B336 Advanced Internet Computing

Writing Wireless Applications with WAP (2)

Learning Objectives

- Within WML -
- Study use of event handlers, variables
- Study tasks, menus and templates

WML: Event Handlers

```
<card id="about" title="BLIX" ontimer="#about2">
  <timer value="100"/>
  <p mode="nowrap">
   <a href="#main">Done</a><br/>
   Blix Publishers aims to offer the best in quality
   reading material, be it fact or fiction.
  </p>
  <p>
   <a href="#main">Done</a><br/>
   <a href="#about2">More</a><br/>
  </p>
 </card>

<card id="about2" title="BLIX">
  <p mode="nowrap">
   <a href="#main">Home</a><br/>
   <a href="allbooks.wml">All Books</a><br/>
   <a href="http://www.blixconf.com">Conferences</a><br/>
   <a href="allauthors.wml">All Authors</a><br/>
   <a href="support.wml">Support</a><br/>
   <a href="membership.wml">Membership</a><br/>
   <a href="contact.wml">Contact Us</a><br/>

  </p>
 </card>
```

after 10 seconds, generate ontimer event using fragment identifier (can also be a URL)

gets there sooner if user chooses More

WML: Variables

- Variables are used to preserve state info across several documents
- Used inside <onevent> and <do> elements

```
<setvar name="N"  value="V"
```

- Variables can also be bound to a value in the <input> and <select> elements
- To include the value into some expression use '$'

```
The timer is $(Timer)
```

- The scope of a variable is set or reset using the newcontext attribute of <card>

WML: Variables

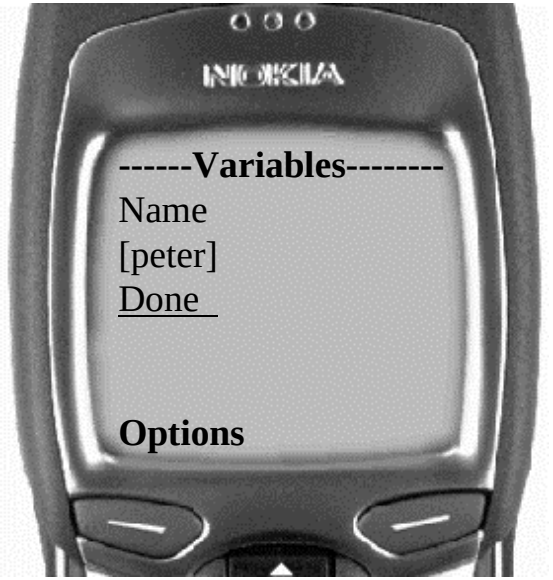
- <newcontext> clears all variables and the history list. So use early to avoid interference from previous services
- Variable names: case sensitive, letter or _ followed by zero or more letters, digits, or undescores.
- Variables can be used inside elements that may contain plain text and in the value of the following:

  - all title, value, ivalue, href and src attributes
  - label attribute of the <do> element
  - name attribute of the <postfield> elements
  - alt and localsrc attributes of the <img> element
  - event handler attributes onpick, ontimer, onenterforward and onenterbackward

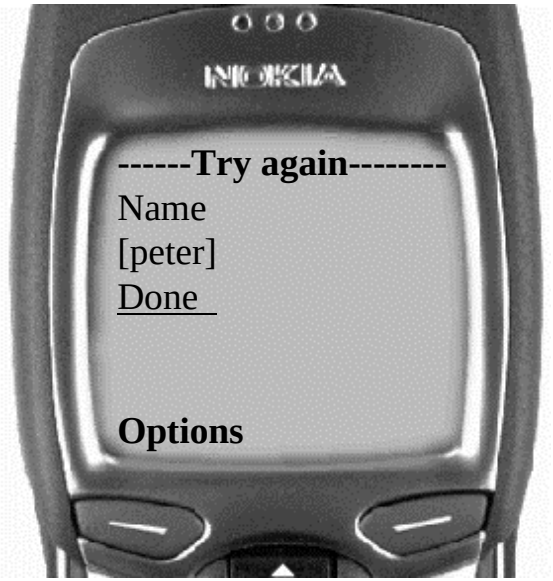
WML: Variables

```
<wml>
<card id="card1" title="$title">
  <onevent type="onenterforward">
   <refresh>
    <setvar name="title" value="Variables" />
   </refresh>
  </onevent>
  <onevent type="onenterbackward">
   <refresh>
    <setvar name="title" value="Try again" />
   </refresh>
  </onevent>
  </p>
  <p>
   Name <input type="text" name="fname" title="Enter name" />
   <anchor> Done
    <go href="#card2"/>
  </p>
 </card>
<card id="card2" title="Variables">
  <p>
   Your name is $(fname)?
   <anchor> Yes
        <go href="next.wml">
   </anchor>
   <anchor> No
        <prev />
   </anchor>
  </p>
 </card>
</wml>
```

triggers onenterbackward type event



card1 (first time)



card1 (after onenterbackwards event)

WML: Tasks and Menus

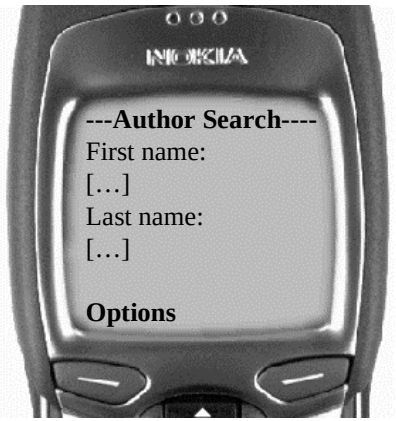
Element	Attributes
Tasks	
<anchor>	id
	class
	title
<do>	id
	class
	type
	label
	name
	optional={true|false}
<go>	id
	class
	href
	sendreferer
	method={post|get}
	accept-charset

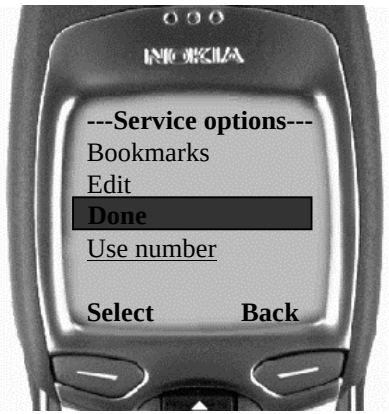
Element	Attributes
Tasks (con't)	
<prev>	id
	class
<refresh>	id
	class
<noop>	id
	class

WML: Tasks and Menus

- Use the <do> element to set up menus within a card
- Each menu option has a label (<12 chars, no spaces) and a task (<go>,<prev>,<noop> or <refresh>)

```
<card id="form" title="Author search" newcontext="true">
  <do type="prev" label="Back">
   <prev/>
  </do>
  <do type="accept" label="Done" >
       <go method="post" href="search">
           <postfield name="type" value="bio" />
           <postfield name="fname" value="$fname" />
           <postfield name="lname" value="$lname" />
       </go>
    </do>
<p>
   <fieldset>
          First name:
          <input name="fname" type="text" title="Enter name"/>
   </fieldset>
   <fieldset>
          Last name:
          <input name="lname" type="text" title="Enter name"/>
   </fieldset>
</p>
</card>
```





WML: Tasks and Menus

- type attributes in <do> element lets browser choose appropriate presentation format
- Always specify one of the following types to allow this:

Type	Description
accept	Positive acknowledgment (acceptance)
prev	Backward history navigation
help	Request for help. May be context-sensitive
reset	Clearing exisiting state
options	Context-sensitive request for options or extra operations
delete	Delete item or choice

WML: The <go> task

- <go> is used to navigate to another resource or post data to a web server

```
<do type="accept" label="Done" >
     <go method="post" href="search">
         <postfield name="fname" value="$fname" />
         <postfield name="lname" value="$lname" />
     </go>
  </do>
```

composes an HTTP-style post request, making these values available to the search resource

- <setvar> elements can also be used to make variables available in the target resource

- When <go> is activated, an onenterforward event is generated and injected into the target resource

WML: The <prev> task

- <prev> is used to navigate to the last entry on the history list
- In WML, history list is a list of old HTTP requests (not a copy of each resource as in HTML)
- Requests are resubmitted and previous document is fetched from network or local cache

```
<do type="prev" label="Back">
    . . .
    <prev/>
</do>
```

← **typical application creates a menu item that is a Back key**

- <setvar> elements can also be used to go back into history with new variable values
- When <prev> is activated, an onenterbackward event is generated and injected into the target resource

WML: The <noop> task

- <noop> does nothing and is therefore useful only as a way of overriding the effects of another <do> element with the same name:

```
<wml>
 <template>
  <do label="Back" name="prev" type="prev">
    <prev/>
  </do>
 </template>

 <card>
  <!-- This card won't display the 'Back' key -->
  <do name="prev" type="prev">
     <noop/>
  <do>
 </card>

 <card>
  <!-- This card will display the 'Back' key from template -->
 </card>
</wml>
```

- <noop> can also be used to override the effects of an <onevent> event binding in a similar fashion

WML: The <refresh> task

- The <refresh> task provides a way of updating variables with new values. Eg:

```
<do type="accept" label="Restart" >
 <refresh>
   <setvar name="Timer" value="Running" />
   <setvar name="T" value="10" />
 </refresh>
</do>
```

variables Timer and T will get new values when Restart option is chosen

- <do> can be located inside a <card> or <template>. When a card is processed, the <do> elements from the template are combined with those from the card, except when a <do> element is defeated using <noop>

WML: Accessing Tasks Inline

- <anchor> element can be used as an alternative way to include a task as a link inside the content:

```
<card id="author" title="Author search" newcontext="true">
  <p mode="nowrap">
   <fieldset>
          First name:
          <input name="fname" type="text" title="Enter name"/>
   </fieldset>
   <fieldset>
          Last name:
          <input name="lname" type="text" title="Enter name"/>
   </fieldset>
   <anchor>Done
       <go method="post" href="search">
           <postfield name="type" value="bio" />
           <postfield name="fname" value="$fname" />
           <postfield name="lname" value="$lname" />
       </go>
   </anchor>
</card>
```

<anchor> takes the place of the <do> element

- This can be helpful when trying to get different handhelds to present services in the same way

References

Anderson, et.al. *Professional XML*. Wrox Press, 2000. Chapter 14.

And see

WAP Forum at http://www.wapforum.org