Widget Description Exchange Service (WIDEX) Framework
draft-stirbu-widex-framework-00

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 23, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document defines a framework 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 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. Architecture Paradigm ........................................... 3
3. The Widex Framework Definition ................................. 4
   3.1. Overview .................................................. 4
   3.2. Components ............................................... 5
      3.2.1. Service Discovery ans Session Setup ................. 5
      3.2.2. Widex Sync .......................................... 6
      3.2.3. Transport ............................................ 6
4. IANA Considerations ............................................. 6
5. Security Considerations ........................................ 6
6. Acknowledgements ............................................... 6
7. References .................................................... 7
   7.1. Normative References .................................... 7
   7.2. Informative References .................................. 7
Appendix A. Deployment Scenarios ................................ 8
   A.1. Simple Widex Elements ................................... 8
   A.2. Multimodal Widex Server ................................ 8
   A.3. Multiple Rendering Engines Widex Renderer ............. 8
   A.4. Hybrid ................................................... 8
Authors’ Addresses ................................................. 8
Intellectual Property and Copyright Statements ................ 9
1. Introduction

This document describes the components and interactions between them of the Widex framework 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 operations. The framework described in this document is intended to fulfil the requirements described in Internet-Draft Widex Requirements [I-D.ietf-widex-requirements].

2. Architecture Paradigm

The Model-View-Controller architectural pattern (MVC) [MVC] divides an interactive application into three components. The model contains the core functionality and data. Views display information to the user. Controllers handle user input. Views and controllers together comprise the user interface. A change-propagation mechanism ensures consistency between the user interface and the model.

Figure 1 describes an extension to the MVC architecture which enables the situations where the user interface and the model may be on different machines.

In the networked MVC architecture, the View is exported on the remote device while a Virtual View is maintained on the server. The change-propagation mechanism that ensures consistency between the user interface and the model is augmented with methods which keep the View synchronised with the Virtual View, synchronisation being done via updates. Additionally, user interactions or gestures are captured by the View copy and passed to the Controller as events.
It is quite common that the View is not needed in the Widex Server, but there are situations (e.g. desktop applications) in which the Widex Server has to maintain a real copy of the View. Therefore, we call the View in the Widex Server to be Virtual.

3. The Widex Framework Definition

3.1. Overview

In the context of Widex working group, 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].

The Widex Framework, described in Figure 2, is based on the networked MVC paradigm described in Section 2. The Widex framework is handling all network related issues involved in the networked MVC architecture, e.g. discovery and matching of Widex Elements, setting up sessions between Widex Elements, marshaling XML DOM updates or events and exchanging them over the wire.

---

Figure 2: Widex Framework Overview
3.2. Components

The Widex Framework has three components:

- Service Discovery and Session Setup
- Widex Sync
- Transport

![Diagram of Widex Framework Components]

Figure 3: Widex Framework Components

3.2.1. Service Discovery ans Session Setup

The Service Discovery and Session Setup component is treated by the Widex Framework as a black box; any service discovery mechanism being able to find matches between a Widex Server and a Widex Renderer and any session setup mechanism able to establish a session between a matching Widex Server and Widex Renderer can be used as part of the framework.

We define a match or compatibility between a Widex Server and Widex Renderer when the Widex Renderer can display the user interface exported by the Widex Server. Typically this means that both devices support the same XML DOM and XML DOM Events specifications and the Widex Renderer has rendering capabilities for the XML user interface language exported by the server.

Quite often, the Widex Server can make better decisions on what user interface to export to a particular Widex Renderer if it has additional information about its hardware capabilities, e.g. screen size, color depth, input methods.

Therefore, the Service Discovery mechanism MUST negotiate the following capabilities:

- supported XML DOM level,
- supported XML DOM Events level,
o supported XML user interface description language,
o supported transport mechanism.

The Service Discovery mechanism SHOULD negotiate the following
capabilities for the Widex Renderer:

o resolutions or display ratio,
o color depth,
o input methods.

The Session Setup mechanism MUST be able to initiate the Widex
Session between the Widex Server and the Widex Renderer using the
selected transport mechanism.

3.2.2. Widex Sync

The Widex Sync component provides the functionality that enables the
marshaling of XML DOM updates and events.

NOTE: The Widex Sync messages will be defined in a separate document.

3.2.3. Transport

The Transport component is treated by the Widex Framework as a black
box; any transport mechanism being able to facilitate the exchange of
Widex Sync messages between a Widex Server and a Widex Renderer can
be used as part of the framework.

The Transport mechanisms MUST carry the Widex Sync messages as string
of bits between the Widex Server and the Widex Renderer.

4. IANA Considerations

This document makes no request to IANA.

5. Security Considerations

The Widex Framework provides no security facilities (e.g.
authentication or privacy) of its own. It relies on the framework
components for all of these abilities.

6. Acknowledgements

TBD.
7.  References

7.1.  Normative References

[DOM2.Core]

[DOM2.Events]

[DOM2.Style]

[DOM3.Core]

[I-D.ietf-widex-requirements]
Stirbu, V. and D. Raggett, "Widget Description Exchange Service (WIDEX) Requirements", draft-ietf-widex-requirements-00 (work in progress), January 2006.


7.2.  Informative References

[MMI.Arch]

Appendix A. Deployment Scenarios

This Appendix describes the possible scenarios for deploying the Widex Framework.

A.1. Simple Widex Elements

In this scenario both devices running the Widex Server and the Widex Renderer are using the same platform.

A.2. Multimodal Widex Server

In this scenario the application running in the Widex Server is a multi-modal application compliant with W3C’s Multimodal Architecture and Interfaces [MMI.Arch], enabling him to interact with several Widex Renderers.

A.3. Multiple Rendering Engines Widex Renderer

In this scenario the Widex Renderer has support for multiple rendering engines, enabling it to interact with several types of Widex Servers.

A.4. Hybrid

In this scenario the application running in the Widex Server is a multi-modal application and at the same time the Widex Renderer has multiple rendering capabilities. It is the job of the Service Discovery and Session Setup component to determine which is the most appropriate modality of interaction.

Authors’ Addresses

Vlad Stirbu
Nokia
Visiokatu 3
Tampere 33720
Finland

Phone: +358 7180 08000
Email: vlad.stibu@nokia.com
this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.