Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2008

C. Boulton
Avaya
T. Melanchuk
Rain Willow Communications
S. McGlashan
Hewlett-Packard

February 23, 2008

A VoiceXML Control Package for the Media Control Channel Framework
draft-boulton-ivr-vxml-control-package-04

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on August 26, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

This document defines a VoiceXML Control Package for the Media Control Channel Framework. This Control Package extends the Basic IVR control package with support for VoiceXML dialogs.

Table of Contents

1. Introduction 3
2. Conventions and Terminology 4
3. Control Package Definition 5
  3.1. Control Package Name 5
  3.2. Framework Message Usage 5
  3.3. Common XML Support 5
  3.4. CONTROL Message Body 5
  3.5. REPORT Message Body 6
4. Element Extensions 7
  4.1. <dialogprepare> 7
  4.2. <dialogstart> 7
  4.3. <event> 8
  4.4. <dialogexit> 8
5. Element Definitions 10
  5.1. <params> 10
6. Examples 11
7. Formal Syntax 13
8. Security Considerations 17
9. IANA Considerations 18
  9.1. Control Package Registration 18
  9.2. URN Sub-Namespace Registration 18
10. Change Summary 19
11. Contributors 20
12. Acknowledgments 21
13. References 22
  13.1. Normative References 22
  13.2. Informative References 22
Authors' Addresses 24
Intellectual Property and Copyright Statements 25

1. Introduction

The Media Control Channel Framework [MCCF] provides a generic approach for establishment and reporting capabilities of remotely initiated commands. The Framework utilizes many functions provided by the Session Initiation Protocol [RFC3261] (SIP) for the rendezvous and establishment of a reliable channel for control interactions. The Control Framework also introduces the concept of a Control Package. A Control Package is an explicit usage of the Control Framework for a particular interaction set.

This specification defines a Control Package for IVR functions using VoiceXML 2.0 dialogs ([VXML20], [VXML21]). As a recognized international standard for IVR dialogs, VoiceXML is used extensively within media server control languages (cf. [CCXML10], [MSML], [RFC4240]).

To ensure interoperability, implementations MUST support VoiceXML 2.0 dialogs. They SHOULD support later versions of VoiceXML (e.g. [VXML21]).

The VoiceXML package extends the basic IVR control package ([BASEIVRCP]) by replacing the basic ivr dialog type with a VoiceXML dialog type. In particular, this package

1. extends <dialogprepare> and <dialogstart> elements to include inline <vxml> dialogs, http related attributes and a <params> element to pass information into the dialog.

2. extends <dialogexit> to provide VoiceXML exit information

Otherwise, this package follows precisely the syntax and semantics of the basic IVR control package.

2. Conventions and Terminology

In this document, BCP 14/RFC 2119 [RFC2119] defines the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL". In addition, BCP 15 indicates requirement levels for compliant implementations.

The additional terms defined in Section 2 of [BASEIVRCP] are used in this document.

3. Control Package Definition

This section fulfills the mandatory requirement for information that MUST be specified during the definition of a Control Framework Package, as detailed in Section 8 of [MCCF].

3.1. Control Package Name

The Control Framework requires a Control Package definition to specify and register a unique name and version.

The name and version of this Control Package is "msc-ivr-vxml/1.0" (Media Server Control - Interactive Voice Response - VoiceXML - version 1.0). Its IANA registration is specified in Section 9.1.

3.2. Framework Message Usage

The Control Framework requires a Control Package to explicitly detail the control messages that can be used as well as provide an indication of directionality between entities. This will include which role type is allowed to initiate a request type.

This package adheres to Framework Message usage defined in Section 3.2 of [BASEIVRCP]. This package extends the dialog control elements as defined in Section 4; additional elements are defined in Section 5. A XML Schema for this package is provided in Section 7.

Implementation of this control package MUST adhere to the syntax and semantics of XML elements defined in this document. In cases where there is a difference in constraints between the XML schema and the textual description of elements, the textual definition takes priority.

3.3. Common XML Support

The Control Framework requires a Control Package definition to specify if the attributes for media dialog or conference references are required.

This package adheres to Common XML Support defined in Section 3.3 of [BASEIVRCP].

3.4. CONTROL Message Body

The Control Framework requires a Control Package to define the control body that can be contained within a CONTROL command request and to indicate the location of detailed syntax definitions and semantics for the appropriate body types.

This package adheres to CONTROL Message Body as defined in Section 3.4 of [BASEIVRCP].

3.5. REPORT Message Body

The Control Framework requires a control package definition to define the REPORT body that can be contained within a REPORT command request, or that no report package body is required. This section should indicate the location of detailed syntax definitions and semantics for the appropriate body types.

This package adheres to REPORT Message Body as defined in Section 3.5 of [BASEIVRCP].

4. Element Extensions

The XML elements used in this package are those defined in Section 4 of [BASEIVRCP] unless otherwise specified in this section.

4.1. <dialogprepare>

This package extends the definition of <dialogprepare> request in Section 4.1.1 of [BASEIVRCP] as follows.

The <dialogprepare> element has the following modified attributes:

src: implementations MUST support the HTTP protocol

type: The default value is "application/voicexml+xml".

The <dialogprepare> element has the following additional attributes:

maxage: string defining a time interval according to the max-age parameter in HTTP. The attribute is optional.

maxstale: string defining a time interval according to the max-stale parameter in HTTP. The attribute is optional.

method: string indicating the HTTP method to use. Permitted values are "post" or "get". The default value is "get". The attribute is optional.

enctype: string identifying the encoding type of the submitted document (when the value of the method attribute is 'post'). The default value is "application/x-www-form-url-encoded". The attribute is optional.

The <dialogprepare> element has the following additional child elements:

<vxml>: contains an inline VoiceXML document in the VoiceXML namespace. The element is optional.

4.2. <dialogstart>

This package extends the definition of <dialogstart> request in Section 4.1.2 of [BASEIVRCP] as follows.

The <dialogstart> element has the following modified attributes:

src: implementations MUST support the HTTP URI protocol

type: The default value is "application/voicexml+xml".

The <dialogstart> element has the following additional attributes:

maxage: string defining a time interval according to the max-age parameter in HTTP. The attribute is optional.

maxstale: string defining a time interval according to the max-stale parameter in HTTP. The attribute is optional.

method: string indicating the HTTP method to use. Permitted values are "post" or "get". The default value is "get". The attribute is optional.

enctype: string identifying the encoding type of the submitted document (when the value of the method attribute is 'post'). The default value is "application/x-www-form-url-encoded". The attribute is optional.

The <dialogstart> element has the following additional child elements defined:

<vxml>: contains an inline VoiceXML document in the VoiceXML namespace. The element is optional.

<params>: an XML data structure (see Section 5.1) to pass parameters into the dialog. The element is optional.

[Editors Note: further work is required to define how connection related information is passed to the VoiceXML interpreter.]

4.3. <event>

Transfer behavior is not defined.

[Editors Note: A later version of this package may specify how <event> is extended with child elements describing transfer events.]

4.4. <dialogexit>

This package extends the definition of <dialogexit> in Section 4.3.2 of [BASEIVRCP] as follows.

The <dialogexit> element has the following additional attributes:

termmode: indicates how the voicexml dialog was terminated. Valid values are: exit, disconnect, or implementation specific string values beginning with "_". The attribute is mandatory.

The <dialogexit> element has the following additional child element:

<params>: parameters returned from the dialog. The element is optional.

5. Element Definitions

5.1. <params>

The <params> element is a general container for parameterized data.

The <data> element has no attributes, but has the following child elements defined:

<param>: specifies a parameter. Multiple instances of this element are permitted. The element is mandatory.

The <param> element has the following attributes:

name: a string indicating the name of the parameter. The attribute is mandatory.

type: a string indicating the type of the value. The attribute is optional. The default is a string type.

value: a string indicating the value of the parameter. separation. The attribute is mandatory.

The <param> element has no children.

[Editors Note: this defintion of <param> may be extended in a later version to allow inline binary data, used reporting recordings when the dialog exits.]

6. Examples

[Editors Note: this section needs further work.]

Example: a request to prepare a dialog where the dialog script is is referenced:

```
<dialogprepare type="application/voicexml+xml"
                  src="http://www.example.com/playprompt.vxml">
</dialogprepare>
```

In the following example, the VoiceXML dialog script is specified inline:

```
<dialogprepare>
   <vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
      <form id='main'>
         <block>
            <audio expr="http://www.example.com/media/prompt1.wav"/>
            <exit/>
          </block>
      </form>
   </vxml>
</dialogprepare>
```

The following example shows a request to start a dialog on a conference where the dialog script is indicated using the src attribute:

```
 <dialogstart conferenceid="conference11"
    type="application/voicexml+xml"
    src="http://www.example.com/playprompt.vxml">
    <params>
      <param name="media"
      value="http://www.example.com/media/prompt1.wav"/>
    </params>
 </dialogstart>
```

Where the parameter "media" would be available in the VoiceXML script as "connection.ccxml.values.media" so different prompts can be played using the same dialog script.

In the following example, the VoiceXML dialog script is specified inline.

```
<dialogstart conferenceid="conference11">
    <vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
     <form id='main'>
        <block>
           <audio expr="http://www.example.com/media/prompt1.wav"/>
           <exit/>
         </block>
      </form>
    </vxml>
 </dialogstart>
```

7. Formal Syntax

This package defines an XML schema which extends the msc-ivr-common.xsd schema defined in Section 9 of [BASEIVRCP].

All elements in the defined schema are in the urn:ietf:params:xml:ns:msc-ivr-vxml namespace.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="urn:ietf:params:xml:ns:msc-ivr-vxml"
 xmlns:fw="urn:ietf:params:xml:ns:control:framework-attributes"
 elementFormDefault="qualified"
 xmlns="urn:ietf:params:xml:ns:msc-ivr-vxml"
 xmlns:vxml="http://www.w3.org/2001/vxml"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">


 <xsd:annotation>
  <xsd:documentation>
   IETF MediaCtrl VoiceXML IVR 1.0 (20080225)

   This is the schema of the VoiceXML IVR Control Package.
   It imports the msc-ivr-common schema (dialogprepare, etc)
   and extends them for VoiceXML support.

   The schema namespace is urn:ietf:params:xml:ns:msc-ivr-vxml

  </xsd:documentation>
 </xsd:annotation>


 <!--
  ##############################################################

  SCHEMA IMPORTS

  ##############################################################
 -->


 <xsd:redefine schemaLocation="msc-ivr-common.xsd">
  <xsd:annotation>
   <xsd:documentation>
    This import brings in the IVR common package Redefinitions: [1]
    Adds http related attributes in dialogprepare [2] Adds http
    related attributes in dialogstart [3] Allow params in dialogstart
    [4] Adds attribute and params to dialogexit
   </xsd:documentation>
```

```
    </xsd:annotation>

    <xsd:attributeGroup name="msc.ivr.common.dialogprepare.attlist">
     <xsd:attributeGroup ref="msc.ivr.common.dialogprepare.attlist" />
     <xsd:attributeGroup ref="httpparams" />
    </xsd:attributeGroup>


    <xsd:group name="msc.ivr.common.dialogstart.mix">
     <xsd:choice>
      <xsd:group ref="msc.ivr.common.dialogstart.mix" />
      <xsd:element ref="params" minOccurs="0" maxOccurs="1" />
     </xsd:choice>
    </xsd:group>


    <xsd:attributeGroup name="msc.ivr.common.dialogstart.attlist">
     <xsd:attributeGroup ref="msc.ivr.common.dialogstart.attlist" />
     <xsd:attributeGroup ref="httpparams" />
    </xsd:attributeGroup>


    <xsd:attributeGroup name="msc.ivr.common.dialogexit.attlist">
     <xsd:attributeGroup ref="msc.ivr.common.dialogexit.attlist" />
     <xsd:attribute name="termmode" type="vxml_termmode.datatype"
      use="required" />
    </xsd:attributeGroup>


    <xsd:group name="msc.ivr.common.dialogexit.mix">
     <xsd:choice>
      <xsd:group ref="msc.ivr.common.dialogexit.mix" />
      <xsd:element ref="params" minOccurs="0" maxOccurs="1" />
     </xsd:choice>
    </xsd:group>

   </xsd:redefine>



   <!--
    ##############################################################

    ELEMENTS

    ##############################################################
   -->
```

```
<!-- params -->

<xsd:attributeGroup name="msc.ivr.vxml.params.attlist">
 <xsd:attributeGroup ref="msc.common.attribs" />
</xsd:attributeGroup>

<xsd:group name="msc.ivr.vxml.params.mix">
 <xsd:choice>
  <xsd:element ref="param" minOccurs="1" maxOccurs="unbounded" />
  <xsd:group ref="msc.common.content" minOccurs="0"
   maxOccurs="unbounded" />
 </xsd:choice>
</xsd:group>

<xsd:group name="msc.ivr.vxml.params.content">
 <xsd:sequence>
  <xsd:group ref="msc.ivr.vxml.params.mix" minOccurs="0"
   maxOccurs="unbounded" />
 </xsd:sequence>
</xsd:group>

<xsd:complexType name="msc.ivr.vxml.params.type">
 <xsd:group ref="msc.ivr.vxml.params.content" />
 <xsd:attributeGroup ref="msc.ivr.vxml.params.attlist" />
</xsd:complexType>

<xsd:element name="params" type="msc.ivr.vxml.params.type" />

<!--  param -->

<xsd:attributeGroup name="msc.ivr.vxml.param.attlist">
 <xsd:attribute name="name" type="xsd:string" use="required" />
 <xsd:attribute name="valuetype" type="xsd:string" />
 <xsd:attribute name="value" type="xsd:string" />
 <xsd:attributeGroup ref="msc.common.attribs" />
</xsd:attributeGroup>

<xsd:group name="msc.ivr.vxml.param.mix">
 <xsd:choice>
  <xsd:group ref="msc.common.content" minOccurs="0"
   maxOccurs="unbounded" />
 </xsd:choice>
</xsd:group>

<xsd:group name="msc.ivr.vxml.param.content">
 <xsd:sequence>
  <xsd:group ref="msc.ivr.vxml.param.mix" minOccurs="0"
   maxOccurs="unbounded" />
```

```
    </xsd:sequence>
   </xsd:group>

   <xsd:complexType name="msc.ivr.vxml.param.type" mixed="true">
    <xsd:group ref="msc.ivr.vxml.param.content" />
    <xsd:attributeGroup ref="msc.ivr.vxml.param.attlist" />
   </xsd:complexType>

   <xsd:element name="param" type="msc.ivr.vxml.param.type" />


   <!--
    ##############################################################

    ATTRIBUTES

    ##############################################################
   -->

   <xsd:attributeGroup name="httpparams">
    <xsd:attribute name="maxage" type="timedesignation.datatype" />
    <xsd:attribute name="maxstale" type="timedesignation.datatype" />
    <xsd:attribute name="enctype" type="xsd:string"
     default="application/x-www-form-urlencoded" />
    <xsd:attribute name="method" type="method.datatype" default="get" />
   </xsd:attributeGroup>


   <!--
    ##############################################################

    DATATYPES

    ##############################################################
   -->

   <xsd:simpleType name="method.datatype">
    <xsd:restriction base="xsd:NMTOKEN">
     <xsd:enumeration value="get" />
     <xsd:enumeration value="post" />
    </xsd:restriction>
   </xsd:simpleType>

   <xsd:simpleType name="vxml_termmode.datatype">
    <xsd:restriction base="xsd:string"></xsd:restriction>
   </xsd:simpleType>

  </xsd:schema>
```

8. Security Considerations

As this control package uses XML markup, implementation MUST address the security considerations of [RFC3023].

9. IANA Considerations

This specification instructs IANA to register a new Media Control Channel Framework Package and a new XML namespace.

9.1. Control Package Registration

Control Package name: msc-ivr-vxml/1.0

9.2. URN Sub-Namespace Registration

XML namespace: urn:ietf:params:xml:ns:msc-ivr-vxml

10. Change Summary

The following are the primary changes between the -04 of the draft and the -03 version.

- o Aligned with the -06 version of the basic ivr control package.
- o Simplified the structure so that only differences with the basic ivr control package are specified.
- o Specified how VoiceXML data is returned in a <dialogexit>
- o replaced <data> with <params>
- o updated references

The following are the primary changes between the -03 of the draft and the -02 version.

- o None

The following are the primary changes between the -02 of the draft and the -01 version.

- o Updated references.

The following are the primary changes between the -01 of the draft and the -00 version.

- o Changes in Basic IVR package version 02 applied.

11. Contributors

Asher Shiratzky from Radvision provided valuable support and contributions to the early versions of this document.

12. Acknowledgments

TBD

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

13.2. Informative References

[BASEIVRCP] Boulton, C., Melanchuk, T., and S. McGlashan, "A Basic Interactive Voice Response (IVR) Control Package for the Media Control Channel Framework", draft-boulton-ivr-control-package-06 (work in progress), February 2008.

[CCXML10] Auburn, R J., "Voice Browser Call Control: CCXML Version 1.0", W3C Working Draft (work in progress), January 2007.

[MCCF] Boulton, C., Melanchuk, T., McGlashan, S., and A. Shiratzky, "Media Control Channel Framework", draft-ietf-mediactrl-sip-control-framework-01 (work in progress), February 2008.

[MSML] Saleem, A., Xin, Y., and G. Sharratt, "Media Session Markup Language (MSML)", draft-saleem-msml-06 (work in progress), February 2008.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC4240] Burger, E., Van Dyke, J., and A. Spitzer, "Basic Network Media Services with SIP", RFC 4240, December 2005.

[RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.

[RFC4722] Van Dyke, J., Burger, E., and A. Spitzer, "Media Server Control Markup Language (MSCML) and Protocol", RFC 4722, November 2006.

[VXML20] McGlashan, S., Burnett, D., Carter, J., Danielsen, P., Ferrans, J., Hunt, A., Lucas, B., Porter, B., Rehor, K., and S. Tryphonas, "Voice Extensible Markup Language (VoiceXML) Version 2.0", W3C Recommendation, March 2004.

[VXML21] Oshry, M., Auburn, RJ., Baggia, P., Bodell, M., Burke, D., Burnett, D., Candell, E., Carter, J., McGlashan, S., Lee, A., Porter, B., and K. Rehor, "Voice Extensible Markup Language (VoiceXML) Version 2.1", W3C Recommendation, June 2007.

Authors' Addresses

Chris Boulton
Avaya
Building 3
Wern Fawr Lane
St Mellons
Cardiff, South Wales CF3 5EA

Email: cboulton@avaya.com

Tim Melanchuk
Rain Willow Communications

Email: tim.melanchuk@gmail.com

Scott McGlashan
Hewlett-Packard
Gustav III:s boulevard 36
SE-16985 Stockholm, Sweden

Email: scott.mcglashan@hp.com

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).