



(12) 发明专利

(10) 授权公告号 CN 102136933 B

(45) 授权公告日 2013. 08. 28

(21) 申请号 201010505878. 2

(22) 申请日 2010. 09. 30

(73) 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
总部办公楼

(72) 发明人 张永靖 卞永刚 黄成 金雷
牟伦建

(74) 专利代理机构 深圳市深佳知识产权代理事
务所(普通合伙) 44285

代理人 彭愿洁 李文红

(51) Int. Cl.

H04L 12/24 (2006. 01)

H04L 29/06 (2006. 01)

(56) 对比文件

CN 101083537 A, 2007. 12. 05, 全文.

CN 101477535 A, 2009. 07. 08, 全文.

CN 101730123 A, 2010. 06. 09, 全文.

审查员 冯骥

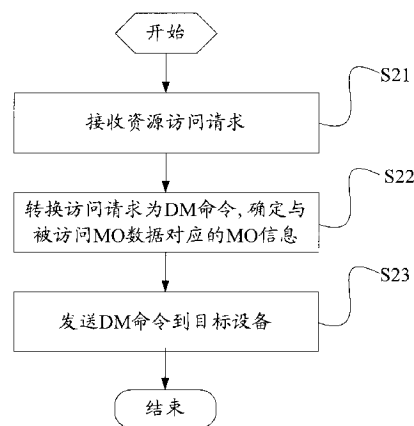
权利要求书3页 说明书15页 附图8页

(54) 发明名称

设备管理方法、中间件及机器通信平台、设备
和系统

(57) 摘要

本发明公开了一种设备管理方法、中间件及机器通信 M2M 平台、设备和系统,该方法包括:通过资源访问接口接收资源访问请求,该资源访问请求包含指示被访问管理对象 MO 数据资源的存储位置的 URI;参考预先建立的 MO 数据资源访问请求与 DM 命令的映射关系,转换对 MO 数据资源的资源访问请求为对应的 DM 命令,并根据预先建立的 MO 数据资源与 MO 信息的映射关系确定与被访问 MO 数据对应的 MO 信息;发送 DM 命令到与 URI 对应的目标设备,以管理与被访问 MO 数据对应的 MO 信息。本发明实施例采用统一的资源访问接口连接 M2M 应用与 M2M 平台,使得 M2M 应用能够访问不同 M2M 平台进行设备管理,实现了端到端的设备管理及相关业务应用。



1. 一种设备管理方法,其特征在于,包括:

通过资源访问接口接收资源访问请求,所述资源访问请求包含:用于指示被访问管理对象 MO 数据资源的存储位置的唯一资源标识符 URI;

参考预先建立的所述 MO 数据资源的资源访问请求与设备管理 DM 命令的映射关系,转换所述资源访问请求为对应的 DM 命令,并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据资源对应的 MO 信息;

发送所述 DM 命令到与所述 URI 对应的目标设备,以管理所述与被访问 MO 数据资源对应的 MO 信息。

2. 根据权利要求 1 所述的方法,其特征在于,还包括:

接收所述目标设备执行所述 DM 命令后生成并返回的结果数据;

返回所述结果数据以响应所述资源访问请求。

3. 根据权利要求 2 所述的方法,其特征在于,所述发送所述 DM 命令到目标设备的过程包括:

判断是否存在管理服务器 DMS 与所述目标设备的会话,若是,则控制所述 DMS 在所述会话中发送所述 DM 命令到所述目标设备;否则,发送会话控制命令给所述 DMS,以控制所述 DMS 建立与所述目标设备的会话并依据建立的会话发送所述 DM 命令到所述目标设备。

4. 根据权利要求 3 所述的方法,其特征在于,所述发送所述 DM 命令到目标设备的过程还包括:

判断预设会话时间内是否存在后续 DM 命令,若存在后续 DM 命令,则继续执行,若不存在,则关闭此次会话。

5. 根据权利要求 3 所述的方法,其特征在于,所述预先建立所述 MO 数据资源与 MO 信息映射关系的过程包括:

将设备中 MO 信息用可扩展标记语言 XML 文档进行描述;

在平台上构建与所述 MO 信息对应的 MO 数据资源,并为所述资源分配 URI。

6. 根据权利要求 3 所述的方法,其特征在于,所述依据建立的会话发送所述 DM 命令到所述目标设备具体为按照资源访问协议封装所述 DM 命令后通过所述建立的会话发送至所述目标设备。

7. 根据权利要求 1 所述的方法,其特征在于,所述预先建立所述 MO 数据资源与 MO 信息映射关系的过程包括:

将设备中的 MO 信息用可扩展标记语言 XML 文档进行描述;

在设备上构建与所述 MO 信息对应的 MO 数据资源,并为所述资源分配 URI。

8. 根据权利要求 7 所述的方法,其特征在于,还包括:

向平台发送指示拥有所述 MO 数据资源的声明。

9. 根据权利要求 1-7 中任意一项所述的方法,其特征在于,所述资源访问请求类型包括:基本资源访问请求,所述基本资源访问请求类型包括:获取请求、替换请求和删除请求。

10. 根据权利要求 9 所述的方法,其特征在于,所述资源访问请求类型还包括:依据超文本链接协议进行扩展的扩展资源访问请求,所述扩展资源访问请求类型包括:执行请求、复制请求、原子操作请求、顺序原子操作请求、异步上报请求、群组设备操作请求、条件操作

请求和复合条件操作请求。

11. 一种中间件,其特征在于,包括:

资源访问请求接收单元,用于通过资源访问接口接收资源访问请求,所述资源访问请求包含:用于指示被访问管理对象 MO 数据资源的存储位置的唯一资源标识符 URI;

控制命令转换单元,用于参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系,转换所述资源访问请求为对应的 DM 命令,并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据资源对应的 MO 信息;

控制命令发送单元,用于发送所述 DM 命令到所述 URI 对应的目标设备,以管理所述与被访问 MO 数据资源对应的 MO 信息。

12. 根据权利要求 11 所述的中间件,其特征在于,还包括:

结果数据接收单元,用于接收所述目标设备执行所述 DM 命令后生成并返回的结果数据;

结果数据返回单元,用于返回所述结果数据至所述资源访问请求的发送端。

13. 根据权利要求 11 或 12 所述的中间件,其特征在于,所述控制命令发送单元包括:

判断单元,用于判断是否存在管理服务 DMS 与所述目标设备间的会话;

会话建立单元,用于当不存在所述 DMS 与所述目标设备间的会话时,控制所述 DMS 建立与所述目标设备的会话;

发送单元,用于依据所述建立的会话发送所述 DM 命令到所述目标设备。

14. 根据权利要求 13 所述的中间件,其特征在于,所述控制命令发送单元还包括:

会话过程判断单元,用于判断预设会话时间内是否存在后续 DM 命令,若存在后续 DM 命令,则指示所述判断单元继续执行,若不存在,则发送指示关闭此次会话的信息。

15. 根据权利要求 13 所述的中间件,其特征在于,所述控制命令发送单元还包括:

封装单元,用于按照资源访问协议封装所述控制命令转换单元转换后的 DM 命令后提供给所述发送单元。

16. 一种机器通信 M2M 平台,其特征在于,具有如权利要求 11-15 任意一项所述的中间件。

17. 一种机器通信 M2M 设备,其特征在于,具有如权利要求 11-15 任意一项所述的中间件。

18. 一种机器通信 M2M 系统,其特征在于,包括:M2M 设备和中间件,其中:

所述中间件用于,接收 M2M 应用通过资源访问接口发送的携带用于指示被访问管理对象 MO 数据资源的存储位置的唯一资源标识符 URI 的资源访问请求,参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系,转换所述资源访问请求为对应的 DM 命令,并在根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据资源对应的 MO 信息后,发送所述 DM 命令到所述 URI 对应的 M2M 设备,以管理所述与被访问 MO 数据资源对应的 MO 信息;

所述 M2M 设备用于,接收所述 DM 命令并执行,获得结果数据,将所述结果数据返回给所述中间件。

19. 根据权利要求 18 所述的系统,其特征在于,所述中间件设置在 M2M 平台上。

20. 根据权利要求 18 所述的系统,其特征在于,所述中间件包括:设置于 M2M 平台上的

平台中间件和设置于所述 M2M 设备上的设备中间件,所述平台中间件和设备中间件配合实现所述中间件的功能。

21. 根据权利要求 20 所述的系统,其特征在于,所述设备中间件与所述设备中的设备管理客户端 DMC 为分体设计,所述设备中间件通过所述 DMC 开放的本地访问接口与其连接;或者,所述设备中间件与所述 DMC 为一体化设计;或者,所述设备中间件与设备管理服务器 DMS 为一体化设计,采用开放移动联盟所定义的设备管理 OMA-DM 协议的访问接口与所述 DMC 相连接。

22. 根据权利要求 21 所述的系统,其特征在于,所述平台中间件与所述 DMS 为一体化设计,或者所述平台中间件与所述 DMS 采用所述 DMS 开放的本地访问接口相连。

设备管理方法、中间件及机器通信平台、设备和系统

技术领域

[0001] 本发明涉及通信领域,尤其涉及一种设备管理方法、中间件及机器通信(Machine-to-Machine Communications, M2M)平台、设备及系统。

背景技术

[0002] M2M 是一种以机器智能交互为核心的网络化的应用与服务。它通过在机器内部嵌入无线或有线通信模块以及应用处理逻辑,实现无需人工干预的数据通信,以满足用户对机器的监控、指挥调度、数据采集和测量等方面的信息化需求。图 1 是一种典型的 M2M 系统架构。其中,各种 M2M 终端(如传感器、微控制器等)直接或经过 M2M 网关远程接入到 M2M 业务平台,而各种 M2M 应用(如电力抄表、智能交通等)则通过 M2M 业务平台所提供的业务能力获取 M2M 终端采集的数据或对 M2M 终端进行远程的控制和管理。

[0003] 远程设备管理是 M2M 系统的一个重要功能,目前,基于广域网的远程设备管理技术主要有开放移动联盟(Open Mobile Alliance, OMA)所定义的设备管理(Device Management, DM)规范。它能够通过操作 M2M 设备(包括 M2M 网关和 M2M 终端)上的管理对象(Management Object, MO)数据,实现对 M2M 设备的远程管理。现有的 DM 系统属于辅助的管理平面功能,与终端用户日常的业务应用过程相对独立。用户发现终端故障后发起 DM 过程(如拨打客服电话),然后由客服或管理员通过控制设备管理服务器(DM Server, DMS)完成 DM 任务。但是,由于 M2M 设备数量巨大,且往往无人值守,需要 M2M 应用经常主动地监测和发现 M2M 设备故障问题,并进行相应的升级维护。因此需要 M2M 平台将 DM 功能作为一种公共业务能力,采用统一的访问接口开放给 M2M 应用,以实现端到端的设备管理及相关业务应用。

发明内容

[0004] 有鉴于此,本发明提供一种设备管理方法、中间件及 M2M 平台、设备和系统,以实现 M2M 应用通过统一的资源访问接口访问不同 M2M 平台的 DM 能力的目的,其具体方案如下:

[0005] 一种设备管理方法,包括:

[0006] 通过资源访问接口接收资源访问请求,所述资源访问请求包含:用于指示被访问管理对象 MO 数据资源的存储位置的唯一资源标识符 URI;

[0007] 参考预先建立的所述 MO 数据资源的资源访问请求与设备管理 DM 命令的映射关系,转换所述资源访问请求为对应的 DM 命令,并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据对应的 MO 信息;

[0008] 发送所述 DM 命令到与所述 URI 对应的目标设备,以管理所述与被访问 MO 数据对应的 MO 信息。

[0009] 一种中间件,包括:

[0010] 资源访问请求接收单元,用于通过资源访问接口接收资源访问请求,所述资源访

问请求包含：用于指示被访问管理对象 MO 数据资源的存储位置的唯一资源标识符 URI；

[0011] 控制命令转换单元，用于参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系，转换所述资源访问请求为与其对应的 DM 命令，并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据资源对应的 MO 信息；

[0012] 控制命令发送单元，用于发送所述 DM 命令到所述 URI 对应的目标设备，以管理所述与被访问 MO 数据资源对应的 MO 信息。

[0013] 一种具有上述中间件的机器通信 M2M 平台和 M2M 设备。

[0014] 一种机器通信 M2M 系统，包括：M2M 设备和中间件，其中：

[0015] 所述中间件用于，接收 M2M 应用通过资源访问接口发送的携带用于指示被访问管理对象 MO 数据资源的存储位置的唯一资源标识符 URI 的资源访问请求；参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系，转换所述资源访问请求为与其对应的 DM 命令，并在根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据资源对应的 MO 信息后；发送所述 DM 命令到所述 URI 对应的 M2M 设备，以管理所述与被访问 MO 数据资源对应的 MO 信息；

[0016] 所述 M2M 设备用于，接收所述 DM 命令并执行，获得结果数据，将所述结果数据返回给所述中间件。

[0017] 从上述的技术方案可以看出，本发明实施例公开的设备管理方法中，通过采用统一的资源访问接口，连接 M2M 应用与 M2M 平台，使得 M2M 应用能够访问不同 M2M 平台的 DM，实现了端到端的设备管理及相关业务应用。

附图说明

[0018] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0019] 图 1 为本发明实施例 1 公开的典型的 M2M 系统结构示意图；

[0020] 图 2 为本发明实施例公开的设备管理方法流程图；

[0021] 图 3 为本发明实施例公开的又一设备管理方法流程图；

[0022] 图 4 为本发明实施例公开的又一设备管理方法流程图；

[0023] 图 5 为本发明实施例公开的又一设备管理方法流程图；

[0024] 图 6 为本发明实施例公开的 MO 数据资源与 MO 信息映射关系建立方法流程图；

[0025] 图 7 为本发明实施例公开的又一 MO 数据资源与 MO 信息的映射关系建立方法流程图；

[0026] 图 8 为本发明实施例公开一种中间件的结构示意图；

[0027] 图 9 为本发明实施例公开另一种中间件的结构示意图；

[0028] 图 10 为图 8 或图 9 中控制命令发送单元的一种结构的示意图；

[0029] 图 11 为图 8 或图 9 中控制命令发送单元的另一种结构的示意图；

[0030] 图 12 为本发明实施例公开的一种 M2M 系统的结构示意图；

[0031] 图 13 为本发明实施例公开的另一种 M2M 系统的结构示意图。

具体实施方式

[0032] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0033] 本发明公开了一种设备管理方法,通过采用资源访问接口实现 M2M 应用访问不同 M2M 平台的 DM 的目的。

[0034] 本发明中所述的资源访问接口可以为现有技术中任何一种基于资源的访问协议接口,例如 HTTP(HyperText Transfer Protocol,超文本传输)接口,XCAP(XML Configuration Access Protocol,XML 配置访问协议)接口,以及在现有的基于资源的访问协议接口基础上进行扩展后的接口。本发明并不限定该方法应用于 M2M 系统,其同样可以应用于其他的通信系统。

[0035] 其具体实施方式如下所述:

[0036] 本发明公开的一种设备管理方法的一种实现流程如图 2 所示,包括以下步骤:

[0037] 步骤 S21、通过资源访问协议接口接收资源访问请求,所述资源访问请求包含:用于指示被访问 MO 数据资源的存储位置的唯一资源标识符 URI。

[0038] 在实现本流程时,要预先根据 DM TND(S(Tree and Description Serialization)规范,建立与 M2M 设备上的 MO 数据信息对应的 MO 数据资源,所述的 MO 数据资源可以为 XML 配置访问协议(XML Configuration Access Protocol,XCAP)XCAP 资源,也可以为其他类型的数据资源,然后利用与数据资源对应的资源访问方法,以及按照该方法建立的资源访问接口,实现对 MO 数据资源的访问。MO 资源可以映射到 M2M 平台或 M2M 设备上,其具体的位置通过 URI 来确定。接收的资源访问请求可以是独立于平台和设备之外的中间件,接收平台转发的 M2M 应用发送的请求,也可以是设置于平台上的中间件直接接收的 M2M 应用发送的请求,同样也可以是设置在平台上的中间件或者平台接收到请求后,转发给设置于设备上的中间件的请求。其具体情况依据实际需要而定。

[0039] 步骤 S22、参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系,转换所述对 MO 数据资源的资源访问请求为与其对应的 DM 命令,并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据对应的 MO 信息;

[0040] 由于 M2M 应用通过资源访问接口发送资源访问请求,其对象为 MO 资源数据,而实际需要的是对与 MO 资源数据对应的 MO 信息的控制,所以,需要按照预先设定的 DM 命令与资源访问请求的映射关系,将资源访问请求转换为 MO 数据信息可以识别的 DM 命令,以实现 MO 信息的管理及操作。

[0041] 现有技术中的资源访问接口包括:HTTP 接口、XCAP 接口等。其中 XCAP 接口对应的资源请求只提供了基本资源访问请求,基本资源访问请求中包括:GET, PUT, DELETE 三种类型,分别表示获取、替换或增加、删除相应的 XCAP 资源的操作,为了能够将现有的 DM 命令映射为对应的 XCAP 请求,还需要对 XCAP 方法进行一定的扩展。例如依据 HTTP 协议进行扩展,得到扩展资源访问请求,扩展资源访问请求的类型包括:执行请求、复制请求、原子操作请求、顺序原子操作请求、异步上报请求、对系统中多个设备进行的群组设备操作请求、条

件操作请求和复合条件操作请求等,以实现 MO 数据资源相应的操作,中间件通过包含在资源访问请求中的数据来判定当前需要对 MO 数据进行的请求类型,然后将其转换为与该请求类型对应的 DM 命令。

[0042] 同时,本发明预先将 MO 信息映射成为了 MO 数据资源,例如 XCAP 资源,使得资源访问请求可以直接作用于资源,然后根据资源访问请求与 DM 命令间的映射关系,以及 MO 信息与 XCAP 资源间的映射关系,实现对 MO 信息的 DM 命令操作。

[0043] 步骤 S23、发送所述 DM 命令到与所述 URI 对应的目标设备,以管理所述与被访问 MO 数据对应的 MO 信息。

[0044] 将转换后的 DM 命令发送到被访问的 MO 数据资源对应的目标设备,该过程可以利用现有技术中设备管理服务器 DMS 与目标设备中设备管理客户端 DMC 之间的开放移动联盟 (Open Mobile Alliance, OMA) 所定义的设备管理 (Device Management, DM) 规范协议完成,也可以利用资源访问接口,如 XCAP 或 HTTP 接口完成。目标设备接收命令后,执行命令,管理相应的 MO 信息,从而实现设备管理功能。

[0045] 本实施例中公开的设备管理方法,采用将 M2M 设备上的 MO 信息映射为 MO 数据资源,并且通过预先建立的资源访问请求与 DM 命令间的映射关系,使得外部 M2M 应用可以通过资源访问接口访问控制 MO 信息,实现了 M2M 应用通过开放的统一接口管理远程设备的目的。

[0046] 本实施例中将 DM 命令发送后,设备可能成功接收 DM 命令,并执行所述 DM 命令后生成结果数据,然后将结果数据返回,以表示完成了一次成功的管理过程,例如,假设资源访问请求的类型为 GET,获取目标设备的运行数据,则设备执行相应的 DM 命令后,其结果数据为目标设备的运行数据。当然,设备也有可能没有成功接收 DM 命令,这种情况下,M2M 应用可以在等待预设时间没有接收到返回消息后继续发送资源访问请求,也可以结束操作,具体实现方式依据应用场景而设定。

[0047] 上述实施例公开的方法可以为独立于平台和设备外的中间件,为实现平台和设备间的通信而进行的操作,同样也可以为设置在平台上的平台中间件以及设置在设备上的设备中间件相互配合,为实现上述目的而进行的操作,其方式都是将 M2M 应用发送的资源访问请求转换为 DM 命令,只是在具体实施过程中,会根据不同的情况而有所差别。例如,当 M2M 应用发送的资源访问请求中,被访问 MO 数据资源存储在平台上时,为了更方便的将资源访问请求转换为 DM 命令,平台中间件可以按照本实施例公开的又一设备管理方法,实现平台和设备间的通信,对设备进行管理,本实施例中以 XCAP 资源访问接口为例实现 M2M 应用与中间件间的连接,其流程如图 3 所示,包括如下步骤:

[0048] 步骤 S31、平台中间件接收 M2M 应用通过 XCAP 资源访问接口发送的资源访问请求,所述资源访问请求包含用于指示被访问 MO 数据资源的存储位置的唯一资源标识符 URI;

[0049] 步骤 S32、根据所述 URI 判断被访问 MO 数据资源的存储位置,若其存储位置为平台,则执行步骤 S33a,若其存储位置为设备,则执行步骤 S33b;

[0050] 步骤 S33a、参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系,转换所述对 MO 数据资源的资源访问请求为与其对应的 DM 命令,并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据对应的 MO 信息,执行步骤 S34;

[0051] 存储位置为平台,平台中间件执行转换 DM 命令以及确定与所述被访问 MO 数据对

应的 MO 信息的操作。

[0052] 步骤 S33b、转发所述资源访问请求给设备中间件,由所述中间件进行处理;由于存储位置为设备,因此为了更加方便地转换 DM 命令,本实施例中采用将资源访问请求转发给设备中间件,由设备中间件来完成后续的操作,具体操作过程将在下文详细介绍。

[0053] 步骤 S34、判断是否存在 DMS 与所述目标设备的 DMC 间的 DM 会话,若是,执行步骤 S35a,若否,执行步骤 S35b;

[0054] 本实施例中采用 DMS 与目标设备的 DMC 间的会话发送 DM 命令,首先判断是否已经存在两者间的 DM 会话。

[0055] 步骤 S35a、依据当前 DM 会话发送所述 DM 命令到目标设备的 DMC。

[0056] 步骤 S35b、发送 DM 会话控制命令给 DMS,控制其建立与所述目标设备的 DMC 的 DM 会话;

[0057] 所述 M2M 设备中的 DMC 尚未与 DMS 建立 DM 会话,指示 DMS 向 DMC 发送 DM 通知消息,以便建立 DMS 与 DMC 间的 DM 会话。本实施例并不限定创建会话的过程如步骤 S35b 所述,同样可以采用其他方案,只要能够建立起 DMS 与 DMC 之间的会话的操作都可行。

[0058] DMC 根据接收到的 DM 通知消息,按照现有技术中 OMA-DM 协议规范的要求向 DMS 发送 DM 会话建立消息,其中携带必要的 M2M 设备信息 (DevInfo),例如 M2M 设备标识 dev1,表明 DMS 与所述 M2M 设备上的 DMC 之间的 DM 会话已经建立。

[0059] 步骤 S36、依据建立的 DM 会话发送所述 DM 命令到目标设备的 DMC。

[0060] DMS 向目标 DMC 发送相应的 DM 命令,例如指示 DMS 获取所述 M2M 设备上的与所述被访问 MO 数据对应的目标 MO 信息。

[0061] 步骤 S37、接收目标设备的 DMC 执行所述 DM 命令后生成并返回的结果数据,以管理所述与被访问 MO 数据对应的 MO 信息;

[0062] DMC 根据接收到的 DM 命令消息,按照现有技术中 OMA-DM 协议规范的要求执行相应的 DM 功能和操作,并向 DMS 返回执行结果数据,例如 M2M 设备上目标 MO 的具体信息。DMS 将来自 DMC 的 DM 执行结果数据返回给平台中间件。

[0063] 步骤 S38、返回所述结果数据。

[0064] 平台中间件将所述结果数据返回给资源访问请求者,如 M2M 应用。

[0065] 本实施例中加入了通过创建 DM 会话发送 MO 控制命令的过程,由于现有技术中的 M2M 系统中, DMS 与 DMC 之间可能已经具有可用于建立 DM 会话的基于 OMA-DM 协议规范的标准接口,因此,本实施例中,重用了所述标准接口,只需要建立 M2M 应用与 M2M 平台间的联系,然后利用所述标准接口所建立的 DM 会话连接发送 MO 控制命令,从而保证了在不改变现有系统中的接口关系的前提下,实现了 M2M 应用通过开放的统一接口管理远程设备的目的。

[0066] 本实施例中,为了避免没有 DM 命令时维护会话而产生不必要的开销,还可以在发送完 DM 命令之后,进一步判断预设会话时间内是否存在后续 DM 命令,若存在后续 DM 命令,则继续执行,若不存在,则关闭此次会话。

[0067] 需要说明的是,上述实施例中转发所述资源访问请求给设备中间件后(步骤 S33b),设备中间件同样也是按照类似上述流程执行,即:转换所述资源访问请求为 DM 命令,以及确定与 MO 数据资源对应的 MO 信息,除此之外,设备中间件相应的执行了其他相关

操作,从而实现可以与平台上的平台中间件相互配合,进而实现平台与设备间的通信,完成 M2M 应用通过平台对设备的管理的目的。具体流程如图 4 所示,包括如下步骤:

[0068] 步骤 S41、接收所述平台中间件转发的资源访问请求;

[0069] 步骤 S42、参考预先建立的所述 M0 数据资源访问请求与 DM 命令的映射关系,转换所述对 M0 数据资源的资源访问请求为与其对应的 DM 命令,并根据预先建立的所述 M0 数据资源与 M0 信息的映射关系确定与所述被访问 M0 数据对应的 M0 信息;

[0070] 步骤 S43、将所述资源访问请求发送给目标设备的 DMC,以管理所述与被访问 M0 数据对应的 M0 信息;

[0071] 将所述 DM 命令发送给其自身所在的目标设备的 DMC。

[0072] 步骤 S44、接收目标设备的 DMC 执行所述 DM 命令后返回的结果数据;

[0073] DMC 执行 DM 命令,对所述 M0 信息进行处理,得到结果数据,将结果数据返回给设备中间件。

[0074] 步骤 S45、将所述结果数据返回给所述平台中间件。

[0075] 需要说明的是,当由设备中间件处理所述资源访问请求时,图 4 所示流程中的步骤 S43-S44 为目标设备的内部模块间的交互过程,步骤 S45 为设备中间件与平台中间件间的交互过程,平台中间件接收到返回的结果数据后,才将其返回给发送请求的 M2M 应用,而不同于图 3 中步骤 S36-S38 中所示的,平台中间件与设备中间件的交互过程,以及平台中间件与 M2M 应用间的交互过程。

[0076] 由此可以看出,本实施例公开的数据控制方法,根据被访问 M0 数据资源的存储位置的不同而处理过程有所不同,当存储在平台上时,其执行 DM 命令转换、确定 M0 信息及发送过程的为平台中间件,当存储在设备上时,其执行 DM 命令转换、确定 M0 信息及发送过程的为设备中间件,其目的是使得平台中间件或设备中间件能够更方便的利用预设的 M0 数据资源访问请求与 DM 命令的映射关系转换成 DM 命令,利用 M0 数据资源与 M0 信息间的映射关系,找到被管理的 M0 信息。

[0077] 当然,本实施例并不限定上述处理过程,同样可以不考虑被访问 M0 数据资源的存储位置,而根据实际情况设定执行相关命令转化及发送的设备。

[0078] 本发明实施例公开的又一设备管理方法的流程如图 5 所示,同样以平台中间件为例,同样采用 XCAP 资源访问接口建立 M2M 应用与 M2M 平台间的联系,此时,在设备上同样设置有设备中间件,所述流程包括以下步骤:

[0079] 步骤 S51、接收通过 XCAP 资源访问接口发送的资源访问请求,所述资源访问请求包含:用于指示被访问 M0 数据资源的存储位置的唯一资源标识符 URI;

[0080] 步骤 S52、参考预先建立的所述 M0 数据资源的资源访问请求与 DM 命令的映射关系,转换所述对 M0 数据资源的资源访问请求为与其对应的 DM 命令,并根据预先建立的所述 M0 数据资源与 M0 信息的映射关系确定与所述被访问 M0 数据对应的 M0 信息;

[0081] 步骤 S53、按照 HTTP 接口协议封装所述 DM 命令;

[0082] 将所述 DM 命令按照 M2M 平台和设备之间的基于资源的 HTTP 接口协议进行封装,以改变其载体,实现 DM 命令在平台和设备间的传递。本实施例中的 HTTP 接口还可以为与基本 XCAP 资源访问请求对应的基本的 XCAP 接口协议,也可以为与扩展 XCAP 资源访问请求协议对应的扩展 XCAP 接口协议。该扩展 XCAP 接口协议可以与图 2 所示实施例中,对 XCAP 请

求进行的扩展相对应,也可以为按照其他方式对基本 XCAP 协议进行的扩展。

[0083] 步骤 S54、发送所述封装后的 DM 命令到目标设备；

[0084] 将封装转换后的 DM 命令发送到被访问的 MO 数据资源对应的目标设备,设备中间件在接收到按照上述 HTTP 协议封装后的 DM 命令后,将其解析为目标设备上的 DMC 可以识别的格式,例如,OMA-DM 协议规范格式,再将其发送给 DMC,使得 DMC 可以识别并执行该 DM 命令,以获得结果数据。

[0085] 步骤 S55、接收目标设备执行所述 DM 命令后生成并返回的结果数据；

[0086] 步骤 S56、返回所述结果数据。

[0087] 将所述结果数据返回给资源访问请求者,如 M2M 应用。

[0088] 本实施例公开的设备管理方法中,增加了对 DM 命令进行封装的过程。本实施例适用于现有的系统中,未部署 DMS 与 DMC 之间单独的通信接口的情况,此时不需要再去单独建立 DMS 与 DMC 之间的连接来发送 DM 命令,而是直接通过 M2M 平台和设备之间的 HTTP 接口协议封装 DM 命令,使其可以直接在 M2M 平台和设备之间传递,减少了 M2M 平台和 M2M 设备之间的通信接口,实现简单方便。

[0089] 本发明实施例公开了一种建立所述 MO 数据资源与 MO 信息间的映射关系的方法,本实施例中,与所述 MO 信息对应的 MO 数据资源建立在 M2M 平台上,其流程如图 6 所示,包括以下步骤：

[0090] 步骤 S61、将 DMC 上的 MO 信息用 XML 文档进行描述；

[0091] 首先根据 DM TNSD 规范,将 M2M 设备上的 MO 信息用 XML 文档进行描述。根据 DM TNSD 规范中提供的方法,任何一个 M2M 设备上的 MO 树都可以用一个 XML 文档描述,其中 MO 树上的每个 MO 节点都对应于 XML 文档中的一个元素,而 MO 节点的子节点和属性则可以由相应的 XML 子元素或属性来描述。

[0092] 步骤 S62、在 M2M 平台上构建与所述 MO 信息对应的 MO 数据资源,并为所述资源分配 URI。

[0093] 为了将用于描述 M2M 设备 MO 的 XML 文档构建成为 M2M 设备或 M2M 平台上的一种数据资源,所述 MO 数据资源为 XCAP 资源,或者其他类型的资源。需要为该 XML 文档分配一个与 M2M 设备或 M2M 平台相关的 URI,例如 <http://example.com/dev1/dm>,这样通过该 URI 的子路径,例如 <http://example.com/dev1/dm/TargetMO>,即可以访问到指定 M2M 设备上的目标 MO 及其属性。因此,在 M2M 平台上创建对应所述 M2M 设备的根 MO 资源,并且为其建立 URI,例如以 <http://domain1/dev1/dm> 为根 URI,然后将所述 M2M 设备上的各具体 MO 节点信息映射为所述根 URI 下的子 MO 资源。另外,文档中的组件(如元素、属性)则构建为相应的子资源,也分配有相应的 URI,即根 URI 的子路径。使得资源访问请求可以细化到针对每一个 MO 信息的节点,细化了访问过程。

[0094] 为了实现利用资源访问方法来实现设备管理的功能,需要将 OMA DM 命令映射为相应的资源访问方法,也就是建立起 MO 信息对应的 DM 命令与所述 MO 数据资源对应的资源访问请求间的映射关系。对于能够实现相同操作的资源访问请求和 DM 命令,可以直接建立起两者间的映射关系,以实现被访问资源响应资源访问请求的结果,与对应的 MO 信息接收与访问请求对应的 DM 控制命令的结果相同的目的。但是由于 DM 命令与资源访问请求并非严格的一一对应,所以对于无法直接映射的 DM 命令,还需要扩展资源请求方法,以提供现有

DM 系统所不支持的设备管理操作和功能。下面以 XCAP 方法为例,对扩展的过程进行描述。

[0095] 为了定义用于访问上述资源的 XCAP 请求方法,根据 XCAP 规范,需要定义如下一些信息:

[0096] 应用唯一标识 (Application Unique ID, AUID):例如 etsi-m2m-dm

[0097] 资源媒体类型 (MIME-type):根据 DM TNSD 规范,可以是 application/vnd.syncml.dmtnds+xml,或 application/vnd.syncml.dmtnds+wbxml;

[0098] 缺省文档命名空间:例如 urn:etsi:m2m:xml:ns:dm

[0099] XML 语法和语义:可以参考 DM TNSD 规范中的语法/语义规定;

[0100] 有效性限制:可参考 DM TNSD 规范以及具体 MO 的规范;

[0101] 访问策略:可参考 DM TNSD 规范及具体 MO 的实施情况;

[0102] 命名习惯:例如可统一规定上述 MO 资源的 URI 根路径为 http://<domain>/<entity>/dm,其中 <domain> 为 M2M 系统的归属域,<entity> 为 XCAP 资源所在 M2M 设备或 M2M 平台的唯一标识,“dm”为固定字符串,这样 M2M 应用能够根据此规则灵活地访问不同位置的用于描述 MO 的 XCAP 资源。

[0103] 现有技术条件下,XCAP 规范只提供了 GET, PUT, DELETE 三种方法,分别表示获取、替换或增加、删除相应的 XCAP 资源的操作,不能完全满足 DM 命令的需求,因此还需要对 XCAP 方法进行一定的扩展。鉴于现有的 XCAP 方法是基于 HTTP 技术的,因此本发明通过引入 HTTP POST 方法来扩展 XCAP 方法,以实现对所有 DM 命令的映射,实施方式见表 1,表 1 为 DM 命令与 XCAP 方法映射表:

[0104] 表 1

[0105]

DM 命令	XCAP/HTTP 方法	说明
<Add>	PUT / POST	增加一个 MO 节点
<Delete>	DELETE	删除一个 MO 节点及其子树
<Get>	GET	获取 MO 节点取值、子树、或子树结构
<Replace>	PUT	更新 MO 节点取值
<Results>	消息体	在响应消息中携带返回内容数据
<Exec>	POST (cmd=exec)	执行一个可执行 MO
<Copy>	POST (cmd=copy) / GET + PUT	复制叶子节点间的数值
<Atomic>	POST (cmd=atomic)	组合多个操作为原子操作
<Sequence>	POST (cmd=sequence)	要求顺序执行的 Atomic 操作
<Alert>	POST (cmd=alert)	异步事件报告

[0106] 具体来说,DM 命令 <Add>, <Delete>, <Get>, <Replace> 都能够直接映射为已有的 XCAP 方法,包括 PUT, DELETE, GET, 其中 <Add> 命令即可以用 PUT 方法实现,也可以采用 POST

方法实现,前者作用于目标 MO 节点 / 属性所对应的 URI,而后者作用于目标 MO 节点 / 属性的父节点所对应的 URI。DM 命令 <Results> 通常用来携带 <Get> 命令的结果内容,可以通过 XCAP 响应消息的消息体来携带实现。

[0107] DM 命令 <Exec> 通常作用于一个可执行 MO 节点,用于指示 DMC 执行某种特定的设备管理操作或任务,本实施例通过 POST 方法并携带 cmd = exec 参数作用于上述 MO 节点对应的 URI (如 http://example.com/dev1/dm/TargetMO) 来实现相同功能的基于资源的设备管理操作方法,例如:

[0108] POST http://example.com/dev1/dm/TargetMO ? cmd = exec

[0109] DM 命令 <Copy> 用于将一个源 MO 节点中的内容复制到一个目标 MO 节点中,本实施例通过 POST 方法并携带 cmd = copy 参数作用于上述目标 MO 节点对应的 URI (如 http://example.com/dev1/dm/TargetMO),并将源 MO 节点的 URI (如 http://example.com/dev1/dm/SourceMO) 携带在消息体中来实现相同功能的基于资源的设备管理操作方法,例如:

[0110] POST http://example.com/dev1/dm/TargetMO ? cmd = copy

[0111] Content-type:...

[0112] Content-length:...

[0113] <SourceURI>http://example.com/dev 1/dm/SourceMO</SourceURI>

[0114] 或者,DM 命令 <Copy> 还可以由 GET 和 PUT 两个方法组合实现,即首先通过 GET 方法获取源 MO 节点的内容,然后通过 PUT 方法将所获取的内容写入目标 MO 节点。

[0115] DM 命令 <Atomic> 与 <Sequence> 都是用来实现上述若干 DM 命令的组合操作,并要求保证原子性,即只要其中一条 DM 命令执行失败,则整个操作失败。不同之处在于,前者对若干 DM 命令的执行顺序没有限制,而后者要求若干 DM 命令必须按照所列的先后顺序执行。本实施例通过 POST 方法并携带 cmd = atomic 或 cmd = sequence 参数作用于目标设备 MO 根节点对应的 URI (如 http://example.com/dev1/dm),并通过在消息体中携带若干 DM 命令的方式,每个命令用一个 <Cmd> 元素描述,从而实现相同功能的基于资源的设备管理操作方法,例如:

[0116] POST http://example.com/dev1/dm ? cmd = atomic

[0117] Content-type:...

[0118] Content-length:...

[0119] <Atomic>

[0120] <Cmd>GET http://example.com/dev 1/dm/TargetMO</Cmd>

[0121] <Cmd>DELETE http://example.com/dev1/dm/TargetMO</Cmd>

[0122] </Atomic>

[0123] 或者,上述实施例中,POST 方法也可以采用相同的参数,如 cmd = trans,然后通过消息体中的 <Atomic> 或 <Sequence> 元素来区分 DM 命令 <Atomic> 与 <Sequence>。

[0124] DM 命令 <Alert> 用于 DMC 向 DMS 上报异步事件信息,本实施例通过 POST 方法并携带 cmd = alert 参数作用于上述 DMS 上专门用于接收异步事件的 URI (如 http://example.com/dms1/dm),并在消息体中描述事件类型或事件码 (如 SERVER-INITIATED MGMT(1200)) 和 / 或相关源 MO 对应的 URI (如 http://example.com/dev1/dm/SourceMO),从而实现相同功能的基于资源的设备管理操作方法,例如:

[0125] POST http://example.com/dms 1/dm ? cmd = alert

[0126] Content-type:...

[0127] Content-length:...

[0128] <Event>SERVER-INITIATED MGMT(1200)</Event>

[0129] <Source>http://example.com/dev1/dm/SourceM0</Source>

[0130] 或者,还可以通过 POST 方法并携带 cmd = alert 参数分别作用于上述 DMS 上用于接收不同类型 / 码的异步事件 URI (如 http://example.com/dms1/dm/1200) 来上报不同的异步事件信息,和 / 或在消息体中描述相关源 M0 对应的 URI (如 http://example.com/dev1/dm/SourceM0),从而实现相同功能的基于资源的设备管理操作方法,例如:

[0131] POST http://example.com/dms1/dm/1200 ? cmd = alert

[0132] Content-type:...

[0133] Content-length:...

[0134] <Source>http://example.com/dev1/dm/SourceM0</Source>

[0135] 基于以上实施例所提供的方法,能够通过基于资源的接口访问方式,实现现有 OMA DM 中的所有针对设备 M0 的设备管理操作和功能。

[0136] 当然,除上述方法外,本发明还公开了如下方法:

[0137] 在 ETSI (European Telecommunications Standards Institute, 欧洲电信标准化协会) 所制定的 M2M 规范中,任何 M2M 设备或 M2M 平台以及它们所提供的业务能力,都可以被抽象为资源并且具有唯一的资源标识,即 URI。同时,多个资源还可以构成一个群组资源,并由共同的群组标识来唯一识别,即群组 URI。因此,当多个 M2M 设备构成一个群组时,本实施例还可以将所述多个 M2M 设备所具有的相同的 M0 资源构建成为该群组的虚拟 M0 资源,然后按照上述任一实施例中的设备管理方法,通过对所述群组 M0 资源的访问操作,实现对该群组中所有 M2M 设备的批量设备管理操作。

[0138] 具体来说,假设 M2M 设备 1、2、3 都具有相同的 M0 节点,且根据实施

[0139] 例 1 中的方法分别构建为如下的 M0 资源:

[0140] http://example.com/dev 1/dm/TargetM0

[0141] http://example.com/dev2/dm/TargetM0

[0142] http://example.com/dev3/dm/TargetM0

[0143] 同时,假设 M2M 设备 1、2、3 共同属于一个群组 group1,其群组 URI 为:

[0144] http://example.com/group 1

[0145] 则,根据本实例中的方法,可以构建该群组的虚拟 M0 资源:

[0146] http://example.com/group1/dm/TargetM0

[0147] 通过对该群组 M0 资源施加相应的 XCAP 方法,就可以实现对该群组中所有 M2M 设备 1、2、3 所具有的相同目标 M0 节点进行批量的获取、替换、删除、添加等设备管理操作和功能。例如:

[0148] GET http://example.com/group1/dm/TargetM0

[0149] 表示请求获取所述群组中所有设备上目标 M0 节点 TargetM0 的信息,则接收该 XCAP 方法的设备管理功能或中间件将分别获取 M2M 设备 1、2、3 上的 TargetM0 信息,并将汇总的结果返回给所述 XCAP 方法的请求者。

[0150] 进一步地,通过扩展上述 XCAP 方法和 / 或消息体中所携带的参数,还可以使得 M2M 应用能够在进行基于资源的设备管理接口访问时,携带所述访问方法的执行条件,当且仅当所述执行条件满足时,才执行相应的设备管理操作。

[0151] 当所述若干执行条件仅用于单独的设备管理操作时,可携带在相应 XCAP 方法的命令行参数中。例如,假设设备 1 用于固件升级的 MO 构建为资源 `http://example.com/dev1/dm/firmware`,同时定义条件参数 `minVer` 表示最小版本号,`man` 表示生产商,`mode` 表示产品型号,则当需要针对设备执行条件固件更新的操作时,应执行如下基于资源的 XCAP 方法:

[0152] PUT

[0153] `http://example.com/dev1/dm/firmware? minVer = 1.0&man = huawei&mode = cdma`

[0154] Content-type:...

[0155] Content-length:...

[0156] {NewF irmwareImage}

[0157] 该方法表示,当且仅当设备 1 的固件版本号不小于“1.0”、且生产商为“huawei”、且型号为“cdma”时,才执行相应的固件升级操作,其中新固件包 {NewFirmwareImage} 由该方法的消息体携带。这样,接收该 XCAP 方法的设备管理功能或中间件将代替 M2M 应用首先获取目标设备的固件版本号、生产商、产品型号的等信息,并与该 XCAP 方法中携带的条件进行比较,若条件均满足时,再执行后续的固件升级操作。从而避免了 M2M 应用需要通过事先多次调用 GET 等方法来获取相关信息并进行本地条件比较,然后才能通过 PUT 方式实施设备固件升级的麻烦。

[0158] 更进一步地,当所述若干执行条件用于多个的设备管理操作时,可以采用类似脚本的方式携带在相应 XCAP 方法的消息体中。例如:

[0159] POST `http://example.com/dev1/dm/firmware? cmd = sequence`

[0160] Content-type:...

[0161] Content-length:...

[0162] <Sequence>

[0163] <Cmd>GET `http://...</Cmd>`

[0164] <Conditions composition = “AND”>

[0165] <minVer>1.0</minVer>

[0166] <man>huawei</man>

[0167] </Conditions>

[0168] <Actions>

[0169] <Cmd>PUT `http://...</Cmd>`

[0170] <Cmd>DELETE `http://...</Cmd>`

[0171] </Actions>

[0172] <Conditions composition = “OR”>...</Condition>

[0173] <Actions>...</Actions>

[0174] </Sequence>

[0175] 其中每一组条件由一个 <Conditions> 元素来描述,其中包含具体的条件子元素,如 <minVer> 表示最小版本号,<man> 表示生产商,<mode> 表示产品型号等,而条件间的组合关系如“或”(OR) 或者“与”(AND) 则可以由 <Conditions> 元素的属性 composition 来描述;相应的设备管理操作则由一个 <Actions> 元素来描述,其中包含若干个 <Cmd> 子元素用于描述具体的基于资源的设备管理方法。此外,还可以有无条件执行的 XCAP 方法,由 <Conditions> 元素之外的 <Cmd> 元素所描述。

[0176] 经过上述改进,资源访问请求中可以包括:被访问 MO 数据资源对应的设备参数,如上述的版本号,生产商及型号等,以实现符合参数的设备的 MO 数据的控制,使得资源访问请求更加细化,对数据的控制更加方便,准确。

[0177] 通过上述可携带条件的基于资源的设备管理方法,可以极大简化 M2M 应用实施设备管理的操作过程。同时,可以理解的是,本实施例只给出了所有可能的执行条件及其组合的一部分示例。任何现有 OMA DM 规范中所定义的 MO 节点及其属性信息,以及与之相关的信息,均可采用本实施例中所给出的方法扩展作为执行条件。

[0178] 本实施例并不限定构建 MO 资源的过程,可以为设备中间件将 DMC 维护的 MO 信息依据基于资源的协议将 MO 信息注册为平台中间件上的 MO 数据资源;也可以为 M2M 设备上的 DMC 将其维护的 MO 信息采用现有技术中 DM 系统中 DMS 与 DMC 之间的协议接口提供给 DMS,并由 DMS 将所述 MO 信息注册为平台中间件上的 MO 数据资源;也可以是设备中间件采用平台中间件和设备中间件之间的资源访问协议,将 DMC 维护的 MO 信息提供给 DMS,并由 DMS 将所述 MO 信息注册为平台中间件上的 MO 数据资源。

[0179] 本实施例中并不限定只能对 XCAP 协议进行扩展,同样可以对其他资源访问协议进行扩展以实现相同的目的。

[0180] 本发明实施例公开了又一建立所述 MO 数据资源与 MO 信息间的映射关系方法,该方法将 MO 数据资源建立在 M2M 设备上,其流程如图 7 所示,包括如下步骤:

[0181] 步骤 S71、将 DMC 上的 MO 信息用可扩展标记语言 XML 文档进行描述;

[0182] 步骤 S72、在 M2M 设备上构建与所述 MO 信息对应的 MO 数据资源,并为所述资源分配 URI,所述 MO 数据资源为 XML 配置访问协议 XCAP 资源。

[0183] 一般来说,为设备上所构建的 MO 数据资源分配 URI 时,可以如本步骤中直接分配 URI,也可以为遵循预定的规则,比如将设备自身的 URI 作为所述 MO 数据资源 URI 的基础部分来分配 URI,平台中间件则能够根据所述 URI 判断所述 MO 数据资源位于设备上,从而能够将对应所述 URI 的资源访问请求转发到相应的目标设备。

[0184] 由于 M2M 应用也有可能根据所述预定的规则请求访问一个与目标设备相关的却并不存在的 MO 数据资源 URI,导致访问失败。此时,为了让 M2M 应用和/或 M2M 平台能够更准确地找到 M2M 设备上所注册的本地 MO 资源,还可以向平台中间件声明设备中间件所注册的本地 MO 数据资源。因此,本实施例中还可以包括:

[0185] 步骤 S73、向所述平台发送指示拥有所述 MO 数据资源的声明。

[0186] 在所述 M2M 平台上声明所述 MO 数据资源。所述声明过程的具体实现方法可以采用 XCAP 资源访问方法或现有技术中任何已知的其它方法。

[0187] 本发明还公开了一种能够实现上述设备管理方法的中间件,其一种结构如图 8 所示,包括:资源访问请求接收单元 81、控制命令转换单元 82 和控制命令发送单元 83,其中:

[0188] 资源访问请求接收单元 81 用于：通过资源访问接口接收外部应用（本实施例以 M2M 应用为例说明）资源访问请求，所述资源访问请求包含用于指示被访问管理对象 MO 数据资源的存储位置的唯一资源标识符 URI。

[0189] 控制命令转换单元 82 用于：参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系，转换所述对 MO 数据资源的资源访问请求为与其对应的 DM 命令，并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据对应的 MO 信息。

[0190] 控制命令发送单元 83 用于发送所述 DM 命令到所述资源访问请求中的 URL 对应的设备（即目标设备），以管理所述与被访问 MO 数据对应的 MO 信息。

[0191] 所述 M2M 设备上的 MO 信息被映射为 MO 数据资源，所述中间件预先建立资源访问请求与 DM 命令间的映射关系，使得外部 M2M 应用可以通过资源访问接口访问控制 MO 信息，从而实现了 M2M 应用通过开放的统一接口管理远程设备的目的。

[0192] 图 9 示出了本发明实施例公开的另一种中间件结构。该中间件包括资源访问请求接收单元 91、控制命令转换单元 92、控制命令发送单元 93、结果数据接收单元 94 和结果数据返回单元 95，其中：

[0193] 资源访问请求接收单元 91、控制命令转换单元 92 和控制命令发送单元 93，与图 8 所示同名单元功能基本相同，在此不再赘述。

[0194] 结果数据接收单元 94 用于：接收目标设备执行所述 DM 命令后生成并返回的结果数据。

[0195] 结果数据返回单元 95 用于向资源访问请求发送者（如 M2M 应用）返回所述结果数据。

[0196] 本实施例所述中间件在实现外部应用（如 M2M 应用）通过开放的统一接口管理远程设备的目的的同时，进一步增加了结果反馈机制，为设备管理提供了方便。

[0197] 上述所述控制命令发送单元 83 或控制命令发送单元 93 的一种结构如图 10 所示，包括：判断单元 1001、会话创建单元 1002 和发送单元 1003，其中：

[0198] 判断单元 1001，用于判断是否存在管理服务 DMS 与所述目标设备间的会话；

[0199] 会话创建单元 1002，用于获取所述判断单元 1001 的判断结果，当判断结果指示不存在所述 DMS 与目标设备的 DMC 间的会话时，控制 DMS 建立与所述目标设备间的会话。

[0200] 发送单元 1003，用于依据所述 DMS 与所述目标设备的 DMC 建立的会话发送 DM 命令到目标设备。

[0201] 本实施例提供的控制命令发送单元能够先判断 DMS 与所述目标设备的 DMC 间的联通情况（即是否建立了会话），并在会话不存在时，进行会话创建，保证 DM 命令能够顺利发送至目标设备。

[0202] 上述所述控制命令发送单元 83 或控制命令发送单元 93 的一种结构如图 11 所示，包括：判断单元 1101、会话创建单元 1102、发送单元 1103 和会话过程判断单元 1104，其中：

[0203] 判断单元 1101、会话创建单元 1102 和发送单元 1103，功能与图 10 中的同名单元基本相同，在此不再赘述。

[0204] 会话过程判断单元 1104，用于在发送单元 1103 发送完一个 DM 命令后，判断预设会话时间内是否存在后续 DM 命令，若存在后续 DM 命令，则指示判断单元 1101 继续工作，若不存在，则发送关闭此次会话的信息，指示 DMS 关闭与目标设备的 DMC 之间的会话。

[0205] 本实施例公开的控制命令发送单元能够避免在不存在后续 DM 命令时维护会话所花费的不必要开销。

[0206] 另外,控制命令发送单元还可以在上述图 10 或图 11 的基础上,进一步增加封装单元,用于按照预设协议封装发送单元 1003 或发送单元 1103 准备发送的 DM 命令。

[0207] 本实施例公开的中间件,可以为独立于 M2M 平台和设备的装置,也可以是集成在 M2M 平台或设备上的装置,同时也可以将其按照不同应用场景,划分其功能,将与平台或者设备连接较为紧密的功能分别设置于平台上的平台中间件和设备上的设备中间件。该中间件采用资源访问接口,如 XCAP 接口,使得外部 M2M 应用可以通过资源访问请求控制 MO 资源,并将对 MO 资源的控制转换为对设备上 MO 信息的控制,实现了 M2M 应用通过开放的统一接口管理远程设备的目的。

[0208] 本实施例并不限定控制命令发送单元的结构,可以根据实际情况而采用其他的结构。

[0209] 本实施例公开的中间件的具体工作过程请参照前述方法实施例,在此不再赘述。

[0210] 另外,本发明还进一步公开了一种 M2M 系统,其一种结构如图 12 所示,包括中间件 1201 和 M2M 设备 1202,其中:

[0211] 中间件 1201 用于执行以下操作:

[0212] 接收外部应用(如 M2M 应用)通过资源访问接口发送的资源访问请求,所述资源访问请求包含:用于指示被访问 MO 数据资源的存储位置的唯一资源标识符 URI;参考预先建立的所述 MO 数据资源访问请求与 DM 命令的映射关系,转换所述对 MO 数据资源的访问请求为与其对应的 DM 命令,并根据预先建立的所述 MO 数据资源与 MO 信息的映射关系确定与所述被访问 MO 数据对应的 MO 信息,发送所述 DM 命令到与资源访问请求中的 URL 对应的 M2M 设备 1202,以管理所述与被访问 MO 数据对应的 MO 信息。

[0213] 所述 M2M 设备 1202 用于,接收所述 DM 命令并执行,获得结果数据,将所述结果数据返回给所述中间件 1201,由所述中间件 1201 返回给发送资源访问请求的外部应用。

[0214] 其中,所述中间件 1201 的结构可以如图 8 或图 9 所示。

[0215] 本发明实施例中的中间件 1201 可以是个独立装置,与 M2M 设备配合完成设备管理操作,还可以设置于 M2M 设备上,作为 M2M 设备的组成部件,与 M2M 设备中的 DMC 配合完成设备管理操作。可以看出,在仅需要完成设备管理的情况下,本发明实施例提供的 M2M 系统可以没有 M2M 平台。

[0216] 此外,针对本发明实施例中的中间件的不同,还存在其他各种结构形式,图 13 示出了另外一种结构的 M2M 系统,该系统包括:M2M 平台 1301 和 M2M 设备 1302,M2M 平台 1301 具有 DMS1303 和平台中间件 1304,M2M 设备 1302 具有 DMC1305 和设备中间件 1306,其中:

[0217] 所述设备中间件与 M2M 设备中的设备管理客户端 DMC 可以为分体设计,所述设备中间件通过所述 DMC 开放的本地访问接口与其连接;或者,所述设备中间件与所述 DMC 为一体化设计;再或者所述设备中间件与所述 DMS 为一体化设计,采用 OMA-DM 协议的访问接口与所述 DMC 相连接。

[0218] 所述平台中间件可以与 M2M 平台中的 DMS 为一体化设计,或者所述平台中间件与所述 M2M 平台中或 M2M 平台外部的 DMS 采用所述 DMS 开放的本地访问接口相连。本实施例并不限定平台中间件和设备中间件与系统中其他部件的连接关系,具体的连接形式可以根

据实际情况而任意设定。

[0219] 本实施例中,平台中间件 1304 和设备中间件 1306 配合完成图 12 所示系统中中间件的功能。本系统的工作过程为:

[0220] 设备中间件 1306 预先将 M2M 设备上的 MO 信息以 XML 文档形式描述后发送到平台中间件,并构建为平台中间件上的 MO 数据资源,平台中间件 1304 接收 M2M 应用发送的资源访问请求,根据资源访问请求中的 URI,查找到对应的目标设备,将资源访问请求依据预先建立的 MO 数据资源访问请求与 DM 命令的映射关系,转换为 DM 命令,并根据预先建立的 MO 资源数据与 MO 信息间的映射关系确定与被访问的 MO 资源数据对应的 MO 信息,控制 DMS1303 按照 OMA-DM 接口协议建立 DMS1303 与 DMC1305 之间的 DM 会话,并在所述会话中将 DM 命令发送给 DMC1305 执行,然后将执行后的结果数据返回给平台中间件 1304,并由平台中间件 1304 返回给 M2M 应用。

[0221] 需要说明的是,在其他的实施例中,DMS 和 DMC 彼此之间不直接连接,两者间的会话由连接 DMS 的平台中间件和连接 DMC 的设备中间件进行通信来实现。

[0222] 上述各个实施例中的 M2M 设备可以为传感器、微控制器、移动或固定终端、网关等各种支持远程设备管理能力的设备和装置,而 M2M 应用可以为电力抄表、智能交通等各种机器通信相关的个人或行业应用。

[0223] 容易理解的是,本发明所提供的方案实际上还可以部署在其它需要通过资源访问接口协议实施设备管理的通信系统中,而不局限于 M2M 应用领域。

[0224] 需要说明的是,上述系统实施例中具有前文所述中间件的 M2M 平台和设备都属于本发明的保护范畴。

[0225] 本说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0226] 专业人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0227] 结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器 (RAM)、内存、只读存储器 (ROM)、电可编程 ROM、电可擦除可编程 ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0228] 对所公开的实施例的上述说明,使本领域专业技术人员能够实现或使用本发明。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本发明的精神或范围的情况下,在其它实施例中实现。因此,本发明将不会被限制于本文所示的这些实施例,而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。

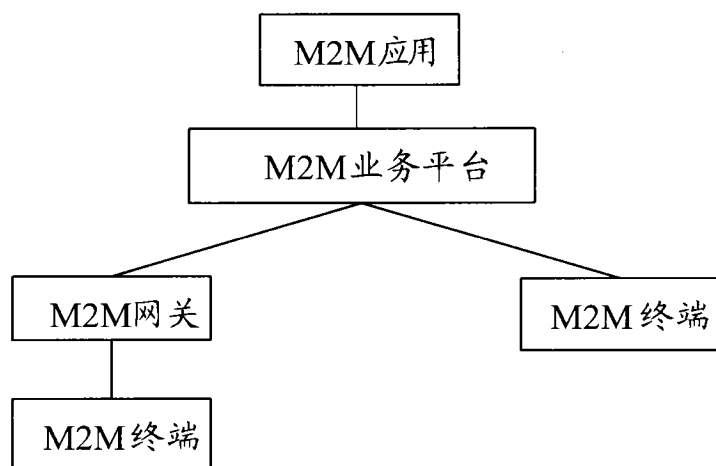


图 1

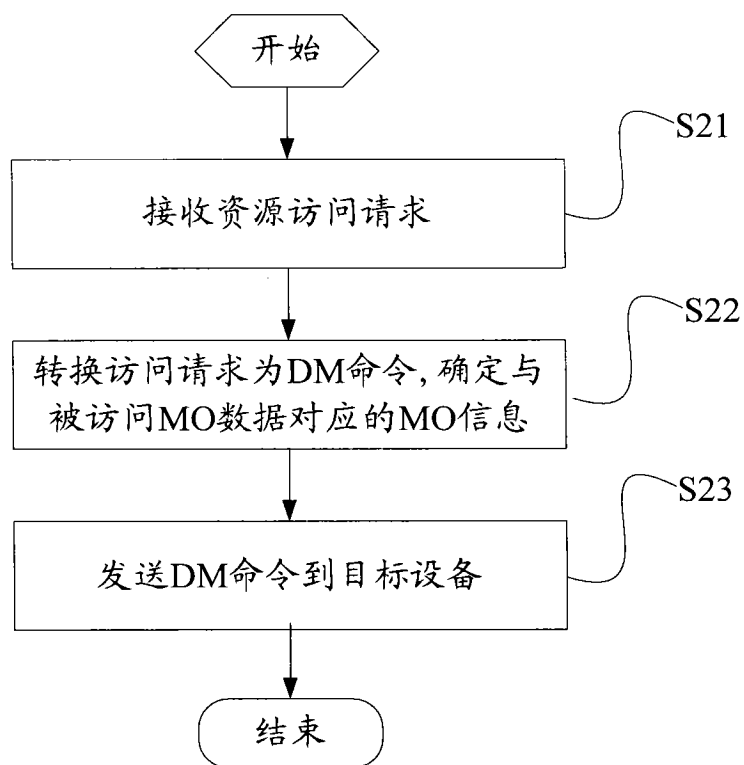


图 2

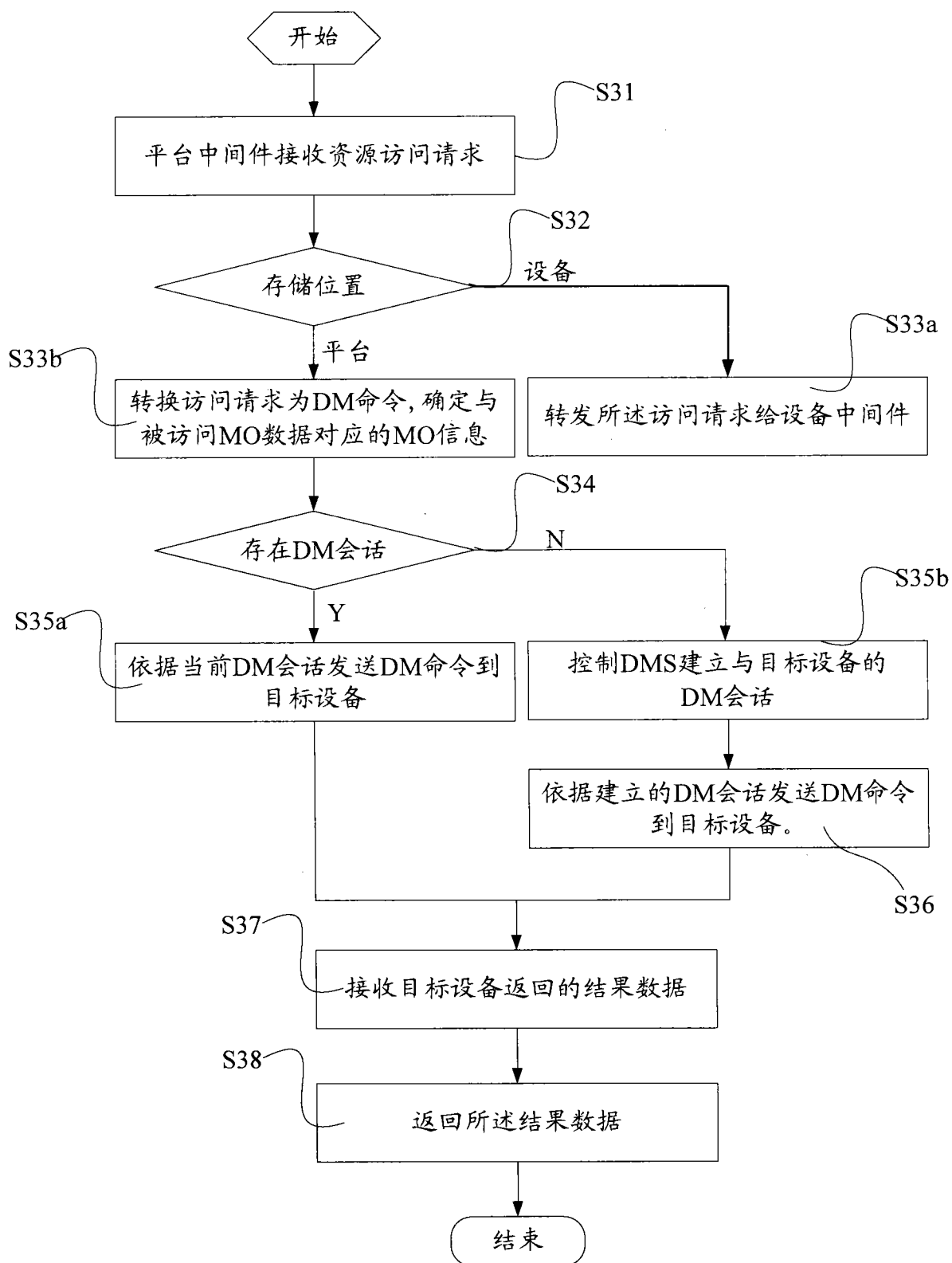


图 3

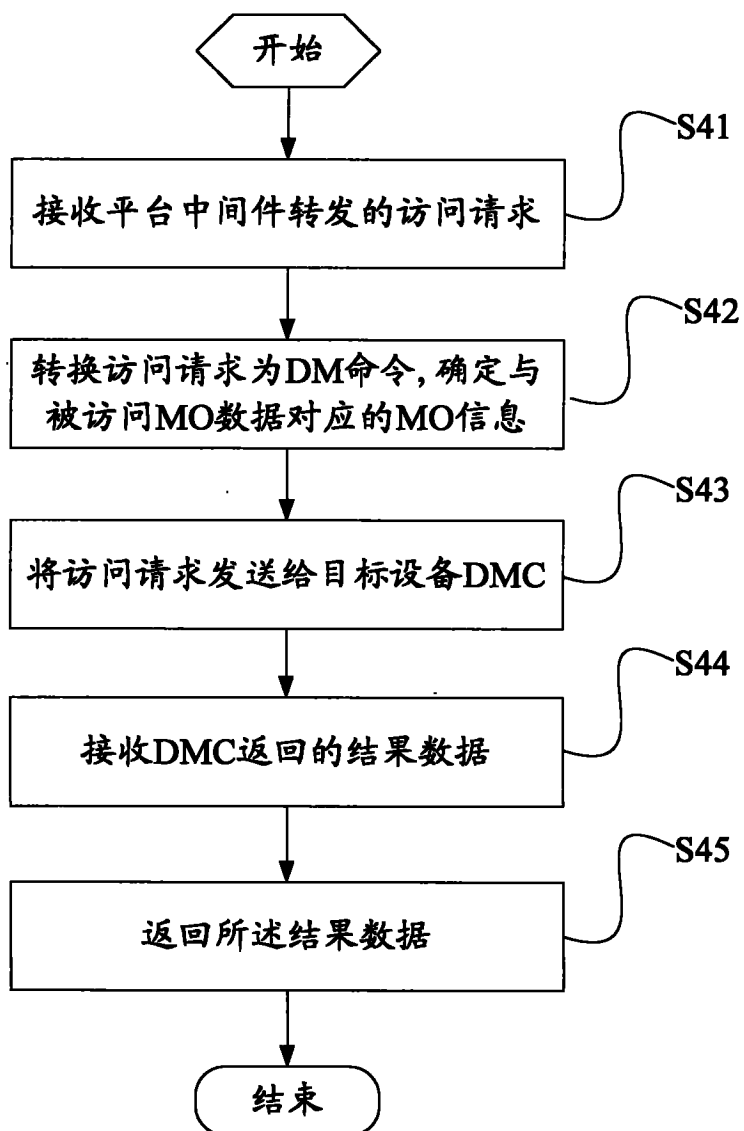


图 4

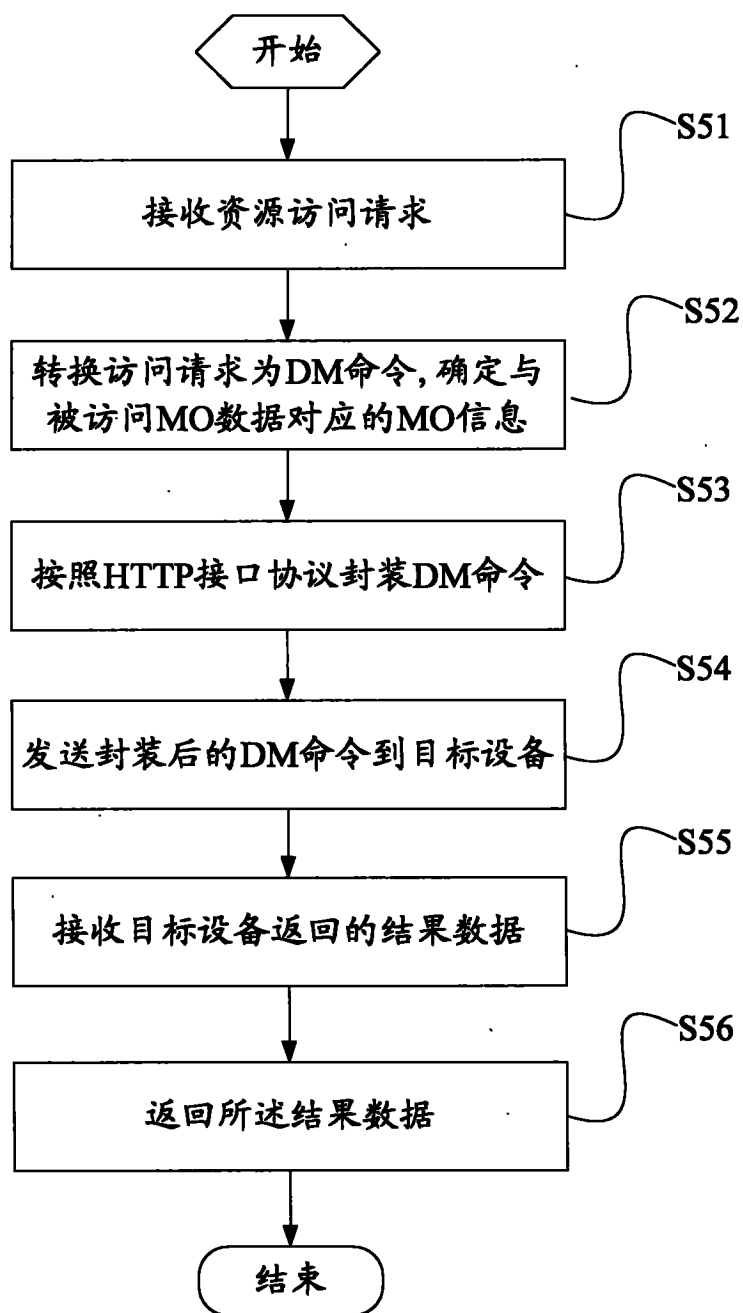


图 5

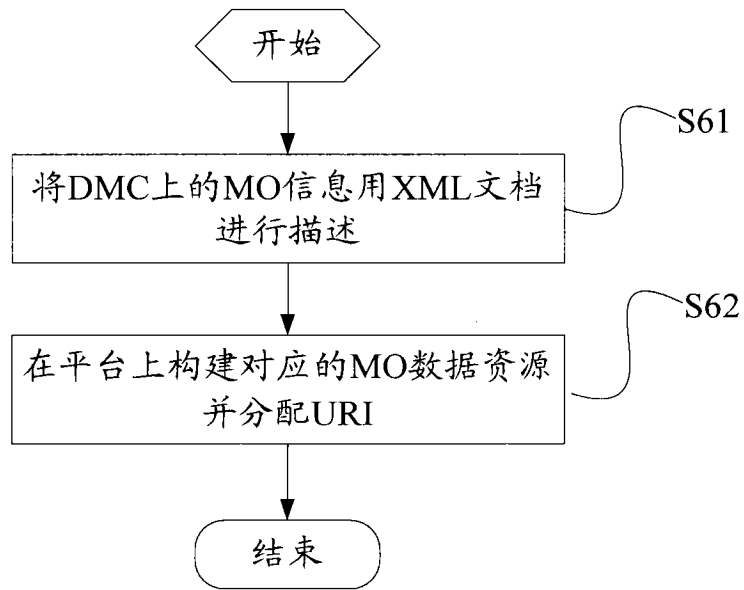


图 6

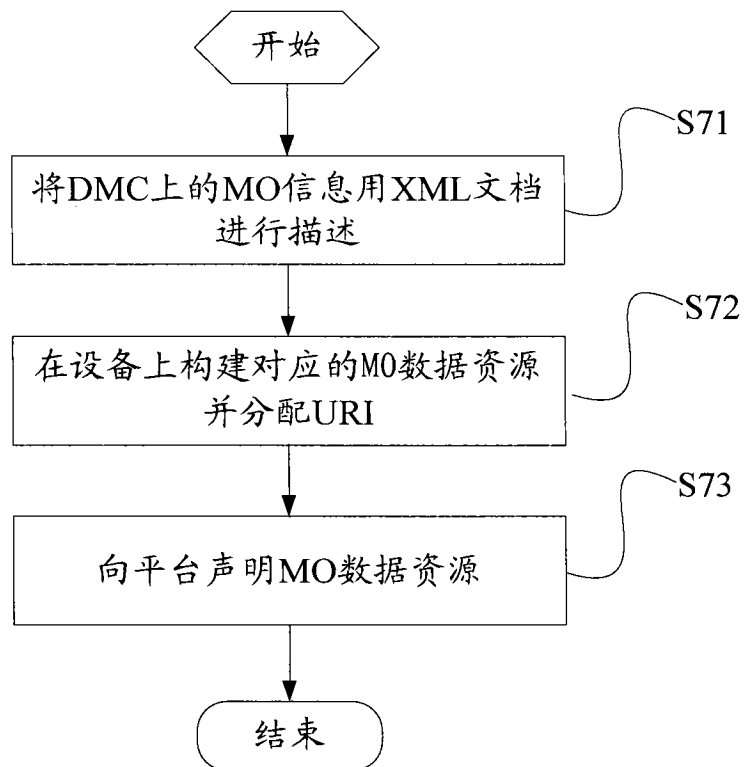


图 7

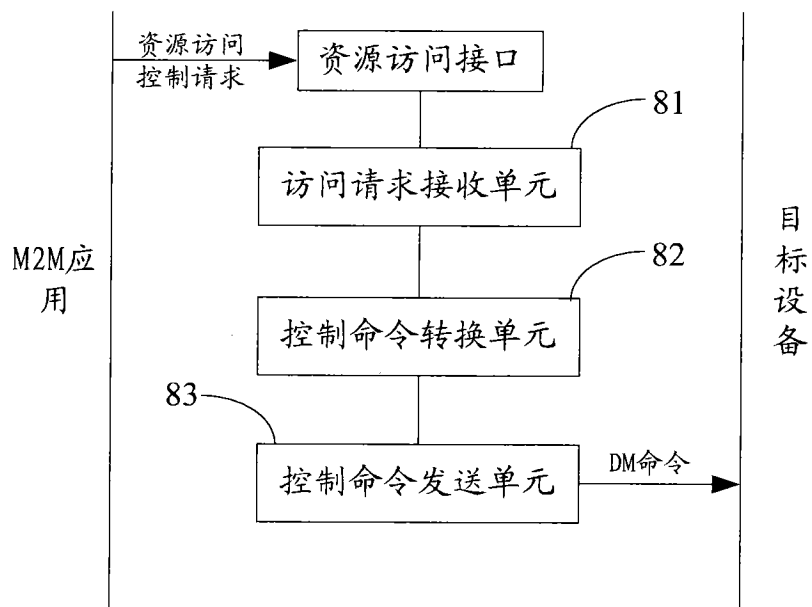


图 8

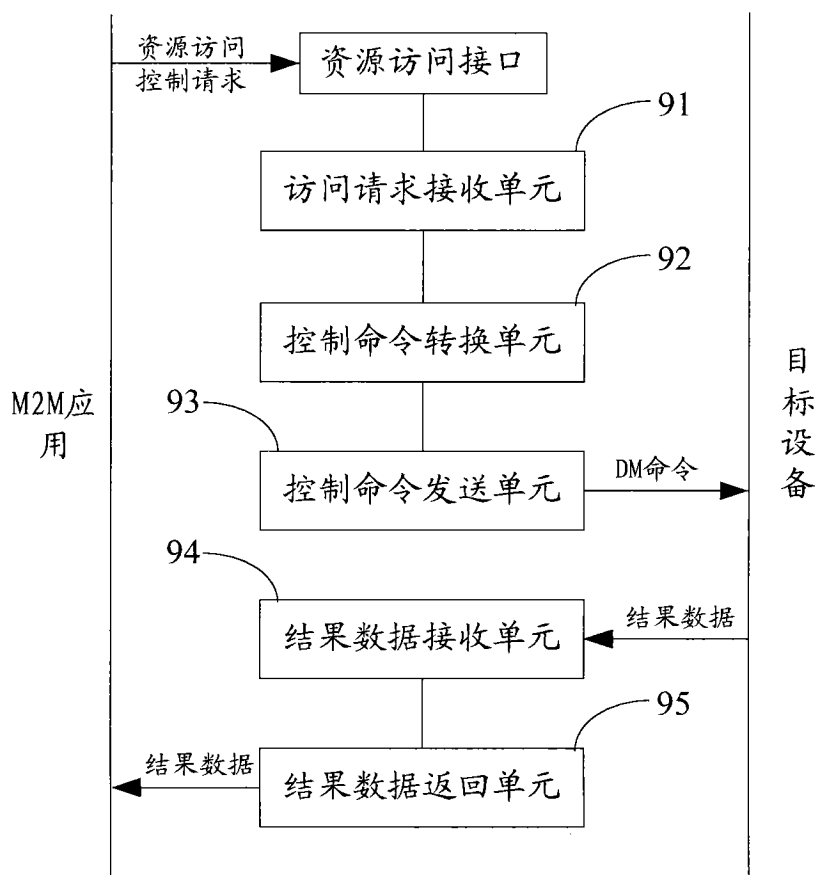


图 9

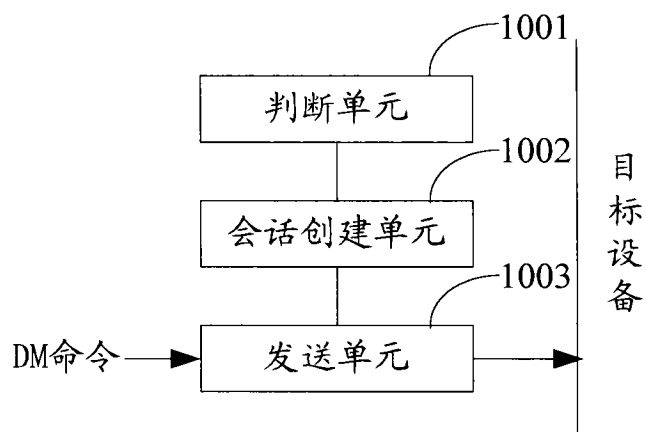


图 10

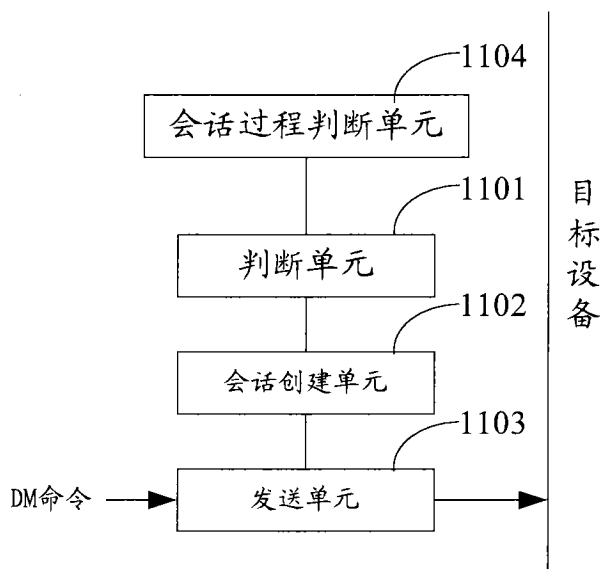


图 11

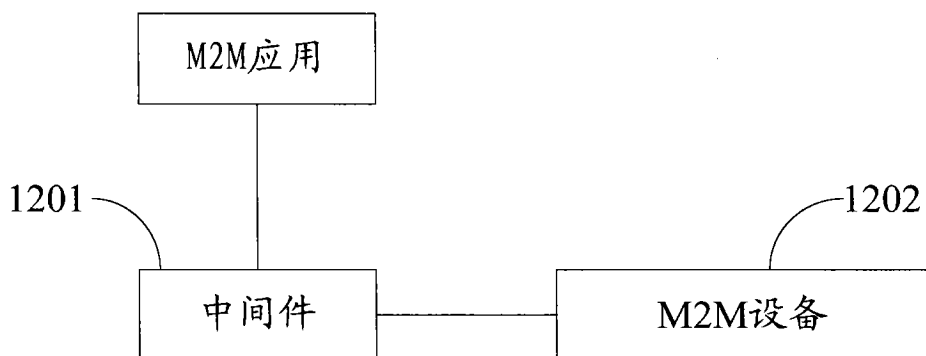


图 12

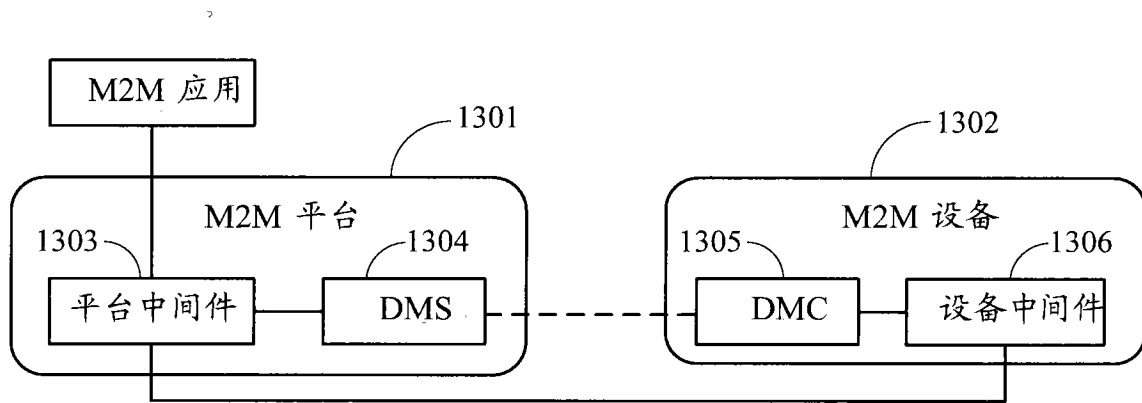


图 13