

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局



(43) 国际公布日
2009 年 6 月 25 日 (25.06.2009)

PCT

(10) 国际公布号
WO 2009/076910 A1

(51) 国际专利分类号:
H04L 12/54 (2006.01)

518129 (CN)。李高峰(LI, Gaofeng) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN)。

(21) 国际申请号: PCT/CN2008/073482

(22) 国际申请日: 2008 年 12 月 12 日 (12.12.2008)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
200710032635.X
2007 年 12 月 14 日 (14.12.2007) CN

(71) 申请人 (对除美国外的所有指定国): 华为技术有限公司(HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN)。

(72) 发明人; 及

(75) 发明人/申请人 (仅对美国): 向海(XIANG, Hai) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN)。伊志权(YI, Zhiquan) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN)。卢剑锋(LU, Jianfeng) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN)。丁宁(DING, Ning) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN)。刘斌(LIU, Bin) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong

(74) 代理人: 广州三环专利代理有限公司(GUANGZHOU SCIHEAD PATENT AGENT CO., LTD); 中国广东省广州市越秀区先烈中路 80 号汇华商贸大厦 1508 室, Guangdong 510070 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 欧洲 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告。

(54) Title: DOCUMENT MANAGEMENT METHOD, SYSTEM AND CORRELATED DEVICE

(54) 发明名称: 一种文档管理方法、系统及相关设备

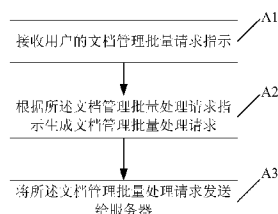


图 1 /Fig. 1

A1 RECEIVING THE INDICATION OF THE DOCUMENT MANAGEMENT BATCH REQUEST FROM THE USER
A2 GENERATING THE DOCUMENT MANAGEMENT BATCH PROCESSING REQUEST ACCORDING TO THE INDICATION OF THE DOCUMENT MANAGEMENT BATCH REQUEST
A3 SENDING THE DOCUMENT MANAGEMENT BATCH PROCESSING REQUEST TO THE SERVER

(57) Abstract: An extensible markup language management request method comprises that: a client sends an operation instruction for batch processing of a plurality of documents or a plurality of nodes of one document to a server via a document management batch processing request, when the document management batch processing request is determined, the server obtains the corresponding document management batch processing information and carries out the corresponding document management batch processing operation. An extensible markup language document management method, an extensible markup language document management client, an extensible markup language document management server and an extensible markup language document management system are also disclosed. The client of the proposal of the invention needs only to send one signaling to carry out the batch processing operation of a plurality of documents or a plurality of nodes of a document, thereby reducing the time delay for operation, improving operation efficiency and the reliability of data operation.

[见续页]



(57) 摘要:

本发明公开了一种可扩展标记语言管理请求方法，客户端将需要对多个文档或单个文档下的多个节点的批量处理操作指示，通过一条文档管理批量处理请求发送给服务端，服务端在确定为文档管理批量处理请求时，获取相应的文档管理批量处理信息并执行相应的文档管理批量处理操作。本发明还提供了相应的可扩展标记语言文档管理方法、可扩展标记语言文档管理客户端、可扩展标记语言文档管理服务器以及可扩展标记语言文档管理系统。由于本发明方案的客户端只需要发送一条信令来进行多个文档或单个文档下的多个节点的批量处理操作，减小了操作时延，提高了操作效率，提高了数据操作的可靠性。

一种文档管理方法、系统以及相关设备

本申请要求了 2007 年 12 月 14 日提交的，申请号为 200710032635.X 发明名称为“一种文档管理方法、系统以及相关设备”的中国申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本发明涉及信息管理技术领域，具体涉及可扩展标记语言（XML, eXtensible Markup Language）文档管理请求方法和相应的 XML 文档管理方法、XML 文档管理系统以及 XML 文档管理客户端和文档管理服务器。

背景技术

XML 是一套定义语义标记的规则，所定义的标记将文档分成许多部件并对这些部件加以标识，符合 XML 规则的文档即为 XML 文档。XML 文档具有易于阅读和编写的优点，是网络应用中描述数据和交换数据的理想格式。

XML 文档管理客户端（XDMC: XML Document Management Client）可使用文档管理请求，在 OMA(Open MOBILE Alliance, 开放移动联盟)标准中的 XML 配置访问协议（XCAP: XML Configuration Access Protocol）中定义的“GET”、“PUT”、“DELETE”等方法，实现对 XML 文档管理服务器（XDMS: XML Document Management Server）上以 XML 格式存储的应用配置数据进行查询、增加和修改、删除等操作。

在实际应用中，用户可通过上述三种请求操作实现对 XDMS 的单个文档或单个文档中的单个节点进行相应的查询、增加和修改、删除等操作。而当用户需要进行对 XDMS 的多个文档或单个文档中的多个节点进行相应的查询、增加和修改、删除等操作时，需在 XDMC 与 XDMS 之间多次反复调用相应请求操作完成对多个文档或单个文档下的多个节点的操作。

发明内容

本发明实施例提供了一种可扩展标记语言文档管理请求方法包括：接收用

户的文档管理批量处理请求指示；根据所述文档管理批量处理请求指示生成文档管理批量处理请求；将所述文档管理批量处理请求发送给服务器。

以及相应的可扩展标记语言文档管理方法，包括：接收文档管理请求；确定所接收到的文档管理请求为文档管理批量处理请求；从所述文档管理批量处理请求中获取相应的文档管理批量处理信息；根据所述文档管理批量处理信息，执行相应的文档管理批量处理操作。

本发明实施例还提供了一种可扩展标记语言文档管理客户端，包括：用户操作单元，用于接收用户的批量处理请求指示；请求生成单元，用于根据所述文档管理批量处理请求指示生成文档管理批量处理请求；请求发送单元，用于将所述文档管理批量处理请求发送给服务器。

和一种可扩展标记语言文档管理服务器，包括：请求接收单元，用于接收文档管理请求；请求解析单元，用于确定所述文档管理请求为文档管理批量处理请求，并从所述文档管理批量处理请求中获取相应的文档管理批量处理信息；请求处理单元，用于根据所述文档管理批量处理信息，执行相应的文档管理批量处理操作。

以及一种可扩展标记语言文档管理系统，包括客户端和服务端，所述客户端包括：用户操作单元，用于接收用户的批量处理请求指示；请求生成单元，用于根据所述文档管理批量处理请求指示生成文档管理批量处理请求；请求发送单元，用于将所述文档管理批量处理请求发送给服务器；所述服务器包括：请求接收单元，用于接收文档管理请求；请求解析单元，用于确定所述文档管理请求为文档管理批量处理请求，并从所述文档管理批量处理请求中获取相应的文档管理批量处理信息；请求处理单元，用于根据所述文档管理批量处理信息，执行相应的文档管理批量处理操作。

上述技术方案中采用客户端将需要对多个文档或单个文档下的多个节点的批量处理操作指示，通过一条文档管理批量处理请求发送给服务端，服务端在确定为文档管理批量处理请求时，获取相应的文档管理批量处理信息并执行相应的文档管理批量处理操作；由于客户端只需要发送一条信令来进行多个文档或单个文档下的多个节点的批量处理操作，减小了操作时延，提高了操作效率，提高了数据操作的可靠性。

附图说明

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。

图 1 是本发明实施例的 XML 文档管理请求方法流程示意图；

图 2 是本发明实施例的 XML 文档管理方法流程示意图；

图 3 是本发明实施例的 XML 文档管理批量增加/修改（PUTS）处理方法流程示意图；

图 4 是本发明实施例的 XML 文档管理批量查询（GETS）处理方法流程示意图；

图 5 是本发明实施例的 XML 文档管理批量删除（DELETES）处理方法流程示意图；

图 6 是本发明实施例的 XML 文档管理批量移动（MOVE）处理方法流程示意图；

图 7 是本发明实施例的 XML 文档管理系统的组成示意图；

图 8 是本发明实施例的 XML 文档管理客户端的结构示意图；

图 9 是本发明实施例的 XML 文档管理服务器的结构示意图。

具体实施方式

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作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。

本发明实施例提供了一种文档管理请求方法，客户端向服务器发送对多个文档或单个文档下的多个节点进行处理操作的文档管理批量处理请求，服务器在确定为文档管理批量处理请求时，进行相应文档管理批量处理操作。本发明实施例还提供了相应的 XML 文档管理方法、XML 文档管理系统、XML 文档管理客户端和 XML 文档管理服务器。下面分别进行详细描述。

参考图 1，是本发明实施例的 XML 文档管理请求方法流程示意图，包括步骤：

A1、接收用户的文档管理批量处理请求指示。

此处，所述文档管理批量处理请求为文档管理批量增加/修改请求指示，或为文档管理批量查询请求指示，或为文档管理批量删除请求指示，或为文档管理批量移动请求指示。对应于现有的 XML 配置访问协议中定义的 PUT（增加/修改）方法，所述文档管理批量增加/修改请求指示是指用户需要对多个文档或单个文档下的多个节点同时进行增加或修改的操作需求，XDMC 可通过提供给用户的操作界面获取用户的批量增加/修改请求指示。对应于现有的 XML 配置访问协议中定义的 GET（查询）方法，所述文档管理批量查询请求指示是指用户需要对多个文档或单个文档下的多个节点同时进行查询的操作需求，XDMC 可通过提供给用户的操作界面获取用户的批量查询请求指示。对应于现有的 XML 配置访问协议中定义的 DELETE（删除）方法，所述文档管理批量删除请求指示是指用户需要对多个文档或单个文档下的多个节点同时进行删除操作需求，XDMC 可通过提供给用户的操作界面获取用户的批量增加/修改请求指示。在现有的 XML 文档管理方法中，实现单个节点的移动是通过 PUT 和 DELETE 两条信令消息实现的，本发明实施例所述的文档管理批量移动请求指示是指用户需要对多个文档或单个文档下的多个节点同时进行移动的操作需求，XDMC 可通过提供给用户的操作界面获取用户的批量移动请求指示。

A2、根据所述文档管理批量处理请求指示生成文档管理批量处理请求。

此处，具体实现时，考虑到与现有的 XML 配置访问协议中定义 ‘PUT’、‘GET’ 以及 ‘DELETE’ 信令方法的结构相兼容，以及符合协议标准的定义习惯，本发明实施例所述的文档管理批量处理请求主要由以下几部分组成：

1、请求行

请求行包括：Method（方法名）、Request-URI、以及 HTTP-Version（HTTP 协议版本号）。

1）Method（方法名）描述与现有技术类似，本发明实施例中定义文档管理批量增加/修改处理请求的方法名为 PUTS，文档管理批量查询处理请求的方法名为 GETS，文档管理批量删除处理请求的方法名为 DELETES，文档管理批量移动处理请求的方法名为 MOVE。显然以上方法名的定义主要是为了区别与现

有的方法名以及具备一定的可读性，本领域技术人员应该明白，选用其他字符也是可以达到相同的目的的。

2) 在现有技术中，标准 XCAP 信令 Request-URI 格式为：XCAP Root + AUID + “ users ” + XUI + filename 。 例如：
http://xcap.example.com/org.openmobilealliance.user-profile/users/sip:zhangshan@huawei.com/index.xml，其中：XCAP Root 为 http://xcap.example.com/，AUID 是 org.openmobilealliance.user-profile，“users”是常量，表示接下来是用户信息（即 XUI），XUI 是 sip:user@huawei.com（sip:表示 XUI 是 tel uri，huawei.com 是 XDMS 归属域名），文件名为 index.xml。

为支持批量处理，本发明要求用一个批处理标记“batch”替代文件名（也可以用其它字符串标记）。如果需要同时操作多个业务，Request-URI 中的 AUID 换成一个不表示任何具体业务的特定字符串，如“00”，把具体要操作的业务放到后面以“xs:schema”开始的消息体中。若当前只操作一种业务，则可以在 Request-URI 中存放真实的业务标识。

基于上述机制，XCAP 批处理请求行的 Request-URI 可以表示为：

http://xcap.example.com/00/users/sip:zhangshan@huawei.com/batch 。

2、请求消息头

名称	类型	说明
Content-Type	media-type	指定消息体的 MIME 类型。
Content-Length	1*DIGIT	消息体的长度， 参考[RFC2616]section 14.13。

每种 XCAP 批量处理请求的消息体格式不同，Content-Type 也就不同。本发明约定几种批处理请求的 Content-Type 定义如下：

GETS: application/gets+xml（对应文档管理批量查询）；

PUTS: application/puts+xml（对应文档管理批量增加/修改）；

DELETES: application/deletes+xml（对应文档管理删除查询）；

MOVE: application/move+xml（对应文档管理批量移动）；

以上 Content-Type 也可以定义成其它值，但需区别于 OMA 已经定义的类型。

3、请求消息体

每种批量处理请求信令的请求消息体格式是不同的，后文对此将有相应描述。

批处理请求消息的命名空间（NameSpace 和 targetNamespace）在消息体起始部分描述。例如：

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="com:huawei:puts" targetNamespace="com:huawei:puts"
elementFormDefault="qualified">，将 PUTS 请求消息体的命名空间（NameSpace）
为"com:huawei:puts"。
```

A3、将所述文档管理批量处理请求发送给服务器。

此处，XDMC 可参考发送现有的其他类型的文档管理请求的方式将所述文档管理批量处理请求发送给 XDMS。

通常 XDMS 会收到 XDMS 对文档管理批量处理请求的处理结果应答消息，在后续实施描述中，将会说明 XDMS 对文档管理批量处理请求的处理结果应答消息对应于不同类型的文档管理批量处理请求文档管理批量处理消息的消息格式是不同的，后文将在具体类型的文档管理批量处理的实施例描述中对 XDMS 对文档管理批量处理消息的接收以及处理过程进行相应描述。

参考图 2，是本发明实施例的 XML 文档管理方法流程示意图，包括步骤：

B1、接收来自客户端的文档管理请求。

此步骤可参考现有文档管理请求的接收方法进行。

B2、确定所接收到的文档管理请求为文档管理批量处理请求。

此处，XDMS 可根据与 XDMC 共同遵守的约定，通过解析收到的文档管理请求来确定其类型。例如文档管理批量处理请求采用新定义的方法名来标识，对应各种类型的文档管理批量处理请求定义由相应的方法名来标识。

B3、从所述文档管理批量处理信息中获取相应的文档管理批量处理信息。

此处，在确定收到的文档管理请求的类型为文档管理批量请求后，XDMS 首先解析所述文档管理请求的消息头，通过识别 Content-Type 类型（每种类型的文档管理批量处理请求对应唯一的 Content-Type 类型标识），以及根据 XDMS 与 XDMC 约定的对应该 Content-Type 类型的消息体格式解码所述文档管理批量处理请求的消息体，获取相应的文档管理批量处理信息。

本发明实施例中不同类型的文档管理批量处理请求的消息体格式是不同

的, 对应具体类型的文档管理批量处理请求, XDMC 基于与 XDMS 约定的消息体格式编码生成相应的文档管理批量处理请求, 同时, XDMC 通过消息头中的 Content-Type 类型标识消息体格式类型, XDMS 接收到文档管理批量处理请求后, 解析其消息头中的 Content-Type 类型标识, 获取标识消息体格式类型, 进而根据对应该消息格式类型的消息体格式解码所述文档管理批量处理请求的消息体。

B4、根据所述文档管理批量处理信息, 执行相应的文档管理批量处理操作。

B5、向客户端返回文档管理批量处理应答消息。

此处, 基于文档管理批量处理操作的执行结果信息, 按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息, 并向文档管理客户端发送所述文档管理批量处理应答消息。

本发明实施例所述的文档管理批量处理应答消息主要由以下几部分组成:

1、应答状态行

名称	类型	说明	
HTTP-Version		HTTP 版本, 取值: HTTP/1.1。	
Status-Code	3DIGIT	返回状态码, 批处理请求执行成功返回 200, 失败则返回 500。	
Reason-Phrase	*<TEXT, excluding CR, LF>	对状态码的简短描述。 成功返回 OK 未成功返回 NOK	

2、应答消息头

名称	类型	说明	备注
Etag	entity-tag	用于比较 同一资源的不 同实体	参 考 [RFC2616]section 14.19, 14.24, 14.26, 14.44 [XCAP]section 7.10, 8.5
Content-Type	media-type	指定消息 体的 MIME 类	

		型	
Content-Length	1*DIGIT	消息体的 长度	参 考 [RFC2616]section 14.13

本发明实施例描述的四种批处理信令的应答消息体 Content-Type 定义如下:

GETS: application/gets-result+xml

PUTS: application/puts-result+xml

DELETES: NA

MOVE: NA

以上 MIME 类型也可以约定其它标记。

Content-Length 用 “(…)” 描述。

3、应答消息体

根据请求返回相应的内容，每种批处理信令的应答消息体格式在后文将有相应描述。

上述方法实施例中采用客户端将需要对多个文档或单个文档下的多个节点的批量处理操作指示，通过一条文档管理批量处理请求发送给服务端，服务端在确定为文档管理批量处理请求时，获取相应的文档管理批量处理信息并执行相应的文档管理批量处理操作；由于客户端只需要发送一条信令来进行多个文档或单个文档下的多个节点的批量处理操作，减小了操作时延，提高了操作效率，提高了数据操作的可靠性。

参考图 3，是本发明实施例的 XML 文档管理批量增加/修改（PUTS）处理方法流程示意图，下面对其进行详细描述。

步骤 s401，XDMC 向 XDMS 发送文档管理批量增加/修改处理请求。

所述文档管理批量增加/修改处理请求信令格式如下：

1、请求行

名称	类型	说明
Method	token	XCAP 请求方法，取值 PUTS
Request-URI	HTTP URI	指定请求操作的 XCAP 资源。
HTTP-Version		HTTP 版本，取值 HTTP/1.1

PUTS 信令请求行如下：

PUTS http://xcap.example.com/00/users/sip:XXX@huawei.com/batch
HTTP/1.1

其中，“batch”表示批量增加或修改文档，“00”是 auid 替代符，表示操作的具体业务在消息中定义,XXX 表示发起请求的用户号码，“sip:”表示这是一个 sip uri,“@huawei.com”为用户 XXX 归属的 XDMS 域名。

2、请求消息头

PUTS 信令请求消息头如下:

Content-Type: application/puts+xml;

Content-Length: (...)

application/puts+xml 是 PUTS 请求的消息体格式。可以是其它标记，但须有别于 OMA 其它定义。

3、请求消息体

PUTS 请求消息体格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema                      xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="com:huawei:puts"                      targetNamespace="com:huawei:puts"
elementFormDefault="qualified">
  <xs:element name="list">
    <xs:complexType>
      <xs:sequence>
        <xs:element                      name="auid"                      type="auidType"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="auidType">
    <xs:sequence>
      <xs:element name="file" type="fileType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

```

</xs:complexType>
<xs:complexType name="fileType">
  <xs:sequence>
    <xs:element name="node" type="nodetype" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="ctype" type="xs:string" use="required"/>
  <xs:attribute name="etag" type="xs:string" use="optional"/>
</xs:complexType>
<xs:complexType name="nodetype">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="uri" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>

```

说明:

PUTS 请求消息体是针对 PUTS 请求消息的批量处理特性定义的;

"com.huawei:puts"为 PUTS 请求消息体的命名空间 (namespace), 可以是其它约定值;

PUTS 请求消息体分 list、auid、file、node 四级, 可以是其它约定属性名: list 为根节点, auid 表示要处理的业务, file 表示要处理的文档, node 表示要处理的节点;

根据 schema 定义, 可以对多个业务 (用 AUID 表示) 分别执行新增或修改多个文档的操作, 也可以对同一篇文档执行增加或修改多个节点的操作;

对于同一请求的增加或修改操作信息, 可以也在一个 list 标签中表达;

auidType: name 为 auid 的名字;

fileType: name 为文档名, ctype 为文档的 cType 类型, etag 为文档版本号。

对 file 中 ctype 属性的特别约定: ctype="application/xcap-el+xml"表示修改节点元素, ctype="application/xcap-att+xml"表示修改节点属性, 也可以约定其它值。

nodetype: uri 属性表示要插入或替换的结点在 xml 文档中的路径, uri 值采用 Xpath 语法。

PUTS 批量处理请求消息的一个示例如下:

PUTS 请求消息

PUTS http://xcap.example.com/00/users/sip:13500012345@huawei.com/batch
HTTP/1.1

...

Content-Type: "application/puts+xml";

Content-Length: (...)

<list xmlns="com:huawei:puts">

 <aud name="ims-pim" >

 <file name="index.xml" ctype="application/ims-pim+xml">

 <ims-pim xmlns="com:huawei:ims-pim">

 <display-name>张三</display-name>

 <tel>58012700</tel>

 </ims-pim>

 </file>

 </aud>

 <aud name="resource-lists">

 <file name="friend.xml" ctype="application/xcap-el+xml"
etags="abbbbcc">

 <node uri="/list/entry[@id=%22m_b6ZCx1%22]">

 <entry id="m_b6ZCx1" xmlns="com:huawei:resource-lists">

 <display-name>xiaoxiong</display-name>

 </entry>

 </node>

```

<node uri="/list/entry[@uri=%2213509090999%22]">
    <entry uri="13509090999"
xmlns="com:huawei:resource-lists">
        <display-name> xiaomao</display-name>
    </entry>
</node>
<node uri="/list/entry[@id=%22m_b6ZA8j%22]">
    <entry id="m_b6ZA8j" uri="13509090999"
xmlns="com:huawei:resource-lists">
        <display-name> xiaoqiang</display-name>
    </entry>
</node>

</file>

</auid>

```

</list>

说明:

此例表示用户 13500012345 请求修改 ims-pim (个人信息的 auid) 的文档 index.xml 的同时, 修改 resource-lists (分组列表的 auid) 的文档 friend.xml;

ims-pim 的 ctype 为 "application/ims-pim+xml", 这有别于消息体的 "application/puts+xml";

ims-pim 的 namespace 为 "com:huawei:ims-pim", 也有别于消息体的 "com:huawei:puts";

对 index.xml 的修改, 未指定版本号 etag, 说明缺省有权限;

friend.xml 的 ctype 为 "application/xcap-el+xml", 表示修改节点;

friend.xml 的 etag 为 "abbbbcc", 表示服务端要做版本控制, 版本号不同则不允许修改节点;

节点路径在通过 node 标签中的 uri 属性描述;

node 下的每对 entry 描述的是 resource-lists 中的 entry 节点。

步骤 s402, 对 PUTS 批量处理请求按照 PUTS 请求消息格式进行解码, 获取 PUTS 批量处理信息。

XDMS 解析 PUTS 批量处理请求的消息头, 获取 Content-Type 类型为 application/puts+xml, 进而根据对应的 PUTS 消息体格式解码 PUTS 批量处理请求消息体, 获取 PUTS 批量处理信息。

步骤 s403, 根据 PUTS 批量处理信息, 执行 PUTS 批量处理操作, 完成相应的文档信息批量增加或修改。

步骤 s404, 对 PUTS 批量处理的结果按照 PUTS 应答消息格式进行编码。

PUTS 应答消息组成如下:

1、应答状态行

格式: HTTP-Version + Status-Code + Reason-Phrase

HTTP 版本, HTTP-Version: HTTP/1.1

状态码, Status-Code: 200 或 500

Reason-Phrase: Status-Code 取 200 时为 OK, 取 500 时为 NOK。

2、应答消息头

Content-Type: application/puts-result+xml

Content-Length: (...)

application/puts-result+xml 是 PUTS 应答消息体格式。可以是其它标记, 但须有别于 OMA 其它定义。

3、应答消息体

PUTS 应答消息体格式如下:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema                                xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="com:huawei:puts-result"            xmlns:ns1="com:huawei:puts-result"
targetNamespace="com:huawei:puts-result" elementFormDefault="qualified">
```

```
  <xs:element name="list">
```

```
    <xs:complexType>
```

```
      <xs:sequence>
```

```
        <xs:element                        name="auid"                type="auidType"
maxOccurs="unbounded"/>
```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="audType">
    <xs:sequence>
        <xs:element name="file" type="fileType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="fileType">
    <xs:sequence>
        <xs:element name="result" type="xs:string"/>
        <xs:element name="etag" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>

```

说明:

PUTS 应答消息体是针对 PUTS 应答消息的批处理特性定义的, 是对 OMA XCAP 的补充;

“com:huawei:puts-result” 为 PUTS 应答消息的命名空间 (NameSpace), 可以是其它约定值;

PUTS 应答消息体包含 list、aud 和 file 三级, list 为根节点, aud 描述业务信息, file 描述文档信息, 也可以是其它约定元素名;

aud 节点: name 为 aud 名字, ctype 为 aud 对应的 schema;

file 节点: name 为文档名, result 为每个文档操作的返回码(参考 OMA XCAP PUT 信令中对返回码的定义), etag 为文档新的版本号;

创建文档或节点成功 result 取 201, 修改文档或节点成功 result 取 200。

PUTS 批量处理应答消息的一个示例如下:

HTTP/1.1 200 OK

Content-Type: "application/puts-result+xml";

Content-Length: (...)

```
<list xmlns="com:huawei:puts-result">
  <aud name="ims-pim">
    <file name="index.xml">
      <result>201</result>
      <etag>reeeedd</etag>
    </file>
  </aud>
  <aud name="resource-lists">
    <file name="friend.xml">
      <result>200</result>
      <etag>reeeedd</etag>
    </file>
  </aud>
</list>
```

从上例可以看出，创建和修改返回码（result）不同，在这点上遵从 OMA 定义。

步骤 s405，向客户端返回应答消息。

参考图 4，是本发明实施例的 XML 文档管理批量查询（GETS）处理方法流程示意图，下面对其进行详细描述。

步骤 s501，发送 GETS 批量处理请求。

所述文档管理批量查询处理请求信令格式如下：

1、请求行

名称	类型	说明
Method	token	XCAP 请求方法，取值 GETS
Request-URI	HTTP URI	指定请求操作的 XCAP 资源。
HTTP-Version		HTTP 版本， 取值 HTTP/1.1

GETS 信令请求行如下:

GETS http://xcap.example.com/00/users/sip:XXX@huawei.com/batch
HTTP/1.1

其中,“batch”表示批量获取文档,“00”是 auid 替代符,表示操作的具体业务在消息中定义,XXX 表示发起请求的用户号码,“sip:”表示这是一个 sip uri,“@huawei.com”为用户 XXX 归属的 XDMS 域名。

2、请求消息头

GETS 信令请求消息头如下:

Content-type: application/gets+xml

Content-Length:(...)

application/gets+xml 是 GETS 请求的消息体格式。可以是其它标记,但须有别于 OMA 其它定义。

3、请求消息体

GETS 请求消息体格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema                    xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="com:huawei:gets"                    targetNamespace="com:huawei:gets"
elementFormDefault="qualified">
  <xs:element name="list">
    <xs:complexType>
      <xs:sequence>
        <xs:element                    name="auid"                    type="auidType"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="fileType">
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="auidType">
```

```

<xs:sequence>
  <xs:element name="file" type="fileType" maxOccurs="unbounded"/>
</xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>

```

说明:

GETS 请求消息体是针对 GETS 请求消息的批处理特性定义的, 是对 OMA XCAP 的补充;

"com.huawei:gets" 为 GETS 请求消息体的命名空间 (namespace), 可以是其它约定值;

GETS 请求消息体分 list、auid、file 三级, 可用来描述一次性获取多个业务的多个文档。其中, list 为根节点, auid 表示要处理的业务, file 表示要处理的文档;

auidType 中 name 为 auid 的名字;

fileType 中 name 为文档名, 可以是通配符, 如用 "*" 表示获取所有文档;

auid 下既指定具体文档名又指定文档名为 "*" 时, 则取该用户 auid 下的所有文档。

GETS 批量处理请求消息的一个示例如下:

GETS 请求消息

GETS http://xcap.example.com/00/users/sip:13500012345@huawei.com/batch
HTTP/1.1

...

Content-type: application/gets+xml

Content-Length:(...)

```
<list xmlns="com.huawei:gets">
```

```
  <auid name="ims-pim">
```

```
    <file name="index.xml"/>
```

```
</audid>
<audid name="resource-lists">
  <file name="file_001.xml"/>
  <file name="*/>
</audid>
</list>
```

上例表示用户 13500012345 请求获取 ims-pim 下的 index.xml 和 resource-lists 下所有文档。

步骤 s502, 对 GETS 批量处理请求按照 GETS 请求消息格式进行解码, 获取 GETS 批量处理信息。

XDMS 解析 GETS 批量处理请求的消息头, 获取 Content-Type 类型为 application/gets+xml, 进而根据对应的 GETS 消息体格式解码 GETS 批量处理请求消息体, 获取 GETS 批量处理信息。

步骤 s503, 根据 GETS 批量处理信息, 执行 GETS 批量处理操作, 完成相应的文档信息批量查询。

步骤 s504, 对 GETS 批量处理的结果按照 GETS 应答消息格式进行编码。

GETS 应答消息组成如下:

1、应答状态行

格式: HTTP-Version + Status-Code + Reason-Phrase

HTTP 版本, HTTP-Version: HTTP/1.1

状态码, Status-Code: 200 或 500

Reason-Phrase: Status-Code 取 200 时为 OK, 取 500 时为 NOK。

2、应答消息头

Content-Type: application/gets-result+xml

Content-Length: (...)

application/gets-result+xml 是 GETS 应答消息体格式。可以是其它标记, 但须有别于 OMA 其它定义。

3、应答消息体

GETS 应答消息体格式:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema                                xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="com:huawei:gets-result"            targetNamespace="com:huawei:gets-result"
elementFormDefault="qualified">
  <xs:element name="list">
    <xs:complexType>
      <xs:sequence>
        <xs:element                name="auid"                type="auidType"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="file">
    <xs:complexType>
      <xs:sequence>
        <xs:any                namespace="##other"                processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="etag" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="auidType">
    <xs:sequence>
      <xs:element ref="file" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="ctype" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

说明:

GETS 应答消息体是针对 GETS 应答消息的批处理特性定义的, 是对 OMA XCAP 的补充;

"com.huawei:gets-result"为 GETS 应答消息体命名空间 (NameSpace), 可以是其它约定值;

GETS 应答消息体消息体分 list、auid 和 file 三级, list 为根节点, auid 描述业务, file 描述文档, 也可以是其它约定名称;

auid 节点: name 为 auid 名字, ctype 为 auid 对应的 schema(即消息体格式);

file 节点: name 为文档名, etag 为文档版本号。

GETS 批量处理应答消息的一个示例如下:

GETS 应答消息

HTTP/1.1 200 OK

Content-Type: "application/gets-result+xml"

Content-Length: (...)

```
<list xmlns="com.huawei:gets-result">
```

```
  <auid name="ims-pim" ctype="application/ims-pim+xml">
```

```
    <file name="index.xml" etag="dffffgg">
```

```
      <ims-pim xmlns="com.huawei:ims-pim">
```

```
        <name>bob</name>
```

```
        <tel>58012700</tel>
```

```
      </ims-pim>
```

```
    </file>
```

```
  </auid>
```

```
<auid name="resource-lists" ctype="application/resource-lists+xml">
```

```
  <file name="file_001.xml" etag="reeeedd">
```

```
    <contact-profiles xmlns="com.huawei:resource-lists">
```

```
      <name>tom</name>
```

```
      <tel>58012701</tel>
```

```
    </contact-profile>
```

```

</file>
<file name="file_002.xml" etag="rccccff">
  <contact-profiles xmlns="com:huawei: resource-lists">
    <name>bill</name>
    <tel>58012702</tel>
  </contact-profiles>
</file>
</auid>
</list>

```

此例返回“ims-pim”中的 index.xml，以及“resource-lists”中的仅有的两篇文档。

步骤 s505，向客户端返回应答消息。

参考图 5，是本发明实施例的 XML 文档管理批量删除（DELETES）处理流程示意图，下面对其进行详细描述。

步骤 s601，发送 DELETES 批量处理请求。

所述文档管理批量删除处理请求信令格式如下：

1、请求行

名称	类型	说明
Method	token	XCAP 请求方法，取值 DELETES
Request-URI	HTTP URI	指定请求操作的 XCAP 资源。
HTTP-Version		HTTP 版本，取值 HTTP/1.1

DELETES 信令请求行如下：

DELETES http://xcap.example.com/00/users/sip:XXX@huawei.com/batch
HTTP/1.1

其中，“batch”表示批量删除文档或节点，“00”是 auid 替代符，表示操作的具体业务在消息中定义，XXX 表示发起请求的用户号码，“sip:”表示这是一个 sip uri，“@huawei.com”为用户 XXX 归属的 XDMS 域名。

2、请求消息头

DELETES 信令请求消息头如下:

Content-Type: application/deletes+xml;

Content-Length: (...)

application/deletes+xml 是 DELETES 请求的消息体格式。可以是其它标记,但须有别于 OMA 其它定义。

3、请求消息体

DELETES 请求消息体格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema                xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="com:huawei:deletes"                targetNamespace="com:huawei:deletes"
elementFormDefault="qualified">
  <xs:element name="list">
    <xs:complexType>
      <xs:sequence>
        <xs:element                name="auid"                type="auidType"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="fileType">
    <xs:sequence>
      <xs:element                name="node"                type="xs:string"                minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="etag" type="xs:string" use="optional"/>
  </xs:complexType>
  <xs:complexType name="auidType">
    <xs:sequence>
      <xs:element name="file" type="fileType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

</xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>

```

说明:

DELETES 请求消息体是针对 DELETES 请求消息的批处理特性定义的。

"com.huawei:deletes"为 DELETES 请求消息的命名空间 (Namespace), 可以是其它约定值;

DELETES 请求消息体分 list、auid、file、node 四级, list 为根节点, auid 为业务标识, file 表示删除的文档, node 表示删除的节点;

按 schema 定义, 可以同时删除多个业务的文档, 也可以同时删除某业务下的多个文档或所有文档 (如用通配符 "*" 作为文档名), 或者删除一篇文档的多个节点;

auidType: name 为 auid 的名字;

fileType: name 为文档名, etag 为文档版本号;

node: 节点的 XPATH 路径。

DELETES 批量处理请求消息的一个示例如下:

DELETES 请求消息

DELETES

http://xcap.example.com/00/users/sip:13500012345@huawei.com/batch HTTP/1.1

...

Content-type: application/deletes+xml

Content-Length:(...)

```
<list xmlns="com.huawei:deletes">
```

```
  <auid name="resource-lists"/>
```

```
    <file name="file_001.xml" etag="eddddkk">
```

```
      <node>/resource-lists/list[@name="friends"]</node>
```

```
      <node>/resource-lists/list[@name="class"]</node>
```

```
    </file>
```

```
<file name="file_002.xml" etag="abbbbcc"/>
</auid>
<auid name="test"><file name="file_003" etag="adddee" /></auid>
<auid name="test2"><file name="*"/></auid>
</list>
```

说明:

此例表示用户 13500012345 请求执行文档批量删除操作;

在 resource-lists 中, 删除文档 file_002.xml 和 file_001.xml 的两个节点, 节点的 XPATH 路径分别为 /resource-lists/list[@name="friends"] 和 /resource-lists/list[@name="class"];

在 test 中, 删除文档 file_003;

在 test2 中, 删除所有文档。

步骤 s602, 对 DELETES 批量处理请求按照 DELETES 请求消息格式进行解码, 获取 DELETES 批量处理信息。

XDMS 解析 DELETES 批量处理请求的消息头, 获取 Content-Type 类型为 application/deletes+xml, 进而根据对应的 DELETES 消息体格式解码 DELETES 批量处理请求消息体, 获取 DELETES 批量处理信息。

步骤 s603, 根据 DELETES 批量处理信息, 执行 DELETES 批量处理操作, 完成相应的文档信息批量删除。

步骤 s604, 对 DELETES 批量处理的结果按照 DELETES 应答消息格式进行编码。

DELETES 应答消息组成如下:

1、应答状态行

格式: HTTP-Version + Status-Code + Reason-Phrase

HTTP 版本, HTTP-Version: HTTP/1.1

状态码, Status-Code: 200 或 500

Reason-Phrase: Status-Code 取 200 时为 OK, 取 500 时为 NOK。

2、应答消息头

Etag: "reeeedd"

...

Content-Length: (...)

DELETES 应答消息无消息体，因此不需要定义 Content-Type。

DELETES 应答消息示例如下：

DELETES 应答消息

HTTP/1.1 200 OK

Etag: “reeeedd”

...

Content-Length: 0

此应答消息表示批量删除成功，存在节点删除的文档新版本号为 reeeddd。

步骤 s605，向客户端返回应答消息

参考图 6，是本发明实施例的 XML 文档管理批量移动（MOVE）处理方法流程图示意图，下面对其进行详细描述。

步骤 s701，发送 MOVE 批量处理请求。

所述文档管理批量移动处理请求信令格式如下：

1、请求行

名称	类型	说明
Method	token	XCAP 请求方法,取值 MOVE
Request-URI	HTTP URI	指定请求操作的 XCAP 资源。
HTTP-Version		HTTP 版本，取值 HTTP/1.1

MIME 信令请求行如下：

MIME

http://xcap.example.com/resource-lists/users/sip:XXX@huawei.com/batch HTTP/1.1

其中，“batch”表示批量移动文档中的节点，resource-lists 表示操作的具体业务，XXX 表示发起请求的用户号码，“sip:”表示这是一个 sip uri，“@huawei.com”为用户 XXX 归属的 XDMS 域名。

2、请求消息头

MIME 信令请求消息头如下：

Content-Type:application/move+xml

Content-Length: (...)

application/move+xml 是 MOVE 请求的消息体格式。可以是其它标记，但须有别于 OMA 其它定义。

3、请求消息体

MOVE 请求消息体格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema                                xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="com:huawei:move"                    targetNamespace="com:huawei:move"
elementFormDefault="qualified">
  <xs:element name="list">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="node" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="node">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="from" type="fromType"/>
        <xs:element name="to" type="toType"/>
      </xs:sequence>
      <xs:attribute name="seq" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="fromType">
    <xs:attribute name="filename" type="xs:string" use="required"/>
    <xs:attribute name="etag" type="xs:string" use="optional"/>
  </xs:complexType>
```

```

<xs:complexType name="toType">
  <xs:attribute name="filename" type="xs:string" use="required"/>
  <xs:attribute name="etag" type="xs:string" use="optional"/>
</xs:complexType>
</xs:schema>

```

说明:

MOVE 请求消息体是针对 MOVE 请求消息的批处理特性定义的, 是对 OMA XCAP 的补充;

"com.huawei:MOVE" 是 MOVE 请求消息的命名空间 (NameSpace), 可以是其它约定值;

MOVE 请求消息体由 list 和 1~N 个 node 节点构成, 每个 node 包含序号 (seq)、源路径 (在 from 标签中, 遵从 XPATH 语法)、目的路径 (在 to 标签中, 遵从 XPATH 语法), 可以是其它约定名称;

from: filename 为源文档名, etag 为源文档版本号;

to: filename 为目的文档名, etag 为目的文档版本号;

MOVE 信令可用于在不同文档之间移动节点, 也可以在同一文档内部移动节点。

MOVE 批量处理请求消息的一个示例如下:

MOVE 请求消息

MOVE

http://xcap.example.com/resource-lists/users/sip:13500012345@huawei.com/batch
HTTP/1.1

...

Content-Type: application/move+xml

Content-Length: (...)

```

<list xmlns="com.huawei:move">
  <node seq="1">
    <from filename="list1.xml" etag="abbbbcc">

```

```
/resource-lists/list[@name="friends"]/entry[@uri="sip:petri@example.com"]  
    </from>  
    <to filename="list2.xml" etag="abbbbdd">
```

```
/resource-lists/list[@name="colleague"]/entry[@uri="sip:petri@example.com"]  
    </to>  
    </node>  
    </list>
```

此例表示用户 13500012345 请求对“resource-lists”业务执行文档间节点移动操作，将 list1.xml 中分组名为“friends”的节点（uri="sip:petri@example.com"）移动到文档 list2.xml 的“colleague”分组下。

步骤 s702，对 MOVE 批量处理请求按照 MOVE 请求消息格式进行解码，获取 MOVE 批量处理信息。

XDMS 解析 MOVE 批量处理请求的消息头，获取 Content-Type 类型为 application/move+xml，进而根据对应的 MOVE 消息体格式解码 MOVE 批量处理请求消息体，获取 MOVE 批量处理信息。

步骤 s703，根据 MOVE 批量处理信息，执行 MOVE 批量处理操作，完成相应的文档信息批量移动。

步骤 s704，对 MOVE 批量处理的结果按照 MOVE 应答消息格式进行编码。

MOVE 应答消息组成如下：

1、应答状态行

格式：HTTP-Version + Status-Code + Reason-Phrase

HTTP 版本，HTTP-Version: HTTP/1.1

状态码，Status-Code: 200 或 500

Reason-Phrase: Status-Code 取 200 时为 OK，取 500 时为 NOK。

2、应答消息头

Etag: "reeeedd"

...

Content-Length: (...)

MOVE 应答消息无消息体，因此不需要定义 Content-Type。

MOVE 应答消息的一个示例如下：

MOVE 应答消息

HTTP/1.1 200 OK

Etag: "reeeedd"

...

Content-Length: 0

步骤 s705，向客户端返回应答消息。

下面对与上述方法实施例相应的 XML 文档管理系统、XML 文档管理客户端以及 XML 文档管理服务器进行详细说明。

参考图 7，是本发明实施例的 XML 文档管理系统的组成示意图，该 XML 文档管理系统可执行上述方法实施例中的 XML 文档管理请求方法以及 XML 文档管理方法，具体实施时，主要包括：

XML 文档管理客户端 1，用于根据用户输入的文档管理批量处理请求指示，生成文档管理批量处理请求，并向服务器 2 发送所述文档管理批量处理请求；在接收到来自服务器 2 的文档管理批量处理应答消息时，读取所述批量处理应答消息中携带的状态码，并根据所述批量处理应答消息的消息头识别是否存在消息体，若存在，则根据相应的批量处理应答消息体格式对消息体进行解码，获取文档管理批量处理结果信息；

XML 文档管理服务器 2，用于接收文档管理请求，确定所述文档管理请求为文档管理批量处理请求，从所述文档管理批量处理请求中获取相应的文档管理批量处理信息，并根据所述文档管理批量处理信息，执行相应的文档管理批量处理操作；基于文档管理批量处理操作的执行结果信息，按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息，并向文档管理客户端 1 发送所述文档管理批量处理应答消息。

参考图 8，是本发明实施例的 XML 文档管理客户端的结构示意图，该客户端可用于执行上述方法实施例中的 XML 文档管理请求方法，主要包括：

用户操作单元 11，用于接收用户的批量处理请求指示；

请求生成单元 12，用于根据所述文档管理批量处理请求指示生成文档管理

批量处理请求;

请求发送单元 13, 用于将所述文档管理批量处理请求发送给服务器;

应答消息接收单元 14, 用于接收来自服务器的批量处理应答消息;

应答消息处理单元 15, 用于读取所述批量处理应答消息中携带的状态码, 并根据所述批量处理应答消息的消息头识别是否存在消息体, 若存在, 则根据相应的批量处理应答消息体格式对消息体进行解码, 获取文档管理批量处理结果信息。

参考图 9 是本发明实施例的 XML 文档管理服务器的结构示意图, 该服务器主要用于执行上述方法实施例中的 XML 文档管理方法, 主要包括:

请求接收单元 21, 用于接收文档管理请求;

请求解析单元 22, 用于确定所述文档管理请求为文档管理批量处理请求, 并从所述文档管理批量处理请求中获取相应的文档管理批量处理信息。所述请求解析单元 22 具体可包括:

消息体格式识别单元 221, 用于解析所述文档管理批量处理请求的消息头, 以识别所述文档管理批量处理请求的消息体格式;

批量处理信息获取单元 222, 用于根据所述消息体格式解码所述文档管理请求的消息体, 获取相应的文档管理批量处理信息。所述批量处理信息获取单元 222 具体可包括:

批量增加/修改处理信息获取子单元 2221, 用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量增加/修改请求消息体格式时, 根据所述文档管理批量增加/修改请求消息体格式解码所述文档管理请求的消息体, 获取相应的文档管理批量增加/修改处理信息;

批量查询处理信息获取子单元 2222, 用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量查询请求消息体格式时, 根据所述文档管理批量查询请求消息体格式解码所述文档管理请求的消息体, 获取相应的文档管理批量查询处理信息;

批量删除处理信息获取子单元 2223, 用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量删除请求消息体格式时, 根据所述文档管理批量删除请求消息体格式解码所述文档管理请求的消息体, 获取相应的文档管理批量删除处理信息;

批量移动处理信息获取子单元 2224, 用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量移动请求消息体格式时, 根据所述文档管理批量移动请求消息体格式解码所述文档管理请求的消息体, 获取相应的文档管理批量移动处理信息。

请求处理单元 23, 用于根据所述文档管理批量处理信息, 执行相应的文档管理批量处理操作。所述请求处理单元具体可包括:

批量增加/修改请求处理子单元 231, 用于根据所述文档管理批量增加/修改处理信息, 执行文档管理批量增加/修改处理操作;

批量查询请求处理子单元 232, 用于根据所述文档管理批量查询处理信息, 执行文档管理批量查询处理操作;

批量删除请求处理子单元 233, 用于根据所述文档管理批量删除处理信息, 执行文档管理批量删除处理操作;

批量移动请求处理子单元 234, 用于根据所述文档管理批量移动处理信息, 执行文档管理批量移动处理操作。

应答处理单元 24, 用于基于文档管理批量处理操作的执行结果信息, 按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息; 向文档管理客户端发送所述文档管理批量处理应答消息。所述应答处理单元 24 具体可包括:

应答消息生成单元 241, 用于基于文档管理批量处理操作的执行结果信息, 按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息;

应答消息发送单元 242, 用于向文档管理客户端发送所述文档管理批量处理应答消息。

通过上述实施例可以看出, 本发明实施例采用客户端将需要对多个文档或单个文档下的多个节点的批量处理操作指示, 通过一条文档管理批量处理请求发送给服务端, 服务端在确定为文档管理批量处理请求时, 获取相应的文档管理批量处理信息并执行相应的文档管理批量处理操作; 由于客户端只需要发送一条信令来进行多个文档或单个文档下的多个节点的批量处理操作, 减小了操作时延, 提高了操作效率, 提高了数据操作的可靠性。

本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,

是可以通过计算机程序来指令相关的硬件来完成。所述的程序可存储于一计算机可读取存储介质中，该程序在执行时，可包括如上述各方法的实施例的流程。其中，所述的存储介质可为磁碟、光盘、只读存储记忆体（Read-Only Memory, ROM）或随机存储记忆体（Random Access Memory, RAM）等。

以上所述是本发明的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也视为本发明的保护范围。

权 利 要 求

1、一种可扩展标记语言文档管理请求方法，其特征在于，包括：
接收用户的文档管理批量处理请求指示；
根据所述文档管理批量处理请求指示生成文档管理批量处理请求；
将所述文档管理批量处理请求发送给服务器。

2、如权利要求 1 所述的方法，其特征在于，所述文档管理批量处理请求为文档管理批量增加/修改请求，或为文档管理批量查询请求，或为文档管理批量删除请求，或为文档管理批量移动请求。

3、一种可扩展标记语言文档管理方法，其特征在于，包括：
接收文档管理请求；
确定所接收到的文档管理请求为文档管理批量处理请求；
从所述文档管理批量处理请求中获取相应的文档管理批量处理信息；
根据所述文档管理批量处理信息，执行相应的文档管理批量处理操作。

4、如权利要求 3 所述的方法，其特征在于，所述确定文档管理请求为文档管理批量处理请求的步骤具体为：

根据所述文档管理请求的方法名确定所述文档管理请求为文档管理批量处理请求。

5、如权利要求 3 或 4 所述的方法，其特征在于，所述从所述文档管理批量处理请求中获取相应的文档管理批量处理信息的步骤具体为：

解析所述文档管理批量处理请求的消息头，以确定所述文档管理批量处理请求的消息体格式；

根据所述消息体格式解码所述文档管理请求的消息体，获取相应的文档管理批量处理信息。

6、如权利要求 3 所述的方法，其特征在于，所述根据所述文档管理批量处

理信息，执行相应的文档管理批量处理操作的步骤之后还包括步骤：

基于文档管理批量处理操作的执行结果信息，按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息，并向所述文档管理请求的发送方发送所述文档管理批量处理应答消息。

7、如权利要求 5 所述的方法，其特征在于，所述文档管理批量处理请求为文档管理批量增加/修改请求，或为文档管理批量查询请求，或为文档管理批量删除请求，或为文档管理批量移动请求。

8、一种可扩展标记语言文档管理客户端，其特征在于，包括：

用户操作单元，用于接收用户的批量处理请求指示；

请求生成单元，用于根据所述文档管理批量处理请求指示生成文档管理批量处理请求；

请求发送单元，用于将所述文档管理批量处理请求发送给服务器。

9、如权利要求 8 所述的客户端，其特征在于，还包括：

应答消息接收单元，用于接收来自服务器的批量处理应答消息；

应答消息处理单元，用于读取所述批量处理应答消息中携带的状态码，并根据所述批量处理应答消息的消息头识别是否存在消息体，若存在，则根据相应的批量处理应答消息体格式对消息体进行解码，获取文档管理批量处理结果信息。

10、一种可扩展标记语言文档管理服务器，其特征在于，包括：

请求接收单元，用于接收文档管理请求；

请求解析单元，用于确定所述文档管理请求为文档管理批量处理请求，并从所述文档管理批量处理请求中获取相应的文档管理批量处理信息；

请求处理单元，用于根据所述文档管理批量处理信息，执行相应的文档管理批量处理操作。

11、如权利要求 10 所述的服务器，其特征在于，所述请求解析单元包括：

消息体格式识别单元，用于解析所述文档管理批量处理请求的消息头，以识别所述文档管理批量处理请求的消息体格式；

批量处理信息获取单元，用于根据所述消息体格式解码所述文档管理请求的消息体，获取相应的文档管理批量处理信息。

12、如权利要求 11 所述的服务器，其特征在于，所述批量处理信息获取单元包括：

批量增加/修改处理信息获取子单元，用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量增加/修改请求消息体格式时，根据所述文档管理批量增加/修改请求消息体格式解码所述文档管理请求的消息体，获取相应的文档管理批量增加/修改处理信息；

批量查询处理信息获取子单元，用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量查询请求消息体格式时，根据所述文档管理批量查询请求消息体格式解码所述文档管理请求的消息体，获取相应的文档管理批量查询处理信息；

批量删除处理信息获取子单元，用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量删除请求消息体格式时，根据所述文档管理批量删除请求消息体格式解码所述文档管理请求的消息体，获取相应的文档管理批量删除处理信息；

批量移动处理信息获取子单元，用于当所述消息体格式识别单元所识别的消息体格式为文档管理批量移动请求消息体格式时，根据所述文档管理批量移动请求消息体格式解码所述文档管理请求的消息体，获取相应的文档管理批量移动处理信息。

13、如权利要求 12 所述的服务器，其特征在于，所述请求处理单元包括：

批量增加/修改请求处理子单元，用于根据所述文档管理批量增加/修改处理信息，执行文档管理批量增加/修改处理操作；

批量查询请求处理子单元，用于根据所述文档管理批量查询处理信息，执行文档管理批量查询处理操作；

批量删除请求处理子单元，用于根据所述文档管理批量删除处理信息，执

行文档管理批量删除处理操作;

批量移动请求处理子单元, 用于根据所述文档管理批量移动处理信息, 执行文档管理批量移动处理操作。

14、如权利要求 10 至 13 任意一项所述的服务器, 其特征在于, 还包括:

应答处理单元, 用于基于文档管理批量处理操作的执行结果信息, 按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息; 向文档管理客户端发送所述文档管理批量处理应答消息。

15、如权利要求 14 所述的服务器, 其特征在于, 所述应答处理单元包括:

应答消息生成单元, 用于基于文档管理批量处理操作的执行结果信息, 按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息;

应答消息发送单元, 用于向文档管理请求发送方发送所述文档管理批量处理应答消息。

16、一种可扩展标记语言文档管理系统, 客户端和服务端, 其特征在于, 所述客户端包括:

用户操作单元, 用于接收用户的批量处理请求指示;

请求生成单元, 用于根据所述文档管理批量处理请求指示生成文档管理批量处理请求;

请求发送单元, 用于将所述文档管理批量处理请求发送给服务器;

所述服务器包括:

请求接收单元, 用于接收文档管理请求;

请求解析单元, 用于确定所述文档管理请求为文档管理批量处理请求, 并从所述文档管理批量处理请求中获取相应的文档管理批量处理信息;

请求处理单元, 用于根据所述文档管理批量处理信息, 执行相应的文档管理批量处理操作。

17、如权利要求 16 所述的系统, 其特征在于, 所述服务器还包括:

应答处理单元，用于基于文档管理批量处理操作的执行结果信息，按照相应的文档管理批量处理操作应答消息格式组装生成文档管理批量处理应答消息；向文档管理客户端发送所述文档管理批量处理应答消息。

18、如权利要求 16 或 17 所述的系统，其特征在于，所述客户端还包括：
应答消息接收单元，用于接收来自服务器的批量处理应答消息；

应答消息处理单元，用于读取所述批量处理应答消息中携带的状态码，并根据所述批量处理应答消息的消息头识别是否存在消息体，若存在，则根据相应的批量处理应答消息体格式对消息体进行解码，获取文档管理批量处理结果信息。

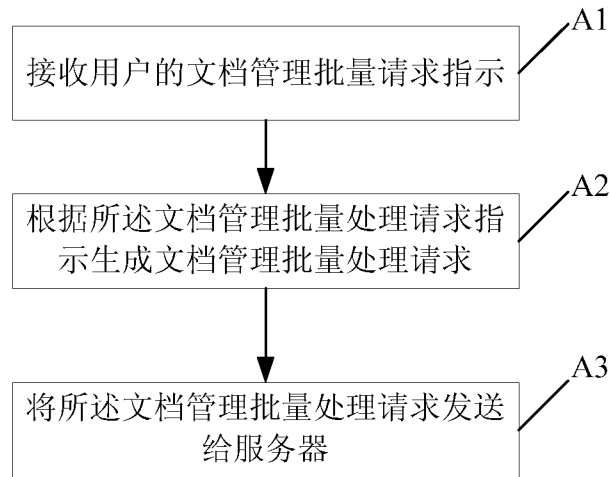


图 1

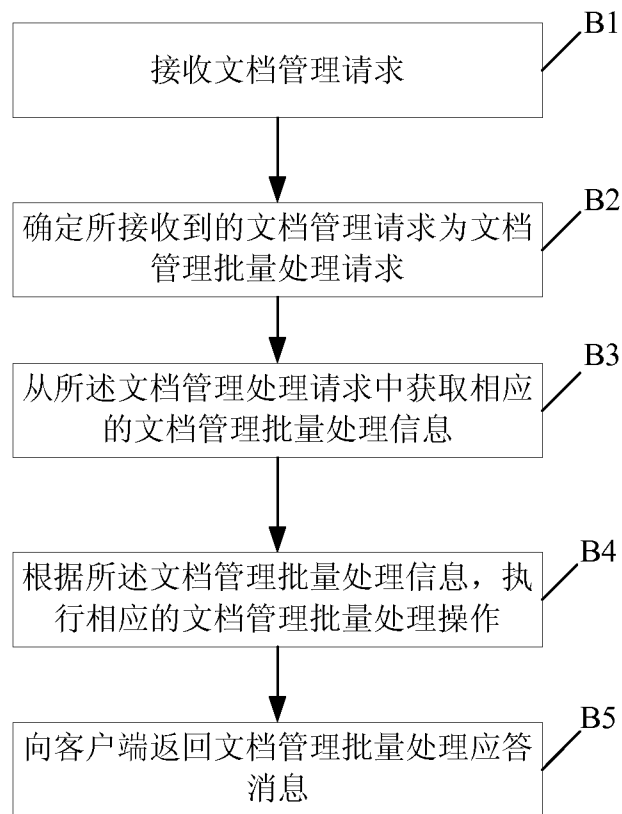


图 2

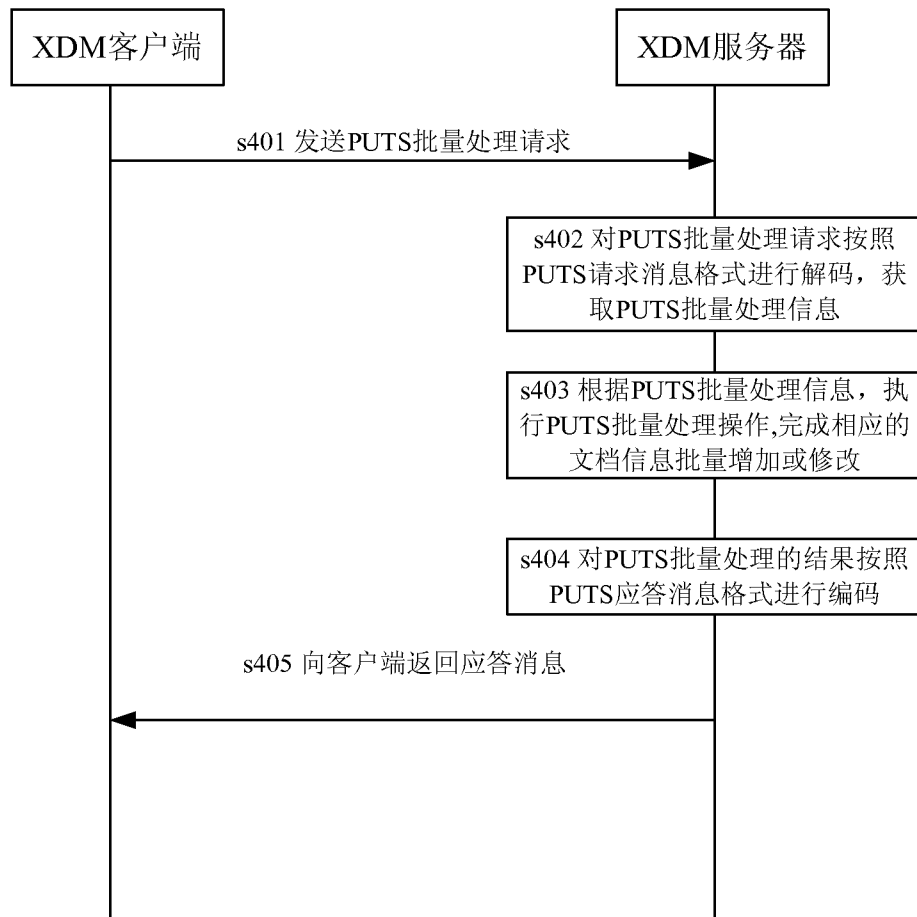


图 3

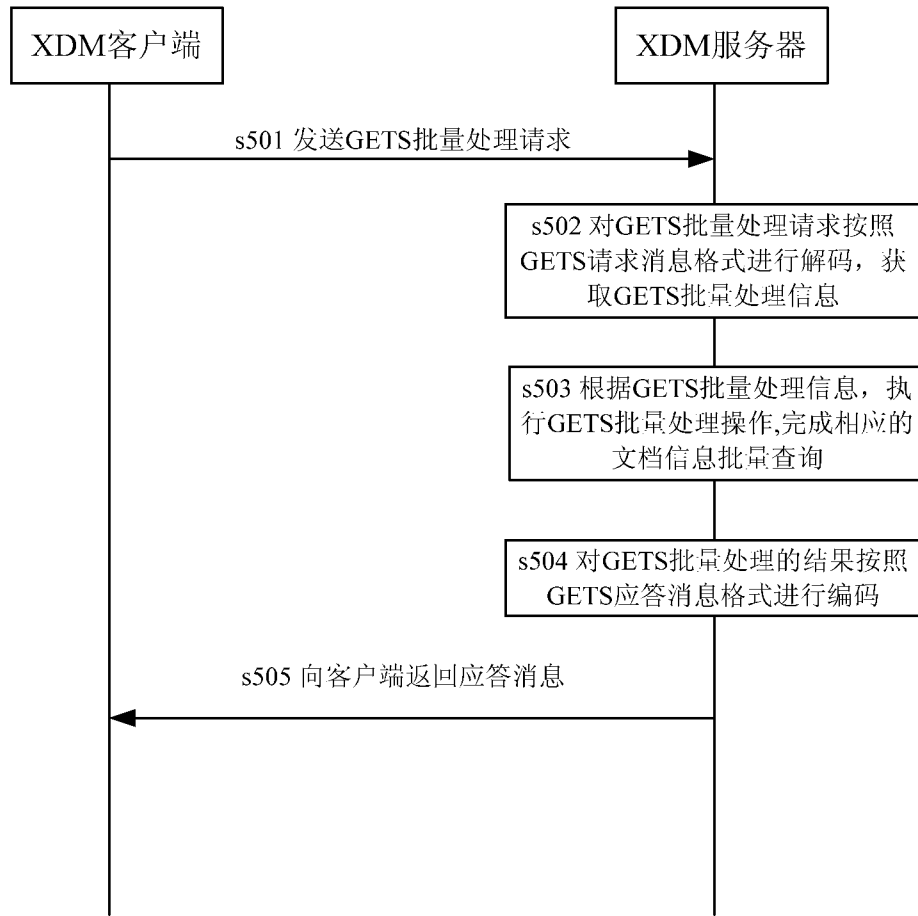


图 4

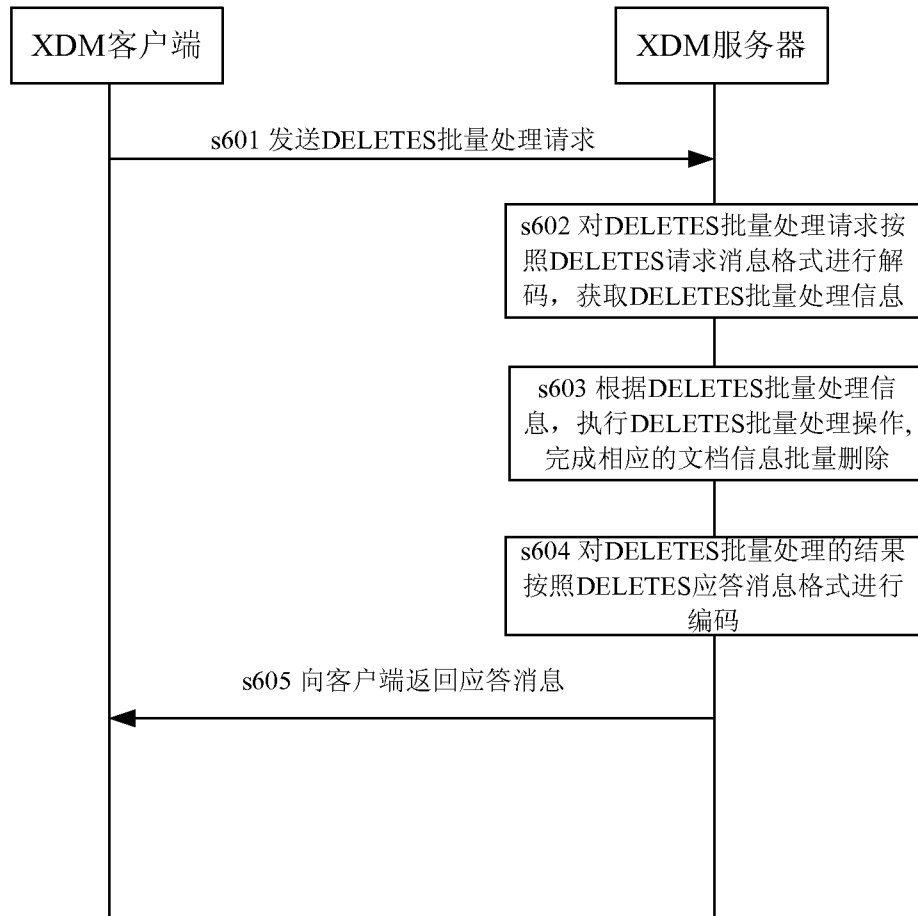


图 5

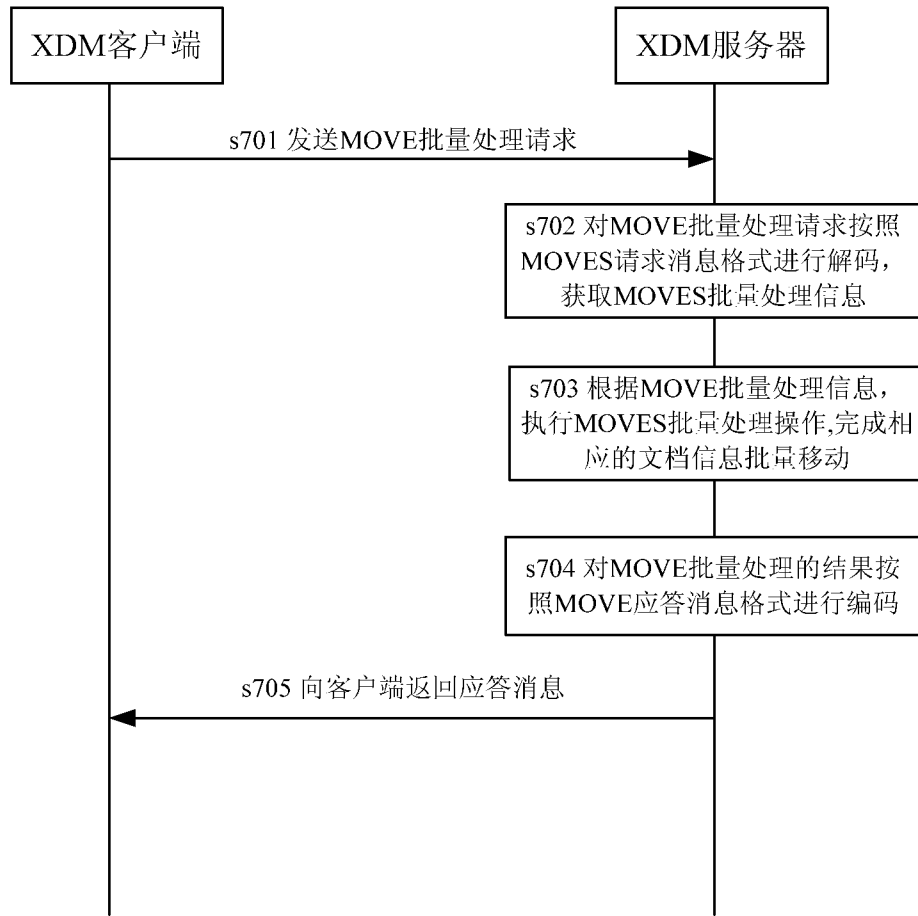


图 6

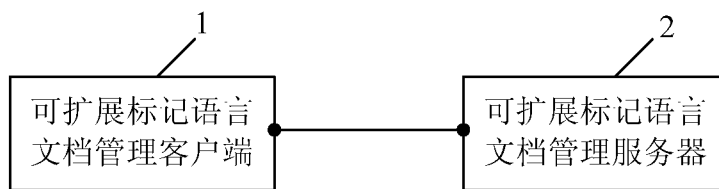


图 7

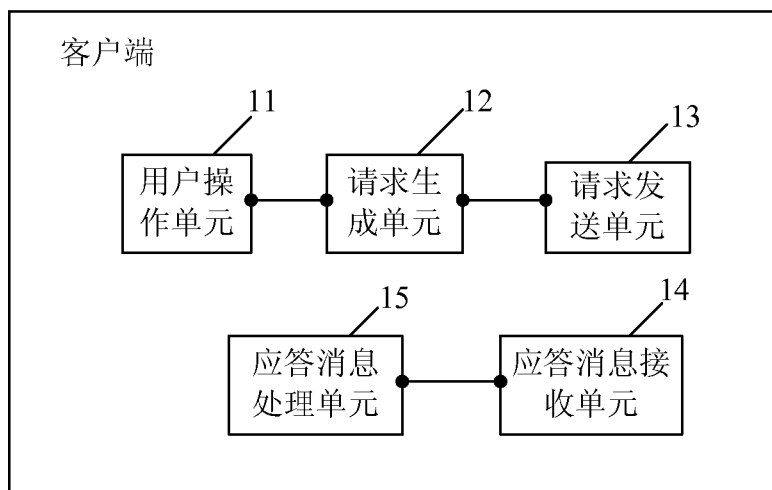


图 8

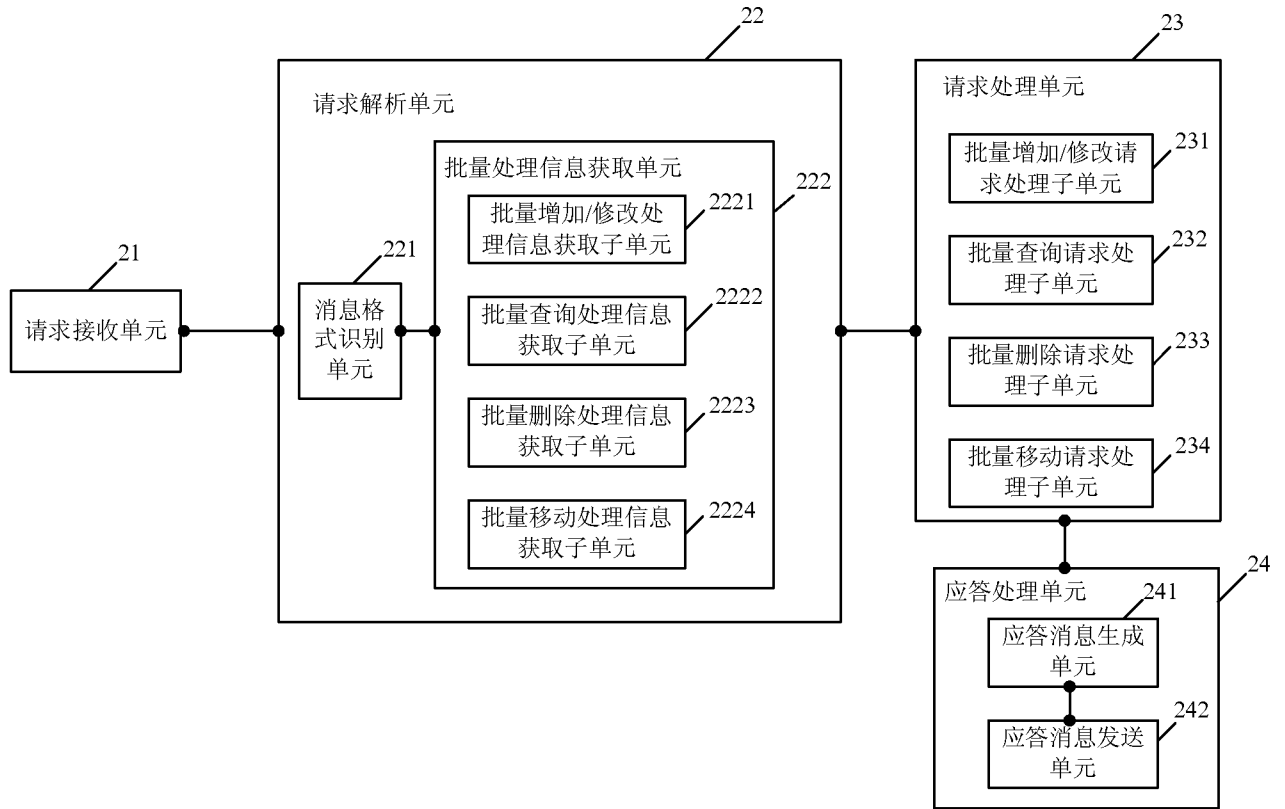


图 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2008/073482

A. CLASSIFICATION OF SUBJECT MATTER

H04L12/54 (2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI;EPODOC;PAJ: XML, extensible makeup language, batch, document, file, process, plurality, request, header, management, put, get, delete, move

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN1690969A(IBM) 02 Nov. 2005(02.11.2005) the description page 7, line 3-page 9, line 15; figures 2-5	1-4,6,8,10,14-17
A	WO2007075234A1(MICROSOFT CORP) 05 July 2007(05.07.2005) the whole document	1-18
A	US2007074215A1(BETHEA T J et al.) 29 March 2007(29.03.2007) the whole document	1-18
PX	CN101197827A(HUAWEI TECHNOLOGY CO LTD) 11 June 2008(11.06.2008) the whole document	1-18

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“A” document defining the general state of the art which is not considered to be of particular relevance	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“E” earlier application or patent but published on or after the international filing date	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“L” document which may throw doubts on priority claim (S) or which is cited to establish the publication date of another citation or other special reason (as specified)	“&” document member of the same patent family
“O” document referring to an oral disclosure, use, exhibition or other means	
“P” document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search
03 March 2009(03.03.2009)

Date of mailing of the international search report
26 Mar. 2009 (26.03.2009)

Name and mailing address of the ISA/CN
The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China
100088
Facsimile No. 86-10-62019451

Authorized officer
WANG, Wei
Telephone No. (86-10)62411506

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/CN2008/073482

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN1690969A	02.11.2005	US2005251575A1	10.11.2005
WO2007075234A1	05.07.2007	US2007143665A1	21.06.2007
		EP1969549A1	17.09.2008
		KR20080084971A	22.09.2008
		CN101331511A	24.12.2008
US2007074215A1	29.03.2007	NONE	
CN101197827A	11.06.2008	NONE	

国际检索报告

国际申请号
PCT/CN2008/073482

A. 主题的分类

H04L12/54 (2006.01)i

按照国际专利分类表(IPC)或者同时按照国家分类和 IPC 两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC: H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

WPI;EPODOC;PAJ: XML, extensible makeup language, batch, document, file, process, plurality, request, header, management, put, get, delete, move

CNPAT;CNKI: XML, 扩展标记语言, 扩展置标语言, 批量, 批处理, 文档, 文件, 请求, 头, 管理

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
X	CN1690969A (国际商业机器公司) 02.11 月 2005(02.11.2005) 说明书第 7 页第 3 行-第 9 页第 15 行; 图 2-5	1-4, 6, 8, 10, 14-17
A	WO2007075234A1(微软公司) 05.7 月 2007(05.07.2007) 全文	1-18
A	US2007074215A1(BETHEA T J 等) 29.3 月 2007(29.03.2007) 全文	1-18
PX	CN101197827A(华为技术有限公司) 11.6 月 2008(11.06.2008) 全文	1-18

☐ 其余文件在 C 栏的续页中列出。

☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期
03.3 月 2009(03.03.2009)

国际检索报告邮寄日期
26.3 月 2009 (26.03.2009)

中华人民共和国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路 6 号 100088
传真号: (86-10)62019451

授权官员
王为
电话号码: (86-10) **62411506**

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2008/073482

检索报告中引用的 专利文件	公布日期	同族专利	公布日期
CN1690969A	02.11.2005	US2005251575A1	10.11.2005
WO2007075234A1	05.07.2007	US2007143665A1	21.06.2007
		EP1969549A1	17.09.2008
		KR20080084971A	22.09.2008
		CN101331511A	24.12.2008
US2007074215A1	29.03.2007	无	
CN101197827A	11.06.2008	无	