FS –Application and Media Content Package Manager

Content

History

Version	Date	Adjustments
A	2006-10-03	Initial version (ERMTERI)

1 Introduction

This document specifies the Application and Media Content Package Manager (AMCPM). Its main purpose is to create Application and a Media Content Packages based on input from a set of Application and Media files, possible to install on a MAS component.

1.1 Glossary

1.1.1 Application Package

The Application package is a container for the Application logic. The logic is described using CCXML, VoiceXML, ECMA scripts and Attribute Mapping[1]

1.1.2 Media Content Package

A Media Content Package is a deliverable that contains a set of media files together with their configuration. A Media Content Package has a one-to-one relation to a Media Content Resource.

1.1.3 Application and Media Content Package relations

An Application Package has one or more related Media Content Packages and a Media Content Package has one related Application Package as show in Figure 1. Typically a Media Content Package contains e.g. all voice prompts for a particular language and voice (e.g. female).

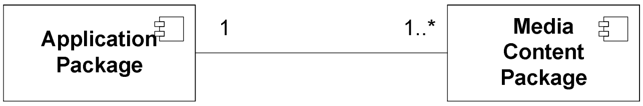

Figure 1 Application and Media Content Package relations

1.1.4 Service

Service as defined when registered in the component register.

2 Function Requirements (Commercial)

The following commercial requirements have been identified:

- It must be possible to handle any language as defined in ref. [8]
- It must be possible to handle any number of concurrent languages.

[1] Attribute Mapping defines the mapping between User Directory Attribute names and those used within the Application

3 Function Specification (Design Related)

3.1 Introduction

The AMCPM is a set of tools and procedures that assist a Service developer to create installable units on the MAS component. The input to this process is a set of CCXML, VoiceXML, ECMA Scripts, recorded media files and their configuration. Most of them are produced by the SCE.

Figure 2 shows the AMCPM interfaces.

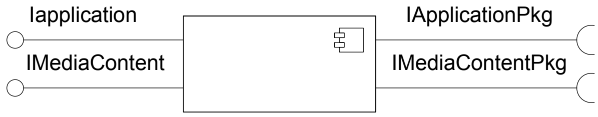

Figure 2 AMCPM interfaces

The AMCPM exports the IApplication and IMediaContent interfaces and imports the IApplicationPkg and IMediaContentPkg interfaces. These interfaces are all **manual interfaces** i.e. used by humans.

3.2 Exported Interfaces

The AMCPM exported interfaces are described in the subsections below. These interfaces are used by clients to create Application and Media Content Packages.

3.2.1 IApplication

This interface consists of a set of files of a certain format describing the Application and an Application package creation tool.

3.2.1.1 Application documents

The client to AMCPM is expected to provide the documents listed in this section. There are different kinds of documents:

1. CCXML documents as defined in ref. [1] . These documents are used when a session is started. It can be any number of CCXML documents however at least one of them has to be considered as the root document and it can be any number of root documents.
2. VoiceXML files as defined in ref. [2] . These files are used when a telephony session has started. It can be any number of VoiceXML files.
3. ECMA script files as defined in ref. [2] .
4. Optionally a configuration file in the format specified in ref. [3]
5. A Number Analysis configuration file defined in ref. [6]
6. A Event configuration files defined in ref. [7]

3.2.1.2 Application Package creation tool

The *makeApplicationPackage* is a tool that creates an Application Package. When the tool is executed it expects the information specified in section 3.2.1.3.

3.2.1.3 Application properties

1. Application name (e.g. "vva_std", "Vodafone" etc.)
2. The Product identity of the Application
3. The R-state of the Product
4. A list of Services supported by this Application
5. For each Service:
   a. The Service name as defined in MCR
   b. The CCXML document that is the root document for this Service
   c. The protocol to use to access this Service (one of xmp or sip)
   d. The protocol port to use to access this Service (any valid port).

3.2.2 IMediaContent

This interface consists of a set of Media related files and a Media Content Package creation tool.

3.2.2.1 Media related files

The client to AMCPM is expected to provide these files. There are different kinds of files:

1. Media files (regular voice or video recordings). In case the Media files consist of both voice and video recordings the audio encodings must be the same for both types. It is assumed that so called system prompts are included here i.e. referenced by the grammar file (see item 4. below)
2. Media Content files, describing the relation between a Media Content Id and a set of Media files and qualifiers.
3. Media Object files, describing the content of each Media file
4. A grammar file, describing how numbers, date, time etc. shall be pronounced.

3.2.2.2 Media Content Package creation tool

The *makeMediaContentPackage* is a tool that creates a Media Content Package. When the tool is executed it expects the information specified in section 3.2.2.5:

3.2.2.3 Grammar creation tool

The Grammar creation tool is a command line based tool used in the MAS target environment, when Grammar files are created. The purpose of the tool is to verify that a manually edited grammar file produces the expected list of spoken words. The tool takes a grammar file (containing the conversion rules) and a data file (containing e.g. a list of numbers) as input and produces a list of spoken words for each input.

Basically the Grammar file is used to control the decomposition of the following input into its language words:

1. Numbers
2. Date
3. Time

For example the number 101 would be decomposed to the following English words: "one", "hundred", "and", "one".

3.2.2.4 Media post processing tool

The Media post processing tool is a WIN32 tool that adapts video prompts to a format feasible for MAS to play.

The tool creates video prompts to be prepared for RTP streaming with the correct header information and frame packetization. The tool takes as input uncompressed RGB24 video sequences (MOV), RGB24 bitmap files (BMP) and G.711 u-Law audio files (WAV). QuickTime MOV file format is generated as output and contains a hint track for the video, a video track for the video sequence and an audio track for the spoken part of the video. The output file contains chunks of interleaved audio, video and hint data.

The MOV files for input can only contain uncompressed RGB24 video sequences and mono 8 kHz 8 bit G.711 u-Law.
The uncompressed RGB24 video frames must not exceed a maximum frame rate of 30 frames per second.
The WAV file for input can only contain mono 8 kHz 8 bit G.711 u-Law.
The BMP can only contain one frame and must be in RGB24 format.
The tool complies with the following requirements:

1. Generates H.263 baseline (Profile 0, Level 10) at a maximum rate of 15 frames per second as output.
2. Generates QCIF format of the video sequence as output.
3. Generates 8 kHz 8-bit G.711 u-law as output.
4. The RTP packets in the hint track generated by the tool comply with RFC2190 Mode A.
5. Synchronizes the audio and video streams by assuming that both start at time 0, i.e. the audio starts when the video starts.

Table 1 below shows possible conversions.

Sound input	Video input	Video output
None	BMP	Video with a still image, default length is one frame.
WAV	BMP	Video with a still image of the same length as the WAV file.
None	MOV	Video with the same length as the MOV file. If the MOV file contains a valid audio track, i.e. a supported audio codec,

		that track is used.
WAV	MOV	Video with the same length as the longest of the WAV and MOV files. If the MOV file contains an audio track that track is discarded.
WAV	None	MOV file with no video sequence.

Table 1 Possible conversion combinations

3.2.2.5 Media Content Package properties

1. The Media Content Package name
2. The Type of Media Content (e.g. "prompt", "swa" etc.)
3. The Product Id
4. The product R-State
5. The Language in the format specified in ref. [4]
6. The Voice variant if applicable (e.g. "male", "female")
7. The Video variant if applicable (e.g. "blue", "green")
8. The audio encoding as a MIME Media Type in the format specified in ref. [5]
9. The video encoding if applicable as a MIME Media Type in the format specified in ref. [5]

3.3 Imported Interfaces

These interfaces specify the structure of Application and Media Content Packages.

3.3.1 IApplicationPkg

This interface consists of a set of Application documents and configuration files. It also contains a definition file describing the properties of the Application.

3.3.1.1 Package structure

The documents and files are organized as shown in Figure 3:

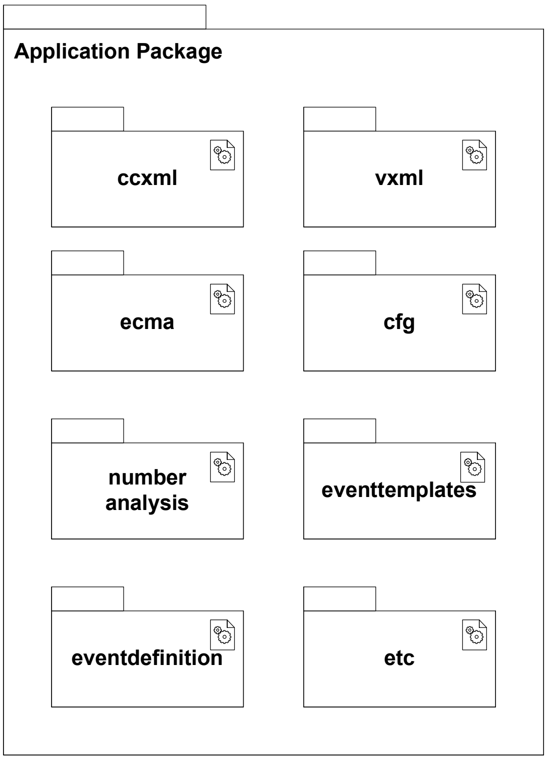

Figure 3 Application Package structure

The libraries shown in Figure 3 have the following content:

ccxml	Contains ccxml files
vxml	Contains VoiceXML files
ecma	Contains ECMA scripts
cfg	Contains Application configuration files
numberanalysis	Contains the Number Analysis configuration files
eventtemplates	Contains email templates used for notification Event.
eventdefinition	Contains the Event Definition file.
etc	Contains the Application properties as specified when the package was created (see section 3.2.1.3)

3.3.2 IMediaContentPkg

This interface consists of a set of Media Content files. It also contains a definition file describing the properties of the Media Content Package.

3.3.2.1 Package structure

The files are organized as shown in Figure 4:

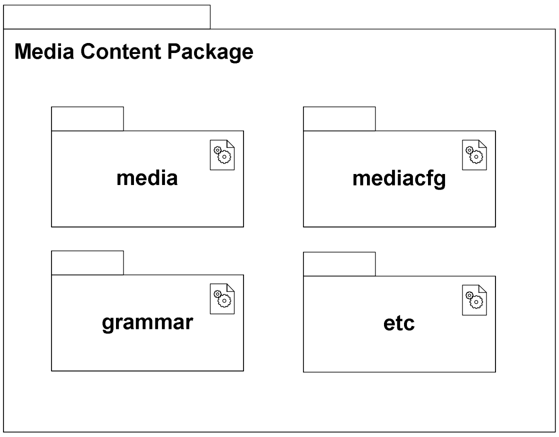

Figure 4 Media Content Package structure

The libraries shown in Figure 4 have the following content:

media	Contains recorded media e.g. wav and mov files.
mediacfg	Contains the Media Content and Media Object configuration files
grammar	Contains the grammar file
etc	Contains the Media Content Package properties as specified when the package was created (see section 3.2.2.5)

3.4 Functions

3.4.1 Create an Application Package

This function makes it possible to create an Application package that will be possible to install on MAS. The function here is a manual procedure that makes use of a tool.

The procedure involves the following steps:

1. Configure the tool with the following properties
    a. The location of the CCXML documents
    b. The location of the VoiceXML documents
    c. The location of the ECMA documents
    d. The location of the Application configuration
    e. The properties of an Application listed in section 3.2.1.3.
2. Start the tool by issuing the following command: *makeApplicationPackage*
3. The tool will create an Application Package delivery file with the following name: *<name>.<productid>.<rstate>.<os>.<ext>*

located in a specific directory.
<os> contains the operating system version
<ext> contains the chosen file format extension e.g. "tar".

3.4.2 Create a Media Content Package

This function makes it possible to create a Media Content Package that will be possible to install on MAS. The function here is a manual procedure that makes use of a tool.

The procedure involves the following steps:

1. Create a Grammar file according to section 3.4.3
2. Post process the media files according to section 3.4.4
3. Configure the tool with the following properties
    a. The location of Media files
    b. The location of Media configuration files
    c. The location of Grammar file(s)
    d. The properties of a Media Content listed in section 3.2.2.3.
4. Start the tool by issuing the following command:
   makeMediaContentPackage
4. The tool will create a Media Content Package delivery file with the following name:
   <name>.<type>.<language>.<productid>.<rstate>.<os>.<ext>
   located in a specific directory.
   <os> contains the operating system version
   <ext> contains the chosen file format extension e.g. "tar"

3.4.3 Create a Grammar file

This function makes it possible to create a Grammar file that will be understood by MAS. The function here is a manual procedure that makes use of a tool.

The procedure involves the following steps:

1. Locate the Grammar file for English (delivered with the Grammar tool) and copy it to a new file for example a grammar file for German.
2. Edit the German grammar file according to ref.[8]
3. Start the Grammar tool and load the German grammar file
4. Enter numbers, date, time etc and verify that the correct words are created. For example for the number 101, in German it would produce the following words: "ein", "hunderd", "und", "ein".
5. When all possible variants have been tested, the Grammar file is considered to be correct and can be stored in the location specified in section 3.4.2.

3.4.4 Post process media files

Depending on the format of the recorded media they might have to be post processed. Currently this concerns video Media files only. This function makes it possible to do this post processing so that the video Media files are feasibly played by MAS. The function here is a manual procedure that makes use of a tool.

The procedure involves the following steps:

1. Locate the video Media files (delivered from the recording studio)
2. Process the Media files using the Media Post processing tool.
3. When all Media files have been post processed they can be stored in the location specified in section 3.4.2.

3.5 Events

3.5.1 Produced Events

The AMCMP doesn't produce any events

3.5.2 Consumed Events

The AMCPM doesn't consume any events

4 External Operation Conditions

4.1 Configuration

As specified in section 3.2.1.3 and in section 3.2.2.5.

5 Capabilities

Intentionally left blank

6 References

[1] CCXML 1.0
www.w3c.org

[2] VoiceXML 2.0
www.w3c.org

[3] FS-Configuration Manager
16/FS-MAS0001 Uen

[4] The LDAP inetOrgPerson Object Class
RFC 2798

[5] IANA Numbers
www.iana.org

[6] FS-Number Analyzer
12/FS-MAS0001 Uen

[7] FS-Traffic Event Sender
11/FS-MAS0001 Uen

[8] FS-Media Content Manager
14/FS-MAS0001 Uen

7 Terminology

AMCPM	Application and Media Content Package Manager
MAS	Media Application Server