XML Events

Adding Behavior To XML Content

T. V. Raman & Rafah Hosn

IBM Research
Outline

- Motivation.
- DOM2 Interface `EventListener`.
- Authoring `EventListener` via XML.
- Bringing XML content to life via the DOM.
Motivation
Bring Static XML To Life
Universal syntax for communicating S-expressions.

- Can encode structured data.
- Can encode structured documents.
- Enable self-describing content.
Syntactic Expressions

*Come to life when bound to behaviors.*

- **Semantics** implemented by consumer.
- XML separates *data* from *behavior.*
Example

\[<\text{person}>...</\text{person}>\]

**Possible behaviors include:**

- Insert appropriate record into a database.
- Generate human-readable display.
- Generate user interface to edit \textit{person}.

**Attached behavior determines interaction semantics.**
Interface EventListener

Defines event dispatch for XML browsers.

- Provides generic eventing framework.
- Generalizes HTML eventing mechanism.
- Framework for adding behavior to XML.
DOM2 Event Propagation
DOM2 Event Propagation

Event flow when click here is activated.

- User agent propagates event:
  - Capture — Event travels from root to target,
  - Target — Event arrives at the target,
  - Bubble — event bubbles back to the root.

Observers can attach to nodes on path the event travels.
DOM2 Events Features

- A generic event system,
- Register event listeners and handlers,
- Route events through a tree structure,
- Access to context information for each event,
- Definition of event flow.
Goals

- Expose DOM event model via XML markup.
- Extend events without modifying DTD.
- Integrate with other XML languages.
Authoring Behavior Via XML Events

**Authoring** — specify a \((\text{element, event, handler})\) triple.

- `<listener>` — directly specify the triple.
- Observer — attributes specify event, handler.
- Handler — attributes specify event, observer.
Specifying Triple Explicitly

```xml
<listener event="activate"
observer="button1"
handler="#doit"/>
```

- Call handler identified by `#doit`
- When event `activate` occurs
- On element `id=button1`
- Or any of its children.
<listener propagate="stop"
    event="activate" observer="block"
    handler="popup" phase="capture"/>

Call handler id=popup

when block receives event activate.

Stop further propagation of this event.
Attaching Attributes To Observer

Events can be authored directly on the observer element.

<anyp-element ev:event="ev:click"
  ev:handler="#clicker"/>
<a href="doc.html"
  ev:defaultAction="cancel"
  ev:event="activate"
  ev:handler="#popper">
The document</a>
Attaching Attributes To The Handler

*Handler can carry event attributes.*

```xml
<script type="..." ev:event="submit" ev:observer="form1">
    return docheck(event);
</script>
```

- Declares script handler for event *submit*
- Arriving at element *id="form1"*. 
XHTML
Generic Application Container
XHTML 1.1 — Generic Application Container.

- Can host multiple namespaces.
- Presentation can be *styled*.
- Can be brought to life via events.
XML Vocabularies

*XML vocabularies define domain-specific markup.*

- Define constructs for encoding data.
- Declarative event handlers.
- Modality-specific event types.
- Use XML events to *bind* these handlers.

*Turn XHTML browser into an application container.*