

## Exercise 4 – Mensch-Maschine-Interaktion 2

### Interaction Tracking With JavaScript

URLs of further resources and source code for this exercise are available from <http://murx.medien.ifi.lmu.de/~richard/mmi2/form>.

A comprehensive introduction to JavaScript programming provided by Yahoo! JavaScript Architect Douglas Crockford can be found as a video at <http://developer.yahoo.com/yui/theater> (second column).

### 1 JavaScript (Ecmascript)

Acquaint yourself with the basic facts about the following features of JavaScript:

- Syntax and semantics: Functions, global/local variables, automatic type conversion etc.
- Embedding JavaScript code in HTML documents (using `<script>` or attribute values)
- Traversing the HTML DOM (document object model) tree
- Changing values in the document tree
- User-generated events: onfocus, onmouseover etc.
- Useful debugging techniques: Changing the status bar text, creating popup messages
- Usage of the Mozilla/Firefox DOM inspector and JavaScript console or the Firefox extension Firebug (<http://www.getfirebug.com/>) for debugging

### 2 Tracking Interactions in HTML Forms

(Per-person homework, 1 week)

To evaluate the usability of a complex web form, it can be of interest to collect information about how people fill in the forms. In particular, it is interesting

- in what order form fields are filled in
- how much time is spent on a particular input field
- whether the contents of an already filled-in field are changed later
- whether copy & paste is used

**Task:** Without changing its appearance, extend a form and add the capability to track user interaction. It should be tracked when a form element is changed, when the mouse is placed on a particular element and when text is selected in a text field. The information should be stored and transmitted along with the “normal”, user-entered data when the form is submitted.

This technique can be used to perform a usability test without additional infrastructure at the client side. However, it is important to inform the the users that they are tracked. A similar approach may be used to monitor regular usage and provide help depending on the users' actions.

Download the sample form ([http://www.medien.ifi.lmu.de/lehre/ss07/mmi2/sample\\_form.html](http://www.medien.ifi.lmu.de/lehre/ss07/mmi2/sample_form.html)) and add interaction tracking code to all fields.

The **action** attribute of the `<form>` element has the following URL:

<http://murx.medien.ifi.lmu.de/~richard/mmi2/http-get-post>

This script will output a list of all values submitted by the form.

### Submission:

- Solutions must include the modified HTML code for the form and a short description of how the code works
- Your JavaScript code must work with Firefox (and should work with Internet Explorer).
- All important aspects of the JavaScript/HTML must be commented.
- Submission is by email to [arnd.vitzthum@ifi.lmu.de](mailto:arnd.vitzthum@ifi.lmu.de). Please use an HTML attachment named *exercise4nachname\_vorname.html* and send the description in the main body of the mail.
- Deadlines for email submission: Tuesday, June 12, 2007, 9 a.m. (members of groups 1-5), Thursday, June 14, 9 a.m. (members of groups 6 and 7) and Friday, June 15, 9 a.m. (members of groups 8-10).

**Example:**(Complete document available as <http://murx.medien.ifi.lmu.de/~richard/mmi2/form>)

```
...
<script type="text/javascript">
<!-- // Comment prevents very old browsers from displaying script
// Return string of the form "2004-12-31,23:59:59"
function datestamp() {
    var now = new Date(); // "var" => is local variable
    return now.getFullYear() + "-" + (now.getMonth()+1) + "-"
        + now.getDate() + "," + now.getHours() + ":" + now.getMinutes()
        + ":" + now.getSeconds();
}
logVal = ""; // Initialised when page loads. Contains entire log of actions

// Called from the form elements below to log things with a timestamp
// @param domObject Reference to a node in the DOM tree
// @param text String to write to log
function writeLog(domObject, text) {
    var objectId = domObject.getAttribute("id");
    var logLine = datestamp() + " " + objectId + " " + text;
    window.status = logLine; // Display logLine in status bar
    logVal = logVal + logLine + "\n"; // Add logLine to interaction log
}
// -->
</script>
...
<form name="mmiform" method="post"
    action="http://murx.medien.ifi.lmu.de/~richard/mmi2/http-get-post"
    onsubmit="document.mmiform.log.value = logVal">
<!-- Example form element. For the JavaScript code in the "onfocus" etc attribute
values, the value of "this" corresponds to the DOM node for <input> -->
<input id="il" name="il" type="text" value="" onfocus="writeLog(this, 'focus')"
    onchange="writeLog(this, 'change')" onselect="writeLog(this, 'select')"
    onmouseover="writeLog(this, 'mouseover')">
<!-- Hidden form element which we use to transmit our log back to the server -->
<input type="hidden" name="log" value="">
<input type="submit" value="Send">
</form>
```