

WIDEX
Internet-Draft
Intended status: Informational
Expires: July 12, 2007

V. Stirbu
Nokia
D. Raggett
W3C/Volantis
January 8, 2007

Widget Description Exchange Service (WIDEX) Requirements
draft-ietf-widex-requirements-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 July 12, 2007.

Copyright Notice

Copyright (C) The Internet Society (2007).

Abstract

This document defines functional requirements for a framework and a protocol used to support XML-based user interfaces, where the user interface and application logic may be on different machines, and coupled via an exchange of XML DOM events and update/mutation operations.

Requirements Language

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](#)].

Table of Contents

| | | |
|----------------------|--|--------------------|
| 1. | Introduction | 3 |
| 2. | Widex Entities | 3 |
| 3. | Widex Terminology | 3 |
| 3.1. | User Interface | 3 |
| 3.2. | Remote User Interface | 4 |
| 3.3. | Multimodal Interaction | 4 |
| 3.4. | Widex Objects | 4 |
| 3.5. | Transport Protocol | 5 |
| 3.6. | Simple vs. Complex User Interfaces | 5 |
| 3.7. | Session | 5 |
| 3.8. | State | 5 |
| 4. | Requirements | 6 |
| 4.1. | Architecture Requirements | 6 |
| 4.2. | Service Discovery and Session Setup Requirements | 6 |
| 4.3. | Widex Objects Requirements | 6 |
| 4.4. | Transport Requirements | 7 |
| 5. | IANA Considerations | 7 |
| 6. | Security Considerations | 7 |
| 6.1. | Privacy Considerations | 7 |
| 7. | Acknowledgements | 8 |
| 8. | References | 8 |
| 8.1. | Normative References | 8 |
| 8.2. | Informative References | 9 |
| | Authors' Addresses | 9 |
| | Intellectual Property and Copyright Statements | 11 |

1. Introduction

With the Internet reaching out to more and more devices, people are increasingly expecting to have access to services at anytime, from anywhere and using any device. Such services are being developed using Web technologies such as XML and distributed across the network rather than resident on any one device.

2. Widex Entities

The following picture shows the primary Widex entities in a simple and basic architecture.

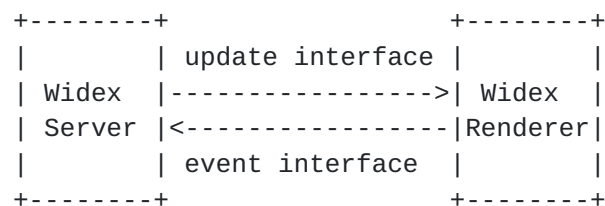


Figure 1: Widex Entities

The two primary Entities are described as follows:

Widex Server (WS):

The entity that holds the application logic.

Widex Renderer (WR):

The entity that renders the user interface.

There might be several Widex Servers in the same device, one for each application that can export the user interface and for each Widex Server there might be several Widex Renderers which are rendering the user interface.

3. Widex Terminology

The terminology and definitions detailed below include both terms that, besides the Widex Entities are used in the requirements section of this document.

3.1. User Interface

In this context, the user interface is understood as XML [[XML1.0](#)] data describing the user interface. Typically, the XML data is manipulated as levels 2 and 3 in the W3C Document Object Model (DOM),

see [[DOM2.Core](#)], [[DOM3.Core](#)], and [[DOM2.Events](#)] (W3C has yet to complete work on DOM3 events). Style information associated with the user interface can be manipulated via the DOM, see [[DOM2.Style](#)].

[3.2.](#) Remote User Interface

The Remote User Interface (RUI) is a technique in which a user interface is rendered on another device than the one that runs the application logic while keeping the user interface synchronised with the application logic.

[3.3.](#) Multimodal Interaction

Multimodal interaction provides the user with multiple modes of interfacing with a system beyond the traditional keyboard and mouse input/output. The most common such interface combines a visual modality (e.g. a display, keyboard, and mouse) with a voice modality (speech recognition for input, speech synthesis and recorded audio for output). However other modalities, such as pen-based input or haptic input/output, may be used.

For example, W3C has developed a Multimodal Interaction Framework [[W3C.NOTE-mmi-framework-20030506](#)] that is intended as a basis for developing multimodal applications in terms of markup, scripting, styling and other resources.

[3.4.](#) Widex Objects

One of the goals is to define Widex Objects (WO) that are used to convey information about interface updates and events. Widex Objects are used to keep the rendered user interface synchronised with the application logic.

There are two types of Widex Objects:

WO Update (WO.Update):

WO.Update object contain description of changes to XML DOM trees. The updates can target individual nodes in order to update their properties and attributes, or can target parts of the DOM tree in order change its structure, e.g. add/delete/replace nodes or branches.

WO Event (WO.Event):

WO.Event objects are primarily used to carry events triggered on the Widex Renderer side so that they can be caught by the application logic event handlers on the Widex Server side. Not all events need to be forwarded to the application logic, as some events have local scope and may be intended for default event

handlers or for local scripted event handlers. The application logic in the Widex Server may only be interested in specific events.

WO.Event object may carry data according to the type of the event, e.g. application defined events with structured payloads as per Extensible Multi-Modal Annotations [[W3C.WD-emma-20050916](#)], which uses XML for annotated interpretations of user input.

A secondary use of WO.Event objects is where the application logic itself raises events that may be caught by event handlers associated with the remote user interface, see for example Web Applications 1.0 [[WhatWG.WebApps1.0](#)].

[3.5.](#) Transport Protocol

The Transport Protocol is a protocol that just transports the Widex Objects as a string of bits, without looking at them.

[3.6.](#) Simple vs. Complex User Interfaces

Simple User Interface:

A simple user interface allows the user interface to be represented with only one XML DOM object. Typically, a simple user interface is associated with a single modality, e.g. vision modality for graphic user interfaces.

Complex User Interface:

A complex user interface may include scripted client-side event handlers or there can be more than one XML DOM in the user interface, e.g. different DOMs for different modalities, but see also SVG's XML Binding Language [[W3C.WD-sXBL-20050815](#)] for the role of shadow DOMs.

[3.7.](#) Session

The Widex Session is initiated between a Widex Server and a Widex Renderer for exchanging information about user interface updates (e.g. WO.Update) and events (e.g. WO.Event) in order to control applications remotely. A Widex Session relate to a single User Interface, which can be simple or complex.

[3.8.](#) State

The Widex Server is maintaining the state of the user interface. Upon ungraceful session break, the Widex Renderer retrieves the current user interface state from the Widex Server when the session is re-established.

Certain UI markup languages allow the remote renderers to operate in disconnected mode. In such cases, the application is responsible for keeping the state on the Widex Server consistent with the changes occurred in the Widex Renderer when it operated in disconnected mode.

4. Requirements

4.1. Architecture Requirements

- o The service discovery and session setup component **MUST** be replaceable.
- o The transport component **MUST** be replaceable.
- o Keeping the user interface and the application logic synchronised **MUST** occur at a fairly loose level that allows for a simple approach to propagating changes.
- o The framework **SHOULD** be stateless.
- o The framework **SHOULD** be consistent with the W3C Multimodal Architecture [[W3C.WD-mmi-arch-20060414](#)].

4.2. Service Discovery and Session Setup Requirements

- o The service discovery mechanism **MUST** be able to discover both Widex Renderers and Widex Servers.
- o The service discovery mechanism **MUST** be able to negotiate the capabilities of both Widex Renderers and Widex Servers, e.g. supported devices physical characteristics, supported UI markup languages, etc.
- o The session setup mechanism **MUST** be able to initiate session establishment from both Widex Renderers and Widex Servers, e.g. remote user interface pull and push.

4.3. Widex Objects Requirements

- o The Widex Objects **MUST NOT** be aware of the semantics of the UI markup language.
- o The Widex Objects **MUST** support client initiated events.
- o The Widex Objects **MUST** support server initiated updates.
- o The Widex Objects **MUST** contain only information meaningful in the context of the Widex Session.
- o The Widex Objects **MUST** indicate the target XML DOM tree when Complex User Interfaces are involved.
- o The Widex Objects **MAY** support multimodal interaction, and it is the responsibility of the application to synchronise modalities and not that of the Widex protocol.

4.4. Transport Requirements

- o The Widex protocol MUST deliver Widex Objects in the order determined by the originator regardless of the underlying network topology. The order is relevant within a single Widex Session. The co-ordination between several Widex Sessions in a Widex Server is outside the scope of this document.
- o The Widex protocol MUST handle each modality within the context of a single Widex Session.
- o The Widex protocol MUST provide reliable delivery of Widex Objects. If this proves to be impractical in a given situation, the protocol MUST provide the means for application to detect such failures.
- o The Widex protocol MUST tolerate limitations in available bandwidth.
- o The Widex protocol MUST tolerate delays caused by network induced latency. Latency and time-outs may need to be handled at the application level.
- o The Widex protocol MUST have support for authentication and secure sessions, e.g. data privacy and integrity.

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

As a means to support remote user interfaces, a number of security considerations need to be addressed, including the potential for unauthorized access to application services, monitoring of interactions by unauthorized third parties, spoofing of application services as a means to support phishing attacks, and denial of service attacks.

6.1. Privacy Considerations

Exposing the capabilities of Widex Servers and Widex Renderers using the appropriate service discovery and session setup mechanism has privacy implications as malicious users may harvest information about physical characteristics and applications hosted by Widex Entities. Implementors should carefully balance which characteristics of the Widex Entities are exposed over the network in accordance with the expected behaviour in their deployment environments as strong privacy measures may lead to degraded usability.

For example, a TV set or a smart screen playing the role of the Widex

Renderer can be very well used as a secondary display. In such a case, the Widex Renderer must be discoverable and its should expose its capabilities so that a Widex Server will be able to provide the user interface that best matches those capabilities instead of falling back to the default user interface, which generally leads to a degraded user experience.

In another example, a Widex Server hosting sensitive applications that are only visible by selected set of users must protect the privacy of the applications during discovery phase so that unauthorised users are not even able to discover the existence of the application before being prevented to initiate Widex sessions.

Privacy enforcement MUST allow implementation according to best common practices of the selected service discovery and session setup mechanism.

7. Acknowledgements

The authors would like to thank Lisa Dusseault and David Black for their comments that made this document more readable.

8. References

8.1. Normative References

[DOM2.Core]

Le Hors, A., Le Hegaret, P., Wood, L., Nicol, G., Robie, J., Champion, M., and S. Byrne, "Document Object Model (DOM) Level 2 Core Specification", W3C Recommendation, November 2000.

[DOM2.Events]

Pixley, T., "Document Object Model (DOM) Level 2 Events Specification", W3C Recommendation, November 2000.

[DOM2.Style]

Wilson, C., Le Hegaret, P., and V. Apparao, "Document Object Model (DOM) Level 2 Style Specification", W3C Recommendation, November 2000.

[DOM3.Core]

Le Hors, A., Le Hegaret, P., Wood, L., Nicol, G., Robie, J., and S. Byrne, "Document Object Model (DOM) Level 3 Core Specification", W3C Recommendation, April 2004.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [XML1.0] Yergeau, F., Bray, T., Paoli, J., Sperberg-McQueen, C., and E. Maler, "Extensible Markup Language (XML) 1.0 (Third Edition)", W3C Recommendation, February 2004.

8.2. Informative References

- [W3C.NOTE-mmi-framework-20030506]
Raman, T., Larson, J., and D. Raggett, "W3C Multimodal Interaction Framework", World Wide Web Consortium NOTE NOTE-mmi-framework-20030506, May 2003, <<http://www.w3.org/TR/2003/NOTE-mmi-framework-20030506>>.
- [W3C.WD-emma-20050916]
Johnston, M., Chou, W., McCobb, G., Raggett, D., and D. Dahl, "EMMA: Extensible MultiModal Annotation markup language", World Wide Web Consortium LastCall WD-emma-20050916, September 2005, <<http://www.w3.org/TR/2005/WD-emma-20050916>>.
- [W3C.WD-mmi-arch-20060414]
Bodell, M., Wahbe, A., Raggett, D., and J. Barnett, "Multimodal Architecture and Interfaces", World Wide Web Consortium WD WD-mmi-arch-20060414, April 2006, <<http://www.w3.org/TR/2006/WD-mmi-arch-20060414>>.
- [W3C.WD-sXBL-20050815]
Ferraiolo, J., Hickson, I., and D. Hyatt, "SVG's XML Binding Language (sXBL)", World Wide Web Consortium WD WD-sXBL-20050815, August 2005, <<http://www.w3.org/TR/2005/WD-sXBL-20050815>>.
- [WhatWG.WebApps1.0]
Hickson, I., "Web Applications 1.0", October 2005.

Authors' Addresses

Vlad Stirbu
Nokia
Visiokatu 3
Tampere 33720
Finland

Phone: +358 7180 08000
Email: vlad.stirbu@nokia.com

Dave Raggett
W3C/Volantis
35 Frome Road
Bradford on Avon, Wiltshire BA15 2EA
United Kingdom

Phone: +44 1225 866240
Email: dsr@w3.org

Full Copyright Statement

Copyright (C) The Internet Society (2007).

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 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).

