An accessible, adaptive and multimodal digital TV framework and corresponding development tool

Pradipta Biswas, Pat Langdon
Engineering Design Centre
Department of Engineering
University of Cambridge,
UK

Carlos Duarte, Jose Coelho,Tiago Guerreiro
Departamento de Informática
Faculdade de Ciências University of Lisbon, Portugal

Christoph Jung
Fraunhofer-Institut für Graphische Datenverarbeitung IGD,
Germany

***Abstract*—This paper presents a tool and a framework to develop accessible and adaptable digital TV interfaces for disabled and elderly users. The development methodology involves disabled and elderly users early in the design process and optimizes interfaces using a simulation system. The simulator complements existing user centred design processes and helps designers to understand, visualize and measure effect of impairments on interaction. The adaptive framework supports a wide variety of applications through its easy-to-use APIs. The system is validated through a series of user trials confirming its usefulness for users with different range of abilities.**

***Keywords*—Human Computer Interaction; User Model; Multimodal Adaptation**

I. Introduction

Recent research on interactive electronic systems can improve the quality of life of many disabled and elderly people by helping them to engage more fully to the world. In particular, during the past few years digital TV has turned from a simple receiver and presenter of broadcast signals to an interactive and personalised media terminal, with access to traditional broadcast as well as internet-based services. Currently available TV panels offer integrated digital processing platforms, with access to standardised hybrid WebTV (or Hybrid TV) portals (e.g. HbbTV [9]). These portals do not only offer access to the internet and legacy web services (like web browser or proprietary portal views on YouTube, Flickr, Facebook, etc.), but also specify content services that are immediately coupled to broadcast content. At the same time it is recognised that disabled or elderly people still face problems when using the above mentioned services. Approximately half of the elderly people over 55 suffer from some kind of functional limitations or impairments (vision, hearing, motor and/or cognitive [7]). For them interaction, especially with digital TV or other consumer electronics devices is sometimes challenging, although accessible ICT applications could make a difference for their living quality. They have the potential to enable or simplify participation and inclusion in their surrounding private and professional communities.

The early attempts of designing systems for people with disabilities was confined to developing isolated system like blind access via Optacon, special video card for low vision access or switch access software for motor impaired users. From late 90s, researchers started to take a more holistic approach like developing Accessibility APIs like Microsoft Accessibility API and standardizing guidelines like Web Content Accessibility Guidelines (WCAG). However services and products for people for disabilities still lags behind mainstream systems. The diverse range of abilities complicates the designing of interfaces for these users. Many inclusive or assistive systems often address a specific class of users and still exclude many users. Lack of knowledge about the problems of disabled and elderly users has often led designers to develop non inclusive systems. As a result, the availability of accessible user interfaces being capable to adapt to the specific needs and requirements of users with individual impairments is very limited. Although there are numerous APIs available for various operating systems or application platforms in web browsers that allow developers to provide accessibility features within their applications, today none of them offers features for automatic adaptation of multimodal interfaces, being capable to automatically fit to the individual requirements of users with different kinds of impairments. Moreover, the provision of accessible user interfaces is still expensive and risky for application developers, as they need special experience and effort for user tests. Many implementations simply neglect the needs of elderly people locking out a large portion of their potential users.

The European project *GUIDE* [13] aims to fill the accessibility, expertise, time, budget and framework gap mentioned above. This is realised through a comprehensive approach for the development and dissemination of multimodal user interfaces capable to intelligently adapt to the individual needs of users with different kinds of physical and age-related impairments. As application platform, *GUIDE* targets connected TVs and Set-Top Boxes (STBs), including emerging application platforms such as HbbTV, and also proprietary STB middleware solutions that integrate broadcast and broadband services. These platforms have the potential to address the special needs of elderly users with applications such as for home automation, communication or continuing education. This paper focuses on the utility of the GUIDE system in developing inclusive applications. It presents

- A simulation system used by GUIDE application developers. The simulation system helps designers to visualize, understand and measure effect on impairment on their designs. It can evaluate designs of interface layouts for different levels of impairment and can be used on paper-pencil prototypes to actual operational systems.

- A framework that sits between end-users and applications, empowering applications with adaptation and multimodal interaction capabilities. This is achieved through a user modeling mechanism that coupled with adaptive fusion and fission mechanisms are capable of (1) perceiving what the user can interact with from an application's interface description; (2) interpreting the user's multimodal commands; (3) transform the user command into an application understandable format; while (4) adapting the application's interface to the user and context.

The following sections present the simulation tool, adaptation system and the framework respectively. The adaptation capabilities of the system are validated through a series of user trials described in section 6 followed by conclusions in section 7.

II. Design Improvement through Simulator

We have used a simulator [4] to improve interface designs of GUIDE applications. The simulator takes a task definition and locations of different objects in an interface as input and then predicts possible eye movements and cursor paths on the screen and uses these to predict task completion times with respect to different user profiles. The simulator embodies both the internal state of an application and also the perceptual, cognitive and motor processes of its user. Figure 1 shows the architecture of the simulator.

- **The Environment model** contains a representation of an application and context of use. It consists of:
- **The Application model** containing a representation of interface layout and application states.
- **The Task model** representing the current task undertaken by a user that will be simulated by breaking it up into a set of simple atomic tasks following the KLM model [5].
- **The Context model** representing the context of use like background noise, illumination and so on.

The Device model decides the type of input and output devices to be used by a particular user and sets parameters for an interface.

The User model simulates the interaction patterns of users for undertaking a task analysed by the task model under the configuration set by the interface model. It consists of a Perception model, a Cognitive model and a Motor Behaviour Model.

The details about users are stored in XML following standardized format developed in EU VUMS cluster [14]. The visual perception model simulates the phenomenon of visual perception (like focusing and shifting attention) by investigating eye gaze patterns (using a Tobii X120 eye tracker) of people with and without visual impairment. The model uses a backpropagation neural network to predict eye gaze fixation points and can also simulate the effects of different visual impairments (like Maccular Degeneration, colour blindness, Diabetic Retinopathy and so on) using image processing algorithms.

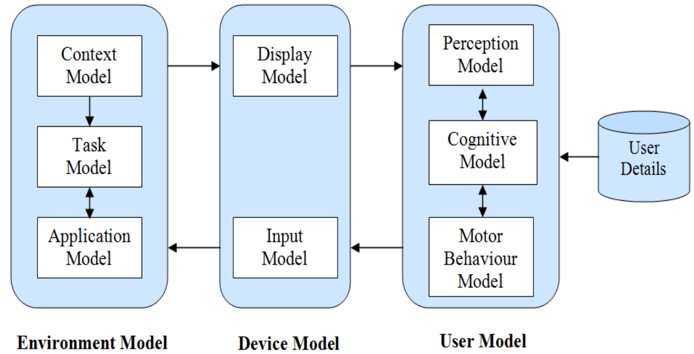


Fig. 1. Architecture of the simulator

The auditory perception model simulates effect of both conductive (outer ear problem) and sensorineural (inner ear problem) hearing impairment. The models are developed using frequency smearing algorithm [12] and are calibrated through audiogram tests.

The cognitive model uses CPM-GOMS model [11] to simulate expert performance. It has a novel and easy-to-use module to simulate performance of novices using two interacting Markov processes.

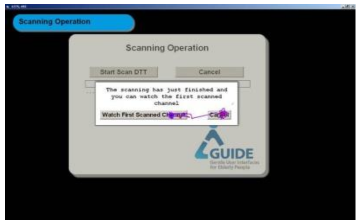
The application of Fitts' law [8] for people with motor impairment is debatable as the assumptions behind Fitts' law are often violated by movement patterns of motor-impaired users. So the motor behaviour model is developed by statistical analysis of cursor traces from motor impaired users and measuring their hand strength using a Baseline 7-pc Hand Evaluation Kit. Based on the analysis, a regression model has been developed to predict pointing time.

The models are calibrated and validated through extensive user studies covering more than 50 users affected by different extents of visual, hearing and motor impairment [4]. The actual and predicted eye gaze patterns, sub-movement profiles in cursor trajectory and task completion times are compared and they are correlated with statistical significance ($p < 0.05$).

The simulator simulates performance of users in a more detailed level than GOMS models, but easier to use than the cognitive architectures as it does not need detailed knowledge of psychology or programming to operate. It has graphical user interfaces to provide input parameters and showing output of simulation. The simulator has already been used to develop a few assistive interaction systems [2]. A demonstration copy of the simulator is available for downloading at the publication section of *GUIDE* website [13]. Interface designers have used the simulator for improving their designs. Figure 2a and b demonstrate such an example. In figure 2a, the font size was smaller and the buttons were close enough to be missed clicked by a person with tremor in hand. The designer chose the appropriate font type (Tiresias in this case) and size and also the inter-button spacing through simulation.

As Figure 2b shows, the new interface remain legible even to moderate visually impaired users, the inter-button spacing is large enough to avoid missed-clicking by moderate motor impaired users. In figure 2b the purple lines show simulated cursor trajectories of users with tremor in hand.

a. Initial interface



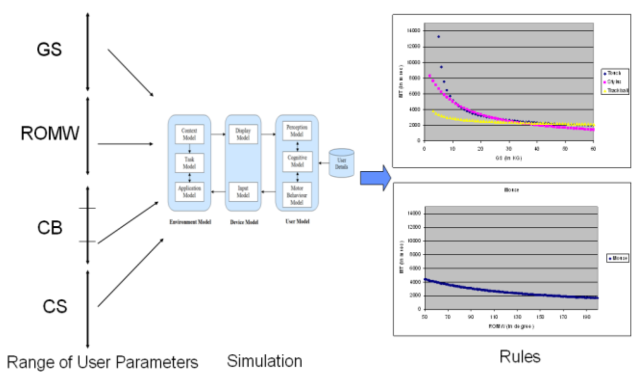
b. Changed Interface with simulation of medium visual and motor impaired profile

Fig. 2. Correcting interface layout

III. Runtime Adaptation

The simulator can predict how a person with visual acuity *v* and contrast sensitivity *s* will perceive an interface or a person with grip strength *g* and range of motion of wrist *w* will use a pointing device. We ran the simulator in Monte Carlo simulation and developed a set of rules relating users' range of abilities with interface parameters (Figure 3). For example the following graph (figure 4) plots the grip strength in kilograms (kg) with movement time averaged over a large range of standard target widths and distances in an electronic screen for three different input devices. The curve clearly shows an increase in movement time while grip strength falls below 10 kg and the movement time turns independent of grip strength while it is more than 25 kg. Similar analyses have been done on fontsize selection with respect to visual acuity and colour selection with respect to different types of dichromatic colour blindness. Taking all the rules together, three sets of parameters can be predicted:

1) *User Interface(UI) parameters*
2) *Adaptation code*
3) *Modality preference*



GS: Grip Strength
ROMW: Active Range of Motion of Wrist
CB: Type of Colour Blindness
CS: Contrast Sensitivity

Fig. 3. Developing runtime user model

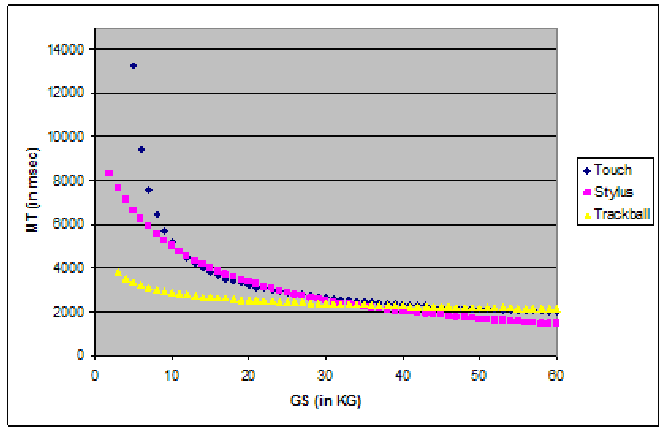


Fig. 4. Relating Movement Time with Grip Strength

In the following sections we briefly describe these prediction mechanisms.

A. User Interface parameter prediction

Initially we selected a set of variables to define a web based interface. These parameters include:

- Button spacing: minimum distance to be kept between two buttons to avoid missed selection
- Button Colour: The foreground and background colour of a button
- Button Size: The size of a button
- Text Size: Font size for any text rendered in the interface

The user model predicts minimum button spacing required from the users' motor capabilities and screen size. The simulation predicts that users having less than 10 kg of grip strength or 80º of Active Range of motion of wrist or significant tremor in hand produce a lot of random movement while they try to stop pointer movement and making a selection in an interface. The area of this random movement is also calculated from the simulator. Based on this result, we calculated the radius of the region of the random movement and the minimum button spacing is predicted in such a way so that this random movement does not produce a wrong target selection. The exact formula is as follows:

```
If users have Tremor, less than 10 kg of Grip
strength or 80° of ROM in wrist

    Minimum button spacing = 0.2 *distance of
target from centre of screen

If users have less than 25 kg of Grip strength

    Minimum button spacing = 0.15 *distance of
target from centre of screen

else

    Minimum button spacing = 0.05 * length of
diagonal of the screen
```

If a user has colour blindness it recommends foreground and background colour blindness as follows:

```
If the colour blindness is Protanopia or Deuteranopia (Red-Green) it recommends

    White foreground colour in Blue background

For any other type of colour blindness it recommends

    White foreground in Black background or vice versa
```

The system stores the minimum visual angle based on the device type, screen size and distance of user from the screen and use it to predict minimum font size for different devices in pixel or point.

B. Adaptation code prediction

The adaptation code presently has only two values. It aims to help users while they use a pointer to interact with the screen like motion sensors or gyroscopic remote. The prediction works in the following way

```
If a user has tremor in hand or less than 10 Kg Grip Strength

    The predicted adaptation will be Gravity Well and Exponential Average

Else

    The predicted adaptation will be Damping and Exponential Average
```

In the first case, the adaptation will remove jitters in movement through exponential average and then attract the pointer towards a target when it is near by using the gravity well mechanism. Details about the gravity well algorithm can be found in a different paper [3, 10]. If the user does not have any mobility impairment, the adaptation will only work to remove minor jitters in movement.

C. Modality prediction

The modality prediction system predicts the best modality of interaction for users. The algorithm works in the following way:

```
If User has Maccular Degeneration or User is Blind
        BestIP = "Voice"
        If DeviceType = TV"
        BestOP = "AudioCaption"
        Else
        BestOP = "ScreenReader"
        End If

ElseIf GRIP STRENGTH < 10Kg Or STATIC TREMOR > 499 Then 'Severe Motor Impairment with vision
        Select Case DeviceType
            Case 'Mobile'
            BestIP = "BigButton"
            Case 'Laptop'
            BestIP = "TrackBall or Scanning"
            Case 'Tablet'
            BestIP = "Stylus"
            Case 'PC'
            BestIP = "TrackBall or Scanning"
            Case 'TV'
            BestIP = "SecondScreenBigButton"
        End Select

        BestOP = "Screen"

ElseIf GRIP STRENGTH < 20Kg Or STATIC TREMOR > 299 Then 'Moderate Motor Impairment with vision
        Select Case DeviceType
            Case 'Mobile'
            BestIP = "BigButton"
            Case 'Laptop'
            BestIP = "TrackBall or Mouse"
            Case 'Tablet'
            BestIP = "Stylus"
            Case 'PC'
            BestIP = "TrackBall or Mouse"
            Case 'TV'
            BestIP = "SecondScreenBigButton"
        End Select

        BestOP = "Screen"

ElseIf ACTIVE RANGE OF MOTION OF WRIST < 100°

        Then

        Select Case DeviceType
            Case 'Mobile'
            BestIP = "Stylus or BigButton"
            Case 'Laptop'
            BestIP = "Trackball or Mouse"
            Case 'Tablet'
            BestIP = "Stylus"
            Case 'PC'
            BestIP = "Trackball or Mouse"
            Case 'TV'
            BestIP = "BasicRemote"
        End Select

        BestOP = "Screen"

Else 'User without visual or motor impairment

        BestIP = "DirectManipulation"
        BestOP = "Screen"
        End If
```

IV. Conceptual Framework

In this section we describe the exploitation of the user model in a run-time software framework (Figure 5).

When a user starts interacting with the system, we use a user initialization application to create a profile for the user. The initialization is a sequence of multi-modal interactive tests, coupled with a basic tutorial on how to use the system. In the individual tests, we do not need an accurate measurement of functional ability; rather an approximate estimation is good enough to fire the user modeling rules. For example we can use the age, gender, height and an assessment on presence of any spasm or tremor in hand of a person to interpolate his objective hand strength data [1] to invoke appropriate adaptation features for him. Additionally, the user can also override the prediction of the system by giving explicit preference about any interface and the user model stores this preference for future use.

After the initialization application, the user can provide input through multiple devices like motion sensors and speech recognizers, meaning he can use multiple modalities like pointing, gesture and speech simultaneously. The signals from recognition based modalities are processed by interpreter modules like a series of points from the motion sensor go through a gesture recognition engine in order to detect gestures. Signals corresponding to pointing modalities go through input adaptation modules. Both interpreter and adaptation modules base their decisions on knowledge stored in the *GUIDE* profiles achieving noise reduction in the input signals or invoke gravity well algorithm.

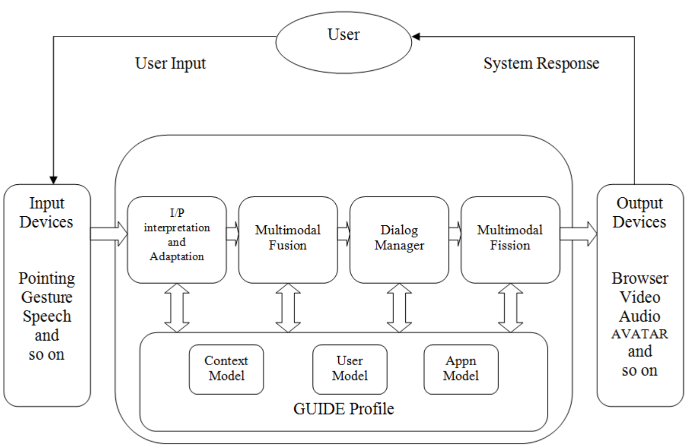


Fig. 5. *GUIDE* Framework

The multimodal fusion module analyzes the raw input signals and the outputs of input interpreters and input adaptation and combines these multiple streams into a single interpretation based on the user, context and application models. The interpretation resulting from the input signals are sent to the dialog manager which couples the framework with applications and decides the application's response.

Finally, this response of the application is fed to the multimodal fission module, which again takes help from the user, context and application models and prepares the output appropriately (like embedding a HTML page in a video with subtitle and voice output) to be rendered in the output devices. The user perceives this output and provides further input.

V. Integration to Other Applications

The proposed framework concept can in principle support any known application environment/runtime (such as native C/C++ runtimes, JAVA, Android, etc.). As a first proof of concept we decided to rely on proven and widely known Web technology, as it is represented by HTML(5) and Web browsers. Applications in this environment are represented in terms of HTML pages, with embedded JavaScript, CSS as well as media objects, like images, videos, etc.

There are several requirements to be considered when creating a UI management layer for an existing application environment. At first, a UI framework should limit interventions with the existing development tools and processes. It should further support the simple integration of legacy applications. An application can be usually considered an independently acting entity, reacting to user input as well as internal or external events. Consequently, a UI framework must support synchronisation with application processes and user I/O. Of course, the UI framework should provide adaptation and UI management services in a transparent manner, so that the developer/application does not need to have any knowledge about UI configurations or UI-related user properties.

Considering the previous requirements, we developed the Web Browser Interface (WBI), which is the basic component in our framework that abstracts the application to the framework and vice versa. The WBI ensures that all UI-related information that is exchanged with the framework is being mapped to the concrete HTML/JS representation in the browser. The WBI can receive events from the framework (like user input, required GUI adaptations, cursor positions, etc.) and forward data from the application to the framework (current UI representation, submission of new user profile data, etc.).

The application developer can access the WBI services through a JavaScript API, which must be embedded as a file in the application's HTML page. In order to fulfill the above mentioned synchronisation requirement, the application has to follow a specific protocol (Figure 6). Whenever the application has finished internal state transitions ("Application phase") and requires new user input, it calls the framework. A sub component of the WBI now queries the HTML DOM for annotated elements (WAI-ARIA) and generates a UIML representation from the elements. Now the framework core starts various adaptation processes and concurrently recognizes multi-modal user input ("Framework phase"). In this phase the WBI might receive instructions from the core to modify the GUI, e.g. by manipulating elements in the HTML DOM (e.g. increase font size for a vision-impaired user). Once the Framework has recognized relevant user input (which maps to available application input slots), the core sends this input to the WBI, which in turn maps the input e.g. to a click event that is emitted on the corresponding HTML element. A user can for example select an item on screen using voice, and thereby click the element. It should be noted that this process is absolutely transparent for the application developer.

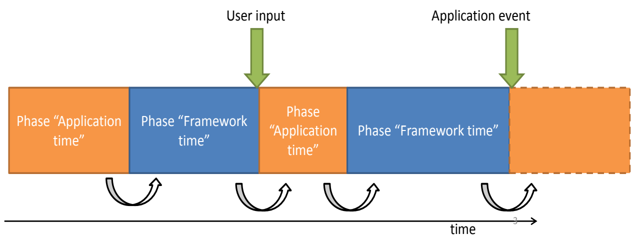


Fig. 6. Interaction of application and framework

The WBI can be employed in two variants, depending on the underlying platform restrictions. When being used with standard browsers (like Mozilla Firefox, Opera, Chrome, etc.) the WBI can be deployed as a shared library plugin in the browser (NPAPI interface). Where this is not possible, the WBI can run as a stand-alone native application, and communicate with the JS API via WebSockets. The basic model of the GUIDE Framework allows the application to always remain in control, regarding internal state changes and updates of the user interfaces. Nevertheless, it is required to synchronise application logic and Framework processes, to avoid interference. In order to not collide with (transparent) Framework processes, the application must call a function before it changes its UI, and another function after it has finished a transition:

```
MyApplication.eventHandlerForUserInputOrA
      nInternalEvent = function() {
           GUIDE.endGetUserInput();
      // ... do something useful, change
         user interface, etc. ...
          GUIDE.beginGetUserInput();
                     }
```

Before using the GUIDE JavaScript API, it has to be initialised by calling the init() function. On the host TV platform, this function embeds the WBI browser plugin in the page and initialises everything.

On a second screen device (like smartphone or tablet), it establishes a network connection to the host instance. Optionally, one can also register event handlers in the GUIDE API, or configure the API by setting available parameters. Finally to use the GUIDE Framework JavaScript API one has to embed the JavaScript file into the application:

```
<html>
<head>
<script type="text/javascript"
src="../../JavaScriptAPI/GuideJavaScriptA
PI.js"></script>
 </head>
```

Figure 7 below shows effects of adaptation on a Smart Home Application for different profiles. It shows different colour contrasts, font sizes and button sizes used for different users.

A few more applications for this DTV based system can be found at http://www-edc.eng.cam.ac.uk/~pb400/GUIDE_DemonstrationVideo.mp4 (110 Mb).

VI. VALIDATION

We have validated the adaptation system of the framework in two stages. The internal validation considered a representative pointing and clicking task and conducted over twelve participants in controlled laboratory settings. It validates the recommendations from the user model. The external validation is performed through an Electronic Program Guide (EPG) application implemented through the GUIDE framework. The following sections presents detail of these studies.

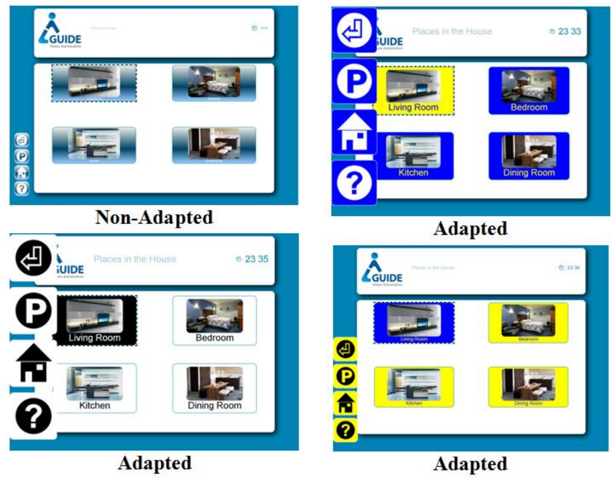


Fig. 7. daptation in GUIDE

A. *Internal Validation*

The internal validation validates the rules of the user model through a simple point and click task. The task is kept simple to ensure the statistical effect we observed in the trial is only due to the experimental conditions and not due to difficulty in learning the task.

B. *Participants*

We collected data from the following twelve users (average age: 56.92 years, male to female ratio 7:5) with physical or age related impairment (Table 1). The selection criteria of participants was either more than 60 years old or having physical impairment. These users were recruited through a local user organization in UK, they all use computers or laptops everyday and volunteered for the study.

C. *Design*

The study simulates a situation of pointing and clicking in a direct manipulation interface. For example, users often click an icon on desktop to open a folder and then click another time to select the required file. This study first showed users a couple of familiar icons and then asked them to click on these two icons from a list of icons. The list of icons was presented in three different ways. In one case they use the default parameter settings (font size, button spacing) of Windows 7 operating system. In the other two cases the layout was adapted following predictions from the user model.We considered two different organizations of icons in the adapted versions – elliptical and rectangular. Figure 8 below shows examples of the control and adapted versions of the icon searching screens.

TABLE I. PARTICIPANTS

Participants	Age	Sex	Impairment
P1	44	M	Tunnel Vision, Spasm in finger
P2	48	M	CerebralPalsy
P3	57	F	CerebralPalsy
P4	34	M	Polio
P5	45	M	Spina Bifida
P6	48	F	Spina Bifida
P7	73	M	Glaucoma, age related dementia
P8	60	M	Age related visual impairment
P9	69	F	Age related visual impairment
P10	65	F	Age related visual impairment
P11	63	M	Age related visual impairment
P12	77	F	Blurred vision due to droops in lower eye-lid, protanomalous colour blindness

The elliptical arrangement requires more visual search time but it has less probability of accidental clicking on wrong icons by motor impaired users as the pointing path does not contain multiple icons. The rectangular arrangement is more familiar than the elliptical one. In both adapted versions, we changed the font size and button spacing according to the algorithms discussed above.

The button labels are also presented in higher contrast for colour blind users. The gravity well and exponential averaging algorithms [3] were activated according to the Adaptation Selection algorithm in previous section. Users also followed predictions from the user model in choosing the appropriate input devices in adapted conditions.

D. *Material*

The study was conducted using a computer and a Tablet device. Both of these devices had Windows 7 operating system. The computer has a 20” screen with 1280 × 1024 pixel resolution while the Tablet had a 10” screen with 1280 × 800 pixel resolution. The participants used a standard mouse and their fingers with the tablet touchscreen in control condition, while they were allowed to use a TrackBall and Stylus in experimental condition based on the prediction from the user model.

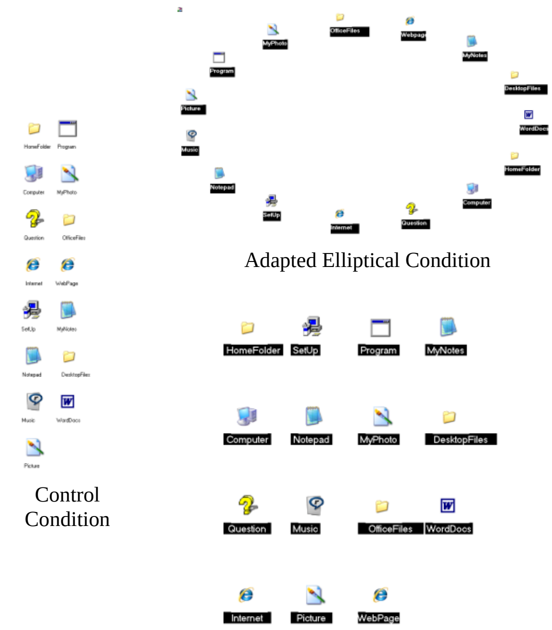


Fig. 8. Icon searching screens

E. *Procedure*

Initially the participants used part of the UIA to create a user profile. Then they undertook the icon searching task. The control (non-adapted) and experimental (adapted) conditions were randomly chosen. For each screen, participants needed to remember two icons and click on them. Each participant used both computer and Tablet. They undertook 10 icon searching tasks under each condition for each device. We measured the time interval between presentations of the screen of icons and the event of clicking on an icon.

F. *Results*

Figures 9 and 10 show average pointing times and number of correct selections for all different conditions. The Y bars signify standard deviation. Initially we found that 9 out of 12 users selected more correct icons and took less time to point in one of the adapted conditions than the control conditions for both PC and Tablet, while they selected the first icon. During selection of the second icon, 10 out of 12 users in PC and 12 out of 12 users for Tablet selected more correct icons and took less time to point in one of the adapted conditions than the control conditions.

We have further analyzed the pointing times and number of correct icon selection through ANOVA and MANOVA tests. The structure of these tests were as follows

$$Device \times Condition \times Selection$$

- Device has two levels: PC and Tablet
- Condition has four levels: Control, Adapted Elliptical, Adapted Rectangular and AdaptedMerged. The last condition aims to merge the different adaptation conditions into one. Users preferred and performed better in one of the elliptical or rectangular conditions, the last condition (AdaptedMerged) considers the better performance (less pointing time and more correct selection) between elliptical and rectangular conditions.
- Selection has two levels: First selection and Second selection, as users needed to select two icons each time.

We have the following significant effects in the Within-Subject Test

1) As expected Selection has a significant effect for both correct icon selection and pointing time. Users selected less number of correct icons and took more time to click the second icon than the first icon.

2) A main effect of Condition for correct icon selection, $F(1.84, 20.24) = 3.74$, $p< 0.05$, $\eta^2 = 0.25$ *after applying Greenhouse-Geisser correction. The effect of condition for pointing time tends to significance,* $F(3, 11)= 2.83$, $p = 0.05$, $\eta^2 = 0.20$.

3) A main effect of Device for correct icon selection, $F(1,11) = 5.21$, $p< 0.05$, $\eta^2 = 0.32$

4) An interaction effect of Device × Selection for correct icon selection, $F(1,11) = 5.52$, $p < 0.05$, $\eta^2 = 0.33$

In the MANOVA test, we have the following significant effects

1) A significant effect of selection for both reaction and correct icon selection as in the within subject test.

2) A main effect of Device for correct icon selection, $F(1,11) = 5.21$, $p< 0.05$, $\eta^2 = 0.32$

3) An interaction effect of Device × Selection for correct icon selection, $F(1,11) = 5.52$, $p < 0.05$, $\eta^2 = 0.33$

4) The effect of Condition tends to significance for pointing time, $F(3, 9) = 3.44$, $p = 0.06$, $\eta^2 = 0.53$

In summary,

1) The control condition packed icons in a small portion of the screen due to reduced font size and button spacing. It may be expected that this condition would require less eye gaze and pointer movements than adapted versions, which spread out the icons throughout the screen. However, users selected more correct icons in one of the adapted conditions than control condition with statistical significance and their pointing times were not compromised due to spreading up the icons in the screen as the differences in pointing times with control condition were not significant, rather on average it was less in AdaptedMerged condition than Control condition. It shows that the increased font size, button spacing and colour contrast helped users to remember, search, point and click on them.

2) Users took more time to remember the second icon, which can be attributed to the fact that many of our participants have age or cerebral palsy related dementia.

3) Users selected more correct icons in the PC than in tablet, especially during selecting the second icon. This is due to the fact that many of our participants had more experience with PC than with Tablet device. However their pointing times to select correct icons were not significantly different between PC and Tablet.

4) Regarding subjective preference, 6 users preferred Elliptical arrangements of buttons while 5 preferred rectangular and one had no preference. All of them preferred

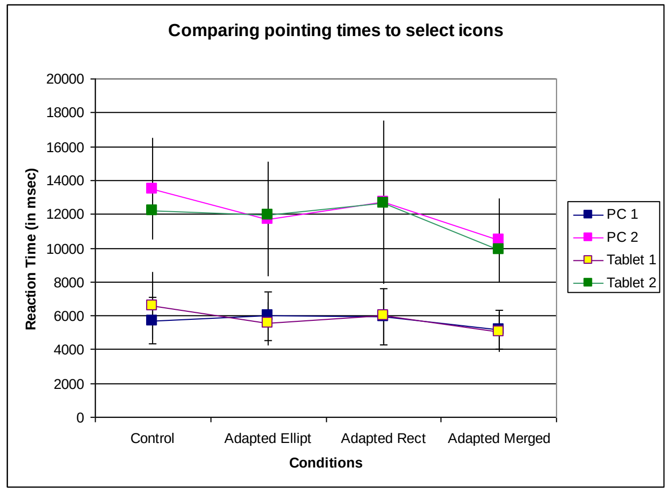


Fig. 9. Comparing average pointing times between control and adapted conditions

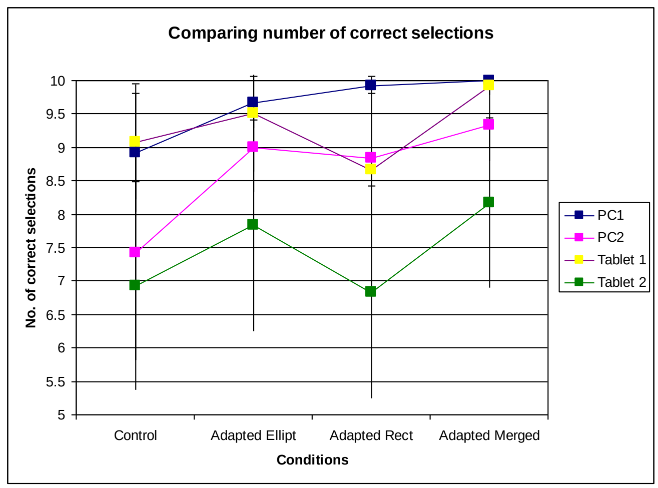


Fig. 10. Comparing number of correct icon selections between control and adapted conditions

one of the adapted conditions more usable than the control condition.

G. External Validation

This study has used the user model implementation within the GUIDE Framework in a setting mimicking users are watching TV at home.

We conducted a study [6] to evaluate the user model's ability to assess users' profiles and adapt interfaces accordingly. Users interacted with the system using TV remote, gesture recognition and second screen based systems, the appropriate modality was chosen by the user model. Herein, we briefly summarize the results obtained in three countries, Germany, Spain and UK, with a total of 40 elderly people, with different age-related disabilities. In this study, all participants created user profiles and used adapted and non-adapted versions of an Electronic Program Guide (EPG) application. The average age of participants was 70.9 years old. To understand the benefits of the user model and the adaptations performed, we analysed the subjective understanding and acceptance of the created profiles and consequent interface changes. Results [6] showed that participants perceived the adaptation during the adapted EPG tasks. It was also found that those were subject to adaptations rated the adaptive version as an improvement over the non-adapted one. The baseline EPG already had improved accessibility features over traditional EPGs due to the use of the simulation tool, a fact that may have reduced the impact of the adaptations in such a short term evaluation. The participants showed to be positive about the adaptations, which is relevant as a requirement for adoption, particularly in the elderly population.

Later, we conducted a task-by-task video analysis of 15 users sampled from all users (6 Spanish users, 6 British users and 3 German users). We first constructed a list of the necessary variables to look for while watching each user interacting with both the user profile creation application and different adapted and non-adapted versions of the EPG. Following that selection, the 15 users more relevant and which cover all user model features were selected. Finally the 15 videos were watched once (several were watched and then revised to make sure all variables were classified in the same manner for every user) and the list of variables was filled accordingly.

The detailed analysis performed with 15 users with different profiles showed that there was a clear distinction between the adapted and non-adapted EPG versions. A higher percentage of the participants (94%) perceived the interface elements and adaptations in the Adapted version without any intervention from the evaluation monitors. In the Non-adapted version, unaided perception of the interface elements was lower (77%). The main reasons for this difference were the visual adaptation mechanisms in the adapted version that helped users in perceiving the interface elements.

During execution of tasks, the adaptations showed to improve the participants' autonomy and overall performance. This was visible in the amount of times they stopped during a trial without being able to continue on their own (11% vs 16%) but also in the percentage of tasks that were accomplished without requiring any help from the test monitor (49% vs 61%). The number of explicit help requests also showed to be higher in the non-adapted version (0.33 times per task) than in the adapted version (0.16 times per task) revealing that the adaptations ease the usage of the EPG and make the user more comfortable. The acceptance ratings of the participants towards the adaptations showed that almost half of the users were satisfied with the adaptations performed (7 participants – 47%) This could seem as a low value but looking in detail only 2 participants (13%) disagreed with the adaptation. The remaining 6 participants were mildly satisfied with the adaptation as they wished it to be more evident (even bigger fonts and buttons and more contrast).

In summary, in an adapted version, supported with an automatically enriched user-model, the users are more effective both in understanding and completing tasks, performing fewer errors, and requiring less help. These results, together with the positive acceptance of the adaptation concepts and their expected impact in the quality of life of its users, validate the approach followed so far and pave the road for the project's future developments, which will be verified in a longitudinal trial for better assessing the effects of adaptation based on our user model.

VII. Conclusions

This paper presents a new method of developing electronic interfaces for elderly users and people with disabilities. Our approach involves simulating interaction patterns for users with a wide range of abilities and then using the prediction to make decisions about design. This enables designers to evaluate the effect of physical impairment on their design in early design phase and customize their products according to different user groups. As they do not need to go for time consuming and expensive user trials at early design stage for all combinations of design alternatives and user categories, it reduces the cost and time to develop inclusive systems. The simulation is used in developing the *GUIDE* framework.

The *GUIDE* Software Framework is designed to integrate various kinds of multi-modal user interface technologies (like gesture control, automatic speech recognition, remote controls, graphical user interfaces, virtual characters, second screen devices, assistive technologies, etc.) and adapt them automatically to the preferences and capabilities of individual users.

Adaptation is based on the user profiles and can select appropriate I/O modalities and combinations, and appropriately configure them for a specific user profile. Further support for the user during interaction is provided by input adaptation and processing of contextual data. Our user trials showed that the development methodology can help designers in developing accessible system and our adaptive system can indeed makes end users more effective both in understanding and completing tasks, performing fewer errors, and requiring less external help.

References

[1] Angst F. et al., Prediction of grip and key pinch strength in 978 healthy subjects, BMC Musculoskeletal Disorders 2010, 11:94

[2] Biswas P. and Langdon P., A new input system for disabled users involving eye gaze tracker and scanning interface, Journal of Assistive Technologies, Vol. 5 Issue 2 June 2011, ISSN: 1754-9450

[3] Biswas P. and Langdon P., Investigating the gravity well algorithm to help in pointing in electronic interfaces by motor-impaired users, Journal of Assistive Technologies, 5(3) September 2011, ISSN: 1754-9450

[4] Biswas P., Robinson P. and Langdon P., Designing inclusive interfaces through user modeling and simulation, International Journal of Human-Computer Interaction, Taylor & Francis, Print ISSN: 1044-7318, DOI: 10.1080/10447318.2011.565718

[5] Card S., Moran T. and Newell A. The Psychology of Human-Computer Interaction. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1983.

[6] Coelho J., Duarte C., Biswas P. and Langdon P., Developing Accessible TV Applications, Proceedings of the ACM SIGACCESS Conference in Accessibility (ASSETS) 2011

[7] Eurostat Statistical books Active ageing and solidarity between generations. A statistical portrait of the European Union 2012

[8] Fitts P.M. "The Information Capacity of The Human Motor System In Controlling The Amplitude of Movement." Journal of Experimental Psychology 47 (1954): 381-391.

[9] HbbTV, Available at: http://www.hbbtv.org, Accessed on 10th January, 2011

[10] Hwang F. et. al. Cursor Characteristics And Haptic Interfaces For Motor Impaired Users. Cambridge Workshop on Universal Access and Assistive Technology 2002. 87-96

[11] John B. E. and Kieras D. "The GOMS Family of User Interface Analysis Techniques: Comparison And Contrast." ACM Transactions on Computer Human Interaction 3 (1996): 320-351.

[12] Nejime Y. and Moore B.C.J., Simulation of the effect of threshold elevation and loudness recruitment combined with reduced frequency selectivity on the intelligibility of speech in noise, J. Acoust. Soc. Am.(1997) 102: 603-615

[13] The *GUIDE* Project, URL: http://www. **anonymized**.eu/, Accessed on 23rd August 2010.

[14] EU VUMS Cluster, Availabe at: http://www.veritas-project.eu/vums/?page_id=56, Accessed on: 12/12/2012