Internet Draft Bert Culpepper

draft-culpepper-sipping-app-interact-reqs-04.txt

October 8, 2003
Expires: April 2004

Robert Fairlie-Cuninghame
Nuera Communications, Inc.

Session Initiation Protocol Based Application Interaction Requirements

Status of this Memo

Abstract

This document defines the high level requirements for a framework and/or one or more mechanisms that support user interaction, via SIP-based user agents, with applications residing on remote network servers. The requirements in this document address the overall features of such a system, without regard to its architecture.

SIP currently supports media-based application interactions using methods such as speech, video and end-to-end telephony-related tones; however, it is desired that more general application interaction models are defined, especially those that are not restricted to the media plane. In addition, it is desired that an application be able to present the user with application-specific user interfaces and information. The user agent should also be able to generate activity indications back to an application to communicate actions on physical or logical user interfaces. The document also defines a number of topic-related terms to assist in disambiguating discussions of the issues.

1. Conventions Used In This Document

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 [2].

2. Motivation

Telecommunications services in circuit-switched networks have utilized end-user indications as the means for users to interact with the services while users are engaged in a call. These end-user indications, such as those produced by a user pressing keys, are sent end-to-end through each of the network entities participating in the call. As communications services move to IP networks, the ability for users to interact with their communications services in a real-time like fashion must also follow. Unlike the legacy circuit-switched networks, nodes hosting many services in IP networks infrequently reside along the path taken by the media.

Users of communications services have become accustomed to control of services through interaction via the communications terminal. The traditional means by which users interact with their communications services in legacy networks is via the use of DTMF generated as a result of the user pressing a key on the terminal's keypad. Because of this, there is a significant desire to duplicate the use of DTMF to support user interaction with services tightly associated with IP communications sessions. The Internet network model for communications separates session control from the session media in that the entities involved in session control are not necessarily tightly coupled to the entities that process media. As the transport of DTMF is provided for in IP networks as a media stream [3], access to these user indications by the network entities involved in the session control is awkward. In addition, limiting user interaction with communications services to input devices that emulate the traditional telephone keypad constrain the user devices unnecessarily.

In addition to legacy application interaction methods such as DTMF, there is a desire for new interaction methods that support the use of web pages, keyboards and other user devices used to access the Internet to be available. These new interaction methods should operate, from a user's perspective, in a consistent and seamless manner with legacy methods such as DTMF.

It is for these reasons a different mechanism than that based on legacy networks is needed to transport user indications for application interaction in IP networks.

The Session Initial Protocol (SIP) [4] has been chosen as the session control protocol for multimedia session establishment within the general Internet and in many other IP-based networks. Because of this choice, it is desirable to have one or more mechanisms supporting user application interaction that works with SIP. As SIP deals with session control and not media transport, the mechanisms should not be limited to the media plane.

3. Use Cases

Network-based services for SIP-based communications, while a SIP session is ongoing, are unlikely to be compelling without the ability for a user to interact with the service. Currently, once a session is established, users are limited to the functions their terminal supports, and network-based services are limited to SIP signaling events.

Some network-based communications services that can benefit from an Application Interaction framework include Pre-paid and Post-paid Calling Cards. These applications require a user to provide an account number and Personal Identification Number (PIN) when accessing the service. The user typically provides this information using the keypad on their telephone, and the information is communicated to the service/application using DTMF. This example, when hosted in an IP network, does not require any new IP functionality, as the end point the user is interacting with at the time of service invocation, is the service entity. However, these services many times have "mid-call" features that are invoked via the user's terminal, and when the media has been redirected away from the service entity.

Another network-based service that can benefit is Mid Call Transfer. This service typically utilizes a key sequence followed by a destination address (telephone number). Here again, the service entity in an IP network will not be in the media path between the end points when the service is accessed.

A SIP-based Application Interaction Framework will also enable new services that take advantage of the IP network capabilities and protocols, without requiring service-specific knowledge to be present in end user devices and intermediate network entities not involved in providing the specific service.

4. Terminology

The following acronyms and terms are used in this document.

Requestor: The agent responsible for requesting user indications or application presentations from the Reporter. The Requestor is normally associated with the Application Entity.

Reporter: The agent responsible for detecting and reporting user activity indications; and optionally presenting a user application component to the user.

UA: SIP User Agent [4].

User Activity Indication (UAI): The message(s) containing the data associated with the reporting of discrete user indications, for instance, a mouse click or button press. It refers to indications relating to discrete stimulus-based interactions rather than media stream-based interactions such as voice or video.

Physical User Interface: The collection of physical input and presentation devices possessed by a device, for instance, a display, speaker, microphone and/or dialpad.

Logical User Interface (LUI): The logical collection of user interface components (see definition below) used by a user to interact with a group of (explicitly) cooperating applications. A logical user interface is independent of all other application interactions occurring on the device.

User Interface Component (UIC): A component (physical or otherwise) used for application interaction. Examples of UICs include: a web-page window, a media-based video window, a speaker, microphone or a key-based input device. A UIC may only generate user activity indications when the user is interacting with the associated logical user interface.

Presentation-based Interaction: A presentation-based UIC will present an application-supplied user interface (or simply application-supplied information) to the user. A presentation-based component will also commonly allow a user to interact directly with the supplied interface through stimulus-based methods. An example is a web-page window & pointing device or simply a display screen with no associated input device.

Media-based Interaction: Media-based interaction refers to user input supplied via UICs that process media (e.g., audio). Media-based UI components allow bi-directional or unidirectional interaction through the media plane, for instance, a speaker or a microphone (unidirectional) or a speaker & microphone combination (bi-directional). Media-based UICs may present application-supplied user interfaces or information to the user; however, these components do not generate discrete user activity indications and merely relay un-interpreted media streams to/from the application. The resulting framework should not alter the normal SIP session semantics but simply allow the media-based SIP session to be associated with a UIC within a logical user interface.

Input-based Interaction: Input-based interaction refers to user input supplied via UICs that do not present an application-supplied interface to the user but rather correspond to a (usually physical) interface possessed by the device, for instance, a dialpad or keyboard. Input-based UICs generate UAIs in response to user actions.

5. End-to-end Verses Asynchronous User Activity Indications

The end-to-end user activity indications currently supported in IP networks require "workarounds" in SIP networks so that applications along the session signaling path have access to the indications. The current solution requires "DTMF forking" be supported by the endpoint, or requires the receiving entity, when it's not the final destination for the sessionÆs media, to re-generate the indication towards the destination. In many scenarios, the indications meant for the application are not used at the destination.

UAIs needed for application interaction on the other hand, are only needed between an endpoint/user and the application within the network. Using end-to-end mechanisms for application interaction, when the application is not itself an endpoint in the session, is problematic as indicated above.

6. General Requirements

R1: The framework MUST support the collection of device/user input generated in the context of a SIP dialog or conversation-space.

R2: The framework MUST support transporting UAIs to network elements independently of the media plane.

R3: The transport mechanism must be sensitive to the limited bandwidth constraints of some signaling planes; for instance, reliability through blind retransmission is not acceptable.

R4: It MUST be possible for multiple network entities or applications to request and receive user activity indications from a user's terminal independently of each other.

R5: The framework MUST provide a means for a network application/entity to indicate its desire to receive user activity indications and/or to present an application interface on the user's terminal.

R6: The framework MUST support a means for a requestor to be able to determine the UICs that are available to the user's UA and/or terminal for application use.

The intent of this requirement is that the presence of a message header, header parameter, or other indicator will be used to indicate the supported UICs of an application entity and SIP UA. For backwards compatibility, the lack of a message header or parameter may result in an assumption that a UA only possesses a minimal UIC such as a traditional telephone keypad.

R7: The framework MUST provide a means for a SIP UA to indicate its capability/intent to fulfill a request for user activity indications.

Here again, the intent of this requirement follows that of R6.

R8: The framework MUST provide a means whereby the Requestor can indicate its desire to only receive a subset of the supported UAIs for any non-trivial UIC.

R9: The framework MUST NOT generate UAIs unless implicitly or explicitly requested by an entity.

R10: The framework SHOULD support devices with a wide range of user interfaces for both presentation-based and input-based interaction modes, for instance, it must support devices that possess a display UIC, as well as those that do not; from devices that only have physical buttons to those that only have display-based pointing devices.

R11: The framework MUST be extensible so that a variety of non key-based user activity indications can be supported now or in the future, for instance, sliders, dials, switches, local voice-commands, hyperlinks, biometrics, etc.

R12: The framework MUST support reliable delivery of UAIs at least as good as the session control protocol.

R13: The framework MUST ensure that the receiver of user activity indications (i.e., the Requestor) can determine their original order of occurrence and detect any missing indications.

R14: The framework MUST allow the user to know which application is associated with each UIC.

R15: The framework MUST provide a mechanism that allows users to have assurances that the user input they are providing is only seen by the application that created the UIC or requested UAIs from the UIC.

R16: The framework must support the ability for each UIC to be associated with a separate LUI. Each LUI may be associated with the same or different applications. For example, a user may want to interact with a voice-recording application and a prepaid calling application within the same call but allow each application to use a different LUI.

R17: The framework MUST allow UICs created through the prescribed mechanism(s) to be updated or removed as desired by the creating application entity.

R18: The framework SHOULD support the termination, by the User Agent, of application interaction resources established via the framework when they are no longer associated with a SIP dialog. There may be cases in which a user authorizes the persistence of application interaction resources beyond the life of the SIP dialog that established them.

R19: For user activity indications, the framework SHOULD support mechanisms to relate the time of occurrence of UAIs to the media in one or more media streams.

Because a primary goal of the framework is to decouple the transport of UAIs from the media transport, it is not practical to require synchronization between UAIs and media. For scenarios where tight synchronization is required, the UAIs should be transported with the media itself. For example, the transport of DTMF generated as a result of a key press on a the keypad of a telephone should be sent as specified in RFC2833 in the same media stream as the media requiring its synchronization. In addition, since UAIs relayed using the framework will not be tightly coupled with a sessionÆs media, the utility of UAI timestamps is an implementation decision. However, some applications may find this capability useful for their services.

R20: The framework MUST provide a mechanism that allows the Requestor to indicate to the Reporter that UAIs for the associated UIC MUST NOT be sent/copied using any other means. The framework MUST provide a mechanism for the Reporter to refuse such a request if it cannot fulfill this guarantee.

This allows the Requestor to be assured of a "private" UIC regardless of the Reporter's level of implementation or user interface.

7. Key-Based Input Specific Requirements

K1: The framework MUST address the collection of DTMF-based UAIs.

K2: The framework MUST address the collection of UAIs for device- and/or user- specific buttons.

K3: For key-based indications, the framework MUST provide some form of indication of key press duration.

K4: For key-based indications, the framework MUST provide some form of indication of a key-press' occurrence in time relative to other key presses.

8. Desirables

D1: The framework SHOULD allow a UA to indicate relative preferences amongst its various supported UICs.

D2: To help manage feature interaction, the framework SHOULD also allow a means of prioritizing user interface component requests from multiple network entities within a single SIP dialog.

9. Acknowledgements

The authors would like to acknowledge the detailed comments and additions to this document by Jonathan Rosenberg of Dynamicsoft,

Inc. and Eric Chueng of AT&T Labs.

10. Authors

Robert Fairlie-Cuninghame
Nuera Communications, Inc.
50 Victoria Rd
Farnborough, Hants GU14-7PG
United Kingdom
Phone: +44-1252-548200
Email: rfairlie@nuera.com

Bert Culpepper
Phone: +1-407-314-2617
Email: bertculpepper@netscape.net

11. References

1 S. Bradner, "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

2 S. Bradner, "Key words for use in RFCs To Indicate Requirement Levels," RFC 2119, Internet Engineering Task Force, Mar. 1997.

3 H. Schulzrinne and S. Petrack, "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals," RFC 2833, Internet Engineering Task Force, May 2000.

4 J. Rosenberg, H. Schulzrinne, et. al., "SIP: Session Initiation Protocol", RFC 3261, June 2002.