

ICS 33.040.30

M 12



中华人民共和国通信行业标准

YD/T 2299-2011

XML 配置接入协议(XCAP)技术要求

Technical requirement for XML configuration access protocol (XCAP)

2011-12-10 发布

2011-12-10 实施

中华人民共和国工业和信息化部 发布

目 次

前 言.....II

1 范围.....1

2 规范性引用文件.....1

3 缩略语.....1

4 XCAP 协议概述.....1

 4.1 XCAP 协议机制.....1

 4.2 XCAP 协议栈架构.....2

5 XCAP 协议构成.....2

 5.1 协议结构.....2

 5.2 URI 构成.....2

6 XCAP 基本流程.....4

 6.1 正常流程.....4

 6.2 异常流程.....7

 6.3 安全性要求.....8

附录 A（资料性附录） 错误信息 XML 格式.....9

前 言

本标准按照GB 1.1-2009给出的规则起草。

本标准由中国通信标准化协会提出并归口。

本标准起草单位：中国电信集团公司、工业和信息化部电信研究院、华为技术有限公司、中兴通讯股份有限公司、上海贝尔股份有限公司。

本标准主要起草人：陆 钢、李惠云、张鹏生、吴宏建、刘文字、万志坤。

XML 配置接入协议（XCAP）技术要求

1 范围

本标准规定了XCAP协议框架，描述其基本能力，包括XCAP协议的原则性要求以及一些基本操作、流程、参数的描述。

本标准适用于使用XCAP协议的通信设备与接口。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

IETF RFC 4825	XML 配置接入协议（The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)
W3C XML	可扩展标记语言（Extensible Markup Language (XML)）

3 缩略语

下列缩略语适用于本文件。

AUID	Application Unique ID	应用唯一标识
HTTP	Hyper Text Transfer Protocol	超文本传输协议
OMA	Open Mobile Alliance	开放移动联盟
SIP	Session Initiation Protocol	会话初始协议
URI	Uniform Resource Identifier	统一资源标识符
XCAP	XML Configuration Access Protocol	XML配置接入协议
XCAPC	XCAP Client	XCAP客户端（简称客户端）
XCAPS	XCAP Server	XCAP服务器（简称服务器）
XDM	XML Document Management	XML文档管理
XML	eXtensible Markup Language	扩展标记语言
XUI	XCAP User Identifier	XCAP用户标识

4 XCAP 协议概述

4.1 XCAP 协议机制

XCAP协议全称是XML配置访问协议（XML Configuration Access Protocol）。其中XML是一套定义语义标记的规则，标记将文档分成多个部件并对这些部件加以标识。XML也是元标记语言，即定义了用于定义其他与特定领域有关的、语义的、结构化的标记语言的句法语言。利用XML可以定义多种信息表示的方式，经常应用于某种信息的格式定义，以保证通信的双方可以方便的传送、解析信息内容，XML应符合的要求由W3C XML给出。

XCAP协议是采用了XML来定义配置信息的描述格式，并且采用HTTP消息作为配置信息相关操作的

承载协议。XCAP结合了XML信息描述方式和HTTP协议，根据业务需求对HTTP协议部分参数进行规定，以实现在分布式通信系统中远程访问配置信息的目的。

4.2 XCAP 协议栈架构

如前所述，XCAP是基于HTTP协议承载XML消息体，用于各类业务XCAP客户端访问其XCAP服务器的相关信息。如图1所示。

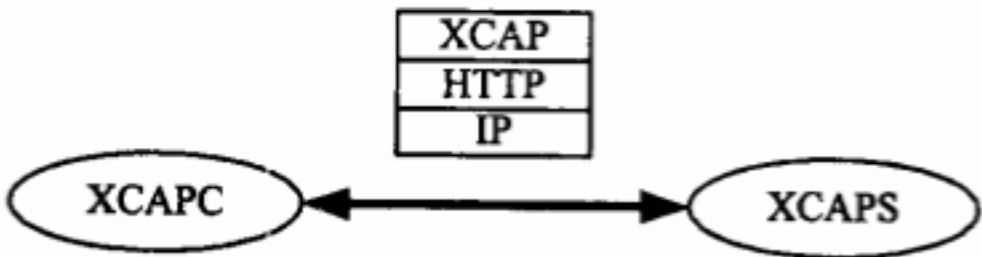


图 1 XCAP 协议栈架构

5 XCAP 协议构成

5.1 协议结构

XCAP协议的核心是XML文档。XML最基本的概念是：结构化标记数据，实现了“文档结构化”的语言规范，即采用DTD(Document Type Definition)文档类型定义的语言标准。XML文档表现如图2所示形式。

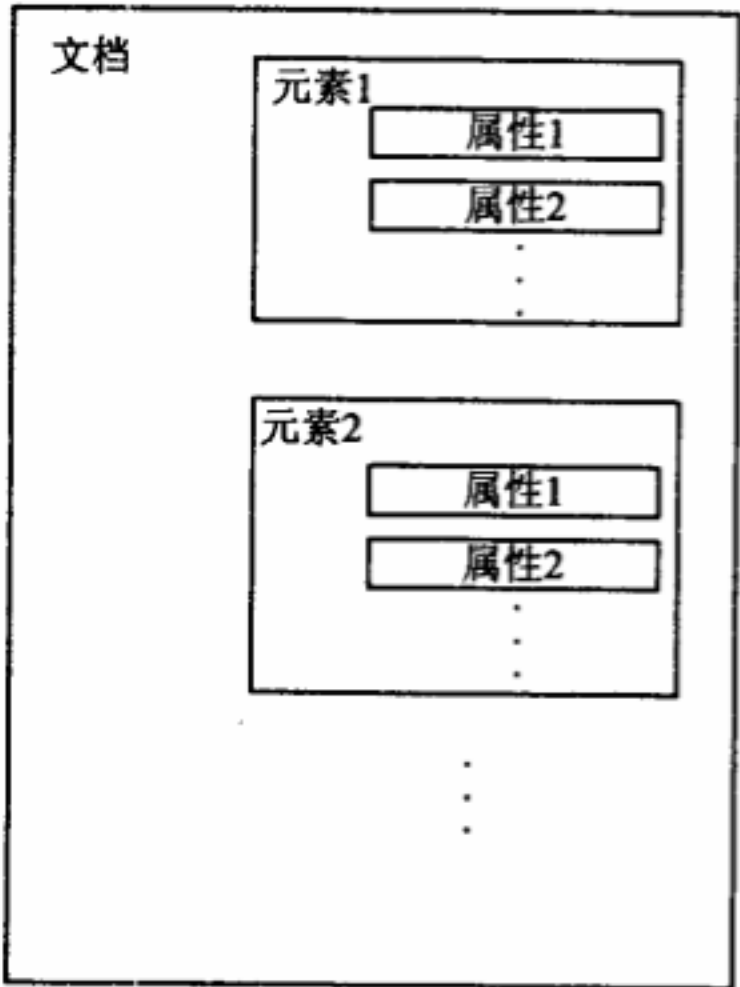


图 2 XML 文档结构

5.2 URI 构成

5.2.1 应用唯一标识（AUID）

XCAP协议为使用XCAP协议的每一个应用规定了唯一的应用标识，此为应用标识明确了该应用的命名空间和应用语法。运营商可以为自己的XCAP应用分配应用唯一标识。应用唯一标识的ABNF语法如下：

```
AUID           = global-auid / vendor-auid
global-auid    = auid
auid           = 1*auid-char
vendor-auid    = rev-hostname "." auid
rev-hostname   = toplabel *( "." domainlabel )
domainlabel   = alphanum
               / alphanum *( alphanum / "-" ) alphanum
```


toplabel = ALPHA / ALPHA *(alphanum / "-") alphanum
 auid-char = auid-unreserved / pct-encoded / sub-delims
 / ":" / "@"
 auid-unreserved = ALPHA / DIGIT / "-" / "_" / "~"

5.2.2 XCAP 根节点 (XCAP Root)

XCAP根节点描述了资源所在的位置，是XML文档存储位置的域名。一般推荐描述形式为“xcap.domain”，例如中国电信规定其XCAP根节点为“xcap.chinatelecom.com.cn”。

5.2.3 文档选择器 (Document Selector)

在文档根节点下的每一个XML文档使用文档选择器唯一标识，文档选择器类似于由多个路径标识符组成的一个完整路径，使用“/”作为路径分隔符。Document Selector的ABNF语法描述如下：

document-selector = AUID"/"doc-type
 doc-type = user-doc/global-doc
 user-doc = "users"/"[XUI]/home-directory
 global-doc = "global"/"[XUI]/home-directory

其中 XUI 为 XCAP 的客户端标识。Home-directory 为自定义的目录文件。

5.2.4 节点选择器 (Node Selector)

节点选择器用于寻址到被访问XML文档的具体元素或元素属性，其ABNF语法描述如下：

node-selector = element-selector ["/" terminal-selector]
 terminal-selector = attribute-selector / namespace-selector /
 extension-selector
 element-selector = step *("/" step)
 step = by-name / by-pos / by-attr / by-pos-attr /
 extension-selector
 by-name = NameorAny
 by-pos = NameorAny "[" position "]"
 position = 1 *DIGIT
 attr-test = ("@" att-name "=" <"> att-value <">) /
 ("@" att-name "=" <> att-value <>)
 by-attr = NameorAny "[" attr-test "]"
 by-pos-attr = NameorAny "[" position "]" "[" attr-test "]"
 NameorAny = QName / "*"; QName from XML Namespaces
 att-name = QName
 att-value = AttValue; from XML specification
 attribute-selector = "@" att-name
 namespace-selector = "namespace::*"
 extension-selector = 1*(%x00-2e / %x30-ff) ; anything but "/"

6 XCAP 基本流程

注：以下流程参考 IETF RFC 4825。

6.1 正常流程

XCAP为客户端提供以下基本操作：建立新文档、替换现有文档，删除现有文档、读取文档，在现有文档中建立新元素，替换文档中现有元素，删除文档中现有元素，读取文档中的元素，为文档中已有元素建立属性，替换文档中的属性，删除文档中的属性及获取文档中的某个属性。

6.1.1 建立、替换文档

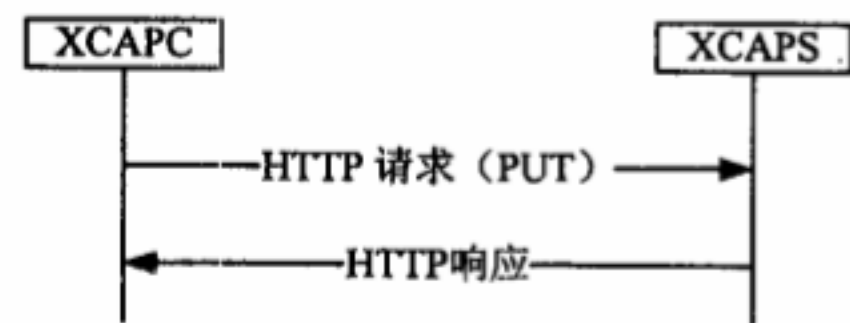


图 3 建立替换文档流程

要创建或替换文件，需要构造一个客户端的URI描述文件放置的位置。此URI 应是一个文件的URI，包含XCAP根 文件选择，然后客户端调用该URI的PUT方法。

如果文档不存在则创建一个新文档，如果文档已经存在则覆盖原有文档。

6.1.2 删除文档

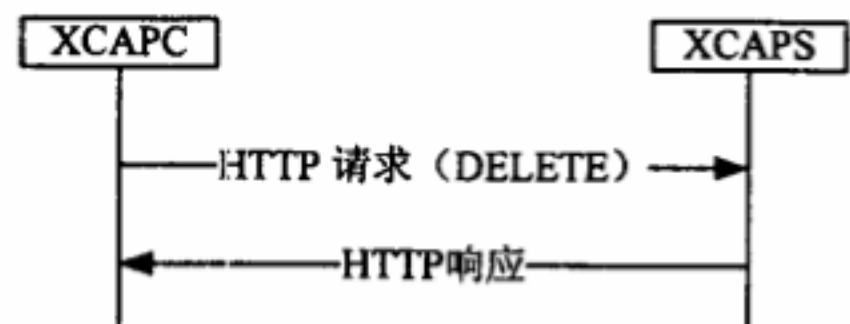


图 4 删除文档流程

客户端构造一个URI引用描述要删除的文件放置的位置，这个URI应是一个文件的URI，客户端然后调用该URI的DELETE操作删除的文件。

6.1.3 读取文档

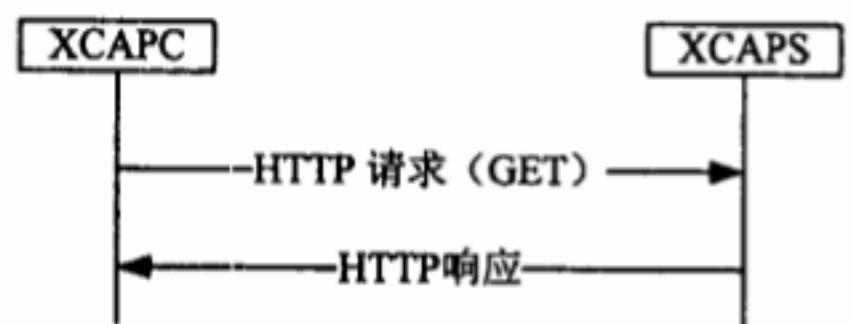


图 5 读取文档流程

客户端构造一个URI引用描述要读取的文件放置的位置，然后调用该URI的GET操作读取文件。

6.1.4 建立、替换元素

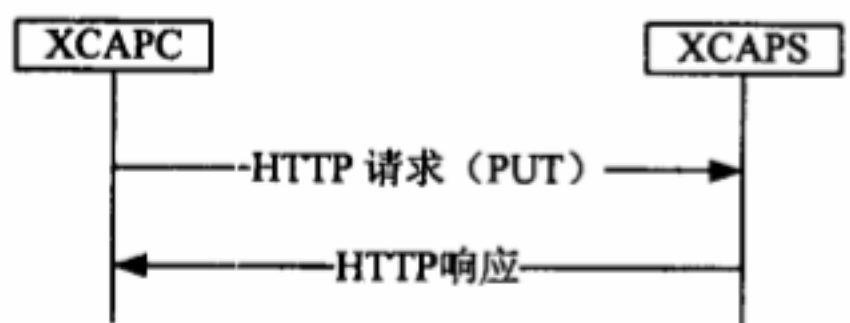


图 6 建立、替换元素流程

要替换一个文档中的XML元素，客户端需要构造URI，其文档选择器指向需要修改的文档，节点选择器指向需要修改的元素。

要创建一个文档中的XML元素，客户端需要构造URI，其文档选择器指向需要修改的文档，节点选

择器指向新增加的元素位置，该位置和现有元素是不匹配的。

通常情况下客户端希望在某个指定位置插入一个元素，此元素和其他元素同属于一个父节点元素，此类操作被称为位置插入，对于此类操作，客户端需要构造一个节点选择器，结构为“parent/*[position][unique-attribute-value]”，“parent”描述待插入元素的父节点，“position”描述元素待插入的位置，“unique-attribute-value”描述待插入元素的属性。

6.1.5 删除元素



图 7 删除元素流程

要删除一个文档中的XML元素，客户端需要构造URI，其文档选择器指向需要删除元素的文档，节点选择器指向需要删除的元素。

如果客户端希望在某个指定位置删除元素，此类操作被称为位置删除。对于此类操作，客户端需要构造一个节点选择器，结构为“parent/*[position][unique-attribute-value]”。其中，“parent”描述待删除元素的父节点，“position”描述待删除元素的位置，“unique-attribute-value”描述待删除元素的属性。

当文档服务器收到客户端构造的删除请求后，将从文档中删除指定元素，包括该元素的属性、子元素。

6.1.6 读取元素

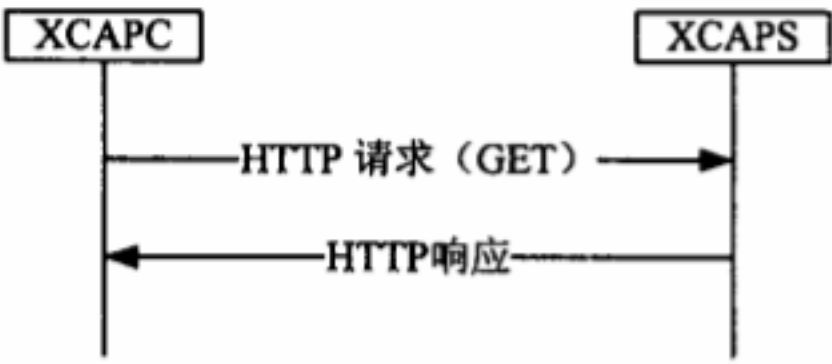


图 8 读取元素流程

要获取一个文档中的XML元素，客户端需要构造URI，其文档选择器指向需要获取元素的文档，节点选择器指向需要获取的元素。

客户端构造一个获取请求消息，服务器收到请求操作成功后，将会回复一个包括该元素XML描述的200 OK消息。

6.1.7 建立、替换属性

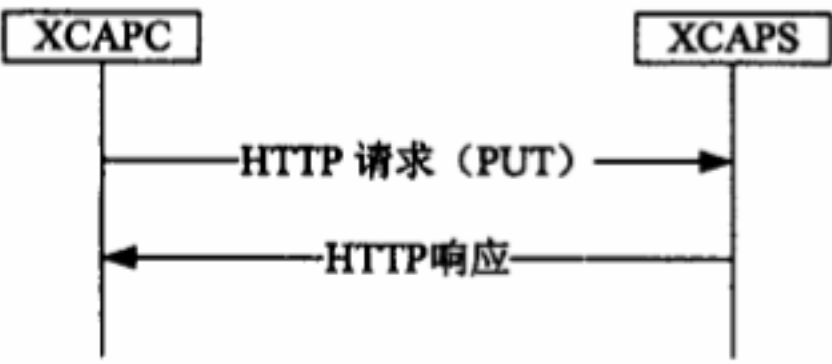


图 9 建立替换属性流程

要创建或替换一个文档中的XML元素，客户端需要构造URI，其文档选择器指向需要包含待创建或修改属性的文档，节点选择器指向需要创建或替换的元素属性，如果匹配则进行替换操作，如果不存在则进行创建操作。

客户端根据URI构造一个包含内容是属性值的HTTP PUT操作。

6.1.8 删除属性

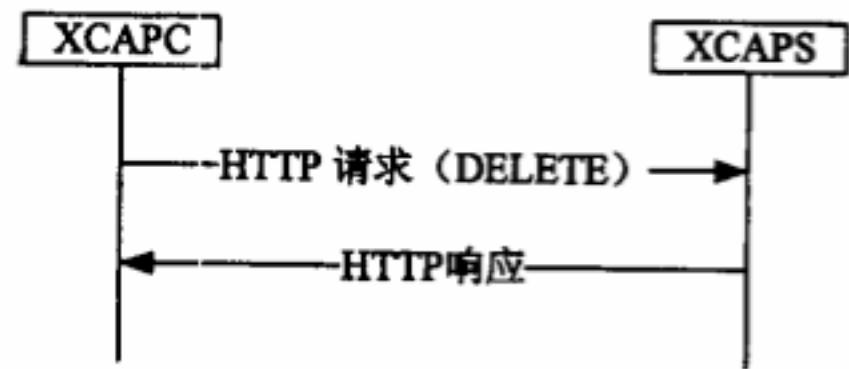


图 10 删除属性流程

要删除一个文档中的XML属性，客户端需要构造URI，其文档选择器指向需要删除属性的文档，节点选择器指向需要删除的元素属性。

当文档服务器收到客户端构造的删除请求后，将从文档中删除指定元素属性。

6.1.9 读取属性

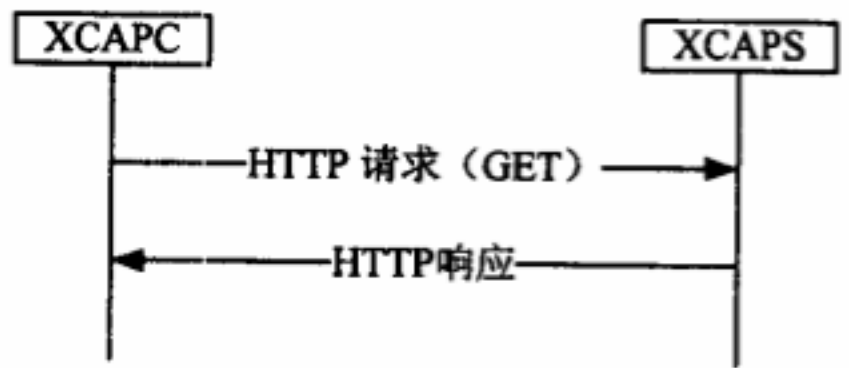


图 11 读取属性流程

要获取一个文档中的XML元素属性，客户端需要构造URI，其文档选择器指向需要获取属性的文档，节点选择器指向需要获取的元素属性。

客户端构造一个获取请求消息，服务器收到请求操作成功后，将会回复一个包括描述该属性 "application/xcap-att+xml" 文档的200 OK消息。

6.1.10 获取命名空间绑定

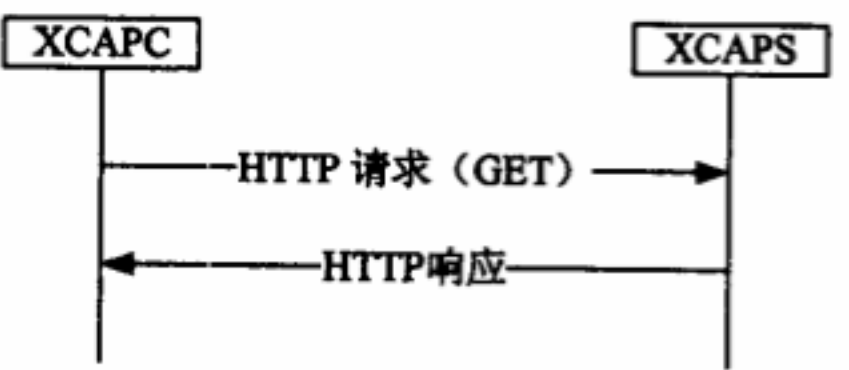


图 12 获取命名空间绑定流程

如果客户端希望在文档中插入一个元素或者属性，而此元素或者属性是文档中其他命名空间的一部分，客户端需要知道该命名空间以构造一个XML请求消息。如果客户端缓存了该文档，则无需该操作。如果客户端并没有缓存全部文档，在客户端进行下一步操作时则需要获取命名空间绑定。

要获取命名空间绑定，客户端需要构造URI，其文档选择器指向需要获取命名空间绑定的文档，节点选择器指向需要获取的命名空间绑定。客户端根据此URI发起一个GET请求，服务器会回复一个包括命名空间定义的 “application/xcap-ns+xml” 文档的响应消息。

6.1.11 可选操作

HTTP协议中定义了几个头域，客户端可以根据这些头域来控制可选的操作过程。“If-None-Match” 和 “If-Match” 头域允许客户端根据其取值来判断下一步如何操作。

当客户端需要增加、修改、删除某个文档元素或属性时，需要在HTTP请求消息带上 “If-Match” Etag

标签（该标签是上次操作文档时，服务器返回的取值，表示目前文档版本）。服务器收到该请求后判断该标签和目前服务器文档版本标签是否一致，如果一致则操作成功，如果不一致，则返回错误消息给客户端，客户端根据此结果发起新的获取文档请求消息，以获取最新版本的文档。

当客户端需要获取某个文档的最新版本时，为了减少网络传输流量，可以利用“**If-No-Match**” Etag 标签携带自己缓存文档的最新版本号，然后发起一个HTTP GET请求。服务器收到该请求消息，将标签和服务器侧文档标签进行比较，如果不一致则返回200 OK效应消息，如果一致，服务器会返回304响应消息。

6.2 异常流程

异常流程如图13所示。

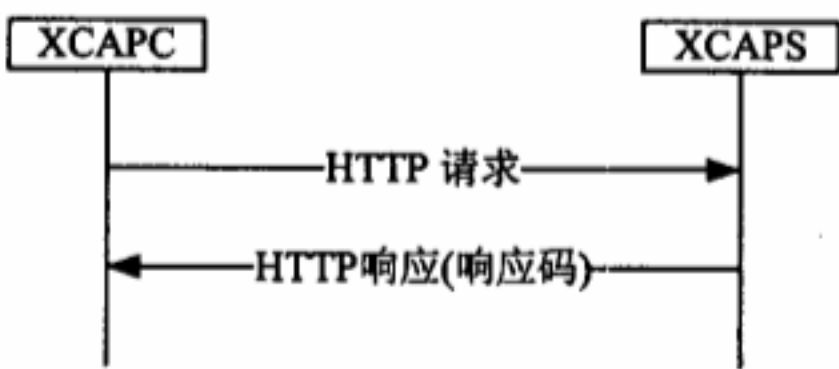


图 13 异常流程

6.2.1 412 响应码

当客户端需要增加、修改、删除某个文档元素或属性时，需要在HTTP请求消息带上“**If-Match**” Etag 标签（该标签是上次操作文档时，服务器返回的取值，表示目前文档版本）。服务器收到该请求后判断该标签和目前服务器文档版本标签是否一致，如果不一致，则返回412错误消息给客户端，客户端根据此结果发起新的获取文档请求消息，以获取最新版本的文档。

6.2.2 404 响应码

如果服务器接收到一个请求消息，无法识别该请求消息中的URI，则服务器回复404错误响应。

如果服务器接收到一个请求消息，无法识别该请求消息中的XUI，则服务器回复404错误响应。

6.2.3 400 响应码

如果服务器接收到一个请求消息，请求消息中的节点选择器不符合schema定义要求，则服务器回复400错误响应。

6.2.4 405 响应码

如果接收到HTTP POST请求消息，或者HTTP PUT请求URI中包括命名空间选择器，则服务器拒绝该请求，回复405错误响应。

6.2.5 409 响应码

如果接收到HTTP PUT请求是操作某个文档，消息体内容不是一个符合XML格式要求的文档，服务器回复409错误响应（包括<no-well-formed>元素）描述错误原因。

如果HTTP PUT请求是针对某个元素，如果消息体内容不符合XML格式要求，服务器回复409错误响应（包括<no-xml-frag>元素）描述错误原因。如果格式符合要求，但是字符不属于UTF-8字符集，则服务器回复409错误响应（包括<not-utf-8>元素）描述错误原因。

如果HTTP PUT请求是针对某个属性，如果消息体内容不符合属性语法定义，则服务器回复409错误响应（包括<not-xml-att-value>元素）描述错误原因。

如果HTTP PUT请求是需要插入或替换某个元素，但是节点选择器并没有正确指向，则服务器回复409错误响应（包括<cannot-insert>元素）描述错误原因。

如果HTTP PUT请求是需要插入或替换某个属性,但是节点选择器并没有正确指向,则服务器回复409错误响应(包括<cannot-insert>元素)描述错误原因。

如果HTTP PUT请求时需要插入一个文档、元素或者属性,服务器会判断该文档是否符合Schema定义,如果不符合,则服务器回复409错误响应(包括<schema-validation-error>元素)描述错误原因。

如果HTTP PUT请求时需要插入一个元素或者属性,服务器会判断该插入操作是否符合唯一性要求,如果不符合,则服务器回复409错误响应(包括<uniqueness-failure>元素)描述错误原因。

如果HTTP PUT请求时需要插入一个元素或者属性,服务器会判断URI和其他数据约束是否符合要求,如果不符合,则服务器回复409错误响应(包括<constraint-failure>元素)描述错误原因。

如果HTTP DELETE请求需要删除一个元素或属性,如果删除后,该请求的URI指向了另外一个元素或属性,则该删除请求不被执行,并且服务器回复409错误响应(包括<cannot-delete>元素)描述错误原因。

如果请求消息请求插入节点没有父节点,导致xml文档不完整,则服务器拒绝该请求,回复409错误响应,并且在该响应消息(包括<no-parent>元素)中详细描述冲突原因。

对于409错误响应,服务器可以在消息体中详细描述错误原因,其MIME类型为“application/xcap-erroot+xml”,可参考附录A的要求。

6.2.6 415 响应码

如果请求消息中的MIME类型不等于应用语法要求的MIME类型,则服务器拒绝该请求,回复415错误响应。

6.3 安全性要求

参考RFC4825,建议XCAP操作采用HTTPS URI作为XCAP root URI,在客户端和服务器之间使用TLS封装方式,采用HTTP Digest方式鉴权。

附 录 A
(资料性附录)
错误信息XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:xcap-error"
  xmlns="urn:ietf:params:xml:ns:xcap-error"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:element name="error-element" abstract="true"/>

  <xs:element name="xcap-error">
    <xs:annotation>
      <xs:documentation>Indicates the reason for the error.
    </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="error-element"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="extension" substitutionGroup="error-element">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##any" processContents="lax"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="schema-validation-error"
    substitutionGroup="error-element">
    <xs:annotation>
```

`<xs:documentation>`This element indicates that the document was not compliant to the schema after the requested operation was performed.`</xs:documentation>`

`</xs:annotation>`

`<xs:complexType>`

`<xs:attribute name="phrase" type="xs:string" use="optional"/>`

`</xs:complexType>`

`</xs:element>`

`<xs:element name="not-xml-frag" substitutionGroup="error-element">`

`<xs:annotation>`

`<xs:documentation>`This indicates that the request was supposed to contain a valid XML fragment body, but did not.`</xs:documentation>`

`</xs:annotation>`

`<xs:complexType>`

`<xs:attribute name="phrase" type="xs:string" use="optional"/>`

`</xs:complexType>`

`</xs:element>`

`<xs:element name="no-parent" substitutionGroup="error-element">`

`<xs:annotation>`

`<xs:documentation>`This indicates that an attempt to insert an element, attribute or document failed because the document or element into which the insertion was supposed to occur does not exist`</xs:documentation>`

`</xs:annotation>`

`<xs:complexType>`

`<xs:sequence>`

`<xs:element name="ancestor" type="xs:anyURI" minOccurs="0">`

`<xs:annotation>`

`<xs:documentation>`Contains an HTTP URI that points to the element which is the closest ancestor that does exist.

`</xs:documentation>`

`</xs:annotation>`

`</xs:element>`

`</xs:sequence>`

`<xs:attribute name="phrase" type="xs:string" use="optional"/>`


```

</xs:complexType>
</xs:element>

```

```

<xs:element name="cannot-insert" substitutionGroup="error-element">
  <xs:annotation>
    <xs:documentation>This indicates that the requested
PUT operation could not be performed because a GET of that resource
after the PUT would not yield the content of the PUT request.
  </xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:attribute name="phrase" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>

```

```

<xs:element name="not-xml-att-value"
substitutionGroup="error-element">
  <xs:annotation>
    <xs:documentation>This indicates that the
request was supposed to contain a valid XML attribute value, but did
not.</xs:documentation>
  </xs:annotation>
<xs:complexType>
  <xs:attribute name="phrase" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>

```

```

<xs:element name="uniqueness-failure"
substitutionGroup="error-element">
  <xs:annotation>
    <xs:documentation>This indicates that the
requested operation would result in a document that did not meet a
uniqueness constraint defined by the application usage.
  </xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>

```

```

<xs:element name="exists" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>For each URI,
element or attribute specified by the client which is not unique,
one of these is present.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence minOccurs="0">
      <xs:element name="alt-value" type="xs:string"
maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>An optional set of alternate values can be
provided.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="field" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="not-well-formed"
substitutionGroup="error-element">
  <xs:annotation>
    <xs:documentation>This indicates that the body of the request was
not a well-formed document.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="phrase" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>

<xs:element name="constraint-failure"
substitutionGroup="error-element">

```

```
<xs:annotation>
```

```
  <xs:documentation>This indicates that the
requested operation would result in a document that failed a data
constraint defined by the application usage, but not enforced by the
schema or a uniqueness constraint.</xs:documentation>
```

```
</xs:annotation>
```

```
<xs:complexType>
```

```
  <xs:attribute name="phrase" type="xs:string" use="optional"/>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<xs:element name="cannot-delete" substitutionGroup="error-element">
```

```
  <xs:annotation>
```

```
    <xs:documentation>This indicates that the requested DELETE
operation could not be performed because it would not be
idempotent.</xs:documentation>
```

```
  </xs:annotation>
```

```
  <xs:complexType>
```

```
    <xs:attribute name="phrase" type="xs:string" use="optional"/>
```

```
  </xs:complexType>
```

```
</xs:element>
```

```
<xs:element name="not-utf-8" substitutionGroup="error-element">
```

```
  <xs:annotation>
```

```
    <xs:documentation>This indicates that request could not be
      completed because it would have produced a document not
      encoded in UTF-8.</xs:documentation>
```

```
  </xs:annotation>
```

```
  <xs:complexType>
```

```
    <xs:attribute name="phrase" type="xs:string" use="optional"/>
```

```
  </xs:complexType>
```

```
</xs:element>
```

```
</xs:schema>
```

服务器能力 XML 格式

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

```
<xs:schema targetNamespace="urn:ietf:xml:params:ns:xcap-caps"
```

```
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```

xmlns="urn:ietf:xml:params:ns:xcap-caps"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:element name="xcap-caps">
  <xs:annotation>
    <xs:documentation>Root element for xcap-caps</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="auids">
        <xs:annotation>
          <xs:documentation>List of supported AUID.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="auid" type="auidType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="extensions">
        <xs:annotation>
          <xs:documentation>List of supported extensions.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="extension" type="extensionType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="namespaces">
        <xs:annotation>
          <xs:documentation>List of supported namespaces.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="namespace" type="namespaceType"/>
          </xs:sequence>

```

```

    </xs:complexType>
  </xs:element>
  <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="audType">
  <xs:annotation>
    <xs:documentation>AUID Type</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="extensionType">
  <xs:annotation>
    <xs:documentation>Extension Type</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="namespaceType">
  <xs:annotation>
    <xs:documentation>Namespace type</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
</xs:schema>

```

中华人民共和国
通信行业标准
XML 配置接入协议(XCAP)技术要求
YD/T 2299-2011

*

人民邮电出版社出版发行
北京市崇文区夕照寺街 14 号 A 座
邮政编码: 100061
宝隆元(北京)印刷技术有限公司印刷
版权所有 不得翻印

*

开本: 880 × 1230 1/16 2012 年 1 月第 1 版
印张: 1.5 2012 年 1 月北京第 1 次印刷
字数: 34 千字

ISBN 978 - 7 - 115 - 2338/ 11 - 289

定价: 15 元

本书如有印装质量问题, 请与本社联系 电话: (010)67114922