固件管理服务器;3) 将固件下载到设备 106;以及(4) 由更新代理 608 更新设备。固件管理服务器可以是 OMA-FUMO 组件 604。设备管理组件 236 可以实现阶段 2, 3 和 4 以支持 linux OS。设备管理 236 还可以集成专有解决方案以实现对所有阶段 1-4 的支持。

[0119] 图 6B 示出了描述用于设备 106 的 FOTA 更新周期的通用过程的示例 600b。外部系统或平台 102 向设备管理组件 236 发送指令以发起固件更新过程(612)。设备管理组件 236 向设备 106 发送发起信号(614),且设备以设备信息响应来用于验证(616)。此后,设备管理组件 236 发送固件更新包和指令来将固件更新到 OMA-DM 客户端 610(618,620)。一旦用新的固件更新了设备 106,设备将最终通知发送到设备管理组件 236(622)。备选地,设备 106 可以发起并发送用于更新包的请求给设备管理组件 236(624)。作为响应,设备管理组件 236 可以将更新包发送到设备管理组件 236(626)。此后,设备 106 向设备管理组件 236 发送最终通知(622)。

[0120] 设备监控组件 264 可以实现上述设备监控功能。为了避免/减少业务服务中断/失败或为了满足设备提供者 108 和用户 112 之间服务的等级协定,设备监控组件 264 验证设备 106 的状态,性能,故障情况和诊断分析。设备监控组件 264 使得设备故障能够通过电子邮件或 SMS,或通过显示故障设备列表的网络控制台 266,通知到服务提供者 108。网络控制台 266 可以在仪表盘上显示故障设备。网络控制台 266 可以是门户网站 260 或 262 的一部分。

[0121] 设备监控组件 264 可以支持故障设备 106 的问题排除。为此,设备监控组件 264 可以独立分析每个设备的元件,例如 SIM 卡,设备硬件和固件,以及应用模块。也可以单独监控设备 106 的其他独立元件。

[0122] 由设备监控组件 264 监控的信息可以包括状态,例如 OK, KO, ON, OFF, 待机, 激活, 未激活, 或未定义。也可以监控网络信道使用,网络存在,工作状态,最后配置更新日期,以及最后重置(重启)日期。可以监控对所有通信接口的网络信道使用,包括 SMS 和 GPRS。网络存在指示 SIM 在 MNO 网(例如 GSM 或 GPRS)上的存在。工作状态指示设备是否配有发现关键环境状况的感测器,环境条件例如温度,湿度,或低电源或其他条件。监控的信息也可以包括性能信息例如平均 CPU 使用和存储器使用。

[0123] 设备监控组件 264 还可以集合和报告监控的信息的可视化。用户 108 或第三方服务提供者 108 可以用状态信息聚合其拥有的所有设备 106 或其子集。用户 112 或第三方服务提供者 108 还可以按客户, VAS, 项目, 或站点聚合设备 106。

[0124] 为了管理服务等级,设备监控组件 264 可以进一步提供 KPI 管理和 SLA 配置,例如

设备性能的实时图形化表示和故障状况的跟踪。组件 264 进一步支持轮询,捕获和实时请求机制。轮询指由平台 102 实现向设备 106 的定期查询。捕获指基于预定义条件,配置或规则,通过来自设备 106 的消息触发事件。事件还可以由设备 106 触发。实时请求指用户 112 或第三方服务提供者 108 对所有监控的实体进行实时监控请求。实时请求可以通过门户网站 260 或 262 实现。监控的实体可以包括任何设备 106。

[0125] 设备监控组件 264 还可以通过 SMS 或电子邮件提供预定义事件的消息通知。设备 264 可以进一步提供监控设备的高级业务视图。

[0126] 图 7-20 示出了由协调器和业务过程管理器(过程管理器)238 实现的过程和动作的工作流。过程管理器 238 可以直接或间接和平台 102 的所有其他组件通信以提供平台的功能。

[0127] 图 7 示出了用于上述设备供应功能的插入设备过程的逻辑流图 700。用户 112 或第三方服务提供者 108 可以请求插入设备动作,以将设备添加到和平台 102 通信的设备 106 之一(702)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据:设备 ID(deviceID),制造者(Manufacture),OEM 编码(OEM code),MSISDNsim,ICCID sim。平台 102 接收请求并将该请求转发到过程管理器 238(704)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。请求可以使用插入设备(insertDevice)接口转发。如果提供所有所需的参数,过程管理器 238 确定设备 ID 是否存在于设备储存库 224 中(706)。如果不能提供所有所需的参数,过程管理器 238 通知错误(708)并且插入设备过程结束(714)。在步骤 706,如果设备 ID 不存在于设备储存库 224 中,在设备储存库 224 中创建新记录。如果设备 ID 已经存在于设备储存库 224 中,使用请求提供的新信息来更新设备储存库 224 中相应于设备 ID 的已有记录(712),以及过程结束(714)。

[0128] 图 8 示出了用于设备供应功能的移除设备过程的逻辑流图 800。用户 112 或第三方服务提供者 108 可以请求移除设备动作来将设备添加到和平台 102 通信的设备 106 之一(802)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据:设备 ID。平台 102 接收请求并将该请求转发给过程管理器 238(804)。平台 102 可以通过第三方网关 238,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。可以使用移除设备(removeDevice)接口转发请求。如果提供所需参数,过程管理器 238 确定设备 ID 是否存在于设备储存库 224 中(806)。如果未提供所需参数,过程管理器 238 通知错误(808)并且过程结束(814)。在步骤 806,如果设备 ID 存在于设备储存库 224 中,过程管理器 238 确定对于和设备 ID 相应的设备而言是否有任何激活的服务(810)。如果没有活动的服务,从设备储存库 224 中移除对应于该设备 ID 的记录(812)并且过程结束(814)。如果对于和设备 ID 对应的设备而言有活动服务,过程管理器 238 通知错误(808)并且过程结束(814)。

[0129] 图 9 示出了用于服务配置功能的激活过程的逻辑流图 900。用户 112 或第三方服务提供者 108 可以请求服务激活动作来将设备 106 配置为使得对于该设备服务被激活(902)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据:设备 ID,服务 Id(serviceID),配置参数(configuration parameters)。平台 102 接收请求并向过程管理器 238 转发请求(904)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。请求可以使用激活服务(activateService)接

口转发:激活服务(设备 ID,服务 Id,会议(conf))。如果提供所有所需的参数,过程管理器 238 确定设备 ID 是否存在于设备储存库 224 中(906)。如果未提供所需参数,过程管理器 238 通知错误(908)并且过程结束(918)。在步骤 906,如果设备 ID 存在于设备储存库 224 中,过程管理器 238 确定服务 Id 是否存在于设备储存库 224 中(910)。如果服务 ID 不存在,过程管理器 238 通知错误(910)并且过程结束(918)。如果存在服务 ID,则过程管理器 238 确定是否有配对服务 ID 和设备 ID 的记录存在于设备储存库 224 中(912)。如果在设备储存库 224 中已经存在这样的配对,过程管理器通知错误(908)并且过程结束(918)。如果在设备储存库 224 中不存在这样的配对,在设备储存库中创建配对服务 ID 和设备 ID 的新记录(914)。下一步,过程管理器 238 将配置消息发送到对应于设备 ID 的设备 106(916)。消息可以使用发送请求(sendRequest)同步接口进行发送,使用如下的参数:

[0130] 请求类型(RequestType):发送(POST)

[0131] 呼叫者(Caller):DCM

[0132] 有效载荷(Payload):

[0133] 领域(Domain):配置(Configuration)

[0134] 参数(Parameters):<配置参数(ConfigParam)数组(array of ConfigParam)>

[0135] 消息类型(MessageType):配置命令.激活(ConfigurationCommand.Activation)

[0136] 状态(Status):ON(对于所有和该服务 ID 相关的领域(for all domain associated to that serviceID))

[0137] 此后,过程结束(918)。

[0138] 图 10 示出了用于服务配置功能的去激活过程的逻辑流图 1000。用户 112 或第三方服务提供者 108 可以请求服务去激活动作,以配置设备 106 以使得对于该设备去激活服务(1002)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据:设备 ID,服务 Id。平台 102 接收请求并将请求转发给过程管理器 238(1004)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。请求可以使用去激活服务(deactivateService)接口转发:去激活服务(设备 ID,服务 Id)。如果提供所有所需的参数,过程管理器 238 确定在设备储存库 224 中是否存在设备 ID(1006)。如果未提供所需参数,过程管理器 238 通知错误(1008)并且过程结束(1018)。在步骤 1006 中,如果设备 ID 存在于设备储存库 224 中,过程管理器 238 确定服务 Id 是否存在于设备储存库 224 中(1010)。如果服务 ID 不存在,过程管理器 238 通知错误(1008)并且过程结束(1018)。如果服务 ID 存在,则过程管理器 238 确定是否在设备储存库 224 中存在配对服务 ID 和设备 ID 的记录。如果在设备储存库 224 中没有这样的配对,过程管理器通知错误(1008)并且过程结束(1018)。如果在设备储存库 224 中已经存在这样的配对,从设备储存库中移除该记录(1014)。下一步,过程管理器 238 向对应于设备 ID 的设备 106 发送配置消息(1016)。消息可以通过 SMS、使用发送请求(sendRequest)同步接口发送,使用如下参数:

[0139] 请求类型(RequestType):发送(POST)

[0140] 呼叫者(Caller):DCM

[0141] 有效载荷:

[0142] 领域:配置

[0143] 参数:空

[0144] 消息类型 (MessageType): 配置命令. 去激活 (ConfigurationCommand.Deactivation)

[0145] 状态:OFF(对于所有和该服务 ID 相关的领域)

[0146] 此后,过程结束 (1018)。

[0147] 图 11 示出了用于服务配置功能的更新设备配置过程的逻辑流程图 1100。用户 112 或第三方服务提供者 108 可以请求更新设备 106 的配置 (1102)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据:设备 ID,服务 Id,配置。平台 102 接收请求并将请求转发给过程管理器 238(1104)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。请求可以使用配置服务 (configureService) 接口转发:配置服务 (设备 ID,服务 Id,配置)。如果提供所有所需的参数,过程管理器 238 确定在设备储存库 224 中是否存在设备 ID(1106)。如果未提供所需参数,过程管理器 238 通知错误 (1108) 并且过程结束 (1118)。在步骤 1106 中,如果设备 ID 存在于设备储存库 224 中,过程管理器 238 确定服务 Id 是否存在于设备储存库 224 中 (1110)。如果服务 ID 不存在,过程管理器 238 通知错误 (1108) 并且过程结束 (1118)。在 (1110),如果服务 ID 不存在,过程管理器 238 通知错误 (1108) 并且过程结束 (1118)。如果服务 ID 存在,则过程管理器 238 确定是否在设备储存库 224 中存在配对服务 ID 和设备 ID 的记录 (1112)。如果在设备储存库 224 中没有这样的配对,过程管理器通知错误 (1108) 并且过程结束 (1118)。如果在设备储存库 224 中已经存在这样的配对,利用与该请求一起提供的信息来更新该记录 (1114)。下一步,过程管理器 238 向对应于设备 ID 的设备 106 发送配置消息 (1116)。消息可以通过 SMS、使用发送请求 (sendRequest) 同步接口发送,使用如下参数:

[0148] 请求类型 (RequestType):发送 (POST)

[0149] 呼叫者 (Caller):DCM

[0150] 有效载荷:

[0151] 领域:配置 (或应用,如果配置是在特定应用领域中的命令)

[0152] 参数:<配置参数 (ConfigParam) 数组>

[0153] 消息类型 (MessageType): 配置命令. 配置 (ConfigurationCommand.Configuration)

[0154] 状态:ON(对于所有和该服务 ID 相关的领域)

[0155] 此后,过程结束 (1118)。

[0156] 图 12 示出了用于服务配置功能的交换服务 / 替代设备过程的逻辑流程图 1200。用户 112 或第三方服务提供者 108 可以请求用新设备 106 来替代设备 106(1202)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据:旧设备 ID(oldDeviceID),新设备 ID(newDeviceID)。平台 102 接收请求并将请求转发到过程 238(1204)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。请求可以使用替代设备 (replaceDevice) 接口转发:替代设备 (旧设备 ID,新设备 Id)。如果提供所有所需的参数,过程管理器 238 确定在设备储存库 224 中是否存在旧或新设备 ID(1206)。如果未提供所需参数,过程管理器 238 通知错误 (1208) 并且过程结束 (1218)。

在步骤 1206 中,如果旧和新设备 ID 存在于设备储存库 224 中,过程管理器 238 确定新设备是否已经有任何激活的服务 (1210)。如果两种服务 ID 都不存在于设备储存库 224 中,过程管理器 238 通知错误 (1208) 并且过程结束 (1218)。在步骤 1210,如果对于新设备而言有服务被激活,过程管理器 238 通知错误 (1208) 并且过程结束 (1218)。如果对新设备而言没有服务被激活,则过程管理器 238 从设备储存库 224 获取和旧设备 ID 相关的服务和配置参数 (1212)。下一步,在设备储存库 224 中更新记录,以将记录从旧设备变成新设备 (1214)。下一步,过程管理器 238 向对应于设备 ID 的设备 106 发送配置消息 (1216)。消息可以通过 SMS、使用发送请求 (sendRequest) 同步接口发送,使用如下参数:

- [0157] 请求类型 (RequestType):发送 (POST)
- [0158] 呼叫者 (Caller):DCM
- [0159] 有效载荷:
- [0160] 领域:配置
- [0161] 参数:<配置参数 (ConfigParam) 数组>
- [0162] 消息类型 (MessageType):配置命令.配置 (ConfigurationCommand. Configuration)
- [0163] 状态:ON(对于所有和该服务 ID 相关的领域)
- [0164] 此后,过程结束 (1218)。

[0165] 图 13 示出了用于设备配置功能的配置请求 (1) 从 SP 到设备过程的逻辑流图 1300。用户 112 或第三方服务提供者 108 可以请求给定设备 106 的当前配置 (1302)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据:设备 ID。平台 102 接收请求并将请求转发给过程管理器 238 (1304)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。请求可以使用接收 SP 请求 (receiveSPRequest) 同步接口转发,使用如下参数:

- [0166] 有效载荷:
- [0167] 领域:<配置>
- [0168] 消息类型 (MessageType):配置查询 (ConfigurationQuery)
- [0169] 目标 (Target):设备 ID

[0170] 如果提供所有所需的参数,过程管理器 238 确定用户 112 或第三方服务提供者 108 是否被授权来请求配置 (1306)。如果未提供所需参数,过程管理器 238 通知错误 (1308) 并且过程结束 (1312)。在步骤 1306,如果用户 112 或第三方服务提供者 108 未被授权,过程管理器通知错误 (1308) 并且过程结束 (1318)。如果用户 112 或第三方服务提供者 108 被授权,过程管理器 238 向对应于设备 ID 的设备 106 发送配置查询 (ConfigurationQuery) 消息 (1310)。消息可以通过 SMS、使用发送请求 (sendRequest) 异步接口发送,使用如下参数:

- [0171] 请求类型 (RequestType):获取 (GET)
- [0172] 呼叫者 (Caller):<服务提供者 ID 或服务提供者 URL>
- [0173] 有效载荷:
- [0174] 领域:配置
- [0175] 消息类型 (MessageType):配置查询 (ConfigurationQuery)

[0176] 会话 ID :< 会话 ID>

[0177] 此后,过程结束 (1312)。

[0178] 图 14 示出了用于服务配置功能的配置请求 (2) 从设备到 SP 过程的逻辑流程图 1400。设备 106 可以请求平台 102 上的最后配置 (1402)。设备 106 可以通过如下接口发送请求 :

[0179] 发送请求 (sendRequest) (

[0180] 请求类型 (RequestType) :获取 (GET)

[0181] 呼叫者 (Caller) :< 设备 ID>

[0182] 有效载荷 :

[0183] 领域 :配置

[0184] 消息类型 (MessageType) :配置查询 (ConfigurationQuery)...)

[0185] 平台 102 接收请求,处理请求,并获取请求的配置 (1404)。如果提供所有所需的参数,过程管理器 238 确定设备 ID 是否存在于设备储存库 224 中 (1406)。如果未提供所需参数,过程管理器 238 通知错误 (1408) 并且过程结束 (1412)。在步骤 1406,如果设备 ID 不存在于设备储存库 224 中,过程管理器 238 通知错误 (1408) 并且过程结束 (1412)。如果设备 ID 存在,过程管理器 238 向对应于设备 ID 的设备 106 发送配置 (Configuration) 消息 (1410)。消息可以使用发送请求 (sendRequest) 同步接口发送,使用如下参数 :

[0186] 响应状态 (ResponseStatus) :OK

[0187] 请求类型 (RequestType) :获取 (GET)

[0188] 有效载荷 :

[0189] 领域 :配置

[0190] 参数 :< 配置参数 (ConfigParam) 数组 >

[0191] 消息类型 (MessageType) :配置命令. 激活 (ConfigurationCommand. Activation)

[0192] 状态 (Status) :ON(对所有和该设备 ID 相关的领域)

[0193] 此后,过程结束 (1412)。

[0194] 图 15 示出了用于服务提供者订阅功能的订阅过程的逻辑流程图 1500。用户 112 或第三方服务提供者 108 可以向应用领域请求订阅,以接收和请求和应用领域相关的事件集 (1502)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据 :服务提供者 ID(servieProviderID),应用领域 ID(applicationDomainID)。平台 102 接收请求并将请求转发给过程管理器 238 (1504)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。可以使用订阅领域接口转发请求。如果提供所有所需的参数,过程管理器 238 确定请求的订阅是否已经存在 (1506)。如果未提供所需参数,过程管理器 238 通知错误 (1508) 并且过程结束 (1512)。在步骤 1506 中,如果已经存在订阅,过程管理器 238 通知错误 (1508) 并且过程结束 (1512)。如果请求的订阅并非已存在,在设备储存库 224 中创建新记录 (1510),并且过程结束 (1512)。

[0195] 图 16 示出了用于设备提供者订阅功能的取消订阅过程的逻辑流程图 1600。用户 112 或第三方服务提供者 108 可以请求取消到应用领域的订阅 (1602)。伴随该请求,用户 112 或第三方服务提供者 108 可以提供如下数据 :服务提供者 ID(servieProviderID),应用领域 ID(applicationDomainID)。平台 102 接收请求并将请求转发给过程管理器 238 (1604)。

平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。可以使用去订阅领域 (unsubscribeDomain) 接口转发请求。如果提供所有所需的参数,过程管理器 238 确定请求的订阅是否已经存在 (1606)。如果未提供所需参数,过程管理器 238 通知错误 (1608) 并且过程结束 (1612)。在步骤 1606 中,如果不存在订阅,过程管理器 238 通知错误 (1608) 并且过程结束 (1612)。如果请求的订阅已存在,在设备储存库 224 中移除该记录 (1610),并且过程结束 (1612)。

[0196] 图 17 示出了用于应用通信功能的事件分发过程的逻辑流图 1700。设备 106 可以触发特定事件并发起到平台 102 的通信 (1702)。请求可以使用发送请求 (sendRequest) 接口发送,具有如下参数:

[0197] 请求类型 (RequestType):发送 (POST)

[0198] 呼叫者 (Caller):<设备 ID(deviceID)>

[0199] 接受者 (Recipient):<接受者 url>

[0200] 有效载荷:

[0201] 领域:<应用领域 (application domain)>

[0202] 消息类型 (MessageType):事件 <ID>(Event<ID>)

[0203] 数据:<数据>

[0204] 设备 106 还可以发回请求的数据 (1704)。由于 SMS 不是双向信道,设备 106 可以使用具有如下参数的、发送请求 (SendRequest) 接口向平台 102 发起新的通信:

[0205] 请求类型 (RequestType):回复 (REPLY)

[0206] 呼叫者 (Caller):<设备 ID>(deviceID)

[0207] 接受者 (Recipient):<接受者 url>

[0208] 有效载荷:

[0209] 领域:<应用领域>(application domain)

[0210] 消息类型 (MessageType):事件 <ID>(Event<ID>)

[0211] 数据:<数据>

[0212] 会话 ID:<会话 id>

[0213] 平台 102 从步骤 1702 或 1704 接收来自设备的消息 (1706),并且确定是否提供所需参数 (1708)。如果未提供所需参数,平台 102 使用发送响应 (sendResponse) 接口向设备 106 发送 NACK,参数 (1710) 如下:

[0214] 响应状态 (ResponseStatus):K0

[0215] 有效载荷:

[0216] 消息类型 (MessageType):确认 (Acknowledge)

[0217] 然后过程结束 (1722)。如果提供所有所需的参数,过程管理器 238 确定是否和请求一起提供了接受者 (1712)。如果提供了接受者,平台 102 使用通知事件 (notifyEvent) 接口将请求转发给接受者 (1714)。下一步,平台 102 使用发送响应 (sendRequest) 接口向设备 106 发送 ACK 信号,伴随如下参数 (1716):

[0218] 响应状态 (ResponseStatus):OK

[0219] 有效载荷:

[0220] 消息类型 (MessageType):确认 (Acknowledge)

[0221] 下一步,过程结束(1722)。在步骤 1712,如果没有提供接受者,平台 102 解析请求并使用通知事件(notifyEvent)接口将事件分发给所有订阅的服务提供者(包括用户 112 和 / 或第三方服务提供者 108)。分发是根据触发的事件所属的领域。下一步,<数据>存储在数据储存库 222 中,或者其他储存库 220(1720)。此后,平台 102 使用发送响应(sendResponse)接口向设备 106 发送 ACK 信号,伴随如下参数(1716):

[0222] 响应状态(ResponseStatus):OK

[0223] 有效载荷:

[0224] 消息类型(MessageType):确认(Acknowledge)

[0225] 下一步,过程结束(1722)。

[0226] 图 18 示出了用于应用通信功能的应用请求过程的逻辑流图 1800。用户 112 或第三方服务提供者 108 可以请求用户或第三方服务提供者订阅的领域中实体的值(1802)。请求可以伴随数据 DCM_Message:<msg>而发送。下一步,平台 102 接收请求并将请求转发到过程管理器 238(1804)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接收请求。请求可以使用接收 SPR 请求(receiveSPRequest)异步接口转发,具有如下参数:

[0227] 有效载荷:

[0228] 领域:<应用领域>(application domain)

[0229] 消息类型(MessageType)

[0230] 应用查询(ApplicationQuery):<ID>

[0231] 如果提供所有所需的参数,过程管理器 238 确定用户 112 或第三方服务提供者 108 是否订阅到请求资源所属的领域(1806)。如果未提供所需参数,过程管理器 238 通知错误(1808)并且过程结束(1812)。在步骤 1806,如果用户 112 或第三方服务提供者 108 不订阅到请求资源所属的领域,则过程管理器 238 通知错误(1808)并且过程结束(1812)。如果用户 112 或第三方服务提供者 108 订阅到领域,则过程管理器 238 通过 SMS、使用发送请求(sendRequest)异步接口将应用查询(ApplicationQuery)消息发送到设备 106(1810)。消息和如下参数一起发送:

[0232] 请求类型(RequestType):获取(GET)

[0233] 呼叫者(Caller):<服务提供者 ID 或服务提供者 url>(serviceProviderID or Service Provider url)

[0234] 有效载荷:

[0235] 领域:<应用领域>(application domain)

[0236] 消息类型(MessageType)

[0237] 应用查询(ApplicationQuery).<ID>

[0238] 会话 ID(ConversationID):<ConversationID>

[0239] 下一步,过程结束(1812)。

[0240] 图 19 示出了设备管理过程的逻辑流图。用户 112 或第三方服务提供者 108 可以请求更新给定模型的所有设备 106 的固件(1902)。请求可以伴随如下的数据:设备模型,固件 url。下一步,平台 102 接收请求并将请求转发给过程管理器 238(1904)。平台 102 可以通过第三方网关 218,垂直解决方案网关 216,门户网站 260/262,或者网络控制台 266 接

收请求。请求可以使用接收 SP 请求 (receiveSPRequest) 异步接口转发,具有如下参数:

[0241] 有效载荷:

[0242] 领域:配置 (Configuration)

[0243] 消息类型 (MessageType):配置命令 (ConfigurationCommand)

[0244] 配置参数 (Configuration Parameters):下载固件 (download firmware) (ON)

[0245] 设备模型 (Device Model):<设备模型 (deviceModel)>

[0246] 固件 url:<url>

[0247] 如果提供所有所需的参数,过程管理器 238 确定是否存在所提供设备模型的设备 (1906)。可以通过搜索设备储存库 224 来确定。如果未提供所需参数,过程管理器 238 通知错误 (1908) 并且过程结束 (1912)。在步骤 1906,如果不存在提供设备模型的设备,过程管理器 238 通知错误 (1908) 并且过程结束 (1912)。如果存在这样的设备,则过程管理器 238 通过 SMS、使用发送请求 (sendRequest) 异步接口向设备 106 发送配置命令 (ConfigurationCommand) 消息 (1910)。消息和如下参数一起发送:

[0248] 请求类型 (RequestType):通知 (Notify)

[0249] 呼叫者 (Caller):DCM

[0250] 有效载荷:

[0251] 领域:配置

[0252] 消息类型 (MessageType):配置命令

[0253] 会议参数 (ConfParameters):下载固件 (ON)

[0254] 会话 ID (ConversationID):<ConversationID>

[0255] 固件 url:<url>

[0256] 下一步,过程结束 (1912)。

[0257] 从平台 102 接收到下载固件通知消息后,设备 106 可以连接到提供的 url 以发起固件下载 (1914)。平台 102 可以从设备 106 接收下载请求并处理该请求 (1916)。接着,平台 102 确定是否存在固件下载包,以及请求下载的设备是否被授权 (1918)。如果没有用于下载的包或者如果设备未被授权,平台 102 通知错误 (1920) 并且过程结束 (1924)。如果可获得包以及设备得到授权,则下载开始 (1922)。如果下载成功,则过程结束 (1924)。如果下载不成功,则平台 102 通知错误 (1920) 并且过程结束 (1924)。

[0258] 图 20 示出了设备监控过程的逻辑流图 2000。监控服务提供者 - 例如设备监控组件 264 的垂直解决方案 202 或其他由第三方服务提供者 108 提供的、用以提供监控服务的垂直解决方案 - 可以执行定期设备监控请求 (2002)。该请求可以伴随如下的数据:<设备 id> 列表 (<Device id>list)。监控服务提供者还可以执行实时设备监控请求 (2004)。请求可以伴随如下数据:<设备 id> 列表。在运行步骤 2002 或 2004 步骤的请求之一以后,运行应用通信 - 应用请求过程以向设备 106 转发设备监控请求 (2006)。

[0259] 设备 106 可以触发错误条件以及发起到监控服务提供者的通信 (2008)。下一步,运行应用通信 - 事件分发过程以向设备 106 分发之前向服务提供者请求的状态数据 (2010)。下一步,数据存储于监控储存库中,监控储存库可以是库 220 之一 (2012)。存储在监控储存库中的数据可以是从设备 106 接收的、响应于设备监控请求的数据,或者是响应于故障条件发送给设备的数据。然后,事件处理器组件 232 基于预定规则来管理从设备

106 捕获的状态信息 (2014)。事件处理器 232 和警告组件 234 通信,以向监控服务提供者提交故障状况。传达故障状况的三个示例渠道是:在网络控制台 266 上显示,经 SMS 消息发送到分发列表,以及向分发列表发邮件。在使用网络控制台 266 的情况下,刷新网络控制台以显示故障条件 (2016),并且过程结束 (2018)。

[0260] 上述多种接口由平台 102 暴露和使用,以和设备 106 或垂直解决方案 202 通信。接口可以是异步或同步接口,例如 SMS, HTTP, SOAP,或具有回调的 SOAP。

[0261] 通信管理器 240 将在下面进一步详细描述。

[0262] 和平台 102 交互的设备 106 可以向通信管理器 240 发送其消息 / 采样。通信管理器 240 可以在 /M2Mweb/reciever 暴露 com. company. amos. m2m. dcm. web. StandardDeviceServlet 伺服小程序。一旦伺服小程序接收到了消息,则选择合适的适配器来从消息中提取数据,并产生设备消息 406。适配器可以存储在 M2M_DCM 储存库中,其可以是库 220 之一。适配器可以根据如下的分类层次存储在 M2M_DCM 储存库中:

[0263] 设备 -> 模型 -> 模型适配器 -> 适配器 (通过通信信道)。

[0264] 一旦产生了设备消息 406,则消息由转换器 244 转换成平台的其余组件可以理解的标准、高抽象级别服务消息 408。转换器可以存储在 M2M_DCM 储存库中,并且可以存储在 M2M_DCM. DeviceToSvcMsgTransformer 表中。可以通过匹配消息中包含的 namespaceURI 参数和 M2M_DCM. DeviceToSvcMsg Transformer. expression 字段来选择合适的转换器。通过调用所选转换器的 transform() 方法来将消息 406 转换为服务消息 408。一旦消息得到转换,将转换的服务消息 408 发送给过程管理器 238,以递送给平台 102 合适的组件。参考图 4 描述的应用管理器 246 可以与过程管理器 238 集成。

[0265] 如下提供了制造者 DUMMY 为名为 MyNewModel 的新设备模型提供支持的示例过程。

[0266] 首先,配置 M2M_DCM 储存库。将记录 (DUMMY) 插入到 M2M_DCM. Manufacture 表中。然后,将记录 (N, MyNewModel, DUMMY) 记录插入 M2M_DCM. Model 表中,其中 N 是 M2M_DCM. Model 表的键值。N 是累进数字。在这一示例中,取值为 4。然后将记录 (4, 1) 插入到 M2M_DCM. Model_adapter 表中。在 M2M_DCM. Adapter 表中,1 是其中特定适配器之一的键值。

[0267] 还需要新的转换器。转换器的目的是为了将从设备 106 接收到的消息转换为标准消息,以发送给过程管理器 238 或平台 102 的其他组件。每个转换器都是直接或间接由 com. company. amos. m2m. dcm. msgprocessing. MessageTransformer 类扩展而来。

[0268] 在 XSL 转换器的情况下,其实现的一个示例是 com. company. amos. m2m. dcm. msgprocessing. impl. AbstractXsltTransformer。AbstractXsltTransformer 是抽象类,并且通过实现返回 XSLT 转换器文件路径的 getXslt() 方法来扩展。转换器在输出上产生针对 XSD 文件 dcmMessage. xsd 进行验证的 XML,如表 4 所示:

表 4

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
  <xs:element name="DCMDOCUMENT">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="EVENT"/>
      </xs:sequence>
      <xs:attribute name="version" use="optional" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="EVENT">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="TIMESTAMP" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="DEVICE" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="TRANSACTION_ID" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="TRIGGER" minOccurs="1" maxOccurs="1"/>
        <xs:element maxOccurs="unbounded" ref="RAW_DATA"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

[0269]

[0270]

```

<xs:element name="TIMESTAMP" type="xs:dateTime"/>
<xs:element name="DEVICE" type="ID"/>
<xs:element name="TRANSACTION_ID" type="xs:string"/>
<xs:element name="TRIGGER">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ID">
        <xs:sequence>
          <xs:element minOccurs="0" ref="DETAIL"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="DETAIL" type="xs:string"/>
<xs:element name="RAW_DATA">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="PARAM"/>
    </xs:sequence>
    <xs:attribute name="impl" use="optional" type="xs:string"/>
    <xs:attribute name="type" use="required" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="PARAM">
  <xs:complexType>
    <xs:attribute name="name" use="required" type="xs:string"/>
    <xs:attribute name="value" use="required" type="xs:string"/>
    <xs:attribute name="javatype" use="required" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:complexType name="ID">
  <xs:sequence>
    <xs:element ref="ID" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ID" type="xs:string"/>
</xs:schema>

```

[0271] 然后将新的记录插入 M2M_DCM.DeviceToSvcMsgTransformer 表中。例如,如果转换器类的新名称是 com.company.amos.m2m.dcm.msgprocessing.impl.MyNewXslTransformer,要转换的XML的根节点由 XpathQuary//:dummy-service[@

domain = 'sampling'], 则将记录 (//:dummy-service[domain = 'sampling'], com.company.amos.m2m.dcm.msgprocessing.impl.MyNewXslTransformer) 插入到 M2m_DCM.DeviceToSvcMsgTransformer 表中。

[0272] 为了在平台 102 中完全支持新的设备类型, 还需要在过程管理器 238 组件上执行一些配置:

[0273] 在 M2M_AIB.device_type 表中插入新设备类型。

[0274] 在 M2M_AIB.repository 表中插入合适的配置, 以及

[0275] 在 M2M_AIB.repository_group 表中插入储存库和组的关系。

[0276] M2M_AIB 数据库可以是为平台 102 保存配置信息的储存库 220 之一。

[0277] 平台 102 可以在例如数据储存库 222 的储存库的专用表中存储从设备发送的数据。M2M_AIB 数据库可以是这样的储存库之一。用于存储数据的表具有定义如下的、名为 id_base 的第一键值域:

[0278] PRIMARY KEY('id_base'), KEY 'fk_1_base1' ('id_base'),

[0279] CONSTRAINT 'fk_1_base1' FOREIGN KEY('id_base') REFERENCES 'base' ('id_base')

[0280] 图 21 示出了数据采集表的一个示例 2100。新的设备类型 DUMMY 数据采集表的一个示例可以定义如下:

[0281]

```

CREATE TABLE 'dummy' (
  'id_base' int(11)NOT NULL,
  'field_1' varchar(255) default NULL,
  'field_2' varchar(255) default NULL,
  PRIMARY KEY('id_base'),
  KEY 'fk_home_metering_base1' ('id_base'),
  CONSTRAINT 'fk_home_metering_base1' FORIEN
KEY('id_base') REFERENCES 'base'

```

[0282]

```

  ('id_base') ON DELETE NO ACTION ON UPDATE NO
ACTION
  ) ENGINE=InnoDB DEFAULT CHAESSET=latin1;

```

[0283] 平台 102 可以使用 JPA (Java Persistence API) 来管理数据访问。M2M-DAL 模块管理对 M2M_AIB DB 的访问。

[0284] 一旦已经创建了数据采集表, 在包 com.company.amos.m2m.dal.entity 中创建具有 JPA 注解的 POJO 类。

[0285] 平台 102 将数据访问暴露为网络服务。为了实现该任务, 特别是为了识别返回的 POJO 属性, 使用自定义的注解 :com.company.amos.m2m.dal.entity.Exclude。@Exclude 注

解设置在 POJO 方法上,其不返回暴露的属性之一的值。

[0286] 针对表 dummy,可以如表 5 所示的那样来实现有注解的 POJO :

[0287]

表 5
<pre>@Entity @Table(name = "dummy") public class Dummy implements java.io.Serializable { private static final long serialVersionUID = -8228360678364527411L; private int idBase; private Base base; private String field1; private String field2; public Dummy() { } public Dummy(Base base) {</pre>

[0288]

```
    this.base = base;
}
public Dummy(Base base, String field1, String field2) {
    this.base = base;
    this.field1 = field1;
    this.field2 = field2;
}
@GenericGenerator(name = "generator", strategy = "foreign", parameters =
@Parameter(name = "property", value = "base"))
@Id
@GeneratedValue(generator = "generator")
@Column(name = "id_base", unique = true, nullable = false)
@Exclude
public int getIdBase() {
    return this.idBase;
}
@Exclude
public void setIdBase(int idBase) {
    this.idBase = idBase;
}
@OneToOne(fetch = FetchType.LAZY)
@PrimaryKeyJoinColumn
@Exclude
public Base getBase() {
    return this.base;
}
@Exclude
public void setBase(Base base) {
    this.base = base;
}
@Column(name = "field_1", length = 255)
public String getField1() {
    return field1;
}
```

```

    }
    @Exclude
    public void setField1(String field1) {
        this.field1 = field1;
    }
    @Column(name = "field_2", length = 255)
    public String getField2() {
        return field2;
    }
    @Exclude
    public void setField2(String field2) {
        this.field2 = field2;
    }
}

```

[0289] 下面将进一步描述事件处理器 232 的细节。

[0291] 第三方服务提供者 108 或用户 112 可以通过调用平台 102 暴露的网络服务接口，通过第三方网关 218 访问事件处理器 232。事件处理器通过使用规则，动作和函数来管理和处理事件。

[0292] 规则是以（用户 id, 服务 id, 设备 id）和相关动作集合为特征的逻辑表示。第三方服务提供者 108 或用户 112 可以将规则和一个或多个设备 106 相关联。规则可以具有一个或多个与之相关的动作。

[0293] 逻辑表示必须由第三方服务提供者 108 或用户 112 供给，并且可以包括零个或更多个函数。

[0294] 函数是事件处理器 232 提供的复杂逻辑表示。对于平台 102 托管的每个服务都存在一组域，使用这组域第三方服务提供者 108 或用户 112 可以构造逻辑表示。域是以相关服务为特征的实体。如果逻辑表达式估值是正的，则运行和特定规则相关的动作集。

[0295] 动作是事件处理器 232 提供的任务，并且能够和零个或 N 个输入参数相关。

[0296] 逻辑表示

[0297] 逻辑表示可以用逻辑运算符，域，函数来构造。域可以通过 findField 服务获取。可以通过 findFunctions 服务获取函数。

[0298] 在逻辑表示中，域由语法 \$(fieldName) 表示，其中 fieldName 是域名。在逻辑表达式中，函数可以用语法 #functionName(input1, input2) 表示，其中 functionName 是函数名，input1, input2 是函数的输入参数。

[0299] 表 6 示出了逻辑表达式的一些示例：

表 6	
Description	Rule
if checkStatus is TRUE then execute actions	\$(checkstatus) == true
if a sampling is received then execute actions	\$(baseid)
if the field temperature of a sampling is greater than 30 degrees then execute actions	\$(temperature) > 30

[0300]

[0301] 动作

[0302] 动作是事件处理器 232 提供的任务并且可以和 0 个或 N 个数据参数相关。

[0303] findActionAndParamiters 服务提供可用动作列表。

[0304] 规则

[0305] 规则是以 (顾客 Id, 服务 Id, 设备 Id) 以及相关动作集为特征的逻辑表达式。第三方服务提供者 108 或用户 112 可以将规则和一个或多个设备相关。规则可以具有一个或多个相关的动作。

[0306] 为了创建新的规则, 第三方服务提供者 108 或用户 112 可以调用 addRule 服务。

[0307] 一旦创建了规则, 第三方服务提供者 108 或用户 112 可以 :

[0308] a. 通过调用 modifyRuleExpression 服务修改相关逻辑表达式,

[0309] b. 通过调用 deactivateRule 服务来去激活规则,

[0310] c. 通过调用 addAction 服务来添加新动作, 或者

[0311] d. 通过调用 activateRule 服务来激活规则。

[0312] 表 7 示出了在平台 102 和设备 106 或者外部网关 104 间交换的 XML 消息的示例。

[0313]

表 7

```

<?xml version="1.0" encoding="UTF-8" ?>
:      <home-metering          domain="sampling"
      xmlns="http://www.domotica.com">
: <event id="SAMPLING" detail="">
      <timestamp>2010-01-13T12:50:00.000</timestamp>
      <senderMacAddress
      protocol="IEEE">008098E91FA2</senderMacAddress>
      <consumption unit="kWh">0.483</consumption>
      <voltage unit="V">111.9</voltage>
      <current unit="A">0.055</current>
</event>
: <event id="SAMPLING" detail="">
      <timestamp>2010-01-13T12:51:00.000</timestamp>
      <senderMacAddress
      protocol="IEEE">008098E91FA2</senderMacAddress>
      <consumption unit="kWh">0.483</consumption>
      <voltage unit="V">222.9</voltage>
      <current unit="A">0.078</current>
</event>
: <event id="SAMPLING" detail="">
      <timestamp>2010-01-13T12:52:00.000</timestamp>
      <senderMacAddress
      protocol="IEEE">008098E91FA2</senderMacAddress>
      <consumption unit="kWh">0.583</consumption>
      <voltage unit="V">333.9</voltage>

```

[0314]

```

    <current unit="A">0.078</current>
  </event>
</home-metering>

```

```

<?xml version="1.0" encoding="UTF-8" ?>
-   <home-metering          domain="sampling"
      xmlns="http://www.domotica.com">
-   <event id="SAMPLING" detail="">
      <timestamp>2010-01-11T12:50:00.000</timestamp>
      <senderMacAddress protocol="IEEE">333</senderMacAddress>
      <presence>1</presence>
    </event>
  </home-metering>

```

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
-   <home-metering          xmlns="http://www.domotica.com"
      domain="sampling">
-   <event detail="" id="SAMPLING">
      <timestamp>2009-11-18T14:46:37.093</timestamp>
      <senderMacAddress protocol="IEEE">333L</senderMacAddress>
      <lightLevel unit="LUM">16.0</lightLevel>
    </event>
-   <event detail="" id="SAMPLING">
      <timestamp>2009-11-18T14:46:39.062</timestamp>
      <senderMacAddress protocol="IEEE">333T</senderMacAddress>
      <temperature unit="C">31.5</temperature>
    </event>
-   <event detail="" id="SAMPLING">
      <timestamp>2009-11-18T14:46:39.171</timestamp>
      <senderMacAddress protocol="IEEE">333H</senderMacAddress>
      <humidity unit="%">39.3</humidity>
    </event>
  </home-metering>

```

[0315] 设备映射

[0316] 平台 102 可以为了保持对所有和平台通信中的设备 106 的追踪而创建和维护设备映射,以有效管理设备。设备映射可以识别和平台 102 通信的设备。对每个识别的设备 106,设备映射可以表示设备用于和平台 102 通信的信道,以及设备是否仅作为仅能向平台发送信息的传感器,设备是否仅能从平台接收命令,设备是否仅能从平台接收命令,或者设备是否能接收命令并作为传感器。设备映射还可以包括其他可以用于管理设备 106 的信息。设备映射可以存储在库 220 之一,例如设备储存库 224。

[0317] 平台可以直接从设备 106 接收填充设备映射所必需的信息。平台 102 还可以从外部网关 104 接收必要的信息,其从连接到自身的设备 106 收集信息,并将收集到的信息提交给平台。在一个实施方式中,外部网关 104 可以创建连接到其自身的所有设备 106 的设备映射,并将映射转发给平台 102(而不是仅仅中继信息),其反过来聚集从其他外部网关接

收的映射以创建和平台相连的所有设备 106 的完整设备映射。

[0318] 网络服务

[0319] 第三方网关组件 218 提供规定暴露给第三方的服务的网络服务接口。在一个实现中,网络服务和第三方网关组件 218 之间的连接是 HTTPS/1.1 上的简单对象接入协议 (SOAP) 或 HTTP/1.1 上的 SOAP。HTTPS 或 HTTP 可以作为用于 SOAP 消息的发送协议。在简单 HTTPS/HTTP 连接中,可以以同步模式运行请求和响应消息,以使得此后网络服务应用运行的网络服务客户端应用实现 SOAP 请求,并且返回适当的 SOAP 响应。做出该请求的客户端可以在发送另一个请求之前一直等到已经接收到一个响应。可存在多个并行的 HTTPS/HTTP 连接以使得网络服务能实现多个请求的并行处理。

[0320] 在一个实现中,连接到第三方网关组件 218 网络服务的外部系统使用 SSL 客户端证书来认证。可以使用应用级用户名 / 密码组合作为客户认证的另一层并且该应用及认证可以特别用于认证内部系统。连接到第三方网关组件 218 的外部系统可以是由第三方服务提供者 108 或用户 112 运行的系统。

[0321] 第三方网关组件 218 网络服务包括提供管理和监控设备的方法的数据访问网络服务。特别地,数据访问网络服务的方法允许提取关于设备运行的信息,例如命令移除,替代,放置,配置,以及和设备相关的事件例如存在和开机 / 关机事件。

[0322] 图 22 示出了数据访问网络服务所用的数据访问网络服务数据模型的一个示例 2200。数据访问网络服务模型表示在多个使得数据访问网络服务能够定义和提取信息的平台实体之间的关系。数据访问网络服务数据模型包括设备组 (device_group) 实体 2202,组 (group) 实体 2204,客户服务 (customer_service) 实体 2206,事件 (event) 实体 2208,设备 (device) 实体 2210,客户 (customer) 实体 2212,服务 (service) 实体 2214,设备类型 (device_type) 实体 2216,以及基础 (base) 实体 2218。

[0323] 数据访问网络服务暴露方法,包括:findCommandByDeviceId; findEventByDeviceId; findEventDetail; 以及 findEventDetailsByDeviceId。

[0324] 表 8 示出了用于为特定设备和参数获取命令列表的 findCommandDeviceId 方法的 findCommandByDeviceId 请求消息的示例。

[0325]

数据元素	类型	强制性	大小	描述
commandType	String (字符串)	N	30	用于在设备上, 服务或通常在事件上识别操作的类型
deviceId	String	Y	30	特定设备的标识符
fromData	dateTime (日期时间)	N		用于提取特定日期的命令列表
Status	String	Y	30	表示设备的实际状态(供给, 活动等)
toDate	dateTime	N		用于提取特定日期之前的命令列表
transactionId	String	Y	255	表示系统 100 中的事务标识符

[0326]

[0327] 表 9 示出了用于 findCommandByDeviceId 方法的 findCommandByDeviceId 响应消息的示例。

[0328]

errorCode	Int (整数)	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括错误码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)
commandId	Int	N	11	请求发现的特定命令的标识符
commandType	String	N	30	描述请求发现的命令的类型
lastDate	dateTime	N		命令运行的最近日期
Status	String	N	30	包含特定命令结果的域

[0329] 表 10 示出了用于为特定设备和参数获取事件列表的 findEventByDeviceId 方法的 findEventByDeviceId 请求消息的示例。

[0330]

数据元素	类型	强制性	大小	描述
deviceId	String	Y	30	特定设备的标识符
eventType	String	N	50	事件的标识符
fromDate	dateTime	N		用于提取从特定日期开始的命令列表
toDate	DateTime	N		用于提取特定日期之前的命令列表
TransactionId	String	Y	255	表示 M2M 系统中事务标识符的域

[0331] 表 11 示出了用于 findEventById 方法的 findEventById 响应消息的示例。

[0332]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
ErrorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)
eventId	Int	N	11	请求找到的特

[0333]

				定事件的标识符
commandType	String	N	30	描述请求找到的命令的类型
eventType	String	N	30	描述请求找到的事件的类型
LastDate	DateTime	N		事件校验的最近日期

[0334] 表 12 示出了用于为特定设备和参数获取事件详细列表的 findeventDetail 方法的 findeventDetail 请求消息的示例。

[0335]

数据元素	类型	强制性	大小	描述
eventId	String	N	30	特定事件的标识符
transactionId	String	Y		表示系统 100 中的事务标识符

[0336] 表 13 示出了用于 findeventDetail 方法的 findeventDetail 响应消息的示例。

[0337]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
ErrorDesc	String	Y	255	包括可能的错

[0338]

				误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)
Name	String	N	50	该域是参数配置的名字
Value	String	N	50	该域是参数配置的值

[0339] 表 14 示出了用于为特定设备 ID 取回事件详细列表的 findEventDetailByDeviceId 方法的 findEventDetailByDeviceId 请求消息的示例。

[0340]

数据元素	类型	强制性	大小	描述
deviceId	String	Y	30	特定设备的标识符
eventType	String	N	50	某事件的标识符
fromDate	dateTime	N		用于提取从特定日期开始的命令列表
toDate	Date Time	N		用于提取特定日期之前的命令列表
TransactionId	String	Y	255	表示 M2M 系统中事务标识符的域

[0341] 表 15 示出了用于 findEventDetailByDeviceId 方法的 findEventDetailByDeviceId 响应消息的示例。

[0342]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。 该编码可以表示错误或成功之一
ErrorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0=成功; 1=错误)
Name	String	N	50	参数配置的名字
Value	String	N	50	参数配置的值

[0343] 第三方网关组件 218 提供设备管理网络服务。图 23 示出了设备管理网络服务数据模型 2300 的一个示例,其定义了多个使得设备管理网络服务能够定义和提取信息的平台实体之间的关系。设备管理网络服务数据模型包括设备组 (device_group) 实体 2302,组 (groups) 实体 2304,客户服务 (customer_service) 实体 2306,设备 (device) 实体 2308,客户 (customer) 实体 2310,以及服务 (service) 实体 2312。

[0344] 设备管理器网络服务提供管理设备和提供新设备 (例如设备配置,以及使能和去使能特征),配置设备 (如,设备的设置和参数的改变),以及远程设备管理 (如,移除/更换设备),以及激活服务的方法。来自设备管理网络服务展示的方法包括:激活;检查状态;配置;细节配置;去激活;插入设备;移除设备;以及移除设备。

[0345] 表 16 示出了用于激活用于特定设备的服务的激活方法的激活请求消息的示例。

[0346]

数据元素	类型	强制性	大小	描述
configurationId	Int	N	11	已有配置的标识符
customerId	String	Y	30	特定客户的标识符
deviceId	String	Y	30	特定设备的标识符
serviceId	Int	Y	11	特定服务的标识符
TransactionId	String	Y	255	表示系统 100 中的事务标识符

[0347] 表 17 示出了用于 Activate(激活)方法的 Activate 响应消息的示例。

[0348]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0349]

[0350] 表 18 示出了用于向特定设备发送 checkStatus 请求的 checkStatus 方法的 checkStatus 请求消息的示例。

[0351]

数据元素	类型	强制性	大小	描述
ConfigurationId	Int	N	11	已有配置的标识符
customerId	String	Y	30	特定客户的标识符
deviceId	String	Y	30	特定设备的标识符
serviceId	Int	Y	11	特定服务的标识符
transactionId	String	Y	255	表示系统 100 中的事务标识符

[0352] 表 19 示出了用于 checkStatus 方法的 checkStatus 响应消息的示例。

[0353]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0354]

[0355] 表 20 示出了用于通过选择配置来配置设备的 configure 方法的 configure 请求消息的示例。

[0356]

数据元素	类型	强制性	大小	描述
ConfigurationId	Int	N	11	已有配置的标识符
customerId	String	Y	30	特定客户的标识符
deviceId	String	Y	30	特定设备的标识符
serviceId	Int	Y	11	特定服务的标识符
transactionId	String	Y	255	表示系统 100 中的事务标识符

[0357] 表 21 示出了用于 checkStatus 方法的 configure 响应消息的示例。

[0358]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0359]

[0360] 表 22 示出了用于通过提供配置参数来配置设备的 configureByDetails 方法的 configureByDetails 请求消息的示例。

[0361]

数据元素	类型	强制性	大小	描述
Name	String	Y	50	参数配置的名字
Value	String	Y	50	参数配置的值
customerId	String	Y	30	特定用户的标识符
deviceId	String	Y	30	特定设备的标识符
serviceId	Int	Y	11	特定服务的标识符
ctransactionId	String	M	255	表示在系统 100 中的事务标识符

[0362] 表 23 示出了用于 configureByDetails 方法的 configureByDetails 响应消息的示例。

[0363]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0364] 表 24 示出了用于对特定设备去激活 (deactivate) 服务的去激活方法的去激活请求消息的示例。

[0365]

数据元素	类型	强制性	大小	描述
ConfigurationId	Int	N	11	已有配置的标识符
customerId	String	Y	30	特定客户的标识符
deviceId	String	Y	30	特定设备的标识符
serviceId	Int	Y	11	特定服务的标识符
transactionId	String	Y	255	表示系统 100 中的事务标识符

[0366] 表 25 示出了用于去激活方法的去激活响应消息的示例。

[0367]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0368] 表 26 示出了用于执行新设备供给的 insertDevice 方法的 insertDevice 请求消息的示例。

[0369]

数据元素	类型	强制性	大小	描述
customerId	String	Y	30	特定用户的标识符
deviceId	String	Y	30	特定设备的标识符
deviceType	String	Y	255	指定设备类型
groupId	Int	Y	11	拥有该客户的组的标识符
iccid	String	N	20	Sim iccd
ipAddress	String	N	255	设备的 ip 地址
Msisdn	String	N	15	和 sim 相关的
				电话号码
transactionId	String	Y	255	表示系统 100 中的事务标识符

[0370]

[0371] 表 27 示出了用于 insertDevice 方法的 insertDevice 响应消息的示例。

[0372]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0373] 表 28 示出了用于移除已有设备的 removeDevice 方法的 removeDevice 请求消息的示例。

[0374]

数据元素	类型	强制性	大小	描述
customerId	String	Y	30	特定用户的标识符
deviceId	String	Y	30	特定设备的标识符
transactionId	String	M	255	该域表示 M2M 系统中的事务标识符

[0375]

[0376] 表 29 示出了用于 removeDevice 方法的 removeDevice 响应消息的示例。

[0377]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0378] 表 30 示出了用于使用新设备替代设备的 replaceDevice 方法的 replaceDevice 请求消息的示例。在一个实现中,新设备必须处于供给状态。

[0379]

数据元素	类型	强制性	大小	描述
customerId	String	Y	30	特定用户的标识符
oldDevice	String	Y	30	已有设备的标识符
newDevice	String	Y	30	特定新设备的标识符
transactionId	String	Y	255	表示系统 100 中的事务标识符

[0380]

[0381] 表 31 示出了用于 replaceDevice 方法的 replaceDevice 请求消息的示例。

[0382]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。 该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0383] 第三方网关组件 218 提供事件处理器网络服务提供方法以管理事件并且对设备向平台（如，外部事件）发送任何平台生成的事件（例如，内部事件）报警。特别得，事件处理器网络服务的方法：获取告警和告警的通知；以及管理规则定义和执行。

[0384] 图 24 示出了定义多个平台实体之间的关系的处理器网络服务数据模型 2400 的示例,该关系使得事件处理器网络服务能够定义和提取信息。事件处理器网络服务数据模型 2400 包括域 (field) 实体 2402,告警 (alarm) 实体 2404,设备规则 (device_rule) 实体 2406,规则动作 (rule_action) 实体 2408,服务功能 (device_function) 实体 2410,状态 (status) 实体 2412,规则 (rule) 实体 2414,设备 (device) 实体 2416,动作 (action) 实体 2418,功能 (function) 实体 2420,服务 (service) 实体 2422,以及客户 (customer) 实体 2424。

[0385] 在一个实现中,从事件处理器网络服务暴露的方法包括 :activeRule ;addAction ;addRule ;deactivateRule ;fineActionAndParameters ;findAlarms ;findFields ;findFunction ;findRules ;modifyRuleExpression ;modifyRule ;removeAction ; 以 及 removeRules。

[0386] 表 32 示出了用于激活用于特定设备的规则的 activeRule 方法的示例 activeRule 请求消息的示例。

[0387]

数据元素	类型	强制性	大小	描述
devices	String	Y	255	标识一个设备
ruleName	String	Y	255	和特定设备相关的规则的名字
transactionId	String	M	255	表示系统 100 中的事务标识符

[0388] 表 33 示出了用于 activeRule 方法的 activeRule 响应消息的示例。

[0389]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回

[0390]

				的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0391] 表 34 示出了用于向特定规则添加动作的 addAction 方法的 addAction 请求消息的示例。

[0392]

表 34-addAction 请求消息				
数据元素	类型	强制性	大小	描述
actionName	String	Y	255	该域是动作的名称
Key	String	N	255	该域是参数键值
Value	String	N	255	该域是参数值
ruleName	String	Y	255	该域是已有的规则名称
transactionId	String	Y	255	该域表示 M2M 系统的事务标识符

[0393] 表 35 示出了用于 addAction 方法的 addAction 响应消息的示例。

[0394]

表 35-activeRule 响应消息				
数据元素	类型	强制性	大小	描述

[0395]

errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0396] 表 36 示出了用于向特定设备添加新的规则和相关动作的 addRule 方法的 addRule 请求消息的示例。

[0397]

数据元素	类型	强制性	大小	描述
actionName	String	Y	255	该域是动作的名称
CustomerId	String	Y	30	特定客户端的标识符
Description	String	N		
Devices	String	Y	255	一个设备
Expression	String	Y		和规则相关的逻辑表达式
Name	String	Y	255	规则的名称
Key	String	N	255	参数键值
Value	String	N	255	参数值
ServiceName	String	Y	255	服务的名称
transactionId	String	M	255	表示系统 100 中的事务标识符

[0398] 表 37 示出了用于 addRule 方法的 addRule 响应消息的示例。

[0399]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0400] 表 38 示出了用于对于所有相关设备去激活特定规则的 deactivateRule 方法的 deactivateRule 请求消息的示例。

[0401]

数据元素	类型	强制性	大小	描述
Device	String	Y	255	该域是一个设备
ruleName	String	Y	255	该域是和一个或多个设备相关的规则的名称
transactionId	String	Y	255	该域表示在 M2M 系统中表示事务标识符

[0402] 表 39 示出了用于 deactivateRule 方法的 deactivateRule 响应消息的示例。

[0403]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)

[0404] 表 40 示出了用于获取所有动作和可获得的相关参数的 findActionAndParameters 方法的 findActionsAndParameters 请求消息的示例。

[0405]

数据元素	类型	强制性	大小	描述
transactionId	String	Y	255	该域表示在系统 100 中表示事务标识符

[0406] 表 41 示出了用于 findActionsAndParameter 方法的 findActionAndParameter 响应消息的示例。

[0407]

[0408]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)
Name	String	N	255	动作的名称
Param	String	N	255	参数类型

[0409] 表 42 示出了用于利用特定参数获取所有告警的 findAlarms 方法的 findAlarms 请求消息示例。

[0410]

数据元素	类型	强制性	大小	描述
BaseId	Int	N	11	实体标识符
deviceId	String	N	255	设备标识符
From	dateTime	N		开始搜索的起始日期
ruleName	String	N	255	规则名称
ServiceName	String	N	255	服务名称
statusType	String	N	45	警告的状态
To	dateTime	N		结束搜索的日期
transactionId	String	Y	255	表示系统 100 的事务标识符

[0411] 表 43 示出了用于 findAlarms 方法的 findAlarms 响应消息的示例。

[0412]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0=成功; 1=错误)
baseId	Int	Y	11	实体标识符
creationDate	dateTime	N		警告日期创建
lastModifyDate	dateTime	N		警告的最后更改日期
ruleName	String	N	255	规则名称
statusType	String	N	45	警告的状态

[0413] 表 44 示出了用于获取用户可以为特定服务构建规则的域列表的 findFields 方法的 findFields 请求消息的示例。

[0414]

数据元素	类型	强制性	大小	描述
ServiceName	String	Y	255	服务名称
transactionId	String	Y	255	表示 M2M 系统中的事务标识符

[0415] 表 45 示出了用于 findFields 方法的 findFields 响应消息。

[0416]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。 该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0=成功; 1=错误)
Fields	String	N	255	该域是一个字符串, 用其可以创建规则

[0417] 表 46 示出了用于获取用户可以构建用于特定服务的规则的功能列表的 findFunctions 方法的 findFunctions 请求消息的示例。

[0418]

数据元素	类型	强制性	大小	描述
ServiceName	String	Y	255	服务名称
transactionId	String	Y	255	表示 M2M 系统中的事务标识符

[0419] 表 47 示出了用于 findFunctions 方法的 findFunctions 响应消息的示例。

[0420]

数据元素	类型	强制性	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。 该编码可以表示错误或成功之一
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0=成功; 1=错误)
Functions	String	N	255	用该功能可以组成规则

[0421] 表 48 示出了用于由客户和服务获取规则的 findRules 方法的 findRules 请求消息的示例。

[0422]

数据元素	类型	强制值	大小	描述
customerId	String	N	255	该域是客户标识符
deviceId	String	N	255	该域是设备标识符
serviceName	String	N	255	该域是设备名
transactionId	String	Y	255	该域表示 M2M 系统

[0423]

				中的事务标识符
--	--	--	--	---------

[0424] 表 49 示出了用于 findRules 方法的 findRules 响应消息的示例。

[0425]

表 49-findRules 响应消息

数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0 = 成功; 1 = 错误)
CreationDate	DateTime	N		规则创建日期
Description	String	N	255	规则的描述
Expression	String	N	65535	规则的逻辑表达式
lastModifyDate	dateTime	N		规则最后修改日期
Name	String	N	255	规则名称

[0426] 表 50 示出了用于修改特定规则表达式的 modifyRule 方法的 modifyRule 请求消息的示例。

[0427]

表 50-modifyRules 请求消息				
数据元素	类型	强制值	大小	描述
Expression	String	Y	65535	规则的逻辑表达式
ruleName	String	Y	255	规则名称
TransactionId	String	Y	255	表示系统 100 中的事务标识符

[0428] 表 51 示出了用于 modifyRule 方法的 modifyRule 响应消息的示例。

[0429]

表 51-modifyRule 响应消息				
数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0 = 成功; 1 = 错误)

[0430] 表 52 示出了用于移除动作和规则之间关系的 removeAction 方法的 removeAction 请求消息的示例。

[0431]

表 52-removeAction 请求消息

[0432]

数据元素	类型	强制值	大小	描述
ActionName	String	Y	255	动作名称
Key	String	N	255	参数键值
Value	String	N	255	参数值
ruleName	String	Y	255	规则名称
TransactionId	String	Y	255	表示系统 100 中的事务标识符

[0433] 表 53 示出了用于 removeAction 方法的 removeAction 响应消息。

[0434]

数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0 = 成功; 1 = 错误)

[0435] 表 54 示出了用于移除特定规则和相关实体的 removeRule 方法的 removeRule 请求消息的示例。

[0436]

数据元素	类型	强制值	大小	描述
------	----	-----	----	----

[0437]

TransactionId	String	Y	255	表示系统 100 中的事务标识符
---------------	--------	---	-----	------------------

[0438] 表 55 示出了用于 removeRule 方法的 removeRule 响应消息的示例。

[0439]

数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0 = 成功; 1 = 错误)

[0440] 图 25 示出了定义多个平台实体之间关系的系统配置网络服务数据模型 2500 的示例,其使得系统配置网络服务能够定义和提取信息。系统配置网络服务数据模型 2500 包括项目组 (project_groups) 实体 2502,客户服务 (customer_service) 实体 2504,组 (groups) 实体 2506,项目 (project) 实体 2508,服务 (service) 实体 2510,客户 (customer) 实体 2512。

[0441] 系统配置网络服务提供管理客户,服务,组和项目间之间关系的方法。特别是系统配置网络服务的方法:关联/去关联组和项目;以及为客户激活/去激活服务。暴露的来自系统配置网络服务的方法包括:associateGroupProject;customerDomainSubscription;customerDomainUnsubscription;以及 deassociateGroupFromProject。

[0442] 表 56 示出了用于将请求中指定的组和项目相关联的 groupProject 方法的 groupProject 请求消息的示例。

[0443]

数据元素	类型	强制值	大小	描述
groupId	Int	Y	10	特定组标识
projected	Int	Y	10	特定项目的标识
TransactionId	String	Y	255	表示 M2M 系统中的事务标识符

[0444] 表 57 示出了用于 groupProject 方法的 groupProject 响应消息的示例。

[0445]

数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0 = 成功; 1 = 错误)

[0446] 表 58 示出了用于为请求中指定的客户激活服务的 customerDomainSubscription 方法的 customerDomainSubscription 请求消息的示例。

[0447]

数据元素	类型	强制值	大小	描述
customerId	String	Y	30	指定客户标识符
ServiceID	Int	Y	10	指定服务标识符
transactionId	String	Y	255	该域表示系统 100 中的事务标识符

[0448] 表 59 示出了用于 customerDomainSubscription 方法的 customerDomainSubscription 响应消息的示例。

[0449]

数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态 (0 = 成功; 1 = 错误)

[0450]

[0451] 表 60 示出了用于对请求中指定用户去激活服务的

customerDomainUnsubscription 方法的 customerDomainUnsubscription 请求消息的示例。

[0452]

数据元素	类型	强制值	大小	描述
customerId	String	Y	30	指定客户标识符
ServiceID	Int	Y	10	指定服务标识符
transactionId	String	Y	255	该域表示系统 100 中的事务标识符

[0453] 表 61 示出了用于 customerDomainUnsubscription 方法的 customerDomainUnsubscription 响应消息的示例。

[0454]

数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码
				相关的状态(0 = 成功; 1 = 错误)

[0455]

[0456] 表 62 示出了用于去连接请求中指定组和项目的 deassociateGroupFromProject 方法的 deassociateGroupFromProject 请求消息的示例。

[0457]

数据元素	类型	强制值	大小	描述
GroupId	Int	Y	10	指定组的标识符
projected	Int	Y	10	指定项目的标识符
transactionId	String	Y	255	该域表示系统 100 中的事务标识符

[0458] 表 63 示出了用于 deassociateGroupFromProject 方法的 deassociateGroupFromProject 响应消息的示例。

[0459]

数据元素	类型	强制值	大小	描述
errorCode	Int	Y	11	包括请求返回的编码。该编码可以表示错误或成功之一。
errorDesc	String	Y	255	包括可能的错误编码的描述
StatusCode	Int	Y	11	描述和错误码相关的状态(0 = 成功; 1 = 错误)

[0460]

[0461] 表 64 示出了在来自每个网络服务的发送响应中可能返回的响应编码的示例。

[0462]

表 64-响应编码			
编码	类型	文本	描述
1000	Error Code(错误编码)	Request successful done	执行成功
1001	Error Code	Events was not found by device id;	发生一个错误或者由请求无法找到
1002	Error Code	<p>O commands was not found by device id:...</p> <p>O Fields was not found for service: ...</p> <p>O Function was not found service: ...</p> <p>O Customer with id “...” was not found</p> <p>O service with name “...” was not found</p> <p>O Rules was not found</p> <p>O “Device with id” “...” was not found</p> <p>O Rule with name “...” was not found</p> <p>O Alarms was not found</p>	发生一个错误或者由请求无法找到

[0463]

		O Action with name “...” was not found	
0	Status Code(状态编码)	-	成功 (和错误编码 1000 相关)
1	Status Code	-	错误 (和表示错误 1001-1002 等错误的错误编码相关)
1008	Error Code	Check status request already opened, wait for request's expire	发生一个错误或者由请求无法找到
2001	Error Code		IB 普通错误
2101	Error Code		IB-数据访问普通错误
2102	Error Code		由调用到 CSM 引起的错误
2103	Error Code	<p>O event not found by id:...</p> <p>O commands was not found by device id: ...</p> <p>O No Group with specified ID found</p> <p>O No project with specified ID found</p> <p>O No customer with specified ID found</p> <p>O service with specified ID found</p> <p>O Some Device associated to Service with status ACTIVE or</p>	坏的输入参数

[0464]

		ACTIVATION_IN_PROGRESS exists for specified Customer	
2104	Error Code		由调用到 DCM 引起的 错误
2105	Error Code		IB-数据访问 普通错误
2106	Error Code		
2107	Error Code		库表指定的文件 无法找到
2108	Error Code		由调用到 EH 引起的错误
2109	Error Code	O events was not found by Device id: ...	无法找到 EB 元件
4001	Error Code		事件管理器普 通错误
4002	Error Code		事件管理器参 数输入错误
4003	Error Code		事件管理器无 法找到值
3001	Error Code		DCM 普通错 误
3002	Error Code		DCM 输入参 数错误

[0465] 表 65 示出了 DataAccess.wsdl 列表的示例。

表 65 - DataAccess.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DataAccessInterface"
targetNamespace="http://services.infobroker.m2m.amos.company.com"
xmlns:tns="http://services.infobroker.m2m.amos.company.com"
xmlns:ns1="infobroker.m2m.amos.company.com/eventType"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns2="infobroker.m2m.amos.company.com/eventId"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
<xs:schema targetNamespace="infobroker.m2m.amos.company.com/eventType" version="1.0"
xmlns:ns1="http://services.infobroker.m2m.amos.company.com"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:import namespace="http://services.infobroker.m2m.amos.company.com"/>
<xs:element name="eventType" type="ns1:eventType"/>
</xs:schema>
<xs:schema targetNamespace="infobroker.m2m.amos.company.com/eventId" version="1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="eventId" type="xs:int"/>
</xs:schema>
<xs:schema targetNamespace="http://services.infobroker.m2m.amos.company.com"
version="1.0" xmlns:ns1="infobroker.m2m.amos.company.com/eventType"
xmlns:ns2="infobroker.m2m.amos.company.com/eventId"
xmlns:tns="http://services.infobroker.m2m.amos.company.com"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:import namespace="infobroker.m2m.amos.company.com/eventType"/>
<xs:import namespace="infobroker.m2m.amos.company.com/eventId"/>
<xs:complexType name="subscribeRequest">
<xs:sequence>
<xs:element name="customerId" type="xs:string"/>
<xs:element name="deviceId" type="xs:string"/>
<xs:element name="eventId" type="xs:int"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="dataAccessResponse">
<xs:sequence>
<xs:element minOccurs="0" name="errorCode" type="xs:int"/>

```

[0466]

[0467]

```

<xs:element minOccurs="0" name="errorDesc" type="xs:string"/>
<xs:element minOccurs="0" name="statusCode" type="xs:int"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="commandRequest">
<xs:sequence>
<xs:element minOccurs="0" name="commandType" type="tns:commandTypes"/>
<xs:element name="deviceId" type="xs:string"/>
<xs:element minOccurs="0" name="fromDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="status" type="tns:statusType"/>
<xs:element minOccurs="0" name="toDate" type="xs:dateTime"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="commandResponse">
<xs:complexContent>
<xs:extension base="tns:dataAccessResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="commands" nillable="true"
type="tns:command"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="command">
<xs:sequence>
<xs:element minOccurs="0" name="commandId" type="xs:int"/>
<xs:element minOccurs="0" name="commandType" type="xs:string"/>
<xs:element minOccurs="0" name="lastDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="status" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="eventRequest">
<xs:sequence>
<xs:element name="deviceId" type="xs:string"/>
<xs:element minOccurs="0" ref="tns:eventType"/>
<xs:element minOccurs="0" name="fromDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="toDate" type="xs:dateTime"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="eventResponse">
<xs:complexContent>
<xs:extension base="tns:dataAccessResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="events" nillable="true"
type="tns:event"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="event">
<xs:sequence>
<xs:element minOccurs="0" name="eventId" type="xs:int"/>
<xs:element minOccurs="0" name="eventType" type="xs:string"/>
<xs:element minOccurs="0" name="lastDate" type="xs:dateTime"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="eventDetailRequest">

```

[0468]

```

<xs:sequence>
  <xs:element minOccurs="0" ref="ns2:eventId"/>
  <xs:element name="TransactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="eventDetailResponse">
  <xs:complexContent>
    <xs:extension base="tns:dataAccessResponse">
      <xs:sequence>
        <xs:element minOccurs="0" name="arrayOfDetail" type="tns:arrayOfDetail"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="arrayOfDetail">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="details" nillable="true"
      type="tns:detail"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="detail">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="attributes" nillable="true"
      type="tns:attribute"/>
    <xs:element minOccurs="0" name="name" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="attribute">
  <xs:sequence>
    <xs:element minOccurs="0" name="name" type="xs:string"/>
    <xs:element minOccurs="0" name="value" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="commandTypes">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PUT_DEVICE"/>
    <xs:enumeration value="REMOVE_DEVICE"/>
    <xs:enumeration value="REPLACE_DEVICE"/>
    <xs:enumeration value="ACTIVATE_SERVICE"/>
    <xs:enumeration value="DEACTIVATE_SERVICE"/>
    <xs:enumeration value="SUBSCRIBE_EVENT"/>
    <xs:enumeration value="UNSUBSCRIBE_EVENT"/>
    <xs:enumeration value="CONFIGURE_DEVICE"/>
    <xs:enumeration value="CHECK_STATUS"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="statusType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PROVISIONING"/>
    <xs:enumeration value="PROVISIONED"/>
    <xs:enumeration value="INSTALLED"/>
    <xs:enumeration value="UNINSTALLED"/>
    <xs:enumeration value="ACTIVATING"/>
    <xs:enumeration value="ACTIVATED"/>
    <xs:enumeration value="ACTIVATION_IN_PROGRESS"/>
    <xs:enumeration value="ACTIVE"/>
    <xs:enumeration value="DEACTIVATING"/>
    <xs:enumeration value="DEACTIVATED"/>
    <xs:enumeration value="DEACTIVE"/>
  </xs:restriction>
</xs:simpleType>

```

[0469]

```

<xs:enumeration value="REPLACING"/>
<xs:enumeration value="REPLACED"/>
<xs:enumeration value="DELETING"/>
<xs:enumeration value="DELETED"/>
<xs:enumeration value="SUBSCRIBED"/>
<xs:enumeration value="UNSUBSCRIBED"/>
<xs:enumeration value="ERROR"/>
<xs:enumeration value="CONFIGURING"/>
<xs:enumeration value="CONFIGURED"/>
<xs:enumeration value="CHECK_IN_PROGRESS"/>
<xs:enumeration value="CHECKED"/>
<xs:enumeration value="CHECK_FAILED"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="eventType">
<xs:restriction base="xs:string">
<xs:enumeration value="SAMPLING"/>
<xs:enumeration value="LOGGING"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
</types>
<message name="DataAccess_unsubscribe">
<part name="request" type="tns:subscribeRequest">
</part>
</message>
<message name="DataAccess_findEventDetailsByDeviceId">
<part name="request" type="tns:eventRequest">
</part>
</message>
<message name="DataAccess_findEventDetail">
<part name="request" type="tns:eventDetailRequest">
</part>
</message>
<message name="DataAccess_findEventByDeviceId">
<part name="request" type="tns:eventRequest">
</part>
</message>
<message name="DataAccess_findCommandByDeviceId">
<part name="request" type="tns:commandRequest">
</part>
</message>
<message name="DataAccess_findCommandByDeviceIdResponse">
<part name="response" type="tns:commandResponse">
</part>
</message>
<message name="DataAccess_subscribe">
<part name="request" type="tns:subscribeRequest">
</part>
</message>
<message name="DataAccess_findEventByDeviceIdResponse">
<part name="response" type="tns:eventResponse">
</part>
</message>
<message name="DataAccess_subscribeResponse">
<part name="response" type="tns:dataAccessResponse">
</part>
</message>
<message name="DataAccess_findEventDetailsByDeviceIdResponse">

```

[0470]

```

<part name="response" type="tns:eventDetailResponse">
</part>
</message>
<message name="DataAccess_findEventDetailResponse">
<part name="response" type="tns:eventDetailResponse">
</part>
</message>
<message name="DataAccess_unsubscribeResponse">
<part name="response" type="tns:dataAccessResponse">
</part>
</message>
<portType name="DataAccess">
<operation name="findCommandByDeviceId" parameterOrder="request">
<input message="tns:DataAccess_findCommandByDeviceId">
</input>
<output message="tns:DataAccess_findCommandByDeviceIdResponse">
</output>
</operation>
<operation name="findEventByDeviceId" parameterOrder="request">
<input message="tns:DataAccess_findEventByDeviceId">
</input>
<output message="tns:DataAccess_findEventByDeviceIdResponse">
</output>
</operation>
<operation name="findEventDetail" parameterOrder="request">
<input message="tns:DataAccess_findEventDetail">
</input>
<output message="tns:DataAccess_findEventDetailResponse">
</output>
</operation>
<operation name="findEventDetailsByDeviceId" parameterOrder="request">
<input message="tns:DataAccess_findEventDetailsByDeviceId">
</input>
<output message="tns:DataAccess_findEventDetailsByDeviceIdResponse">
</output>
</operation>
<operation name="subscribe" parameterOrder="request">
<input message="tns:DataAccess_subscribe">
</input>
<output message="tns:DataAccess_subscribeResponse">
</output>
</operation>
<operation name="unsubscribe" parameterOrder="request">
<input message="tns:DataAccess_unsubscribe">
</input>
<output message="tns:DataAccess_unsubscribeResponse">
</output>
</operation>
</portType>
<binding name="DataAccessBinding" type="tns:DataAccess">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="findCommandByDeviceId">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>

```

[0471]

```
</operation>
<operation name="findEventByDeviceId">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </output>
</operation>
<operation name="findEventDetail">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </output>
</operation>
<operation name="findEventDetailsByDeviceId">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </output>
</operation>
<operation name="subscribe">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </output>
</operation>
<operation name="unsubscribe">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
  </output>
</operation>
</binding>
<service name="DataAccessInterface">
  <port name="DataAccessPort" binding="tns:DataAccessBinding">
    <soap:address location="http://192.168.83.208:18080/M2M/InfoBrokerDataAccess"/>
  </port>
</service>
</definitions>
```

[0472] 表 66 示出了 DeviceManager.wsdl 列表的示例。

[0473]

表 66 - DeviceManager.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DeviceManagerInterface"
targetNamespace="http://services.infobroker.m2m.amos.company.com"
xmlns:tns="http://services.infobroker.m2m.amos.company.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
<xs:schema targetNamespace="http://services.infobroker.m2m.amos.company.com"
version="1.0" xmlns:tns="http://services.infobroker.m2m.amos.company.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xs:complexType name="deviceManagerRequest">
<xs:sequence>
<xs:element minOccurs="0" name="configurationId" type="xs:int"/>
<xs:element name="customerId" type="xs:string"/>
<xs:element name="deviceId" type="xs:string"/>
<xs:element name="serviceId" type="xs:int"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="deviceManagerResponse">
<xs:sequence>
<xs:element minOccurs="0" name="errorCode" type="xs:int"/>
<xs:element minOccurs="0" name="errorDesc" type="xs:string"/>
<xs:element minOccurs="0" name="statusCode" type="xs:int"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="removeDeviceRequest">
<xs:sequence>
<xs:element name="customerId" type="xs:string"/>
<xs:element name="deviceId" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="replaceDeviceRequest">
<xs:sequence>
<xs:element name="customerId" type="xs:string"/>
<xs:element name="oldDevice" type="xs:string"/>
<xs:element name="newDevice" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="insertDeviceRequest">
<xs:sequence>
<xs:element name="customerId" type="xs:string"/>
<xs:element name="deviceId" type="xs:string"/>
<xs:element name="deviceType" type="xs:string"/>
<xs:element name="groupId" type="xs:string"/>
<xs:element minOccurs="0" name="iccid" type="xs:string"/>
<xs:element minOccurs="0" name="ipAddress" type="xs:string"/>
<xs:element minOccurs="0" name="msisdn" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="deviceManagerConfigurationRequest">
<xs:sequence>
<xs:element name="arrayOfDetail" type="tns:arrayOfDetail"/>

```

[0474]

```

<xs:element name="customerId" type="xs:string"/>
<xs:element name="deviceId" type="xs:string"/>
<xs:element name="serviceId" type="xs:int"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="arrayOfDetail">
<xs:sequence>
<xs:element maxOccurs="unbounded" name="details" type="tns:detail"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="detail">
<xs:sequence>
<xs:element maxOccurs="unbounded" name="attributes" type="tns:attribute"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="attribute">
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="value" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="DeviceManager_replaceDeviceResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_checkStatusResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_deactivate">
<part name="request" type="tns:deviceManagerRequest">
</part>
</message>
<message name="DeviceManager_activate">
<part name="request" type="tns:deviceManagerRequest">
</part>
</message>
<message name="DeviceManager_configure">
<part name="request" type="tns:deviceManagerRequest">
</part>
</message>
<message name="DeviceManager_removeDeviceResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_configureResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_deactivateResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_insertDevice">
<part name="request" type="tns:insertDeviceRequest">
</part>

```

[0475]

```

</message>
<message name="DeviceManager_insertDeviceResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_replaceDevice">
<part name="request" type="tns:replaceDeviceRequest">
</part>
</message>
<message name="DeviceManager_configureByDetailsResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_removeDevice">
<part name="request" type="tns:removeDeviceRequest">
</part>
</message>
<message name="DeviceManager_configureByDetails">
<part name="request" type="tns:deviceManagerConfigurationRequest">
</part>
</message>
<message name="DeviceManager_activateResponse">
<part name="response" type="tns:deviceManagerResponse">
</part>
</message>
<message name="DeviceManager_checkStatus">
<part name="request" type="tns:deviceManagerRequest">
</part>
</message>
<portType name="DeviceManager">
<operation name="activate" parameterOrder="request">
<input message="tns:DeviceManager_activate">
</input>
<output message="tns:DeviceManager_activateResponse">
</output>
</operation>
<operation name="checkStatus" parameterOrder="request">
<input message="tns:DeviceManager_checkStatus">
</input>
<output message="tns:DeviceManager_checkStatusResponse">
</output>
</operation>
<operation name="configure" parameterOrder="request">
<input message="tns:DeviceManager_configure">
</input>
<output message="tns:DeviceManager_configureResponse">
</output>
</operation>
<operation name="configureByDetails" parameterOrder="request">
<input message="tns:DeviceManager_configureByDetails">
</input>
<output message="tns:DeviceManager_configureByDetailsResponse">
</output>
</operation>
<operation name="deactivate" parameterOrder="request">
<input message="tns:DeviceManager_deactivate">
</input>
<output message="tns:DeviceManager_deactivateResponse">
</output>

```

[0476]

```
</operation>
<operation name="insertDevice" parameterOrder="request">
<input message="tns:DeviceManager_insertDevice">
</input>
<output message="tns:DeviceManager_insertDeviceResponse">
</output>
</operation>
<operation name="removeDevice" parameterOrder="request">
<input message="tns:DeviceManager_removeDevice">
</input>
<output message="tns:DeviceManager_removeDeviceResponse">
</output>
</operation>
<operation name="replaceDevice" parameterOrder="request">
<input message="tns:DeviceManager_replaceDevice">
</input>
<output message="tns:DeviceManager_replaceDeviceResponse">
</output>
</operation>
</portType>
<binding name="DeviceManagerBinding" type="tns:DeviceManager">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="activate">
<soap:operation soapAction="">
<input>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="checkStatus">
<soap:operation soapAction="">
<input>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="configure">
<soap:operation soapAction="">
<input>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="configureByDetails">
<soap:operation soapAction="">
<input>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal" namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="deactivate">
```

[0477]

```

<soap:operation soapAction="">
  <input>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
  </input>
  <output>
    <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
  </output>
</operation>
<operation name="insertDevice">
  <soap:operation soapAction="">
    <input>
      <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
    </output>
  </operation>
<operation name="removeDevice">
  <soap:operation soapAction="">
    <input>
      <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
    </output>
  </operation>
<operation name="replaceDevice">
  <soap:operation soapAction="">
    <input>
      <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://services.infobroker.m2m.amos.com"/>
    </output>
  </operation>
</binding>
<service name="DeviceManagerInterface">
  <port name="DeviceManagerPort" binding="tns:DeviceManagerBinding">
    <soap:address location="http://127.0.0.1:8080/M2M/InfoBrokerDeviceManager"/>
  </port>
</service>
</definitions>

```

[0478] 表 67 示出了 EventHandler.wsd1 列表的示例。

表 67 - EventHandler.wsd1

[0479]

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EventHandlerInterface"
  targetNamespace="http://services.infobroker.m2m.amos.com"
  xmlns:tns="http://services.infobroker.m2m.amos.com"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <xs:schema

```

[0480]

```

targetNamespace="http://services.infobroker.s2m.amos.company.com"
version="1.0"
xmlns:tns="http://services.infobroker.s2m.amos.company.com"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
<xs:complexType name="addRuleRequest">
<xs:sequence>
<xs:element minOccurs="0" name="actionName" type="xs:string"/>
<xs:element name="customerId" type="xs:string"/>
<xs:element minOccurs="0" name="description" type="xs:string"/>
<xs:element maxOccurs="unbounded" name="devices" type="xs:string"/>
<xs:element name="expression" type="xs:string"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="parameters">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="entry">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="key" type="xs:string"/>
<xs:element minOccurs="0" name="value" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="serviceName" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="eventHandlerResponse">
<xs:sequence>
<xs:element minOccurs="0" name="errorCode" type="xs:int"/>
<xs:element minOccurs="0" name="errorDesc" type="xs:string"/>
<xs:element minOccurs="0" name="statusCode" type="xs:int"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="addActionRequest">
<xs:sequence>
<xs:element name="actionName" type="xs:string"/>
<xs:element name="parameters">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="entry">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="key" type="xs:string"/>
<xs:element minOccurs="0" name="value" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ruleName" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="fieldRequest">
<xs:sequence>

```

[0481]

```

<xs:element name="serviceName" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="fieldResponse">
<xs:complexContent>
<xs:extension base="tns:eventHandlerResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="fields"
nillable="true" type="xs:string"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="functionRequest">
<xs:sequence>
<xs:element name="serviceName" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="functionResponse">
<xs:complexContent>
<xs:extension base="tns:eventHandlerResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="functions"
nillable="true" type="xs:string"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="eventHandlerRequest">
<xs:sequence>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="actionResponse">
<xs:complexContent>
<xs:extension base="tns:eventHandlerResponse">
<xs:sequence>
<xs:element minOccurs="0" name="actionParams"
type="tns:listKeyListType"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="listKeyListType">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="action"
type="tns:keyListType"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="keyListType">
<xs:sequence>
<xs:element minOccurs="0" name="name" type="xs:string"/>
<xs:element maxOccurs="unbounded" minOccurs="0" name="param"
type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="findRuleRequest">
<xs:sequence>

```

[0482]

```

<xs:element minOccurs="0" name="customerId" type="xs:string"/>
<xs:element minOccurs="0" name="deviceId" type="xs:string"/>
<xs:element minOccurs="0" name="serviceName" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="findRuleResponse">
<xs:complexContent>
<xs:extension base="tns:eventHandlerResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="rules"
nillable="true" type="tns:rule"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="rule">
<xs:sequence>
<xs:element minOccurs="0" name="creationDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="description" type="xs:string"/>
<xs:element minOccurs="0" name="expression" type="xs:string"/>
<xs:element minOccurs="0" name="lastModifyDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="name" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="alarmRequest">
<xs:sequence>
<xs:element minOccurs="0" name="baseId" type="xs:int"/>
<xs:element minOccurs="0" name="deviceId" type="xs:string"/>
<xs:element minOccurs="0" name="from" type="xs:dateTime"/>
<xs:element minOccurs="0" name="ruleName" type="xs:string"/>
<xs:element minOccurs="0" name="serviceName" type="xs:string"/>
<xs:element minOccurs="0" name="statusType" type="tns:statusType"/>
<xs:element minOccurs="0" name="to" type="xs:dateTime"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="alarmResponse">
<xs:complexContent>
<xs:extension base="tns:eventHandlerResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="alarms"
nillable="true" type="tns:alarm"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="alarm">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="alarmhistories"
nillable="true" type="tns:alarmhistory"/>
<xs:element name="baseId" type="xs:int"/>
<xs:element minOccurs="0" name="creationDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="lastModifyDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="ruleName" type="xs:string"/>
<xs:element minOccurs="0" name="statusType" type="tns:statusType"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="alarmhistory">
<xs:sequence>

```

[0483]


```

<xs:element minOccurs="0" name="creationDate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="statusType" type="tns:statusType"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="removeRuleRequest">
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="activeDeactiveRuleRequest">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="devices"
nillable="true" type="xs:string"/>
<xs:element name="ruleName" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="modifyRuleRequest">
<xs:sequence>
<xs:element name="expression" type="xs:string"/>
<xs:element name="ruleName" type="xs:string"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="statusType">
<xs:restriction base="xs:string">
<xs:enumeration value="ACTIVE"/>
<xs:enumeration value="DEACTIVE"/>
<xs:enumeration value="CREATED"/>
<xs:enumeration value="LOADED"/>
<xs:enumeration value="EXECUTED"/>
<xs:enumeration value="ERROR"/>
<xs:enumeration value="CLOSED"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
</types>
<message name="EventHandler_modifyRuleExpressionResponse">
<part name="response" type="tns:eventHandlerResponse">
</part>
</message>
<message name="EventHandler_removeRuleResponse">
<part name="response" type="tns:eventHandlerResponse">
</part>
</message>
<message name="EventHandler_removeRule">
<part name="request" type="tns:removeRuleRequest">
</part>
</message>
<message name="EventHandler_modifyRuleExpression">
<part name="request" type="tns:modifyRuleRequest">
</part>
</message>
<message name="EventHandler_removeAction">
<part name="request" type="tns:addActionRequest">
</part>
</message>
<message name="EventHandler_findFunctions">
<part name="request" type="tns:functionRequest">

```

[0484]

```
</part>
</message>
<message name="EventHandler_removeActionResponse">
  <part name="response" type="tns:eventHandlerResponse">
  </part>
</message>
<message name="EventHandler_findFieldsResponse">
  <part name="response" type="tns:fieldResponse">
  </part>
</message>
<message name="EventHandler_addAction">
  <part name="request" type="tns:addActionRequest">
  </part>
</message>
<message name="EventHandler_addActionResponse">
  <part name="response" type="tns:eventHandlerResponse">
  </part>
</message>
<message name="EventHandler_findActionsAndParamiters">
  <part name="request" type="tns:eventHandlerRequest">
  </part>
</message>
<message name="EventHandler_findActionsAndParamitersResponse">
  <part name="response" type="tns:actionResponse">
  </part>
</message>
<message name="EventHandler_findFields">
  <part name="request" type="tns:fieldRequest">
  </part>
</message>
<message name="EventHandler_activeRule">
  <part name="request" type="tns:activeDeactiveRuleRequest">
  </part>
</message>
<message name="EventHandler_findAlarms">
  <part name="request" type="tns:alarmRequest">
  </part>
</message>
<message name="EventHandler_findAlarmsResponse">
  <part name="response" type="tns:alarmResponse">
  </part>
</message>
<message name="EventHandler_findRules">
  <part name="request" type="tns:findRuleRequest">
  </part>
</message>
<message name="EventHandler_findFunctionsResponse">
  <part name="response" type="tns:functionResponse">
  </part>
</message>
<message name="EventHandler_addRuleResponse">
  <part name="response" type="tns:eventHandlerResponse">
  </part>
</message>
<message name="EventHandler_activeRuleResponse">
  <part name="response" type="tns:eventHandlerResponse">
  </part>
</message>
<message name="EventHandler_addRule">
  <part name="request" type="tns:addRuleRequest">
```

[0485]

```
</part>
</message>
<message name="EventHandler_deactiveRule">
<part name="request" type="tns:activeDeactiveRuleRequest">
</part>
</message>
<message name="EventHandler_findRulesResponse">
<part name="response" type="tns:findRuleResponse">
</part>
</message>
<message name="EventHandler_deactiveRuleResponse">
<part name="response" type="tns:eventHandlerResponse">
</part>
</message>
<portType name="EventHandler">
<operation name="activeRule" parameterOrder="request">
<input message="tns:EventHandler_activeRule">
</input>
<output message="tns:EventHandler_activeRuleResponse">
</output>
</operation>
<operation name="addAction" parameterOrder="request">
<input message="tns:EventHandler_addAction">
</input>
<output message="tns:EventHandler_addActionResponse">
</output>
</operation>
<operation name="addRule" parameterOrder="request">
<input message="tns:EventHandler_addRule">
</input>
<output message="tns:EventHandler_addRuleResponse">
</output>
</operation>
<operation name="deactiveRule" parameterOrder="request">
<input message="tns:EventHandler_deactiveRule">
</input>
<output message="tns:EventHandler_deactiveRuleResponse">
</output>
</operation>
<operation name="findActionsAndParamiters" parameterOrder="request">
<input message="tns:EventHandler_findActionsAndParamiters">
</input>
<output message="tns:EventHandler_findActionsAndParamitersResponse">
</output>
</operation>
<operation name="findAlarms" parameterOrder="request">
<input message="tns:EventHandler_findAlarms">
</input>
<output message="tns:EventHandler_findAlarmsResponse">
</output>
</operation>
<operation name="findFields" parameterOrder="request">
<input message="tns:EventHandler_findFields">
</input>
<output message="tns:EventHandler_findFieldsResponse">
</output>
</operation>
<operation name="findFunctions" parameterOrder="request">
<input message="tns:EventHandler_findFunctions">
</input>
```

[0486]

```

<output message="tns:EventHandler_findFunctionsResponse">
</output>
</operation>
<operation name="findRules" parameterOrder="request">
<input message="tns:EventHandler_findRules">
</input>
<output message="tns:EventHandler_findRulesResponse">
</output>
</operation>
<operation name="modifyRuleExpression" parameterOrder="request">
<input message="tns:EventHandler_modifyRuleExpression">
</input>
<output message="tns:EventHandler_modifyRuleExpressionResponse">
</output>
</operation>
<operation name="removeAction" parameterOrder="request">
<input message="tns:EventHandler_removeAction">
</input>
<output message="tns:EventHandler_removeActionResponse">
</output>
</operation>
<operation name="removeRule" parameterOrder="request">
<input message="tns:EventHandler_removeRule">
</input>
<output message="tns:EventHandler_removeRuleResponse">
</output>
</operation>
</portType>
<binding name="EventHandlerBinding" type="tns:EventHandler">
<soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="activeRule">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="addAction">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="addRule">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>

```

[0487]

```
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="deactiveRule">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="findActionsAndParamiters">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="findAlarms">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="findFields">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="findFunctions">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="findRules">
```

[0488]

```

<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="modifyRuleExpression">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="removeAction">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="removeRule">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
</binding>
<service name="EventHandlerInterface">
<port name="EventHandlerPort" binding="tns:EventHandlerBinding">
<soap:address
location="http://127.0.0.1:8080/M2M/InfoBrokerEventHandler"/>
</port>
</service>
</definitions>

```

[0489] 表 68 示出了 SystemConfiguration.wsdl 列表的示例。

[0490]

表 68 - SystemConfiguration.wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="SystemConfigurationInterface"

```

[0491]

```

targetNamespace="http://services.infobroker.m2m.amos.company.com"
xmlns:tns="http://services.infobroker.m2m.amos.company.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
<types>
<xs:schema
targetNamespace="http://services.infobroker.m2m.amos.company.com"
version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexType name="customerServiceRequest">
<xs:sequence>
<xs:element name="customerId" type="xs:string"/>
<xs:element name="serviceId" type="xs:int"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="systemConfigurationResponse">
<xs:sequence>
<xs:element minOccurs="0" name="errorCode" type="xs:int"/>
<xs:element minOccurs="0" name="errorDesc" type="xs:string"/>
<xs:element minOccurs="0" name="statusCode" type="xs:int"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="groupProjectRequest">
<xs:sequence>
<xs:element name="groupId" type="xs:int"/>
<xs:element name="projectId" type="xs:int"/>
<xs:element name="transactionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message
name="SystemConfiguration_deassociateGroupFromProjectResponse">
<part name="response" type="tns:systemConfigurationResponse">
</part>
</message>
<message name="SystemConfiguration_customerDomainSubscription">
<part name="request" type="tns:customerServiceRequest">
</part>
</message>
<message name="SystemConfiguration_deassociateGroupFromProject">
<part name="request" type="tns:groupProjectRequest">
</part>
</message>
<message
name="SystemConfiguration_customerDomainUnsubscriptionResponse">
<part name="response" type="tns:systemConfigurationResponse">
</part>
</message>
<message name="SystemConfiguration_customerDomainUnsubscription">
<part name="request" type="tns:customerServiceRequest">
</part>
</message>
<message name="SystemConfiguration_customerDomainSubscriptionResponse">
<part name="response" type="tns:systemConfigurationResponse">
</part>
</message>
<message name="SystemConfiguration_associateGroupToProject">

```

[0492]

```

<part name="request" type="tns:groupProjectRequest">
</part>
</message>
<message name="SystemConfiguration_associateGroupToProjectResponse">
<part name="response" type="tns:systemConfigurationResponse">
</part>
</message>
<portType name="SystemConfiguration">
<operation name="associateGroupToProject" parameterOrder="request">
<input message="tns:SystemConfiguration_associateGroupToProject">
</input>
<output
message="tns:SystemConfiguration_associateGroupToProjectResponse">
</output>
</operation>
<operation name="customerDomainSubscription" parameterOrder="request">
<input message="tns:SystemConfiguration_customerDomainSubscription">
</input>
<output
message="tns:SystemConfiguration_customerDomainSubscriptionResponse">
</output>
</operation>
<operation name="customerDomainUnsubscription" parameterOrder="request">
<input message="tns:SystemConfiguration_customerDomainUnsubscription">
</input>
<output
message="tns:SystemConfiguration_customerDomainUnsubscriptionResponse">
</output>
</operation>
<operation name="deassociateGroupFromProject" parameterOrder="request">
<input message="tns:SystemConfiguration_deassociateGroupFromProject">
</input>
<output
message="tns:SystemConfiguration_deassociateGroupFromProjectResponse">
</output>
</operation>
</portType>
<binding name="SystemConfigurationBinding"
type="tns:SystemConfiguration">
<soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="associateGroupToProject">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="customerDomainSubscription">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>

```

[0493]


```

<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="customerDomainUnsubscription">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
<operation name="deassociateGroupFromProject">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</input>
<output>
<soap:body use="literal"
namespace="http://services.infobroker.m2m.amos.company.com"/>
</output>
</operation>
</binding>
<service name="SystemConfigurationInterface">
<port name="SystemConfigurationPort"
binding="tns:SystemConfigurationBinding">
<soap:address
location="http://127.0.0.1:8080/M2M/InfoBrokerSystemConfiguration"/>
</port>
</service>
</definitions>

```

[0494] 平台 102 可支持的示例性服务的一些特征在下面给出：

[0495] 服务：家庭控制与智能计量

[0496] 1. 终端用户通过垂直门户加入住宅建筑产品并使用移动支付管理产品支付订阅。

[0497] 2. 第三方服务提供者或用户配置和升级 / 降级家庭控制器功能。

[0498] 3. 终端用户将控制器应用下载到移动电话上，该应用实现软面板集成信息（计量，状态），控制（打开 / 关闭），或其他功能。

[0499] 具有自己移动电话的终端用户将能够：控制能源成本；优化能量节省，例如，为了根据人在或不在房间保持恒定的光的灯调光器；保持定义阈值下的环境值（温度，光，空气质量）以及接收告警；为每个设备监控实时状态，为异常值接收警告；检测并回应房间中人的存在，以及其他功能。

[0500] 服务：企业建筑的控制与安全

[0501] 1. 企业通过垂直门户加入住宅控制与安全产品，并使用移动支付管理产品支付订阅。

[0502] 2. 服务提供者能够配置和升级 / 降级建筑控制器功能。

[0503] 3. 企业建筑将用电缆接入以进行注册：

[0504] 雇员访问，可考虑为徽章 / 面部识别 / NFC 或无线射频识别技术 (RFID) 条。

- [0505] 能源消耗,和阈值相比以减少浪费并管理告警。
- [0506] 环境值(温度,光线,空气质量)以将其保持在定义的阈值以下。
- [0507] 用于打开/关闭灯组,供热站的日程表。
- [0508] 优化能源节省。
- [0509] 企业管理部门管理和监控所有控制和安全设备的使用,例如移动电话。
- [0510] 服务:远程健康诊断
- [0511] 1. 终端用户/医生在病人家里通过数字笔技术或通过健康诊断垂直平台配合或提供医院治疗。
- [0512] 2. 终端用户/病人能够测量向架构 100 报告的如下设备值:
- [0513] 血糖测计仪
- [0514] 血压
- [0515] 体重
- [0516] 心跳,或者其他健康数据。
- [0517] 3. 终端用户/病人将数据发送到医院数据中心。
- [0518] 4. 终端用户/医生基于最后的测量将正确的疗法发送给病人。
- [0519] 服务:药房自动化
- [0520] 1. 基于健康医疗情境下的诊断报告或在医疗检查期间,医生通过移动设备应用准备药方。
- [0521] 2. 通过医疗虚拟凭证创建“病人条形码”将药方发送给病人。
- [0522] 3. 病人通过医生指定的医疗虚拟凭证在药房请求药品。
- [0523] 4. 病人使用销售设备的遥控移动点为药品付款。
- [0524] 服务:健康授权
- [0525] 1. 健身房创建技术环境,并且服务自动化终端用户到设备的通路。
- [0526] 2. 终端用户通过零售移动门户订阅服务并接收用于移动应用下载的 SMS。
- [0527] 3. 用户将以“使用时支付”或订阅方式付款。通过移动资金管理服务管理支付。
- [0528] 4. 针对订阅模式,终端用户通过其移动应用接收通过条形码启用的服务许可。
- [0529] 5. 终端用户通过其个人服务许可访问服务(例如,健身设备);智能健身设备识别终端用户,跟踪训练表现并建议如何改进训练。
- [0530] 6. 个人教练能够访问终端用户的记录以正确调整训练。
- [0531] 7. 终端用户能够通过健身房中的售货亭访问训练表以及训练表现。
- [0532] 8. 通过健康诊断设备,监控体重,心跳,以及血压;当由于训练达到临界值时,通过 SMS 发信号给医生和个人教练两者;更新医疗记录和训练表两者。
- [0533] 9. 健身房管理员能够给终端用户分配会员积分和折扣/礼券。
- [0534] 服务:集中策略管理
- [0535] 1. 终端用户通过移动数据采集或垂直保险门户参与用于车辆(CAR)和家庭(HOME)和生活(LIFE)的保险策略。
- [0536] 2. 终端用户访问保险门户并选择每个策略段更便宜的策略。
- [0537] CAR 策略:由于“行驶时支付”或其他选项可以减少策略花费;
- [0538] HOME 策略:由于家里的安全功能可以减少策略花费;

- [0539] LIFE 策略 :由于和用户相关的个人追踪或监控设备可以减少策略花费。
- [0540] 3. 保险公司将基于选定策略激活控制设备。
- [0541] 4. 为每个超过的阈值建议保险平台以用于确定支付的下面的费用并通知到终端用户。
- [0542] 5. 对于每个获得赔付的客户,保险公司指定用于采购新策略的虚拟凭证折扣。
- [0543] 服务 :车队管理
- [0544] 1. 监控整个车队的业务经营者参与该服务以 :
- [0545] 实时追踪每辆车 (例如, GPS 位置, 速度, 燃料, 传感器)
- [0546] 管理盗窃 / 事故警告
- [0547] 设定地理围栏区域并管理相关警告
- [0548] 例如在工作调度方面支持驾驶员
- [0549] 当火车离开预定目标道路 / 方向时获取警告
- [0550] 2. 每辆车提供的数据收发器在预定时间间隔发送, 使得 M2M 平台能够产生车队管理服务。
- [0551] 3. 可以向整个车队或一个驾驶员递送 SMS 消息
- [0552] 服务 :交货凭证自动化
- [0553] 1. 物流公司用自己的车队递送货物并且其需要用交货凭证证实 / 自动化 / 数字化每个货运 ;每辆车都配备数据收发器, 和车辆传感器, GPS 天线, 数字笔相连。
- [0554] 2. 驾驶员将货物运送给收货人并且让收货人签署交货文档 ;数据收发器将数据发送回移动平台。
- [0555] 3. 公司后台可以 :
- [0556] 实施监控装运 (GPS 位置, 交货状态)
- [0557] 查看 / 验证 / 转发收货人使用驾驶员的数字笔签署的交货文档。
- [0558] 监控反诈骗报告
- [0559] 监控警告 / 警告驾驶员动作 (平均和最高速度, 事故, 等)
- [0560] 为车队设置地理围栏区域
- [0561] 服务 :智能购物
- [0562] 1. 终端用户通过移动数据采集和 / 或零售商站点订阅服务 ;在订阅后, 终端用户接收用于移动应用下载的 SMS 并且零售管理员在架构 100 上提供支持应用。
- [0563] 2. 终端用户管理其自己的家庭设备以配置在其手持应用上可见的“购物规则”。
- [0564] 3. 在商店的终端用户然后通过产品以及现金注册遥控阅读器上的遥控接口计算消费产品的数量。
- [0565] 4. 终端用户使用其电话上的移动资金管理遥控应用支付购物车并且手机其会员卡上的点数。
- [0566] 5. 零售管理员将折扣凭证发送给终端用户电话。
- [0567] 6. 终端用户访问奖励目录以兑换经历不年个在下一次购物访问期间花费任何凭证。
- [0568] 如下描述了网络控制台 266 的示例实现。
- [0569] 图 26 示出了用于架构 100 的示例网络控制台界面 2600。网络控制台界面 2600 可

以托管于平台使能器 204 中并且可以通过其他界面的网络平台界面暴露给第三方。网络控制台界面 2600 为每个客户提供客户表 2602。客户表 2602 包括客户链接列表 2604, 分层浏览链接 2606, 地理浏览链接 2608, 以及状态报告 2610。

[0570] 状态报告 2610 可以向客户显示运行或普通状态设备的数量和百分比, 或其他状态数据。链接 2604-2608 可以实现为超链接或其他指针。经过客户链接的点击, 例如, 网络控制台接口 2600 可以显示架构 100 提供给选定客户的服务填充的树视图对象。

[0571] 网络控制台视图 2600 还提供搜索界面 2612。搜索界面 2612 接受搜索样式, 包括用于查找架构 100 管理的资源的通配符。作为响应, 网络控制台视图 2600 找到匹配资源并生成搜索结果界面 3802, 例如图 38 示出了那样。网络控制台界面 2600 还包括引出高级搜索界面 3902 的高级搜索链接 2614, 例如图 39 示出了那样。

[0572] 网络控制台界面 2600 进一步包括引出如图 40 所示定期报告界面的定期报告链接 2614, 引出如图 42 所示实时报告界面的实时报告链接 2616, 引出如图 43 所示管理界面的管理链接 2616, 以及引出如图 48 所示警告界面的警告链接 2618。

[0573] 图 27 示出了示例树视图对象 2702。树视图对象 2702 显示向电信客户提供的服务列表。如图 27 所示的示例中, 树视图对象 2702 表示汽车跟踪服务 2704, 远程监控对象 2706, 以及其他提供给电信的服务。

[0574] 图 28 示出了扩展为列表项目的树视图对象 2702, 例如项目 TI-SPLB 2802。通常来说, 树视图对象 2702 填充有资源 (如移动设备, 软件, 或其他资源) 安装的站点, 或表示设备 (如移动设备, 在物理区域如城市中移动) 当前位置的站点。主窗口 2804 改变以表示属于项目和所选站点的设备列表。

[0575] 图 29 示出了树视图对象 2702 扩展了项目对象以表示属于该项目的站点 (如, 用于 Rome 2902 的站点)。设备列表 2904 通过列出和站点相关或属于站点的列表来响应项目和站点选择。设备列表 2904 可以包括设备标识符 2906, MSISDN 编号 2908, 站点位置或描述 2910, 设备状态 2912, 或其他信息。设备列表 2904 的每个条目, 如设备 ID 2906, 可以是设备更多详细状态信息的链接。

[0576] 图 30 示出了树视图对象 2702 进一步展开以表示特别和选定站点相关的设备 3002。如例所示, 设备列表 2904 可以更新以表示特定于选定站点的设备 3002。

[0577] 图 31 示出了树视图对象 2702 进一步展开项目对象以表示选定设备的详细状态信息链接 3102。在图 31 所示的例子中, 详细状态信息链接 3102 包括硬件状态链接, 订户身份模块链接, 以及应用状态链接。每个链接可以将指示主窗口 2804 显示硬件或链接表示的软件的详细状态。如图 31 所示, 细节状态可以包括如运行状态, 供电级别, 存储器使用, CPU 负载, 激活状态, 设备状态以及其他期望的状态信息的状态数据。可以将状态报告保存在文件中, 并采集当前或历史 (例如, 10 天的历史) 设备状态。

[0578] 图 32 示出了网络控制台界面 2600 可以包括用于指示系统保存任何设备生成的日志信息的文件的数据日志下载按钮 3202。

[0579] 图 33 示出了订户身份模块 (SIM) 详细状态窗口 3302。状态窗口 3302 可以表示任何期望的 SIM 状态信息, 例如 SIM 是否激活, 是否具有连通性, MSISDN 或其他状态数据。

[0580] 图 34 示出了应用模块详细状态窗口 3402。状态窗口 3402 可以表示任何期望的软件模块状态信息, 例如功能状态, 当前运行, 或其他状态信息。

[0581] 图 26 示出了网络控制台界面 2600 包括地理浏览链接 2608。当选择链接 2608 时,主窗口 2802 可变化以表示设备位置页面,如图 35 所示。

[0582] 在图 35 中,设备位置页面 3502 用设备图标 3504 表示资源方位的地理显示。设备位置页面 3502 包括服务过滤盒 3506。服务过滤盒 3506 中的录入指示网络控制台界面 2600 示出属于录入的服务的设备位置页面 3502 中的图标。将鼠标移动到图标上(或者进行不同的选择动作)将引起设备位置页面 3502 显示具有在给定资源或临近区域中资源的任意期望的资源状态信息的弹出箱。

[0583] 图 36 示出了在相同地理区域中(如同省,县,建筑,邮编,或其他区域)资源的资源状态窗口 3602 的示例。资源的选择将生成状态显示页面,例如图 37 所示。

[0584] 图 38 示出了列出匹配搜索标准的每个资源的搜索结果界面 3802 的示例。搜索结果界面 3802 可以提供资源选择器 3804 以及状态请求按钮 3806 以指示架构 100 运行实时查询以对于每个资源获取最新可用数据。最新状态更新域 3808 表示正在显示的状态信息的时间戳。

[0585] 图 39 示出了高级搜索界面 3902 的示例。高级检索界面 3902 为任意期望的搜索标准提供数据录入元件。作为示例,高级搜索界面 3902 包括用于客户名称,MSISDN,项目名称,以及其他标准的搜索标准。

[0586] 图 40 示出了定期报告界面 4002 的示例。界面 4002 包括用于定义任意期望报告的报告生成标准。在图 40 所示的示例中,界面 4002 为每天,每周,或每月日程安排提供报告计时选项 4004,包括在报告中的状态域 4006(例如设备 id 和项目),用于选择哪个设备添加加入报告的资源过滤器 4008,以及指定谁将通过邮件接收报告拷贝的接受者电子邮件地址域 4010。界面 4002 还表示之前定义的报告的已有报告列表 4012,其可以删除或改变。

[0587] 图 41 示出了将新定义的报告 4102 添加到已有报告列表 4012 中以用于以后的访问。

[0588] 图 42 示出了实时报告界面 4202 的示例。界面 4202 提供包括在报告中的状态域 4204(例如设备 id 和项目),选择哪个设备添加在报告中的资源过滤器 4206,以及指定状态的时间范围的日期范围过滤器 4208。根据界面 4202 设定的标准,基于提交按钮的激活生成实时报告。

[0589] 图 43 示出了管理界面 4302 的示例。一旦用户作为管理员登入,管理界面 4302 为管理资源和客户提供资源选择界面。例如,资源选择界面 4304 提供用于管理的客户选择 4306 和设备选择 4308。

[0590] 图 44 示出了来源于 IP 摄像头设备类型选择的管理页面 4402 的示例。界面 4304 生成匹配设备 4404 的列表,并且包括详细日程安排选项 4406 和分布式日程安排选项 4408。详细日程安排选项 4406 使得可以根据指定分钟,小时和日子在一个或多个设备上直接调度状态升级检查。分布式调度选项 4408 使得用户可以指定检查资源的时间范围,并且是否可以每天,每周,每月或依据其他进行检查。时间表检查按钮 4410 小时使得能够以详细方式或分布式方式两者之一设定调度频率的页面。

[0591] 图 45 示出了根据管理页面 4402 的客户选择,可以显示表示提供给选定客户的服务的数浏览对象 4502。可以使用上述树浏览对象 4502 继续指定的设备选择。

[0592] 图 46 示出了详细调度页面 4602。当详细调度选项 4406 选中时,管理界面可以显

示详细调度页面 4602。详细调度页面 4602 可以提供分钟,小时,一周中某天,或者其他调度条件的调度条目。

[0593] 图 47 示出了分布式调度页面 4702。当分布式调度选项 4408 选中时,管理界面可以显示分布式调度页面 4702。分布式调度页面 4702 可以为时间范围,频率或其他选项提供调度条目,这样状态检查可以使用指定频率在该时间范围内执行。

[0594] 图 48 示出了警告界面 4802 的示例。警告界面 4802 提供用于获取资源警告的资源选择界面 4804。例如,资源界面 4804 提供客户选择 4806 和设备选择 4808。

[0595] 图 49 示出了促进对于特定客户警告通信的警告配置界面 4902,通知信道选择器 4904 使得用户能够指定警告是否通过电子邮件, SMS,或其他信道发送。出于这样的原因,界面 4902 提供用于电子邮件地址和 SMS 号码的录入域 4906 和 4908。界面 4902 还使得警告能够基于警告的定时(例如,在调度检查之后,在实时检查之后,或当警告状态达到特定状态,例如“完成”或“出局”)发送给不同的接受者。

[0596] 上面的系统,模块,组件和逻辑可以以不同的形式实现。功能可以在一个系统中实现或跨多个系统功能分区的实现。作为另一个示例,模块,组件,系统和逻辑可以以计算机可执行指令或数据结构来实现,可存储于,分布在,或从许多不同类型的及其可读介质中读取。机器可读介质可包括 RAM, ROM, 硬盘, 软盘, CD-ROM, 信号, 例如从网络接收到或划分为几个部分并在多个跨网通信的分组中接收的信号。系统可以在软件,硬件,或软硬件结合体中实现。

[0597] 进一步地,系统可以使用额外的,不同的,或更少的组件加以实现。作为一个示例,可以用微处理器,微控制器, DSP, 专用集成电路 (ASIC), 程序指令, 分立模拟或数字逻辑,或其他类型电路或逻辑的结合体来实现处理器或任何其他逻辑,模块或组件。作为另一个示例,存储器可以是 DRAM, SRAM, 闪存或任何其他类型的存储器。系统可以分布在多个组件中,例如在多个处理器和存储器中,可选的包括多个分布式处理系统。逻辑,例如程序或电路,可以结合在或拆分在多个程序中,分布在多个存储器和处理器中,以及可以在功能库中或作为功能库实现,例如动态链接库 (DLL) 或其他共享库。

[0598] 组件和系统之间的接口例如核 SDP 可以包括传输控制协议 (TCP), 实时传输协议 (RTP) 或其他传输逻辑。网络网关可以基于因特网协议 v4, v6 (即 IPv4 或 IPv6) 或其他网络层协议路由信息。数据链路层可以包括有线或无线链接,例如 IEEE 802. 11, WiFi, WiMAX。异步传输模式 (ATM), 光纤分布数据接口 (FDDI), 以太网,或其他光纤,同轴电缆,双绞线或其他物理层上的数据链接层。

[0599] 系统和逻辑之间的接口以及系统中的模块可以以多种方式实现。例如,系统之间的接口可以是网络服务,简单对象访问协议,或企业服务总线接口。接口的其他例子包括消息传递,例如发布/订阅消息,共享存储,以及远程过程调用。

[0600] 在 SDP DS 中运行的硬件和软件平台可以广泛的变化。如例所示,断电可以运行 Windows CE(TM) 操作系统, JAVA ME(TM) 系统, Symbian(TM) 操作系统, Palm(TM) 操作系统。硬件平台可以和普通目的操作平台一起实现,例如从 Sun 微系统, HewlettPackard, 或国际商业机器 (International Business Machine, IBM), 以及运行中的 Unix, Windows(TM), Linux 或其他操作系统可获得的。

[0601] 由于已经描述了发明的多个实施方式,对本领域普通技术人员而言在本发明范围

内可以有更多实施方式和实现。因此,本发明除了按照所附的权利要求和其等同体之外不为其所限。

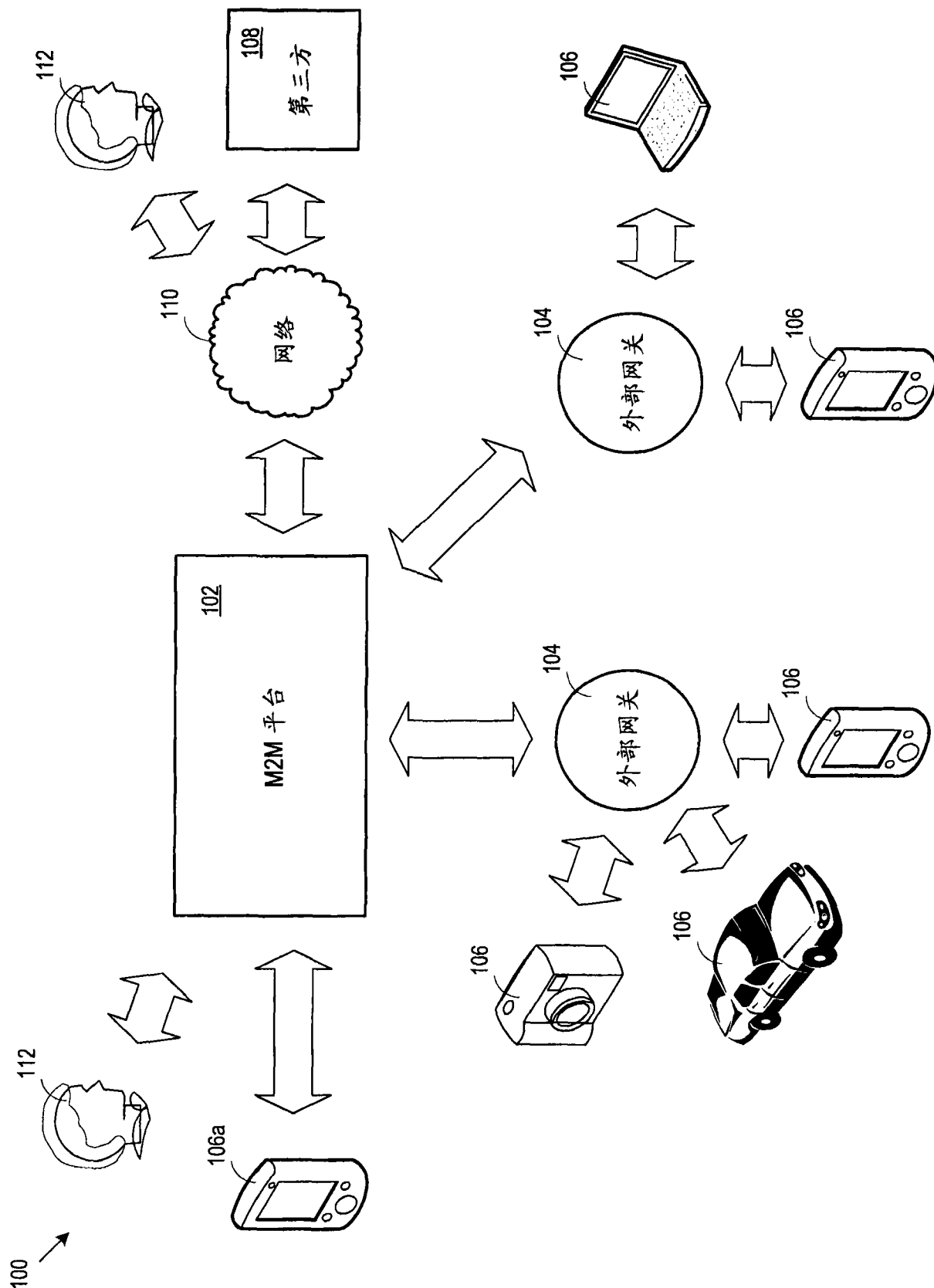


图 1

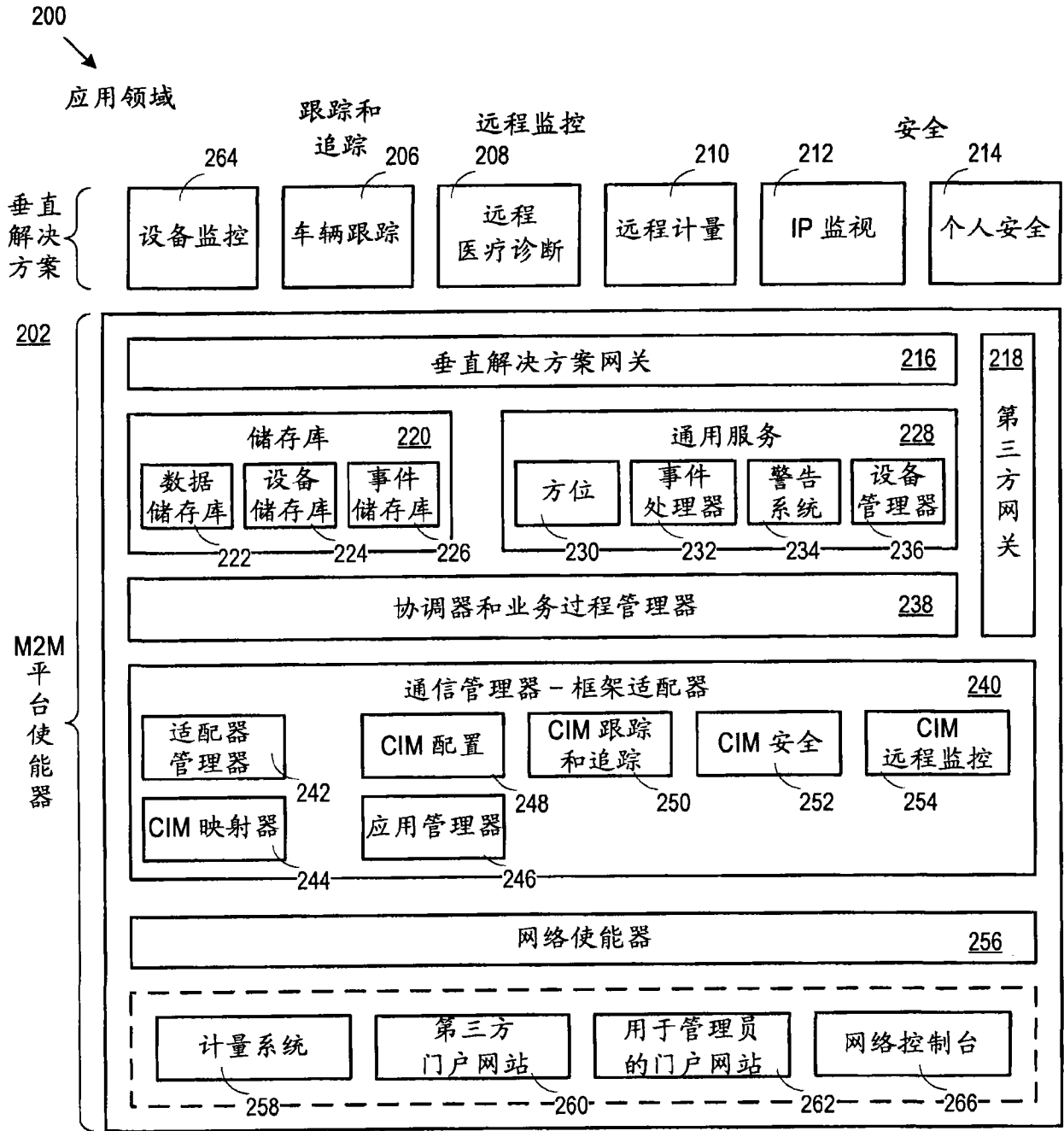


图 2

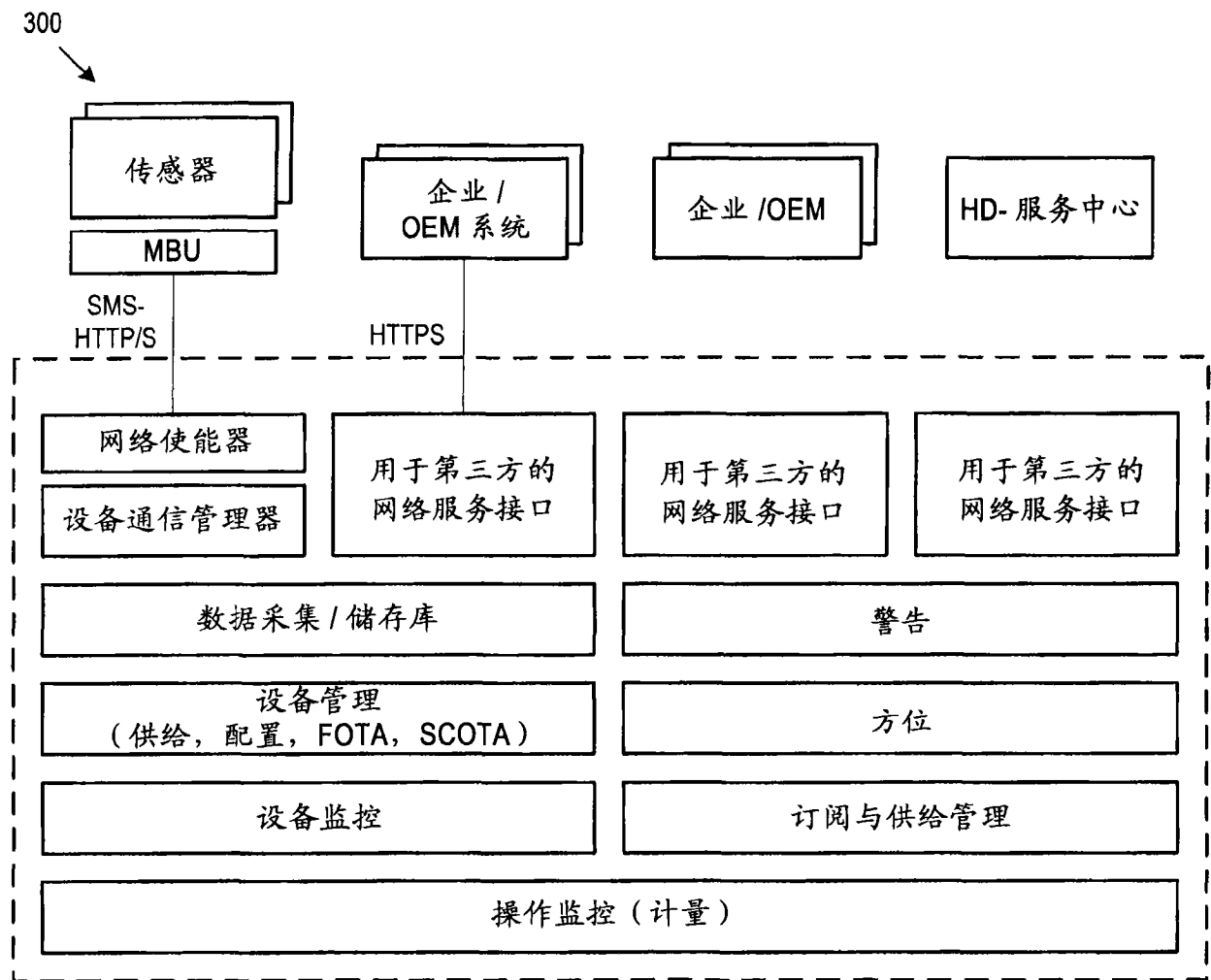


图 3

400
↙

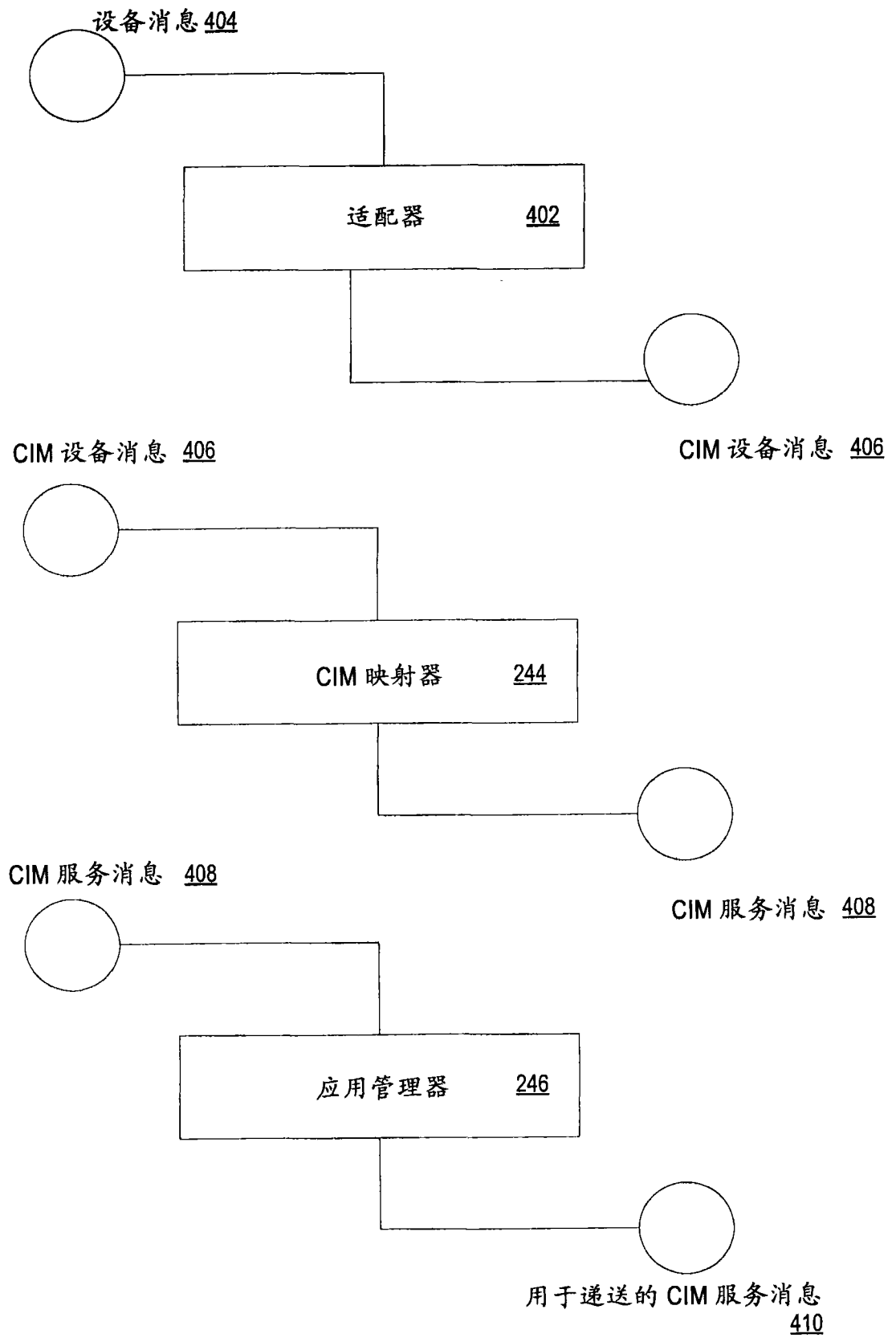


图 4

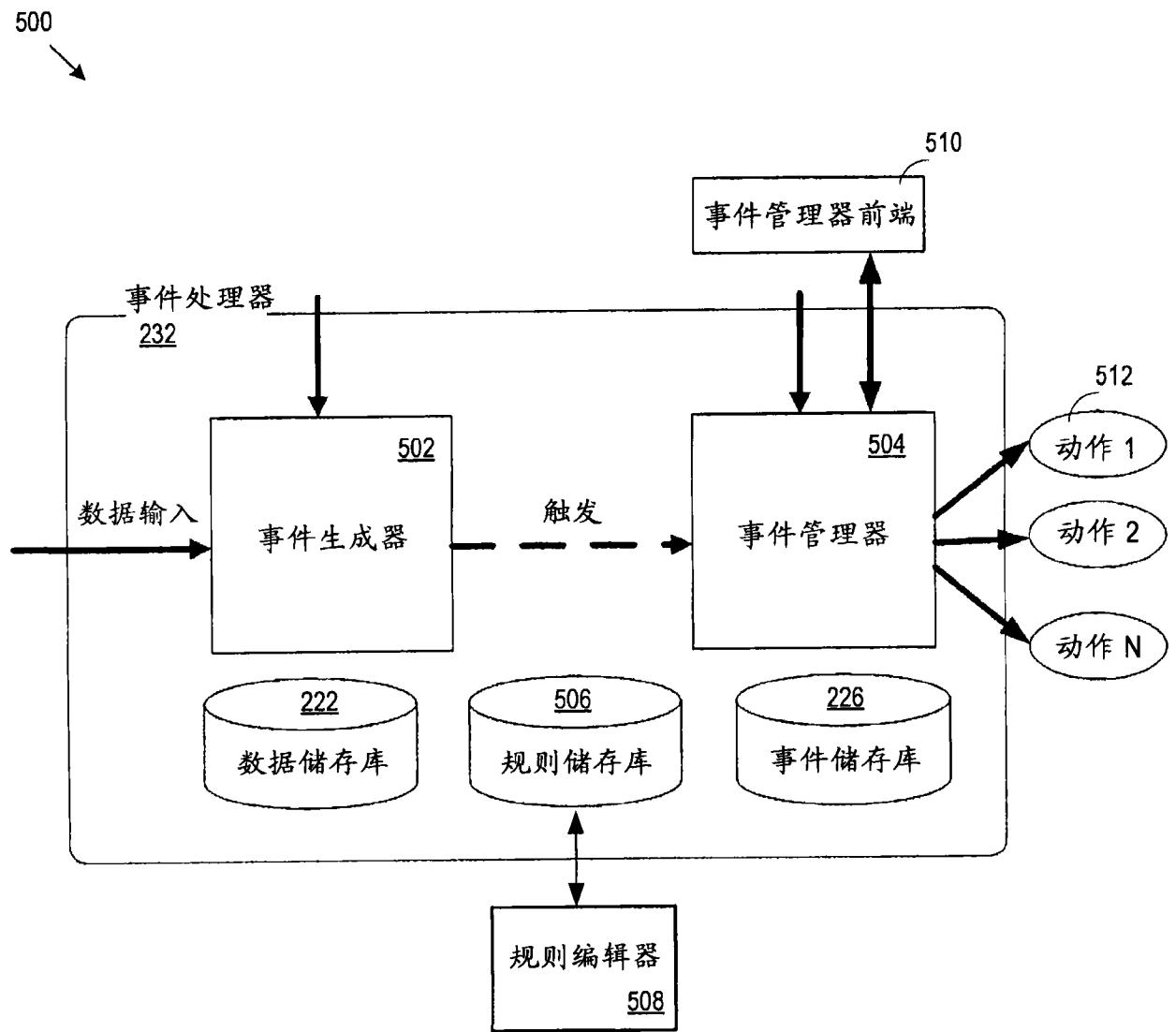


图 5

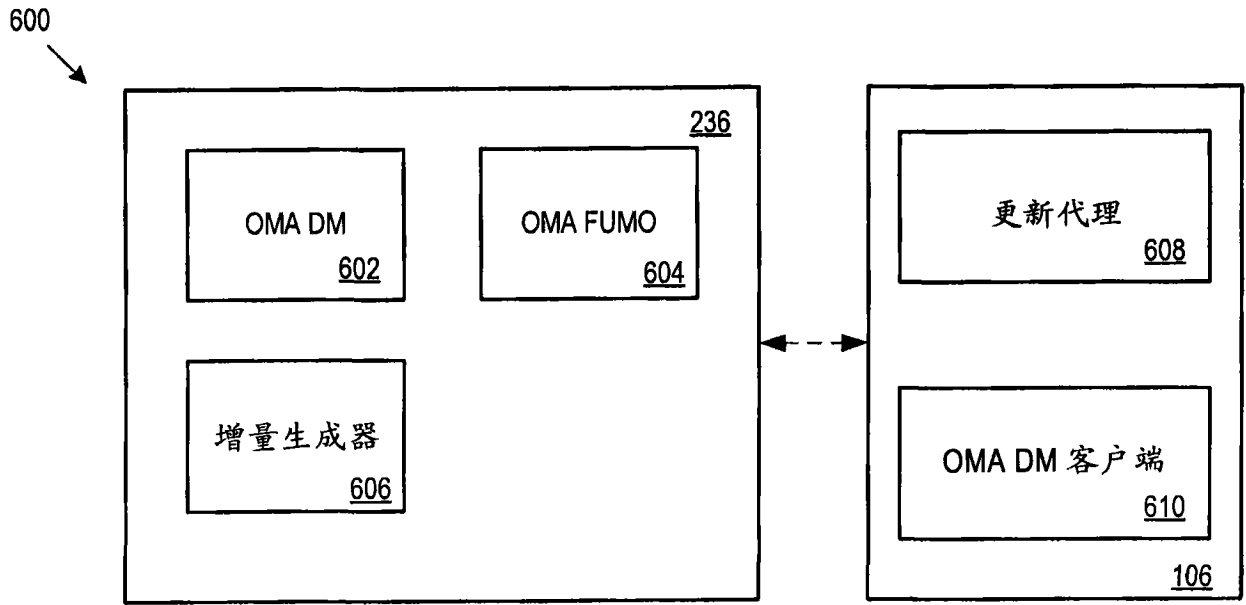


图 6A

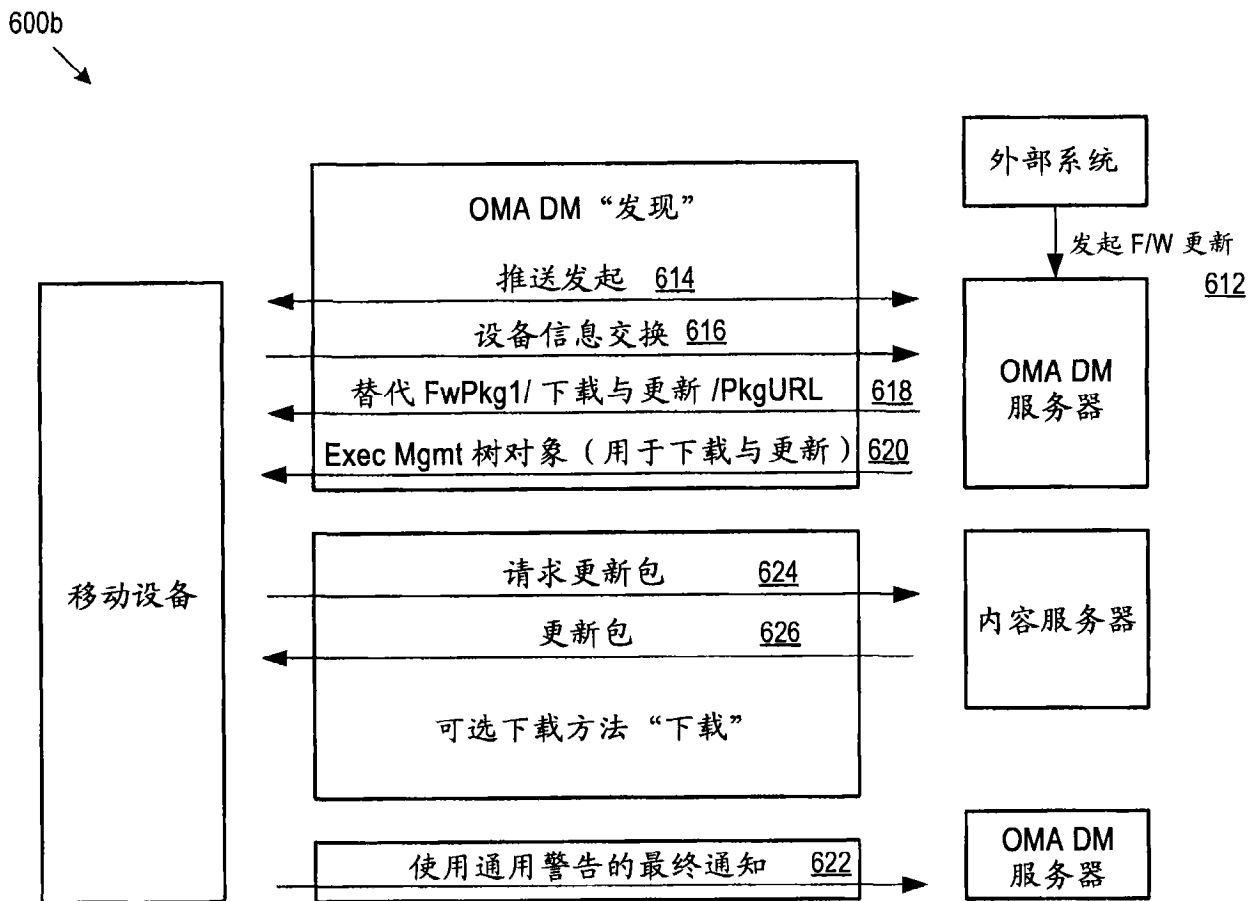


图 6B

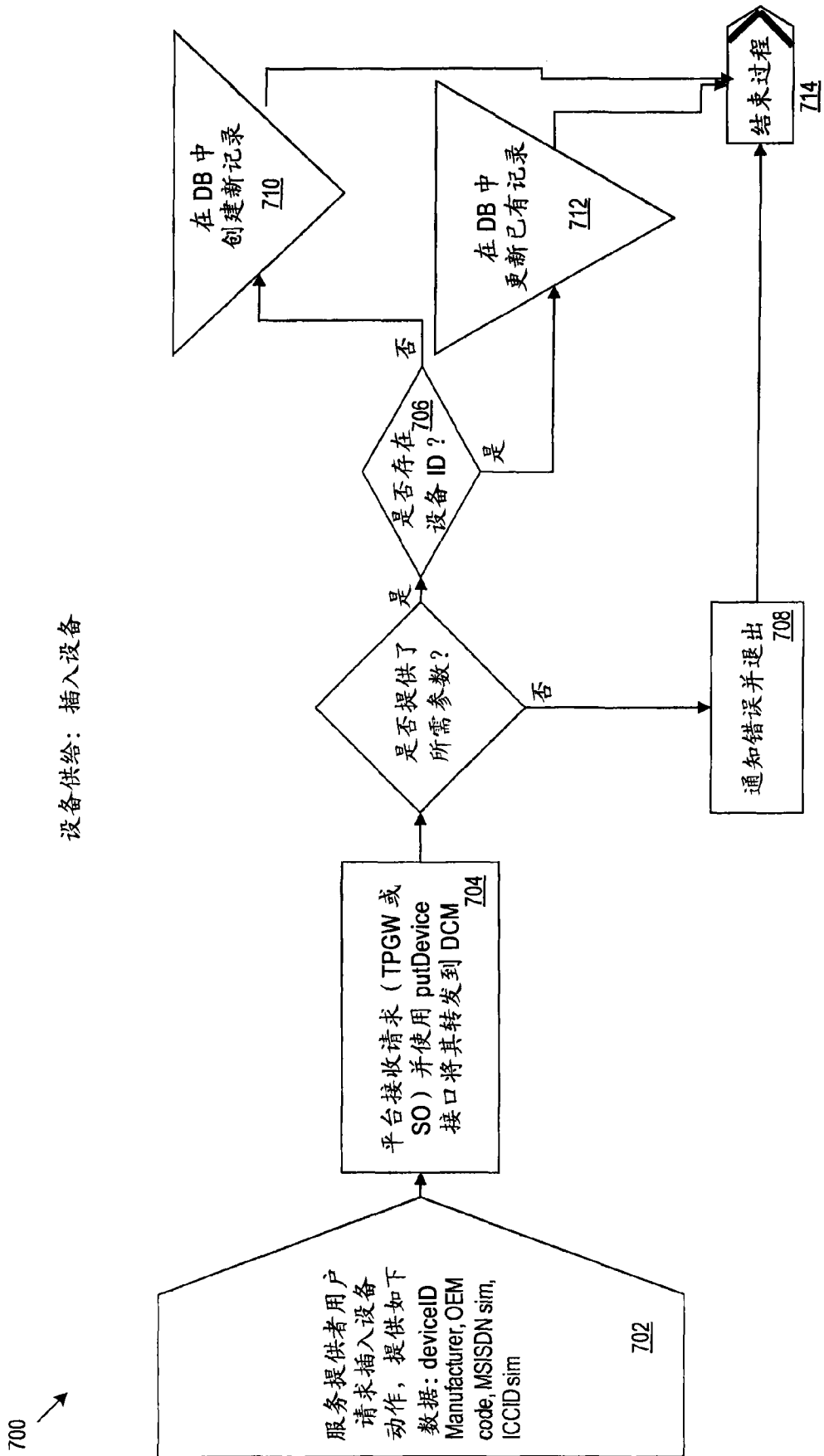


图 7

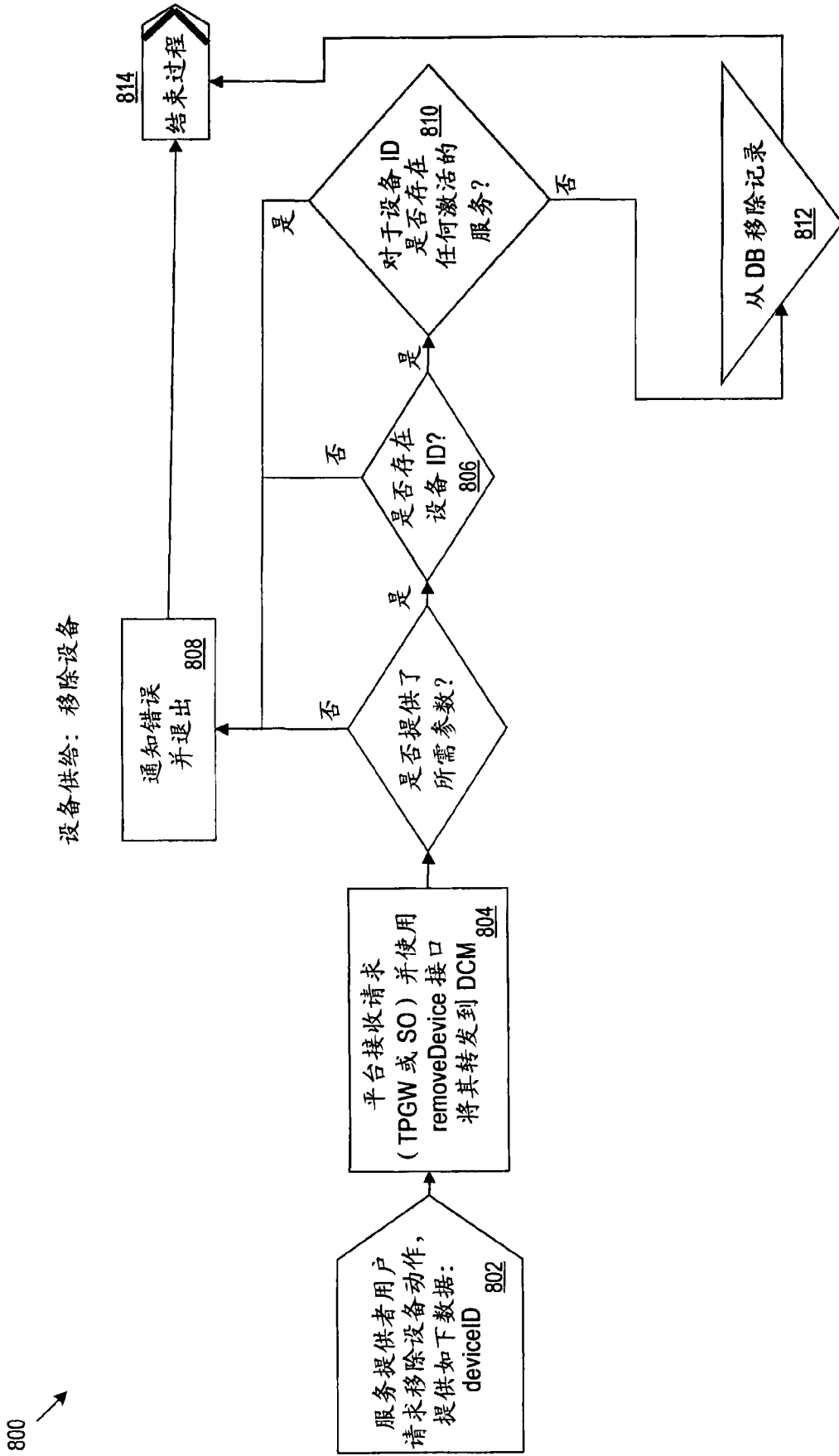


图 8

900

服务配置: 激活

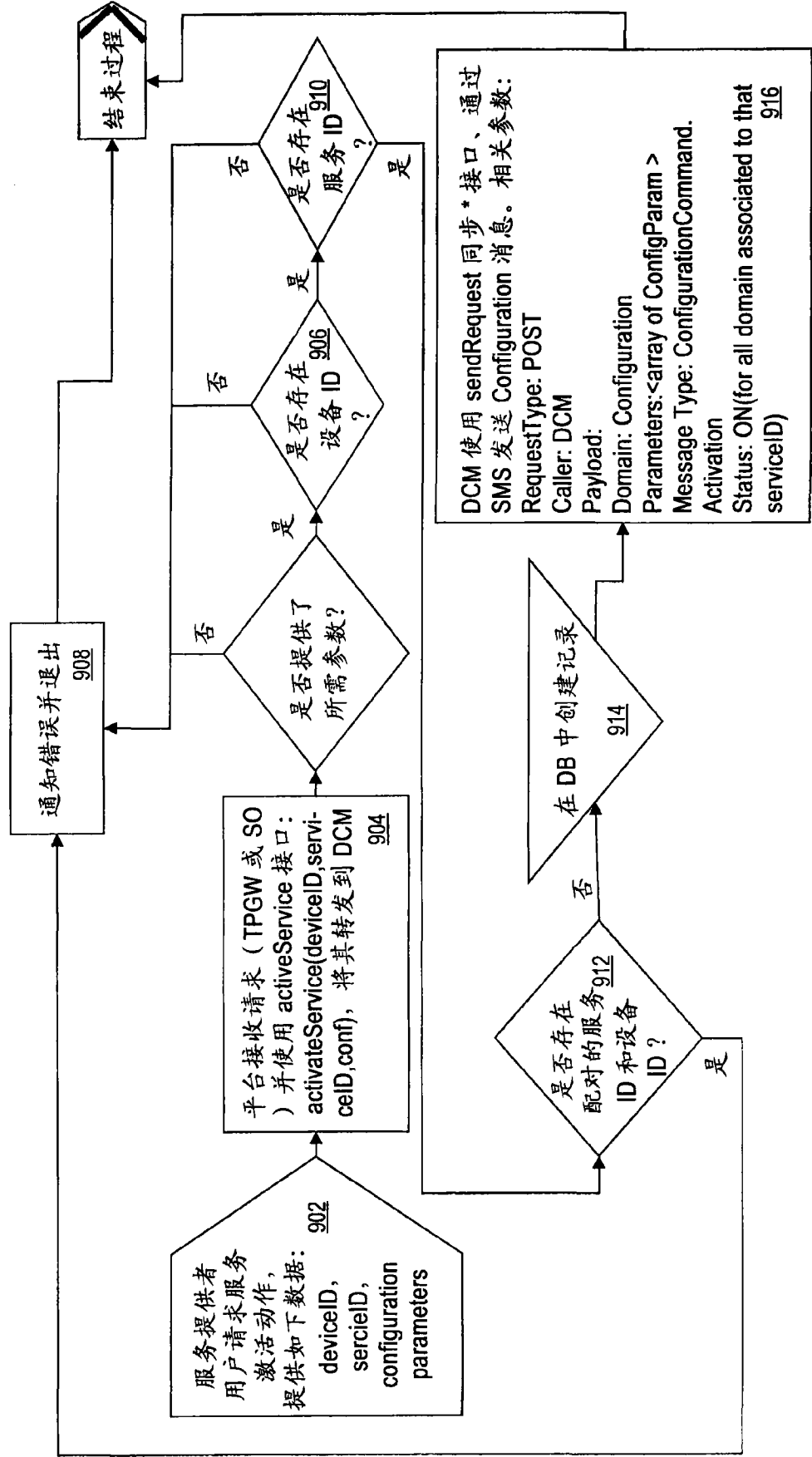


图 9

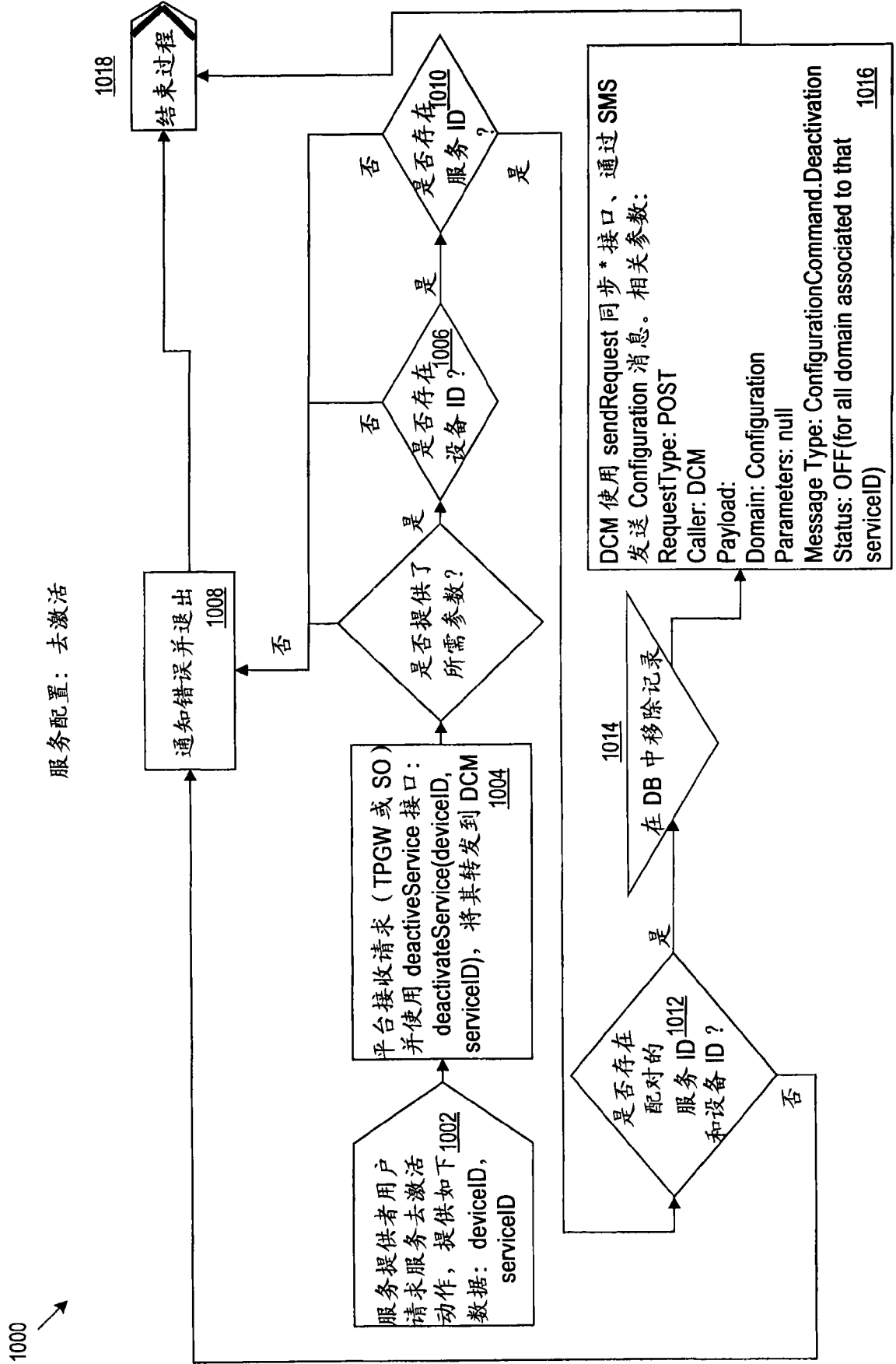


图 10

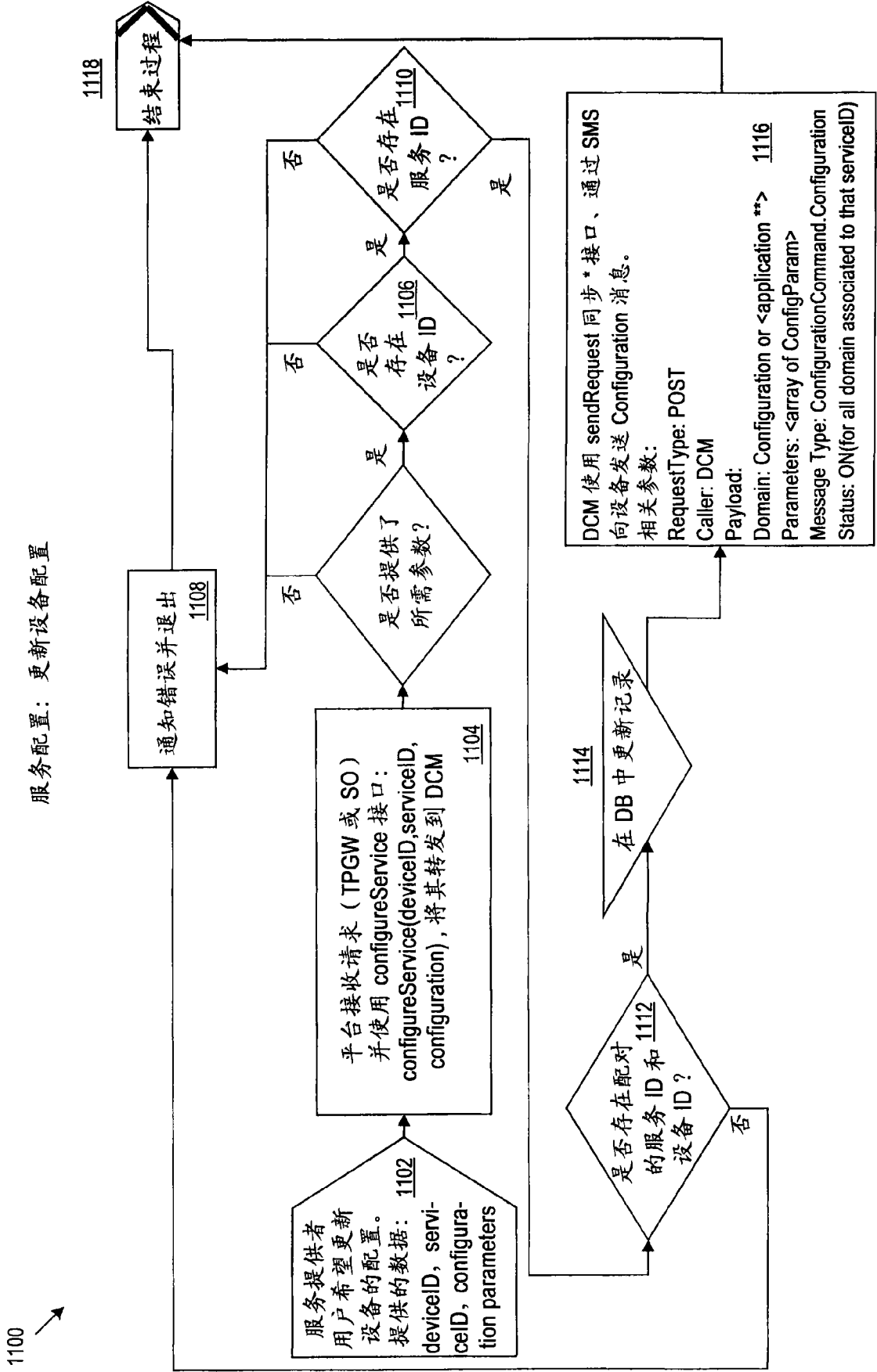


图 11

1200 ↗

服务配置: 交换服务 (替代设备)

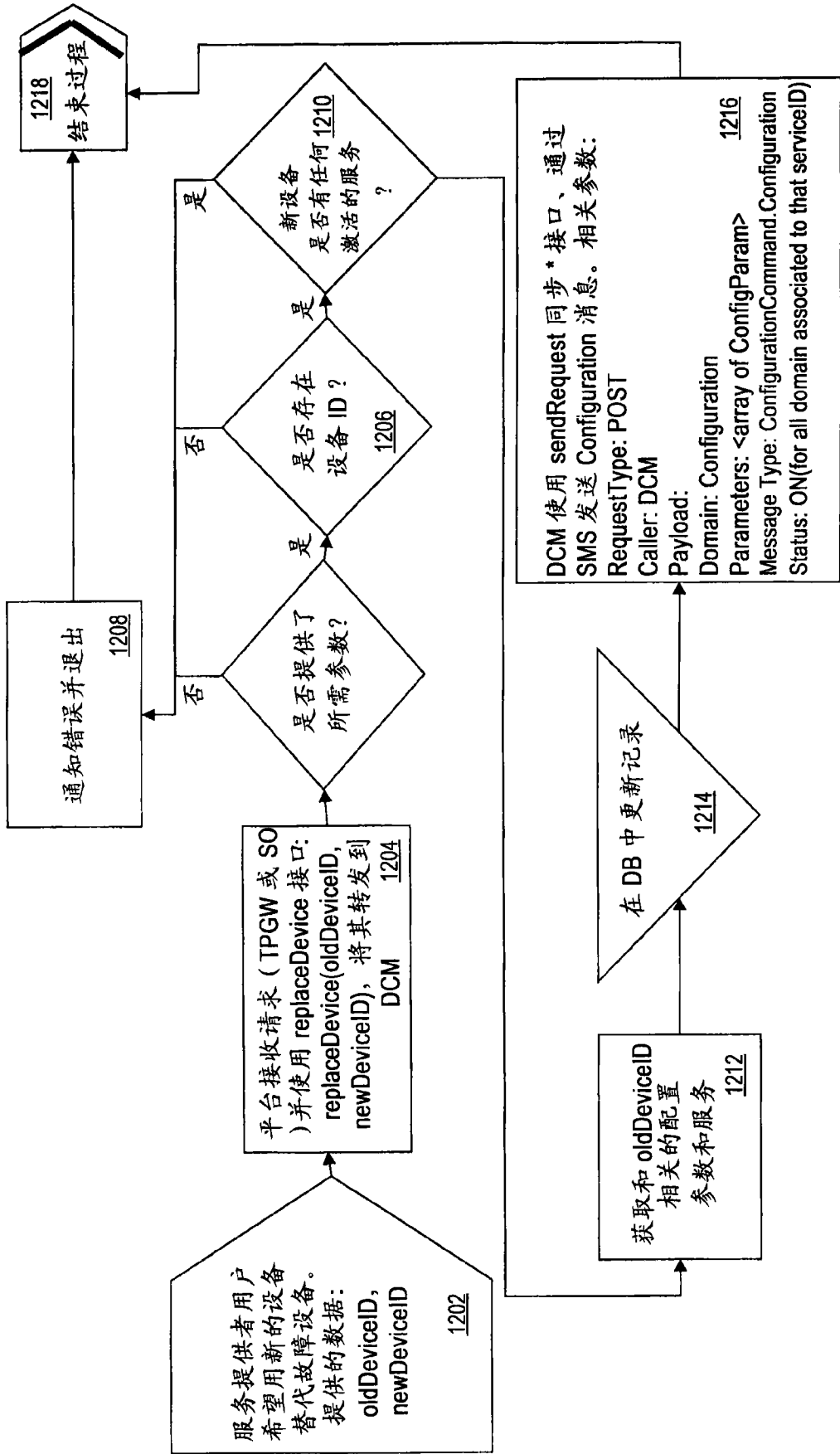


图 12

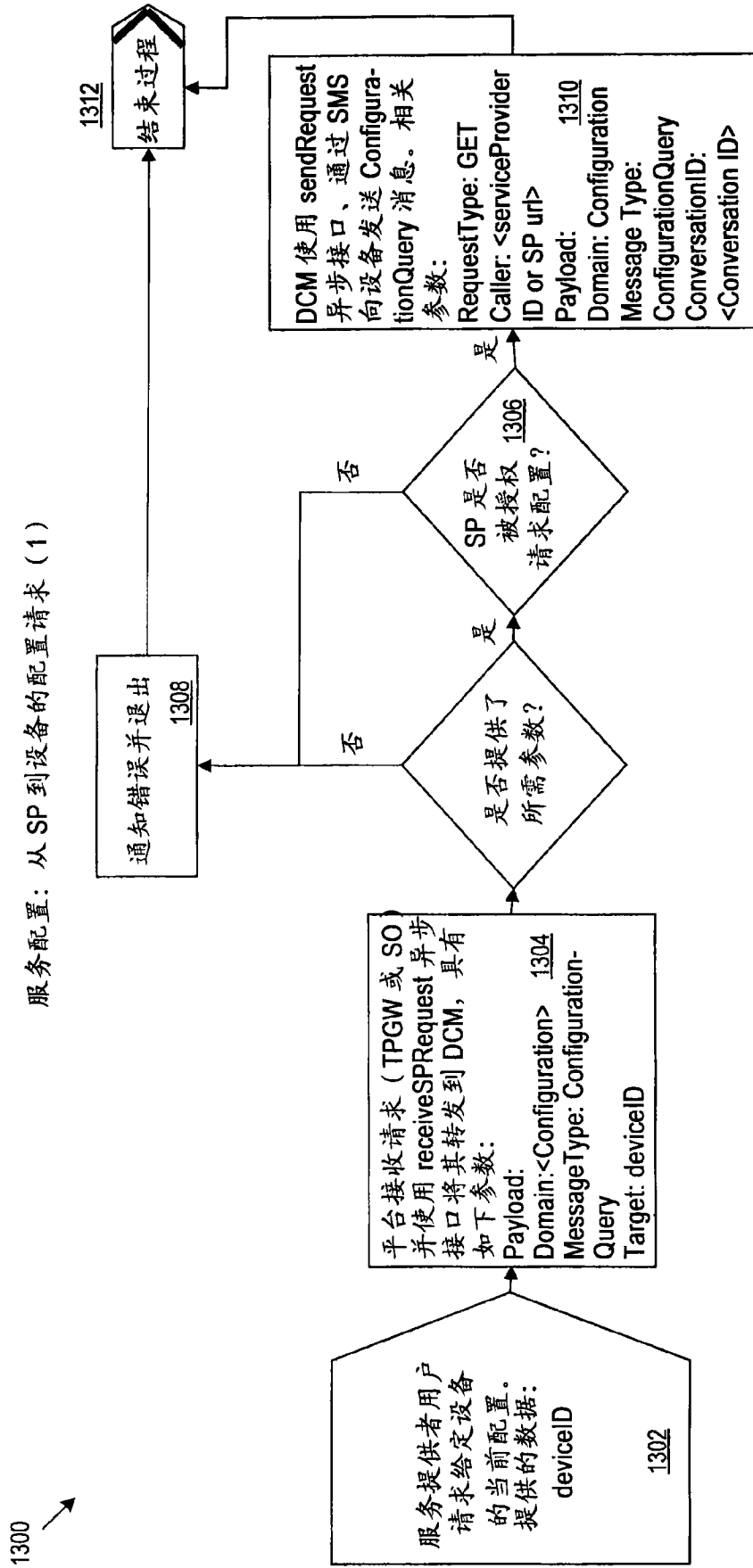


图 13

1400

服务配置：从设备到 SP 的配置请求 (2)

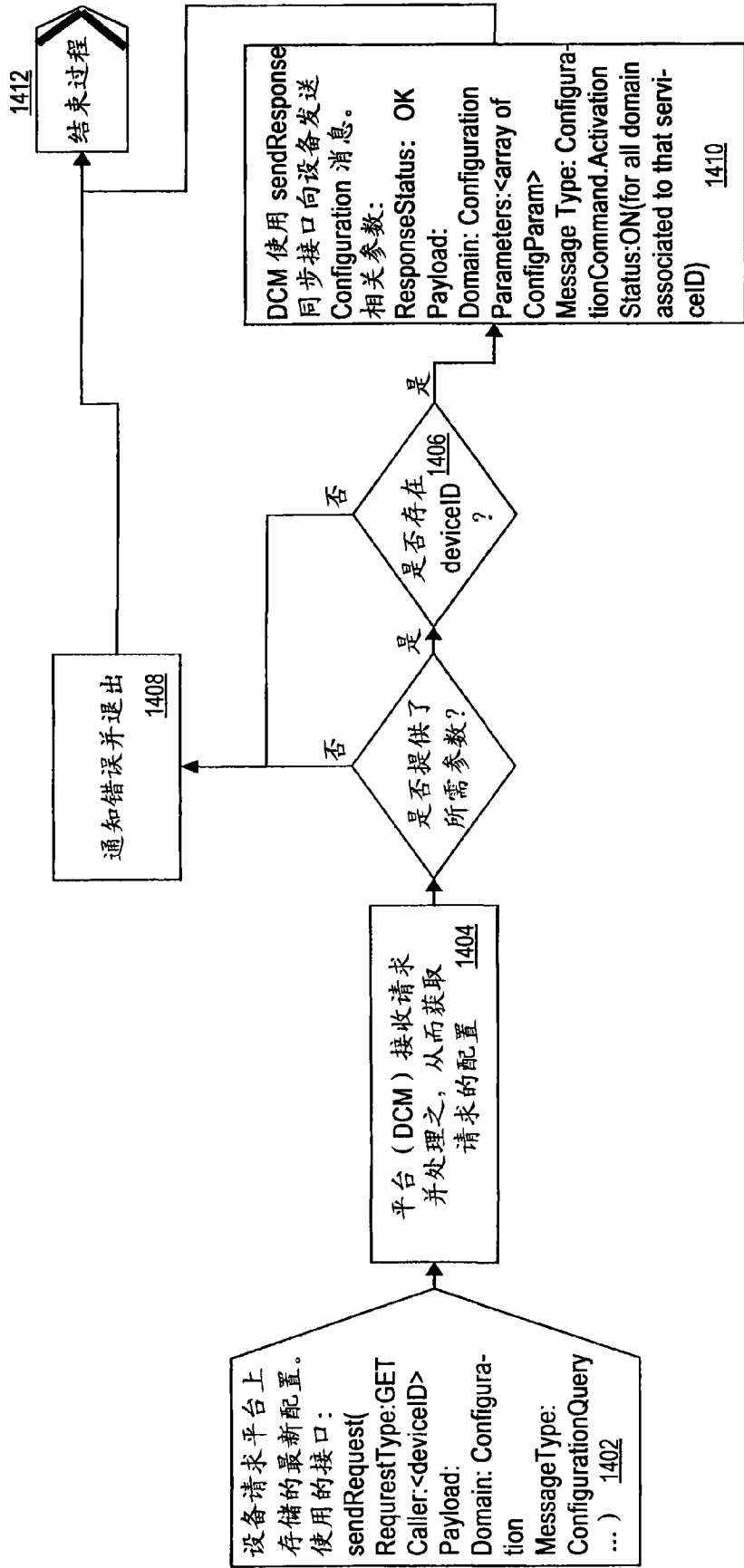


图 14

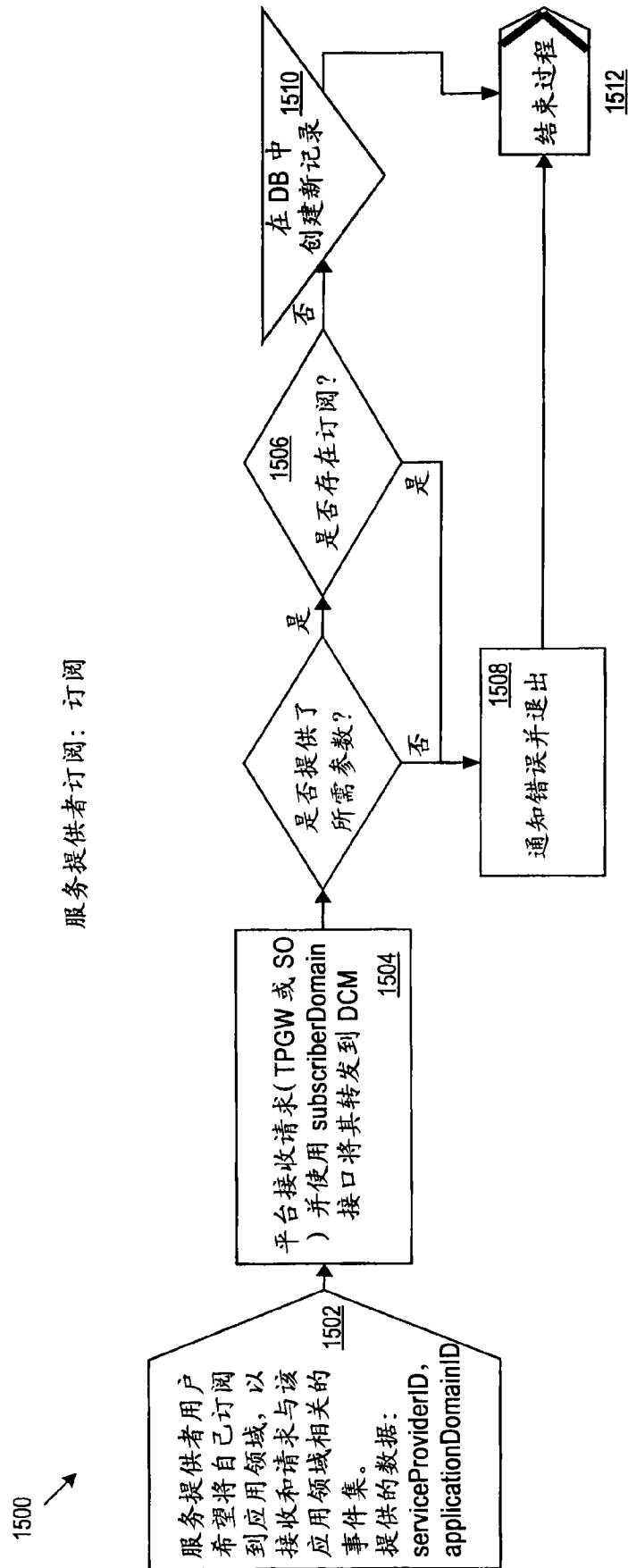


图 15

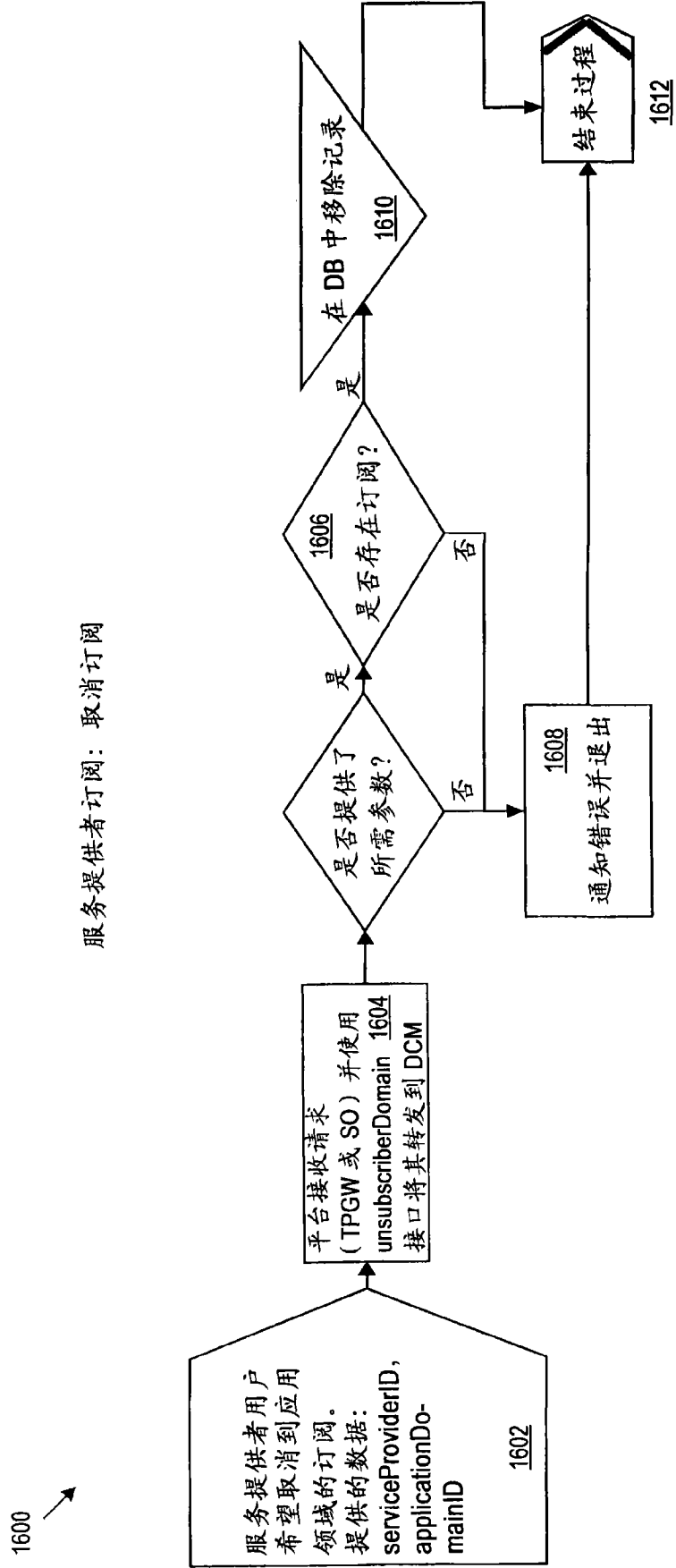


图 16

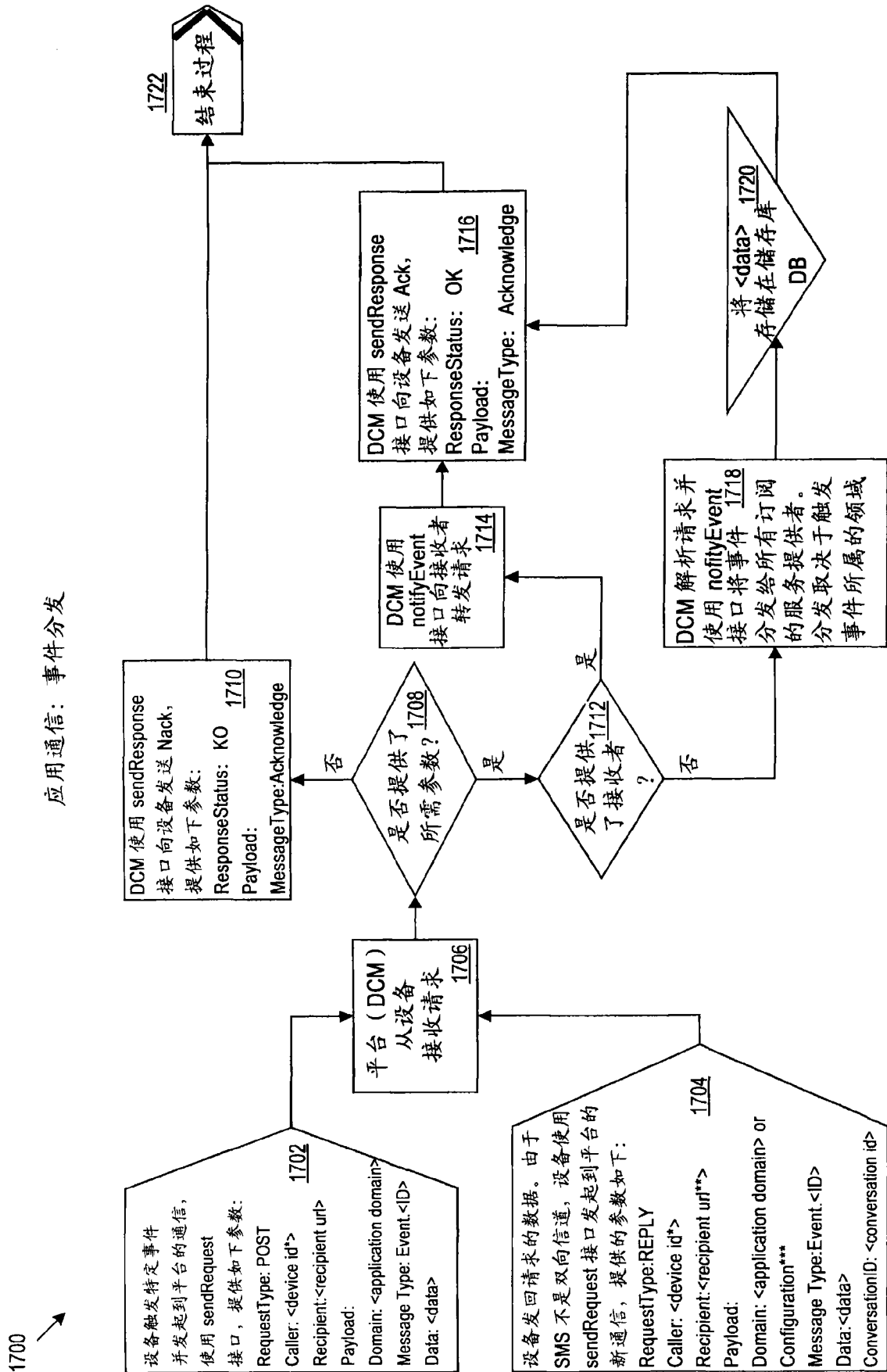


图 17

1800 ↗

应用通信: 应用请求

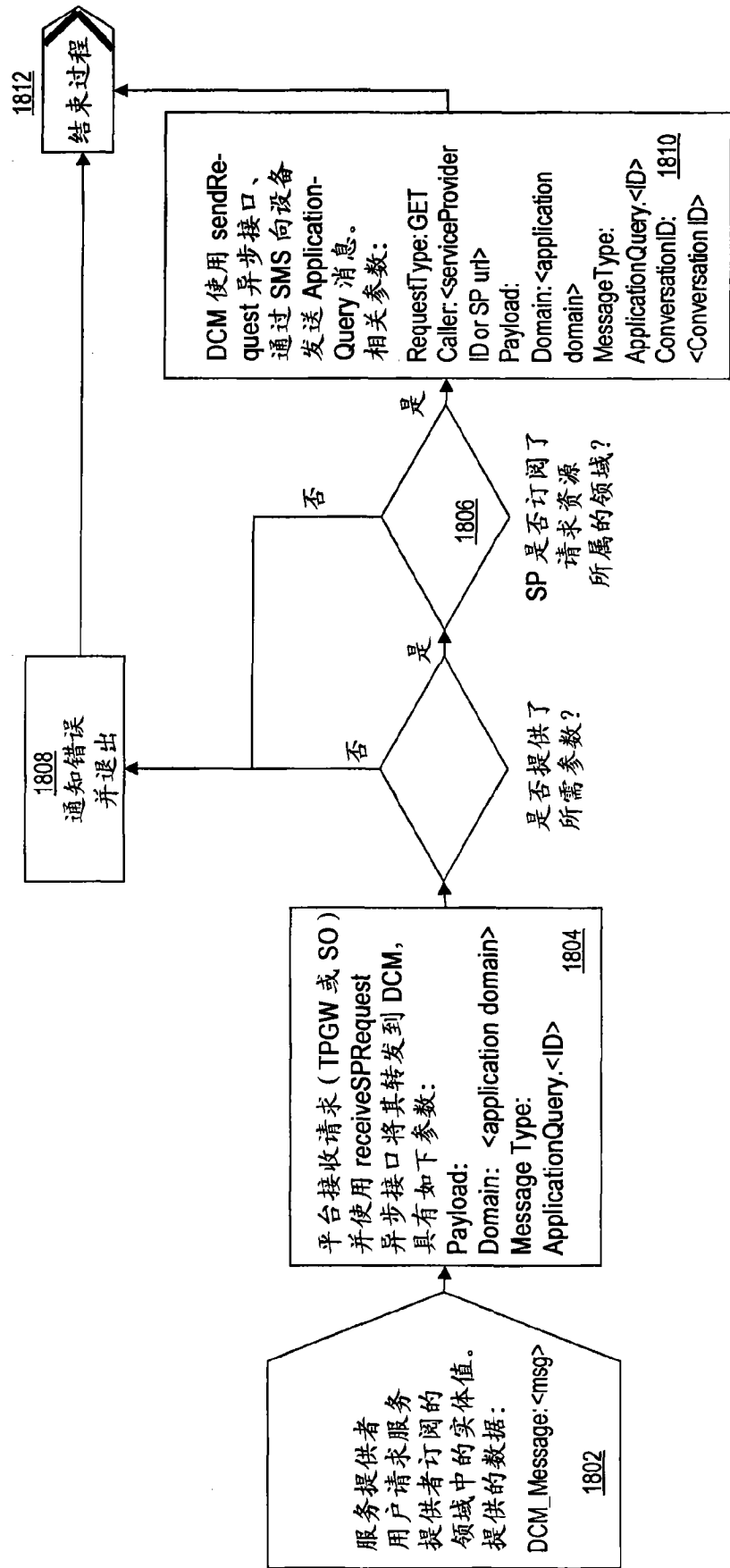


图 18

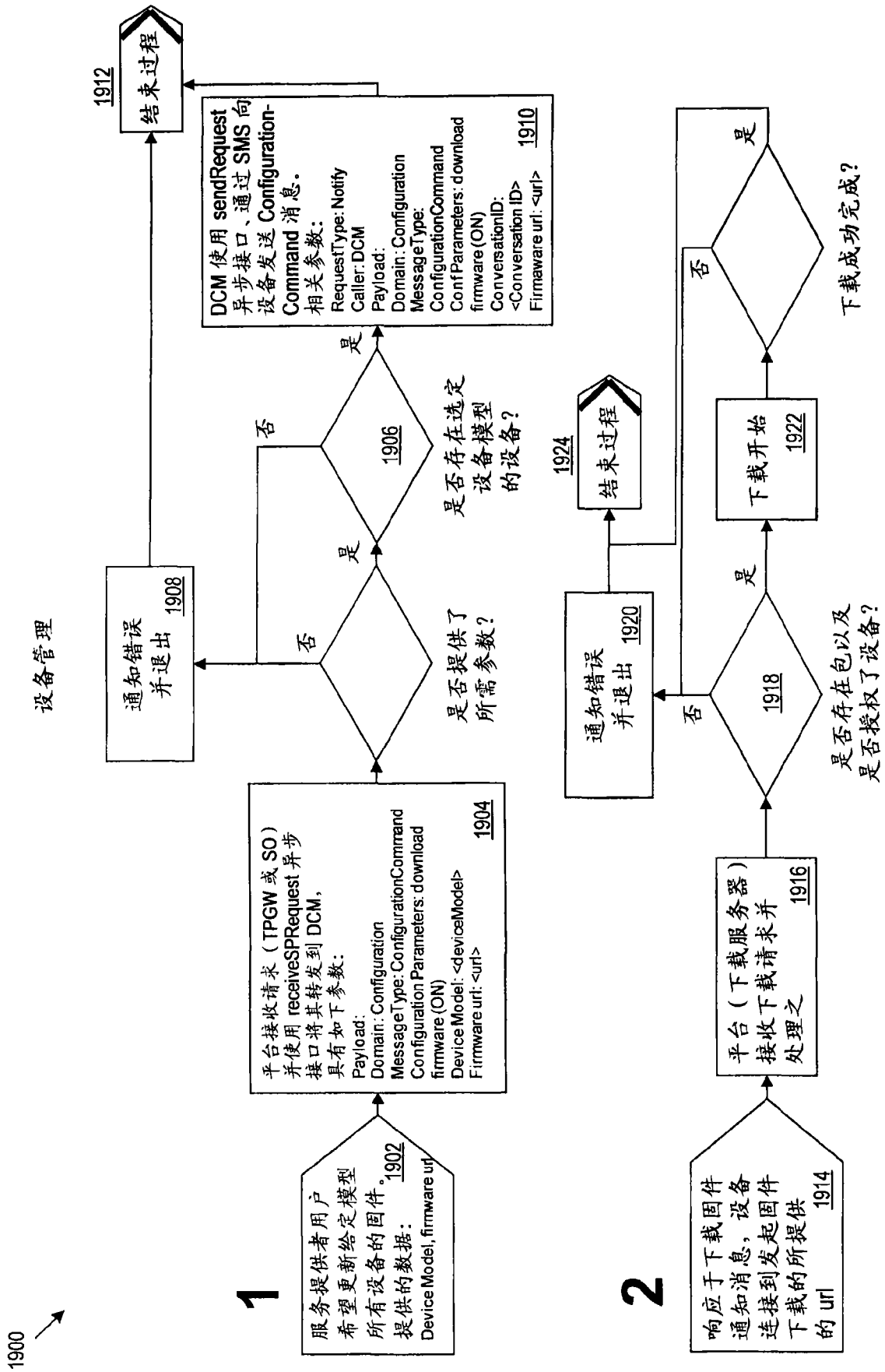


图 19

2000 →

设备监控

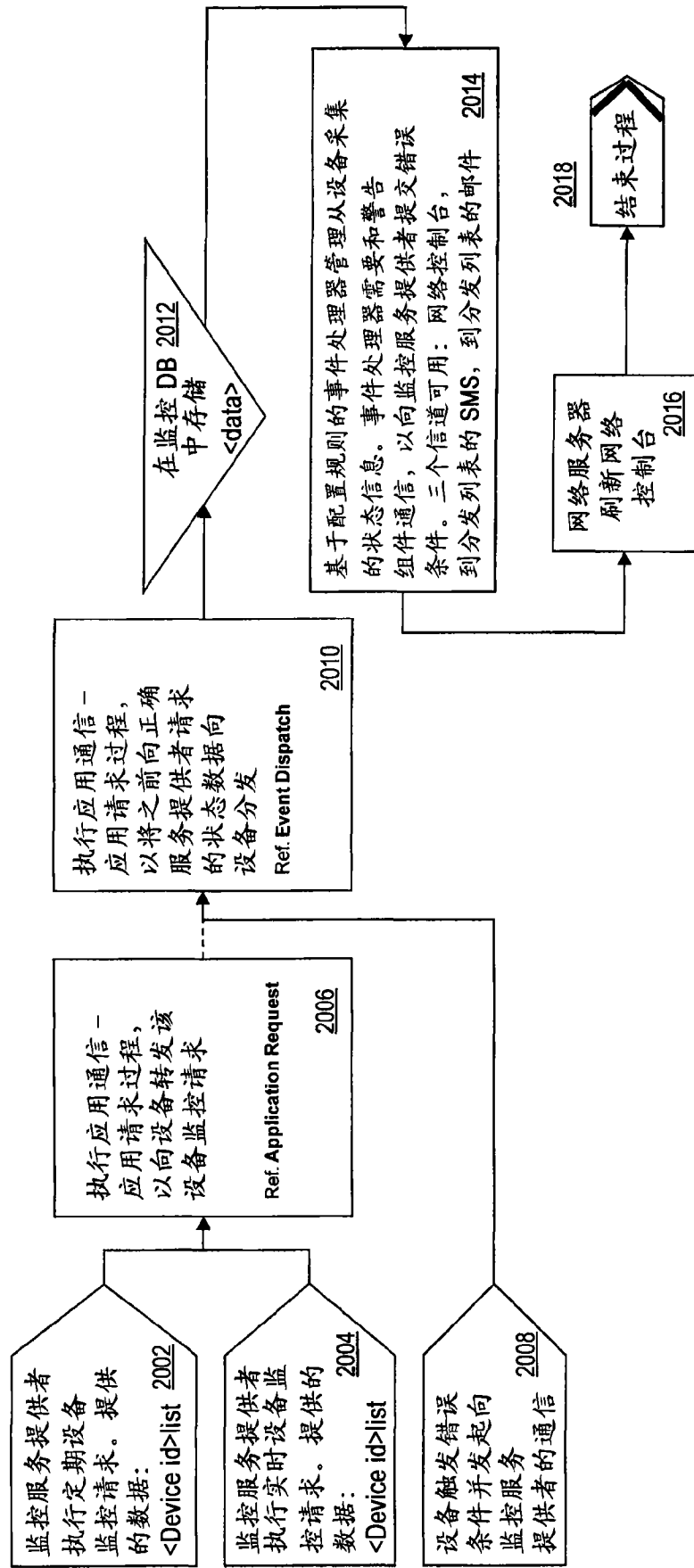


图 20

2100

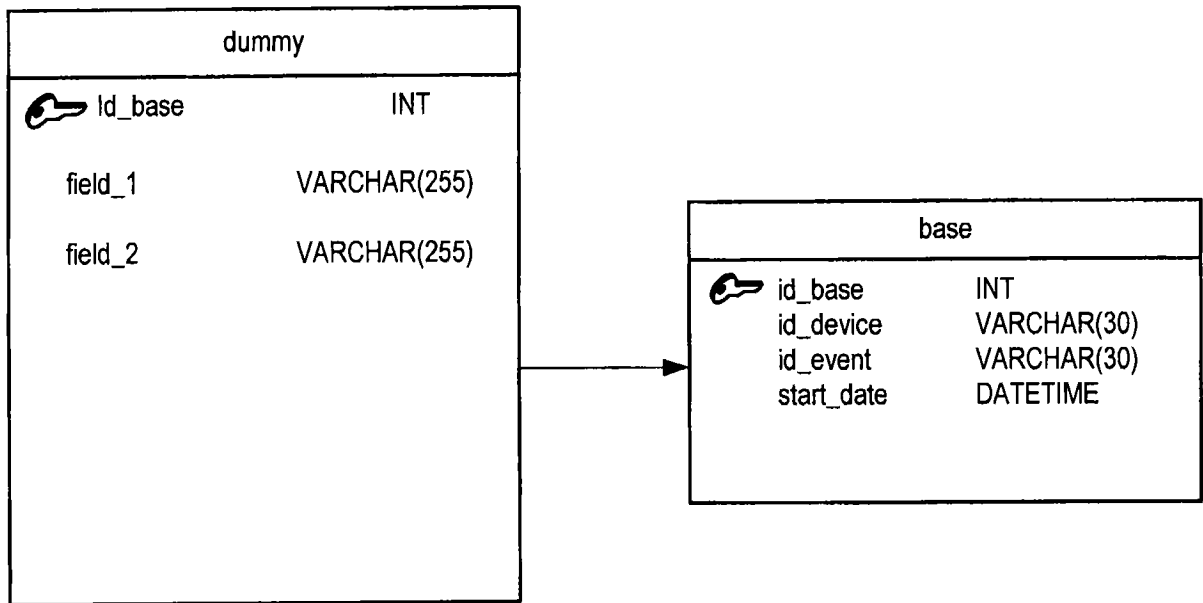


图 21

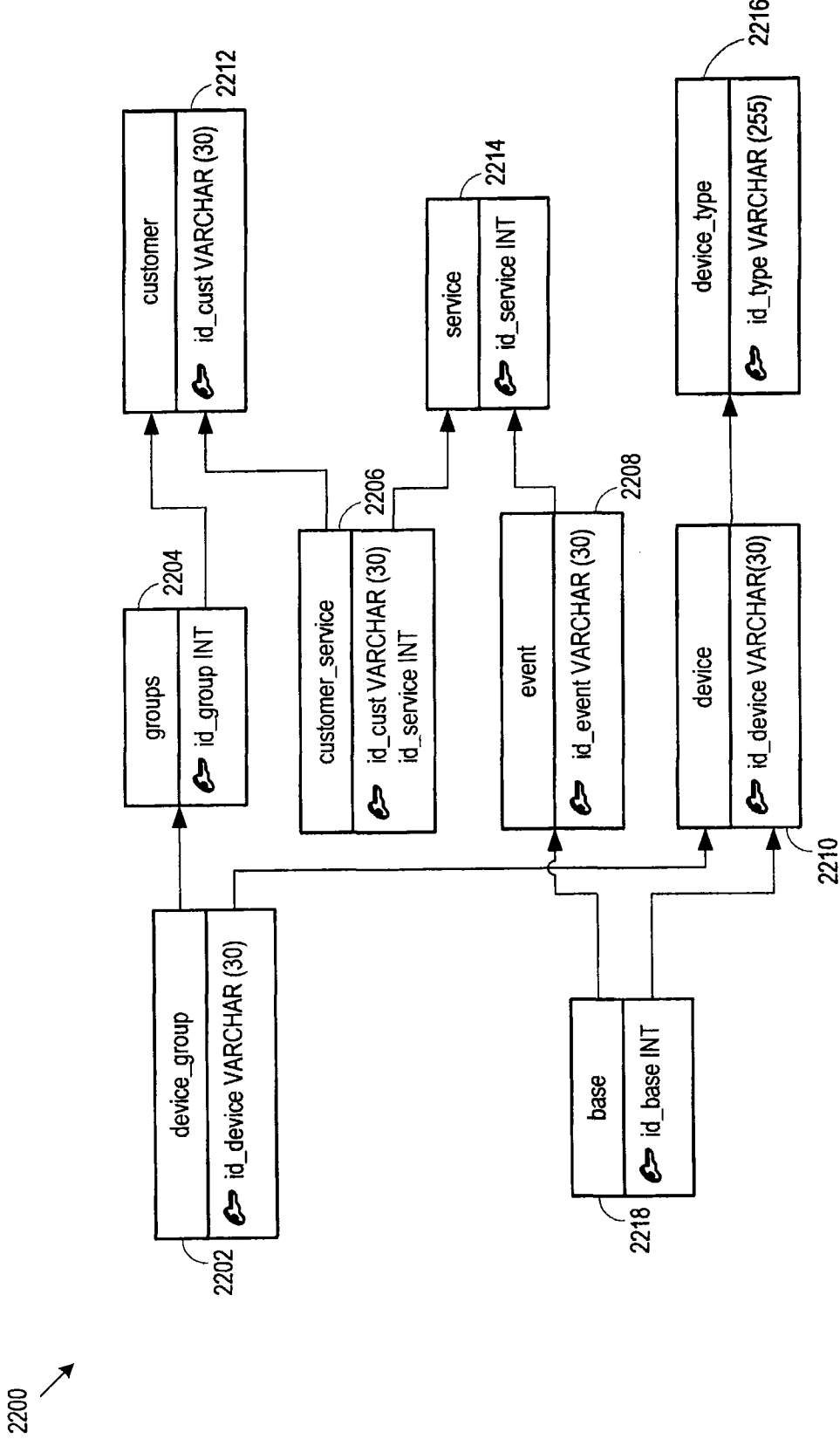


图 22

2300 →

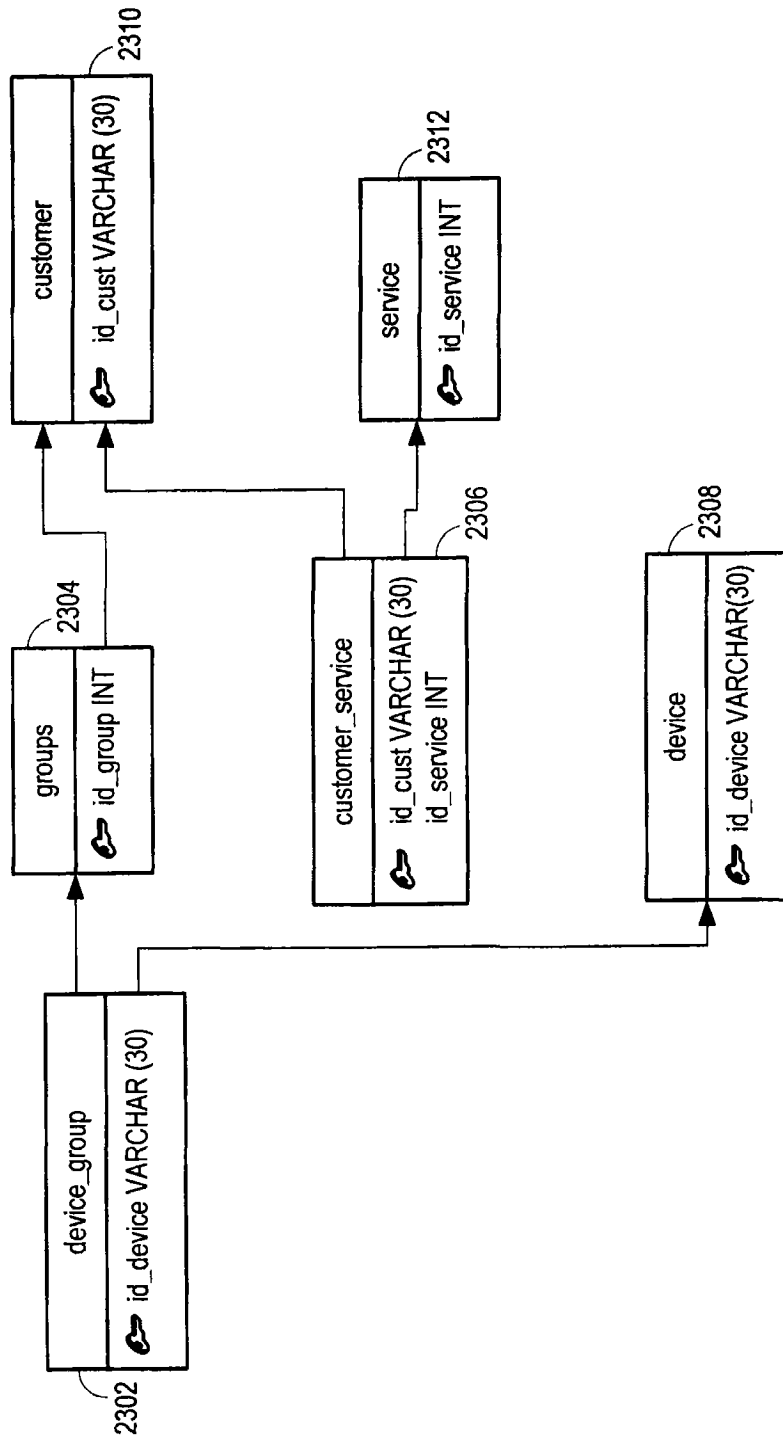


图 23

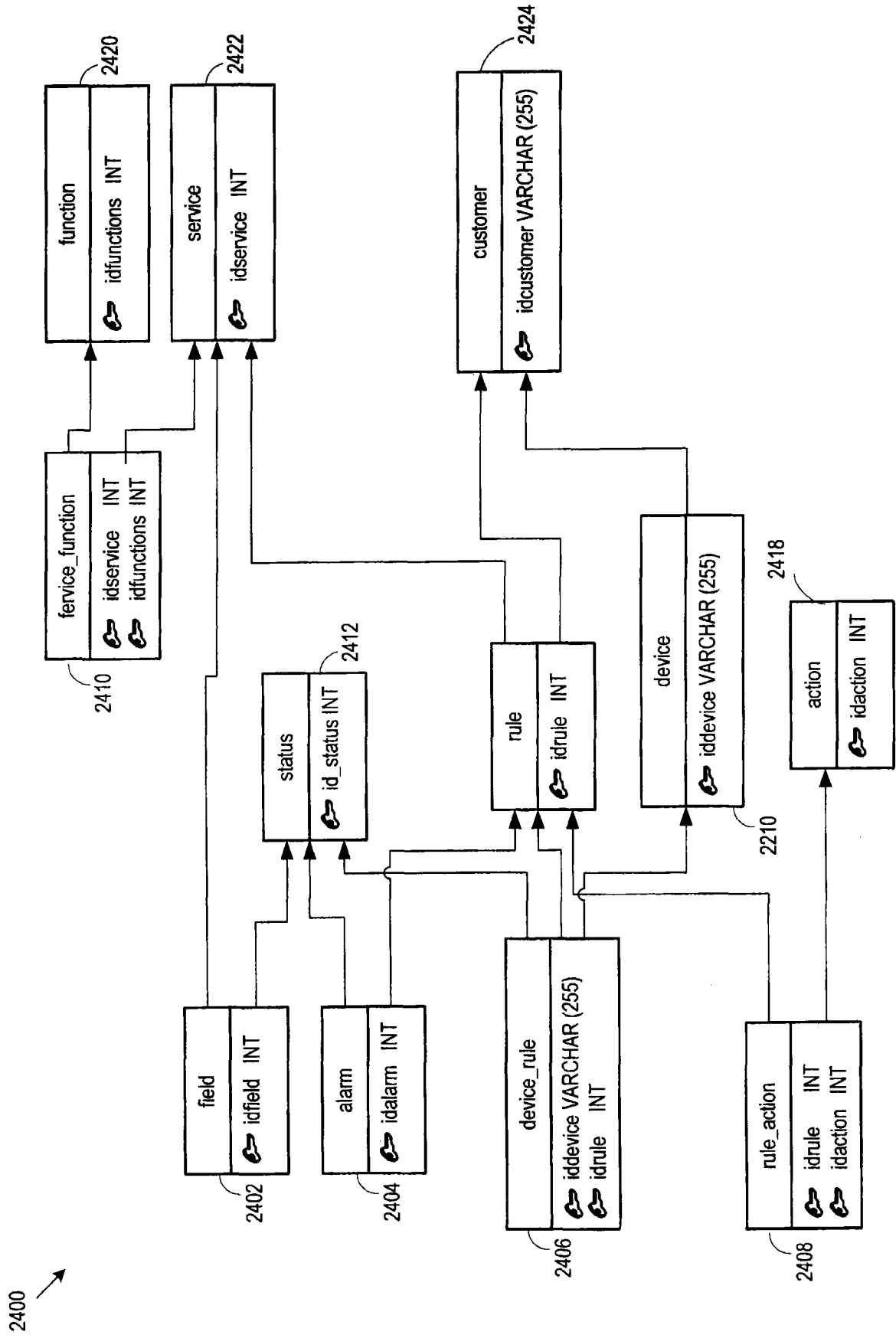


图 24

2500 ↗

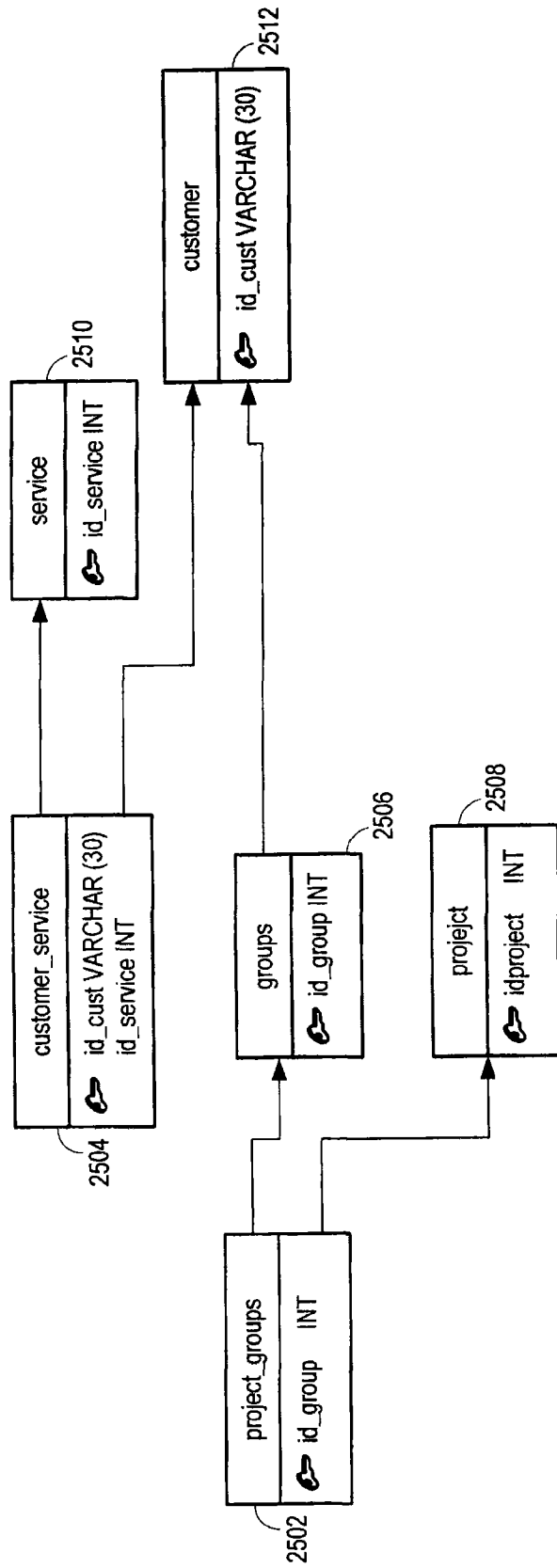


图 25

2600 →

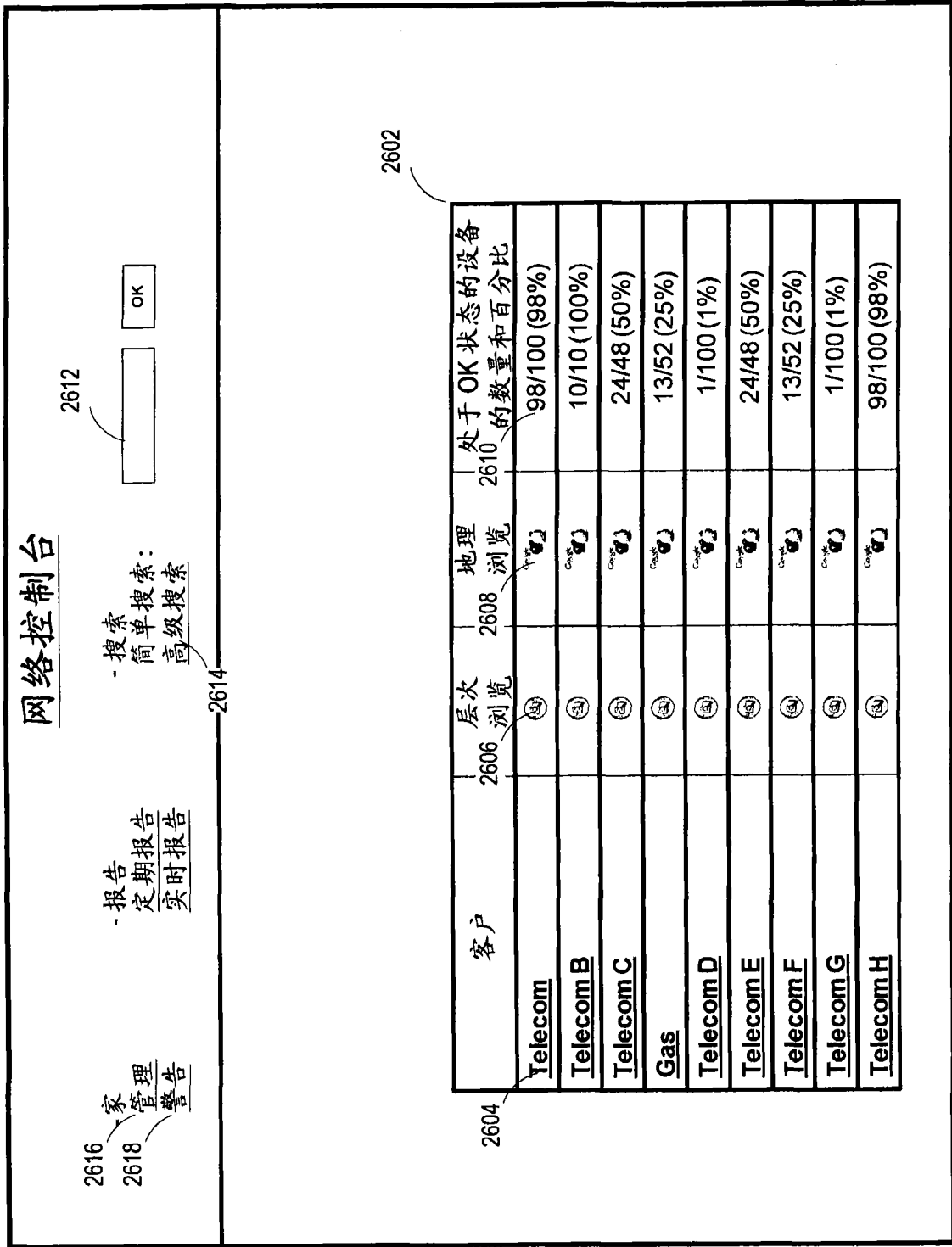


图 26

2600 →

网络控制台

- 家

- 报告

 - 定期报告

 - 实时报告

- 搜索

 - 简单搜索

 - 高级搜索

OK

Selected Client: *Telecom2704*

VAS: Vehicle Tracking 2706

VAS: Remote Surveillance

VAS: Remote Metering

VAS: Personal Tracking

VAS: Health Diagnostics

客户	层次浏览	地理浏览	处于OK状态的设备的数量和百分比
<u>Telecom</u>			98/100 (98%)
<u>Telecom B</u>			10/10 (100%)
<u>Telecom C</u>			24/48 (50%)
<u>Gas</u>			13/52 (25%)
<u>Telecom D</u>			1/100 (1%)
<u>Telecom E</u>			24/48 (50%)
<u>Telecom F</u>			13/52 (25%)
<u>Telecom G</u>			1/100 (1%)
<u>Telecom H</u>			98/100 (98%)

图 27

2600 ↗

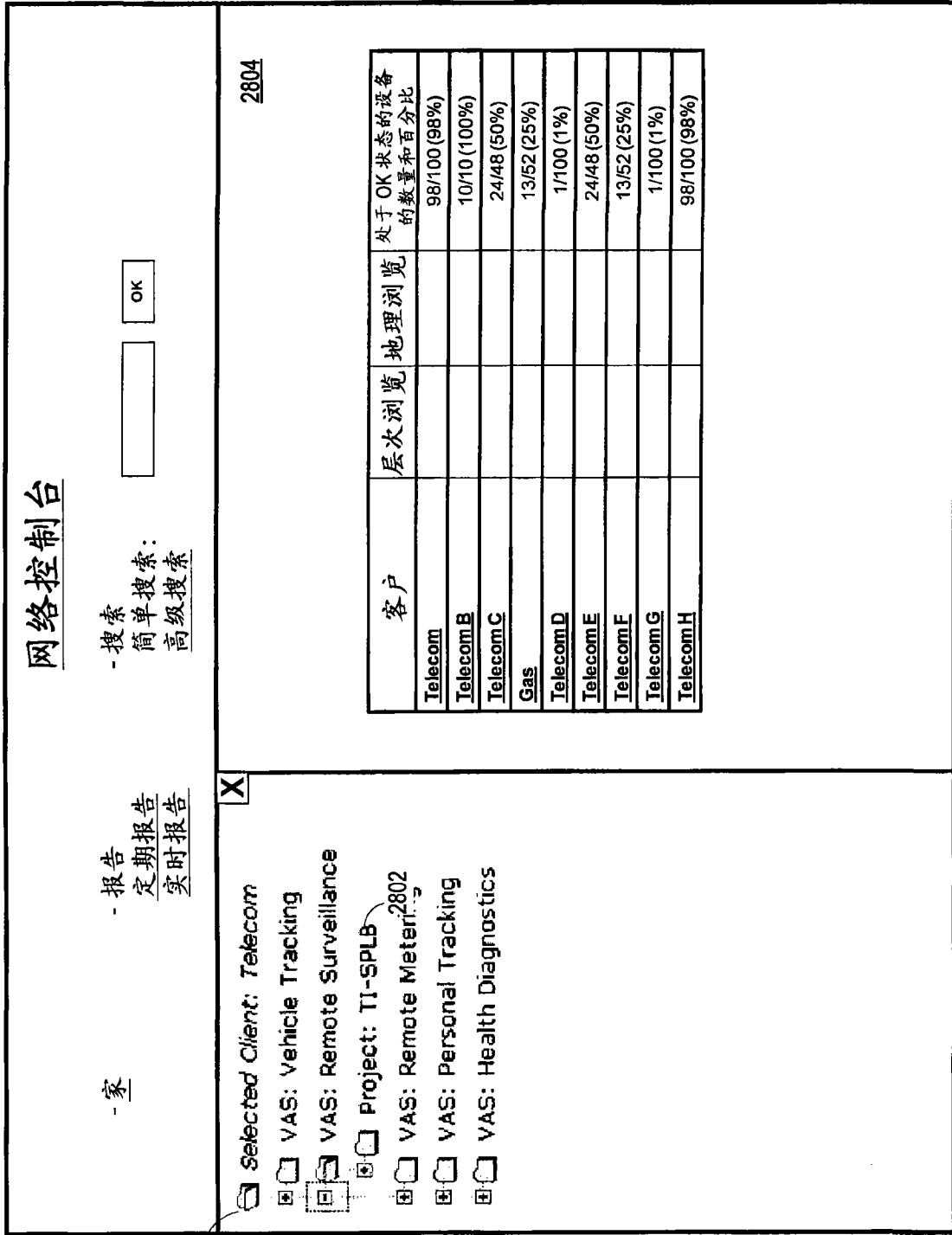


图 28

2600 →

网络控制台

- 家

- 报告
定期报告
实时报告

- 搜索
简单搜索
高级搜索

OK

移动设备

Client: Telecom
VAS: Remote Surveillance
Project: TI-SPLB

2904

检查	设备 ID	MSISDN	站点	状态
√	JR6789	3358789001	Rome (41.901375,12.501165)	OK
√	JR6790	3358789221	Rome (41.901375,12.501165)	KO
√	JR6791	3358789234	Rome (41.901375,12.501165)	OK
√	DR1720	3390909090	Rome (41.906964,12.485254)	不存在
√	DR1721	3345481000	Rome (41.906964,12.485254)	不存在
√	DR1722	3342381040	Rome (41.906964,12.485254)	OK
√	DR1723	3345481123	Rome (41.906964,12.485254)	OK
√	DR1724	3345423400	Rome (41.906964,12.485254)	OK
√	DR1725	3345423500	Rome (41.906964,12.485254)	OK
√	DR1726	3345423600	Rome (41.906964,12.485254)	OK

2906

检查选定状态 检查所有状态

约 22 个结果中的结果 1-10
页面 | << 1 2 3 >> |

2702

- Selected Client: Telecom
- VAS: Vehicle Tracking
- VAS: Remote Surveillance 2902
- Project: TI-SPLB
- Site: Rome (41.901375,12.501165)
- Site: Rome (41.906964,12.485254)
- VAS: Remote Metering
- VAS: Personal Tracking
- VAS: Health Diagnostics

图 29

2600 ↗

网络控制台

- 家

- 报告

定期报告

实时报告

- 搜索

简单搜索:

高级搜索

移动设备

Client: Telecom

VAS: Remote Surveillance

Project: TI-SPLB

Site: Rome (41.901375, 12.501165)

2904

检查

检查	ID 设备	MSISDN	状态
<input type="checkbox"/>	JR6789	3358789001	OK
<input type="checkbox"/>	JR6790	3358789221	KO
<input type="checkbox"/>	JR6791	3358789234	OK

约 3 个结果中的结果 1-3

页面 <<1 >>1

2702

- Selected Client: Telecom
- VAS: Vehicle Tracking
- VAS: Remote Surveillance
- Project: TI-SPLB
- Site: Rome (41.901375, 12.501165)
- Mobile Equipment: JR6789
- Mobile Equipment: JR6790 3002
- Mobile Equipment: JR6791
- Site: Rome (41.906964, 12.485254)
- VAS: Remote Metering
- VAS: Personal Tracking
- VAS: Health Diagnostics

图 30

2600 ↗

网络控制台

家

- 报告

定期报告

实时报告

- 搜索

简单搜索:

高级搜索:

OK

Selected Client: Telecom

VAS: Vehicle Tracking

VAS: Remote Surveillance

Project: TI-SPLB

Site: Rome (41.901375, 12.501165)

Mobile Equipment: JR6789

Device HW: JR6789

SIM: 3358789001

Application: Mod.1

Mobile Equipment: JR6790

Mobile Equipment: JR6791

Site: Rome (41.906964, 12.485254)

VAS: Remote Metering

VAS: Personal Tracking

VAS: Health Diagnostics

2804

最后 10 天报告

移动设备: JR6789

Client: Telecom

VAS: Remote Surveillance

Project: TI-SPLB

Site: Rome (41.901375, 12.501165)

Device Brand: Teit

Device Type: GT863-PY

移动设备: JR6789	
移动设备聚集状态	概要
最后更新	OK
SIM 状态	20090729 11:32am
设备状态	OK
应用状态	OK

SIM	
MSISDN	3358789001
APN	lbox.tim.it
网络上存在性	Present
激活状态	Active
SMS 信道	OK
GPRS 信道	OK
应用	
模块 1 状态	OK

Device HW	
设备 ID	JR6789
设备日期时间	20090803 11:12am
最后重置	20090701 10:23am
最后配置更新	20090711 11:35am
操作状态	OK
供电级别	High
存储器使用	Low
CPU 负载	Low

立即检查状态

显示调度会议

2702

图 31

2600 ↗

网络控制台

- 报告

定期报告

实时报告

- 搜索

简单搜索

高级搜索

移动设备: JR6789

Client: Telecom
 VAS: Remote Surveillance
 Project: TI-SPLB
 Site: Rome (41.901375,12.501165)
 Device Brand: Teit
 Device Type: GT863-PY

移动设备: JR6789

移动设备聚集状态	概要
最后更新	OK
SIM 状态	20090729 11:32am
设备状态	OK
应用状态	OK
	OK

Device HW

设备 ID	JR6789
设备日期时间	20090803 11:12am
最后重置	20090701 10:23am
最后配置更新	20090711 11:35am
操作状态	OK
供电级别	High
存储器使用	Low
CPU 负载	Low

Log Messages

[20090721184043]: Configuration error
 [20090721183911]: Low power supply level
 [20090701113011]: Configuration updated
 [20090701103031]: Reboot completed
 [20090701103011]: Device is rebooting
 [20090701103009]: Reboot command received

3202

Selected Client: Telecom

- VAS: Vehicle Tracking
- VAS: Remote Surveillance
- Project: TI-SPLB
- Site: Rome (41.901375,12.501165)
- Mobile Equipment: JR6789
 - Device HW: JR6789
 - SIM: 3358789001
 - Application: Mod.1
- Mobile Equipment: JR6790
- Mobile Equipment: JR6791
- Site: Rome (41.906964,12.485254)
- VAS: Remote Metering
- VAS: Personal Tracking
- VAS: Health Diagnostics

立即检查状态

图 32

2600 ↗

网络控制台

- 家
- 搜索
- 报告

简单搜索:
OK

高级搜索
定期报告
实时报告

移动设备: JR6789

Client: Telecom
 VAS: Remote Surveillance
 Project: TI-SPLB
 Site: Rome (41.901375,12.501165)
 Device Brand: Teit
 Device Type: GT863-PY

移动设备: JR6789

移动设备聚集状态
 最后更新: 20090729 11:32am
 SIM 状态: OK
 设备状态: OK
 应用状态: OK

SIM

MSISDN	3358789001
APN	ibox.tim.it
网络上存在性	Present
激活状态	Active
SMS 信道	OK
GPRS 信道	OK

Log Messages

[20090721184741]: GPRS Network error
 [20090721184043]: Network Configuration error

下载系统日志文件

清除日志

X

Selected Client: Telecom

- VAS: Vehicle Tracking
- VAS: Remote Surveillance
- Project: TI-SPLB
- Site: Rome (41.901375,12.501165)
- Mobile Equipment: JR6789
 - Device HW: JR6789
 - SIM: 3358789001
 - Application: Mod.1
- Mobile Equipment: JR6790
- Mobile Equipment: JR6791
- Site: Rome (41.906964,12.485254)
- VAS: Remote Metering
- VAS: Personal Tracking
- VAS: Health Diagnostics

图 33

2600 ↗

网络控制台

- 报告:
- 定期报告,
- 实时报告

- 搜索:
- 简单搜索
- 高级搜索

移动设备: JR6789

Client: Telecom
VAS: Remote Surveillance
Project: TI-SPLB
Site: Rome (41.901375,12.501165)
Device Brand: Tellit
Device Type: GT863-PY

移动设备 JR6789	概要
移动设备聚集状态	OK
最后更新	20090729 11:32am
SIM 状态	OK
设备状态	OK
应用状态	OK

应用

模块 1 状态

立即检查状态

应用

OK

Log Messages

[20090721184953]: Video recorded

[20090721184923]: Start Video recording

[20090721184923]: Detect motion

[20090721184245]: Video recorded

[20090721184215]: Start Video recording

[20090721184214]: Detect motion

下载应用日志文件

X

Selected Client: Telecom

VAS: Vehicle Tracking

VAS: Remote Surveillance

Project: TI-SPLB

Site: Rome (41.901375,12.501165)

Mobile Equipment: JR6789

Device HW: JR6789

SIM: 3358789001

Application: Mod.1

Mobile Equipment: JR6790

Mobile Equipment: JR6791

Site: Rome (41.906964,12.485254)

VAS: Remote Metering

VAS: Personal Tracking

VAS: Health Diagnostics

图 34

2600 ↗

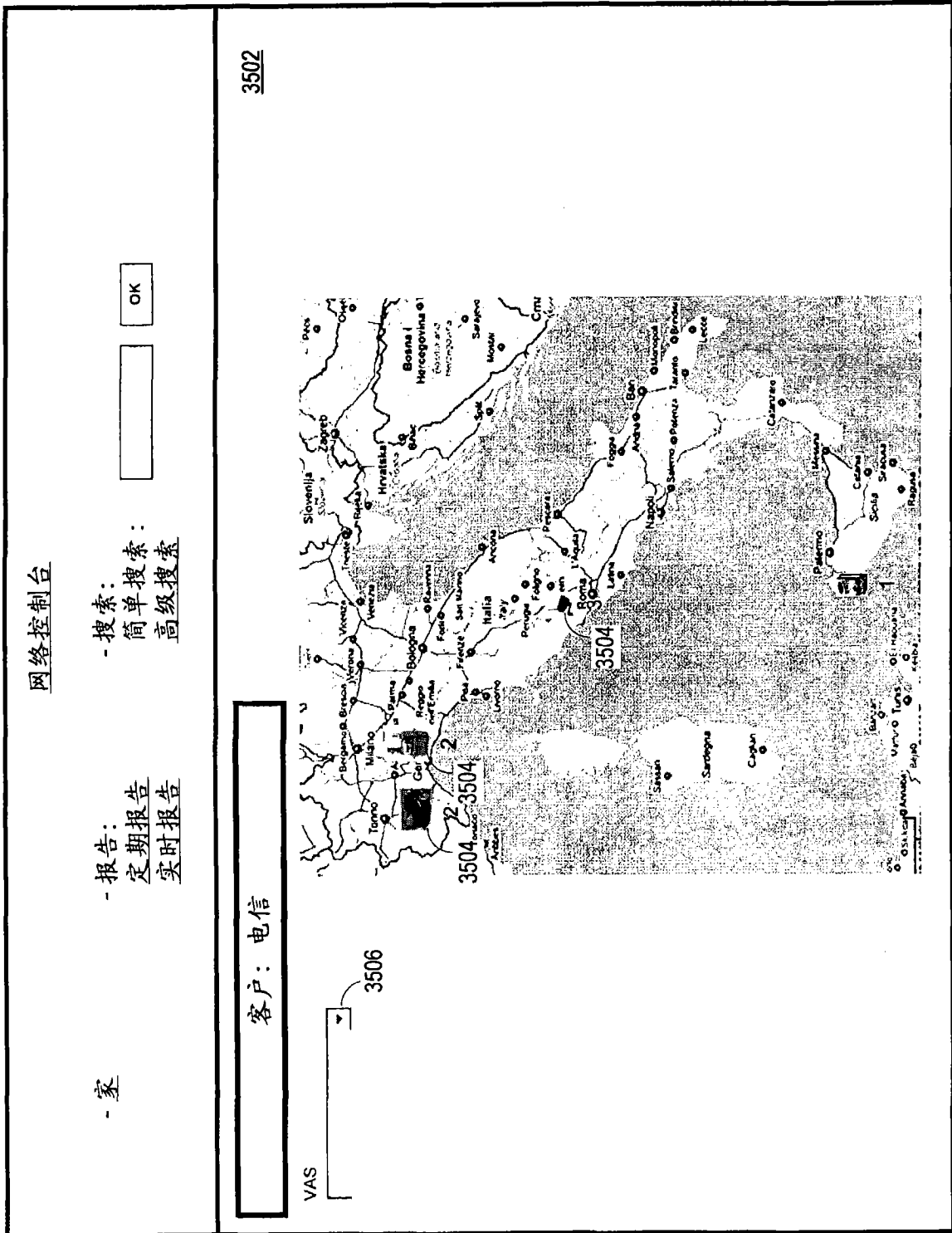


图 35

2600 ↗

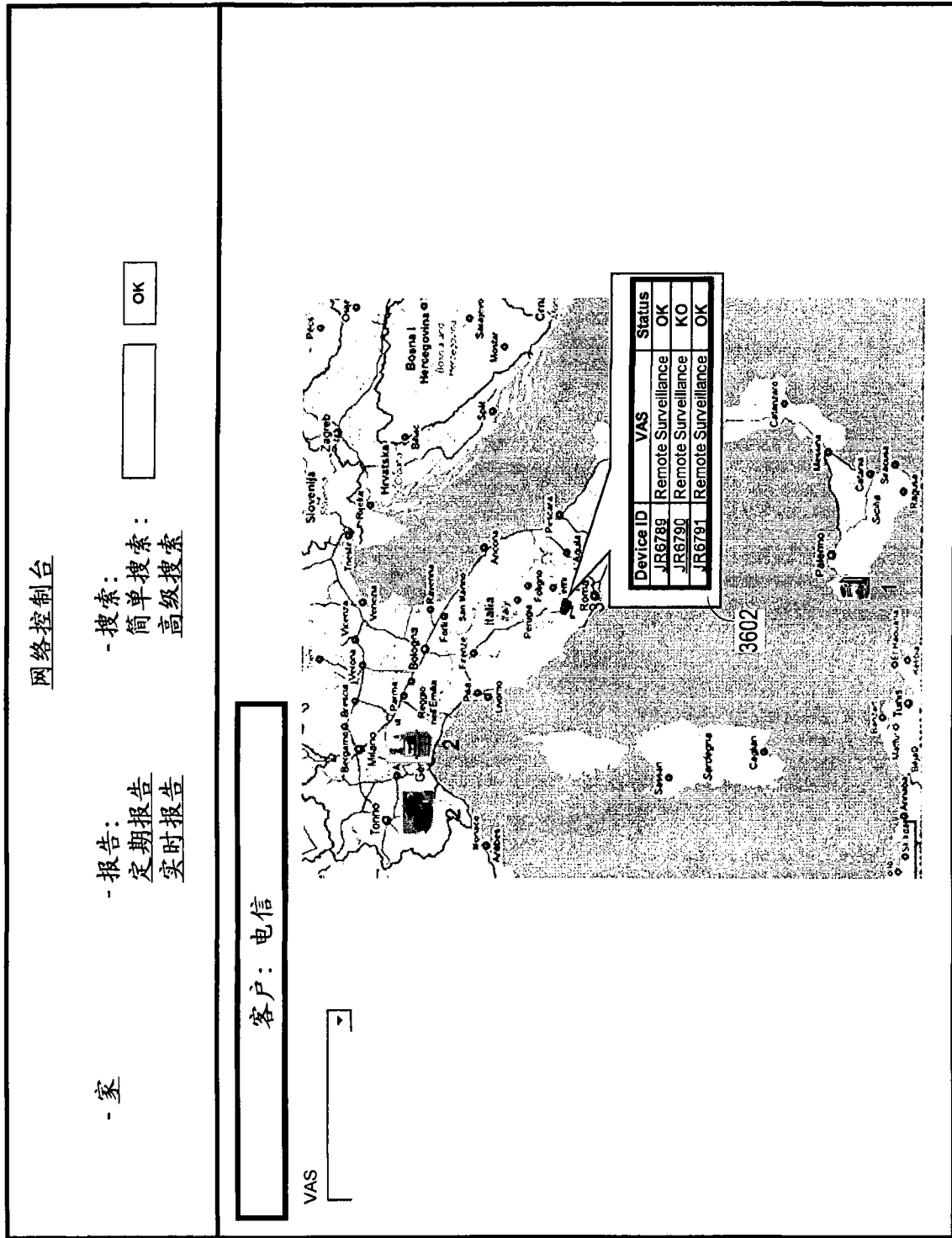


图 36

2600 ↗

- 家

网络控制台

- 报告: - 搜索:

定期报告 简单搜索:

实时报告 高级搜索

移动设备: JR6789

Client: Telecom
 VAS: Remote Surveillance
 Project: TI-SPLB
 Site: Rome (41.901375,12.501165)
 Device Brand: Telit
 Device Type: GT863-PY

Last 10 days Report

概要
OK
20090729 11:32am
OK
OK
OK
OK

SIM	Device HW
MSISDN	设备 ID
APN	设备日期时间
网络上存在性	最后重置
激活状态	最后配置更新
SMS 信道	操作状态
GPRS 信道	供电级别
应用	存储器使用
模块 1 状态	CPU 负载
立即检查状态	

图 37

2600 →

网络控制台

- 家

- 报告:

定期报告

实时报告

- 搜索:

简单搜索

高级搜索

结果

检查	设备 ID	MSISDN	设备类型	设备品牌	客户	VAS	项目	站点	最后更新	状态
3804	DR1721	3345481000	IP Camera	Genetec	Telecom	Remote Surveillance	T1-SPLB	Rome (41.906964,12.485254)	20090730 11:30am	NOT Present
	DR1722	3342381040	IP Camera	Genetec	Telecom	Remote Surveillance	T1-SPLB	Rome (41.906964,12.485254)	20090720 06:30pm	OK
	DR1723	3345481123	IP Camera	Exacq	Telecom	Remote Surveillance	T1-SPLB	Rome (41.906964,12.485254)	20090729 10:20am	OK
	DR1724	3345423400	IP Camera	Exacq	Telecom	Remote Surveillance	T1-SPLB	Rome (41.906964,12.485254)	20090728 07:30am	OK

约 4 个结果中的结果 1-4

页面 | < < 1 > >

3802

3808

3804

3806

3806

图 38

2600 ↗

网络控制台

- 家

- 报告: 定期报告 实时报告

- 搜索: 简单搜索 高级搜索

OK

高级检索标准

设备 ID	设备类型	设备品牌	状态
MSISDN	客户	VAS	项目
之前的最后重置	之后的最后重置	之前的最后配置	之后的最后配置

提交 清除

3902

图 39

2600 →

网络控制台

家 报告 搜索:
 定期报告 简单搜索:
 实时报告 高级搜索

定期报告

报告名称	频率	用户	创建日期	命令
报告电信	每天	m.abc	20090723	<input checked="" type="checkbox"/> ⊖ <input checked="" type="checkbox"/>
报告全部	每周	g.xyz	20090721	<input checked="" type="checkbox"/> ⊖ <input checked="" type="checkbox"/>

4002

定义报告

调度: 每天 每周 每月

4004

域选择

报告名称:

4006

设备ID 设备品牌 设备类型 MSISDN 客户 VAS
 项目 最后重置 最后配置 供电级别 CPU负载 存储器使用
 更新

4008

移动设备选择

设备ID: 设备品牌:
 客户: VAS: 项目:

分发列表

abc@telecom.com
xyz@telecom.com

4010

4012

图 40

2600 ↗

网络控制台

— 报告 — — 搜索 — OK

— 定期报告 — 简单搜索: 高级搜索

— 实时报告 —

报告名称	频率	用户	创建日期	命令
新报告	每天	g.Leopardi	20090730	
报告电信	每天	m.Rossi	20090723	
报告全部	每周	g.Verdi	20090721	

4102

定义报告

— 报告名称 — 报告名称 新报告

— 频率 — 每天 每周 每月

— 设备选择 —

设备 ID 设备品牌 设备类型 MSISDN 客户 VAS
 项目 最后重置 最后配置 供电级别 CPU 负载 存储器使用
 更新

— 移动设备选择 —

设备 ID 设备品牌 设备类型 项目
 客户 VAS 项目

— 分发列表 —

abc@telecom.com
xyz@telecom.com

提交 清除

图 41

2600 →

网络控制台

- 家 - 报告: - 搜索: OK

定期报告 简单搜索: 高级搜索

实时报告

4202 定义报告

4204 域选择

设备 ID 设备品牌 设备类型 MSISDN 客户 VAS

项目 最后重置 最后配置更新 供电级别 CPU 负载 存储器使用

4206 移动设备选择

设备 ID 设备品牌

客户 VAS 项目

4208 日期范围

从 至

July 2010
1234567
...

提交 清除

图 42

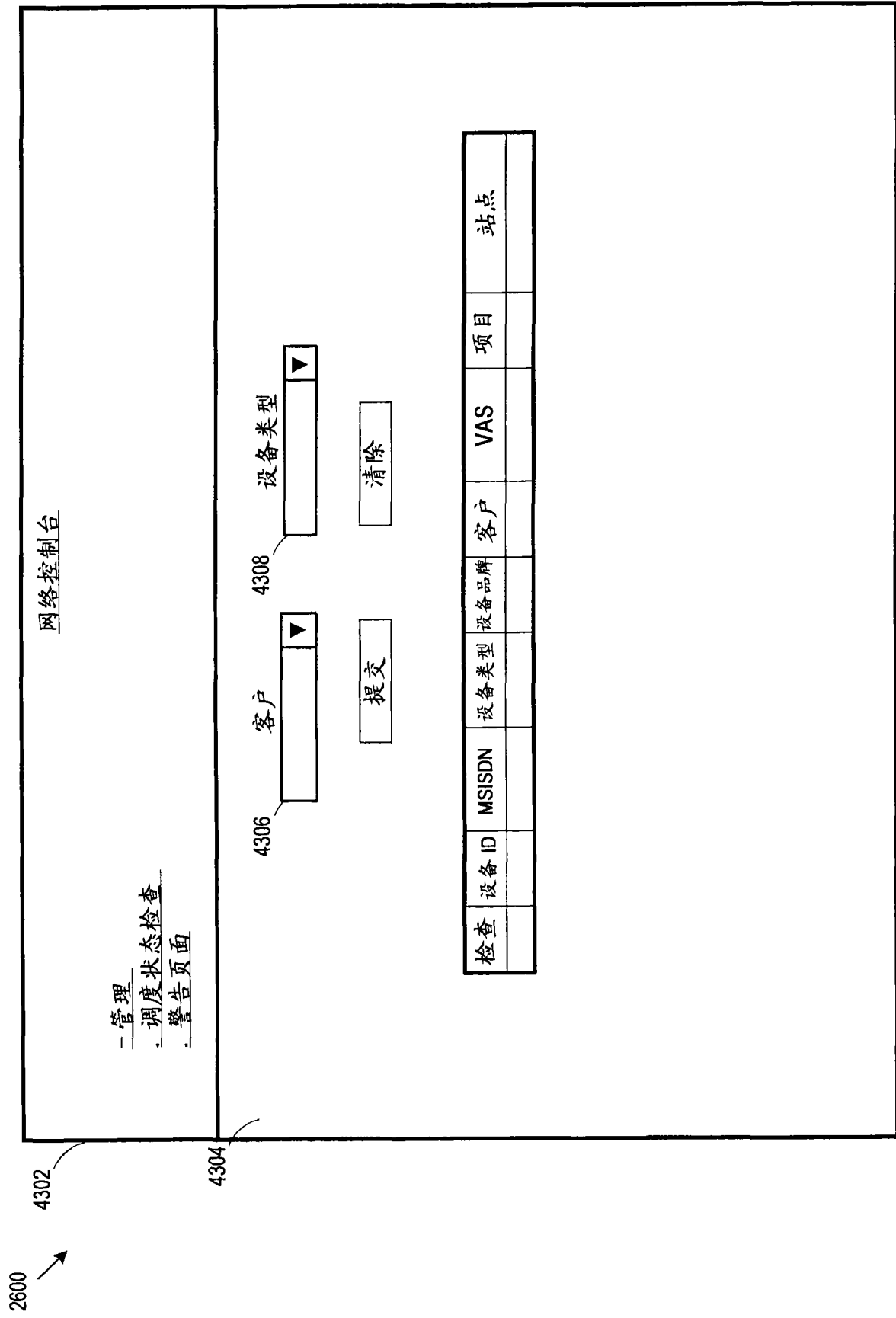
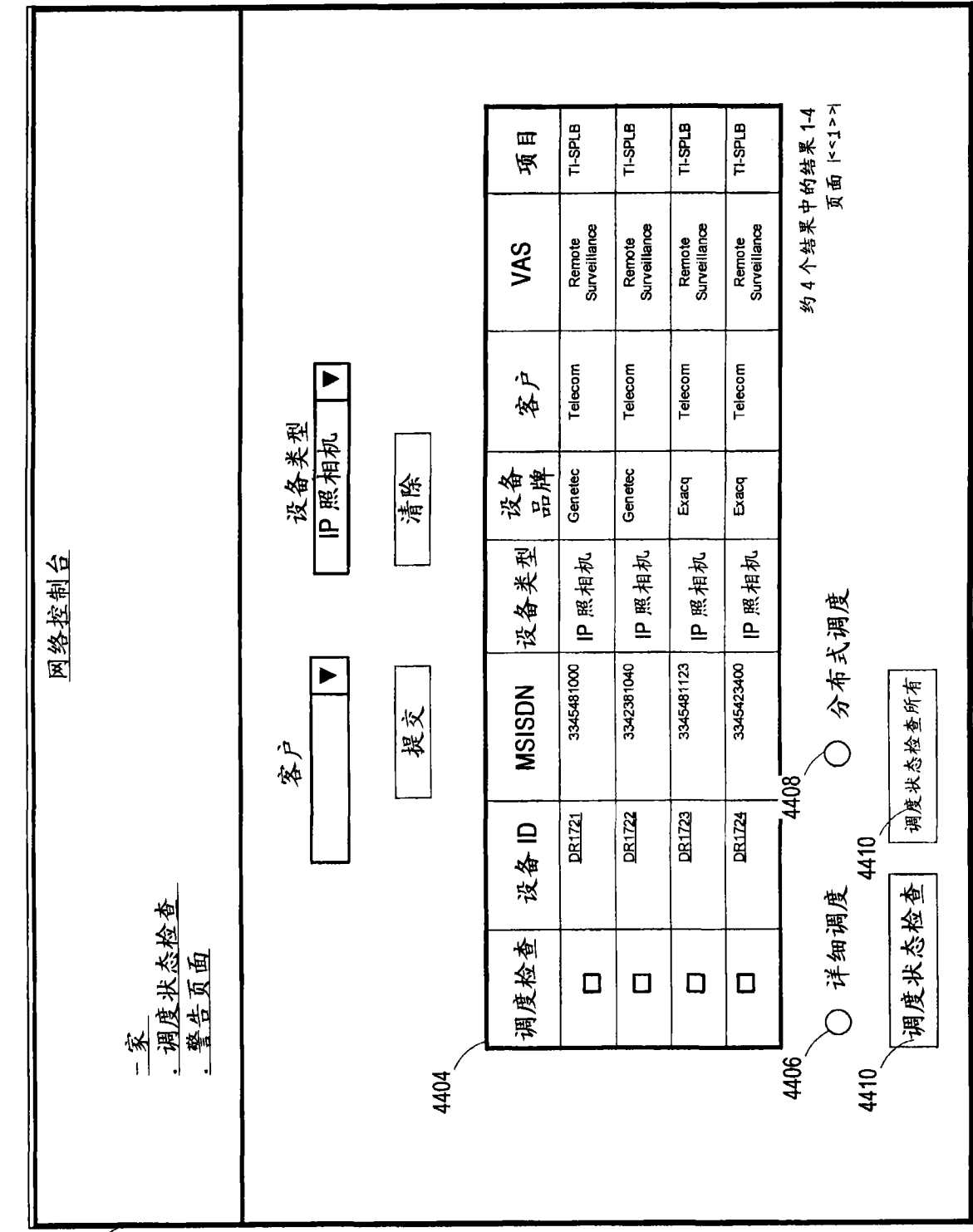


图 43



2600

图 44

2600



网络控制台

[家](#)
[调度状态检查](#)
[警告页面](#)

4502

Selected Client: **Telecom Italia**

- VAS: Vehicle Tracking
- VAS: Remote Surveillance
- VAS: Remote Metering
- VAS: Personal Tracking
- VAS: Health Diagnostics

调度检查	设备 ID	MSISDN	设备类型	设备品牌	客户	VAS	项目
<input type="checkbox"/>	DR1721	3345481000	IP 照相机	Genetec	Telecom	Remote Surveillance	Ti-SPLB
<input type="checkbox"/>	DR1722	3342381040	IP 照相机	Genetec	Telecom	Remote Surveillance	Ti-SPLB
<input type="checkbox"/>	DR1723	3345481123	IP 照相机	Exacq	Telecom	Remote Surveillance	Ti-SPLB
<input type="checkbox"/>	DR1724	3345423400	IP 照相机	Exacq	Telecom	Remote Surveillance	Ti-SPLB

[详细调度](#) [分布式调度](#)

约 4 个结果中的结果 1-4 页面 | << 1 >> |

调度状态检查

调度状态检查所有

图 45

2600 →

网络控制台

家
调度状态检查
警告页面

4602

调度

分钟	5, 15, 25, 35, 45, 55
小时	1am, 2am, 3am, 4am, 5am, 6am
一月中的天	None
月份	Jul
一周中的天	Sat, Sun

提交 清除

图 46

2600 ↗

网络控制台

[家](#)
[调度状态检查](#)
[警告页面](#)

调度

时间范围

从 至

频率

每天
 每星期
 每月

4702

图 47

2600 →

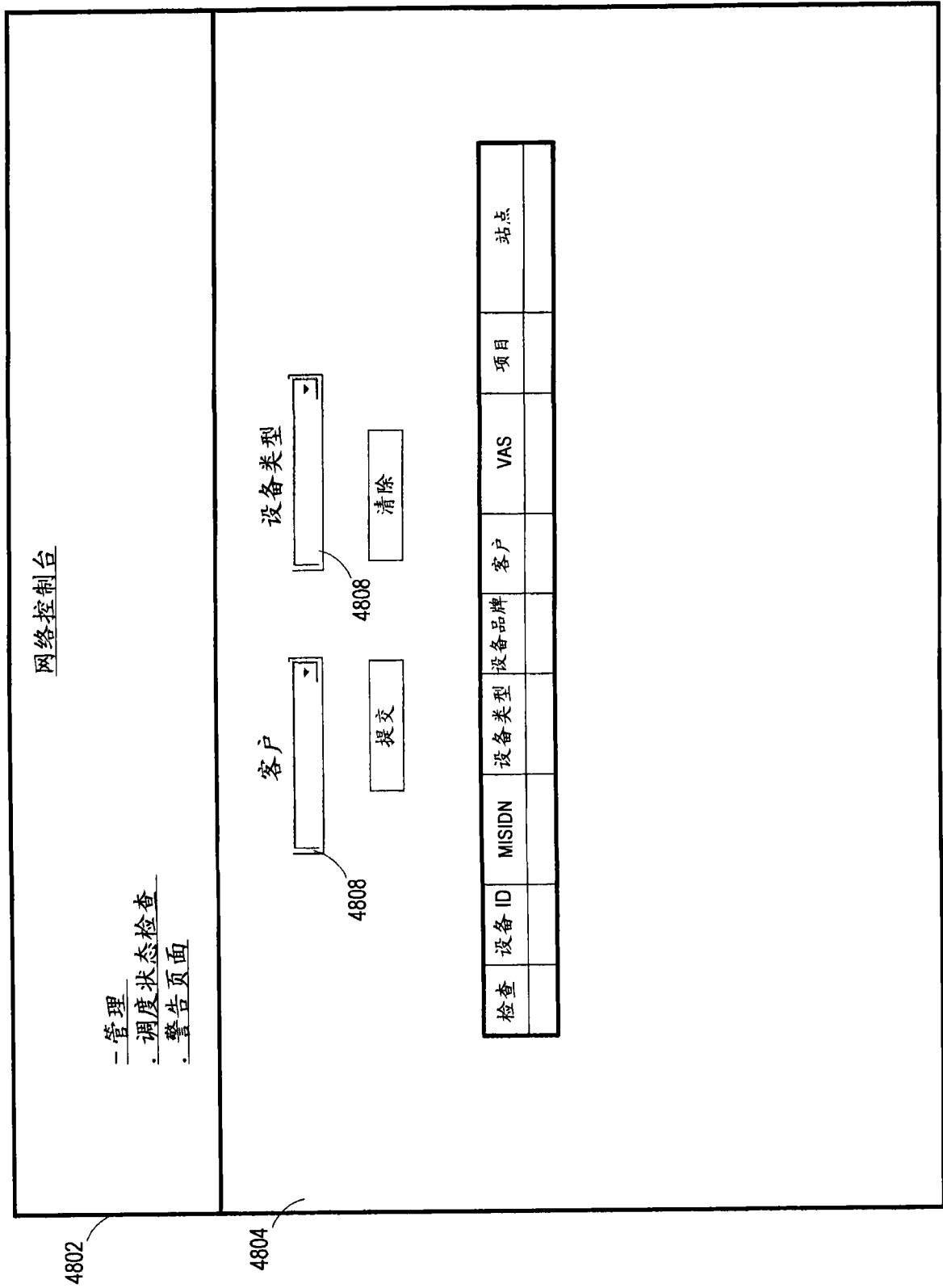


图 48

2600 ↗

网络控制台

管理
· 调度状态检查
· 警告页面

警告

客户

4904 邮件警告 4904 SMS 警告

4906 4908

如果定期检查后回答丢失, 则警告

如果实时检查后回答丢失, 则警告

如果状态是 KO, 则警告

邮件地址: 电话号码:

邮件地址: 电话号码:

邮件地址: 电话号码:

4902

图 49