## Human-Computer Interaction

### Exploring the terrain and seeking quick inspiration
- **Typeform**
  - Visually engaging, easily shareable surveys
- **Betterific**
  - Crowd sourced inspiration
- **Voice**
  - Quick input and inspiration from voice polls

### Recruiting research participants
- **Clarity**
  - Find practitioner-experts in a given business area
- **PivotPlanet**
  - Find practitioner-experts in a given business area
- **Ethnio**
  - Intercept website or app users in the moment

### Obtaining deep insights into people’s lives and everyday contexts
- **24tru***
  - Remote, video-based diary studies and web-based discussions
- **Revelation***
  - Remote diary studies, live video and web-based discussions
- **Crowdtap***
  - Crowd sourced inspiration

### Eliciting feedback on concepts, insights, or value propositions
- **Usabilla**
  - Visual feedback on web prototypes or live websites
- **Optimizely**
  - A/B and multivariate testing platform
- **Unbounce**
  - Concepts or value prop. testing on landing pages

### Analyzing and synthesizing research findings
- **Murally**
  - Visual, collaborative organization of material
- **NarrativeClip**
  - Life-logging and remote diary studies

---

*We found these platforms to be useful for more than one research activity. For example, Crowdtap is useful for all five research phases.*
“If you are not embarrassed by the first version of your product, you’ve launched too late.”

Reid Hoffman
How we can evaluate the usability (quality) of interactive systems?
UI quality

Presentation of information

multiplicity of devices & representations
UI quality

Presentation of information

input/output reusability

use output produced by one action as input for another
UI quality

Ordering of task planning

multiplicity of user roles

regular user versus administrator
UI quality

Ordering of task planning

multiplicity of execution paths

menu versus toolbox versus shortcuts
UI quality

Ordering of task planning

non-preemptiveness

degree of freedom for users to decide what’s next
UI quality

Ordering of task planning

reachability

possibility to navigate in the system
(e.g., undo, redo, breadcrumb pattern)
UI quality

Ordering of task planning

observability *versus* browsability
UI quality

Adaption of interaction

reconfigurability

system ability to support user personalization
UI quality

Adaption of interaction

reconfigurability

system ability to support user personalization

configuration versus personalization

affects system’s function & performance

relevant to a certain user
UI quality

Adaption of interaction

adaptivity

system ability to support automated adaptation
UI quality

Adaption of interaction

migrability

system ability to transfer responsibility from one user to another, among users, among users and systems/platforms
UI quality

Adaption of interaction

plasticity

system ability to adapt to the context of use while preserving predefined usability properties
UI quality

Standards

ISO/IEC 9126-3 (2003) – internal measures
UI quality

Quality in use is the user’s view of the quality of the software product when it is used in a specific environment and a specific context of use.

UI quality

External quality is the totality of characteristics of the software product from an external view

UI quality

External quality is the totality of characteristics of the software product from an external view measured and evaluated in the testing phase within a simulated environment, by using external metrics
UI quality

*Internal quality* is the totality of attributes of a product that determine its ability to satisfy stated and implied needs when used under specified conditions.

UI quality

**Internal quality** is the totality of attributes of a product that determine its ability to satisfy stated and implied needs when used under specified conditions.

remains unchanged until the system redesign
UI quality

For details, consult:

Usability evaluation

Criteria
Scapin & Bastien, 1997; Vanderdonckt, 1995

compatibility
consistency
work load
adaptation
dialog control
guidance
error management
Usability evaluation

Important aspects

*a priori* and/or *a posteriori*

design *versus* testing (evaluation)
Usability evaluation

Example:

the lack of explanatory messages in the case of Web links

Click here to go to the UAIC main page.
Click here to visit our HCI Website.
Click here for details about this event.
Usability evaluation

Example:

the lack of explanatory messages in the case of Web links

Click [here](#) to go to the UAIC main page.
Click [here](#) to visit our HCI Website.
Click [here](#) for details about this event.

“click here” anti-pattern
Usability evaluation

Example:

adoption of different placement strategies for content and navigational methods
“Optimal” placement of links for the main page of a Website (Shaikh & Lenz, 2006)
“Optimal” placement of links for the subsidiary pages (Shaikh & Lenz, 2006)
“Optimal” placement of the internal search tools (Shaikh & Lenz, 2006)
“Optimal” placement of the advertisements (Shaikh & Lenz, 2006)
Usability evaluation

One of the most used general technique: eye tracking

30—60 minute interviews, in which users are asked to complete real-life tasks online, while the eye tracking monitor captures their eye movements


www.nngroup.com/reports/how-to-conduct-eyetracking-studies/
What areas of the page draw users’ attention?
Do users notice and use key navigation elements?
Do users notice key marketing elements and do they recall them?

www.evocinsights.com
Are users successful in completing a particular task?
Which pages, ads, videos, or images do users prefer?

www.evocinsights.com
What do users expect to find but overlook on the site?
Usability evaluation

One of the most used general technique: eye tracking

several open-source software solutions:
  EyeWriter – eyewriter.org/developer/
  ExpertEyes – code.google.com/p/experteyes/
  Gaze Tracking – sourceforge.net/projects/gazetrackinglib/
  OGAMA (OpenGazeAndMouseAnalyzer) – www.ogama.net
  PyGaze – www.pygaze.org
Usability evaluation

PUI – Plastic User Interfaces

adaptation to the context of use while satisfying predefined usability properties of interest
D. Thevenin, J. Coutaz & G. Calvary, 2004
Usability evaluation

PUI – Plastic User Interfaces

levels of adaptation:
  lexical
  sintactic
  semantic
performed tasks
concerning the user goals
Accessibility

successful access to information and IT applications by people having special needs

visual, hearing, motor, cognitive, seizure disorders
Accessibility

realities:
20% of US population have a certain type of disability
10% of persons are having severe problems

4% of world-wide humans have major sight problems
discussion

general usability
affects all users

variables
inconveniences for some barriers for others

physical barriers
affects only disabled people
discussion

Touch

one arm

arm injury

new parent
See

don't see
blind
cataracts
distracted driver
Hear

deaf

ear infection

bartender
Speak

- non-verbal
- laryngitis
- heavy accent
cognitive issues
Accessibility

offering alternative means in order to facilitate the completion of users’ tasks for people having temporary (on short/long term) or permanent problems
Accessibility

general strategies:

textual descriptions of multimedia content (images, audio, animations, video, 3D)
Accessibility

general strategies:

a proper (logical) organization of the content and navigational paths
Accessibility

general strategies:
support for keyboard-only interaction
Accessibility

general strategies:

using standardized formats
Accessibility

guidelines and tools – concerning Web applications:

www.w3.org/WAI/

www.webaim.org
WAVE (Web Accessibility Evaluation Tool)

Uh oh! WAVE has detected 19 accessibility errors.

The following are present in the head section or apply to this page in general:

- ERROR: Form label missing
  A form `<input>`, `<select>`, or `<textarea>` does not have a corresponding label.
  (Note: Labels are not required for hidden, image, submit, reset, or button form elements.)
Accessibility

see also Ire Aderinokun,

[https://bitsofco.de/the-accessibility-cheatsheet/](https://bitsofco.de/the-accessibility-cheatsheet/)
Internationalization (I18N) and localization (L10N)

**language**
linguistic preferences of the users

**locale**
cultural preferences concerning number and date formatting, currency, etc.
Internationalization (I18N) and localization (L10N)

“If the user can’t read the description of the preference, he/she doesn’t even have a chance to make a choice.”

Achim Ruopp, 2007
Internationalization (I18N) and localization (L10N)

it is desirable to use complete phrases in order to give a better translation
Internationalization (I\textsubscript{18}N) and localization (L\textsubscript{10}N)

Web guidelines & resources:

www.w3.org/International/

www.globalbydesign.com/
discussion

avoid absolute width (texts in other languages could have different lengths)

avoid pictures (use translatable text)
Usability evaluation (Rogers, 2007)

analytical evaluation
controlled experiment
field study
formative evaluation
heuristic evaluation
predictive evaluation
summative evaluation
usability laboratory
usability studies
user testing
Multiple existing heuristics introduced by Jakob Nielsen (1994)

www.nngroup.com/articles/ten-usability-heuristics/
### Multiple existing heuristics proposed by Bruce Tognazzini (revised in 2014)

<table>
<thead>
<tr>
<th>Category</th>
<th>Heuristic</th>
<th>Category</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aesthetics</td>
<td>Discoverability</td>
<td>Metaphors</td>
<td>Protect Users’ Work</td>
</tr>
<tr>
<td>Anticipation</td>
<td>Efficiency of the User</td>
<td>Readability</td>
<td>Simple Work</td>
</tr>
<tr>
<td>Autonomy</td>
<td>Explorable Interfaces</td>
<td>Simplicity</td>
<td>State: Track it</td>
</tr>
<tr>
<td>Color</td>
<td>Fitts’ Law</td>
<td>Visible Interfaces</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td>Human-Interface Objects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defaults</td>
<td>Latency Reduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Learnability</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[asktog.com/atc/principles-of-interaction-design/](asktog.com/atc/principles-of-interaction-design/)
Multiple existing heuristics
context: mobile computing (E. Bertini et al., 2006)

1. Visibility of system status & device findability
2. Match between system and the real world
3. Consistency and mapping
4. Good ergonomics & minimalist design
5. Ease of input, screen readability and glancability
6. Flexibility, efficiency of use and personalization
7. Aesthetic, privacy and social conventions
8. Realistic error management
Usability evaluation (Rogers, 2007)

The evaluator could not substitute the real users

an UI has usability problems
only if its final common users have these problems

code inspection vs. code testing
User testing

UI quality evaluation could be performed by users
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the focus is on usability
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the participants are (potential) end users
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

there is a real product/service to be evaluated
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the participants perform tasks, usually while thinking aloud
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the data are recorded and further analyzed
Test Plan for: Product/Application Component

Persona Type Tested: Primary/Secondary/Negative/Supplemental/Served/Customer

Context: an environment (location) where the user is comfortable without distractions

Duration: XX minutes

Complete the following before starting: Greeting and Consent & User Background Information.

Overall Objectives

List the objectives/goals of the usability test; what you want to study or to evaluate. Be specific in what you want achieve, to evaluate with this test (e.g. obtain quality or quantitative regarding certain product/application components, accessibility evaluation, heuristic evaluation, cognitive walk-through etc.).

- Assess the overall effectiveness of the product/application for different types of users performing basic, common tasks;
- Identify obstacles to completing a certain task or using a certain component;
- Evaluate consistency and standards;
- Use the References to guide you in writing your objectives.

Scenario

Use one scenario for each test. Do not use multiple scenarios for a usability test as it can confuse and frustrate the user. The scenarios used in the usability tests have a direct connection with the scenarios built for the personas document.

Tasks

Write out the tasks for each persona type. Avoid complex tasks. Make sure the user is able to complete the test in the specified time interval. Assign an estimated completion time for each task.

- Task 1 + estimated time to complete it;
- Task 2 + estimated time to complete it;
- Task 3 + estimated time to complete it;
- As many tasks needed.

A user might complete a task in a shorter or greater time than the estimated time. If the completion time exceeds the estimated time then it there is a problem. Also a user might skip the

Notes for Test Coordinator

- Remind the user that you would like them to ‘think aloud’ so that you can understand what they are thinking, about the task or the product/application;
- Read the user the scenario for the test;
- Read the user the tasks he has to complete. If the tasks are complex, you may want to put a sheet of paper in front of the user with each task written out;
- Record the user’s actions, comments, questions, and body language (write them in the Test Observations section accompanied by

Test Results for: Product/Application Component

Success Criteria

A successful design has been achieved when:

- 50% of users...
- 80% of users...
- And so on...

Summary

List major findings in list or table format for quick and easy reading. E.g. "A number of users (specify the number) did/find___________."

- Finding 1;
- Finding 2;
- Finding 3;
- These finding reflect the observation notes of the observer.

Demographics

User Number | Gender | Age Range | Persona Type | Technical Level | Background questions answers
---|---|---|---|---|---
User 1 | | | | | |
User 2 | | | | | |
User 3 | | | | | |
User 4 | | | | | |

Literature recommends testing no more than five users and running as many small tests as you can afford, according to Nielsen, J. (1994). Usability Engineering, Academic Press Inc, p 165.

Interaction Notes

Enter a detailed descriptions of the participants' interactions in this table. The table can be separated by task, category, or whatever makes the most sense for this test.

<table>
<thead>
<tr>
<th>User</th>
<th>Task 1</th>
<th>Task 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>Task Result description + completion time</td>
<td>Task Result description + completion time</td>
</tr>
<tr>
<td>User 2</td>
<td>Task Result description + completion time</td>
<td>Task Result description + completion time</td>
</tr>
<tr>
<td>User 3</td>
<td>Task Result description + completion time</td>
<td>Task Result description + completion time</td>
</tr>
<tr>
<td>User 4</td>
<td>Task Result description + completion time</td>
<td>Task Result description + completion time</td>
</tr>
</tbody>
</table>

profs.info.uaic.ro/~stefan.negru/usability/
User testing

Considered metrics:
effectiveness, efficiency & satisfaction
User testing

Considered metrics

effectiveness, efficiency & satisfaction

+ fun, challenge & stimulation
User testing: planning (Adaptive Path, 2001)

- 2 weeks  Determine test audience, start recruiting immediately
- 2 weeks  Determine feature set to be tested
- 1 week   Write first version of guide, discuss with team, check on recruiting
- 3 days   Write second version of guide, recruiting should be completed
- 2 days   Complete guide, schedule practice test, set up and check equipment
- 1 day    Do practice test in the morning, adjust guide/tasks as appropriate
-           Test (usually 1-2 days, depending on scheduling)
+1 day    Discuss with observers, collect copies of all notes
+3 days   Watch all video recordings, take notes
+1 week   Combine notes, write analysis
+1 week   Present to team, discuss and note directions for further research
User testing

Methods:

formative evaluation

field study

controlled experiment
User testing

Formative evaluation

finding problems for next iteration of the design project
User testing

Formative evaluation

prototype/implementation is evaluated within a controlled environment (lab), with focus on specific tasks
User testing

Formative evaluation

users, facilitators, observers offers qualitative data (usability issues)
User testing

Field study

tries to find problems with respect to a given context
User testing

Field study

evaluates the (preliminary) UI in a concrete context, with focus on real tasks
User testing

Field study

evaluates the (preliminary) UI in a concrete context, with focus on real tasks

offers qualitative annotations
User testing

Controlled experiment

to test a hypothesis

e.g., interface X is easier to be used than interface Y
User testing

Controlled experiment

could be used to evaluate a preliminary UI, in lab rigorous conditions, with focus on specific tasks

has one or more conditions (independent variables) and measures (dependent variables)
User testing

Controlled experiment

gives quantitative information:
time of reaction, error rate, user satisfaction,...


exp-platform.com/Documents/GuideControlledExperiments.pdf
User testing

Advices + aspects of interest (Lukas Mathis, 2011)

- do not influence the tester
- avoid stressful situations
- consider the ethics of conducting tests
User testing

A/B testing

comparing 2 versions of an UI element or an entire Web page in a time interval to see which performs better
User testing

A/B testing

comparing 2 versions of an UI element or an entire Web page in a time interval to see which performs better

performance criterion = conversion rate (from visitors to goal achievers)
User testing

[www.alistapart.com/articles/a-primer-on-a-b-testing/](http://www.alistapart.com/articles/a-primer-on-a-b-testing/)

consult also [elem.com/~btilly/effective-ab-testing/](http://elem.com/~btilly/effective-ab-testing/)
User testing

A/B testing

multivariate testing – different versions of individual parts of the design are tested at the same time
A multivariate test (Julian Gaviria, 2017)

**Variant A**
- Single Offer
- with Intro Message

**Variant B**
- Top 3
- Converting eBooks

**Variant C**
- Single Offer w/ Intro
- Msg + Top 2 eBooks

Website Converted at **8.59%** with 108 leads generated out of 1,258 visitors (+22.3% Improvement).

Website Converted at **8.14%** with 109 leads generated out of 1,339 visitors (+16.0% Improvement).

Website Converted at **10.12%** with 127 leads generated out of 1,255 visitors (+44.2% Improvement).

[julian.is/article/exit-intent-popups/]
User testing

A/B testing

several tools:
  Convert
  Optimizely
  Unbounce
  Vanity – [vanity.labnotes.org](http://vanity.labnotes.org)
  Visual Website Optimizer
<table>
<thead>
<tr>
<th>Company</th>
<th>Increase</th>
<th>Experiment Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majestic Wine</td>
<td>201%</td>
<td>Redesigning category webpage increases leads generated. Majestic Wines revamped their category page design to increase online enquiries for their Wedding services by 201%.</td>
</tr>
<tr>
<td>Server Density</td>
<td>144%</td>
<td>A/B testing between different pricing structures increases revenue by 114%. Server Density A/B tested between per unit and packaged pricing plans. The winning plan reduced free signups but increased the Average Order Value (AOV), and consequently revenue by 114%.</td>
</tr>
<tr>
<td>Taylor Gifts</td>
<td>111%</td>
<td>Redesign of ecommerce product page increases conversions. Conversion Optimization Agency Trinity Insight used Visual Website Optimizer to test a better version of the ecommerce product page. This led to a 111% increase in conversions.</td>
</tr>
</tbody>
</table>

real case studies: vwo.com/resources/casestudy/
User testing

Pilot study

a trial run of an experimental procedure, not expected to produce valid research data
User testing

Remote testing

by using a screen-sharing software

User testing

Remote testing

the tester’s environment cannot be typically controlled

also, the tester might get distracted during the test
Predictive evaluation

Having a correct model of the interaction between users and computers, we can predict the usability of a system, without the need of designing and/or testing a concrete UI.
Predictive evaluation

User testing could only relieve certain problems, but can not explaining them
Predictive evaluation

Classical models:

probabilistic – e.g., Bayes networks

GOMS (Goals, Operators, Methods, Selection rules)

CMP-GOMS (Cognitive-Motor-Perceptual)
task graph used by CMP-GOMS model

the critical path is the path of tasks having the longest time of execution
Predictive evaluation

Directions of research

objectives beyond productivity

- **hedonomics** (Hancock *et al.*, 2005)
Predictive evaluation

Directions of research

RITE – Rapid Iterative Test and Evaluation

used by Microsoft (M. Medlock et al., 2002, 2005)

focused on fixing usability problems rather than finding them
Usability evaluation

For further information, study:

www.interaction-design.org/encyclopedia/usability_evaluation.html

www.uxpin.com/studio/ebooks/guide-to-usability-testing/
How about designing interactivity by using model-driven approaches?
Reality

Major differences between software engineering and the effective UI development
Reality

Target-applications & domains of interest

personal productivity, business, entertainment, etc.
Reality

Notations and engineering tools

(formal) methods for software development
Reality

Multiple interaction paradigms

textual, graphical, multimedia, natural, 3D,...
Evolution of the interactive apps: context of use = (U, P, E)

Jean Vanderdonckt, 2006
# Reality

Multiple models to be considered *(Abrams *et al.*, 2001)*

<table>
<thead>
<tr>
<th>Application 1</th>
<th>Platform #1</th>
<th>Platform #2</th>
<th>Platform #3</th>
<th>Platform #4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UI #1</td>
<td>UI #2</td>
<td>UI #3</td>
<td>UI #4</td>
</tr>
<tr>
<td>Application 2</td>
<td>UI #5</td>
<td>UI #6</td>
<td>UI #7</td>
<td>UI #8</td>
</tr>
<tr>
<td>Application 3</td>
<td>UI #9</td>
<td>UI #10</td>
<td>UI #11</td>
<td>UI #12</td>
</tr>
</tbody>
</table>
Multiple models to be considered (Abrams et al., 2001)
Model-based UI design

MDA (Model-Driven Architecture)

www.omg.org/mda/
Model-based UI design

Model(s) facilitate(s) an abstract view of the interaction separation of concerns, ability of correlation parsability, editability if possible, human readability
Model-based UI design

Models

explicitly capture knowledge about UI and interactive applications with appropriate abstractions
Model-based UI design

Method

structures the definition and use of underlying models in a stage-wise approach

example: agile usability (Scott Ambler, 2008)

www.agilemodeling.com/essays/agileUsability.htm
Model-based UI design

Supporting tools

support the use of the method by providing tools for models and their related operations

ideally, one model should be supported by at least one tool
Models used in the processes of UI design (P. Forbrig et al., 2004)
Using notations to model tasks to be performed by users (L. Marucci et al., 2004)
Process Reference Framework for the development of plastic user interfaces
S = Source context of use

T = Target context of use

Jean Vanderdonckt, 2006
Bouillon (2006)
Model-based UI design

CADUI – Computer-Aided Design of User Interface

(formal) descriptions of interactive systems, in terms of existing meta-models, used to design and deploy multiple user interfaces
Model-based UI design

Jean Vanderdonckt
Model-based UI design

At the abstract level, the specification of the interactive system is given by CIM (Computation-Independent Model)
Model-based UI design

At the abstract level, the specification of the interactive system is given by CIM (Computation-Independent Model)

- mission statement
- function reference tree
- use cases
- interaction requirements
- concurrent tasks trees
Model-based UI design

At the abstract level, the specification of the interactive system is given by CIM (Computation-Independent Model)

see also “Requirements Engineering from an HCI Perspective” (A. Sutcliffe, 2015) www.interaction-design.org/encyclopedia/requirements_engineering.html
Model-based UI design

The platform independent description is based on PIM (Platform-Independent Model)
Model-based UI design

The platform independent description is based on PIM (Platform-Independent Model)

- Conceptual Model
  - object model
  - dynamic model
  - functional model
  - presentation model
Model-based UI design

The development focused on a specific platform is based on **PSM (Platform-Specific Model)**
Model-based UI design

The development focused on a specific platform is based on PSM (Platform-Specific Model)

uses software tools of transformation (compilation) based on an application model
Model-based UI design

The concrete implementation is using CM (Code Model)
Model-based UI design

The concrete implementation is using CM (Code Model)

concerns creation/generation of the source-code:
- interface tier
- application tier
- persistence tier
- integration
Model-based UI design
Model-based UI design

Task and Domain

Concrete User Interface 1 (2-D Desktop) → Rendering → Final User Interface

Concrete User Interface 2 (2-D small display) → Rendering → Final User Interface

Concrete User Interface 3 (auditory) → Rendering → Final User Interface

Task and Domain

Abstract User Interface

T1 → T2 → T3 → T4 → T5

T6 → T7

Concrete User Interface

Abstract User Interface

Concrete User Interface
Model-based UI design

(Meta)languages for describing UI

MARIA
(Model-based lAnguage foR Interactive Applications)

UIML – User-Interface Modeling Language

UsiXML – USer Interface eXtensible Markup Language

chess.eecs.berkeley.edu/pubs/940/dreams-18-09-2012.pdf
Model-based UI design

(Meta)languages for describing UI

FXML (JavaFX)
Glade XML
JSX (React.js)
XAML – eXtensible Application Markup Language
XUL – eXtensible User-interface Language
Model-based UI design

MDA (Model-Driven Architecture)

- Computing Independent Model (CIM)
- Model to Model
- Platform Independent Model (PIM)
- Model to Model
- Platform Specific Model (PSM)
- Model to Code
- Source code

UsiXML

- UsiXML models: task, domain
- Graph transformations
- UsiXML model: Abstract user interface
- Graph transformations
- UsiXML model: Concrete user interface
- Rendering
- Final user interface

Jean Vanderdonckt, 2006
designing the abstract UI of a computer game user task modelling via a specific tool (S. Tofan, A. Pradais, S. Buraga, 2009)
Support for multiple displays
(Grolaux & Vanderdonckt, 2005)

detach ▶ migrate ▶ plastify
case study
Support for multiple displays  
(Grolaux & Vanderdonckt, 2005)

- detach
- migrate
- plastify

for migration, rules and/or design patterns could be used
Support for multiple displays
(Grolaux & Vanderdonckt, 2005)

the use of rules – example:

\[ \forall x \in T_s : x = \text{input and} \\
(x.\text{type} = \text{“text” or} x.\text{type} = \text{“password” or} x.\text{type} = \text{NULL}) \Rightarrow \text{AddNode (“textComponent”, idText)} \]

where idText = NodeValue (T_t)
case study

substitution  removal  moving

examples of actions that could be performed for migration and/or creating PUI – revisit responsive Web design
Model-based UI design

For other examples, read G. Meixner, G. Calvary, J. Coutaz (Eds.), *Introduction to Model-Based User Interfaces*, W3C Working Group Note (2014)

www.w3.org/TR/mbui-intro/
No more programming: only models

Target Applications, Domains
- Business applications
- Embedded systems
- Information systems
- Web, desktop, mobile apps
- Automated, batch systems

Notations & Engineering Tools
- Physical models
- World models
- Scene model
- Task model, context model, UML,…
- Entity-relationship model
- Attribute model
- State-transition diagrams
- Screen definitions
- Nothing

User Interface Interaction Techniques
- Graphical User Interfaces
- Mixed Reality User Interfaces
- Virtual Reality User Interfaces
- Multi-platform User Interfaces
- Character UIs
- No Interaction Technique

Invisible UI
- Embedded UIs
- Tangible UIs
- Embodied UIs
- Virtual Reality User Interfaces
- Mixed Reality User Interfaces
- Multi-platform User Interfaces
- Character UIs
- No Interaction Technique

All Applications
- Business applications
- Information systems
- 3D Applications
- Web, desktop, mobile apps
- Automated, batch systems
- Embedded systems
- Command & control systems, games

In near future
interactivity in the context of software engineering

UI quality, testing, models
next episode: data visualization