MAKING FORMS ACCESSIBLE

Norbert Kuhn, Stefan Naumann, Stefan Richter, Michael Schmidt and Andreas Truar*
Institute for Software Systems in Economics, Environment and Administration
University of Applied Sciences Trier, Umwelt-Campus Birkenfeld, Birkenfeld, Germany

Keywords: Accessibility, Adaptable User Interfaces, Access to e-Government forms and processes.

Abstract: We observe that paper is still the dominant means to transport information. In particular, for the communication between governmental authorities and citizens paper based documents still play a vital role. In this paper we discuss ideas how documents can be designed so that they can adapt almost automatically to an actual reader. Therefore, we describe an approach to design documents and governmental forms so that they may change their appearance in accordance to the preferences of their recipients. This allows the creation of documents that can better be understood by different target groups, among them elderly people, people with visual impairments or people with dyslexia.

1 INTRODUCTION

In recent years much effort has been spent in Human Computer Interfaces to improve access for handicapped persons to computer systems. To a major extent these activities were enforced by legislative constraints that exist in the US, e.g. the Americans with Disabilities Act (United States of America, 1990) as well as in the European Union (European Commission, 2000), and in its member countries (e.g. (Bundesrepublik Deutschland, 2006)). In particular, governmental authorities are in charge to provide accessible solutions. However, to a large amount current realizations allow only to download particular forms, to print them, and to send it back to the governmental institution after some information has been inserted. This yields a point of media disruption which is difficult to handle for many users with particular handicaps. Therefore, we discuss in more detail an approach to build interfaces to governmental forms.

In this paper we focus on the requirements of visual impaired humans when they communicate with public authorities. For this kind of communication electronic information systems are increasingly used. However, when an authority has to contact a citizen usually this is still done by sending paper documents. In the context of governmental processes these are predominantly forms. While for the web based information systems accessibility aspects are often considered in e-Government platforms, for the procedure of filling paper forms support for disabled persons is almost missing. Hence, we believe that an integrated view to printed and electronic media is of particular value for disabled people. Moreover, what is sometimes mentioned in the context of efforts to improve accessibility, this may also improve access for users without handicaps.

We suggest document representations based on XML structures and the implementation of communication by using web services. This guarantees independence from software and hardware platforms. For electronic forms different XML representations have been developed, among them the XForms standard from W3C ((World Wide Web Consortium, 2007)), or the XFA-Standard ((World Wide Web Consortium, 1999)), which built the basis for Adobe's Acroforms format ((Adobe Corp., 2003)). We can use these standards for certain parts of our document representation, e.g. XForms for describing the the different fields of a form. Acroforms allows even more information to be described, e.g. information for speech synthesis of a document. However, in our setting we need to store even more information, e.g. to control the OCR software to read a form containing information which has been already filled in.

*This work is supported by the German Federal Ministry of Research and Technology under grant 1771X07.

Kuhn N., Naumann S., Richter S., Schmidt M. and Truar A.
MAKING FORMS ACCESSIBLE.
DOI: 10.5220/0001823504030406
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

2 SYSTEM ARCHITECTURE

In order to demonstrate our ideas of interfaces which are able to handle forms which may either be available as printed version or in electronic form we implemented a prototypical system called GUIDO (Generating User-specific Interactive Documents). This system maintains documents that adapt their appearance according to the needs of an actual user. We concentrate on users which have problems to perceive information in a conventional document representation. This group comprises blind people, people with low vision, defective vision, color deficiencies or daltonism, but also illiterates and immigrant people.

All these users pose quite different requirements to the design of and the interaction with an information system. In order to be able to economically develop adequate interfaces the system should react almost automatically to the actual user's needs.

Therefore, our system consists of different components. The core component is a document database where all the digital documents of a governmental authority are stored. A database entry for a document consists of different data. This encompasses an image of the document which can be used to print it or display it on a screen. Furthermore, we use a semantic description of the document, including information about the fields of a form, their type, and possible relations between them, which ones are already filled with information and which ones have to be completed by the current user. Also a link to the workflow management in the authority should be present. We call the universe of information associated with a governmental document or form the Generic Document Structure (cf. chapter 2.1).

When a user wants to process a form which he has received from an authority we assume that he starts communication with the governmental server by sending an image of this form to the server. Together with the image, the device profile and his user profile are sent (cf. chapter 4).

2.1 The Generic Document Structure

Forms in administrative processes represent information in a structured form. We map this structure into the Generic Document Structure. It addresses aspects like layout and semantics, as well as particular functions to process a form.

The layout part for a document is concerned with its graphical elements, e.g. its pages, blocks, paragraphs and words. They may be regarded as elements of a GUI which we describe in a form similar to approaches known e.g. from the eXtensible Application Markup Language (XAML, (Microsoft MSDN, 2008)) or the XML User Interface Language (XUL, (The Mozilla Foundation, 2008)). A user agent which is able to understand such a language is able to draw the form as a GUI. The features of the user agent are communicated by the device profile.

Within the semantics part we consider contextual relationships between layout elements and the meaning of specific fields. One example can be the gender information in a form which is often given by two checkboxes (female/male). The exclusive choice between the two checkboxes is stored in the semantics part of the generic document structure. In the function part we define rules to verify the user's input against the desired semantics. This can e.g. check whether the given city code may be correct and whether it matches the name of the city given in another field. Additionally, we may include information for the process to which the form belongs to. Such an extension allows to describe what happens to the form after it has been completed and sent by the user.

2.2 The Server Workflow

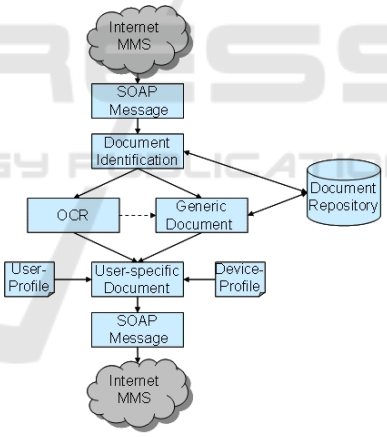

Figure 1: The server workflow.

Figure 1 sketches the computation steps on the server side when a user wants to process a document. The GUIDO system aims to derive for him a suitable representation of the governmental document with which he is currently concerned with. We call this representation the User-specific Interactive Document. The client can specify the reference to the Generic document structure either by a URL or by an image of the printed form. Given the image, which is transmitted together with the information specified below via the SOAP message in the upper part of figure 1, the server tries to retrieve the suitable document template from its document repository.

The server also needs information about the user's perceptual and physical capabilities. This data is contained in the user profile. For instance it can specify that the user wants to receive no image but an audio version of the document. With this profile the user can customize the document representation to his personal requirements. Besides the user's physical capabilities a user specific answer needs to consider the user's device configuration. The technical configuration is stored in the device profile. For example the audio representation can either be constructed on the server or on the user's device. The choice can be based on the actual bandwidth available for the client-server communication and on the features of the user's device. If the client computer has no speech synthesis program available the audio information has to be computed on the server. Bandwidth or preference parameters will then determine the compression rate of the audio information.

3 USER AGENTS

The user can access the document service with different devices which may have different hard- and software configurations. This has to be considered for the delivery of the service. In the following we sketch our implementation of a User Agent on a smart phone. A description of a desktop user agent may be found e.g. in (Kuhn et al., 2008).

Today a wide range of mobile devices with multimedia capabilities is available, among them cell phones and PDAs (Personal Digital Assistants). An important difference between cell phones and PDAs concerns the display. The size of cell phone displays fairly varies and is often too small to display large content. In many cases these are only black and white dot-matrix displays without color. PDA displays are more homogeneous in their size and colored. Displaying an image on a small dot-matrix display makes no sense, whereas playing an audio stream is easily possible. When using a device with a QVGA screen resolution we may also display an image. For example, figure 2 shows the representation of a governmental form on a smart phone device. The word with the focus (the one, which is actually read) is surrounded by a colored box.

Another fact to consider is the lack of a keyboard on most mobile devices. Instead, we can use the microphone that mobile phone devices have always built-in. We will record the user's voice and perform speech recognition. The recognition process can be executed on the device itself or on the server that hosts the document repository. All these differences between the many devices are represented in the device profile.

Figure 2: User Agent on a smart phone.

4 USER PROFILE AND DEVICE PROFILE

To transform forms into a user specific format we use the user and the device profile. The user profile defines personal preferences for the visual and the audible representation of a document. An example of an entry in the user profile would be the preferred color combination of the document when it is displayed on a device (e.g. black-white, black-cyan, cyan-yellow, or none). For the audio result the user profile contains the compression method and the sampling rate. The following code shows some parameters of the user profile.

```
<User>
  <OCR> true/false </OCR>
  <TTS> true/false </TTS>
  <COLORTRANSFORMATION>
   string
  </COLORTRANSFORMATION>
  <TTS_SPEED>speed </TTS_SPEED>
  <TARGET_LANGUAGE>
   language
  </TARGET_LANGUAGE>
  <TTS_VOICE> voicename </TTS_VOICE>
  <COMPRESSION_RATE>
   compression
  </COMPRESSION_RATE>
</User>
```

The user profile is almost the same no matter which hardware is used. This is different for the device profile. It contains information about the client that is used to display the document. Depending on

the hardware and software configuration of the device it may be different for each client. An example for an entry is the screen resolution which influences the document representation. E.g., we consider display dimensions, resolution or colors. To produce a sound file it is important to know which audio formats are accepted by the client. This information is also included in the device profile. The following code shows an excerpt from a device profile.

```
<Device>
      <DEVICE_RESOLUTION_HEIGHT>
       screenheight
      </DEVICE_RESOLUTION_HEIGHT>
      <DEVICE_RESOLUTION_WIDTH>
       screenwidth
      </DEVICE_RESOLUTION_WIDTH>
      <AUDIOFORMAT>
       format
      </AUDIOFORMAT>
      <OCR>true/false </OCR>
      <GUIDO_PLAYER>
       useragent
      </GUIDO_PLAYER>
      <NETWORK_CONNECTION>
       connection
      </NETWORK_CONNECTION>
   </Device>
```

5 CONCLUSIONS

Since the beginnings of the 1990s, when the GUIs developed more and more, the development of interfaces for disabled people is always a couple of steps behind (Glinert and York, 2008). To shorten this distance we regard our approach as something similar to what Vanderheiden calls a *pluggable user interface* (Vanderheiden, 2008). He states that the ability to invoke any assistive technologies or special features that are needed directly from the Net to use on whatever displays are nearby, may be the most effective means of access.

At the moment the GUIDO system provides humans with reading disabilities with much better access to information. In particular, access to printed documents, which are still important in business processes or in government applications, is substantially improved. In this paper we have sketched our ideas to automatically adapt the output of a centralized information portal of a governmental authority to the special need of a citizen looking for information. Furthermore, the service can also be used for simpler tasks, e.g. when a visual impaired person comes to a bus stop and wants to read the schedule.

Currently, we focus mainly on the group of users with visual disabilities. This group of users will increase in number at least in many European countries due to the demographic development. A further possible target group are people with difficulties in reading comprehension, e.g. immigrants or dyslexic people.

REFERENCES

Adobe Corp. (2003). Acrobat Forms API Reference, Technical Note #5181. http://www.adobe.com/devnet/acrobat/pdfs/ FormsAPIReference.pdf. visited: 2009/01/19.

Bundesrepublik Deutschland (2006). Allgemeines Gleichbehandlungsgesetz der Bundesrepublik Deutschland. http://bundesrecht.juris.de/agg/. visited: 2009/01/19.

European Commission (2000). Gleichbehandlung Behinderter in Beruf und Bildung: Richtlinie des Rates 2000/78/EG vom 27. November 2000.

Glinert, E. P. and York, B. W. (2008). Computers and People with Disabilities (reprint from 1992). In *ACM Trans. Access. Comput. Vol. 1, Issue 2*, page Article No. 7 (October 2008).

Kuhn, N., Richter, S., Schmidt, M., and Truar, A. (2008). Improving Accessibility to Governmental Forms. In *Human-Computer Interaction Symposium*, volume 272/2008, pages 118–128, Milano, Italy. Springer Boston.

Microsoft MSDN (2008). Windows Presentation Foundation - XAML. http://msdn.microsoft.com/en-us/library/ms747122.aspx. visited: 2009/01/19.

The Mozilla Foundation (2008). XML User Interface Language (XUL) - Project Documentation. http://developer.mozilla.org/en/docs/XUL. visited: 2009/01/19.

United States of America (1990). Americans with Disabilities Act, Public Law 336 of the 101st Congress. http://www.ada.gov/pubs/ada.htm. visited: 2009/01/19.

Vanderheiden, G. C. (2008). Ubiquitous accessibility, common technology core, and micro-assistive technology. In *ACM Trans. Access. Comput. Vol. 1, Issue 2*, page Article No. 10 (October 2008).

World Wide Web Consortium (1999). XML Forms Architecture. http://www.w3.org/1999/05/XFA/xfa-template.html. visited: 2009/01/19.

World Wide Web Consortium (2007). XForms 1.0 (Third Edition). http://www.w3.org/TR/xforms/. visited: 2009/01/19.