Mobile messaging, and Short Message Service (SMS) in particular, is a crucial communication channel for reaching out to your users. Messaging is also a central part of the consumer mobile experience. However, implementing mobile messaging applications is difficult and expensive due to barriers involved with interacting with closed telco services. This article reviews the background and challenges of mobile messaging, and discusses several technical approaches to address these challenges. After reading this article, you will be ready to incorporate interactive text messaging into your own applications.

Messaging is a key part of the mobile experience. In 2010, Americans sent and received over 2.1 trillion text messages on their mobile phones. Worldwide, it is estimated that active mobile messaging users outnumber email users two to one, making mobile messaging perhaps the most pervasive, effective communication channel today. Furthermore, compared with other communication channels such as email, mobile messaging comes with much less spam and is much more likely to be read by its recipients immediately after delivery.

However, unlike email, which is an open Internet standard, mobile messaging is channeled through closed telco infrastructures. That has made mobile messaging services difficult and expensive to develop. In this article, I introduce you to several low cost, or even free, techniques for incorporating mobile messaging into your applications.

The barrier to entry for mobile messaging

The "closed" nature of mobile messaging is a significant barrier to entry for developers. Because of the barrier, however, applications that can handle mobile messaging have distinct advantages. For instance, Twitter started as a text messaging company—hence the 140 char limit to tweets—and then built on its success to become the giant social platform it is today. Another example is Facebook, which also had an active SMS program very early on (sending text to FBOOK from your phone).
What about "push" notification

Mobile Push 101

Your users can't survive without being notified of even the littlest update.* The mobile Push service on IBM Cloud makes it easy to get those notifications into their waiting hands. Learn more.

* According to your marketing department.

With the rise of superphones like the iPhone and Android phones, it is now also possible to send messages directly to those phones via the standard TCP/IP network and bypass the carriers. Those messages are called push messages. Push messages are sent via Internet servers controlled by Apple and Google. Push messages are designed from the ground up to communicate with applications. They can send text, media files, and application-specific data such as alert sounds and badges to display on the application icon. Push notifications are excellent for smartphone apps, but they are much less pervasive and less reliable than carrier-based mobile messaging.

Mobile messaging basics

SMS is the most common mobile messaging channel. Practically every mobile phone can send and receive SMS messages, which are limited to 160 characters. Considering variations across different carriers, a safe limit for SMS is 140 characters.

The Multimedia Messaging Service (MMS) is an enhancement to SMS that lets users send and receive photos and short video clips via their phones. The message size is typically limited to 300KB. Under the hood, MMS uses SMS to signal users, and once the user opens the message, the phone retrieves the multimedia content via a standard email protocol. MMS is very popular among young people, but its overall market penetration is still relatively low, partly due to incompatible media payload formats across different carriers.

SMS and MMS are used mostly for peer-to-peer communication, enabling users to send messages to each other. For application developers, we are primarily interested in sending and receiving messages from and to applications. In industry jargon, we are interested in Mobile Terminated (MT) messages—sent from an application to a mobile device—and Mobile Originated (MO) messages—sent from a mobile device to an application. In order to send and receive messages to and from mobile phones, the application needs to interact with gateway servers managed by telcos.

Sending SMS and MMS for free

Sending an MT message via SMS or MMS to a phone number is actually quite simple if you know the recipient's phone number and wireless carrier. Almost all wireless carriers have email gateways that receive email messages and forward them to handsets as SMS or MMS messages. For example, for AT&T subscribers, you can send them SMS messages by emailing number@txt.att.net. AT&T will truncate the email message to 140 characters and forward to the phone number. Figure 1 shows how the message appears on the phone.
Similarly, email with photo attachments can be forwarded to phones as MMS messages via the carrier's MMS email gateway. Table 1 shows the SMS and MMS email addresses for major carriers in the US. Replace the word "number" in the email address with the recipient's 10-digit phone number (e.g. 5125551234@txt.att.net).

**Table 1. SMS and MMS gateway email addresses for major wireless carriers in the US**

<table>
<thead>
<tr>
<th>Carrier</th>
<th>SMS gateway</th>
<th>MMS gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp;T</td>
<td><a href="mailto:number@txt.att.net">number@txt.att.net</a></td>
<td><a href="mailto:number@mms.att.net">number@mms.att.net</a></td>
</tr>
<tr>
<td>Verizon</td>
<td><a href="mailto:number@vtext.com">number@vtext.com</a></td>
<td><a href="mailto:number@vzwpix.com">number@vzwpix.com</a></td>
</tr>
<tr>
<td>T-Mobile</td>
<td><a href="mailto:number@tmomail.net">number@tmomail.net</a></td>
<td><a href="mailto:number@tmomail.net">number@tmomail.net</a></td>
</tr>
<tr>
<td>Sprint</td>
<td><a href="mailto:number@messaging.sprintpcs.com">number@messaging.sprintpcs.com</a></td>
<td><a href="mailto:number@pm.sprint.com">number@pm.sprint.com</a></td>
</tr>
<tr>
<td>Virgin Mobile</td>
<td><a href="mailto:number@vmobl.com">number@vmobl.com</a></td>
<td><a href="mailto:number@vmpix.com">number@vmpix.com</a></td>
</tr>
</tbody>
</table>
The email gateway approach is good for sending one-off alerts or reminder-type messages, but for most other use cases, it has severe limitations:

- The message comes from random phone numbers and is badly formatted. It looks unprofessional.
- The user cannot reply to the message or send any information back to your application. That eliminates whole categories of applications, including the most successful SMS apps such as Twitter and American Idol.
- Since the message will come from a different phone number every time, the user cannot bookmark the phone number to associate it with your application.
- You will have to ask the user to provide information about his or her wireless carrier during user registration in order to map to the correct email gateway.

In order to interact with users in a professional fashion, you need to send and receive messages from a consistent phone number that can be identified with your service. Traditionally, that is where shortcode comes into play.

**Using shortcode**

You probably see shortcodes every day. These are the 5-digit numbers your see in restaurants, at sporting events, or on real estate for-sale signs. They ask you to "text 12345" to get more information or a coupon etc. Shortcodes are managed by "message aggregators" on behalf of telcos. Major aggregators in the US include mblox, Sybase 365, and others. You can register a shortcode with one of those aggregators, and they provide an HTTP-based web service API that allows you to send messages via your shortcode to any phone number (MT), and receive callbacks if anyone sends a message to your shortcode (MO). That sounds simple, but using shortcodes this way has some serious barriers:

- First, the cost of obtaining a dedicated shortcode is very high. It costs several thousand dollars per month, plus a large startup fee and per-message fee.
- Second, shortcodes are regulated by the Mobile Marketing Association (MMA). The MMA requires all applications to be pre-approved by each carrier. It is a lengthy and expensive process.

Fortunately, there are companies that provide shared shortcodes, which lets you take advantage of shortcodes at low cost. A leading provider in this space is TextMarks. TextMarks has a shortcode that is very easy to remember: 41411. Since it is shared by many applications, each application is distinguished via a keyword. For instance, I registered the keyword `conf` with TextMarks, and created a callback URL for the keyword (see Figure 2). The callback URL can contain templates that reference parts of the incoming SMS. For instance, \p refers to the incoming message sender's phone number, and \v refers to the message text following the keyword.
Figure 2. Register a keyword and callback URL in TextMarks

TextMarks will now forward every message that starts with CONF to the callback URL. For example, if someone texts "conf Michael Jack" to 41411 from his mobile phone 5125551234, my callback URL will receive a GET request as follows:

**Listing 1. Sample GET request**

```
http://app.ringful.com/conf?
   attendees=Michael+Jack&
   phonenumber=15125551234
```

The application handles the request. It could parse the message, save conference attendees into the database, alert the attendees, and then generate a response back to the message sender. The HTTP response body from the callback URL will be sent back to the user as the reply text message. That makes it very easy to build SMS applications that respond to user input (for example, returning a coupon for a restaurant or the sale price of a house).

Once the user texts your keyword, he or she will be registered as a subscriber of that keyword. Using its developer API, TextMarks also lets you send SMS messages to your subscribers at any time, and you can send messages to all subscribers at once or one at a time. The following API call sends a message to all subscribers to your keyword:

**Listing 2. API call to send a message to all subscribers to your keyword**

```
POST TO: http://dev1.api2.textmarks.com/GroupLeader/broadcast_message/
Parameters:
   auth_user=YOUR_USERNAME
   auth_pass=YOUR_PASSWORD
   api_key=API_KEY_FROM_TEXTMARKS
   tm=YOUR_KEYWORD
   msg=The+message+to+send+out
```
The following call sends a message to a single subscriber to your keyword:

**Listing 3. API call to send a message to a single subscriber to your keyword**

```
POST TO: http://dev1.api2.textmarks.com/GroupLeader/send_one_message/
Parameters:
  auth_user=YOUR_USERNAME
  auth_pass=YOUR_PASSWORD
  api_key=API_KEY_FROM_TEXTMARKS
  tm=YOUR_KEYWORD
  to=RECIPIENT_PHONE_NUMBER
  msg=The+message+to+send+out
```

Although TextMarks is very useful, it's still weird for users to remember and prefix their text messages with your keyword every time they text you. Also, you cannot send MT messages to anyone who has not texted your keyword first (that is, non-subscribers). The whole notion of shortcodes and keywords is designed for large-scale message broadcasts, as opposed to the one-to-one interactions that many applications need. For one-to-one interactions, long code is probably the best and most economical choice.

### Using long code

The term "long code" refers to regular ten-digit phone numbers. Instead of renting shortcodes for thousands of dollars per month, you can actually rent regular phone numbers for as little as $1 USD per month (or even get them for free in the case of Google Voice). Because those phone numbers are not associated with physical phones, they are also called "virtual numbers." You can then send and receive mobile messages via those virtual numbers.

**Twilio is not just mobile messaging**

The Twilio API also lets you make and receive voice phone calls from your virtual numbers; it supports Skype-like VoIP calls. See Twilio’s website and API documentation for more information (see Related topics for a link).

A leading provider of web services that interact with long codes is Twilio. Using the Twilio API, you can receive any message sent to that phone number, and send messages to any mobile phone number, even international ones.

Twilio provides an administration console that lets you manage multiple virtual numbers. Since those numbers only cost $1 USD per month, you might as well get one for each of your applications so that the user never needs to use any kind of keyword to distinguish between applications. Twilio does have a fee of 1 cent per SMS sent or received via their API.

The Twilio web service API is well designed and easy to use. Twilio has further simplified it by providing SDKs to developers in various programming languages. Here are some examples based on their Java SDK. The listing below shows how to send a message to a mobile phone number.
Listing 4. Sending an SMS message to any mobile number via the Twilio API

```java
public static void sendSms (String from, String to, 
    String msg) throws Exception {
    TwilioRestClient client = 
        new TwilioRestClient YOUR_API_KEY, null);
    String path = "/2010-04-01/Accounts/"+
        client.getAccountSid()+"/SMS/Messages";

    Map<String, String> vars = 
        new HashMap<String, String> ();
    vars.put("From", from);
    vars.put("To", to);
    vars.put("Body", msg);

    TwilioRestResponse tresp = 
        client.request(path, "POST", vars);
    if (tresp.isError()) {
        throw new Exception ("Twilio response error: 
            + tresp.getResponseText());
    }
}
```

For MO messages, Twilio works the same way as TextMarks: it forwards the message to a callback URL registered in the administration console, and replies to the user with the HTTP response body from the callback URL. For example, if a user sends "Hello World" to my virtual number, my registered callback URL will receive an HTTP GET as follows:

```
http://my.callback.com/process?
    From=5125551234&To=3215554567&Body=Hello+World
```

Notice that the Twilio call has a To parameter that identifies the message recipient's virtual number. That is needed because a single Twilio account could register multiple virtual numbers. My servlet at the callback URL can then handle the incoming message and generate a response.

What about MMS?

So far, except for the MMS email gateways, all the services I've discussed handle SMS interactions. MMS is inherently much more complex than SMS, because the media content must be adapted for each of the phones on each carrier. For instance, each device has a different screen resolution and expects a different video format, and each network has its limit in terms of maximum message sizes.

The difficulty of MMS is underlined by the fact that even Twitter only started to support MMS on limited US carriers in September, 2011—years after its SMS service became a blockbuster success.

Fortunately, new startups such as Hook Mobile are developing and marketing new web services that support inter-carrier MMS messaging. Hook Mobile's MMS API is not yet open to the public (you have to sign up as a partner), but this is certainly an exciting space to watch!

For smartphone users, push messaging is an attractive alternative to MMS, so I will cover push in a future article.
Related topics

- The MMA Mobile Advertising Guidelines is the industry's guideline on permitted approaches and processes to engage consumers via mobile messaging campaigns.
- TextMarks is a leading provider of low-cost SMS solutions based on their shortcode 41411.
- Hook Mobile is a leading provider of cross-carrier MMS solutions.
- Twilio provides long code-based SMS and voice APIs through its easy-to-use web services API.

© Copyright IBM Corporation 2011
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)