

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200610083834.9

[51] Int. Cl.

G06F 17/28 (2006.01)

G06F 17/30 (2006.01)

[43] 公开日 2007 年 12 月 12 日

[11] 公开号 CN 101086731A

[22] 申请日 2006.6.5

[21] 申请号 200610083834.9

[71] 申请人 李 钢

地址 412000 湖南省株洲市北区贺家土电厂
宿舍 11 栋 406 室

[72] 发明人 李 钢 刘智美

权利要求书 3 页 说明书 12 页 附图 7 页

[54] 发明名称

基于服务器的多国即时在线翻译系统

[57] 摘要

本发明是一种多国即时在线翻译系统，即时在线网页翻译技术(以下简称在线翻译)是一种新型的基于服务器的即时网页翻译技术，该技术利用 WEB 服务器截获客户提交的 URL 请求，并将 URL 转交给后台翻译服务器，翻译服务器首先获取 URL 内容，并将其进行翻译处理，之后再将翻译结果通过 WEB 服务器转发给客户，从而使客户能够在线即时的获得网页内容的翻译结果，极大的提高了整个互联网的使用范围、使用效率和使用价值。

1、一种新型的基于服务器的即时在线网页翻译系统，其特征是利用服务器（或服务器集群）架设大型的服务型网站，利用 WEB 服务器截获用户的 URL 请求，WEB 服务器再将 URL 转发给后台翻译服务器，翻译服务器采用网页预存、翻译引擎、预料库、工作终端计算机的语种回调、分布式校正和录入等技术将翻译后的网页转发给 WEB 服务器，再由 WEB 服务器发送给用户。

2、根据权利要求 1 所述系统，其特征在于基于服务器的即时在线翻译系统采用预存技术：

当用户点击了一个链接（或者输入一个 URL）之后，基于服务器的即时在线翻译系统并不马上抓取和翻译这个 URL 所指向的网页，而是使用既定的算法（MD5）根据 URL 计算得出一个关键字，使用该关键字检索数据库，如果数据库中已经保存了这个网页，基于服务器的即时在线翻译系统就可以直接从数据库中取出网页内容并发送给用户；否则，基于服务器的即时在线翻译系统马上抓取 URL 指向的网页，将网页内容进行翻译处理，使用既定的算法（MD5）根据 URL 计算出关键字，将关键字和内容一同保存到数据库中，之后再将内容发送给用户，这样，链接（URL）一旦被人访问过一次之后，这个链接（URL）及所指向的网页内容都会被我们的服务器记录下来，之后所有的访问都是看到我们预存后的网页。

3、根据权利要求 1 所述系统，其特征在于基于服务器的即时在线翻译系统采用可挂接网页分析引擎：

首先，分析引擎将解析网页内容，确定网页的源语种以调用相应的翻译引擎，然后，分析引擎还必须分析网页中的每一个链接，并将这些链接进行处理，使链接直接指向系统处理网页翻译的页面，当然处理后的链接还必须包含原始链接，并且系统采用动态的可挂接分析引擎，当服务器抓取到一个网页之后，

服务器根据网页内容调用相应的分析引擎，确保网页被正确的解析。

4、根据权利要求 1 所述系统，其特征在于基于服务器的即时在线翻译系统采用可挂接翻译引擎：

网页被分析引擎处理之后，相应的翻译引擎就会被调用，翻译引擎将网页内容加载到内存，并分析网页的每一个标签，遇到文字标签时，解析得到标签内的语句（词），通过基于服务器的即时在线翻译系统的翻译算法，配合以语种语料库将标签内的语句（词）翻译成目标语种，并进行替换，直至网页中最后一个标签被分析（翻译）之后，便完成了网页的翻译，此时即可将网页内容回送给服务器进行处理。

5、根据权利要求 1 所述系统，其特征在于基于服务器的即时在线翻译系统采用自学习语种语料库：

翻译引擎在工作期间，会不断向语种语料库查询语法结构、基本语意等信息，当语料库中不存在翻译引擎所需要的信息时，语料库会自动记录这个新词（句子），并要求工作终端对语料库进行语法解释和语意扩充。

6、根据权利要求 1 所述系统，其特征在于基于服务器的即时在线翻译系统采用语种翻译 即时回调：

工作终端运行分布式程序的客户端，并在运行时向服务端申请相应的语种回调服务，即向服务端发送服务类型、服务 ID 和回调请求，并向服务端暴露一个回调接口，该接口提供回调响应方法等，翻译引擎被调用时，会自动连接服务端程序，并通过服务端提供的接口向服务端查询与翻译引擎语种对应的工作终端的服务 ID，并保存起来，在翻译的过程当中，翻译引擎会通过服务端提供的接口给相应的工作终端（用先前保存的服务 ID 来确定）发送 各种回调消息。

7、根据权利要求 1 所述系统，其特征在于基于服务器的即时在线翻译系统

采用分布式语料校正和录入：

服务端同样运行在服务器上，可以和语种翻译 即时回调程序运行在同一个程序中，但必须提供语料库管理接口和语料修改和扩充接口，用于客户端调用接口的方法查询对应的语料库和对语料库进行修正和扩充操作；

客户端也一样可以和语种翻译即时回调运行在同一个分布式程序中，收到翻译引擎发送来的语料库扩充消息后，客户端会立即在工作终端发出警告信息，提醒工作终端必须马上对语料库进行修正或扩充；并自动查询服务端，得到新词（句子）的具体保存位置，弹出新词（句子）修正或扩充窗口，在工作终端完成编辑之后，自动调用服务端接口将新词（句子）的信息保存到语料库。

基于服务器的多国即时在线翻译系统

技术领域

目前，网页翻译技术主要可分为以下两种实现手段：

一. 在客户端计算机上安装 WEB 浏览器插件，这种方式实现网页的翻译有以下弊端：

1. 软件的升级更新不便，不能及时通知用户：

因为安装软件时需要复制大量的语种翻译引擎及语种语料库到客户端计算机，所以软件中的任何一个语种翻译引擎、语种语料库做了调整（哪怕是少量），都会需要制作更新包。在软件的不断升级和完善过程中，这个过程的重重复率越高时，维护成本也会越高。另外，软件升级更新时并不能及时通知用户进行更新。

2. 不能全面兼容所有 WEB 浏览器，开发者不能把工作重点有效的放在翻译技术上：

随着 INTERNET 的不断深入我们的生活，相应的 WEB 浏览器也越来越多，而不同的 WEB 浏览器中实现插件的方法也不尽相同（甚至有的浏览器不支持插件）。因此，这将导致开发者为了使软件能兼容各种浏览器，而疲于研究各种浏览器的插件开发，不能有效的专注于翻译技术的研究。

3. 不利于知识产权以及核心技术的保护：

因为大量的语种翻译引擎以及语种语料库被复制到客户端计算机中，这些都是翻译软件的具体实现和基本数据，而语种语料库则更可能是明文的（即没有采取任何加密手段）。这样一来，核心技术和知识产权的危险性将会大大增加。

二. 安装 WEB 浏览器插件，插件采用在线的语种语料库翻译网页内容。

这种方式同样存在第一种方式中的前两种弊端，但增强了知识产权和核心技术的保护，而这种方式有一个致命的缺陷：因为采用了在线语种语料库，所以在翻译过程中，需要不停的向服务器发送请求，此时当用户的网络不稳定或者网络状况不好的时候，翻译速度将不堪忍受，或者干脆无法翻译。

另外，最近有些公司干脆在自己制作的 WEB 浏览器中支持网页翻译，但同样存在上述各种弊端。

目前软件业做翻译（机器翻译）软件的公司不多，多年来国内有点名气也就东方快车、雅信、金山快译等几个软件，这些软件也提供了基于浏览器的客户端的网页翻译功能。而基于服务器的即时在线网页翻译技术则在国内（甚至国际上）都无应用先例，该技术解决了目前的翻译技术存在的各种不足：

1. 不需要用户进行升级更新，直接在服务器上将语种翻译引擎和语种语料库进行更新后，所有用户即可享受更精确的翻译服务。
2. 不必理会用户使用什么 WEB 浏览器以及什么版本，因为从服务器中发送出去的就是 HTML（XML）源文件，所有的浏览器都可以正确的解析并显示。
3. 翻译引擎和语料库不会分散到客户端计算机中，而是集中在服务器上，可以更利于技术的保密以及知识产权的保护。
4. 只要客户端计算机能连接上服务器，客户即可享受翻译服务，而不会遭遇翻译速度慢的难堪。

背景技术

在线即时翻译由三个重要因素来决定她的可行性：

1. 服务器以及相关设施。这是在线翻译的基础、在线翻译的运行平台。在

线翻译必须借助服务器以及服务器软件之间的协同合作来完成网页（当然，不仅仅是网页）的翻译。

2. 服务器之间如何实现协同合作。目前，国际上已经有相当成熟的方案：如远程调用（RPC）、WEBSERVICE、DCOM、CORBA 等等。

3. 翻译技术。这是最核心的内容，该部分内容除了实现必须的语种翻译功能外，还需要实现分布式校正、录入等功能。

发明内容

本发明—基于服务器的即时网页在线翻译系统采用了网页预存、可挂接网页分析引擎、可挂接翻译引擎、自学习语种语料库、语种翻译即时回调以及分布式语料校正和录入技术。

一. 网页预存（图1）

为了给用户提供稳定高效的翻译服务，基于服务器的即时在线翻译系统采用预存技术。

因为很有可能某一个页面能引起大部分人的兴趣，如此高频率的点击让服务器每次都重复的抓取和翻译，必将浪费大量的服务器资源，对点击这个页面的用户而言，这种等待也将是漫长（甚至是遥遥无期）的，因此这样做是十分不明智的。

当用户点击了一个链接（或者输入一个 URL）之后，基于服务器的即时在线翻译系统并不马上抓取和翻译这个 URL 所指向的网页，而是使用既定的算法（MD5）根据 URL 计算得出一个关键字，使用该关键字检索数据库，如果数据库中已经保存了这个网页，基于服务器的即时在线翻译系统就可以直接从数据库中取出网页内容并发送给用户；否则，基于服务器的即时在线翻译系统马上抓

取 URL 指向的网页，将网页内容进行翻译处理，使用既定的算法（MD5）根据 URL 计算出关键字，将关键字和内容一同保存到数据库中，之后再将内容发送给用户。这样，链接（URL）一旦被人访问过一次之后，这个链接（URL）及所指向的网页内容都会被我们的服务器记录下来，之后所有的访问都是看到我们预存后的网页，这样既节省了服务器资源，也加快了用户的浏览速度。

另外，用户可能分布在不同的地区（甚至可能在不同国家），而如果所有网页都预存在同一台数据库服务器的话，首先会大大加重服务器的负荷，还有一个更重要的问题就是不同地区的用户因连接速度（由所在的网络、经过的路由器以及线路稳定状况等因素决定）不同而将让他们的耐心倍受煎熬。因此，提供网页预存的数据库服务器必须安装大型分布式数据库，将数据库服务器安置在不同的区域，并安装分布式数据库，各个数据库服务器之间相互协作，为用户提供稳定高效的服务。

二. 可挂接网页分析引擎（图 2）

在基于服务器的即时在线翻译系统的服务器抓取到一个网页的内容之后，会遇到以下两个问题：

1. 网页的源语种是什么语种？
2. 网页中势必存在链接，怎么确保这些链接的网页也能被基于服务器的即时在线翻译系统翻译？

网页分析引擎就是用于解决这些问题的：

首先，分析引擎将解析网页内容，确定网页的源语种以调用相应的翻译引擎。不同格式的文件可以采用不同的方法获取源语种，例如 HTML 格式文件会在 HEAD 标签内包含一个带有值为 Content-Language、属性为 http-equiv 的 META 标签，该标签的 content 属性值就是源语种。当然，还有一些网页可能没有这个 META

标签，但没关系，HEAD 标签内还会有另外一个值为 Content-Type、属性为 http-equiv 的 META 标签，该标签的属性值中会包含一个 charset 子项，虽然该子项值是说明了网页的字符集，但是我们可以根据字符集来确定网页所使用的源语种。

然后，分析引擎还必须分析网页中的每一个链接，并将这些链接进行处理，使链接直接指向系统处理网页翻译的页面，当然处理后的链接还必须包含原始链接，否则系统也不知道用户点击（打开）的是哪个网页了。例如：网页中包含一个指向 <http://www.whitehouse.gov> 的链接，而系统用于处理翻译的页面为 <http://www.xxx.com/x.jsp>，那么最终我们将替换网页中的链接为 <http://www.xxx.com/x.jsp?url=http://www.whitehouse.gov>。

另外，为什么称可挂接分析引擎呢？

因为基于服务器的即时在线翻译系统所面对的 INTERNET 充斥着各式各样不同格式的网页，如：基于 HTML 格式的 .HTM、.HTML、.SHTML；基于 XML 格式的 .XML、.XSL 等。对于不同格式的网页如果采用同一种解析方式必将产生歧义（甚至根本无法解析），所以基于服务器的即时在线翻译系统必须采用动态的可挂接分析引擎，当服务器抓取到一个网页之后，服务器根据网页内容调用相应的分析引擎，确保网页被正确的解析。

三. 可挂接翻译引擎（图 3）

因为语种不同，所采用的翻译手段必定不同，因此基于服务器的即时在线翻译系统同样采取动态可挂接的方式提供不同的翻译引擎以处理不同语种间的翻译。

网页被分析引擎处理之后，相应的翻译引擎就会被调用。

翻译引擎将网页内容加载到内存，并分析网页的每一个标签，遇到文字标

签时，解析得到标签内的语句（词），通过基于服务器的即时在线翻译系统的翻译算法，配合以语种语料库将标签内的语句（词）翻译成目标语种，并进行替换。

直至网页中最后一个标签被分析（翻译）之后，便完成了网页的翻译，此时即可将网页内容回送给服务器进行处理。

四. 自学习语种语料库（图 4）

翻译引擎被启动之后，与之对应的语种语料库便会被加载。

翻译引擎在工作期间，会不断向语种语料库查询语法结构、基本语意等信息。当语料库中不存在翻译引擎所需要的信息时，语料库会自动记录这个新词（句子），并要求工作终端对语料库进行语法解释和语意扩充。

语种语料库和翻译引擎是相互对应的，当一个语种的翻译引擎被挂接起来时，其对应的语种语料库都将被翻译引擎自动加载，在翻译过程中新扩充的语法和语意信息会被实时的保存到语料库中，供以后使用。

五. 语种翻译 即时回调（图 5）

服务器上运行分布式程序的服务端，服务端是一个全局服务程序，提供一系列接口用于其他程序（本地或远程程序）与其交互。其中必须提供远程终端管理接口、远程终端消息转发接口等。服务端充当本机程序和远程终端分布式协同工作的桥梁，记录远程终端注册的服务类型及服务 ID，并在必要时为本机程序向远程终端发送回调消息。

工作终端运行分布式程序的客户端，并在运行时向服务端申请相应的语种回调服务，即向服务端发送服务类型、服务 ID 和回调请求。并向服务端暴露一个回调接口，该接口必须提供回调响应方法等。

翻译引擎被调用时，会自动连接服务端程序，并通过服务端提供的接口向

服务端查询与翻译引擎语种对应的工作终端的服务 ID，并保存起来。在翻译的过程当中，翻译引擎会通过服务端提供的接口给相应的工作终端（用先前保存的服务 ID 来确定）发送各种回调消息。例如：开始翻译时，翻译引擎会给工作终端发送开始翻译消息，在消息中包含所翻译的网页的 URL、内容大小及由 URL 计算来的关键字等；在翻译过程中向工作终端发送正在翻译消息，消息包含源语种语句（词）及目标语种语句（词）；在遇到新词（句子）而无法翻译时，向工作终端发送语料库扩充消息，消息中则必须包含该新词（句子）以及新词（句子）应该存放的语种语料库的路径；而在翻译完成后向工作中断发送结束翻译消息，消息包含了翻译后的内容大小和所保存的位置（分布式数据库所在的服务器名以及数据库名）等。

六. 分布式语料校正和录入（图 6）

服务端同样运行在服务器上，可以和语种翻译 即时回调程序运行在同一个程序中，但必须提供语料库管理接口和语料修改和扩充接口，用于客户端调用接口的方法查询对应的语料库和对语料库进行修正和扩充操作。

客户端也一样可以和语种翻译即时回调运行在同一个分布式程序中，收到翻译引擎发送来的语料库扩充消息后，客户端会立即在工作终端发出警告信息，提醒工作终端必须马上对语料库进行修正或扩充；并自动查询服务端，得到新词（句子）的具体保存位置，弹出新词（句子）修正或扩充窗口，在工作终端完成编辑之后，自动调用服务端接口将新词（句子）的信息保存到语料库。

当然，客户端还会收到其他回调消息。如正在翻译消息，因为消息中包含了两个语种的语句（词）对照，工作终端很容易就可以对翻译结果进行修正编辑，并可以将修正结果更新到语料库。

将上述技术合理有效的组织在一起，便可以圆满的完成基于服务器的即时

在线网页翻译技术(图7)。

首先,基于服务器的即时在线翻译系统在服务器上创建一个WEB页面(或者提供一个WEBSERVICE)用于提供翻译服务的出入口(即所有的翻译请求都发往这里,而最终翻译结果也由这里送出),这个页面接受必要的URL参数外,还可以接受其他辅助参数如语种ID等。这个WEB页面将所接受的参数发送给翻译服务器,并进入等待状态,一直到翻译服务器送回翻译结果为止。

接着网页预存就发挥功效了,如果页面被预存了,直接读出预存并回传给WEB页面,如果没有预存,翻译服务器则会抓取URL指定的网页,并将控制权交给网页分析引擎,网页分析引擎工作完毕之后,翻译服务器继续将控制权交给翻译引擎,翻译引擎则自动加载语种语料库,并查询相关工作终端的信息等,在翻译引擎工作期间,语种语料库、分布式录入、翻译回调等模块相互协作,并最终完成翻译服务。之后翻译引擎将翻译结果,回传翻译服务器,翻译服务器将结果预存之后,再回传给WEB页面。至此,一个页面的翻译即告完成。

附图说明

图1:

当用户点击了一个链接(或者输入一个URL)之后,基于服务器的即时在线翻译系统并不马上抓取和翻译这个URL所指向的网页,而是使用既定的算法(MD5)根据URL计算得出一个关键字,使用该关键字检索数据库,如果数据库中已经保存了这个网页,基于服务器的即时在线翻译系统就可以直接从数据库中取出网页内容并发送给用户;否则,基于服务器的即时在线翻译系统马上抓取URL指向的网页,将网页内容进行翻译处理,使用既定的算法(MD5)根据URL计算出关键字,将关键字和内容一同保存到数据库中,之后再内容发送给用户

户。这样，链接（URL）一旦被人访问过一次之后，这个链接（URL）及所指向的网页内容都会被我们的服务器记录下来，之后所有的访问都是看到我们预存后的网页。

图 2:

首先，分析引擎将解析网页内容，确定网页的源语种以调用相应的翻译引擎，然后，分析引擎还分析网页中的每一个链接，并将这些链接进行处理，使链接直接指向系统处理网页翻译的页面。

图 3:

翻译引擎将网页内容加载到内存，并分析网页的每一个标签，遇到文字标签时，解析得到标签内的语句（词），通过基于服务器的即时在线翻译系统的翻译算法，配合以语种语料库将标签内的语句（词）翻译成目标语种，并进行替换。直至网页中最后一个标签被分析（翻译）之后，便完成了网页的翻译，此时即可将网页内容回送给服务器进行处理。

图 4:

翻译引擎在工作期间，会不断向语种语料库查询语法结构、基本语意等信息。当语料库中不存在翻译引擎所需要的信息时，语料库会自动记录这个新词（句子），并要求工作终端对语料库进行语法解释和语意扩充。语种语料库和翻译引擎是相互对应的，当一个语种的翻译引擎被挂接起来时，其对应的语种语料库都将被翻译引擎自动加载，在翻译过程中新扩充的语法和语意信息会被实时的保存到语料库中，供以后使用。

图 5:

服务器上运行分布式程序的服务端，服务端是一个全局服务程序，提供一系列接口用于其他程序（本地或远程程序）与其交互。其中必须提供远程终端管理接口、远程终端消息转发接口等。服务端充当本机程序和远程终端分布式

协同工作的桥梁，记录远程终端注册的服务类型及服务 ID，并在必要时为本机程序向远程终端 发送回调消息。

图 6:

服务端同样运行在服务器上，可以和语种翻译 即时回调程序运行在同一个程序中，但必须提供语料库管理接口和语料修改和扩充接口，用于客户端调用接口的方法查询对应的语料库和对语料库进行修正和扩充操作。

图 7:

首先，基于服务器的即时在线翻译系统在服务器上创建一个 WEB 页面（或者提供一个 WEBSERVICE）用于提供翻译服务的出入口（即所有的翻译请求都发往这里，而最终翻译结果也由这里送出），这个页面接受必要的 URL 参数外，还可以接受其他辅助参数如语种 ID 等。这个 WEB 页面将所接受的参数发送给翻译服务器，并进入等待状态，一直到翻译服务器送回翻译结果为止。如果页面被预存了，直接读出 预存并回传给 WEB 页面，如果没有预存，翻译服务器则会抓取 URL 指定的网页，并将控制权交给网页分析引擎，网页分析引擎工作完毕之后，翻译服务器继续将控制权交给翻译引擎，翻译引擎则自动加载语种语料库，并查询相关工作终端的信息等，在翻译引擎工作期间，语种语料库、分布式录入、翻译回调等模块相互协作，并最终完成翻译服务。之后翻译引擎将翻译结果，回传翻译服务器，翻译服务器将结果预存之后，再回传给 WEB 页面。至此，一个页面的翻译即告完成。

具体实施方式

以用户点击了 `x.asp?url=http://www.aol.com/` 为例：

1. 用户点击这个链接之后，系统的 WEB 服务器则会收到该请求，请求会直接转发到 `x.asp` 页面。那么系统在 `x.asp` 中直接通过 DCOM 调用翻译服务器的接

口 `IServiceTranslator` 的 `translateURL` 方法，并把 URL 参数值传入该方法，然后进入等待状态。

2. `translateURL` 首先将 URL 参数进行 MD5 运算，并查询 SQL SERVER 2000 数据库是否存在该页面的预存内容，如果已经预存，则直接取出内容发回给 `x.asp`。进入步骤 11。

3. 网页没有被预存，则先将 `http://www.aol.com/` 页面内容抓取下来，简单分析一下网页格式之后，加载对应的网页分析引擎（美国在线目前是 HTML 格式）`htmlParser.dll`，并调用接口 `IHtmlParser` 的 `parseHTML` 方法对该网页内容进行处理。

4. 网页内容被分析引擎处理后，其相应的翻译引擎也被确定（分析得出该网页源语种为英文，目标语种我们规定为中文），因此 `en2chs.dll` 被加载，其接口 `ILanguageTranslator` 的 `translateIt` 方法被调用，网页内容作为参数传入了该方法。

5. `en2chs.dll` 加载时，配置文件 `en2chs.cfg` 会被其自动加载，并根据 `en2chs.cfg` 中的信息调用语料管理模块 `langResMgr.dll` 中 `ILanguageResourceManager` 接口的 `loadLanguageResource` 方法来加载相关语料库。

6. `en2chs.dll` 同时还会调用分布式协同模块 `MTCComm.dll` 中 `IMachineTranslateCommunit` 接口的 `queryLanguageCallback` 方法查询相关语种的工作终端，并将其服务 ID 保存起来。

7. `translateIt` 方法分析作为参数传进来的内容，解析出每一个文字标签，根据已经加载的语料库对文字标签的内容进行翻译，每一次翻译都会通过 `ImachineTranslateCommunit` 接口的 `sendLanguageCallbackMessage` 方法向工

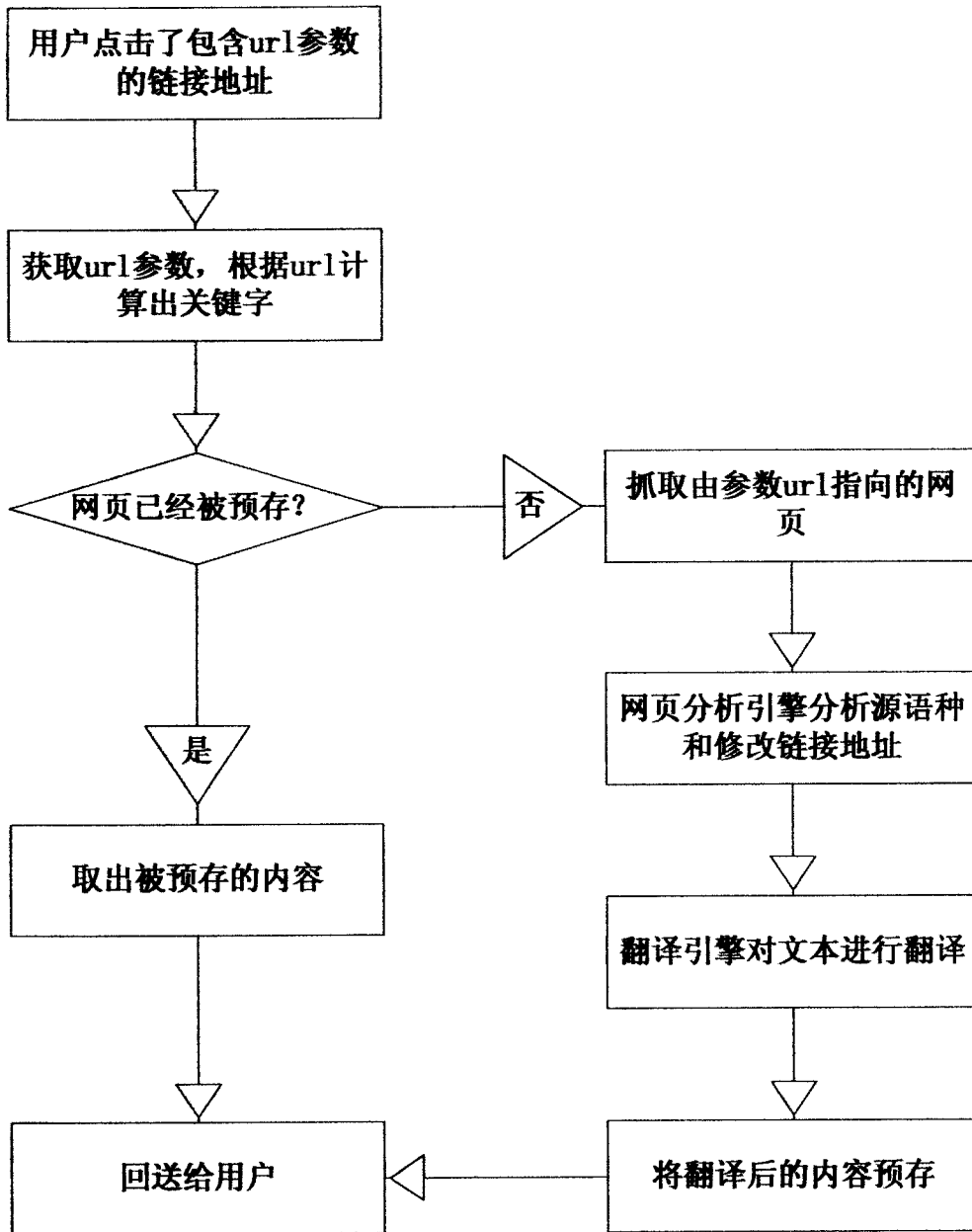
作终端发送消息。在翻译前和翻译后也会发送相关消息。特别需要注意的是，当语料库不能满足翻译需求时，他会向工作终端发送警告级别的语料扩充消息，并一直等待工作终端的响应。

8. 工作终端被启动时就立即通过 DCOM 调用服务端的 IMachineTranslateCommunit 接口的 registerLanguageCallback 方法注册相关语种的回调服务，并暴露出 ILanguageCallback 接口。该接口用于接收并由服务端调用 sendLanguageCallbackMessage 方法发送来的消息。

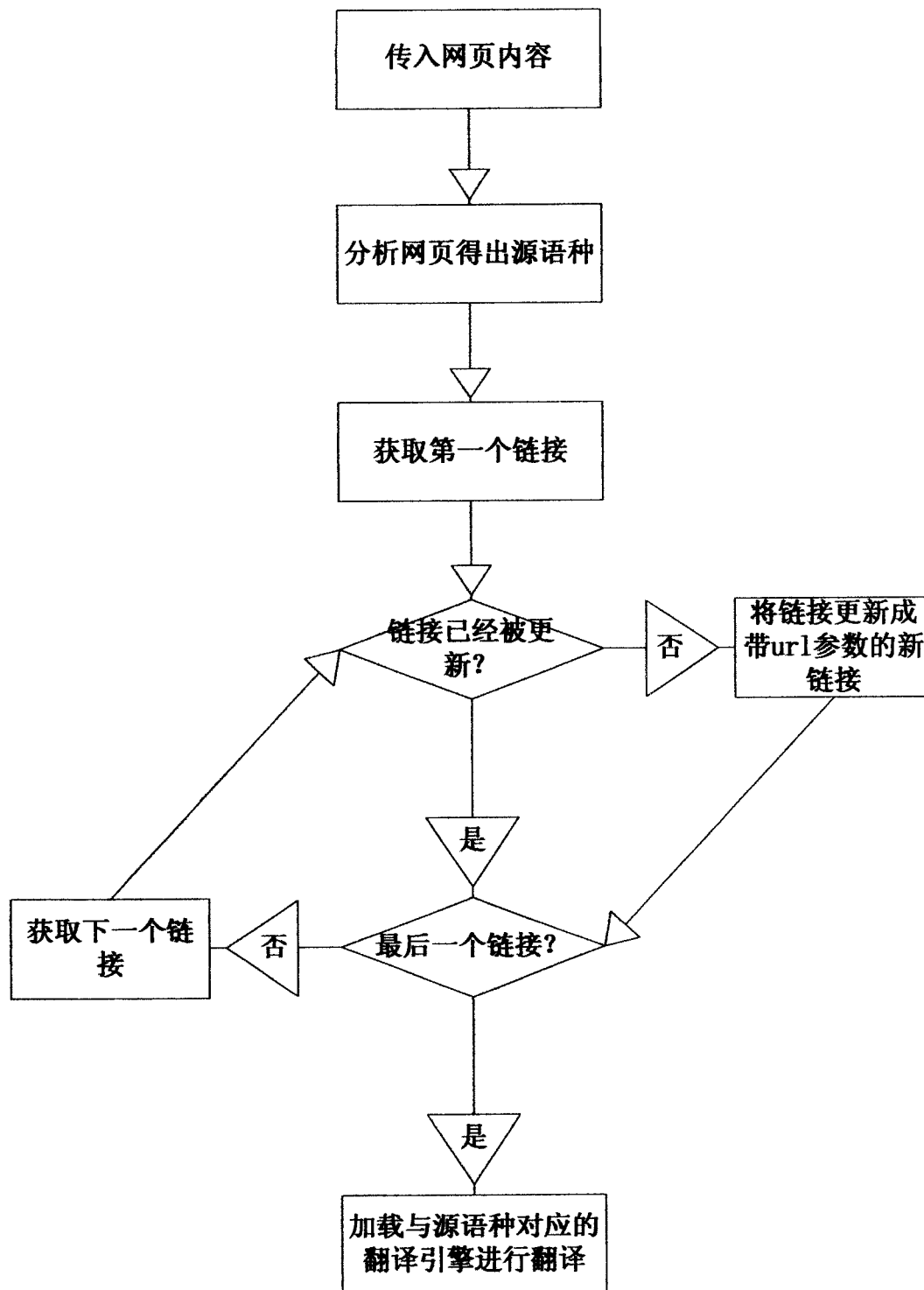
9. ILanguageCallback 收到普通的回调消息可以不予理会。但当收到语料扩充消息时，必须马上作出响应更新语料库，通过 DCOM 调用服务端的 ILanguageResourceManager 接口的 appendLanguageResource 方法对语料库进行扩充。

10. translateIt 完成翻译之后，将内容返回给 translateURL, translateURL 将返回的内容保存到 SQL SERVER 2000 数据库之后，再将内容回传给 x.asp。

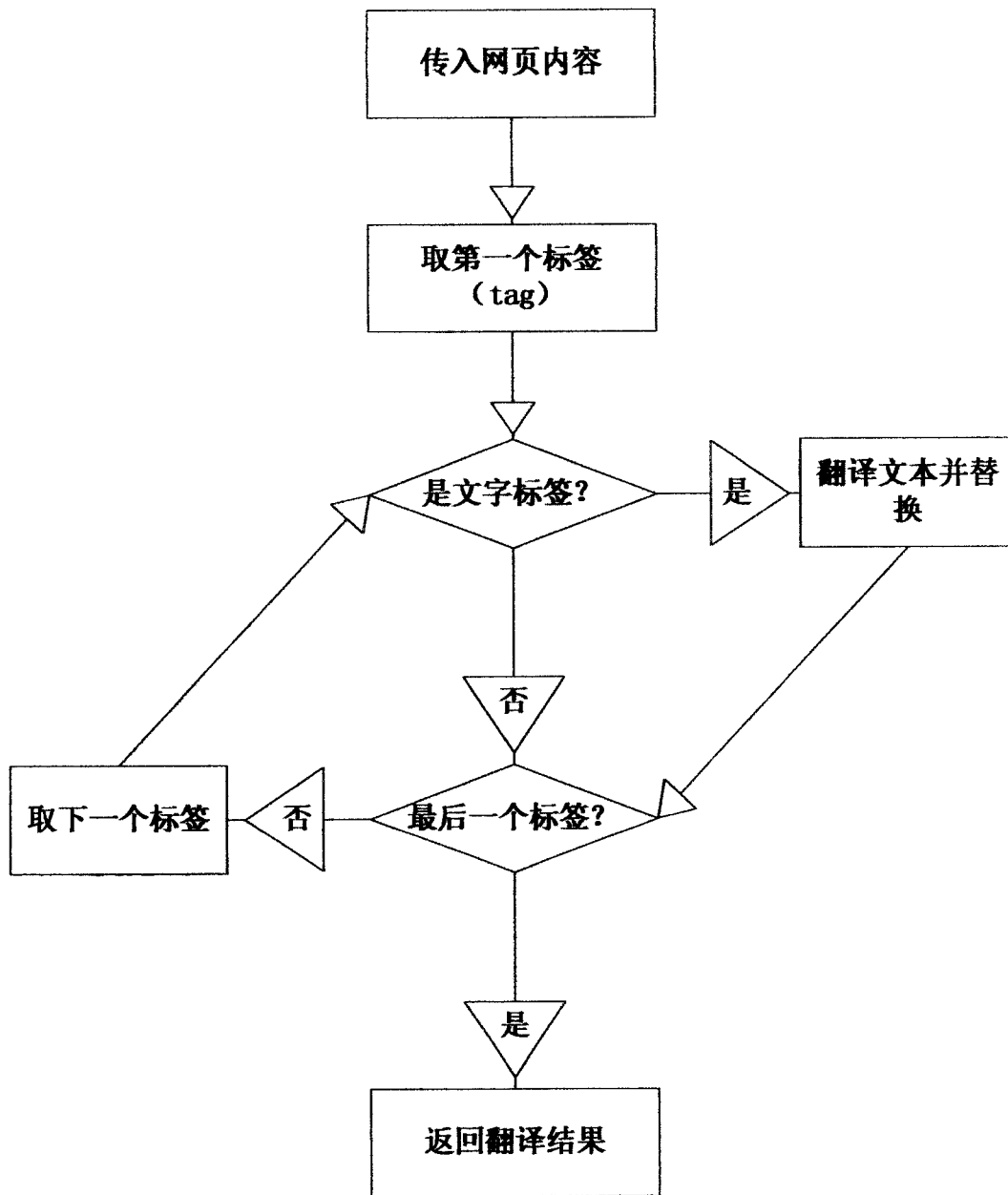
11. x.asp 收到返回信息之后，将信息再回传给客户。此时客户看到的就是被系统翻译过的美国在线的网页了。



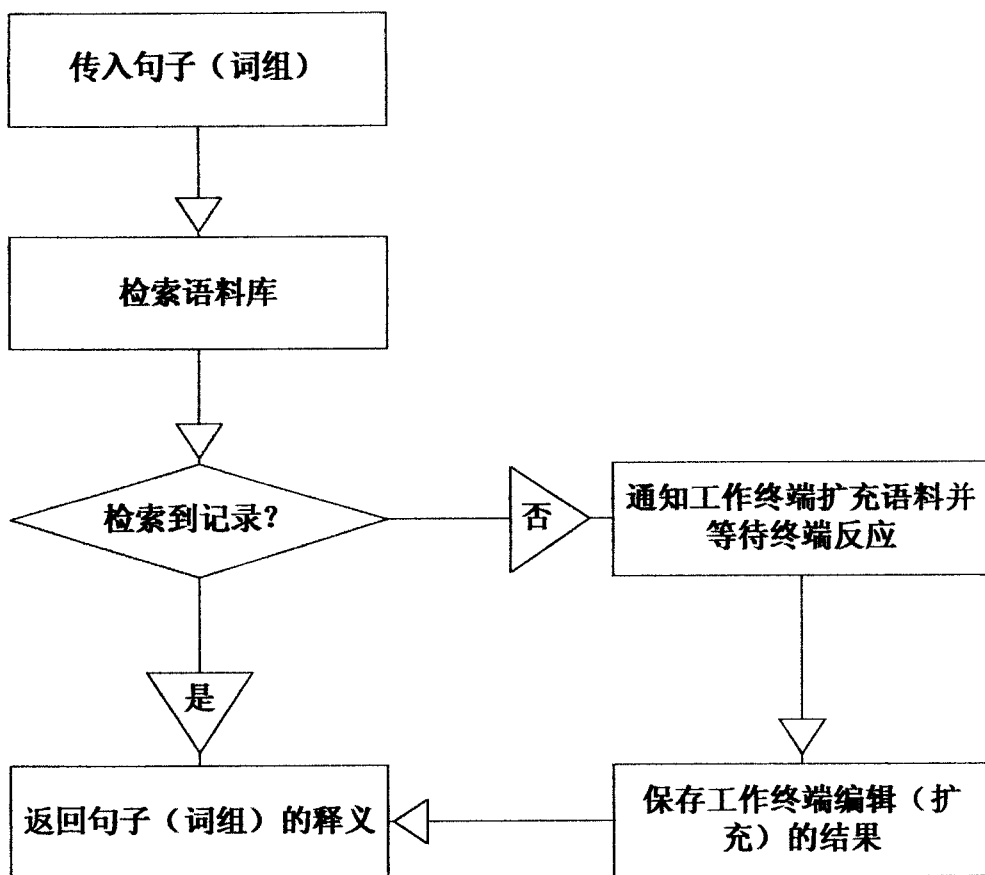
附图 1



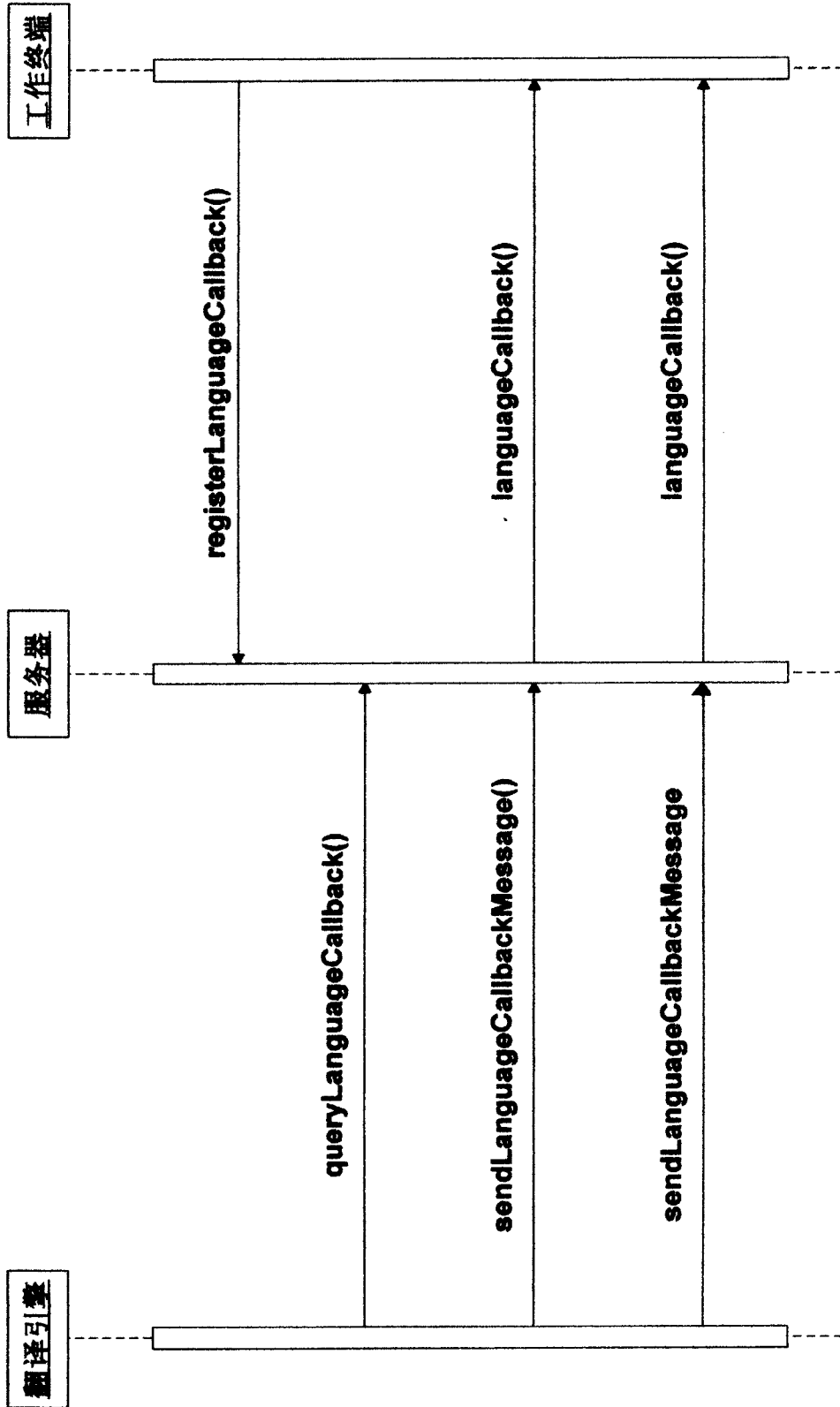
附图 2



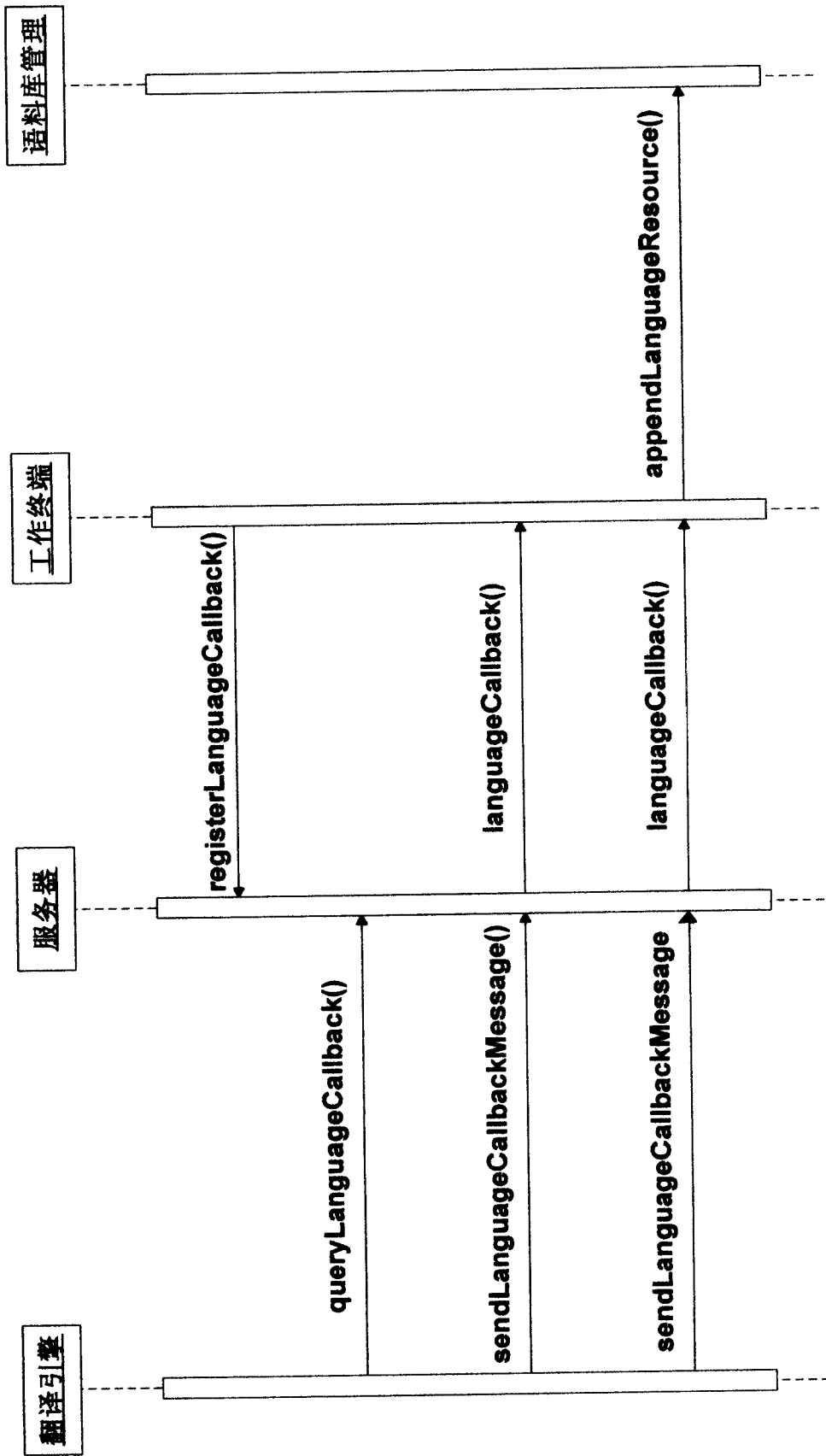
附图 3



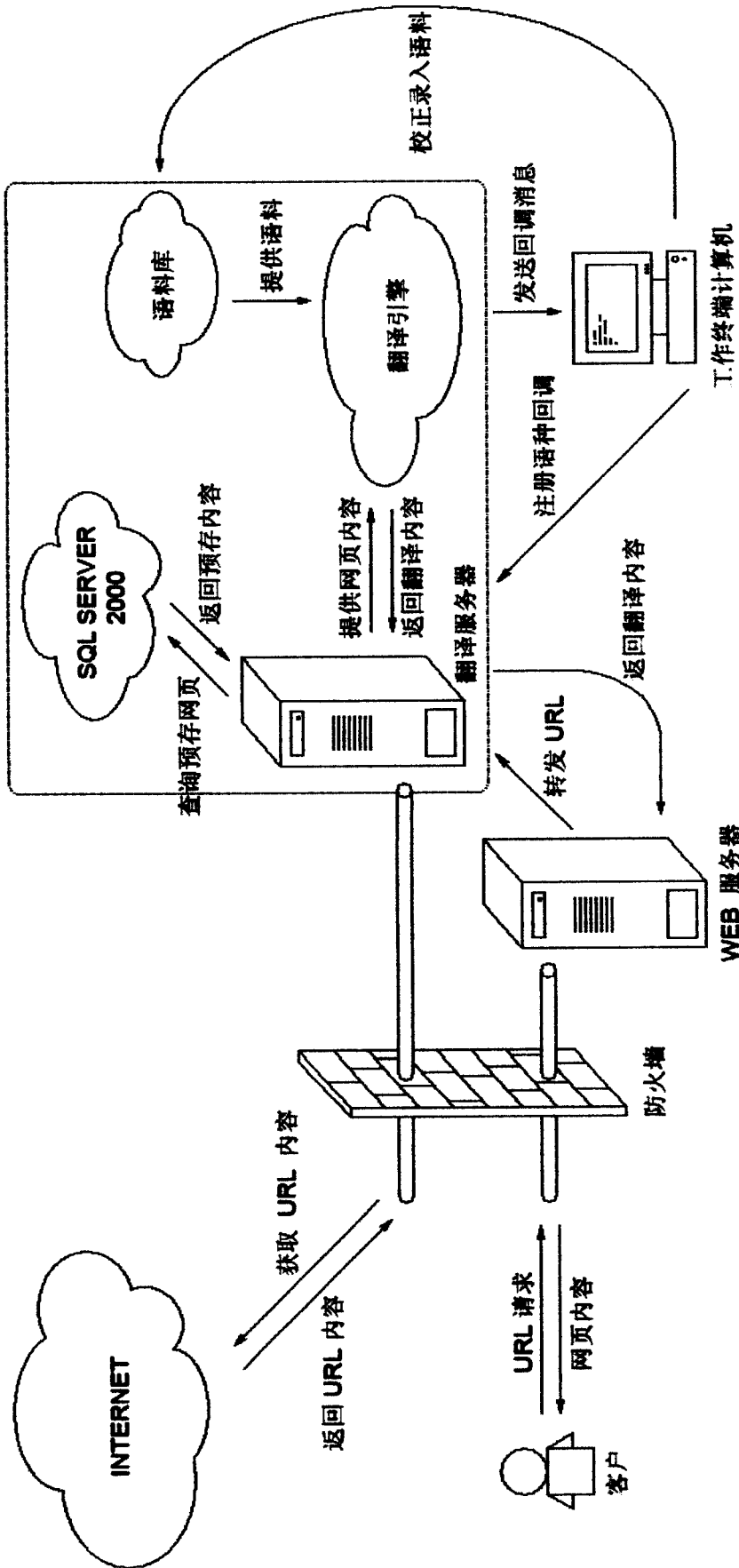
附图 4



附图 5



附图 6



附图 7