INTERNET-DRAFT
Intended Status: Standards Track
Expires: July 18, 2011

R. Shekh-Yusef
Avaya
C. Jennings
Cisco Systems
A. Johnston
Avaya
F. Audet
Skype

January 14, 2011

Action Referral in the Session Initiation Protocol (SIP)

draft-yusef-dispatch-action-ref-00

Abstract

This specification defines action referral that allows an application to make a high level request to a User Agent to perform an action, and let the the User Agent execute the action as it sees fit. Action referral uses the SIP REFER method with a Refer-To header field containing a URN, which indicates the requested action.

This specification also defines the option tag 'action-ref' to allow the UA to indicate its support for action referral.

Status of this Memo

Copyright and License Notice

Table of Contents

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

To simplify discussions of the REFER method and its extensions, the three terms below are being used throughout the document:

- REFER-Issuer: the UA issuing the REFER request

- REFER-Recipient: the UA receiving the REFER request

- REFER-Target: the UA designated in the Refer-To Uniform Resource Identifier (URI), which, for this specification, is a Uniform Resource Name (URN)

2. Introduction

The Session Initiation Protocol (SIP) [RFC3261] provides users with the ability to initiate, manage and terminate multimedia sessions. Many deployments have SIP applications in the SIP signaling path that get involved in the setup and management of these multimedia sessions.

A SIP application is a program running on a SIP network element that provides some value-added function to a user. Some of these applications need mechanisms to monitor or/and control a SIP User Agent (UA).

SIP already provides an extensible framework, (SIP)-Specific Event Notification [RFC 3265], by which SIP UAs can monitor remote UAs and get indications that certain events have occurred. For example, the following are widely used event packages:

- Dialog package - call states

- Registration package - phone status

- Conference package - conference status

SIP also provides a method for requesting UAs to perform certain task, i.e., REFER [RFC3515]. This REFER method is limited, as today it does not support the following:

- REFER does not allow for a UA to request another UA to respond to requests, e.g.,

    - A UA cannot request another UA to answer a call

    - A UA cannot request another UA to reject a call

- REFER does not allow for a UA to request another UA to invoke actions, e.g.,

    - REFER does not allow for a UA to request another UA to place a call on hold or to mute it

Action referral allows an application to make a high level request to a SIP [RFC3261] User Agent (UA) to perform an action or, and let the User Agent execute the action as it sees fit.

Action referral uses the SIP REFER method [RFC3515] with a Refer-To header field containing a URN [RFC2141] to indicate the requested action. Action referral is consistent with the SIP call control framework [RFC5850] and is a natural expansion of the Framework for Application Interaction [RFC5629] which allows for referral to SIP and HTTP resources.

OPEN QUESTION:

The REFER method seems to be already overloaded with various capabilities. Take a look at the following draft draft-worley-sip-many-refers-00 for more details.

Should a new SIP method be defined for this purpose?

3. Use Cases

This section describes these applications for which the Action Referral can be useful.

3.1. Third-Party/Proxy Application

Action referral is useful for a wide range of third party or proxy applications that need to remotely control or influence a User Agent, e.g. Contact Center environment. In pre-SIP environments, these environments have been using "Computer Telephony Integration": for example, traditional PBXs use CTI protocols such as CSTA [ECMA269] to provide this functionality. CSTA works fine for legacy PBXs with legacy phones but is problematic in a SIP environment. For example, SIP includes totally new capabilities such as presence and instant messaging. SIP also supports multiple users with multiple devices operating at once, and with complex User Interfaces. Furthermore, multiple applications may want to simultaneously wish to interact with the device. Because of the lack of a native mechanism to achieve such control for SIP, implementers have had to implement such techniques as mapping CSTA’s ASN.1 encoding to XML then encapsulate it into SIP INFO requests in order to tunnel it to a SIP B2BUA [ECMA323], which then maps it to proprietary device control protocols or to SIP with proprietary and incompatible extensions. This document provides a clean and native way to meet the requirements.

3.2. Loosely Coupled UAs

Action referral is useful for collections of loosely coupled User Agents which would like to present a coordinated user experience. Among other things, this allows User Agents which handle orthogonal media types but which would like to be present in a single conversation to add and remove each other from the conversation as needed. This is especially appropriate when coordinating conversations among organizers, general purpose computers, and special purpose communications appliances like telephones, Internet televisions, in-room video systems, electronic whiteboards, and gaming devices. For example using action referral, an Instant Messaging client could initiate a multi-player gaming session and an audio session to a chat conversation. Likewise a telephone could add an electronic whiteboard session to a voice conversation. Finally, a computer or organizer could cause a nearby phone to dial from numbers or URIs in a document, email, or address book, allow users to answer or redirect incoming calls without removing hands from the computer keyboard, place calls on hold, and join other sessions on the phone or otherwise.

3.3. Examples

In the following examples, Alice has two SIP clients; a PC with a soft client and a desk phone. Alice prefers to answer incoming audio calls on her desk phone.

3.3.1. Answer an audio call

In this example, Bob makes an audio call to Alice and the proxy forks the call to both of Alice’s clients. Alice decided that she wants to answer the call on her desk phone remotely using her soft client.

To do that, the soft client sends a REFER with a well-defined URN, which instructs the desk phone to answer the call. As a result, the desk phone sends a 200 OK to answer the incoming call, and the proxy then cancels the INVITE to the soft client.

```
   Alice                 Alice                Proxy                  Bob
    PC                 Desk Phone
    |                     |                     | INVITE Alice        |
    |                     |                     |<--------------------|
    |                     | INVITE Alice        |                     |
    |                     |<--------------------|                     |
    | INVITE Alice        |                     |                     |
    |<------------------------------------------|                     |
    | REFER Refer-To: urn:sip-action:call:answer|                     |
    |-------------------->|                     |                     |
    | 202                 |                     |                     |
    |<--------------------|                     |                     |
    | NOTIFY [Trying]     |                     |                     |
    |<--------------------|                     |                     |
    | 200 OK              |                     |                     |
    |-------------------->|                     |                     |
    |                     | 200 OK              |                     |
    |                     |-------------------->| 200 OK              |
    |                     |                     |-------------------->|
    |                     |                     | ACK                 |
    |                     |                     |<--------------------|
    |                     | ACK                 |                     |
    |                     |<--------------------|                     |
    | NOTIFY [200 OK]     |                     |                     |
    |<--------------------|                     |                     |
    | 200 OK              |                     |                     |
    |-------------------->|                     |                     |
    | CANCEL              |                     |                     |
    |<------------------------------------------|                     |
    |                     |                     |                     |
```

3.3.2. Answer an A/V call on two separate devices

In this example, Bob makes a video call to Alice. Alice interested in accepting the video using her soft client on her PC and accepting the audio on her desk phone.

To do that, the soft client sends a REFER with the URN urn:sip-action:audio:answer which asks the desk phone to return its audio answer to the incoming call. The desk phone returns its audio answer in the body of the sipfrag message in the NOTIFY request. The soft client then aggregates the audio SDP answer from the desk phone with the video answer from the soft client and sends it to Bob’s phone. As a result a video call is established, with the video streaming between Alice’s PC and Bob’s phone while the audio is streaming between Alice’s desk phone and Bob’s phone.

```
   Alice                Alice                Proxy                  Bob
    PC                Desk Phone
    |                     |                     | INVITE Alice [A/V]  |
    |                     |                     |<--------------------|
    |                     | INVITE Alice [A/V]  |                     |
    |                     |<--------------------|                     |
    | INVITE Alice [A/V]  |                     |                     |
    |<------------------------------------------|                     |
    | REFER Refer-To: urn:sip-action:audio:answer                     |
    |-------------------->|                     |                     |
    | 202                 |                     |                     |
    |<--------------------|                     |                     |
    | NOTIFY [100 Trying] |                     |                     |
    |<--------------------|                     |                     |
    | 200 OK              |                     |                     |
    |-------------------->|                     |                     |
    | NOTIFY [200 OK [Audio SDP answer]]        |                     |
    |<--------------------|                     |                     |
    | 200 OK              |                     |                     |
    |-------------------->|                     |                     |
    | 200 OK [A/V SDP answer]                   |                     |
    |------------------------------------------>|                     |
    |                     |                     | 200 OK [A/V]        |
    |                     |                     |-------------------->|
    |                     | CANCEL              |                     |
    |                     |<--------------------|                     |
    |                     |<======audio==============================>|
    |<============================video==============================>|
    |                     |                     |                     |
```

3.3.3. Click-to-Dial

In this example, while browsing the web on her PC, Alice clicks on Bob’s SIP URI, intending to initiate a call to Bob. Alice’s web browser passes the SIP URI to the soft client on Alice’s PC. The soft client is configured with the URI of Alice’s desk phone. A REFER is sent to the desk phone, which results in Alice’s desk phone initiating a call to Bob.

Alice decided to cancel the call to Bob because he is no answering. To do that, the soft client sends a REFER request with the URN urn:sip-action:call:terminate which results in Alice’s desk phone sending a CANCEL request to terminate the call to Bob.

```
          Alice                 Alice                  Bob
           PC                 Desk Phone
            |                     |                     |
            | REFER Refer-To: Bob |                     |
            |-------------------->|                     |
            | 202 Accepted        |                     |
            |<--------------------|                     |
            |                     | INVITE              |
            |                     |-------------------->|
            |                     | 180 Ringing         |
            |                     |<--------------------|
            |                     |                     |
            | REFER Refer-To: urn:sip-action:call:terminate
            |-------------------->|                     |
            | 202 Accepted        |                     |
            |<--------------------|                     |
            |                     |                     |
            |                     | CANCEL              |
            |                     |-------------------->|
            |                     |                     |
```

4. Overview

A prototypical action referral flow looks as per section 4.1 of [RFC3515]. The Refer-To URI in the REFER message includes a URN describing the action.

Action referral is sent to a GRUU when a specific instance of a UA is the desired target. When the action referral needs to be correlated to a specific dialog, the Target-Dialog header field is used [RFC4538].

Some primitives require a second dialog identifier (such as ConferenceCalls which causes the media from two dialogs to be mixed). To address the multiple dialogs need, one REFER is sent per dialog with a URN with a bridge action telling the phone to put the dialog on the bridge. While this requires multiple requests to be sent, the requests can be sent overlapped at the same time. This approach is flexible enough to allow adding and removing parties to the conference in very simple and straightforward way without changes to existing standard behavior.

4.1. URN Structure

The Namespace Identifier (NID) of the URN is intended to be in the formal space and assigned by IANA, as per the procedures of [RFC3406]. An alternative would be to use the service URN space [RFC5031]. Until this is resolved, this document will use the following namespace: "sip-action".

The Namespace Specific String (NSS) of the URN includes the action name, and may be followed by a semi-colon and additional action-specific parameters.

The action name might consist of a number of categories that start from the most general category and go down to more specific category, with the last value representing the request.

For example, the structure of the NSS of the urn:sip-action:call:answer has two categories: ’service’ and ’call’ and the request ’answer’.

The reason behind the above structure is to avoid the creation of a namespace with a very long list of unrelated actions.

The above structure allows grouping of related actions under one category to allow application to apply actions, e.g. enable/disable, to the whole category.

4.2. URN Categories

This document defines the following categories as part of the NSS of the URN:

- call: to allow access to call actions available on a SIP UA, e.g. answer a call, decline a call, ...

- conference: to allow access to conference actions available on a SIP UA, e.g. add, remove, ...

This document defines the following actions:

- Answer call - urn:sip-action:call:answer
- Terminate call - urn:sip-action:call:terminate
- Decline call - urn:sip-action:call:decline
  The REFER-Recipient returns 603 Decline
- Ignore call - urn:sip-action:call:ignore
  The REFER-Recipient stops ringing and clears the call
- Send call to VM - urn:sip-action:call:sendvm
  Stop ringing and send the call to VM.
- Hold call - urn:sip-action:call:hold
- Unhold call - urn:sip-action:call:unhold
- Mute call - urn:sip-action:call:mute
- Unmute call - urn:sip-action:call:unmute
- Conference - urn:sip-action:conference:add
  - urn:sip-action:conference:remove

Note that the very important "Make call" CTI primitive does not require a action referral URN since it is accomplished by sending a normal REFER with a URI identifying the resource (e.g., a sip, sips or tel URI).

This specification defines the above list as an initial set of URNs, to be registered with IETF, for use with this specification.

In order to add any new action URN to the list above, it must go through the "IETF review" process as defined in RFC 5226.

5. User Agent Behavior

5.1. Dialog usage

This document attempts to avoid using multiple dialog usages, for the reasons described in [RFC5057]. Therefore, this document will make use of the GRUU [RFC5627], and the Target-Dialog header field [RFC4538] to associate an existing INVITE usage with a REFER arriving on a new dialog to facilitate authorization of that REFER.

In many use cases of action referral, receiving notifications about the status of a REFER request are superfluous, as the Refer issuer often maintains a long duration subscription to the dialog package [RFC4235]. Suppression of the REFER notifications is done with the norefersub option-tag, defined in section 7 of [RFC4488]. When the norefersub option tag is present, a REFER request which would have created a new subscription and dialog becomes a standalone transaction instead, eliminating a multiple dialog usage. Each such standalone REFER transaction use a new (unique) Call-ID header field value.

In the most common usage, the controller maintains a long duration subscription to the dialog package, and sends REFER requests in separate dialogs. Each REFER would include the norefersub option-tag in a Supported header field.

In some cases, the controller does not maintain a dialog package subscription for the REFER-Recipient. This might be the case for a "webdialer" or other application which associates with other UAs on an adhoc and intermittent basis. An initial REFER request is sent to start a new dialog, which is followed by notifications for the refer event type (the norefersub option-tag is not used in this case).

5.2. Capabilities Indications

A UA compliant to this specification SHOULD indicate its support by including the option tag ’action-ref’ in the Supported header of all requests it sends.

The UA MAY also indicate its support for this action referral by adding the ’action-ref’ value to the extentions parameter in the Contact header field of its REGISTER request, as described in [RFC 3840].

5.3. Addressing the relevant parties

REFER requests contain a number of URIs which need to address the appropriate parties. A list of the relevant fields include the Request-URI, To URI, From URI, Contact URI, Refer-To URI, and the Referred-By URI, as well as the Target-Dialog itself. This section attempts to clarify what needs to be placed in each field.

Action referral applies to dialogs or sessions on a specific UA, which requires the use of GRUU [RFC5627] for a single UA. Contact URIs for a UA can be discovered by subscribing to the registration package [RFC3680] for the relevant AORs.

The To header field in the REFER request normally contains the same URI as in the Request-URI. The From identifies the AOR of the controller. The Refer-To URI is the action referral URN.

Many uses of action referral require that the REFER-Recipient take some action in the context of an existing dialog. For example, the controller might want the REFER-Recipient to terminate an existing dialog. To select the appropriate dialog from which to source the request, the Target-Dialog header specified in [RFC4538] is used.

5.4. Monitoring Device Specific Events

Some applications need a mechanism to monitor events on a SIP UA that are device specific.

The following is a list of some these device specific events:

1. Hook status
2. Transducer status (handset, headset, speaker)
3. Active call
4. Volume status
5. Mute status
6. MWI light status
7. IM light status

A separate document that defines a new event package will be created to address this need.

5.5. Request Authorization

When a UA receives a request to invoke an action, it needs to authorize that request. Some requests can be authorized based on the identity of the sender, the request method, local policy, etc.

All action referral requests MUST be challenged using the digest authentication mechanism described by [RFC 3261]. In most cases, the same user will be logged in to the different devices using the same credentials provided in the REGISTER requests.

6. Call Flows

This sample provides non-normative sample calls flows for some of the actions listed in Section 4. It is important to understand that the actual "realization" of the action (i.e., the actual procedures invoked) are the sole responsibility of the Refer-Recipient. This document in no way attempts to standardize those procedures, and the call flow below are merely examples.

In all cases, the "controller" (i.e., the Refer-Issuer) could be Alice’s PC, PDA, or a third party application. The controlled device is Alice’s phone (i.e., the Refer-Recipient). The Refer-Target is obviously the action referral URN. In all cases, it is assumed that the controller is subscribed to Alice’s Phone’s dialog package.

The call flows in this document use the following conventions. The dialog each message is sent in is shown on the left hand side. Selected Request-URI and header fields are shown. The contents of message bodies are shown for dialog-info+xml, sdp, and sipfrag message bodies. For responses, the method is shown in parentheses. For reference, the messages are labeled F1, F2, etc.

6.1. Answer Call Operation

In message 1, Bob makes a call to Alice’s Phone. A notification of "trying" is sent to the controller. Alice’s phone automatically sends a "ringing" to Bob. Another notification of "early" is then sent to the controller. The controller then tells the phone to answer the call. Alice’s phone sends a notification of "confirmed" to the controller.

```
       Controller                       Alice                        Bob
           |<<< Controller subscribed  >>>|                            |
           |<< to Alice’s dialog events >>|                            |
   dialog1 |                              | F1  INVITE sip:Alice-AOR   |
           |                              |<---------------------------|
   dialog2 | F2 NOTIFY sip:Controller-GRUU|                            |
           |    dialog-info+xml: dialog1=trying                        |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F3  200 (NOTIFY)             |                            |
           |----------------------------->|                            |
   dialog1 |                              | F4  180 (INVITE)           |
           |                              |--------------------------->|
   dialog2 | F5 NOTIFY sip:Controller-GRUU|                            |
           |    dialog-info+xml: dialog1=early                         |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F6  200 (NOTIFY)             |                            |
           |----------------------------->|                            |
           |                              |                            |
   dialog3 | F7  REFER sip:Alice-GRUU     |                            |
           |     To: sip:Alice-GRUU       |                            |
           |     Refer-To: urn:sip-action:call:answer                  |
           |     Target-Dialog: dialog1   |                            |
           |----------------------------->|                            |
           |                              |                            |
   dialog3 | F8  202 (REFER)              |                            |
           |<-----------------------------|                            |
   dialog1 |                              | F9  200 (INVITE)           |
           |                              |--------------------------->|
           |                              |                            |
   dialog1 |                              |     F10 ACK                |
           |                              |<---------------------------|
   dialog2 | F11 NOTIFY sip:Controller-GRUU                            |
           |     dialog-info+xml: dialog1=confirmed                    |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F12 200 (NOTIFY)             |                            |
           |----------------------------->|                            |
```

In the following flow, the application is in the signaling path between Alice and Bob and is aware of a dialog that was established between the UAs.

Alice calls Bob, and Bob’s UA is ringing. At this stage, the application decides that it wants to ask Bob’s UA to answer the call. The application sends a REFER request asking Bob’s UA to answer the call, which results on Bob’s UA sending an 200 OK and a call is established with Alice.

```
            Alice                 Application                  Bob
              | F1 INVITE              |                        |
              |----------------------->|                        |
              |                        | F2 INVITE              |
              |                        |----------------------->|
              |                        | F3 180 Ringing         |
              |                        |<-----------------------|
              | F4 180 Ringing         |                        |
              |<-----------------------|                        |
              |                        |                        |
              |                        | F5 REFER sip:Bob-GRUU  |
              |                        |    To: sip:Bob-GRUU    |
              |                        |    Refer-To:           |
              |                        |     urn:sip-action:call:answer
              |                        |    Target-Dialog: dialog1
              |                        |----------------------->|
              |                        | F6 202 (REFER)         |
              |                        |<-----------------------|
              |                        |                        |
              |                        |                        |
              |                        | F7 200 OK              |
              |                        |<-----------------------|
              | F8 200 OK              |                        |
              |<-----------------------|                        |
              | F9 ACK                 |                        |
              |----------------------->|                        |
              |                        | F10 ACK                |
              |                        |----------------------->|
              |                        |                        |
              |                        |                        |
              |                        | F11 NOTIFY             |
              |                        |<-----------------------|
              |                        |                        |
              |                        | F12 200 OK             |
              |                        |----------------------->|
              |                        |                        |
```

6.2. Terminate Call Operation

Terminate Call is a perfect example of an action whose treatment (and consequently, the resulting call flow) depends on the situation, for example, the state of the dialog between the remote parties.

Alice’s Phone and Bob are currently in an established dialog. The controller tells Alice’s phone to terminate the call with Bob’s phone.

```
       Controller                    Alice                          Bob
           |<< Controller subscribed to >>|<<< Established dialog1 >>>>|
           |<<< Alice’s dialog events >>>>|                            |
           |                              |                            |
   dialog3 | F1  REFER sip:Alice-GRUU     |                            |
           |     To: sip:Alice-GRUU       |                            |
           |     Refer-To: urn:sip-action:call:terminate               |
           |     Target-Dialog: dialog1   |                            |
           |----------------------------->|                            |
           |                              |                            |
   dialog3 | F2  202 (REFER)              |                            |
           |<-----------------------------|                            |
   dialog1 |                              | F3  BYE sip:Bob-GRUU       |
           |                              |--------------------------->|
           |                              |                            |
   dialog1 |                              | F4  200 (BYE)              |
           |                              |<---------------------------|
           | F5  NOTIFY sip:Controller-GRUU                            |
           |     dialog-info+xml: dialog2=local-bye                    |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F6  200 (NOTIFY)             |                            |
           |----------------------------->|                            |
```

Terminate Call in Established Dialog Call Flow Example

If Alice’s Phone and Bob are in an early dialog with Bob calling Alice, the call flow could be as follows.

```
          Controller                    Alice                        Bob
           |<< Controller subscribed to >>|                            |
           |<<< Alice’s dialog events >>>>|                            |
   dialog1 |                              | F1  INVITE sip:Alice-AOR   |
           |     (dialog2)                |<---------------------------|
           |                              |                            |
   dialog2 | F2 NOTIFY sip:Controller-GRUU|                            |
           |     dialog-info+xml: dialog1=trying                       |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F3  200 (NOTIFY)             |                            |
           |----------------------------->|                            |
   dialog1 |                              | F4  180 (INVITE)           |
           |                              |--------------------------->|
   dialog2 | F5 NOTIFY sip:Controller-GRUU|                            |
           |    dialog-info+xml: dialog1=early                         |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F6  200 (NOTIFY)             |                            |
           |----------------------------->|                            |
           |                              |                            |
   dialog3 | F7  REFER sip:Alice-GRUU     |                            |
           |     To: sip:Alice-GRUU       |                            |
           |     Refer-To: urn:sip-action:call:terminate               |
           |     Target-Dialog: dialog1   |                            |
           |----------------------------->|                            |
           |                              |                            |
   dialog3 | F8  202 (REFER) (dialog3)    |                            |
           |<-----------------------------|                            |
   dialog1 |                              | F9  480 (INVITE)           |
           |                              |--------------------------->|
           |                              |                            |
   dialog1 |                              | F10 ACK                    |
           |                              |<---------------------------|
   dialog2 | F11 NOTIFY (Controller-GRUU) |                            |
           |     dialog-info+xml: dialog1=rejected                     |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F12 200 (NOTIFY)             |                            |
           |----------------------------->|                            |
```

Terminate Call in Early Dialog Call Flow Example

If Alice’s Phone and Bob are in an early dialog with Alice calling Bob, the call flow could be as follows.

```
          Controller                    Alice                        Bob
           |<< Controller subscribed to >>|                            |
           |<<< Alice’s dialog events  >>>|                            |
   dialog1 |                              | F1  INVITE sip:Bob-AOR     |
           |                              |--------------------------->|
   dialog2 | F2 NOTIFY sip:Controller-GRUU|                            |
           |    dialog-info+xml: dialog1=trying                        |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F3  200 (NOTIFY)             |                            |
           |----------------------------->|                            |
   dialog1 |                              | F4  180 (INVITE)           |
           |                              |<---------------------------|
   dialog2 | F5 NOTIFY sip:Controller-GRUU|                            |
           |    dialog-info+xml: dialog1=early                         |
           |<-----------------------------|                            |
           |                              |                            |
   dialog2 | F6  200 (NOTIFY)             |                            |
           |----------------------------->|                            |
           |                              |                            |
   dialog3 | F7  REFER sip:Alice-GRUU     |                            |
           |     To: sip:Alice-GRUU       |                            |
           |     Refer-To: urn:sip-action:call:terminate               |
           |     Target-Dialog: dialog1   |                            |
           |----------------------------->|                            |
           |                              |                            |
   dialog3 | F8  202 (REFER)              |                            |
           |<-----------------------------|                            |
   dialog1 |                              | F9  CANCEL                 |
           |                              |--------------------------->|
   dialog1 |                              | F10 200 (CANCEL)           |
           |                              |<---------------------------|
           |                              |                            |
   dialog1 |                              | F11 487 (INVITE)           |
           |                              |<---------------------------|
   dialog1 |                              | F12 ACK                    |
           |                              |--------------------------->|
           |                              |                            |
   dialog1 | F13 NOTIFY sip:Controller-GRUU                            |
           |     dialog-info+xml: dialog1=rejected                     |
           |<-----------------------------|                            |
   dialog2 | F14 200 (NOTIFY)             |                            |
           |----------------------------->|                            |
```

Terminate Call Initiated Call Flow Example

7. Security Considerations

The functionality described in this document allows an authorized party to manipulate SIP sessions and dialogs in arbitrary ways. Any user agent that accepts these types of requests needs to be very careful in who it authorizes to send these types of requests. The same security considerations as [RFC3515] apply.

8. IANA Considerations

TODO: Need to register urn namespace according to procedures of [RFC3406].

9. Acknowledgments

TODO

10. References

10.1. Normative References

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

10.2 Informative References

[RFC2141] Moats, R., "URN Syntax", RFC 2141, May 1997.

[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.

[RFC3406] Daigle, L., van Gulik, D., Iannella, R., and P. Faltstrom, "Uniform Resource Names (URN) Namespace Definition Mechanisms", BCP 66, RFC 3406, October 2002.

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[RFC3680] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Registrations", RFC 3680, March 2004.

[RFC4235] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", RFC 4235, November 2005.

[RFC4488] Levin, O., "Suppression of Session Initiation Protocol (SIP) REFER Method Implicit Subscription", RFC 4488, May 2006.

[RFC4538] Rosenberg, J., "Request Authorization through Dialog Identification in the Session Initiation Protocol (SIP)", RFC 4538, June 2006.

[RFC5031] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services", RFC 5031, January 2008.

[RFC5057] Sparks, R., "Multiple Dialog Usages in the Session Initiation Protocol", RFC 5057, November 2007.

[RFC5629] Rosenberg, J., "A Framework for Application Interaction in the Session Initiation Protocol (SIP)", RFC 5629, October 2009.

[RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", RFC 5627, October 2009.

[RFC5850] Mahy, R., "A Call Control and Multi-party usage framework for the Session Initiation Protocol (SIP)", RFC 5850, May 2010.

[ECMA269] ECMA International, "Services for Computer Suported Telecommunications Communications Applications (CSTA) Phase III", Standard ECMA-269, December 2006.

[ECMA323] ECMA International, "XML Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III", Standard ECMA-323, December 2006.

[TR87] ECMA International, "Using CSTA for SIP Phone User Agents (uaCSTA)", Technical Report TR/87, June 2004.

Author's Addresses

Rifaat Shekh-Yusef
Avaya
250 Sidney Street
Belleville, Ontario K8P 5B7
Canada

Phone: +1-613-967-5267
Email: rifatyu@avaya.com

Cullen Jennings
Cisco Systems
170 West Tasman Drive
Mailstop SJC-21/2
San Jose, CA 95134
USA

Phone: +1-408-902-3341
Email: fluffy@cisco.com

Alan Johnston
Avaya
St. Louis, MO 63124
USA

Email: alan.b.johnston@gmail.com

Francois Audet
Skype
USA

Email: francois.audet@skype.net