Identification of Individual Over-The-Air Generic Content Download for Mobile Phones

Krešimir Bokunić
VIPnet d.o.o., Zagreb, Croatia
kbokunic@vip.hr

Leo Budin
University of Zagreb,
Faculty of Electrical Engineering and Computing, Zagreb
Department of Electronics, Microelectronics, Computer and Intelligent Systems
leo.budin@fer.hr

Marin Golub
University of Zagreb,
Faculty of Electrical Engineering and Computing, Zagreb
Department of Electronics, Microelectronics, Computer and Intelligent Systems
marin.golub@fer.hr

Abstract: *This paper represents a technical overview of a current state in the Open Mobile Alliance (OMA) Over The Air (OTA) Generic Content Download method for mobile phones. It shows currently used download methods like Basic HTTP Download and MIDlet Download method. It describes architecture of the OMA Download system and different OMA Download processes. The paper analyzes Status Report Functionality in Download process. It points out problem with Status Report reception, and gives solution to the problem that could also be applied on the Mobile Digital Rights Management systems.*

Keywords: *OMA OTA, Basic HTTP Download, MIDlet Download, OMA Download, Status Report*

1. INTRODUCTION

In the last few years mobile phones have developed from simple, mainly voice enable devices to the rich multimedia capability devices and have become important part of our daily lives. Mobile phones with advanced multimedia and networking capabilities have introduced new business opportunities for mobile operators and content providers by enabling downloads of rich media content such as games, images and music. In a short time, content downloading to a mobile phone has become one of the most popular and most important mobile data service. In 2002 the Open Mobile Alliance has announced specification for Over The Air Generic Content Download than can be used for downloading all kinds of content to mobile devices. The goal of the specification was to define a technology and procedures for confirmed download that could be used for delivering higher-value media contents to mobile devices

2. OPEN MOBILE ALLIANCE GENERIC CONTENT DOWNLOAD

The Open Mobile Alliance is an open standardization body dedicated to defining an open standard based framework for mobile service, devices and enablers and with a membership of more than 250 worldwide companies.

In 2002 OMA has defined and published the first version of standard for Over The Air Generic Content Download. The specification concentrates on technology and process that can be used for confirmed download of all kinds of content to mobile devices. It goals were also to; enable both automated and manual client driven capability negotiation, avoid fragmentation of content space by enabling that content for mobile devices with different capabilities could be published using a consistent concept, create commonality between the download process of all type of media, enable

confirmed and reliable download transactions, enable mechanism that can be easily extended with new functionality and to cerate a solution that is quickly and easy to implement and deploy.
OMA Download is based on two older and successful download methods: Basic HTTP[1] Download and MIDlet[2] Download that will be explained next.

3. BASIC HTTP DOWNLOAD METHOD

Basic HTTP Download is method for transferring low-value media objects from Web servers to mobile devices. Basic HTTP Download method is based on Wireless Session Protocol (WSP), a binary Wireless Application Protocol (WAP) variant of the Web HTTP protocol. This protocols transfer objects from the Web server to the device in one single network transaction, a GET request and response. Because of its simplicity, all WAP enable phones support Basic HTTP Download.

Although, Basic HTTP download is implemented and used in the big number of the present download systems and business models its functionality is insufficient in the following situations:

- When it is important that the mobile device makes capability check to make sure it supports the media object and that the user is given an opportunity to confirm the download based on the results of the capability checks, before media object is requested and downloaded from the server.
- When it is important that the content provider or mobile operator gets an indication of the outcome of the download process, after the media object has been requested from the server

In the Basic HTTP Download, standard HTTP method for content negotiation driven by the server is used, and the server makes most of the decision before the download starts. The need for the user and the device to make decisions before the actual download starts, make HTTP content negotiation insufficient. Another limitation of this type of download is that after the download, there is no confirmation back to the server about outcome of the download process. Without this confirmation, server does not know the success of the download process, which can be used as the basic for billing and quality assurance. This knowledge is necessary to the content providers and the mobile operators to find out which media objects have been successfully downloaded and installed on the devices and which have not, so they could bill right users.
Although, this download method has many weaknesses, it is the basic for MIDlet Download and OMA Download methods, and in many cases, because of its simplicity, it is still used for downloading of low-value media contents.

4. MIDlet DOWNLOAD METHOD

MIDlet Download method is based on the Basic HTTP Download method but it expands it by adding more functionality before and after media object transfer. It also expands Basic HTTP Download method by introducing the concept of content descriptor file containing the metadata about content object, which is downloaded by the device before actual media object.
MIDlet Download method was firstly designed for downloading Java^{TM} MIDlet programs, but it can be also used as the basic for downloading of other content types.

MIDlet Download method extends Basic HTTP Download with two additional steps, one before the media object is transferred and one after:

- Before downloading of the MIDlet, a description of the MIDlet is downloaded. The description is a small file, the Java^{TM} Application Descriptor (JAD). It contains metadata about the MIDlet, like; size, type, vendor, origin, and instruction to the client on mobile device, Java^{TM} Application Manager (JAM), for how and from where to download the MIDlet

[1] HTTP – HyperText Transfer Protocol
[2] MIDlet – JAVA^{TM} Mobile Information Device Profile program

program that is stored in Java^{TM} Archive file (JAR). It also contains Uniform Resource Locator (URL) on which JAM can send notification about outcome of the download process.
- After downloading of the MIDlet, a status report, that in fact is the notification about outcome of the download process, is posted to the URL provided in the JAD file.

The MIDlet file itself is downloaded using Basic HTTP Download.

With the introduction of these two additional steps, before and after the download, MIDlet Download method splits download process in three steps. The first step is downloading of the small file that is used like descriptor of the main media object file. The second step is downloading of the actual media object file. And an additional optional third step is sending notification about success of the download process based on the information from descriptor file.

With these extensions to the Basic HTTP Download method, MIDlet Download provides a good download experience to the users by allowing them to start and control download process and it enables the devices to know in advance and accept only the types of content that they support. It also provides confirmation back to the server about outcome of the download process, which is necessary for content providers and mobile operators, and is used for billing and quality assurance.

Although, this download method does not have weaknesses as Basic HTTP Download, and can be used as a basic for downloading of other content types it is mainly used only for downloading of Java^{TM} MIDlet objects. The reason for that is the metadata from the descriptor file that are designed to satisfy needs of Java^{TM} MIDlet objects and not other object types.

On the other hand, in the MIDlet Download method, some specific elements, such as Mobile Information Device Profile (MIDP) version are used, and because of them, it cannot be fully replaced with the general download framework, such as OMA Download, that does not have those elements. Because of that, some older mobile devices that have clients for MIDlet Download method require the continuation of the MIDlet Download servers, and they are not compatible with OMA Download standard.

5. OPEN MOBILE ALLIANCE OVER THE AIR GENERIC CONTENT DOWNLOAD METHOD

The OMA OTA Generic Content Download method is also based on Basic HTTP Download but with additional features like tools for content negotiation, well-formalized metadata presentation and application layer confirmation of the object installation.

The negotiation model allows both the client device and the user to evaluate if a download is to proceed to the phase where the media object is transferred to the device. This is enabled by the use of the Download Descriptor (DD) file, which is downloaded first. This is the file, similar to JAD file from MIDlet Download method that contains the collection of attributes, used to describe the media object at the specific URL. The defined attributes are specified to allow the Download Agent to identify, retrieve, and install media objects. Some of the attributes allow the Download Agent to compare the currently available resources on the client device to the metadata representing the media object to be downloaded. Another set of attributes allow the user to evaluate if the media object is worthwhile for him and if he wants to complete the transaction or abort it before the media object has been transferred. Download Descriptor also contains attribute with the information about the location on which to send conformation about outcome of the download process and the media object installation.

The purpose of OMA Download is to provide service similar to MIDlet Download. The difference between these two download methods is that OMA Download is not designed specifically for the downloading of Java^{TM} MIDlet objects, or any other specific media types. OMA Download is a general download framework and can be used for downloading of any type of media object. However, some media types have specific requirements on the downloading procedure and because of that it is also possible to extend the general download framework with media type specific behaviour by extending DD attribute set. These extensions can also be used to trigger additional steps in the download procedure.

The OMA OTA Generic Content Download extends Basic HTTP Download with two additional steps, similar to the MIDlet Download;

- Before download, a descriptor file of the media object is downloaded. In this case it is called Download Descriptor. It contains metadata about media object, instructions to the Download Agent in the device, how to download the media object and URL on which agent can send notification about outcome of the download process.
- After download, a status report is posted to the URL provided in DD file, indicating the outcome of the download. After download, user can also be directed to a Web location provided in the Download Descriptor.

In the OMA Download method, the media object file is downloaded using basic HTTP Download.

The OMA OTA Generic Content Download, based on the concept of Download Descriptor, includes two basic scenarios:

- OMA Download with Separate Delivery of Download Descriptor and media object, in which Download Descriptor containing metadata related to the download transaction is delivered separately to the device from the media object it references. This download scenario is a process that includes three separate request-reply interaction pairs. One pair for; descriptor delivery, media object delivery and the application level transaction confirmation.
- OMA Download with Co-Delivery of Download Descriptor and media object, in which Download Descriptor and media object are delivered combine in one delivery package. This download scenario is the process where the DD file is delivered together with the media object within a single request-reply interaction. This delivery can be followed by an optional application level transaction confirmation.

In the both scenarios there may or may not be an Installation Notification sent, depending on what was requested in the Download Descriptor. If the Installation Notification is not requested and sent, then OMA Download becomes an unreliable download mechanism similar to the Basic HTTP Download, for what it had not been designed.

The OMA OTA Download method for media object transfer to mobile device with the use of Installation Notification supports a pay-per-transaction business model where the confirmation of the successful installation of the media object triggers server side billing actions and quality assurance of the service. Because of this, OMA Download method is by the mobile operators and content providers preferred media object download method.

6. BENEFITS OF OPEN MOBILE ALLIANCE GENERIC CONTENT DOWNLOAD METHOD

The main benefits of the OMA Generic Content Download over the Basic HTTP Download for the content providers and mobile operators are:

- The download can be confirmed, with the status report posted back to the download server after the download. It can be used for monitoring the quality of the service, and as the basis for billing. This also enables providing more valuable content for download.
- The Download Descriptor files of the downloadable media object can be published on the presentation server in a format independent of the presentation language supported by the devices downloading the media objects.
- After the download users can be directed to the Web location provided in the Download Descriptor file.

And for the users, the main benefits of the OMA Generic Content Download are:

- The minimized probability of downloading the media object that device does not support, because there is a capability check made before the device starts download of the object.
- Based on the information from the Download Descriptor user is given a chance to confirm other reject the download.

- The Download Descriptor can be used as the basic for a familiar and friendly user interface in the device and for making always the same download procedure independently of the type of the downloadable media object, the content provider or the mobile operator.

Because of its benefits and already said business model advantages there is intension in mobile domain to use OMA Generic Content Download as the main download method for all types of content, and it slowly replaces older download methods like Basic HTTP Download method deployed at the content providers and the mobile operators.

7. OPEN MOBILE ALLIANCE DOWNLOAD ARCHITECTURE

The OMA Download architecture for supporting OMA OTA Generic Content Download is based on the logical separation of presentation server, download server and content storage on the mobile operator or the content provider side. Usually download server has a separate part for receiving Installation Notification from the user devices. In order to support OMA OTA Download the mobile devices must contain OMA compliant Download Agent.

This kind of architecture allows a simple presentation server without complex transaction functionality. The content storage is used as the storage for any type of the content that can be downloaded. The download management and billing transaction functionality is complex and are usually concentrated and implemented in the download server.

The download and presentation server can be deployed in two ways; centralized deployment with a strong association between presentation and download server, or as decentralized deployment with a low level of integration between download and presentation server. The decentralized deployment is maybe better because it allows different presentation servers with low level of complexity from different content providers to be connected to one download server with full content management and billing logic deployed by the mobile operator.

This type of logical architecture enables the implementation of confirmed and reliable transactions between the server entity and the Download Agent included in the user devices. It is also designed to allow any type of content to be downloaded and it supports preferred pay-per-transaction business model.

8. DOWNLOAD DESCRIPTOR AND INSTALLATION NOTIFICATION

The most important entity in OMA Download method is Download Descriptor file. It contains attributes with metadata about media object and its location. These attributes allow the Download Agent included in mobile device to identify, retrieve, and install media object. It can also be used by the application, the Content Handler that is actually processing the media object. One of the functions of DD is to allow the device to verify that the desired media object is suitable for the device before being downloaded. It can also supply the Content Handler with media object specific attributes.

The Download Descriptor mandatory attributes are:

- *type* – the MIME[3] media type of the content object,
- *size* – the number of bytes to be downloaded from the URL,
- *objectURI* – the Universal Resource Identifier (URI) or URL from which media object can be downloaded.

And the optional attributes are:

- *installNotifyURI* – the URI or URL to which Installation Notification, status report is to be sent,
- *nextURL* – the URL to which the client should navigate in the case user continue browsing action after the download,
- *DDversion* – the version of the DD technology,
- *name* – a user readable name of the media object that identifies the object to the user,

[3] MIME – Multipurpose Internet Mail Extensions

- *description* – a textual description of the media object,
- *vendor* – the name of the media object provider,
- *infoURL* – the URL with further description of the media object,
- *iconURI* – the URI of an icon associated to the object,
- *installParam* – an installation parameter associated with the downloaded media object.

The most important attribute for enabling confirmed and reliable transactions between the download server entity and the Download Agent from the user device is *installNotifyURI* attribute. It also enables pay-per-transaction business model. User mobile device may send Installation Notification that is the status report of the download and installation process of the media object on the device. The purpose of the status report through the use of Installation Notification is to provide a download server with the indication that the media object has been properly received and installed. Successful status Installation Notification is sent after the device has successfully received and installed media object and the user can use it. Unsuccessful status Installation Notification is sent if something went wrong during the transfer, reception or installation process and it means that media object has been rejected and deleted from the device. The sending of the Installation Notification is valid and must be done, only if it has been explicitly requested through the use of *installNotifyURI* attribute in the DD file. In that case, the media object must not be released for use at the device unless the sending of the Installation Notification succeeds.

9. STATE MANAGEMENT OF DOWNLOAD TRANSACTIONS

During the download and installation process one download transaction consists of three steps. The first step is discovering and downloading Download Descriptor file. The second step is downloading the actual media object file that is defined in the DD file. And the third step is sending Installation Notification on the URI defined in the DD file. The state management is the most important part in the download transaction process that uses Installation Notification for the enabling of pay-per-transaction business model. In this case download server must be able to associate three different things; the offer to download the media object that means downloading Download Descriptor, the actual retrieval of the media object, and the reception of the status report through the use of Installation Notification. Without well formed and defined association between media object download and status report reception the download server cannot know which Installation Notification corresponds to which download process and pay-per-transaction business model cannot be used.

In the OMA OTA Generic Content Download specification this association between these three steps is not defined and is left to be solved by the individual implementation.

We propose implementation that is based on dynamic *installNotifyURI* generation through the use of some sort of the unique user identification number associated with random number generation.

The most important association between these three steps is association between individual media content download and Installation Notification reception. We propose that for each download transaction, unique Download Descriptor is created. All the attributes from DD can stay the same for all the download transactions of the same media object except *installNotifyURI* attribute. For each download transaction, *installNotifyURI* attribute must be different and we propose dynamic generation of the unique number that forms the part of URI to which Installation Notification is sent. This unique number can be generated for example, by using a part of International Mobile Subscriber Identity in the combination with the random number generation. The random number could be used to hide user's identity and for differentiation of the different downloads made by the same user, and the part of International Mobile Subscriber Identity could be used to decrease the possibility of generating the same unique number for more download transactions. On the content provider or mobile operator side this unique number must form a part of the URL on which Installation Notification is expected and is associated with the unique user that has requested and started download process. The URL generated in this why must exists on the download server where Installation Notification will be received. It is associated with only one URL from DD file and is used only for reception of the status report from one download transaction. In this way, when the Installation Notification arrives, mobile operators or content providers know which individual media object download it references and dependently on the returned status report, can conclude that individual user transaction.

With the use of dynamic *installNotifyURI* generation for each individual download transaction it is possible to use pay-per-transaction business model based on *installNotifyURI* attribute, so each media object download could be associated and charged to the right user.

The proposed concept of dynamically generated part of the URI can also be used for the unique identification of the media object, by generating unique part of the *objectURI* attribute from which media object could be downloaded. However it is not necessary because one media object is the same for all the users and transactions so its location can also stay the same.

The proposed dynamic *installNotifyURI* generation for each individual download transaction is also very important when Mobile Digital Rights Management (MDRM) systems must be applied on the OMA Download servers. In the MDRM systems after successful download of encrypted media object, rights object containing content encryption key and content usage rights must be sent to the user device. In order to send rights object to proper user and mobile device, there must be association between content object download, Installation Notification reception and rights object delivery. This association can also be achieved by the use of dynamic *installNotifyURI* generation for each individual download transaction because it can connect downloaded media object, end user and rights object delivery to the right user after the successful Installation Notification of the media object has been received by the MDRM system.

10. CONCLUDING REMARKS

Today, higher value media content downloads have become one of the most important business models for mobile operators and content providers. They are very important for the revenue generation and their importance in growing. In order to enable this, reliable download methods that enable pay-per-transaction business models are necessary. To solve this problem OMA has designed and specified content independent OTA Generic Content Download method. This OMA Download method has many benefits and advantages compared to older download methods like Basic HTTP or MIDlet download method. It has much more functionality than the Basic HTTP Download, in order to protect end users and devices from downloading unsupported media objects that could not be used by the device and to enable confirmed and reliable downloads and transactions between the server entity and the Download Agent included in the device, which is important for enabling pay-per-transaction business model.

MIDlet Download method is on the other hand, very similar to the OMA Download method. They both use descriptor file for providing devices with the necessary metadata about media object and they both use the concept of sending to the server the status report with the outcome of download process, but MIDlet Download method is design primarily for downloading of Java^{TM} MIDlets and not for all type of content like OMA Download.

The OMA OTA Generic Content Download specification has one incomplete but very important part. The association between three download steps; Download Descriptor download, media object download and Installation Notification reception is not defined and must be solved during individual implementation in order to enable usage of the pay-per-transaction business model, which is important and necessary for allowing higher value content download.

With the proposed dynamic *installNotifyURI* generation for each individual download transaction that connects individual media object download, the reception of Installation Notification and the end user, mentioned problem can be solved.

The proposed dynamic *installNotifyURI* generation for each individual download transaction could also be applied when Mobile Digital Rights Management systems are going to be implemented on the OMA Download servers. In this case it is important to associate successful download of encrypted media object with sending of the rights object containing content encryption key and content usage rights to the right user device. In order to enable sending rights object to proper user and mobile device, there must exist association between content object download, Installation Notification reception and rights object delivery. This association can also be achieved with the proposed dynamic generation of the *installNotifyURI.*

11. REFERENCES

[1] Becker, E.; Buhse, W.; Gunnewig, D.; Rump, N. (2002): *Digital Rights Management: Technological, Economic, Legal and Political Aspects*, M&T Books, New York

[2] Ortiz, C. E. (2002): *Introduction to OTA Application Provisioning,* SUN Microsystems, available from: http:// developers.sun.com

[3] Rosenblatt, B.; Trippe, B.; Mooney, S. (2003): *Digital Rights Management; Business and Technology*, Springer-Verlag, Berlin

[4] Open Mobile Alliance^{TM} (2002): *Generic Content Download Over The Air Specification Version 1.0*, OMA-Download-OTA-v1_0.*, available from: http://www.openmobilealliance.org/

[5] Open Mobile Alliance^{TM} (2002): *Download Architecture Version 1.0*, OMA-Download-ARCH-v1_0.*, available from: http://www.openmobilealliance.org/

[6] Open Mobile Alliance^{TM} (2002): *Digital Rights Management Version 1.0*, OMA-Download-DRM-v1_0.*, available from: http://www.openmobilealliance.org/

[7] SUN Microsystems (2001): *Over The Air User Initiated Provisioning Recommended Practice, version 1.0*, available from: http://java.sun.com/

[8] SUN Microsystems (2000): *Mobile Information Device Profile version 1.0*, available from: http://java.sun.com/

[9] SUN Microsystems, *Java 2 Platform, Micro Edition*, available from: http://java.sun.com/j2me/docs/index.html

[10] WAP Forum^{TM} (2001): *Specification of WAP Conformance Requirements*, WAP-221-CREQ-*, available from: http://www.openmobilealliance.org/

[11] WAP Forum^{TM} (2000): *HTTP State Management Specification*, WAP-223-HTTPSM-*, available from: http://www.openmobilealliance.org/

[12] WAP Forum^{TM} (2001): *Wireless Session Protocol - WSP*, WAP-230-WSP-*, available from: http://www.openmobilealliance.org/

[13] WAP Forum^{TM} (2001): *Wireless Profiled HTTP*, WAP-229-HTTP-*, available from: http://www.openmobilealliance.org/

[14] WAP Forum^{TM} (2001): *Wireless Application Protocol Architecture Specification*, WAP-210-WAPArch-*, available from: http://www.openmobilealliance.org/