

# ITU-T Technical Report

(11/2023)

## TR.CUP

---

**Concept of a common user profile format used  
to personalize audiovisual media**





# Technical Report ITU-T TR.CUP

## Concept of a common user profile format used to personalize audiovisual media

### Summary

A common user profile (CUP) format used to personalize audiovisual media is intended to assist users who have different ranges of abilities to make audiovisual content more accessible. The CUP is targeted at broadband, digital TV, computer and smartphone software and web-based audiovisual systems. The CUP is independent of the device or application being used and is intended through a software agent<sup>1</sup> stored in the user's device or devices, to personalize the media experience by adapting the devices interface parameters such as the displayed captions font size, colour and contrast, the audio dialogue balance, volume and equalization, the size and position of on-screen elements, etc.

The CUP application can also be used to create synthetic profiles which can be used to simulate the effect of parameter adjustment on different devices allowing developers to assess features on different persons with different ranges of abilities.

### Keywords

Accessibility, audiovisual, common user profile format, metaverse, personalization, user model.

### Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

### Change Log

This document contains the ITU-T Technical Report TR.CUP on “Common User Profile (CUP) format used to personalize audiovisual media” approved as the new work item at the ITU-T Study Group 9 meeting held in Bogota, 14-23 Nov. 2023.

<b>Editor:</b>	Pradipta Biswas Indian Institute of Science India	Tel: +91-80-22932625 E-mail: <a href="mailto:pradipta@iisc.ac.in">pradipta@iisc.ac.in</a>
	Avinash Agarwal Ministry of Communications India	Tel: +91-9868171222 Email: <a href="mailto:avinash.70@gov.in">avinash.70@gov.in</a>

© ITU 2024

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

---

<sup>1</sup> A software agent is defined as a digital entity that perceives its environment and takes actions that maximize its chance of successfully achieving its goals.

## Table of Contents

	<b>Page</b>
1 Scope.....	3
2 Overview.....	3
2.1 Common user profile potential .....	3
2.2 Common user profile hierarchy .....	4
2.3 Existing standardization .....	5
3 Interface adaptation .....	6
3.1 Personalization of audiovisual content .....	6
4 Use-case study examples for CUP implementations .....	7
4.1 Use-case of a geotagging application .....	7
4.2 Use-case adjusting screen layout of a virtual reality application .....	10
4.3 Defining classes in <i>C#</i> .....	14
4.4 Use-case – studies from the EU VUMS cluster .....	15
Bibliography.....	19

# Technical Report ITU-T TR.CUP

## Concept of a common user profile format used to personalize audiovisual media

### 1 Scope

This Technical Paper presents an overview of a common user profile (CUP) format used to personalize audiovisual media with example use-case studies that demonstrate how media can be adapted and personalized for users with different ranges of capabilities.

### 2 Overview

A one-size-fits-all approach to accessible media is rarely satisfactory but it is difficult to anticipate the personal accessibility requirements of the wide range of abilities of users when producing audiovisual media for distribution via broadcasting, on-demand services or the Internet.

Developing different versions of the same content targeted at people with different needs is not scalable, especially for large websites. Ways to adapt the same content for different users based on a user profile have had some success.

The SUPPLE project at University of Washington [b-Gajos], the Inclusive User Model at University of Cambridge [b-Biswas-1], IBM web adaptation technology [b-IBM] and AVANTI browser [b-Stephanidis] are notable examples, mostly targeted at users with different ranges of visual or motor impairment.

A user profile is an essential component for any personalization or adaptation process and since 2010 there have been various attempts to create a CUP format. The EU VUMS (Virtual User Modelling and Simulation) cluster [b-Biswas-2] took the ambitious step of publishing an exhaustive set of anthropometric, visual, cognitive, auditory, motor and user interface related parameters for adapting the human-machine interfaces of automobile, consumer electronics, audiovisual media etc.

The ITU Focus Group on Smart TV published a more compact set of parameters for creating user profiles. The technical report for Smart TV [b-ITU-T-FG SMART] describes a potential methodology to define the CUP but does not require that exactly same set of variables be used.

The rapid advance of technology and computing power has enabled users to access information through multiple devices with different sets of applications and software platform options. Ideally, an accessibility service should be provided to all devices and applications irrespective of the underlying hardware. The responsive design of websites can be considered an example of automatic adaptation of layout based on screen size and platform of deployment. However, information about a user's requirements is essential to personalize the content and layout of both text and audiovisual content.

#### 2.1 Common user profile potential

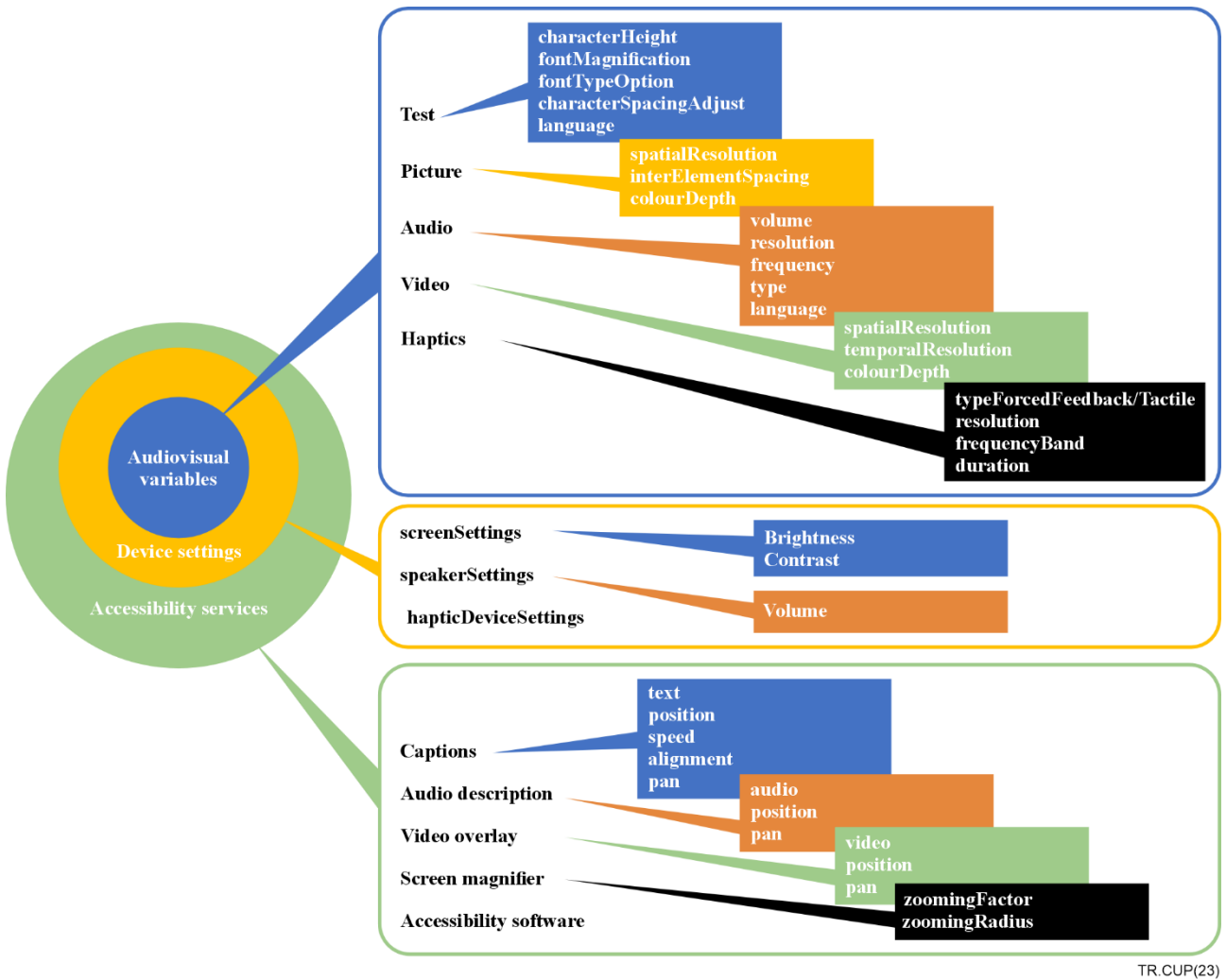
A user profile can be defined as an instance of a user model and a user model can be defined as a machine-readable description of a user's needs. When applied to personalized accessibility, a CUP can be said to have the following advantages:

- 1) Personalizing a user's device interface content and layout for different applications;
- 2) Offering personalized accessibility services to all devices and platforms;
- 3) Improving usability by sharing personalized content settings and interfaces across different platforms and devices;
- 4) Adapting the quality of accessibility services (e.g., font size of captions) across multiple media applications;

- 5) Secure sharing of personalization data between service providers or content developers;
- 6) Enabling AI agents to automatically personalize or evaluate user interfaces.

## 2.2 Common user profile hierarchy

Structurally, three hierarchical levels are used to describe variables (Figure 1). The innermost layer is the baseline text, audio, video, etc. The next layer describes the device media rendering options such as 2D or 3D display, speakers, haptic devices and so on. It also integrates the device properties for the range of adjustable settings available to the profile. Finally, the outermost layer describes various accessibility services and their associated properties.



TR.CUP(23)

**Figure 1 – Hierarchy for describing variables**

Examples of a sample CUP code describing captions for 2D and 3D rendering are given in Figure 2.

Caption	3D Caption
<pre> &lt;service&gt;Caption   &lt;text&gt;     &lt;characterSize&gt;12&lt;/CharacterSize&gt;     &lt;fontMagnification&gt;1.2&lt;/fontMagnification&gt;     &lt;fontTypeOption&gt;Calibri&lt;/fontTypeOption&gt;     &lt;characterSpacingAdjust&gt;1&lt;/characterSpacingAdjust&gt;     &lt;Language&gt;English&lt;/Language&gt;   &lt;/text&gt;   &lt;deviceSettings&gt;     &lt;screenSettings&gt;       &lt;brightness&gt;100&lt;/brightness&gt;       &lt;contrast&gt;50&lt;/contrast&gt;     &lt;/screenSettings&gt;   &lt;/deviceSettings&gt;   &lt;type&gt;simplified&lt;/type&gt;   &lt;position&gt;default&lt;/position&gt;   &lt;alignment&gt;default&lt;/alignment&gt;   &lt;speed&gt;default&lt;/speed&gt; &lt;/service&gt; </pre>	<pre> &lt;service&gt;3D Caption   &lt;text&gt;     &lt;characterSize&gt;12&lt;/CharacterSize&gt;     &lt;fontMagnification&gt;1.2&lt;/fontMagnification&gt;     &lt;fontTypeOption&gt;Calibri&lt;/fontTypeOption&gt;     &lt;characterSpacingAdjust&gt;1&lt;/characterSpacingAdjust&gt;     &lt;Language&gt;English&lt;/Language&gt;   &lt;/text&gt;   &lt;deviceSettings&gt;     &lt;screenSettings&gt;       &lt;brightness&gt;100&lt;/brightness&gt;       &lt;contrast&gt;50&lt;/contrast&gt;     &lt;/screenSettings&gt;   &lt;/deviceSettings&gt;   &lt;type&gt;simplified&lt;/type&gt;   &lt;position&gt;default&lt;/position&gt;   &lt;alignment&gt;default&lt;/alignment&gt;   &lt;speed&gt;default&lt;/speed&gt; &lt;/service&gt; </pre>

Figure 2 – Example profile for captions

### 2.3 Existing standardization

[b-ISO/IEC 24756:2009] provides the principles for managing user profiles.

[b-Biswas-2], EU VUMS (Virtual User Modelling and Simulation) cluster provides a comprehensive list of variables.

[b-ISO 9241-302:2008] provides a comprehensive terminology for electronic visual displays and explains the terms and definitions used in the other parts of ISO 9241.

[b-ISO 9241-307:2008] establishes test methods for the analysis of a variety of visual display technologies, tasks and environments. It uses the measurement procedures of [b-ISO 9241-305:2008] and the generic requirements of [b-ISO 9241-303:2011] to define compliance routes suitable for the different technologies and intended contexts.

[b-ISO/IEC 20071-23:2018] provides guidance for producers, exhibitors and distributors on the visual presentation of alternatives to audio information in audiovisual content, such as captions/subtitles.

The profiles activation, storage etc. can be made compatible with [b-ISO/IEC 24756] by following the naming defined in [b-ISO/IEC 24756].

Additionally, [b-ITU-T H.702], [b-ITU-R BT.2420] and [b-ITU-R BT.2447-1] provide guidance and examples of applications that can be addressed by the common user profile.

A service provider can add additional variables required by a specific application or for future technology developments.

Any user profile creation application should run locally to populate fields. It is not necessary to present all fields to the user separately. For example:

- Automated completion for colour blindness may predict a colour combination contrast;
- Visual acuity can predict minimum font size and modality;
- Audiometry data can anticipate relative dialogue to background balance and loudness.

### 3 Interface adaptation

Figure 3 shows examples of interface adaptation across multiple devices and platforms using a CUP format. The image colour, contrast, font size and inter-element spacing of icons can be adjusted across Smart TV, desktop and laptop computers, smartphones and low-end mobile phones based on an applied CUP.



Font size and colour contrast adaptation for disaster warning application in a computer

Home automation application for a smart TV

Font size and colour contrast adaptation for disaster warning application in a smartphone

e-Agri application for low end mobile phones

TR.CUP(23)

**Figure 3 – Interface examples using CUP format**

#### 3.1 Personalization of audiovisual content

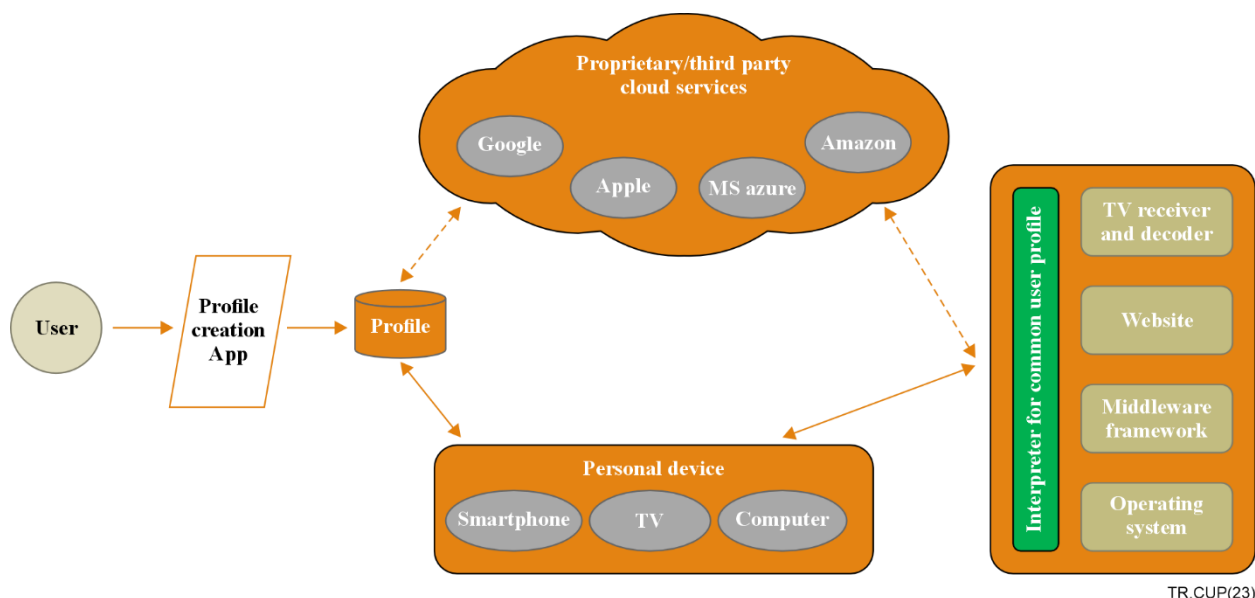
In practice, a user would set up a CUP by:

- Creating a profile using a profile creation application (PCA).
- Deciding to enable the option to upload the profile to a third-party cloud application. This is not mandatory and is indicated by the dashed lines in Figure 4.
- Transfer the profile from the PCA and store it in a personal device (or in another location on the PCA device if both applications are on the same device).
- Create multiple profiles for different devices if needed.

After the user has completed the process:

- An interpretation application on the user's device will adapt the user interface based on the user's profile;
- The interpretation application shall understand the variables defined in the CUP in order correctly interpret the personalization parameters for the content.





**Figure 4 – Sequence of CUP implementation actions**

#### 4 Use-case study examples for CUP implementations

During the development stage of a CUP settings template or a user application it is useful to reference use-case scenarios that include multiple devices or platforms or both.

##### 4.1 Use-case of a geotagging application

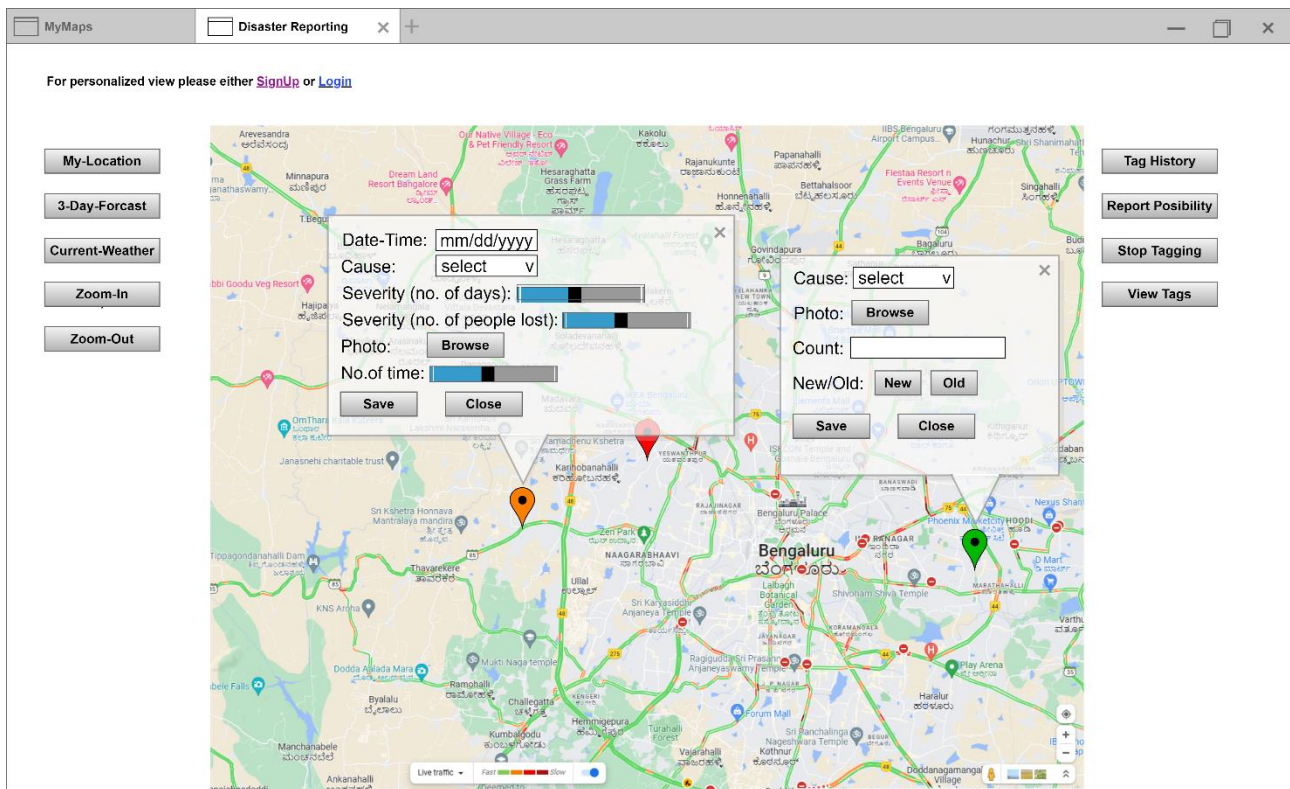
*A use-case study of a geotagging application for three different platforms, personalized using the CUP format parameters.*

The application offers tagging an exact location on Google Maps (Figure 5) overlaid with customized labels. The tagged data can be shared with other users who can access the exact longitude and latitude of the tags.

The application also includes the current and next three days weather forecasts and has a current traffic layer option when zooming in to a given point or tag. For example, a travel website can allow different users to tag various locations such as ATMs, places to visit, medical centre, etc. that have accessible features or facilities.

Similar applications are designed for reporting civic and health-care issues for the quick identification of serious problems. The geospatial and temporal clustering of information provided by elderly users will be useful for the quick identification and resolution of critical civic and health-care problems.

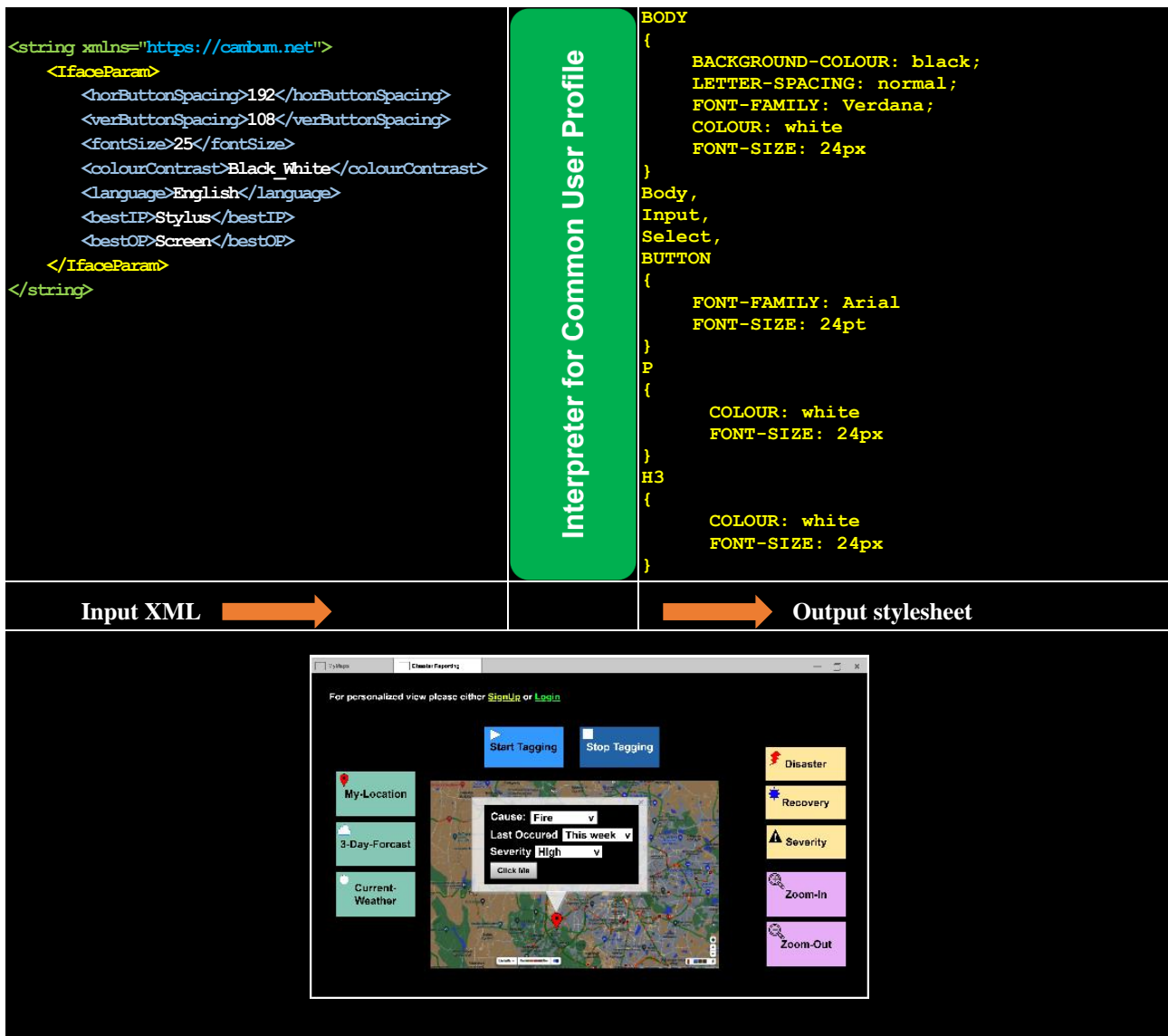
The application is deployed as a webpage (<https://cambum.net/CambUM/GeoTag/>) and an Android app for smartphone (Figure 6).



**Figure 5 – Non-adapted default view of geotagging web application action<sup>2</sup>**

The web application for the CUP format is used to select an appropriate cascading stylesheet (CSS) from a pre-determined template list which has been developed based on earlier research on inclusive user modelling [b-Biswas-1] and the EU GUIDE [b-Biswas-2] project. The stylesheets ensured the adaptation of screen content is within the bound set by the geometry of screen elements, for example, the font size of a button caption is not enlarged enough to spill out from a button control.

<sup>2</sup> The designations employed and the presentation of material on this map do not imply the expression of any opinion whatsoever on the part of ITU and of the Secretariat of the ITU concerning the legal status of the country, territory, city or area or its authorities, or concerning the delimitation of its frontiers or boundaries.



**Figure 6 – CUP example rendering implementation input and output with example render**

Another scenario is a disaster reporting tool for users as well as the local, regional and national authorities.

Users in affected areas or who can see the possibility of an incident e.g., rising water levels, potential landslides or avalanches, can tag exact locations. This information can be accessed by local authorities and other users which will help them maintain their safety or allow pre-emptive action to mitigate the impact.

For a visually impaired user, the rule-based system selects a stylesheet based on font size requirement and type of colour blindness of users. The user PCA calculates and stores minimum visual angle requirements by estimating distance to screen from type of device and screen resolution.

For this application, the pop-up tagging windows are shown in yellow-blue colour contrast for red-green colour-blind users and in reverse contrast for any other type of colour blindness (Figure 7).



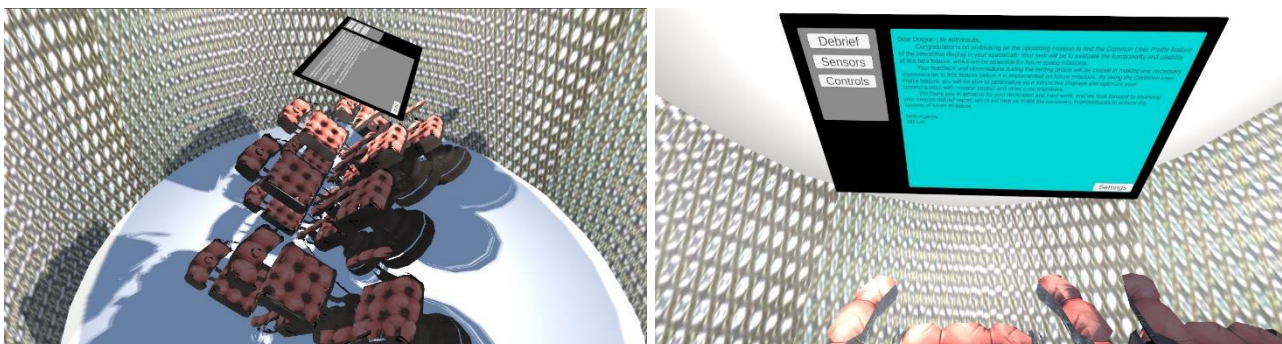
**Figure 7 – Personalized implementation examples for people with visual impairment**

This application can be further enhanced at any time by adding additional data to the map interface which will allow many more user options to be included.

## 4.2 Use-case adjusting screen layout of a virtual reality application

*A case study of an application to adjust visual appearance for virtual reality applications using the CUP format parameters.*

For a virtual reality application based on the SpaceX Dragon space shuttle, modelled in Unity, was used. The interior design (Figure 8) of the Dragon are well suited for this type of application as it is described as spacious, sleek and comfortable with touchscreen minimalistic control surfaces which are highly intuitive and responsive, allowing astronauts to quickly access critical information and adjust as needed. The seating configuration can be customized for each mission.



**Figure 8 – Virtual reality simulation SpaceX crew capsule**

A CUP accessibility setting enables users of the VR application, or virtual 'astronauts', with visual impairments to customize the displays to meet their specific needs such as adjusting the font size,

contrast and colour scheme of the text on the screens. This helps ensure all crew members can read the displays and access critical information without difficulty.

A CUP accessibility option would enable customization of the displays to suit personal preferences such as for language, units of measurement, layout and accessibility. These adaptations would ensure all virtual crew members can easily navigate the displays and operate systems with confidence, regardless of their individual preferences, minimize the potential for errors and help astronauts carry out their mission with confidence and ease.

NOTE – There is also no technological reason why this VR application could not also be applied to future real life SpaceX capsule design, or similar craft for long term colonization missions. Figure 9 shows CUP simulation options overlaid on an image of the real capsule screens.



**Figure 9 – Simulation of CUP adaptation on the actual SpaceX crew capsule screens**

The 3D virtual application was developed to include the feature set of a CUP for visually impaired users. The font size, font colour and background colour can be changed based on the user's preference and these parameters can be loaded using XML files.

The XML files containing this data makes it easier to change the parameters quickly without opening the application itself or rebuilding the application with required changes included.

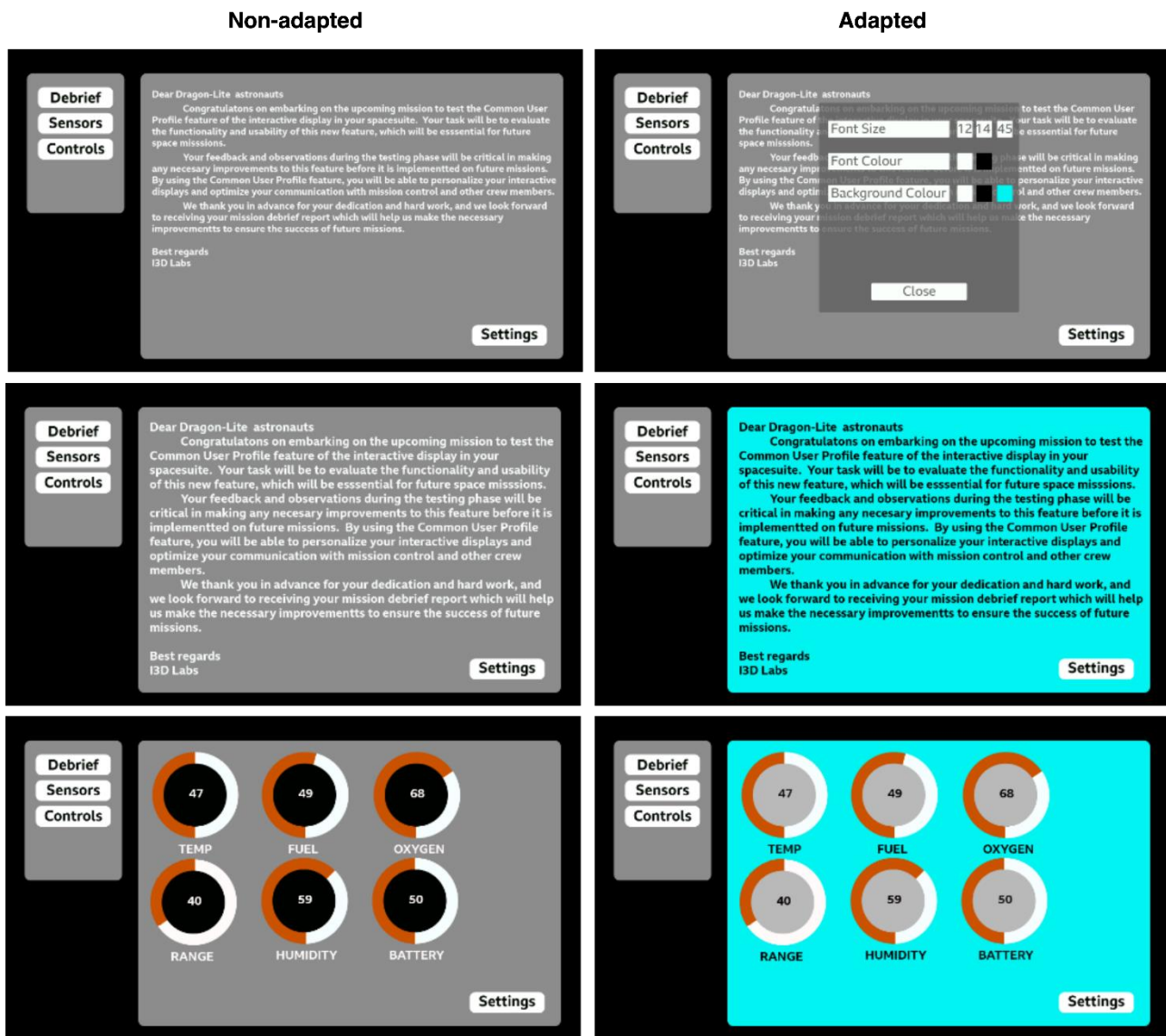


Figure 10 – Non-adapted and simulated CUP adapted screens

Figure 10 demonstrates the visual impairment construct comparing the non-adapted and the adapted version.

#### 4.2.1 XML construct for the SpaceX VT application

*This section outlines the process of building the XML application files to implement the screen parameters.*

The XML creation process can be described in four stages. Figure 11 provides example content for steps 1-3

Step-1 Create an XML file which holds the various values such as font size, font u, background colour, foreground colour, in this case,

- font size: 12,14,16
- text colour: black, white
- background: black, cyan

Step-2 A class is created for each of the features, such as text size or colour, which assigns a variable to the respective XML tag bearing the actual value. This means the only variable will be the XML value.

Step-3 Creation of a colour object with the A, R, G, B Alpha values assigned with the variables that were created during in the Class file.

1 - XML	2 - XML classes	3 - Accessing XML data
<pre> &lt;trigger&gt;   &lt;font&gt;     &lt;size0&gt;12&lt;/size0&gt;     &lt;size1&gt;14&lt;/size1&gt;     &lt;size2&gt;16&lt;/size2&gt;   &lt;/font&gt;   &lt;colour&gt;     &lt;a&gt;1&lt;/a&gt;     &lt;r&gt;0&lt;/r&gt;     &lt;g&gt;0&lt;/g&gt;     &lt;b&gt;0&lt;/b&gt;   &lt;/colour&gt;   &lt;bgcolor&gt;     &lt;a&gt;1&lt;/a&gt;     &lt;r&gt;0&lt;/r&gt;     &lt;g&gt;0&lt;/g&gt;     &lt;b&gt;0&lt;/b&gt;   &lt;/bgcolor&gt;   &lt;cyancolour&gt;     &lt;a&gt;1&lt;/a&gt;     &lt;r&gt;0&lt;/r&gt;     &lt;g&gt;152&lt;/g&gt;     &lt;b&gt;166&lt;/b&gt;   &lt;/cyancolour&gt;   &lt;whitecolour&gt;     &lt;a&gt;1&lt;/a&gt;     &lt;r&gt;255&lt;/r&gt;     &lt;g&gt;255&lt;/g&gt;     &lt;b&gt;255&lt;/b&gt;   &lt;/whitecolour&gt; &lt;/trigger&gt; </pre>	<pre> [xmlRoot]ElementName = "colour") public class ColourXml {   [XmlElement(ElementName = "a")]   public float aFloat;   [XmlElement(ElementName = "r")]   public float aFloat;   [XmlElement(ElementName = "g")]   public float aFloat;   [XmlElement(ElementName = "b")]   public float aFloat; }  [xmlRoot]ElementName = "font") public class FontXml {   [XmlElement(ElementName = "size0")]   public int Size0;   [XmlElement(ElementName = "size1")]   public int Size1;   [XmlElement(ElementName = "size2")]   public int Size2;   [XmlElement(ElementName = "style")]   public string style; } </pre>	<pre> private void TextToBlack(triggerxml triggerxml) {   float red = triggerxml.colour.rfloat   float green = triggerxml.colour.gfloat   float blue = triggerxml.colour.bfloat   blacktext.a=alpha;   blacktext.r=red;   blacktext.g=green;   blacktext.b=blue;   foreach(var text in texts)   {     text.GetComponent&lt;TextMeshProUGUI&gt;().colour =     blacktext;     // text.GetComponent&lt;text&gt;().colour = blacktext   } } </pre>

Figure 11 – Example XML creation (steps 1 to 3)

Step-4 Using the GetComponent function (Figure 12) to obtain the desired text elements and assign the variable from the Class file which holds the values in the XML file.

```

private void Sizeone(Triggerxml triggerxml)
{
  foreach (var text in text)
  {
    text.GetComponent<TextMeshProUGUI>().fontSize = triggerxml.font.size0)
  }
}

private void Sizetwo(Triggerxml triggerxml)
{
  foreach (var text in text)
  {
    text.GetComponent<TextMeshProUGUI>().fontSize = triggerxml.font.size1)
  }
}

private void Sizethree(Triggerxml triggerxml)
{
  foreach (var text in text)
  {
    text.GetComponent<TextMeshProUGUI>().fontSize = triggerxml.font.size2)
  }
}

```

Figure 12 – Example of using the XML data GetComponent (step 4)

### 4.3 Defining classes in C#

The respective classes should also be defined in C# where we can define the variables: font size, colour, background colour, where font is a nested XML tag as shown in Figure 13.

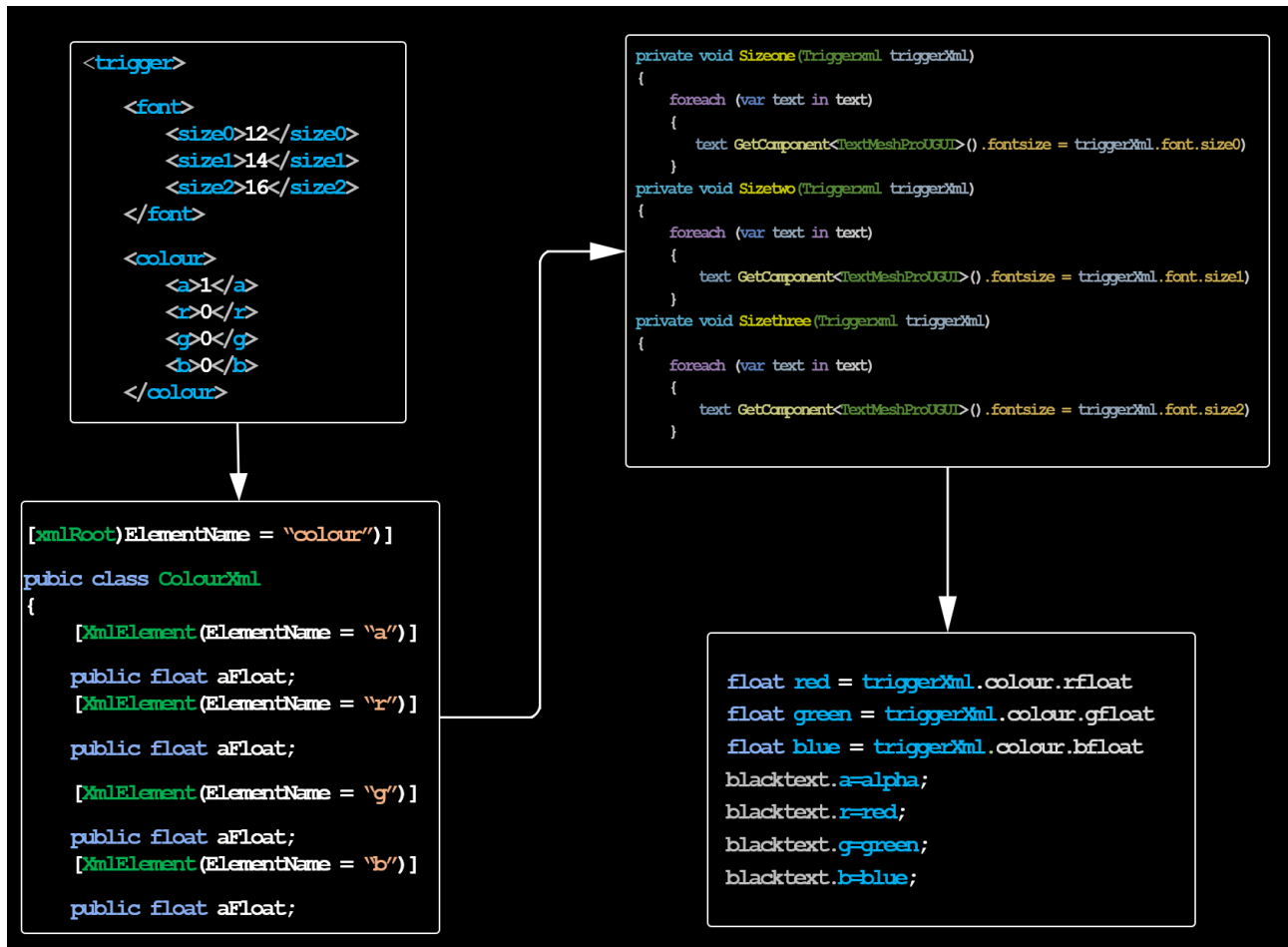


Figure 13 – XML process map overview

To let C# know these are meant to be associated with XML files, the annotations described in Figure 14 could be used.

In this example following classes were created:

- FontXml contains nested font sizes as size increments;
- BGColorXml contains colour data in ARGB format (alpha transparency, red, green and blue);
- CyanColorXml contains colour data in ARGB format;
- WhiteColorXml contains colour data in ARGB format.

C# has built-in functions to import the XML files and C# classes into the C# objects which was used to import into the Unity project.

The XML is then imported into the project scene with a script attached to an object in the scene. The XML path is defined in the Start () method of the script. The next step is to define functions and assign the font and colour properties to text and background objects using the Class to define what is assigned to the object.

Initially the script required all text on every page to be referred to manually in the inspector which means a different script for each different screen of an application. As many text object datatypes are



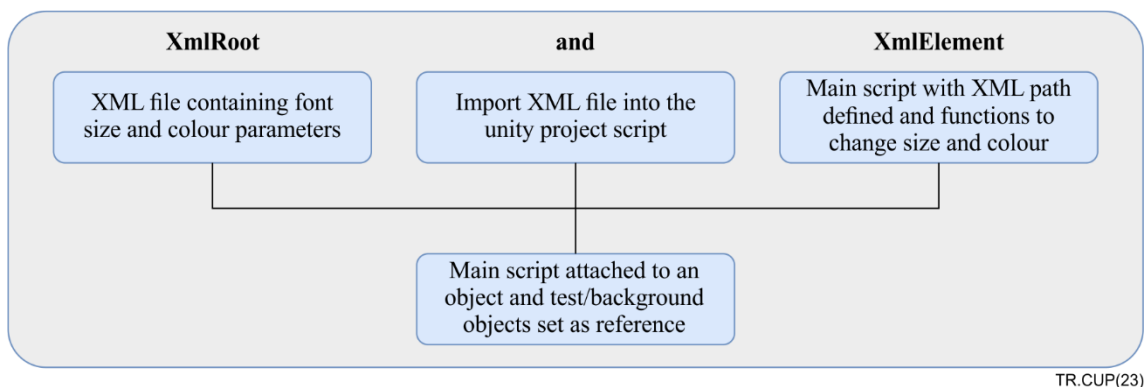
used in any given user interface (UI), this would mean that if a UI has 12 text fields, then 12 objects of the datatype 'Text' needed to be created. This will result in a large amount of redundant code.

Changing this approach to one where the script is designed to automatically find text objects autonomously and stores them in an unsorted list. When any new text field is added it be tagged 'text' so the automation can detect it, eliminating the repeated manual action referring text objects for each UI page.

Using the current XML files, the automation script has the potential to work as a plug-and-play option in any UI interface developed using Unity. Buttons will have to be referred to script function for the desired changes in the UI.

#### 4.4 Use-case – studies from the EU VUMS cluster

*A use-case study that considers a representative persona and shows examples of simulation and adaptation for different applications.*



**Figure 14 – Defining in C# and import to Unity**

The CUP format allows different applications and devices to share a single profile and simulate and adapt interface based on this common profile.

Subject A: Mr John Brown is a 70-year-old male with spinal cord injuries and glaucoma. Due to functional limitations, John has difficulty walking, grasping and reading, which requires the use of some assistive devices, including a wheelchair and reading glasses. He is not colour-blind but has age-related hearing impairment requiring a higher loudness threshold for high frequency sound. His scores in cognitive tests such as trail making and digit symbol tests show no signs of cognitive impairment.

Figure 15 is a screenshot of the GUIDE Home Automation application and Figure 16 simulates how John perceives the screen. The black spots are due to glaucoma and the blue line shows the movement of the cursor while he operates the application using a direct manipulation input device such as a gyroscopic remote or trackball. The message box predicts task completion time.



**Figure 15 – GUIDE application screen**



**Figure 16 – GUIDE application screen simulation perception of subject A**

When the CUP is used to adapt interfaces, Table 1 presents a set of interface parameters predicted by the GUIDE system for this particular user.

GUIDE interfaces use these parameters to adapt application interfaces by updating a UIML (user interface markup language) description of interface layouts.

**Table 1 – Interface parameter prediction for common persona**

Device	Button spacing		Minimum font Size	Colour contrast	Best modality	
	Horizontal	Vertical			Input	Output
	Size in pixels					
Mobile	48	80	26	Any	BigButton	Screen
Laptop	128	80	24	Any	TrackBall	Screen
Tablet	128	80	23	Any	Stylus	Screen
TV	200	120	58	Any	Second screen BigButton	Screen

Figure 17 shows adaptation of the same home automation application for different user profiles.

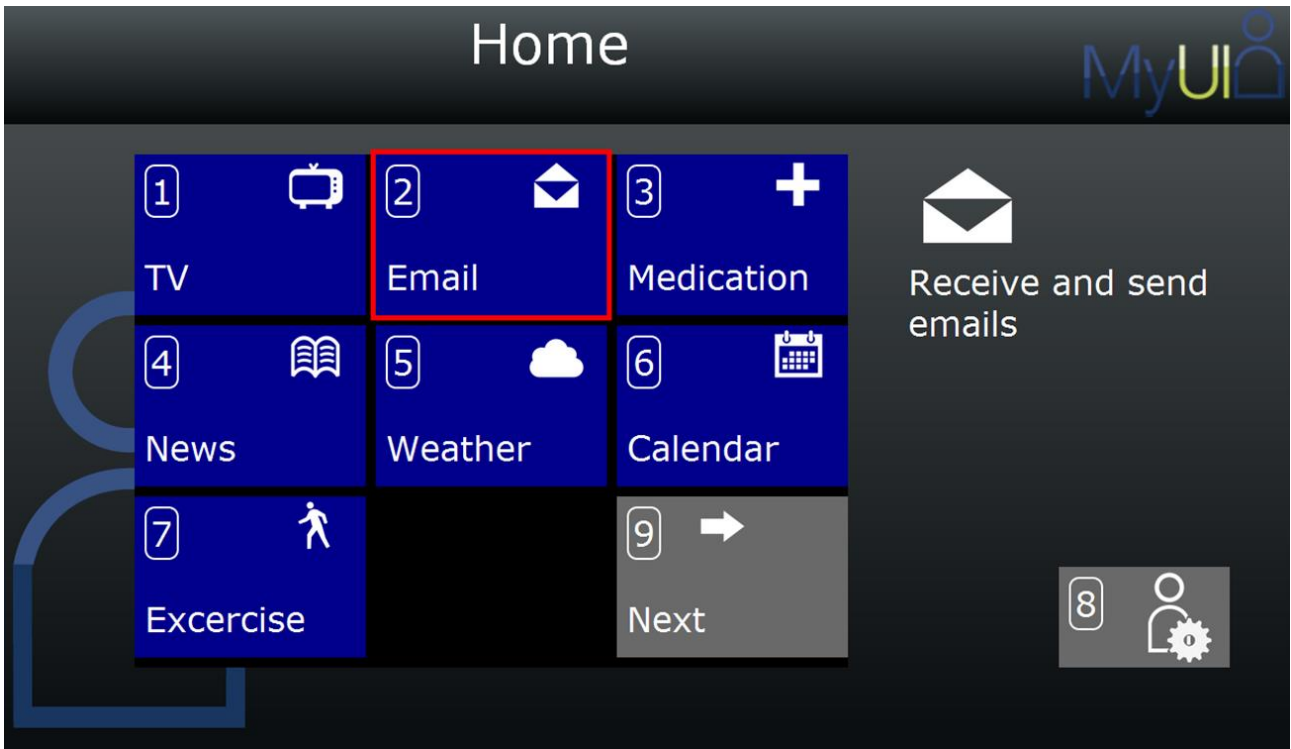


**Figure 17 – Home automation application for different user profiles**

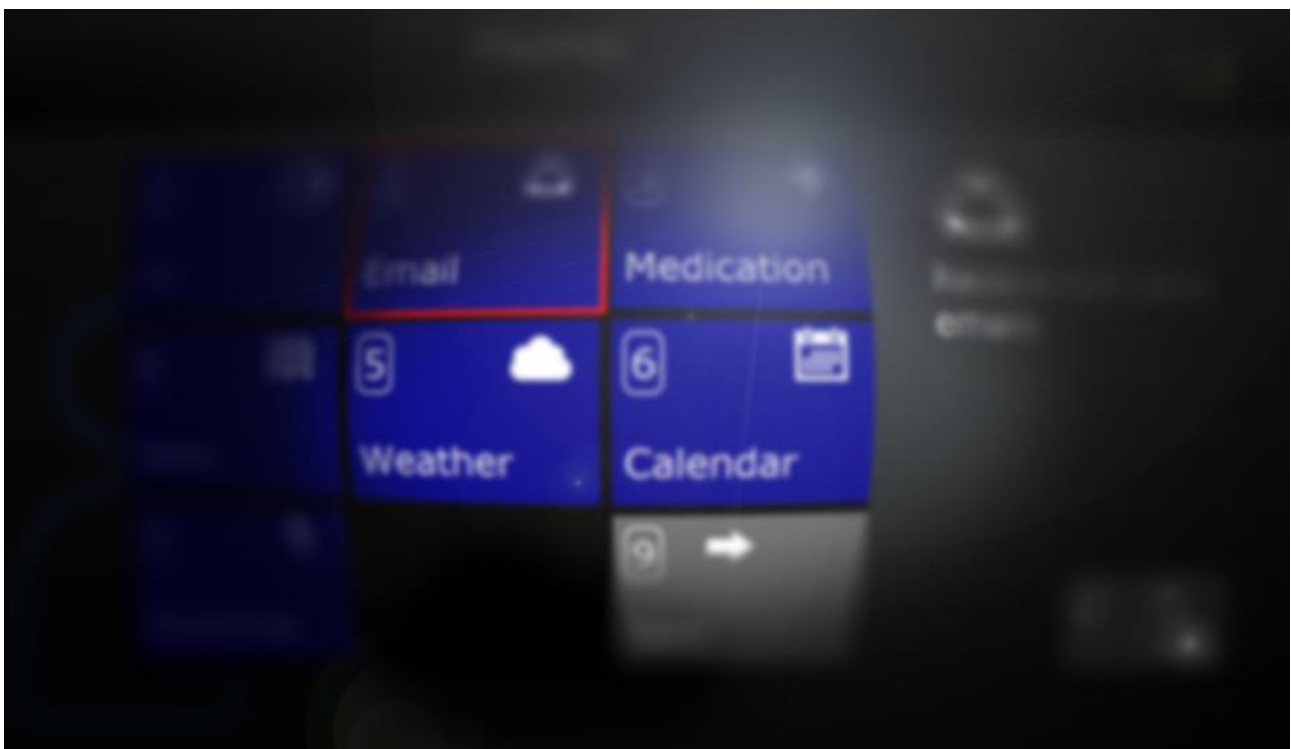
The EU MyUI system used the same profile to adapt the main menu of the MyUI adaptive user interface demonstrating how it would look for subject A. In this simulation the following parameters were adapted:

- Font size is increased to address visual perceptual impairment (testing had already suggested difficulties in reading were not due to cognitive impairments);
- Numeric key navigation was enabled in addition to simple cursor navigation, to address motor function impairment (grasping). This resulted in displaying the respective numbers on every interactive element;
- The number of displayed interactive elements (menu buttons) has been reduced to a subset with ten options (keys 0-9) as enabling numeric key navigation requires more screen space.

Figure 18 shows the adapted the MyUI home page menu simulation and Figure 19 simulates how subject A perceives it.



**Figure 18 – MyUI home page adaptation simulation**



**Figure 19 – MyUI home page interface as perceived by subject A**

## Bibliography

- [b-ITU-T H.702] Recommendation ITU-T H.702 (2020), *Accessibility profiles for IPTV systems*.
- [b-ITU-T-FG SMART] ITU-T Focus Group on Smart TV (31/03/2012), *Technical Report*.  
[http://www.itu.int/dms\\_pub/itu-t/opb/fg/T-FG-SMART-2013-PDF-E.pdf](http://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-SMART-2013-PDF-E.pdf)
- [b-ITU-R BT.2420] Report ITU-R BT.2420 (2020), *Collection of usage scenarios of advanced immersive sensory media systems*.
- [b-ITU-R BT.2447-1] Report ITU-R BT.2447-1 (2022), *Artificial intelligence systems for programme production and exchange*.
- [b-ISO/IEC 20071-23:2018] ISO/IEC 20071-23:2018, *Information technology – User interface component accessibility – Part 23: Visual presentation of audio information (including captions and subtitles)*.
- [b-ISO/IEC 24756:2009] ISO/IEC 24756:2009, *Information technology – Framework for specifying a common access profile (CAP) of needs and capabilities of users, systems, and their environments*.
- [b-ISO 9241-302:2008] ISO 9241-302:2008, *Ergonomics of human-system interaction – Part 302: Terminology for electronic visual displays*.
- [b-ISO 9241-303:2011] ISO 9241-303:2011, *Ergonomics of human-system interaction – Part 303: Requirements for electronic visual displays*.
- [b-ISO 9241-305:2008] ISO 9241-305:2008, *Ergonomics of human-system interaction – Part 305: Optical laboratory test methods for electronic visual displays*.
- [b-ISO 9241-307:2008] ISO 9241-307:2008, *Ergonomics of human-system interaction – Part 307: Analysis and compliance test methods for electronic visual displays*.
- [b-Biswas] Biswas P., Robinson P., Langdon P. (2012), *Designing inclusive interfaces through user modelling and simulation*, International Journal of Human-Computer Interaction Vol. 28, No. 1, 1-33.
- [b-Biswas-2] Biswas, P., Kaklanis, N., Mohamad, Y., Peissner, M., Langdon, P., Tzovaras, D. and Jung, C. (2015), An Interoperable and Inclusive User Modeling Concept for Simulation and Adaptation. In: Biswas, P., Duarte, C., Langdon, P., Almeida, L. (eds), *A Multimodal End-2-End Approach to Accessible Computing. Human-Computer Interaction Series*, London, Springer.
- [b-Gajos] Gajos K.Z., Wobbrock J.O. and Weld D. S (2007), *Automatically generating user interfaces adapted to users' motor and vision capabilities*, ACM Symposium on User Interface and Software Technology (UIST), 231-240.

[b-IBM]

IBM (2007), *Web Adaptation Technology*.

[b-Stephanidis]

Stephanidis C., Paramythis A., Sfyraakis M., Stergiou A., Maou N., Leventis A., Paparoulis G. and Karagiannidis C. (1998), *Adaptable and Adaptive User Interfaces for Disabled Users in the AVANTI Project*, *Intelligence in Services and Networks*, Vol. LNCS-1430, 153-166.

---