

## 第13章 无线电话应用规范

### 13.1 范围

无线应用协议 ( Wireless Application Protocol , WAP ) 是WAP论坛经过不断努力得到的成果, 它提供了一个业界技术规范, 以便开发出适用于各种无线通信网络的应用和业务。 WAP论坛的工作范围就是为各种业务和应用制定一系列的技术规范。无线市场正在快速增长, 新的用户不断增多, 新的业务不断涌现。为了给运营商和生产者提供一个面对先进业务、多种类业务和快速灵活业务生成的商机, WAP制定了一系列传输层、会话层和应用层协议。有关WAP体系结构更多的信息, 请参阅“无线应用协议体系结构规范” (Wireless Application Protocol Architecture Specification) [WAP]。

本规范定义了无线电话应用 ( Wireless Telephony Application , WTA )。 WTA用户代理是WAE用户代理的扩展, 它使用 WTA接口[WTAI]。 WTA 用来定义各种无线电话应用, 这些应用是本地系统与底层电话网络基础设施之间的接口。

WTA 接口有三种不同的类型: 前两种类型——通用网络类型和特殊网络类型是专供网络的运营商使用, 因为移动网络的运营商需要用它们来控制和管理为用户提供的服务; 第三种类型是公共的, 它由一组 WTA函数构成。例如, 对一个移动电话呼叫进行初始化的函数, 可供内容的开发者生成内容时使用。

无线电话应用有以下四个主要目标:

先进的终端用户服务 WTA使网络的运营商能够向终端用户提供统一接口的先进服务。

提高网络的利用率 网络运营商利用WTA可以增加网络服务的用途。

协同工作 用WTA接口编写的各种WAP应用可以运行在各种电话设备上, 协同工作的能力不受终端类型的限制。

与网络无关的应用 WTA内容开发者编写的电话应用可以在使用不同的协议的各种网络上运行。

### 13.2 WTA概述

WTA使WML和WMLScript编写的内容可以利用设备和移动网的电话特性。 WTA服务器、移动网络和WTA客户端可以被看作是一种客户端——WTA服务器的应用。 WTA服务器作为主要的内容生成器, 被连接到网络服务可控的移动网络上。内容由 WTA服务器定制, 并可以下载到客户端。

WTA从三个方面扩展了基本的WAE应用:

- WTA为WTA服务器提供了将内容“推”到设备的手段。
- WTA为移动网络事件提供了触发设备显示内容的方法。
- WTA提供了可以通过WML或WMLScript访问的电话函数, 这些函数位于设备上。

WTA服务器使用WSP中的PUSH特性[WSP], 将内容“推”到WTA客户端。原则上讲,

内容可以采取任意的表示形式，但实际上存在着两种根本不同的内容格式：标准的 WAE 内容格式（如 WML、WMLScript 或 WBMP）和特殊的 WTA 内容格式，称为 WTA 事件。这种 WTA 框架使客户端的实现非常灵活，只要客户端能够支持被推标准内容或者 WTA 事件（或者两者都支持）就可以了。WTA 服务器要根据用户代理的特性和描述来决定具体的操作。

### 13.2.1 以服务器为中心和以客户端为中心的模式

“推” WTA 事件和“推”内容的根本差异见图 13-1 和图 13-2。在命名这个特殊的模型时，用到了相关的网络成员其中，两个重要的网络成员是 WTA 服务器和 WTA 客户端。在这两个模型中，内容都是由 WTA 服务器生成，再传送到 WTA 客户端。

以服务器为中心的模式（图 13-1）：WTA 服务器处理移动网络中发生的事件（如来话呼叫），并向客户端提供内容，以处理特定的任务，它会根据发生的事件来设置内容。

以客户端为中心的模式（图 13-2）：WTA 客户端处理从移动网络中接收到的事件（如来话呼叫）。WTA 客户端上的事件绑定，可以将接收到的事件与处理特定任务的可执行内容联系在一起。

在同一个网络结构中，把以服务器为中心的模式与以客户端为中心的模式相结合，有利于运营商针对不同的移动网络和应用选择最合适的模式。

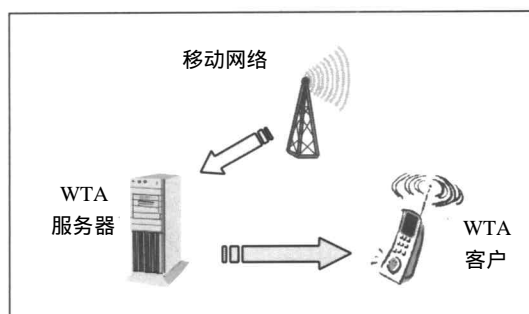


图13-1 以服务器为中心的模式

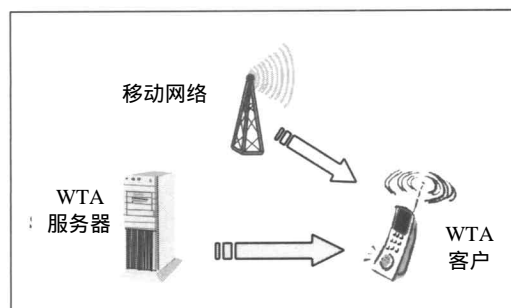


图13-2 以客户端为中心的模式

### 13.2.2 用户代理特征

WAP 代理会被告知可以使用的 WTA 特性，如可用的 WTAI 库 [WTAI] 及选中的 WTA 网络编

程模型（即，是以服务器为中心还是以客户端为中心的模式）。

### 13.2.3 WTAI 函数调用

提供了访问设备和移动网络特性的函数 [WTAI]。设备特性的一个例子是呼叫控制的逻辑指示，移动网络特性则包括呼叫控制和网络文本。

### 13.2.4 WTA事件

图13-3描述了WTA的事件转换。WTA的事件转换是一个过程，它的用户代理首先把网络事件转换为WTA事件，然后再转换为由客户端采取的相应操作（例如，把一个来话呼叫事件转换成振铃信号）。这样，根据WTA用户代理的状态和当前的WTA上下文（如果有的话），WTA事件与对应的WTA事件处理相匹配。

图13-3还解释了客户端接收WTA事件的两种方法：第一种方法（用后缀“A”表示）是由设备的操作系统接收，当操作系统检测到一个事件之后，将该事件传给WAE的基本服务；第二种方法（用后缀“B”表示）是通过从WTA服务器接收一个“推”过来的WTA事件。网络事件（见[WTAI]规范）会被转换成抽象的WTA事件来表示，如表13-1所示。

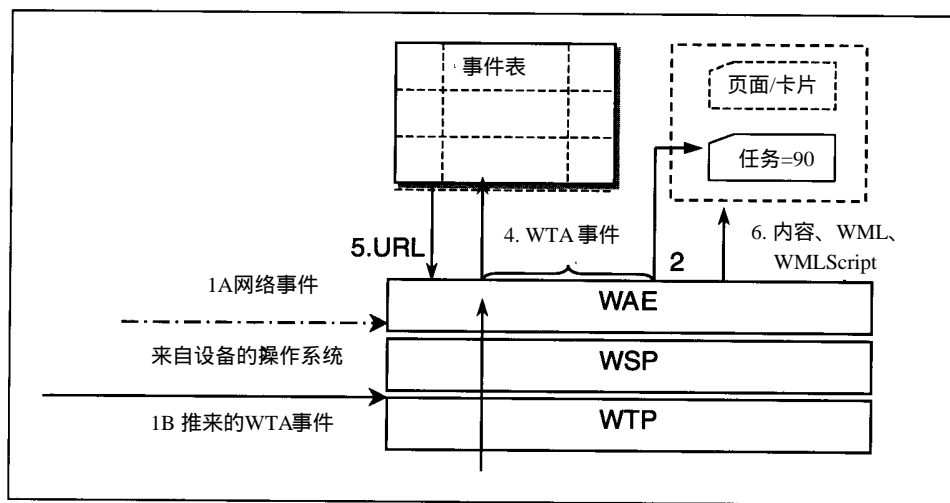


图13-3 WTA事件的转换

表13-1 事件流程的排列

1A	设备的操作系统接收到一个网络事件，把它转换成WTA事件，并将它插入到WAE中
1B	使用WSP的“推”操作，从WTA服务器接收到一个WTA事件的内容格式
2	当存在一个用户代理的上下文时，WTA事件必须首先与现有的WML事件绑定相匹配
3	在WML事件绑定参数task=go的情况下，必须被装入由URL指定的资源
4	如果与WML事件绑定不匹配，意味着必须检测WTA事件，以确定事件表中是否存在一个匹配
5	如果在事件表中找到一个匹配，返回与之相关的URL，这个URL从WTA事件表绑定中抽取
6	对于一个新的上下文，WTA用户代理被重新启动，并根据URL指出的位置装入内容。任何随WTA事件接收到的数值都存储在由参数列表标识出来的变量中

存储在事件表中的一个URL可以是相对的（如，一个卡片名称），在这种情况下，基本的URL必须在事件表中提供。

### 13.3 支持的内容格式

这一节描述了在接收内容时，WTA客户端必须支持的内容格式，以及在不同的（以客户端为中心或是以服务器为中心）模式下，必须进行的操作。

下面描述了内容传送的三种基本方法：

- 移动网络事件 它是一种能够被处理的、网络信号的特殊表示形式。WTA用户代理遇到的网络事件，都是以WTA事件这种抽象的形式出现（抽象意味着不需要具体的WTA事件内容）。
- 确认的内容 必须向发送方传输确认信息的内容，例如事件表。
- 无确认的内容 无须向发送方传输确认信息的内容，例如WML页面。

图13-4给出了WTA客户端必须支持的内容格式，这与WTA客户端的工作模式，也就是以服务器为中心还是以客户端为中心有关。以服务器为中心模式支持的内容格式，也一定能被以客户端为中心的模式支持。

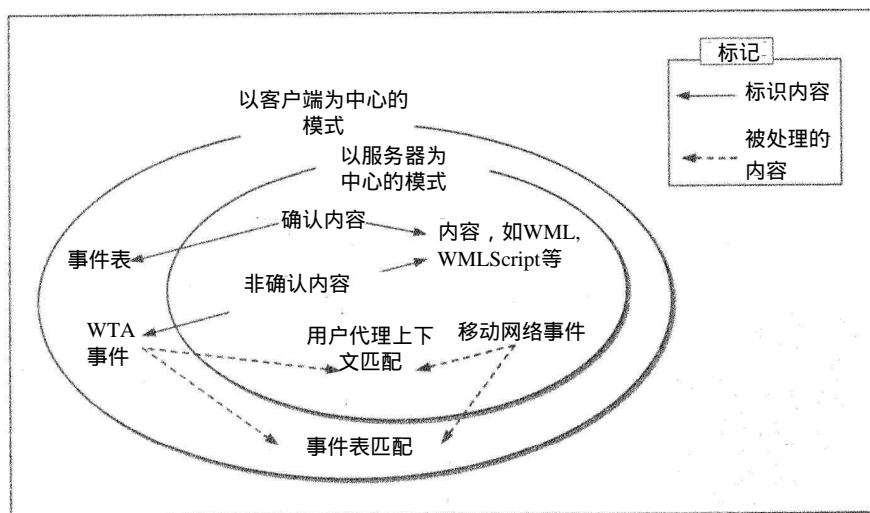


图13-4 WTA内容格式关系图

图13-4是在以客户端为中心和以服务器为中心的模式下，以支持的内容格式范围为基础建立的。以服务器为中心模式能支持的内容格式在以客户端为中心的模式中都能支持，反之则不然，后者能支持的事件表和WTA事件，前者则不能支持。

在图13-4中，内容传送的方法用黑体字表示，箭头表示用给定的方法能够传送的内容格式，斜体字表示了发生在内容格式上的转换过程，这些内容格式是从一些事件（如WTA事件或移动网络事件）转换过来的。这里只给出了网络事件的逻辑框图，实际上，在用户代理或事件表处理之前，所有的网络事件都必须先抽象成WTA事件。

#### 13.3.1 性能要求

在这一节中有两个术语：“立即”（immediately）和“无回应”（unattended），这两个术语是指：在WTA客户端或WTA服务器上，一个给定的实现对响应实时性期望的描述。“立即”

是指用户代理必须忽略其他任何内容的执行过程，开始一个新的内容；“无回应”是指从 WTA 服务器的角度来看，在事先规定的时间内，没有收到来自 WTA 客户端的确认响应。超时限制由应用的承载网络决定。关于承载特性（比如，为等待 WTA 客户端传来的确认内容而设置的超时时间）的建议，请参阅 WAP 体系结构的文件[WAPARCH]。

### 13.3.2 内容的序列化

处理内容的模式必须能提供一个有序的处理过程，以确保能按接收的顺序处理所有的内容：只有在当前的内容被处理完成之后，才能继续处理其他的内容。一个 WTA 事件处理完成的标志是：一个事件被解析成了 URL 或者其他的形式，如通过 WML 事件绑定，与一个给定的任务绑在了一起，并已经开始执行；对于 WML 和 WMLScript 这样的内容，只要这些内容被装入了上下文并开始执行，就可以认为处理过程已经完成。

### 13.3.3 以服务器为中心的模式

在这个模式中，禁止 WTA 服务器向 WTA 客户端发送 WTA 事件和事件表，WTA 用户代理和 WTA 服务器要遵循下面的原则。

#### 1. 处理移动网络事件时采取的步骤

- 1) 在接收到一个网络事件并将其转换成一个 WTA 事件后，要立即为这个网络事件提供服务。
- 2) 如果存在一个用户代理上下文，必须进行“事件匹配”，目的是在用户代理的上下文中找到一个 WML 事件绑定。例如，与 WML 元素 ONEVENT 相匹配的事件名。

#### 2. 处理有确认内容时采取的步骤

- 1) WTA 用户代理必须立即为这个内容提供服务。

2) 若 WTA 用户代理能识别这个内容，必须向 WTA 服务器发送表示“成功”的确认信息。

3) 若 WTA 用户代理不能识别内容，必须向 WTA 服务器发送表示“失败”的确认信息。

4) 若内容无回应，WTA 服务器必须像接收到表示“失败”的确认信息一样，进行处理。

#### 3. 处理无确认内容时，采取的步骤

- 1) 终端用户必须能够得知这个内容，并能够通过一定的途径检测到。如何通知终端用户的问题与具体的实现有关，WAP 没有规定。

2) 除非 WTA 用户代理中的上下文已经结束，否则不能处理这个内容。换句话说，新的内容不能打断用户代理正在执行的其他内容，但终端用户可以决定是结束用户代理中的当前内容和上下文，还是关注新的内容。

### 13.3.4 以客户端为中心的模式

13.3.3 节中给出的原则也同样适用于以客户端为中心的模式，以客户端为中心的模式也必须能够处理移动网络事件和像 WML、WMLScript 这样的内容。此外，以客户端为中心的模式要求 WTA 用户代理还能够支持 WTA 事件和事件表。

在以客户端为中心的模式中，WTA 客户端要遵循下面的原则。

#### 1. 处理事件表时采取的步骤

- 1) WTA 用户代理必须立即为事件表提供服务。

2) 若 WTA 用户代理成功地建立起事件表, 必须向 WTA 服务器发送表示“成功”的确认信息[WSP]。成功地建立起事件表意味着, 对于编码格式中需要的元素, 可以用事件表进行校验, 参见第 13.10 节。在送回一个确认的同时, 新建的事件表生效。

3) 若 WTA 用户代理由于收到一个移动网络事件或者由于校验失败而不能建立事件表, 必须向 WTA 服务器发送表示“失败”的确认信息[WSP]。

4) 若事件表的内容无回应, WTA 服务器必须像接收到“失败”的确认信息一样, 遵循内部原则进行相应的处理。WTA 服务器上所有的具体实现都有特定的内部规则。

## 2. 处理有确认 WTA 事件时采取的步骤

1) WTA 用户代理必须立即为 WTA 事件提供服务。

2) 事件匹配确定是否能够在已经存在的 WTA 用户代理的上下文中找到一个匹配。如果相对于编码表格式所需的元素, 经校验一个 WTA 事件(参见第 13.10 节)是正确的, 并且能找到事件绑定的一个匹配, 必须向 WTA 服务器发送一个表示“成功”的结果代码。

3) 如果在当前的 WTA 用户代理上下文中找不到匹配, 或者相对于编码格式中的所需的元素(参见第 10 节), 这个 WTA 事件不可能通过校验, 这时必须在事件表中进行事件匹配。若能够找一个匹配, 就必须向 WTA 服务器发送一个表示“成功”的结果代码。

4) 如果在事件表中不能找到匹配, 必须向 WTA 服务器发送一个表示“失败”的结果代码。

5) 若 WTA 用户代理由于收到一个移动网络事件或者由于校验失败而不能处理一个 WTA 事件, 必须向 WTA 服务器发送表示“失败”的结果代码。

6) 对于无回应(unattended)的 WTA 事件, WTA 服务器必须像接收到了“失败”信息一样, 遵循同样的原则进行处理。

## 3. 处理无确认的 WTA 事件时采取的步骤

1) WTA 用户代理必须立即为 WTA 事件提供服务。

2) 按照第 13.6 节描述的过程处理 WTA 事件。

## 13.4 WTA 的 URI 和 URL

承载网络和电话特性的多样性使得无线设备千差万别, 有许多标准(如 GSM)支持设备访问移动网络的电话特性。URI 形成了一个统一的命名模型, 用来标识与设备的内部结构和移动网络无关的特性。

在 WTA 中, URI 在下面的情况中使用:

- 在定义象逻辑指示这样的本地资源标识时 [WTAI]。
- 在定义象移动端发起的呼叫这样的电话特性时 [WTAI]。

### URL 的使用

WTA 用户代理建立在 WAE 标准的用户代理基础之上, 因此在定义和使用 URL 时, 二者遵循相同的规则[WML]。

在 WTA 中, URI 还用在下面的情况中:

- 当在 WTA 事件表中定义 URL 时。
- 当在 WTA 服务器上定义已知的资源(如事件表)时。



## 13.5 用户代理的状态模型

与标准的 WAE 用户代理一样，WTA 用户代理也提供对自身状态管理的支持。

WTA 用户代理能够对下面的自身状态进行管理：

与用户代理上下文有关的特定呼叫状态 在 WTA 用户代理的上下文有意被终止的情况下（诸如接收或建立通话这样的 WTAI 函数），必须与处理这个通话采取的行动有关。

与具体实现有关的状态 这些状态（例如，当前电源的存储状态、使用的内存及 SMS 的通信设置等）与用户代理的具体实现方法和行为有关。

### 13.5.1 用户代理上下文

WTA 用户代理建立在标准的 WAE 用户代理的基础之上，因此处理用户代理上下文遵循相同的原则。此外，当接收到了 WTA 的事件参数时，WTA 用户代理还能够对结束浏览器的上下文和生成隐含变量提供支持。

### 13.5.2 结束上下文

WTAI 的“退出”（Exit）函数 [WTAI] 表示必须结束用户代理上下文，这时用户代理必须完成如下的操作：

- 根据通话的模式，处理活动的通话。通话模式与上下文的生存周期有关 [WTAI]。
- 结束所有内容的处理，如结束对 WML 和 WMLScript 的处理。
- 删除浏览器的上下文，清除原先存储在这个上下文中的全部数据。

### 13.5.3 变量

WTA 用户代理建立在标准的 WAE 用户代理的基础之上，因此变量请求规则与标准的 WAE 用户代理相同。

在下列的情况中，WTA 用户代理还支持隐含变量的生成：

- 若一个 WTA 事件与携带着一个 WTA 事件绑定的用户代理上下文捆绑在一起，随这个事件接收到的选中参数能被参量化，用这些参数可以实现对用户代理上下文的访问（见 10.6.3 节）。
- 若一个 WTA 事件以 WTA 事件表绑定的格式与事件表相匹配，随这个 WTA 事件接收到的选中参数可以被参量化，用这些参数可以实现对用户代理上下文的访问（见 10.6.2 节）。

## 13.6 WTA 事件和导航

WTA 事件包含着存储在当前的 WTA 上下文中的数据，这样做的过程被称为 WTA 事件绑定，如何进行绑定要视 WTA 用户代理的情况而定。

当处理 WTA 事件时，有三种截然不同的状态：

用户代理上下文与一个 WTA 事件相匹配 用户代理上下文中存在与 WTA 事件相匹配的事件绑定。

没有有效的用户代理上下文或 WTA 事件匹配 WTA 事件表中存在与 WTA 事件相匹配的 WTA 事件绑定。

既没有有效的用户代理上下文、也没有与事件表绑定相配的 WTA 事件匹配 如果这个 WTA 事件是从移动网事件转换过来的，它会被转发给设备的操作系统。

### 13.6.1 事件处理器的优先顺序

下面的步骤必须按给定的顺序进行：

**用户代理上下文** 如果存在用户代理的上下文，就要利用事件匹配来确定这个 WTA 事件是否与用户代理上下文中的一个事件绑定相匹配。

**事件表** 如果存在事件表，就要利用事件匹配来确定这个 WTA 事件是否与事件表中的一个事件绑定相匹配。

**设备操作系统** 从移动网事件转换过来的 WTA 事件必须继续转发给设备的操作系统。然而，WAP 不要求设备的操作系统必须处理这个事件。

### 13.6.2 WTA 事件表

#### 1. 编制事件表

客户端能够用事件表进行编程，以响应从移动网或 WTA 服务器传来的事件。接收到的 WTA 事件要转换成可以在 WTA 用户代理上运行的 URL。利用 WTA 事件的内容可以编制 WTA 事件表，在以客户端为中心的模式中，WTA 事件表使用从 WTA 服务器“推”过来的事件内容进行编程，并且在 WTA 会话建立的初期被请求（见图 13-5）。

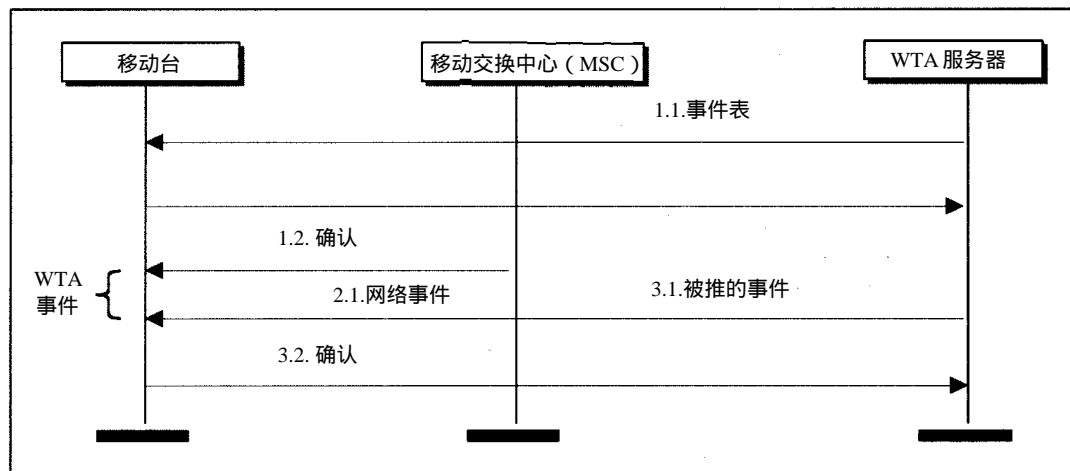


图13-5 事件表编程

示例1 服务器“推”一个将要由 WTA 客户端建立的事件表。

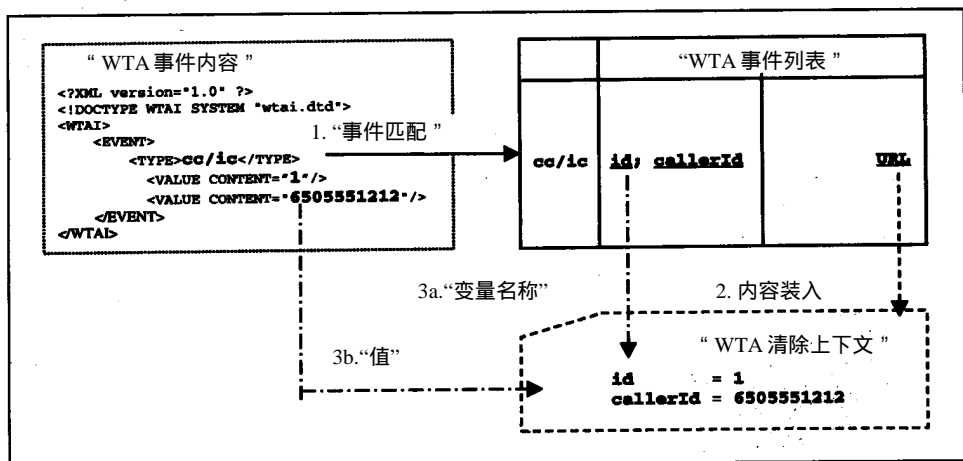
示例2 网络事件（或消息）产生。在这种情况下，WTA 客户端将实际的网络事件转换成较常见的 WTA 事件。如果事件表中存在事件绑定，所选中的参数都存储在新的用户代理上下文中，然后调用相关的 URL。

示例3 服务器下“推”一个 WTA 事件。对这个事件的处理方法与从网络事件转换过来的事件处理方法相同。

#### 2. WTA 事件表绑定



- 查找事件表，寻找与这个 WTA 事件相匹配的事件绑定。
- 若找到一个匹配，用户代理就调用存储在事件表中的给定 URL，将 WTA 内容装入一个“干净”上下文中。这里的“干净”是指在这个用户代理的上下文中没有事先设定的变量或状态（见图 13-6）。
- WTA 事件表中包括一些变量名，这些变量要由 WTA 事件携带的参数来赋值。



### 13.6.3 在用户代理上下文中的WTA事件绑定

当使用用户上下文的事件绑定时，要遵循下面的原则（见图 13-7）：

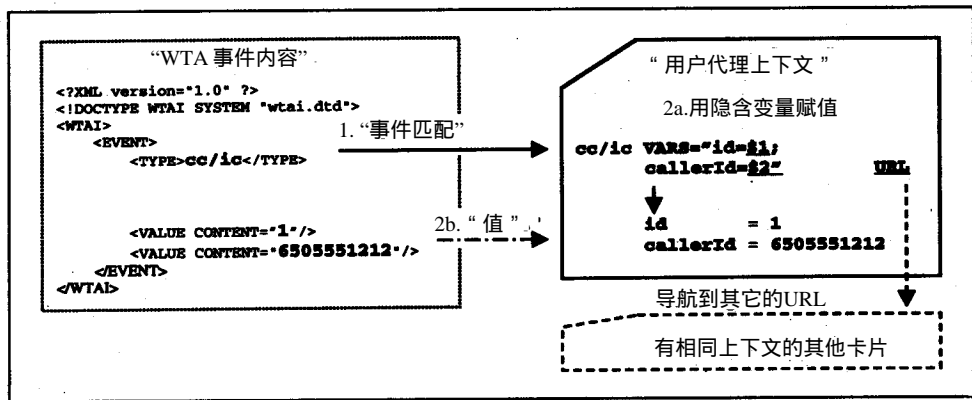


图13-7 将事件数据绑定到用户代理上下文中

- 为了与特殊的WTA事件相匹配，必须考虑存储在WTA用户代理上下文中的事件绑定。
- 随WTA事件接收到的选中参数必须能以隐含生成的变量名字的形式，被用户代理的上下文使用。
- WTA用户代理的上下文必须能把这些隐含变量赋给其他的变量，或者在调用URL时将隐含变量作为参数。

## 13.7 用户代理语义

WTA用户代理建立在标准的WML用户代理基础之上，它也使用WML和WMLScript语言。WAE为WTA用户代理识别内容，能够进行识别基于这样一个事实：接收的内容都经过WTA访问设备上的专用WTP端口。为了能利用WTAI电话函数接口[WTAI]访问设备和移动网络的电话特性，在WTA用户代理上运行的内容被严格地限定。

WTA网络事件是WTA用户代理接收来自移动网络响应的方法。利用“推”WTA内容的功能，还可以从WTA服务器接收WTA事件和其他的内容。WTA网络事件来自网络——客户端之间的信号，它们又被重新转换成更一般、更抽象的WTA事件。

下面详细说明WTA用户代理的行为特点：

- 在一个给定的时间内，只能有一个有效的WTA用户代理上下文。
- 无论是在以服务器为中心模式中还是在以客户端为中心的模式中，WTA的用户代理都必须能够接收“推”过来的内容。WTA事件只能用在以客户端为中心的模式中。
- 在处理“推”过来的内容（WTA事件例外）之前，要清除全部已有的WTA用户上下文。
- 当内容发出终止指令时，WTA用户代理的上下文必须立即结束。
- 处理WTA事件时，要遵循既定的优先级。

### 13.7.1 低存储空间行为

WTA是为硬件资源有限（如存储空间受到很大的限制）的设备设计的，因此，在出现故障（比如由于存储空间不足而导致的故障）的情况下，设计者确切地估计设备的行为状态是十分重要的。

#### 1. 限制历史记录

用户代理可以限制历史记录栈的大小（也就是历史导航信息的长短）。在历史记录栈被耗尽的情况下，用户代理要删除最不常用的历史记录信息。

建议所有用户代理只保持最小的历史记录栈（10个记录）。

#### 2. 限制用户代理上下文的大小

在某些情况下，设计者可能在浏览器的上下文中定义了过多的变量，以致于内存耗尽。在这种情况下，用户代理可以通过回收缓存和存放历史记录的内存（参见13.7.1节中“限制历史记录”），试着获得更多的存储空间。如果这样做无济于事，用户代理耗尽了全部的存储空间，就会向用户报错。

### 13.7.2 错误的处理

为了有良好的一致性，用户代理必须遵守本规范定义的错误条件。

### 13.7.3 未知的 DTD

在用未知的DTD进行编码的WTA事件内容格式中，可能包含一些用户代理无法辨认的元素或属性，在这种情况下，用户代理在处理 WTA事件时，就好象这些无法辨认的记号或属性不存在一样。

## 13.8 WTA 会话的管理

本节描述建立和结束一个 WTA会话所需的操作。通过 WTA会话，用户代理可以方便地同 WTA服务器和移动网络打交道。在13.8.1节和13.8.2节中，术语“WTA会话”表示在一个连接 WTA客户端和WTA服务器的安全端口[WTLS]上的WSP会话。

如果WTA客户端支持以客户端为中心的模式，建立一个 WTA会话还包括建立事件表的过程，这时，为了下载这个事件表，服务器必须提供一个可靠的连接。有关更详细的信息，请参阅WSP[WSP]和WTLS[WTLS]规范。

WTA客户端开始一个WTA会话时，通常使用下面的常见 WSP和WTLS服务：

- 开始一个新的WSP会话。
- 请求内容（事件表）。
- 开始一个WTLS的安全会话。
- 结束WSP会话。

### 13.8.1 WTA会话开始

开始操作时，WTA服务器可以检测到在与这个服务器以及移动网络进行通信的 WTA客户端。

1) 在断定一个WTA会话已经开始之后，WTA服务器遵循下列规则：

- 从WAP网关得到一个指示，这个指示表明已经建立起到 WTA客户端的WTA会话的。
- 通过“推”或“拉”，建立了一个成功的内容交换（比如恢复了一个原有的会话）。

2) 启动WTA会话需要获得下列信息：

- 所需的承载（从优先性或可用性方面考虑）。
- 需要的安全级别，这与承载网络有关[WAPARCH]。
- WTA服务器地址（特定的承载，如MSISDN）。
- 已知的WTA服务器端口号[WSP]。
- WTA用户代理的特性[WAE]。

3) 启动WTA用户代理必须采取如下的步骤（下面用到的关键字“有条件的”表示这一步骤要在支持以客户端为中心的模式中用到）：

(1) 请求一个新的WTA会话 向WSP会话层请求一个和WTA配套使用的安全WSP会话。

(2) 检查现有的事件表（有条件的） WTA客户端必须检查是否存在一个事件表。

- 丢失事件表（可能是由于存储数据丢失或者有意的从设备中删除了） 向WTA服务器请求新的事件表。这个请求或者去掉 http报头中的“最近修改时间”字段，或者包含一个空白的“最近修改时间”字段。

- 存在事件表 向WTA服务器请求新的事件表。这个请求必须包含 http报头的“最近修改时间”字段，以指出当前事件表内容修改的日期。

(3) 请求已知资源 WTA 事件表（有条件的） 必须使用已知的资源名 “ wtatable.wtae ” 和 WTA 服务器的域名构成一个请求的 URL。

例：http://www.operator.com/wtatable.wtae

(4) 来自 WTA 服务器的应答（有条件的） WTA 服务器必须用一个新的事件表内容作为应答，或者在 HTTP 报头 [WSP] 中包含一个结果代码，这个代码中包含的被请求内容与已经存储在客户端中的内容相同。

(5) 调整事件表（有条件的） 如果 WTA 服务器想要删除任何一个已经存在的事件表，必须通过应答向客户端发送一个空白事件表。

(6) 发给 WTA 服务器的确认信息（有条件的） 通过使用带有应答确认结果的 PDU，向 WTA 服务器报告 WTA 客户端建立事件表的结果 [WSP]。

(7) 启动过程的结束（有条件的） 可以使用下面的准则来判断 WTA 客户端的状态：

- 确认 “ 成功 ” 成功的确认表示 WTA 客户端中已经存在一个新的事件表，并准备好提供服务了。
- 确认 “ 失败 ” 失败的确认表示原先 WTA 客户端的事件表状态与 WTA 客户端请求的新事件表相同，也就是存在 “ 最近修改时间 ” 的信息。
- 确认丢失：如果 WTA 服务器没收到对新事件表应答的确认，应用规则与 “ 确认失败 ” 相同。
- WTA 服务器必须设定 WTA 客户端的模式，也就是以服务器为中心还是客户端为中心，基于得到的关于 WTA 用户代理特性的报告 [WAE]。

### 13.8.2 WTA 会话的终止

终止进程可以让 WTA 服务器轻松地检测到 WTA 客户端是否已经终止了它与 WTA 服务器之间的 WTA 会话。

可以使用下面的准则来判断 WTA 客户端是否终止了会话：

- WTA 服务器收到了 WAP 网关的指示，指出 WTA 客户端已终止了 WTA 会话。
- WTA 服务器因会话不存在而无法与 WTA 客户端联系，就认为 WTA 客户端已终止了 WTA 会话。

### 13.8.3 可靠的内容 “ 推 ” 操作

WTA 服务器可以将内容、页面（ deck ）和事件下推（ PUSH ）到客户端的 WTA 端口，这种 PUSH 服务由 WSP 层提供 [WSP]。在 WTA 规范中，可靠的 WSP 服务通常作为确认的内容被引用。

WTA 端口用于接收 “ 推 ” 来的内容， WTA 服务器可以选择让 WTA 客户端在确切地接收到 “ 推 ” 来的内容之后回送一个确认消息。为了能够处理没有用户响应的情况， WTA 服务器必须有自己的超时控制。 WTA 客户端一次只能接收一个 “ 推 ” 来的内容，每接收一次就必须向 WTA 服务器发送一个确认消息。

## 13.9 WTA 参考信息

本节定义了用于描述 WTA 事件和 WTA 事件表的内容格式。 WTA 事件是 XML 1.0 版本的应用。

### 13.9.1 文档标识符

这些标识符还未在 IANA 或 ISO9070 登记注册。

#### 1. SGML 公众标识符

--//WAPFORUM//DTD WTA 1.0//EN

#### 2. WTA 事件媒体类型

文本形式：

text/x-wap.wtae

标记过的格式：

application/x-wap.wtaec

这些类型还未在 IANA 注册，只是试验性的媒介类型。

### 13.9.2 文档类型定义 (DTD)

```
<!ELEMENT WTAI ( ( EVENTTABLE )+ | ( EVENT ) ) >
<!ELEMENT EVENTTABLE ( TYPE | URL | VAR )* >
<!ELEMENT EVENT ( TYPE | VAR | VALUE )* >
<!ELEMENT TYPE (#PCDATA) > -Event type- >
<!ELEMENT URL (#PCDATA) > -Event Table URL- >
<!--Variable declaration -->
<!ELEMENT VAR EMPTY>
<!ATTLIST VAR
    NAME ID #REQUIRED
>
<!--Variable value -->
<!ELEMENT VALUE EMPTY>
<!ATTLIST VALUE
    NAME IDREF #REQUIRED
    CONTENT CDATA #REQUIRED
>
```

## 13.10 WTA事件的二进制编码

WTA事件用紧凑的二进制表示来编码，这种内容格式建立在 WAP的二进制XML内容格式 ( WAP Binary XML Content Format ) 的基础之上[见WBXML]。

### 13.10.1 扩展记号

#### 1. 全局扩展记号

全局扩展记号[见WBXML]用来表示WTA事件表变量。WML变量的规则同样适用于WTA事件变量。

#### 2. 标签记号

WTA事件定义了一组单字节记号，这些记号与 DTD中定义的标签相对应，所有这些记号都定义在代码页1中。

#### 3. 属性记号

WTA事件还定义了一组单字节记号，这些记号与 DTD中定义的属性名和属性值相对应，所有这些记号也都定义在代码页 1中。

13.10.2 编码语义

1. 对变量编码

所有的变量引用必须转换成变量引用记号（例如 EXT\_I\_0）。

2. 文档的有效性

在标记一个WTA事件的过程中，会出现 XML文档有效性（见 [XML]）的问题，文档的有效性是基于 WTA事件中对文件类型（ DOCTYPE）的声明，在标记的过程中，应该向用户通报可用的格式或检测到的原页面中有效性方面的错误。

13.10.3 数字常量

1. 标签记号

表13-2中的记号代码表示代码页 0中的标签。所有的数字都是十六进制的。

表13-2 标签记号

标 签 名	记 号	标 签 名	记 号
EVENT	5	URL	8
EVENTTABLE	6	WTAI	9
TYPE	7		

注：在最终方案公布之前，记号的分配有可能改变。

2. 属性开始记号

表13-3中记号的代码表示代码页 0中的属性开始。所有的数字都是用十六进制的。

表13-3 属性开始记号

属 性 名	属性值前缀	记 号
NAME		5
VALUE		6

注：在最终方案公布之前，记号的分配有可能改变。

3. 属性值记号

没有为WTA事件定义附加的属性值。采用的规则与 WML同样[见WML]。

13.10.4 WTA编码举例

1. WTA事件

下面是做了标记的 WTA事件的例子，它演示了变量编码、属性编码和字符串表的使用。  
源WTA事件为：

```
<WTAI>
  <EVENT>
    <TYPE>cc/ic</TYPE>
    <VALUE    CONTENT=" 1 " />
```



```
<VALUE    CONTENT="6505551212" />
</EVENT>
</WTAI>
```

经过标记的格式（数字用十六进制表示）如表 13-4所示。这个例子只用了内联字符串，并且假定字符编码使用了以空白符（ NULL ）结尾的字符串格式，字符编码集是 UTF-8。

表13-4 经过标记的WTA事件举例

记 号 串	描 述
00	WBXML版本号
01	ENDWTAI元素结束

2. WTA事件表

下面是经过标记的WTA事件表的例子，它演示了变量编码、属性编码和字符串表的使用。源WTA事件表如下：

```
<?XML version="1.0" ?>
<!DOCTYPE    WTAI    SYSTEM    "wtai.dtd">
<WTAI>
  <EVENTTABLE>
    <TYPE>cc/ic</TYPE>
    <URL>http://foo.com/call</URL>
    <VAR    NAME="id" />
    <VAR    NAME="callerID" />
  </EVENTTABLE>
  <EVENTTABLE>
    <TYPE>cc/if</TYPE>
    <URL>http://foo.com/flash</URL>
    <VAR    NAME="string" />
  </EVENTTABLE>
</WTAI>
```

经过标记的格式（数字用十六进制表示）如表 13-5所示。这个例子只用了内联字符串，并且假定字符编码使用了以空白符（ NULL ）结尾的字符串格式，字符编码集是 UTF-8。

表13-5 经过标记的WTA事件

记 号 串	描 述
00	WBXML版本号
01	CWML元素结束END

13.11 WTA示例

创建这些服务的方法有多种，这里仅仅是给出了一个应用示例，不一定是最简便的处理方法。

13.11.1 提供的条件

这个例子假设WAP的移动台（ MS ）不曾预先装入任何WTA内容，下面的问题是如何激活

它的WTA特性。图13-8给出了MS和WTA服务器之间的WAP事务处理顺序。

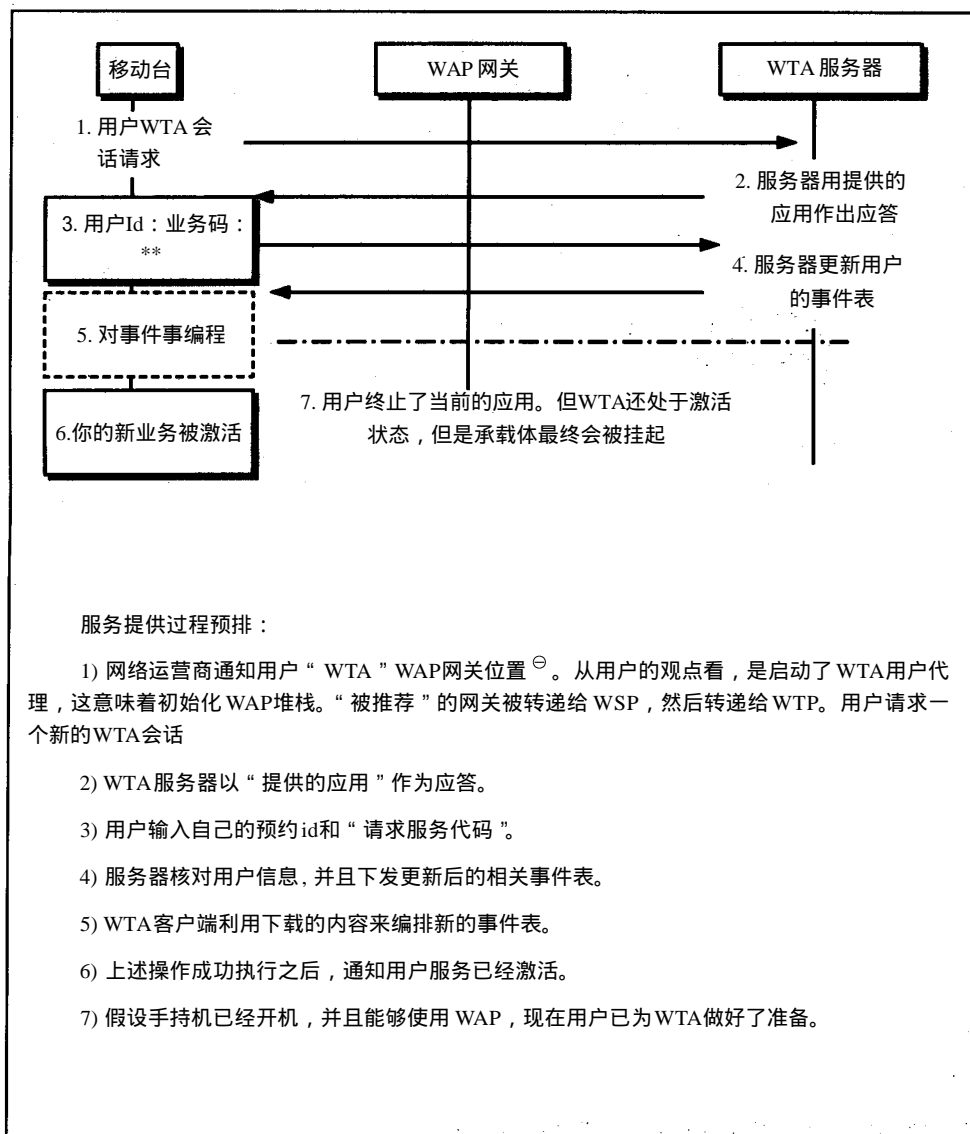


图13-8 WTA服务“提供过程”

### 13.11.2 来话呼叫

“来话呼叫”WTA的应用是通过使用WTA体系结构中的以客户端为中心模式的特性，开发应用程序的一个示例。这个例子假设以WML和WMLScript格式编写的内容已经存储在WAP客户端上了，它演示了一个“一般性”应用的WTA事件如何激活一个呼叫应用的处理过程，如何为被叫方提供增值服务（如以菜单的形式提供预呼选项）。为了在移动电话上处理呼叫和解释标准的呼叫过程，必须替换正常的移动网络呼叫处理。图13-9给出了作为示例的服务网

① WAP网关或WAP服务器的位置信息既可以手工输入到设备中也可以作为配置的一部分。

络轮廓。

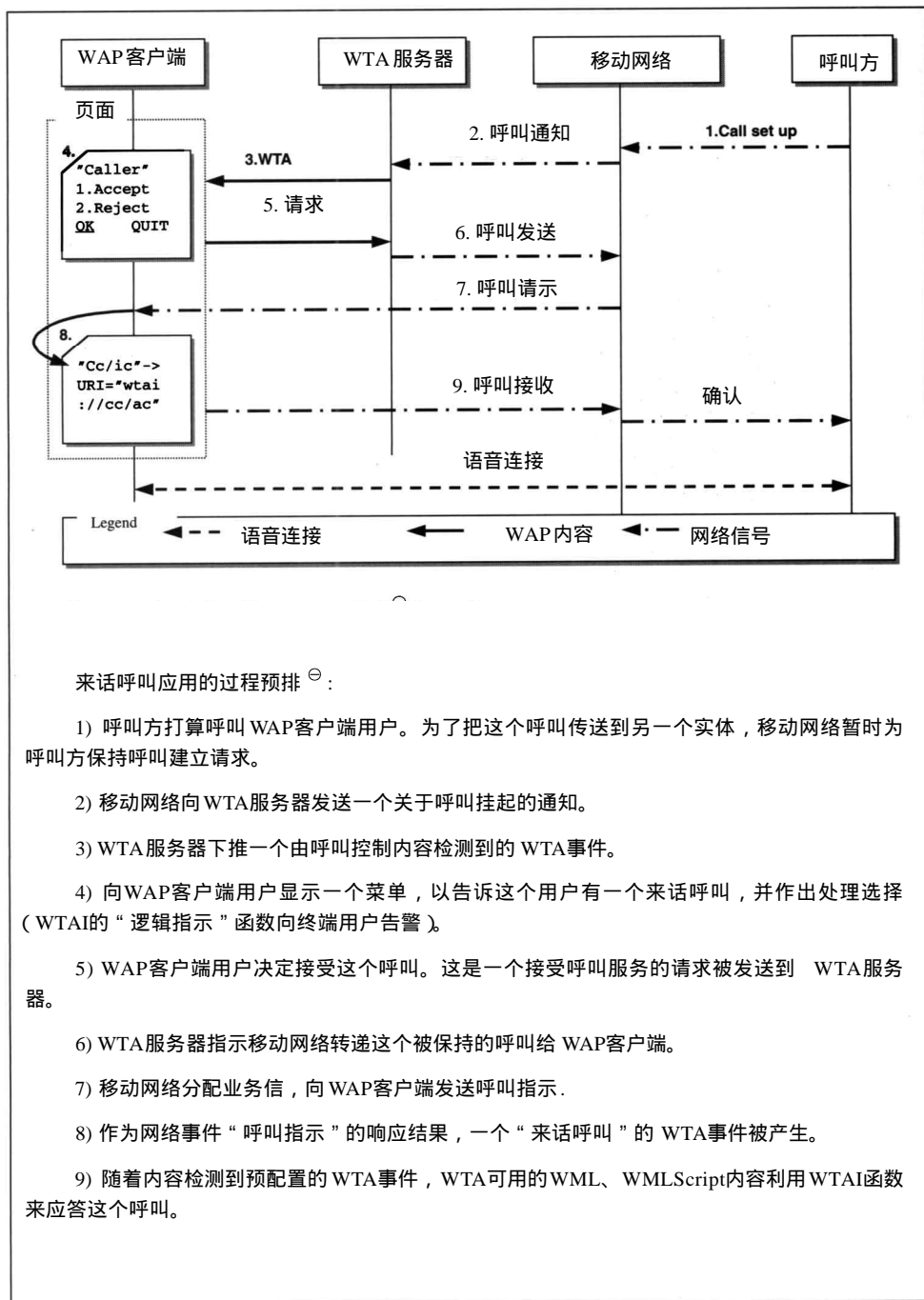


图13-9 WTA的“来话呼叫”

这个例子到移动网络通信的双方建立了实际的通话为止（图 13-10）。

① 标准移动电话人机接口不能干涉或产生标准“来话呼叫”通告。

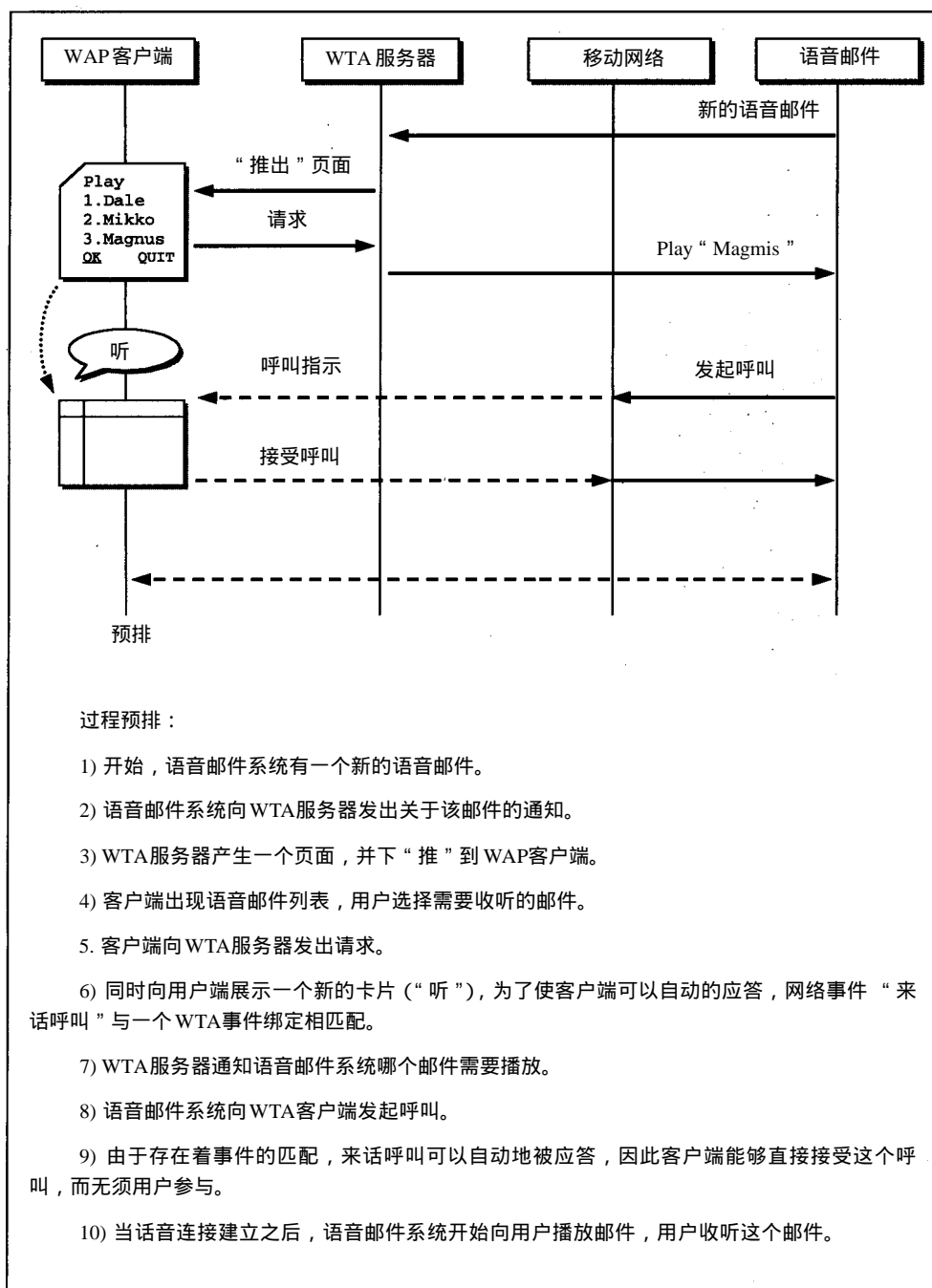


图13-10 语言邮件顺序

### 13.11.3 语音邮件

这是一个在WTA服务器和WTA客户端之间交流语音邮件的应用示例，这个应用开启了标准的浏览工具，最终在WTA服务器的帮助下激活了语音连接过程。

### 13.12 术语定义

本规范采用了下列技术术语：

**作者 (Author)** 作者是一个人或是一个应用程序，它编写或生成了无线标记语言 WML、无线标记语言脚本 WMLScript 或其他的内容。

**卡片 (Card)** 一个单独的 WML 导航和用户接口的单元，它可能包含显示给用户的信息、收集用户输入的指令等等。

**清除 (Clear)** 用于 WTA 上下文的连接。意思是原先存储在 WTA 用户上下文中的所有变量都被清除。

**客户端 (Client)** 客户端是向服务器发出连接请求的设备或应用程序。

**客户端为中心 (Client-Centric)** 由 WTA 应用的客户端来处理外部事件（如网络事件）。

**内容 (Content)** 内容是源服务器生成或存储的数据（或事件）。典型的是在响应用户请求时，内容由用户代理显示或解释。

**内容编码 (Content Encoding)** 当被用作动词时，内容编码指的是把数据对象从一种格式转换为另外一种格式的行为。通常，目标格式需要的物理空间比原格式要少，更易于处理或存储，和/或被加密。当被用作名词时，内容编码指的是一种特殊的格式或编码的标准或处理。

**内容格式 (Content Format)** 内容的实际表示。

**内容生成器 (Content Generator)** 一个服务器，它生成内容或格式化内容，通常位于源服务器端。

**页面 (Deck)** 一组 WML 卡片 (Card)。一个 WML 页面就是一个 XML 文档，它可以包含 WMLScript。

**设备 (Device)** 一个网络实体，能够发送和接收信息包的，并且有一个唯一的地址。在一个给定的上下文或跨越多重上下文，一个设备既可作为客户端，也可作为服务器。例如，一个设备作为其他服务器的客户端时可充当其他客户端的服务器。

**设备操作系统 (Device OS)** 设备中的标准操作系统，它不是 WAP 规范的一部分。

**隐含变量 (Implicit Variables)** 为了一个呼入的 WTA 事件，由用户代理自己创建的变量。

**源服务器 (Origin server)** 它作为一种服务器，是给定资源（或称内容）存储或将被生成的地方。它通常被看作是 Web 服务器或 HTTP 服务器。

**资源 (Resource)** 它是一个可以被 URL 识别的网络数据对象或服务，可以用多种表述格式所表达（例如，多种语言、数据格式、数据块尺寸和分辨率）或以其他方式进行变化。

**复位 (Reset)** 在连结 WTA 上下文时，将 WTA 用户代理的状态设置为历史记录栈为空的状态，给定的实现状态被设置为已知的状态。

**服务器 (Server)** 它是一种被动地等待一个或多个客户端连接请求的设备（或应用程序），它可以接受或拒绝来自客户端的连接请求。

**以服务器为中心 (Server-Centric)** 来自不同源点的内容都在这个服务器上终止。

**终端 (Terminal)** 终端向用户提供了用户代理的能力，它具有发出请求和接收信息的能力，也被称作移动终端或移动台。

**用户 (User)** 用户是一个通过用户代理观看、聆听或使用资源的人。

**用户代理 (User Agent)** 用户代理是可以解释 WML、WMLScript 或其他内容的软件和设备。包括文本浏览、语音浏览和搜索引擎等。

**Web 服务器 (Web Server)** 用作 HTTP 服务器的网络主机。

**无线标记语言 (WML)** 无线标记语言是一种超文本标记语言, 用于表示在窄带设备 (如: 电话) 中传输的信息。

**无线标记语言脚本 (WMLScript)** 用来对移动设备进行编程的一种脚本语言, 它是 JavaScript 脚本语言的扩展子集。

**WTA 上下文 (WTA Context)** 一个变量的完整子集, 它包含 WTA 用户代理的内容和状态。

**WTA 事件上下文绑定 (WTA Event Context Binding)** 在 WML 的 ONEVENT 的结构中, 使用类型属性规定的 WTA 事件名。

**WTA 事件 (WTA Event)** 一种以内容的形式发出的通知, 公布发送者自身状态已经改变的消息。从 WTA 源服务器可以接收到 WTA 事件, 移动网络发生的事件 (如入呼叫) 也可以生成一个 WTA 事件。

**WTA 事件表 (WTA Event Table)** 一种特定的 WTA 内容格式, 它带有一系列有序的 WTA 事件表绑定, 总是以与设备有关的格式存储在设备上。

**WTA 事件表绑定 (WTA Event Table Binding)** 一个 WTA 事件名, 它捆绑着相关的 URL 和一系列可选的参数名。

**WTA 事件句柄 (WTA Event Handler)** 用于匹配事件的实体, 可以是 WTA 上下文、WTA 事件表或者设备操作系统。

**WTA 事件匹配 (WTA Event Matching)** 一个过程。它把 WTA 事件名与作为 WTA 事件表绑定存储的 WTA 事件名相比较, 或者与 WTA 环境中用 WTA 事件环境绑定存储的 WTA 事件名相比较, 这个过程称为 WTA 事件匹配。如果该 WTA 事件名与存储的 WTA 事件名相同, 那么我们说找到了一个匹配。

**可扩展标记语言 (XML)** 可扩展标记语言是一个万维网联盟 (W3C) 的标记语言标准, WML 是其中的一种。XML 是 SGML 的一个有限子集。

### 13.13 缩略语

本规范采用了下列缩略语:

API	Application Programming Interface	应用编程接口
BNF	Backus-Naur Form	Backus-Naur 范式
CGI	Common Gateway Interface	公共网关接口
ECMA	European Computer Manufacturers Association	欧洲计算机制造商协会
ETSI	European Telecommunication Standardization Institute	欧洲电信标准化协会
GSM	Global System for Mobile Communication	全球移动通信系统
HTML	HyperText Markup Language	超文本标记语言
HTTP	HyperText Transfer Protocol	超文本传输协议
IANA	Internet Assigned Number Authority	因特网域名分配权威机构



MSC	Mobile Switching Centre	移动交换中心
MSISDN	Mobile Station International Subscriber Device Number	移动电台国际用户设备号
RFC	Request For Comments	请求注解
SGML	Standardized Generalized Markup Language	标准通用标记语言
URI	Uniform Resource Identifier	统一资源标识符
URL	Uniform Resource Locator [RFC1738]	统一资源定位器 [参见 RFC1738]
WAE	Wireless Application Environment	无线应用环境
WAP	Wireless Application Protocol [WAP]	无线应用协议
WSP	Wireless Session Protocol [WSP]	无线会话协议
WTA	Wireless Telephony Applications	无线电话应用
WTAI	Wireless Telephony Applications Interface	无线电话应用接口
WTP	Wireless Transaction Protocol	无线事务处理协议
WWW	World Wide Web	万维网
W3C	World Wide Web Consortium	万维网联盟
XML	Extensible Markup Language [XML]	可扩展标记语言

### 13.14 参考标准

- [RFC2119] "Key Words for Use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997  
URL: <ftp://ds.internic.net/rfc/rfc2119.txt>
- [RFC1738] "Uniform Resource Locators (URL)", T. Berners-Lee, et al., December 1994  
URL: <ftp://ds.internic.net/rfc/rfc1738.txt>
- [RFC1630] "Uniform Resource identifiers (URI)", T. Berners-Lee, et al., June 1994  
URL: <ftp://ds.internic.net/rfc/rfc1630.txt>
- [WAE] "Wireless Application Environment Specification", WAP Forum, 1998  
URL: <http://www.wapforum.org/>
- [WAEARCH] "Wireless Application Environment Architecture Overview", WAP Forum, 1998  
URL: <http://www.wapforum.org/>
- [WAP] "Wireless Application Protocol Architecture Specification, version 0.9", WAP Forum, 1997  
URL: <http://www.wapforum.org/>
- [WML] "Wireless Markup Language", WAP Forum, 1997  
URL: <http://www.wapforum.org/>
- [WMLScript] "WMLScript Language Specification", WAP Forum, 1998  
URL: <http://www.wapforum.org/>

- [WAESdLib] "WMLScript Standard Libraries Specification", WAP Forum, 1997  
URL: <http://www.wapforum.org/>
- [WSP] "Wireless Session Protocol Specification", WAP Forum, 1997  
URL: <http://www.wapforum.org/>
- [WTAI] "Wireless Telephony Application Interface Specification", WAP Forum, 1997  
URL: <http://www.wapforum.org/>
- [XML] "Extensible Markup Language (XML), W3C Proposed Recommendation 10-February-1998, REC-xml-19980210", T. Bray, et al., February 10, 1998  
URL: <http://www.w3.org/TR/REC-xml>