HTTP Bindings and REX status report

Dave Raggett <dsr@w3.org>

HTTP Binding

- Implementation of REX using JavaScript and Ajax
- REX documents downloaded from server in response to HTTP POST method
- MIME type as per RFC 3023
  - text/xml, application/xml or application/rex+xml
- RFC 3023 recommends when using text/xml to use UTF-8
  - Content-type: text/xml; charset="utf-8"
  - <?xml version="1.0" encoding="utf-8"?>

HTTP Binding

- Application logic runs as a server-side process
  - e.g. using SCXML, an XML based event driven state machine language
  - One running instance per application session
- HTTP requests routed to this process on basis of a session id
  - http://example.com/scxml?session=5783
- HTTP POST requests made from JavaScript via either XMLHttpRequest or getURL (certain SVG players)
- Session id is dynamically included in the Web page when it is generated by the server
  - e.g. as an XHTML <meta /> element or as a JavaScript global variable
- Web page request may initiate new session or attach to an existing session

JavaScript-based REX implementation

- XHTML or SVG document links to external script "rex.js"
- document's onload event used to initiate request to server for events which are passed in response to HTTP POST request
- Local event handlers e.g. on window object can be used to forward events to server as part of HTTP POST request
- Script supports
  - sending and receiving events via HTTP
  - XPath subset for targeting received events
  - forwarding of selected local events to server
  - deferred dispatch of events sent by server (timeStamp)

Efficient delivery of events

- In many cases server events will occur as a reaction to events raised as a result of user input
- This makes it desirable to get the server to pass the events it raises in the HTTP response for the HTTP request that passed the UI events to the server
- The Widex Renderer also needs to poll the server for unexpected events
- Server delays response until it has an event
- Avoiding sending too many HTTP requests by packaging multiple events into the same request (REX document)
- Similar concerns for HTTP responses

Streamed processing of XML

- Processing XML on the fly without waiting until the complete document has been downloaded
  - One REX document instead of many
- Using HTTP for event streams
  - Server starts to send response before request has finished
  - Avoids packaging problem and simplifies code
  - Send each event immediately by writing it to the open HTTP connection
- But sadly, you cannot do this with Ajax (XMLHttpRequest)
  - XMLHttpRequest waits until response is complete before invoking call back
  - Solution involves a browser extension, e.g. a plugin

Avoiding the need for the script

- It would be nice if there was a way to avoid the need to use a linked script
- One approach is to extend the UI markup language with a declarative means to invoke the remote event support
- Another would be to extend HTTP in some way
  - new request header indicating Widex capability?
  - new response header with URI for event stream?
  - but what about effects of HTTP proxies?
- In some cases, an external software agent could configure the browser appropriately, e.g. in response to user running an application from a menu, or in response to SMS or SIP

HTTP Binding for Widex

- Can be done now without changes to HTTP
- Use XML in HTTP request or response
  - Possibly together with efficient encoding as is being developed by W3C EXI WG
- There could be potential benefits to extending HTTP for use with existing UI markup languages without need to change them
- Role of streamed processing of XML and the means for HTTP servers to start response before request body is complete

REX Status

W3C Remote Events for XML
http://www.w3.org/TR/rex/

- Updated Working Draft well behind expected schedule
- Initial draft only covered DOM Mutation events
- Responsibility split between Web API and SVG WGs
- France Telecom have just disclosed several patents and exempted claims under the W3C Patent Policy
- See http://www.w3.org/2004/01/pp-impl/38482/stat
- A W3C Patent Advisory Group is now expected
- This is likely to slow down work on REX ...

Ways Forward?

- Could wait for W3C to resolve the matter
- Alternatively, Widex WG could develop a specification covering all DOM events
  - at least as an interim measure, pending outcome of W3C discussions
  - Leave open long term ownership - i.e. possibility of work reverting to W3C at some future point
- Spec would define XML serializations for DOM events along with processing model, e.g. for time stamps

Open issues in REX specification

- WD's emphasis on streamed XML processing
- Terminology for role of mutation events
- Support for copy-of
- Semantics of event element's position attribute
- Definition of time stamp mechanism
- Processing model for mutation events
- Definition of XPath subset
- Support for dynamic addition/removal of event listeners
- Which set of events to specify serialization for
- Relationshop to DMSP (see Thursday's BoF)

Streamed processing of XML

- Recommended by initial Working Draft
- When using REX to animate SVG the files could potentially be rather large
- Streaming reduces memory needs and load times
- Less critical when applying REX to remote UI
  - UI events are small in size
  - Server responds by updating part or whole of UI, but still quite small
- Spec should be even handed ...

Copy-of

- Adaptation of XSLT copy-of feature
- Makes it easier to copy sub-trees and to wrap existing elements within new ones

```
<rex xmlns='http://www.w3.org/2006/rex'
 xmlns:html='http://www.w3.org/1999/xhtml'>
  <event target='html:html/html:body/html:table[3]'
   name='DOMNodeRemoved'>
    <html:div>
      <copy-of select="."/>
    </html:div>
  </event>
</rex>
```

Position attribute

- Defines where in target nodes content to insert the content of the event element
- Spec doesn't define it sufficiently precisely
- Does it count all DOM nodes or only element nodes?

```
<event target='/html/body/table[2]'
name='DOMNodeInserted' position='7'>
  <tr xmlns='http://www.w3.org/1999/xhtml'>
    <td>Rover</td>
    <td>Alpine Labrador</td>
    <td class='food'>bone</td>
  </tr>
</event>
```

Description of example suggests position only counts elements, but this isn't defined anywhere.

Time Stamps

Proposed change following discussion at 65th IETF

- Time stamps are relative to current time anchor
- Default anchor is start of session
- Use timeRef="anchor" to define this event as new anchor
- Time defined relative to a reference time base, e.g. an audio stream
- Pausing the audio will then pause application of updates
- Are all mutation events so paused or just a selected subset?
- Can the timebase be warped as envisaged in SMIL?

Processing of Mutation events

- Some people object to the terminology
  - DOM Mutation events signal changes to a DOM
  - REX proposes their use to cause changes
  - But this makes sense when reflecting changes from server's DOM to the renderer's DOM
- REX currently mandates that mutation events sent by the server be dispatched as such on target nodes
- Seems more appropriate to just apply the updates and leave it to the local DOM to raise the resultant mutation events

Using REX to add/remove event listeners

How can the server do this?

- One way is to update markup (or linked script) to bind the event listeners
- It is however desirable to provide a means to do this directly
- Proposal to add a pair of events to DOM3 to signal addition/removal of event listeners
- Web API WG has frozen the DOM3 Event spec, so this would need to be specified separately, e.g. in REX

Using REX to add/remove event listeners

Event IDL definitions as composition of Event and addEventListenerNS (removeEventListenerNS) interfaces:

```
interface AddListenerEvent : Event {
   void     initAddListenerEventNS(in DOMString namespaceURI,
                                   in DOMString typeArg,
                                   in boolean canBubbleArg,
                                   in boolean cancelableArg,
                                   in EventListener listener,
                                   in boolean useCapture,
                                   in DOMObject evtGroup);
};

interface RemoveListenerEvent : Event {
   void     initRemoveListenerEventNS(in DOMString namespaceURI,
                                      in DOMString typeArg,
                                      in boolean canBubbleArg,
                                      in boolean cancelableArg,
                                      in EventListener listener,
                                      in boolean useCapture);
};
```

Event packaging

- REX allows multiple events to be packaged as single document
- How does the Widex Renderer determine whether to hold back an event or to send what it has right now?
- Is there a need for a means to control this?
- Answer: no - this should be left to implementations to decide

Which events to include in Widex/REX specification?

- REX WD currently only DOM Mutation events
- Proposal is to cover all DOM2 and DOM3 events
- This involves defining mapping from IDL to XML
- Most IDL attributes are values such as boolean, long or string
  - screenX/screenY, ctrlKey/shiftKey, button
- UIEvent has view attribute that references a window
- MouseEvent has reference to DOM node (mouseover/out)
- DOM3 Events not yet a W3C Recommendation, but no additional problems

Relationship to DMSP?

Proposed distributed multimodal synchronization protocol, see draft-engelsma-dmsp-01.txt

- Eventing mechanism specific to coupling visual UI markup language (XHTML) with VoiceXML
- Covers large number of events
- Strong potential overlap with Widex/REX
- Defines its own binary encoding as alternative to XML
- BoF on Thursday proposing a Distributed Multimodal WG

The DMSP BoF proposes that the IETF charters a new Distributed Multimodal Working Group will develop the protocols necessary to control, coordinate, and synchronize distributed modalities in a distributed Multimodal system. There are several protocols and standards necessary to implement such a system including DSR and AMR speech compression, session control, and media streaming. However, the DM WG will focus exclusively on the synchronization of modalities being rendered across a network, in particular Graphical User Interface and Voice Servers.
