How to create accessible SW

Zdeněk Míkovec

dept. of computer graphics and interaction

Czech Technical University in Prague

What it means accessible?

- Visual impairment
  - control: keyboard navigation
  - presentation: audio/haptic; large graphics
  - data input: keyboard, gestures
- Motor impairment
  - control: large controls for direct manipulation (mouse, touch), keyboard navigation
  - presentation: <no limitations>
  - data input: keyboard, virtual keyboard, voice, direct manipulation
- Hearing impairment
  - control: keyboard/mouse navigation
  - presentation: visual/haptic
  - data input: keyboard, direct manipulation (mouse, touch)

What it means accessible?

- What about combinations?
  - Deaf-blind impairment
  - Motor-blind impairment
  - Motor-deaf impairment

Accessible RIA

RIA

- What is Rich Internet Application (RIA)?
  - New approach on Internet
  - Websites look more like desktop applications
    - Online documents
    - E-mail
    - Instant messaging
    - Calendar
    - Social networking
- Number of RIA rapidly increasing
- Possible benefits for handicapped people working with Internet

RIA +/-

- RIA pros
  - Higher comfort
    - Suggestions
    - Immediate feedback
  - Better user experience
    - Drag&Drop
    - Sophisticated components (tree, collapsible panel)
  - Dynamics
    - Chat
    - Shared data
    - Collaboration

- RIA cons
  - Not transparent
    - Many concurrent changes
    - Complex components
  - Unclear structure
    - Content (not in HTML)
    - Semantics in code of scripts
  - Problematic keyboard navigation
  - Difficult controlling of components
  - Dependency on JavaScript

WAI-ARIA

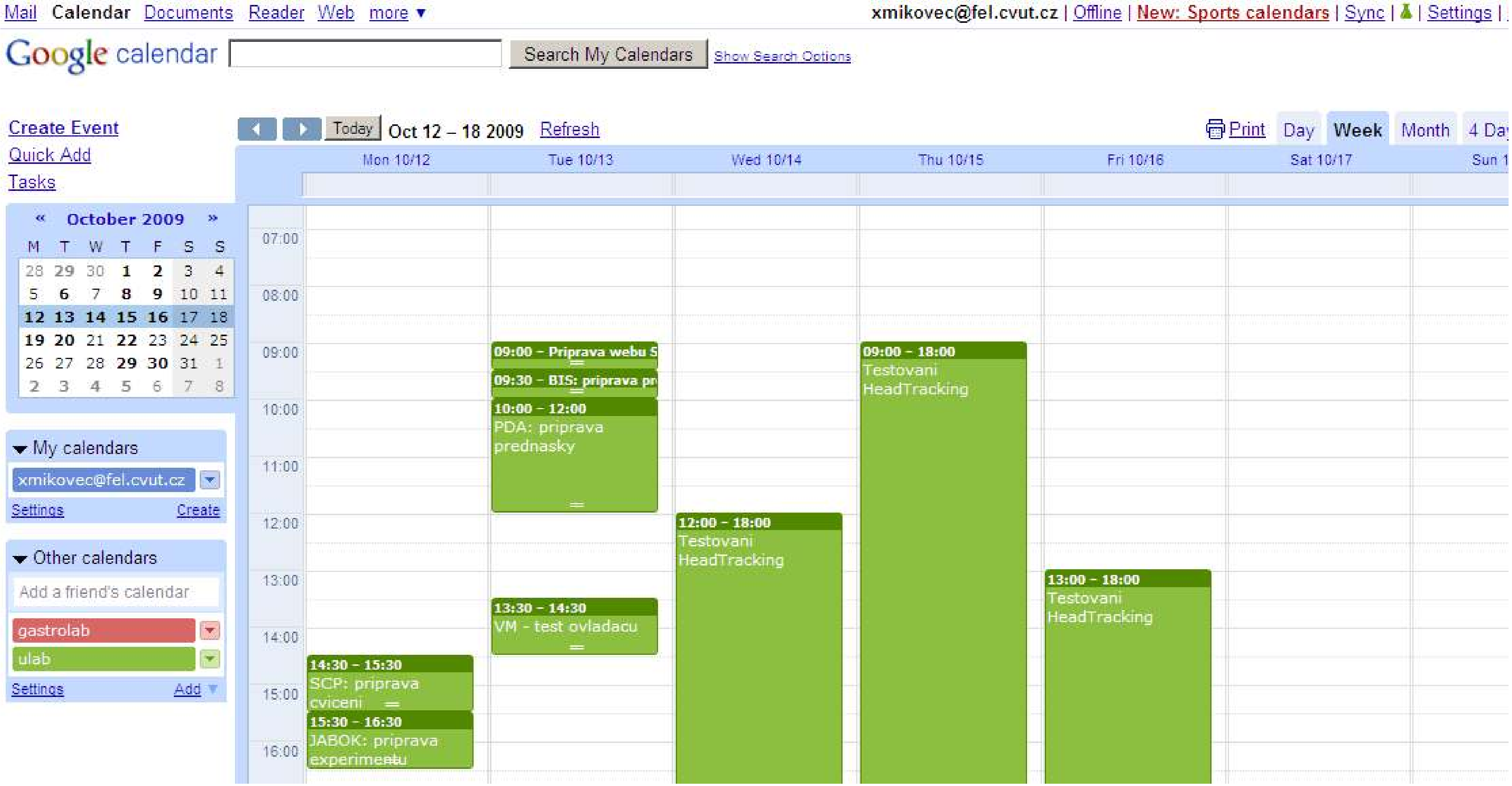
- RIA is less accessible than "standard" web pages
  - How this can be solved?
- Web Accessibility Initiative (WAI)
  - part of W3C consortium
- Accessible Rich Internet Application suite of W3C (WAI-ARIA)
  - Working draft
  - Recommendations on how to make RIA accessible
    - Additional metadata
    - Bring back the semantics into HTML code
  - No restrictions on current functionality

a11y issues tackled by ARIA

- Keyboard-only operation impossible
- Orientation in content
  - Missing appropriate labels — aria-labeledby
  - Hiding of some content without announcement — aria-expanded
    - Tab panels
    - Collapsible panels
- Controlling of UI components
  - Wrong or missing tab order
  - Focus stacking or disappearing

    tabindex
  - Wrong or missing status announcement
    - Checkbox
    - Radio Button — checked
    - Slider

RIA and a11y



Oct 12 – 18 2009

October 2009

M	T	W	T	F	S	S
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

My calendars: xmikovec@fel.cvut.cz

Other calendars: gastrolab, ulab

Mon 10/12 · Tue 10/13 · Wed 10/14 · Thu 10/15 · Fri 10/16 · Sat 10/17 · Sun 1

07:00 · 08:00 · 09:00 · 10:00 · 11:00 · 12:00 · 13:00 · 14:00 · 15:00 · 16:00

Mon 10/12:
- 14:30 – 15:30 SCP: priprava cviceni
- 15:30 – 16:30 JABOK: priprava experimentu

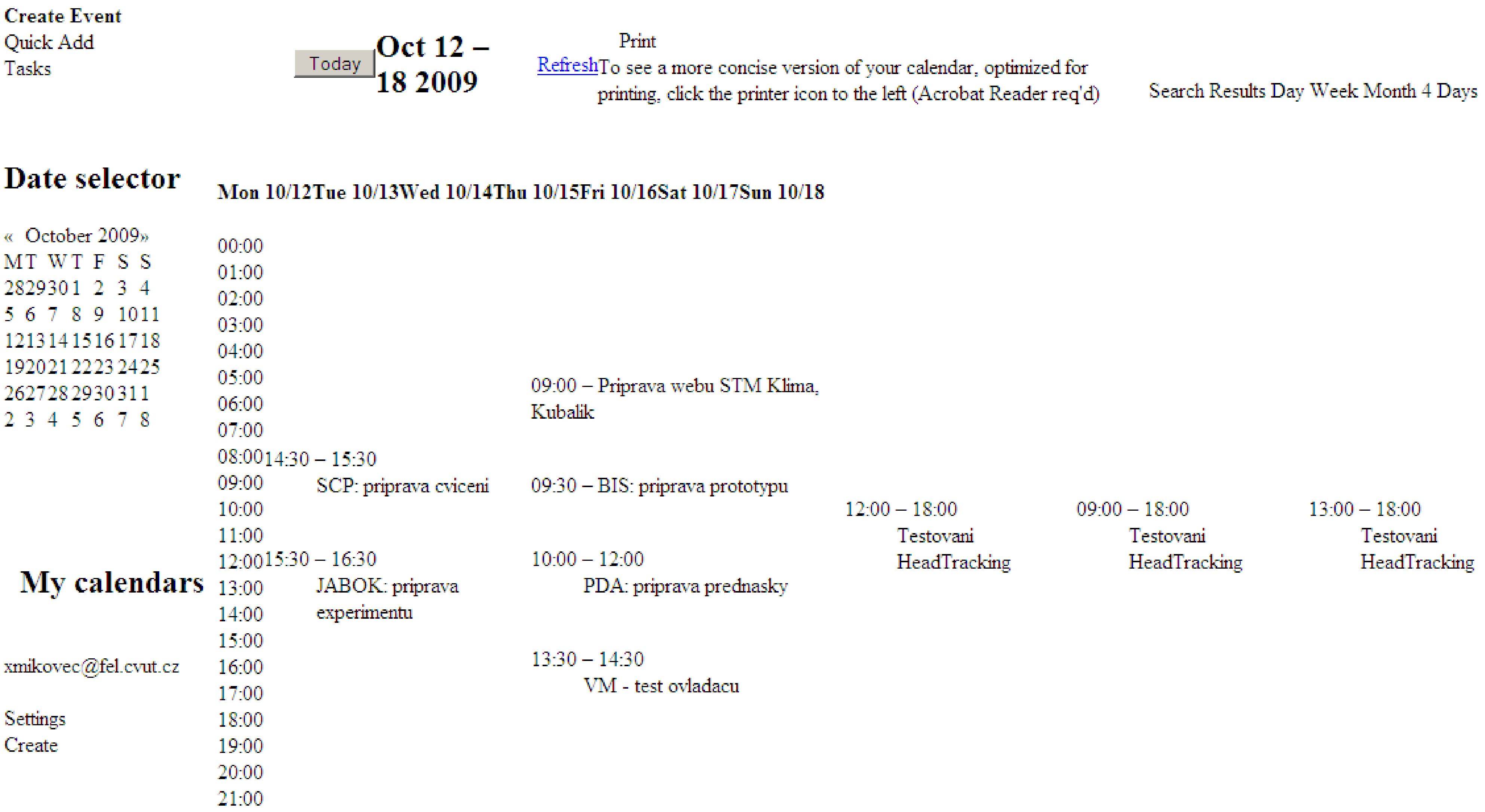
Tue 10/13:
- 09:00 – Priprava webu S
- 09:30 – BIS: priprava pr
- 10:00 – 12:00 PDA: priprava prednasky
- 13:30 – 14:30 VM - test ovladacu

Wed 10/14:
- 12:00 – 18:00 Testovani HeadTracking

Thu 10/15:
- 09:00 – 18:00 Testovani HeadTracking

Fri 10/16:
- 13:00 – 18:00 Testovani HeadTracking

RIA and a11y



Create Event
Quick Add
Tasks

Today

Oct 12 – 18 2009

Print

Refresh To see a more concise version of your calendar, optimized for printing, click the printer icon to the left (Acrobat Reader req'd)

Search Results Day Week Month 4 Days

Date selector

« October 2009»

M	T	W	T	F	S	S
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

My calendars

xmikovec@fel.cvut.cz

Settings
Create

Mon 10/12Tue 10/13Wed 10/14Thu 10/15Fri 10/16Sat 10/17Sun 10/18

00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00

14:30 – 15:30
SCP: priprava cviceni

15:30 – 16:30
JABOK: priprava experimentu

09:00 – Priprava webu STM Klima, Kubalik

09:30 – BIS: priprava prototypu

10:00 – 12:00
PDA: priprava prednasky

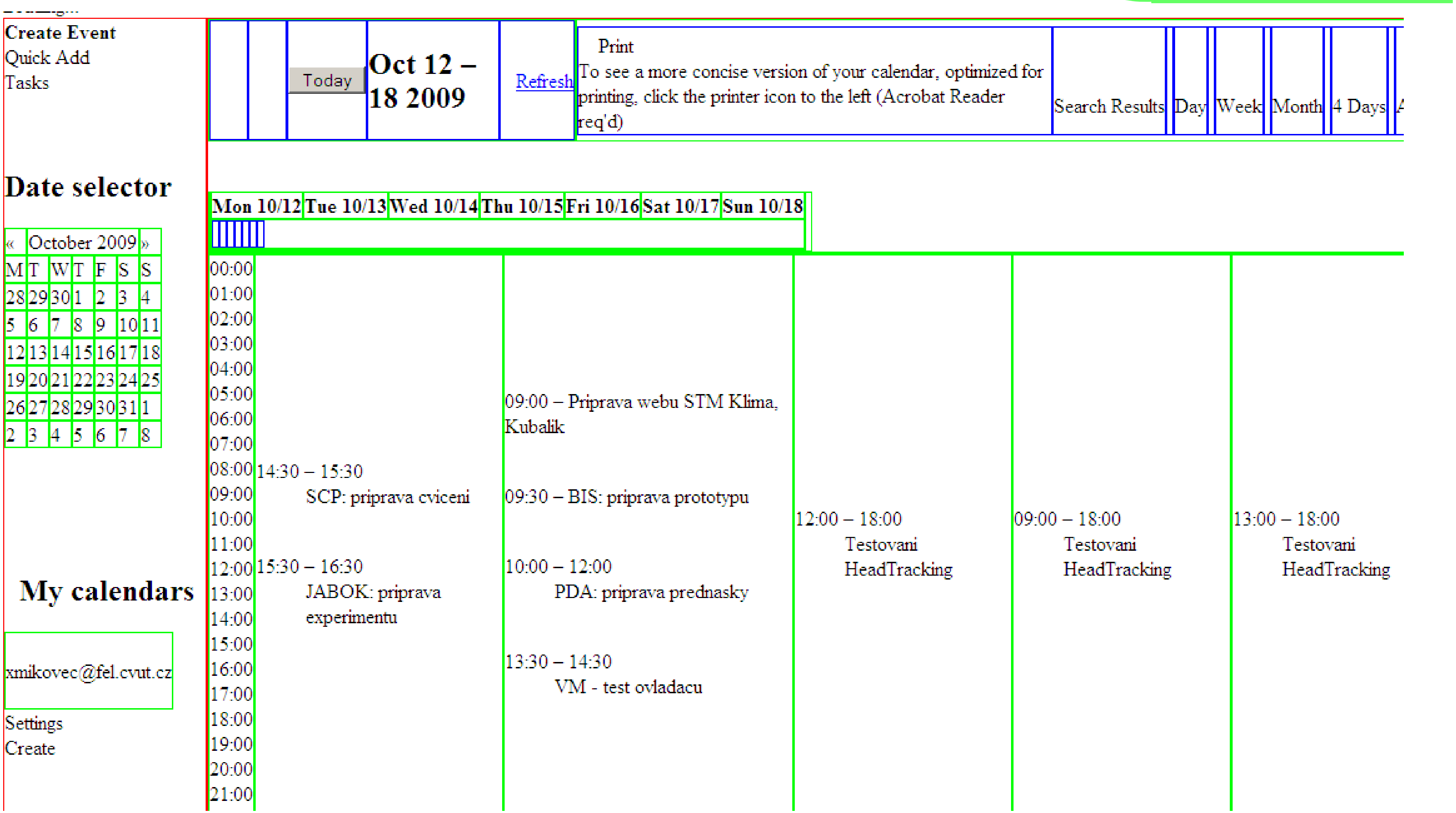
13:30 – 14:30
VM - test ovladacu

12:00 – 18:00
Testovani HeadTracking

09:00 – 18:00
Testovani HeadTracking

13:00 – 18:00
Testovani HeadTracking

RIA and a11y



Create Event
Quick Add
Tasks

Today

Oct 12 – 18 2009

Refresh

Print
To see a more concise version of your calendar, optimized for printing, click the printer icon to the left (Acrobat Reader req'd)

Search Results | Day | Week | Month | 4 Days

Date selector

«	October 2009	»				
M	T	W	T	F	S	S
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Mon 10/12 | **Tue 10/13** | **Wed 10/14** | **Thu 10/15** | **Fri 10/16** | **Sat 10/17** | **Sun 10/18**

00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00

14:30 – 15:30
SCP: priprava cviceni

15:30 – 16:30
JABOK: priprava experimentu

09:00 – Priprava webu STM Klima, Kubalik

09:30 – BIS: priprava prototypu

10:00 – 12:00
PDA: priprava prednasky

13:30 – 14:30
VM - test ovladacu

12:00 – 18:00
Testovani HeadTracking

09:00 – 18:00
Testovani HeadTracking

13:00 – 18:00
Testovani HeadTracking

My calendars

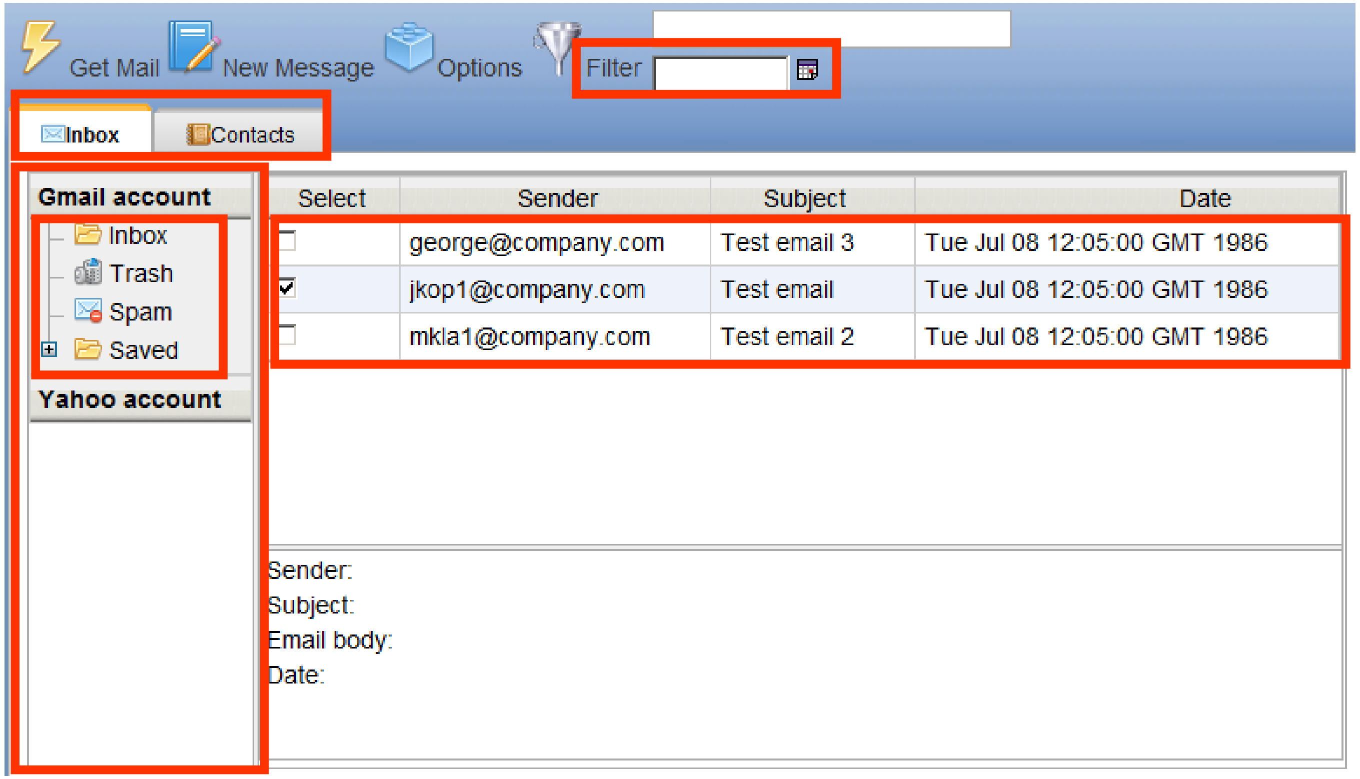
xmikovec@fel.cvut.cz

Settings
Create

Creation of ARIA

Rich environment of RIA

- Modern RIA applications are build from components
  - Tree
  - Tabs
  - Accordion
  - Grid
  - etc.



Select	Sender	Subject	Date
□	george@company.com	Test email 3	Tue Jul 08 12:05:00 GMT 1986
☑	jkop1@company.com	Test email	Tue Jul 08 12:05:00 GMT 1986
□	mkla1@company.com	Test email 2	Tue Jul 08 12:05:00 GMT 1986

Sender:
Subject:
Email body:
Date:

Rich world of RIA

- Web environment is extremely variable
- Accessibility depends on:
  - Type of OS
    - Windows, Linux, Mac, …
  - Type of Web browser
    - Firefox, IE, Safari, Chrome, …
  - Type of Screen reader
    - Jaws, NVDA, Orca, …

- No configuration is 100% ARIA compliant

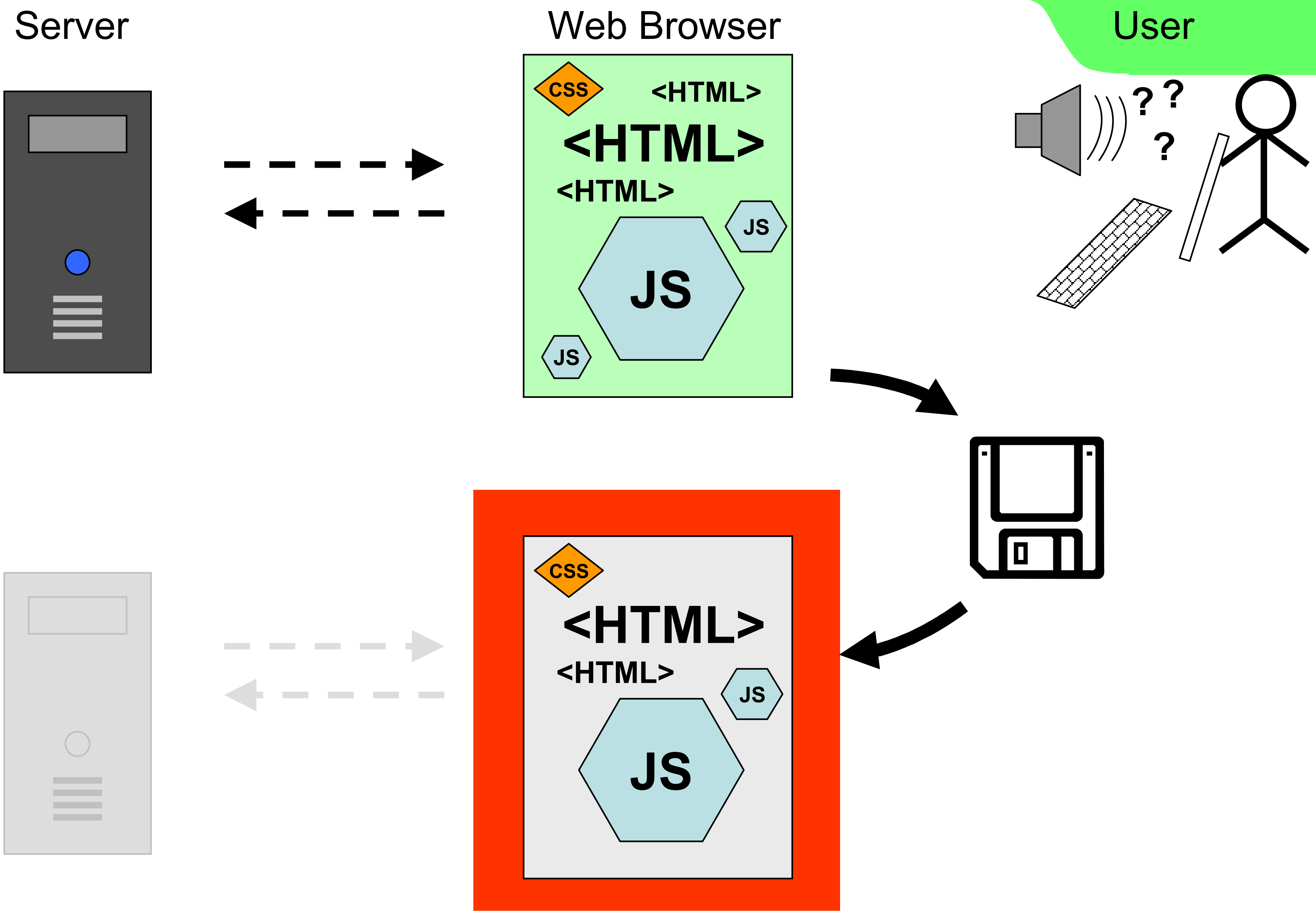
Three steps towards accessible RIA

1. Accessibility of RIA components
2. Accessibility of RIA applications
3. Testing of application accessibility

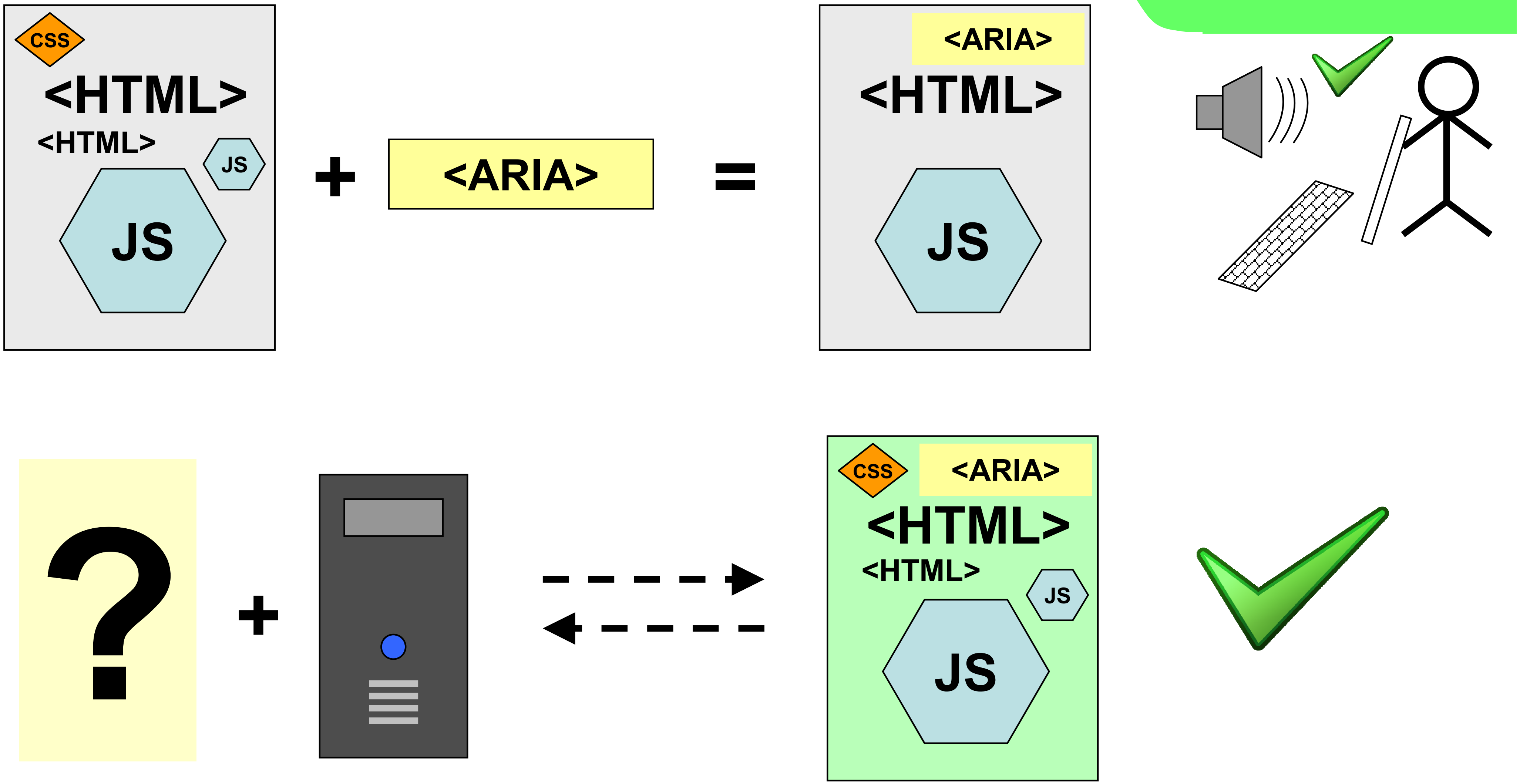
Three steps towards accessible RIA

1. **Accessibility of RIA components**
2. Accessibility of RIA applications
3. Testing of application accessibility

Offline component prototype



Offline component prototype



Accessibility of RIA components

1. Create offline component prototype
2. Simplify the component architecture
3. Add WAI-ARIA attributes into offline component prototypes
   - Implementing ARIA attributes
   - Implementing keyboard navigation
4. Implement changes back to the server
   - Test whether results are accessible

Three steps towards accessible RIA

1. Accessibility of RIA components
2. **Accessibility of RIA applications**
3. Testing of application accessibility

Issues to be implemented

- Navigation on the page
- Relationships between components
- Dynamic changes of presented information

- Created set of 11 heuristics based on Nealson’s usability heuristics

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible
6. Create proper linkage of controls, labels and messages
7. Distinguish all components
8. Define complete keyboard operation and where possible, standardize
9. Define document structure with ARIA landmarks
10. Provide a logical tab order
11. Use buttons for functions and links for linking

Heuristics

Screen readers and another assistive technologies use **several browsing modes**. Make sure all parts of the web page are accessible at least with **“virtual cursor”** and **“forms mode”**. In forms mode all information in the form area must be linked to one of the form elements as a label or description.

1.

2.

3.

4.

5. oon as

6. s

7. Distinguish all components

8. Define complete keyboard operation and where possible, standardize

9. Define document structure with ARIA landmarks

10. Provide a logical tab order

11. Use buttons for functions and links for linking

Heuristics

1. Design with screen reader modes in mind
2.
3.
4.
5. soon as
6. s
7. Distinguish all components
8. Define complete keyboard operation and where possible, standardize
9. Define document structure with ARIA landmarks
10. Provide a logical tab order
11. Use buttons for functions and links for linking

> Icons and other similar visual elements that carry information to the user should have a textual alternative available. The only **exception** is when a non-textual element is used for **decoration** or layout purposes.

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4.
5. soon as
6. s
7. Distinguish all components
8. Define complete keyboard operation and where possible, standardize
9. Define document structure with ARIA landmarks
10. Provide a logical tab order
11. Use buttons for functions and links for linking

Headings are the only elements with various levels of importance. They are often **used to scan the content** and should be used when possible to denote sections.

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. soon as
6. s
7. Distinguish all components
8. Define complete keyboard operation and where possible, standardize
9. Define document structure with ARIA landmarks
10. Provide a logical tab order
11. Use buttons for functions and links for linking

When showing larger section **move focus** to the section. When showing a **tooltip** all content should be **connected as description**.

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible

Use **on-the-fly validation** where possible. Use **live regions** to communicate asynchronous messages.

6. s
7.
8. Define complete keyboard operation and where possible, standardize
9. Define document structure with ARIA landmarks
10. Provide a logical tab order
11. Use buttons for functions and links for linking

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible
6. Create proper linkage of controls, labels and messages
7. [illegible]
8. [illegible] le,

Connect menus with corresponding **dynamically loaded sections** using aria-controls.

9. Define document structure with ARIA landmarks
10. Provide a logical tab order
11. Use buttons for functions and links for linking

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible
6. Create proper linkage of controls, labels and messages
7. Distinguish all components
8. Define complete keyboard operation and where possible,
9.
10. Provide a logical tab order
11. Use buttons for functions and links for linking

All components that have their **Roles** identified in WAI-ARIA should be marked using appropriate Role.

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible
6. Create proper linkage of controls, labels and messages
7. Distinguish all components
8. Define complete keyboard operation and where possible, standardize
9. Define document structure with ARIA landmarks
10.
11.

Use design patterns defined in WAI-ARIA or DHTML Style Guide to determine the proper **keyboard navigation** before implementing your own.

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible
6. Create proper linkage of controls, labels and messages
7. Distinguish all components
8. le,
9. Define document structure with ARIA landmarks
10. Provide a logical tab order
11. Use buttons for functions and links for linking

Identify as **many** common structure parts as possible and apply WAI-ARIA **landmark roles** to them.

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible
6. s
7.
8. le,
9.
10. Provide a logical tab order
11. Use buttons for functions and links for linking

Menus should be **close** in the means of tab order **to the sections** they are affecting. Tab order is important as it is **used to quickly scan** the page for interactive components. If the tab order is faulty, the mental model of the web page will likely be incorrect.

Heuristics

1. Design with screen reader modes in mind
2. Provide text alternative for all non-textual elements
3. Use headings to mark important areas
4. Handle hidden section appropriately
5. Communicate important information and feedback as soon as possible
6. Create proper linkage of controls, labels and messages
7. Distinguish all components
8. le,
9.
10.
11.

Make clear **distinction between buttons and links**. For all functions that are available on the page use buttons. For navigation purposes and for linking to other pages or anchoring, use links.

Three steps towards accessible RIA

1. Accessibility of RIA components
2. Accessibility of RIA applications
3. **Testing of application accessibility**

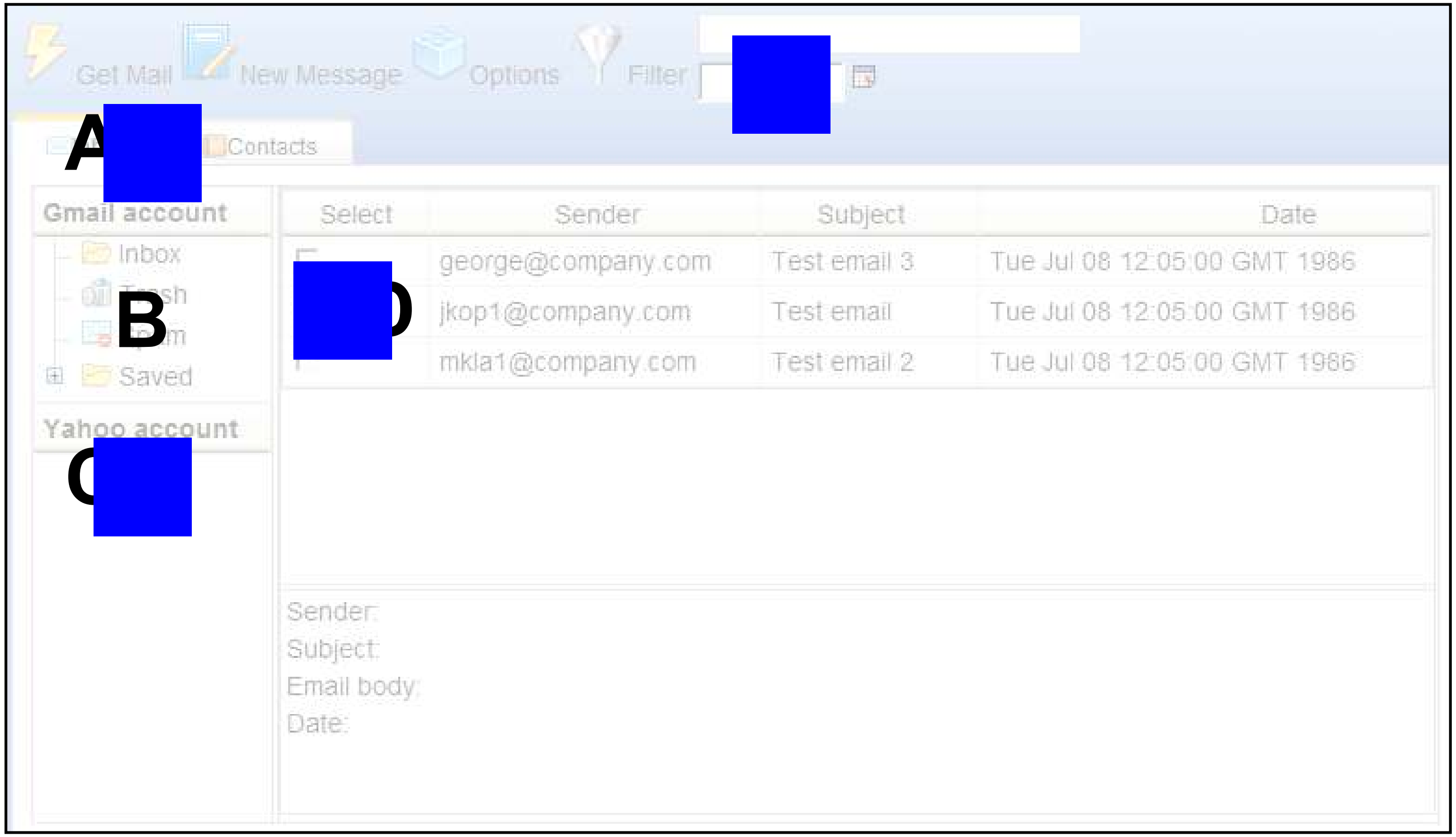
Testing of application accessibility

- Developer is typically NOT:
  - Blind user
  - Used to operate screen reader
- Need for accessibility testing with blind users

- Early stages of development means:
  - Poor accessibility
  - Need for support of accessibility testing

View of blind user

- User sees some components just partially or they seem missing

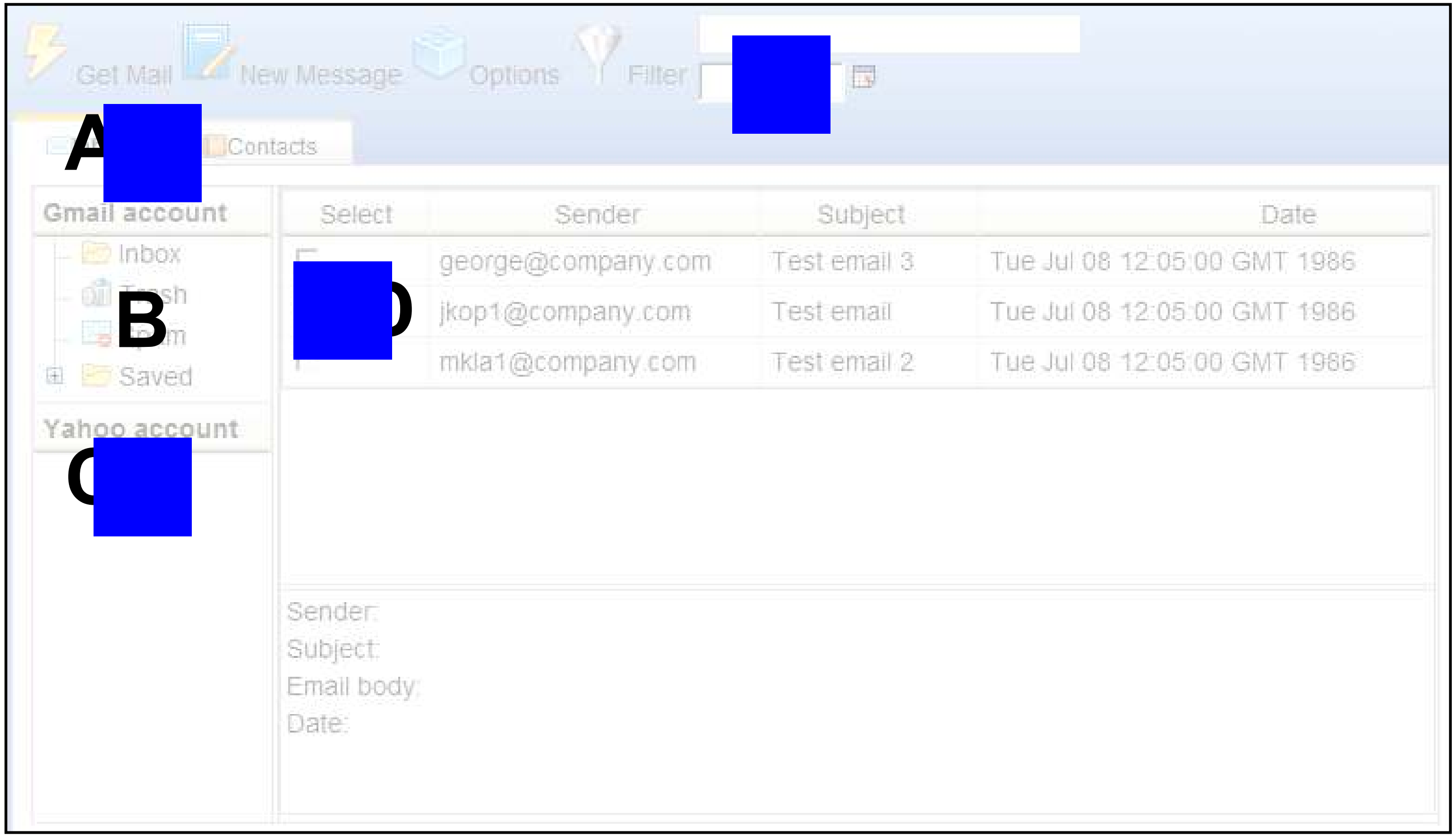


Select	Sender	Subject	Date
	george@company.com	Test email 3	Tue Jul 08 12:05:00 GMT 1986
	jkop1@company.com	Test email	Tue Jul 08 12:05:00 GMT 1986
	mkla1@company.com	Test email 2	Tue Jul 08 12:05:00 GMT 1986

Sender:
Subject:
Email body:
Date:

View of developer

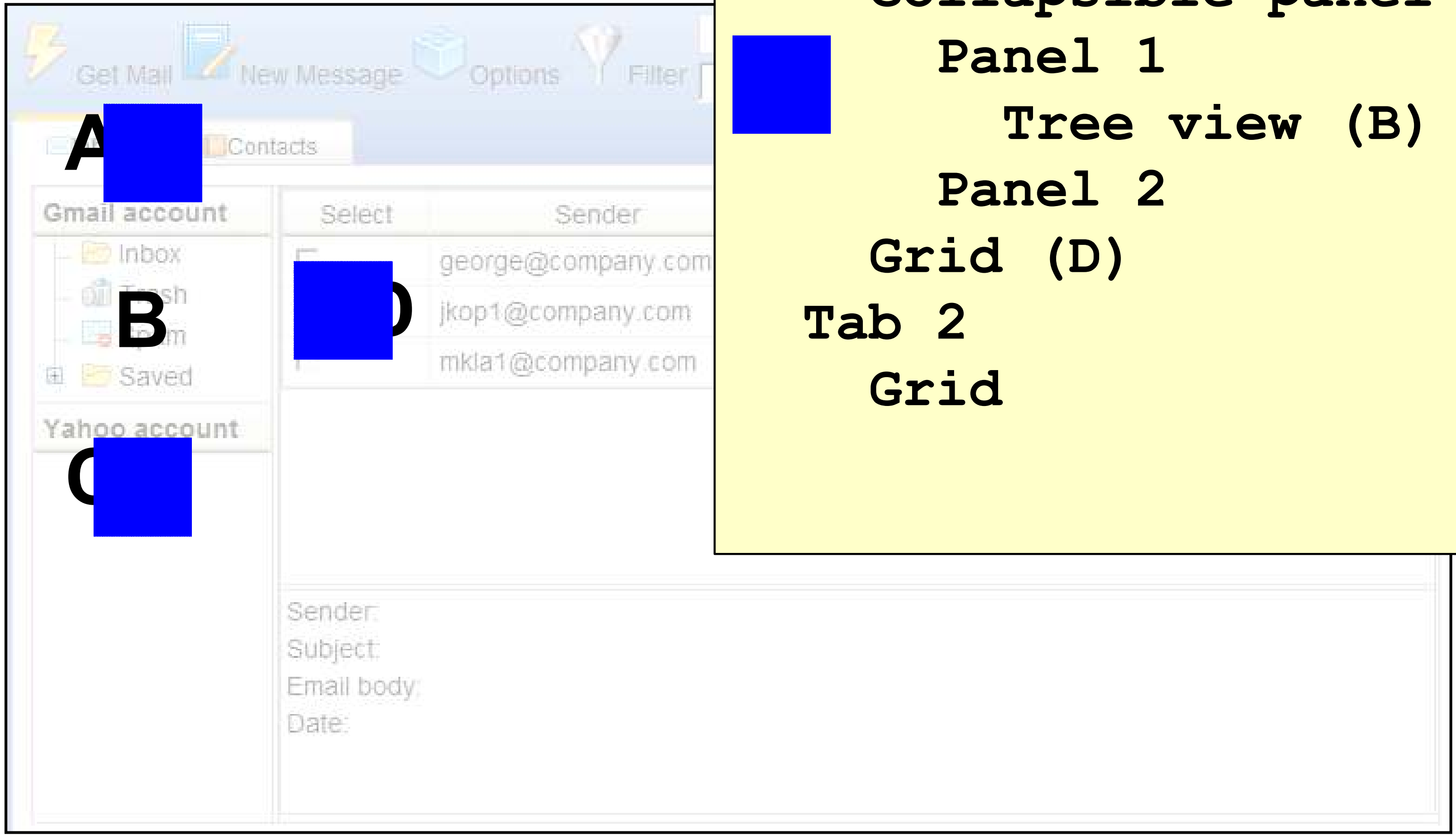
- **Developer sees all the components**



Select	Sender	Subject	Date
	george@company.com	Test email 3	Tue Jul 08 12:05:00 GMT 1986
	jkop1@company.com	Test email	Tue Jul 08 12:05:00 GMT 1986
	mkla1@company.com	Test email 2	Tue Jul 08 12:05:00 GMT 1986

Sender:
Subject:
Email body:
Date:

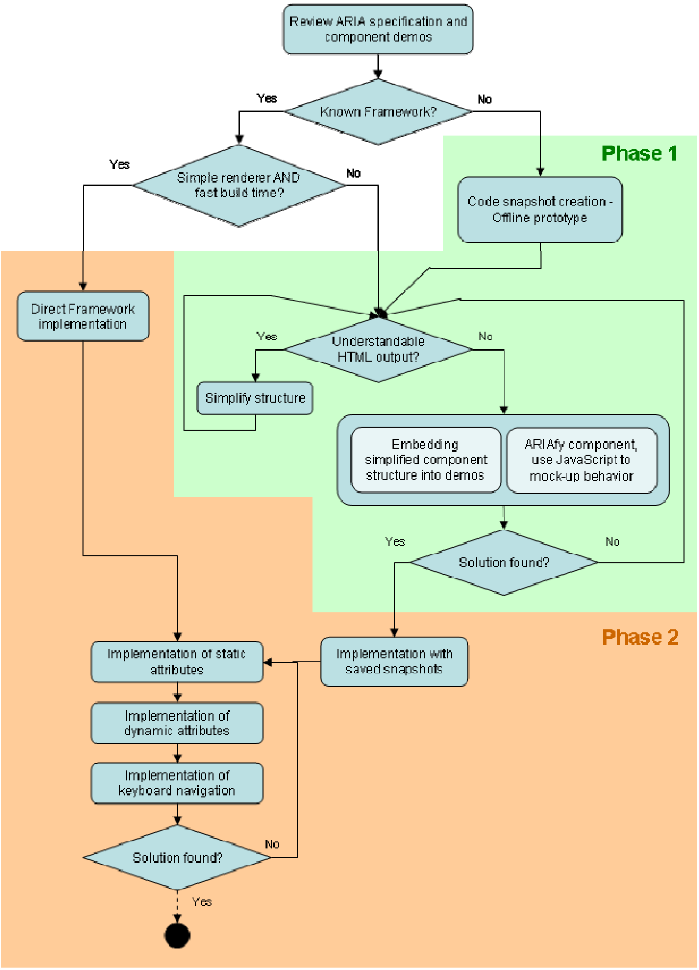
View of user with description



- Datepicker (E)
- Tablist (A)
  - Tab 1
    - Collapsible panel (C)
      - Panel 1
        - Tree view (B)
      - Panel 2
    - Grid (D)
  - Tab 2
    - Grid

RIA accessibility - summary

- RIA accessibility is complicated and complex process
  - Has to be treated in phases
- Valid testing is complicated
  - Support of blind tester needed



Phase 1

Phase 2

Review ARIA specification and component demos

Known Framework?

Yes

No

Simple renderer AND fast build time?

Yes

No

Code snapshot creation - Offline prototype

Direct Framework implementation

Understandable HTML output?

Yes

No

Simplify structure

Embedding simplified component structure into demos

ARIAfy component, use JavaScript to mock-up behavior

Solution found?

Yes

No

Implementation with saved snapshots

Implementation of static attributes

Implementation of dynamic attributes

Implementation of keyboard navigation

Solution found?

No

Yes

Simulation and inspection tools

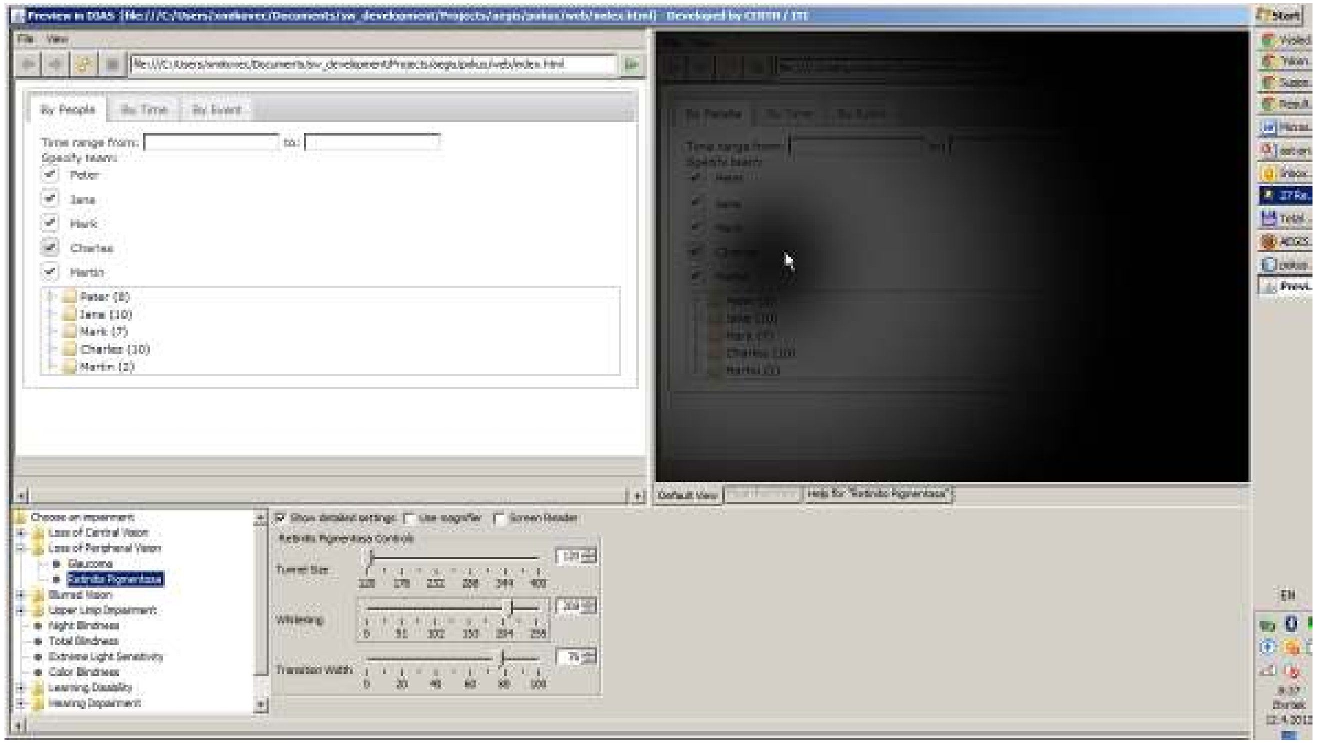
Color checkers

- Simulates color blindness and other visual impairments

- Web applications
  - http://webaim.org/resources/contrastchecker/
  - http://www.snook.ca/technical/colour_contrast/colour.html
- Firefox extensions
  - Web Developer
  - Color Checker

DIAS: Disability Impairment Approximation Tool

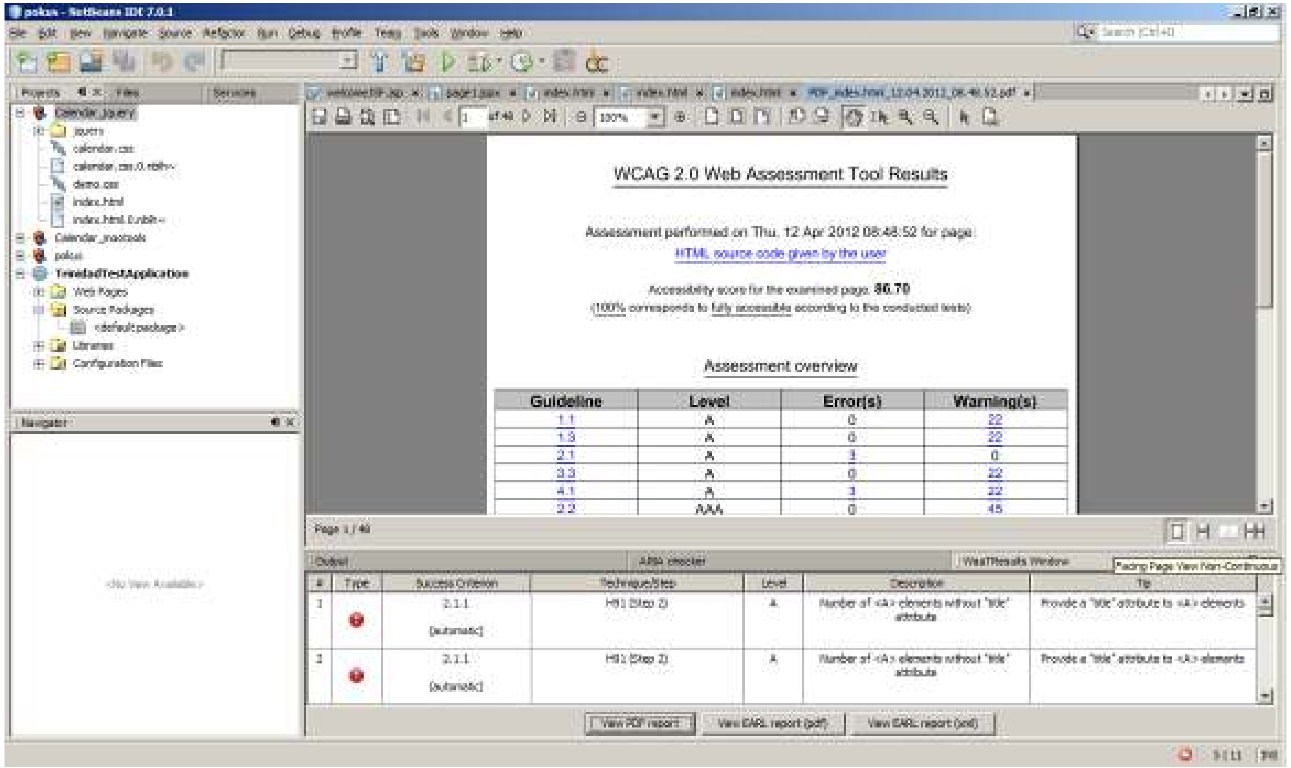
- **Simulation of impairments**
- **Inspection of the code**
- **Integration with IDE (NetBeans)**
- **http://sourceforge.net/projects/diasnb/**



WaaT: Web A11y assessment tool

- Assesses the a11y of web applications
- Integrated with IDE (NetBeans)

- http://sourceforge.net/projects/waat/



WCAG 2.0 Web Assessment Tool Results

Assessment performed on Thu, 12 Apr 2012 08:48:52 for page:
HTML source code given by the user

Accessibility score for the examined page: **86.70**
(100% corresponds to fully accessible according to the conducted tests)

Assessment overview

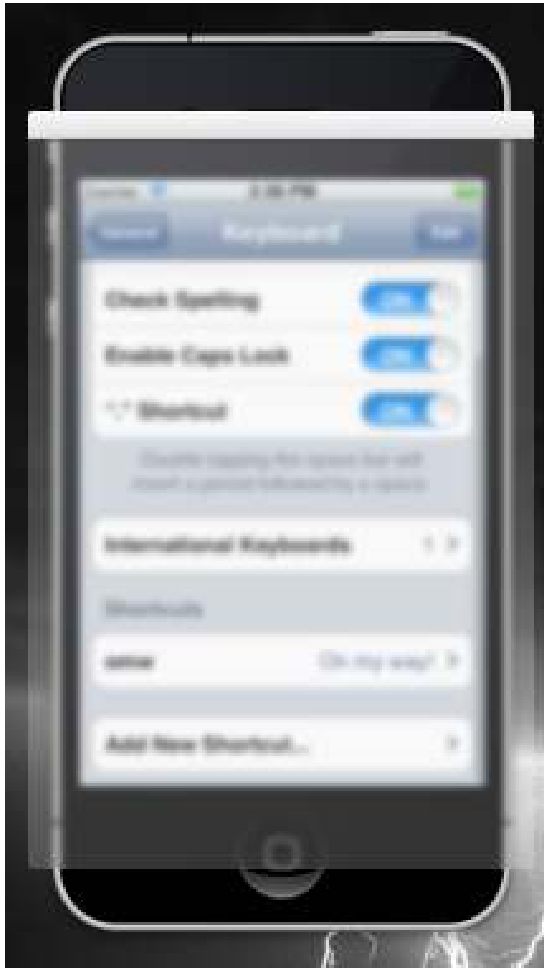
Guideline	Level	Error(s)	Warning(s)
1.1	A	0	22
1.3	A	0	22
2.1	A	1	0
3.3	A	0	22
4.1	A	1	22
2.2	AAA	0	45

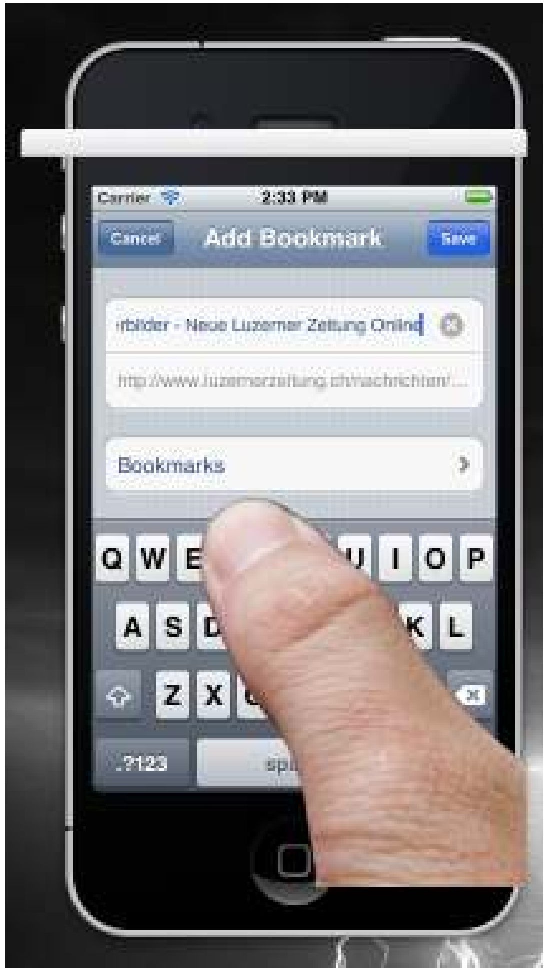
#	Type	Success Criterion	Technique/Step	Level	Description	Tip
1	●	2.1.1 [automatic]	H91 (Step 2)	A	Number of <A> elements without "title" attribute	Provide a "title" attribute to <A> elements
2	●	2.1.1 [automatic]	H91 (Step 2)	A	Number of <A> elements without "title" attribute	Provide a "title" attribute to <A> elements

MIS tool

- Simulates visual impairments
- Simulates mobile environment

- https://cent.felk.cvut.cz/hci/accessible/index.php?page=mis




Carrier
2:33 PM
Cancel
Add Bookmark
Save
rbilder - Neue Luzerner Zeitung Online
http://www.luzernerzeitung.ch/nachrichten/...
Bookmarks
Q W E U I O P
A S D K L
Z X
.?123
spa

Computer vs mobile environment

- Computer
  - Keyboard, mouse
  - Large LCD monitor
  - Steady environment (indoors)
    - Artificial lightning
    - Fixed position
    - Planned activity

- Mobile
  - Touch screen, HW buttons
  - Small display
  - Changing environment (outdoor)
    - Sun, darkness
    - Movements
    - Frequent unpredictable interruption

Simulation of mobile environment in office?

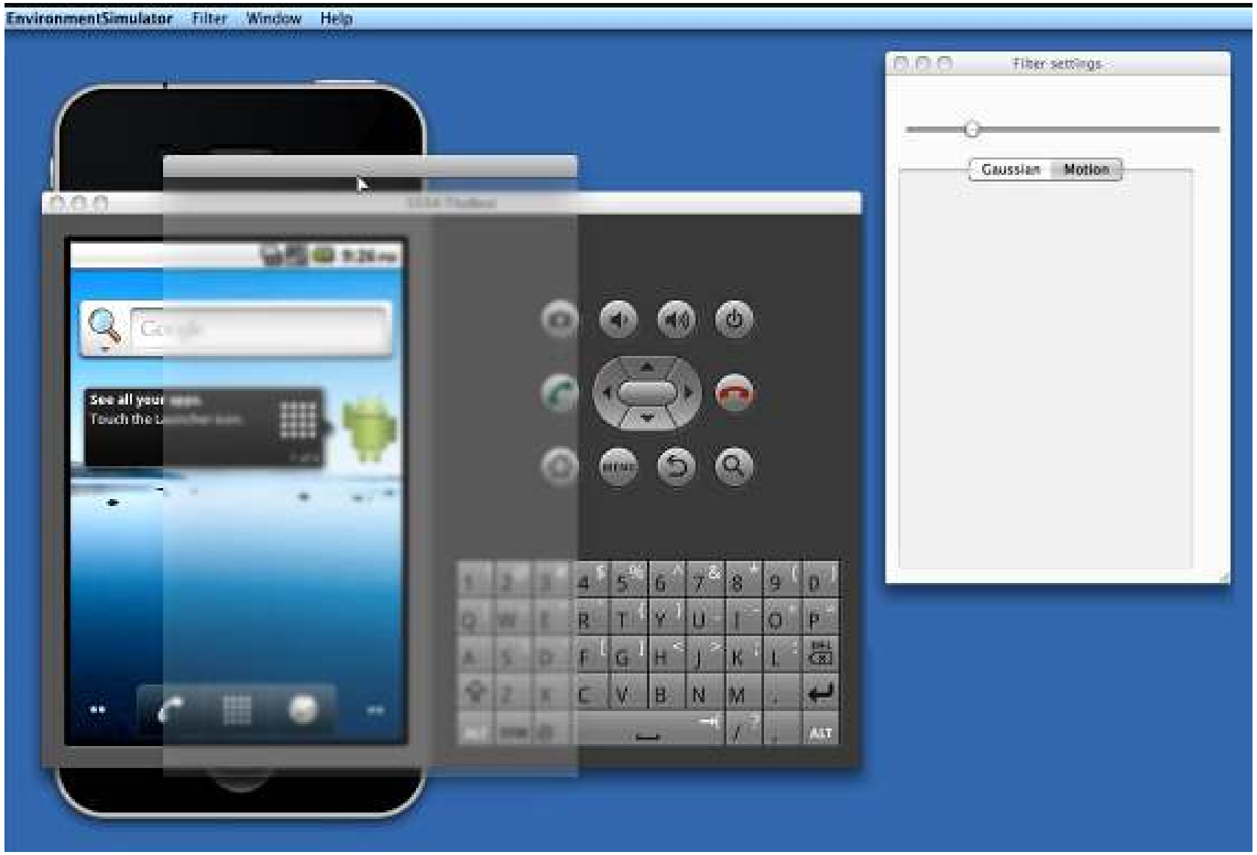
- More factors should be taken into account
- Combination of factors also important
- Difficult to simulate in office environment

Hard to imagine

Simulation desired

Mobile Impairment Simulation tool

- Filter overlay window
- Independent on mobile platform
  - Android
  - BlackBerry
  - iOS
  - Symbian
  - Windows 7
  - …

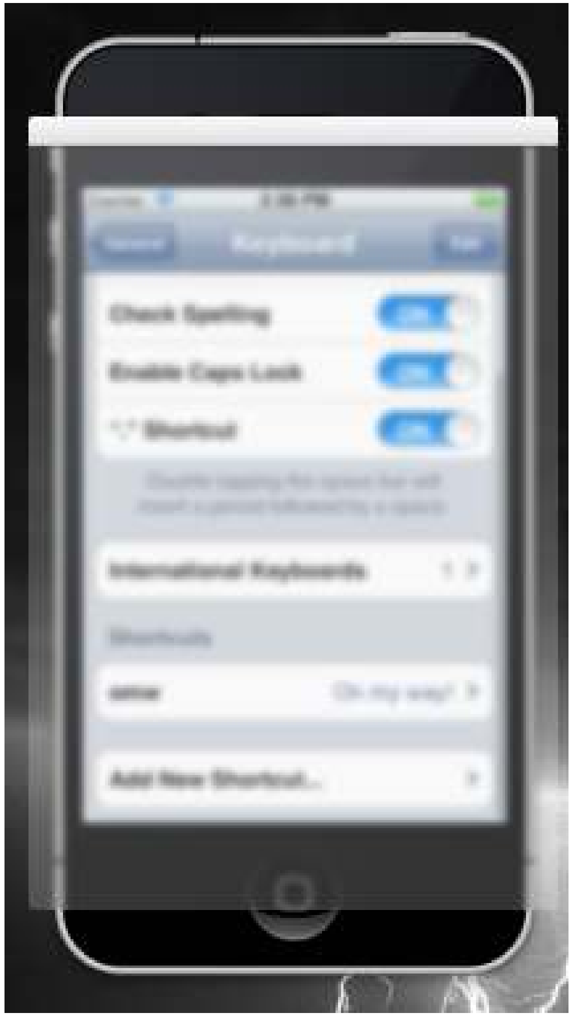


Categories of simulated issues

- Visual impairment
  - Tunnel vision
  - Blurred vision
  - Color blindness
- Occlusion of the display
  - Finger occlusion
- Reflection on the display
  - Static reflections
  - Display tremor
- Combined simulations

Visual impairment

Blurred vision



Carrier 2:35 PM

General Keyboard Edit

Check Spelling ON

Enable Caps Lock ON

"." Shortcut ON

Double tapping the space bar will insert a period followed by a space.

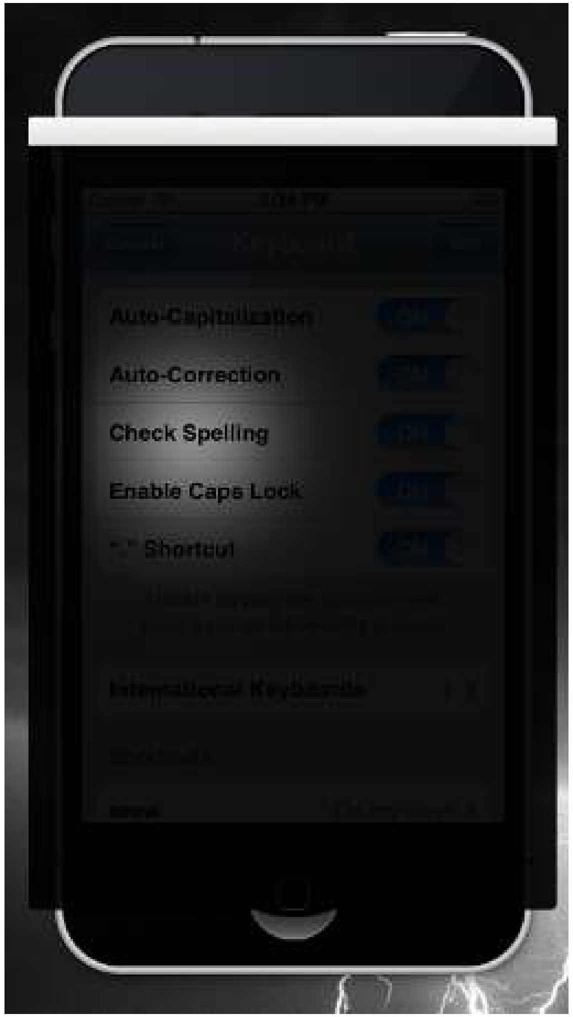
International Keyboards 1 >

Shortcuts

omw On my way! >

Add New Shortcut... >

Tunnel vision



Carrier 2:35 PM

General Keyboard Edit

Auto-Capitalization ON

Auto-Correction ON

Check Spelling ON

Enable Caps Lock ON

"." Shortcut ON

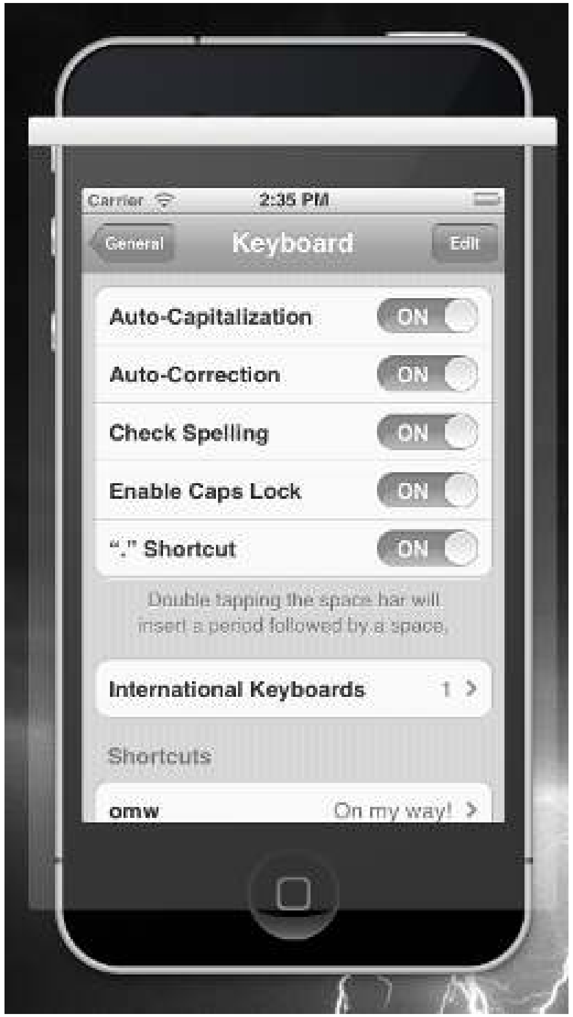
Double tapping the space bar will insert a period followed by a space.

International Keyboards 1 >

Shortcuts

omw On my way! >

Color blindness



Carrier 2:35 PM

General Keyboard Edit

Auto-Capitalization ON

Auto-Correction ON

Check Spelling ON

Enable Caps Lock ON

“.” Shortcut ON

Double tapping the space bar will insert a period followed by a space.

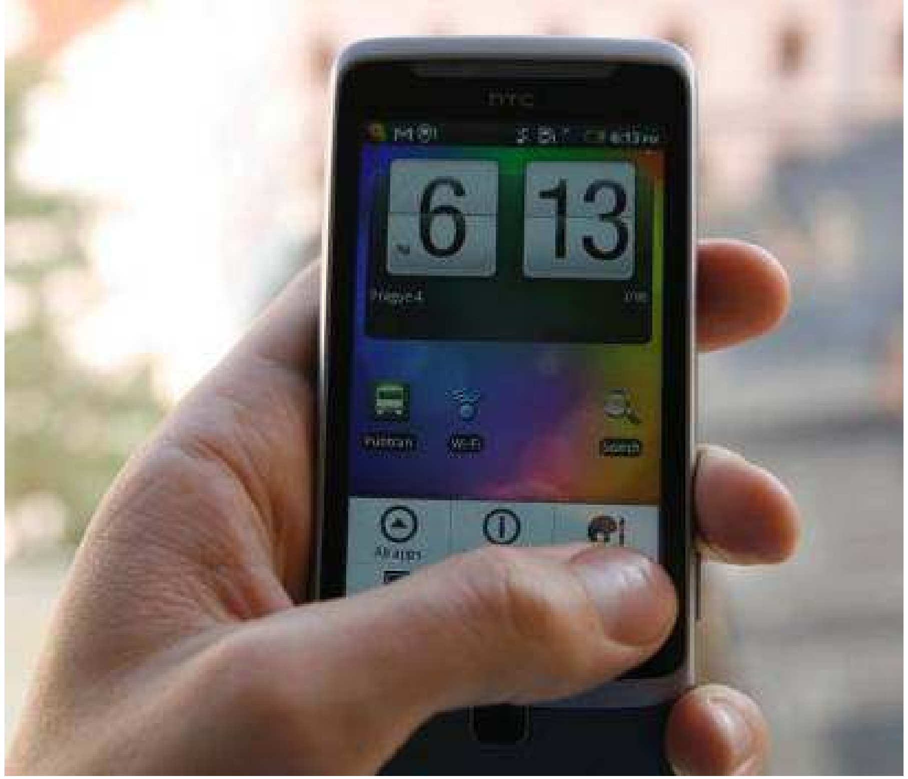
International Keyboards 1 >

Shortcuts

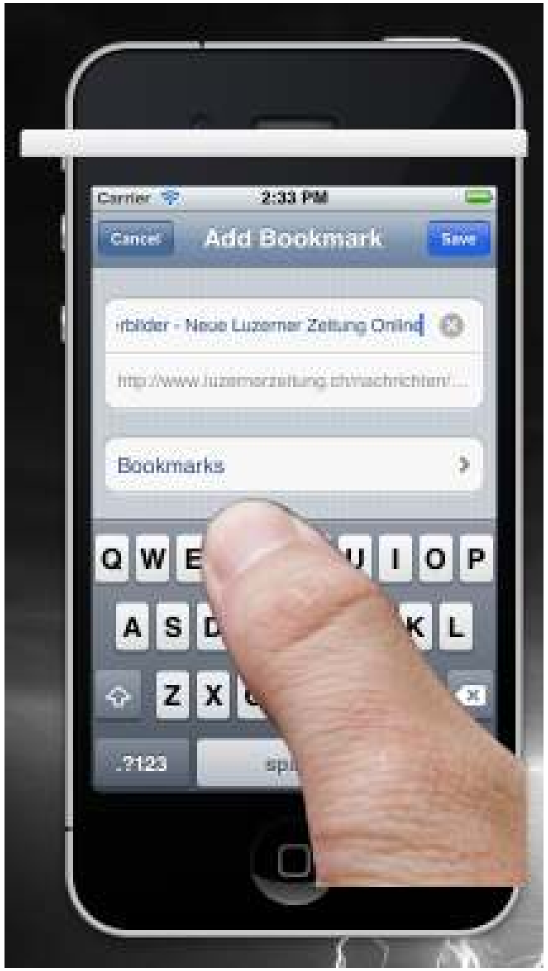
omw On my way! >

Occlusion of the display

Real world

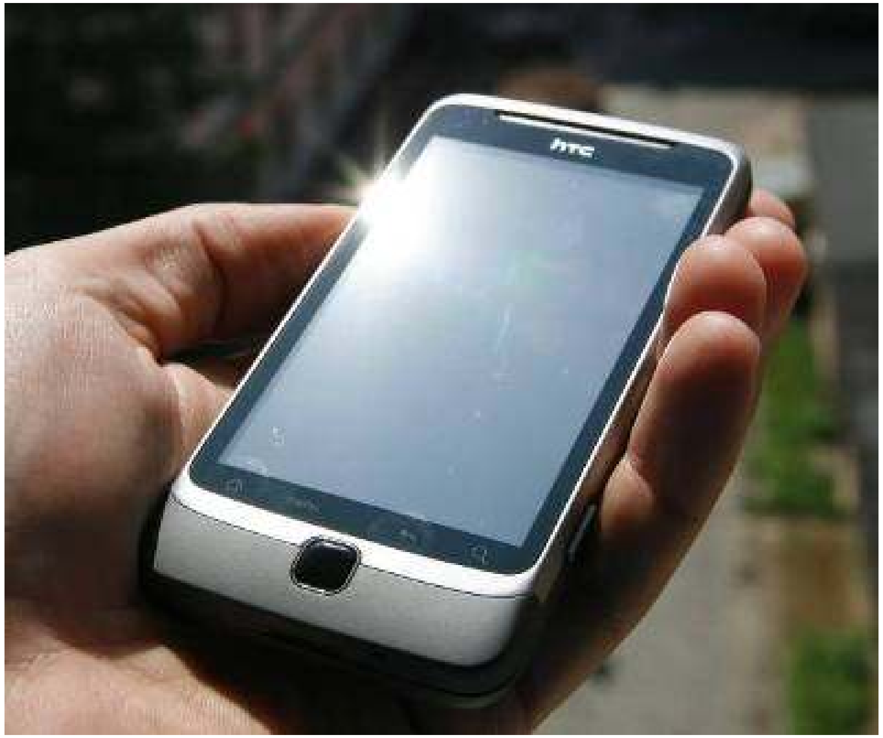


Simulation

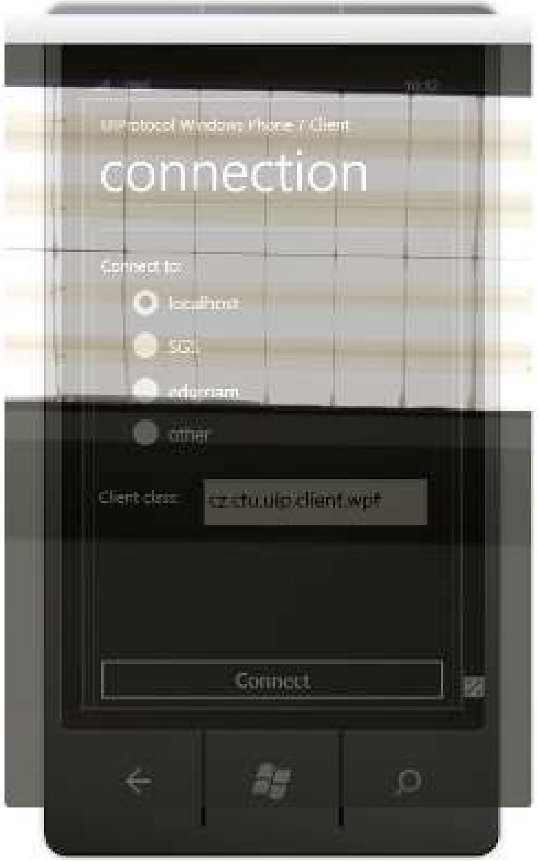

Carrier
2:33 PM
Cancel
Add Bookmark
Save
rbilder - Neue Luzerner Zeitung Online
http://www.luzernerzeitung.ch/nachrichten/...
Bookmarks
Q W E U I O P
A S D K L
Z X
.?123
sp

Reflection on the display

Real world



Simulation



UIProtocol Windows Phone 7 Client

connection

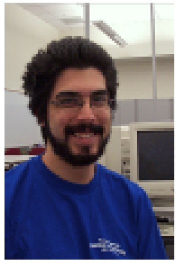
Connect to:

- ● localhost
- ○ SGL
- ○ eduroam
- ○ other

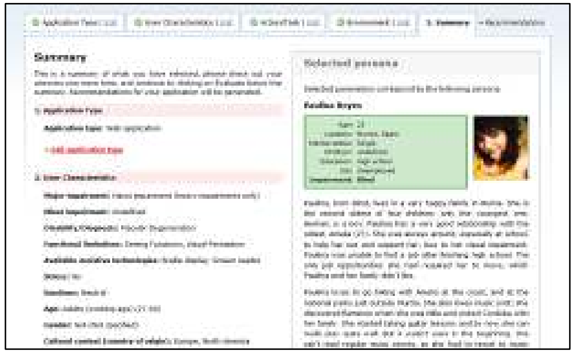
Client class: cz.cvut.uip.client.wpf

Development tools supporting accessibility

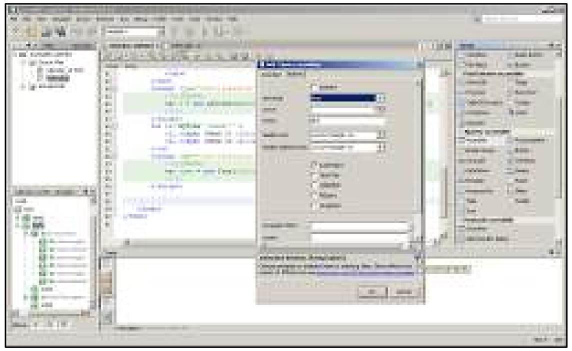
Development process with a11y support



Developer



Accessibility Advisor



ARIA developer

Final applications

Development tools supporting a11y

- AEGIS Accessibility Advisor
  - https://cent.felk.cvut.cz/hci/aegis/public/accessibility-advisor/ (experimental version)
- AEGIS Developer tool
  - https://cent.felk.cvut.cz/hci/aegis/?page=aadt
- SUCESS ARIA Checker
  - https://cent.felk.cvut.cz/hci/coe/?page=aria_checker

Thank you

Zdeněk Míkovec

Czech Technical University in Prague

xmikovec@fel.cvut.cz