Leibniz Universität Hannover

# Interactive Systems (ISY)

## Auditorium Exercise 08

Human-Computer Interaction Group

Jan Feuchter
jan.feuchter@hci.uni-hannover.de

# Lectures

| Session | Date | Topic | Details |
|---|---|---|---|
| 1 | 2.4. | Introduction | human performance, empirical research, modeling |
| 2 | 9.4. | Interaction elements | input devices, interaction elements, states, layouts |
| 3 | 16.4. | Event handling | events, bindings, reactive programming, scene graph |
| 4 | 23.4. | Scene graphs | event delivery, coordinate systems, nodes, animation, concurrency |
| 5 | 30.4. | Interaction techniques | alignment and pointing techniques |
| 6 | 7.5. | Interaction techniques | |
| 7 | 14.5. | Web-based user interfaces | document object model, client-server issues |
| | 21.5. | Pfingstwoche | |
| 8 | 28.5. | Web-based user interfaces | reactive Programming for the Web |
| 9 | 4.6. | Experiments and data analysis | designing experiments, hypothesis testing |
| 10 | 11.6. | Modeling interaction | descriptive and predictive models, keystroke-level model, regression |
| 11 | 18.6. | Visualization | visual encodings, perceptual accuracy, treemaps, dynamic queries |
| 12 | 25.6. | Human-Centered AI | introduction to human-centered AI, human control and automation, examples |
| 13 | 2.7. | Deep learning in HCI | guidelines for human-AI interaction, neural networks |
| 14 | 9.7. | Deep learning in HCI | convolutional and recurrent NNs, face recognition, gesture recognition |

Lecture Recap

# DOCUMENT OBJECT MODEL (DOM)

Leibniz Universität Hannover

# Document Object Model (DOM)

- Browsers represent parsed HTML / CSS as a tree
- W3C specified an API for this tree
  - Accessing and modifying nodes
  - Events
  - XML namespaces
- Programmatic access via JavaScript
  - Browser differences

https://dom.spec.whatwg.org

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

```
interface Node {
    short nodeType (ELEMENT, ATTRIBUTE, ...);
    String nodeName;
    Node parentNode, firstChild;
    Node previousSibling, nextSibling;
    Node insertBefore(Node newChild, Node refChild);
... };
interface Document extends Node {
    Text createTextNode(String data);
    Element getElementById(String elementId);
... };
interface Element extends Node {
    String tagName;
    String getAttribute(String name);
    void setAttribute(String name, String value);
    NodeList getElementsByTagName(String name);
... };
```

# Event Flow

- User input or target.dispatchEvent(event) causes event to propagate through the DOM tree to target
  - event.stopPropagation()
- Propagation path is list of "current event targets", last item is target
- Event phases
  - Capture phase: from window to target's parent
  - Target phase: event reaches event target
  - Bubble phase: from target's parent to window

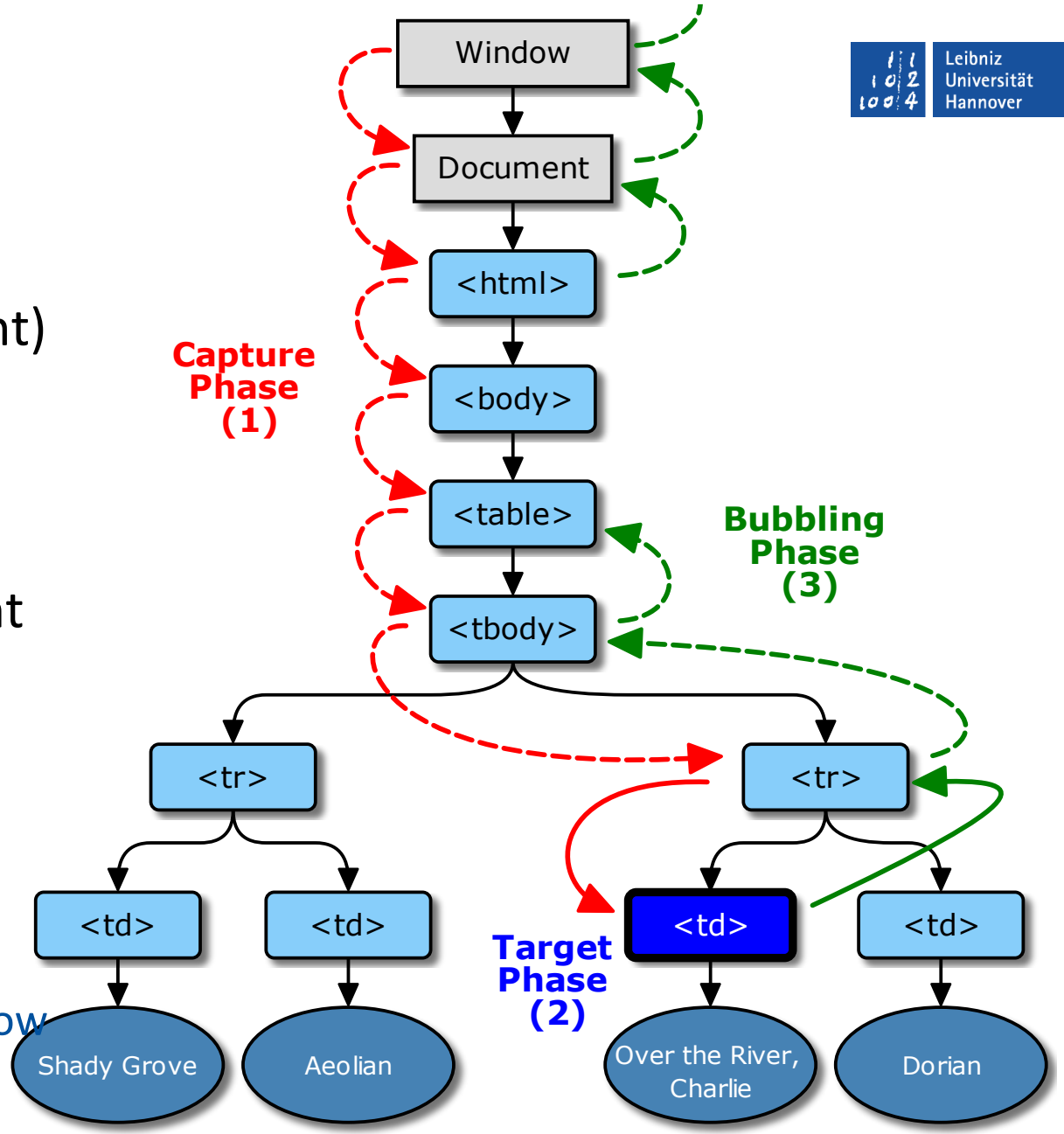
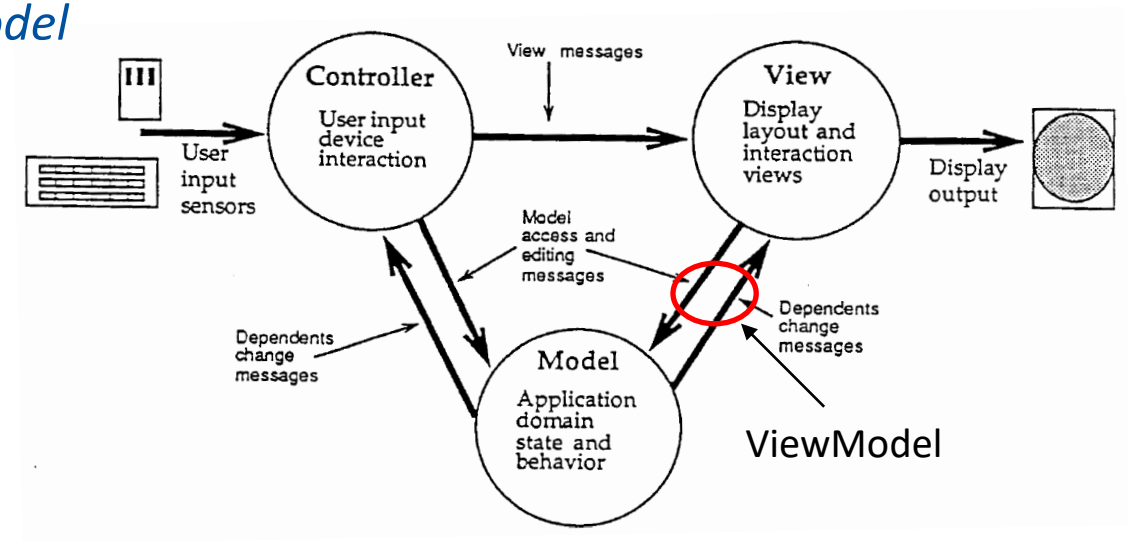



Image source: https://www.w3.org/TR/uievents/

# REACTIVE PROGRAMMING FOR THE WEB (IN THE WILD)

# Frontend Library: Vue.js

- Vue introduces a layer between *View* and *Model*
  - *ViewModel*
  - Related term: MVVM (Model View ViewModel) Architecture
- *ViewModel* layer mediates bidirectional synchronization
  - *Model* only interacts with the *ViewModel* layer
  - Properties in the *View* are bound to the *ViewModel*
- Uses only JavaScript's facilities behind the scenes
  - No compilation/transpilation step required



Leibniz Universität Hannover

# Vue.js: Calculator Example

```
const viewModel = new Vue({
   el: "#calculator",
   data: { b: 2, c: 1 },
   computed: {
      a() { return b + c; }
   }
});

<div id="calculator">
   <input v-model="b" type="number" />
   <input v-model="c" type="number" />
   <p>{{ a }}</p>
</div>
```

# Vue.js

```
Vue.createApp({
  data () {
    return {
      b: 2,
      c: 1,
    }
  },
  computed: {
    a() {
      return this.b + this.c
    },
  },
}).mount('#calculator')
```

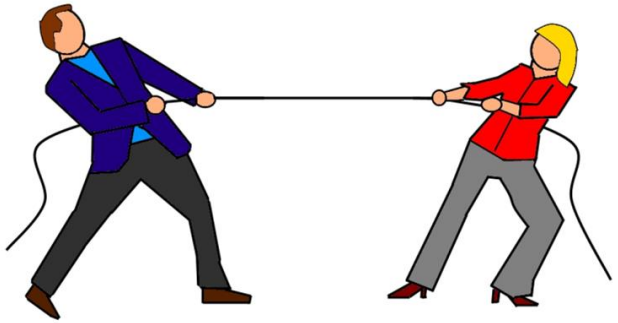
```
<div id="calculator">
  <input v-model="b" type="number"> +
  <input v-model="c" type="number"> =
  <p>
    {{ a }}
  </p>
<div>
```

Wiederholung

# EXPERIMENTE

# Tradeoff between Internal and External Validity

Internal validity



External validity

- There is tension between internal and external validity
- The more the test environment and experimental procedures are “relaxed” (to mimic real-world situations), the more the experiment is susceptible to uncontrolled sources of variation, such as pondering, distractions, or secondary tasks

MacKenzie: Human-Computer Interaction - An Empirical Research Perspective.

# Controlled Experiments

- A procedure for getting empirical data that helps in supporting or rejecting an experimental hypothesis
  - Quantitative, empirical method
  - Can never prove a hypothesis, can only provide statistical support
- Steps
  - Formulate hypothesis/research question
  - Design experiment, pick variable and fixed parameters
  - Choose participants
  - Run experiment
  - Analyze the results
  - Interpret results to accept or reject hypothesis

# Independent and Dependent Variables

- Independent variables (IVs)
  - Experimentally controlled
  - Also called: factors
  - Example factor: font size
  - Values that IVs can have: levels
  - Example levels of factor font size: 10pt, 14pt, 18pt
- Dependent variables (DVs)
  - Measured
  - Typically: execution time, error rates, subjective preferences
- Confounding variables
  - Not controlled, but might have an influence on DVs
  - Also called: hidden variable

# Experiment Design

- (Research) Question
  - Can chocolate help students
- Hypothesis
  - Chocolate helps students
- Design an experiment to test this hypothesis
  - Give chocolate to students and see what happens

Example: Markus Krause

# Experiment Design

- (Research) Question
  - Can chocolate help students
  - Can chocolate help students concentrate in class
- Hypothesis
  - Chocolate helps students
  - Chocolate helps students to concentrate in class
- Design an experiment to test this hypothesis
  - Give chocolate to students and see what happens
  - Give chocolate to students and measure concentration

Example: Markus Krause

# Experiment Design

- (Research) Question
  - Can chocolate help students
  - Can chocolate help students concentrate in class
  - Can chocolate help students better remember a lecture
- Hypothesis
  - Chocolate helps students
  - Chocolate helps students to concentrate in class
  - Chocolate helps students to better remember a lecture
- Design an experiment to test this hypothesis
  - Give chocolate to students and see what happens
  - Give chocolate to students and measure concentration
  - Give chocolate to students and ask them specific questions after the lecture to see how much they can remember

Example: Markus Krause

# Controlling Enviroment Variables

- Control variables
  - What is experimentally modified
  - Kept constant
  - More control → less generalizable
  - Example: Number of targets

**Improves internal validity**
**Compromises external validity**

- Random variables
  - What is allowed to vary randomly
  - More variability → more generalizable
  - Example: Time of day during experiment

**Compromises internal validity**
**Improves external validity**

- Confounding variables
  - Environmental factors that might influence the experiment

Example: Markus Krause

# Participants

- Should be similar to target users
  - Age, education, computer and domain expertise, etc.
- Use the same number of participants as used in similar research[1]
  - Typically at least 10 participants (rule of thumb)
  - More if need to capture small effects
- Get assigned tasks they (repeatedly) execute
- Get assigned conditions
- Follow a predefined procedure

[1] Martin. Doing psychology experiments (6th ed.). Pacific Grove, CA. Belmont, CA: Wadsworth, 2004.

# Between-Group vs. Within-Group

- Between-group
  - Each subject only does one condition of the experiment
  - There are at least 2 conditions
    (manipulated form & control, to isolate effect of manipulation)
  - \+ No learning effect across conditions
  - \- But requires more users
- Within-group
  - Each subject does all the conditions of the experiment
  - \+ Less users required, individual differences canceled out
  - \- But often learning effect across conditions

| Participant | Test Condition |
|---|---|
| 1 | A |
| 2 | A |
| 3 | B |
| 4 | B |
| 5 | C |
| 6 | C |

| Participant | Test Condition | | |
|---|---|---|---|
| 1 | A | B | C |
| 2 | A | B | C |

# Latin Squares

**2 x 2**

| | |
|---|---|
| A | B |
| B | A |

**3 x 3**

| | | |
|---|---|---|
| A | B | C |
| B | C | A |
| C | A | B |

**4 x 4**

| | | | |
|---|---|---|---|
| A | B | C | D |
| B | C | D | A |
| C | D | A | B |
| D | A | B | C |

**5 x 5**

| | | | | |
|---|---|---|---|---|
| A | B | C | D | E |
| B | C | D | E | A |
| C | D | E | A | B |
| D | E | A | B | C |
| E | A | B | C | D |

MacKenzie: Human-Computer Interaction - An Empirical Research Perspective.

# Order Effects, Counterbalancing

- Only relevant for within-subjects factors
- Order effects
  - Learning effects, practice effects, fatigue effects, sequence effects
- Order effects offset by counterbalancing
  - Participants divided into groups
  - Test conditions are administered in a different order to each group
  - Order of administering test conditions uses all possible permutations (n! for n conditions) or a subset of the possible permutations (Latin square)
- Latin square
  - Each condition occurs precisely once in each row and column

MacKenzie: Human-Computer Interaction - An Empirical Research Perspective.

# ASSIGNMENT 8

# Project & Experiment

- Assignment 7: Plan an Experiment ✔
  - Think of a new interaction technique
  - And how to evaluate your research question
- Assignment 8: Implement a Prototype
  - Implement your idea using either JavaFX or Web Technologies (or Android ☺)
  - Follow-up errors won’t loose you points
- Assignment 9: Conduct an Experiment
  - Other members of the course act as participants
  - And you take part in a few experiments as well
- Assignment 10: Analyse Results
  - Answer your research question
  - Give a short presentation of your project and results (~5 minutes)

# Your Prototype

- Should serve only to test your hypothesis
  - Keep it as simple as possible
- Should generate randomized participant IDs
- Should save measurements and general info into as .CSV file
- Should be somewhat portable
  - In the next assignment, you can participate in each other's experiments
  - Distributing the exported project is sufficient,
    you don't need to generate a .jar file.
- Should use either JavaFX or one of the web technologies discussed in the lecture
  - Exceptions possible in individual cases depending on the experiment

- Any follow up errors between the four assignments won't cost points