Support Method to Elicit Accessibility Requirements

Junko Shirogane

Tokyo Woman's Christian University, Japan
junko@lab.twcu.ac.jp

Abstract. Various accessibility guidelines have been developed to meet the increased demand for accessible software, but due to the numerous elements within these guidelines, applying all elements to target software is burdensome and expensive. Additionally, whether all the elements should be applied depends on the software's purpose and target end users, who do not often clearly recognize difficulties. Moreover, accessibility requirements elicited in the late software development phase cannot always be applied. To ensure that these requirements are implemented properly, they must be elicited in the early software development phase by considering end users' conscious and unconscious characteristics. Here a method to elicit accessibility requirements in the early software development phase is proposed. Specifically, end users complete checklists, which are designed to determine disabilities with respect to guidelines. Then guideline elements are prioritized and applied to the target software as specified by the accessibility requirements.

Keywords: requirements elicitation, accessibility requirements, accessibility guideline.

1 Introduction

Accessibility requirements for software have been increasing. For examples, Section 508 [1] in the United States and JIS X 8341 [2] in Japan have been implemented. Accessibility means that various people, including the challenged and elderly people, can use software and websites easily. User interfaces are especially important because people directly interact with them. Most software uses GUIs (Graphical User Interfaces), which people operate visually, but people with visual limitations have difficulty and require specific support tools and devices. Thus, software must be developed considering the end users' characteristics (e.g., disabilities, age, etc.) to design accessible software and the proper support tools.

Many guidelines have been developed to realize accessibility (e.g., Web Content Accessibility Guidelines 2.0 (WCAG 2.0) [3] and JIS X 8341-3). In addition, companies and organizations have developed their own accessibility guidelines where problematic situations and their resolutions are described in detail. Consequently, the vast number of elements in guidelines is an issue. When guidelines are applied to software, the applicability of each element must be confirmed.

D. Zowghi and Z. Jin (Eds.): APRES 2014, CCIS 432, pp. 210–223, 2014.

This confirmation process is necessary for general use software, but not all software is intended for general use (e.g., for institutional use). In these cases, some elements are not required based on the software's purpose and the end users' characteristics.

In addition, some guideline elements can be applied in the late software development phase, whereas some cannot. To resolve these problems, it is necessary to elicit accessibility requirements in the early development phase and determine which elements within the guidelines to apply.

Moreover, it is possible that the accessibility requirements are not appropriately elicited. Because end users are often unable to recognize difficulties when using software, they cannot identify current disabilities. Hence, accessibility requirements should be elicited while considering these people.

In this research, we propose a method to analyze end users' characteristics and elicit accessibility requirements in the early development phase. Concretely, checklists are used to analyze the operational situations and problems of end users. Based on this analysis, guideline elements are elicited as accessibility requirements with priorities. The proposed method can elicit end users' accessibility requirements directly and appropriately, and implement the all required accessibility requirements while simultaneously reducing costs and burden on software developers.

This paper is organized as follows. Section 2 describes related works, while the features of the proposed method is described in section 3. Section 4 shows the support strategies to challenged users. Section 5 provides a detailed description of this proposed method. Section 6 describes the simulation of this proposed method, and section 7 concludes our paper.

2 Related Works

Requirements can be classified into functional and non-functional requirements. Functional requirements describe how to process inputs, while non-functional requirements define attributes that software should satisfy (e.g., security, reliability, usability, etc.). There are various types of non-functional requirements, and accessibility requirements are non-functional requirements. Strategies of eliciting non-functional requirements differ from the types. Although many studies have examined security requirements (a type of non-functional requirements) [4][5], few have focused on accessibility requirements.

Baguma et al. have proposed a method to integrate accessibility requirements with functional requirements [6]. Functional requirements and non-functional requirements, including accessibility requirements, have been analyzed by User Centered Design (UCD) techniques [7]. In this analysis, user group profiles, personas, and scenarios are documented. User group profiles are characteristics of users. Personas are concrete examples of typical users, while scenarios describe how personas use the products. Then, accessibility requirements (AR) graphs are described using the Non-Functional Requirements (NFR) goal graphs approach [8]. Finally, use case diagrams, including accessibility requirements, are

described. Because this method describes accessibility requirements and functional requirements in the same diagrams, their relationships can be clarified. However, this approach does not consider the priorities of accessibility requirements.

AccessOnto is an ontology-based tool kit for accessibility requirements [9] that provides a repository of accessibility guidelines and a specification language to describe accessibility requirements in user requirements documents. Items related to user interfaces, such as user agents, languages, guidelines, checkpoints, and user characteristics, are defined. Although AccessOnto easily describes requirements specifications, including accessibility requirements, accessibility requirements are not elicited.

Minon et al. have proposed a method to integrate accessibility requirements into a user interface development method [10]. Accessibility requirements, which are elicited using accessibility guidelines and standards, such as WCAG [3] and ISO 9241-171:2008 [11], are described as task models of UsiXML (USer Interface eXtensible Markup Language) [12] of UIDL (User Interface Description Language) [13]. The task models are transformed into an Abstract User Interface (AUI) model of UsiXML that includes accessibility requirements. Because this method has high affinity with UIDL, accessibility requirements are easily integrated into user interface development methods. However, accessibility requirements are elicited using existing accessibility guidelines and standards. Because situations and levels of disabilities vary by end user, costs and burdens hinder software development. Thus, it is necessary to prioritize requirements after eliciting them from end users.

3 Features of the Proposed Method

Elicitation of Accessibility Requirements in the Early Software Development Phase

Regardless of their importance, some requirements elicited in the late software development phase cannot be realized. Although accessibility requirements to change color and font size using GUIs may be reasonable, preparing specific functions and support tools may not. Thus, it is necessary to elicit specific accessibility requirements in the early software development phase.

We assume that this proposed method is used in the requirements elicitation phase, which is part of the early software development phase. Hence, all requirements should be implementable.

Realization of Software Based on Detailed End Users' Characteristics

Because situations and levels of disabilities vary from person to person, the requirements differ. Software must be developed based on end users' situations and levels of disabilities.

In the proposed method, the situations and levels of disabilities are analyzed in detail. Then the accessibility requirements are prioritized, allowing software to be more appropriately developed for end users.

Reduction of Costs and Burdens

Eliciting requirements in the late software development phase causes iterations and the development returns to the early phase. In addition, applying accessibility guidelines requires software developers to confirm the numerous elements in the guidelines, which is expensive and burdensome.

In this proposed method, accessibility requirements can be elicited in the early software development phase. Because end users' characteristics are analyzed in detail, the scope of accessibility requirements can be tailored to the end users. These features can reduce the cost of burden on software developers.

4 Support Strategies to Challenged People

Many guidelines have been developed to make software accessible, but each person has a different level and situation of disability. When challenged people use software, they often use specific support tools, devices, and functions, which correspond to their situation. The availability of these tools and functions may be included in accessibility requirements.

4.1 Disabilities

Typical support strategies depend on the type of disability. Below are typical disabilities and their support strategies.

Blind Users. This is one kind of visual impairment in which people have completely lost their eyesight. Part of blind users can feel light, but cannot identify anything with their eyes.

Two types of basic support exist for blind users. One is that all contents should be described with text that is compatible with screen readers and braille displays. Screen readers are software to read texts on display, while braille displays convert texts into braille. Most blind people use screen readers. The other is that all contents should be operated with a keyboard. Blind users cannot identify where controls (e.g., links or buttons) are on display, so it is difficult to use a mouse.

Users with Weak Eyesight. Weak eyesight is another kind of visual impairment. In this disability, eyesight is barely corrected even if users wear glasses or contact lenses. Users can roughly identify things with their eyes, but their vision becomes inaccurate beyond a certain level.

There are two basic supports for users with weak eyesight. One is that small text and icons should not be used. These users can identify large texts and icons. The other is a function to adjust text size. Because the level of eyesight weakness varies by user, the text size must be adjustable to suit individual needs.

Users with Color-Vision Impairments. Color-vision is a third type of visual impairment. In this disability, users misidentify some colors (e.g., red, green, and blue). For example, users with this disability often have difficulty distinguishing between red and green.

There are two types of basic support. One is that contents should not be represented solely by color. For example, a description like "something is represented with red" should not be used. The other is a function to adjust the colors. Because color limitations differ according to the user, the color must be adjustable to suit individual needs.

Users with Hearing Impairments. In this disability, users have difficulty or cannot hear voices or sounds. For these users, the basic support is that all contents of software should be represented as text that they can read with their eyes.

Physically Disabled Users. In this disability, users have limited control of their hands, arms, and/or fingers. These users often use various support tools and devices, such as a software keyboard or a track ball. Basic support is that the software must be compatible with these support tools and devices.

Elderly Users. Although elderly users are not considered as challenged users, they have similar difficulties with challenged users by aging. Thus, it is possible to support elderly users with similar strategies for challenged users.

4.2 Accessibility Guidelines

Accessibility guidelines include detailed descriptions and resolutions of common difficulties of challenged users. In the guidelines, each element describes a specific issue or its resolution. Elements often include implementation. Because many guidelines have been developed (e.g., WCAG 2.0 [3] and JIS X 8341-3 [2]), elements must be prioritized.

5 Elicitation of Accessibility Requirements

In the proposed method, accessibility guidelines are prepared by initially employing checklists to analyze end users' characteristics. Associations between the questions in these checklists and elements in guidelines are specified. Then based on the end users' responses, the levels and situations of difficulties are analyzed, and the strength of relevance between difficulties of end users and guideline elements are calculated as numerical values. Finally, the necessary guideline elements are extracted and prioritized as accessibility requirements. Figure 1 shows the architecture of the proposed method.

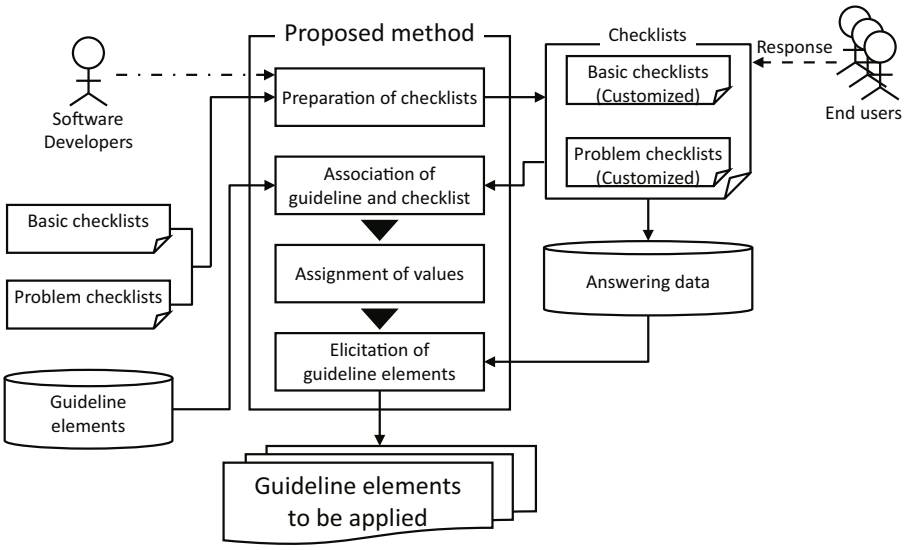

Fig. 1. Architecture of the proposed method

5.1 Preparation of Checklists and Accessibility Guidelines

Checklists. There are two types of checklists: basic and problem. Table 1 shows an example of a basic checklist.

Table 1. Example of basic checklists

No.	Question	Selections for response			
Situations of disabilities and abilities					
A-1	What is your eyesight?	Over 1.0	Over 0.3 and under 1.0	Under 0.3	None
A-2	Can you use braille?	No	Hardly	Almost can	Can
Usages of support tools and devices					
B-1	Do you use a screen reader?	No	Sometimes	Often	Always
B-2	Do you use a braille display?	No	Sometimes	Often	Always
B-3	Do you use a voice input system?	No	Sometimes	Often	Always
B-4	Do you use software to adjust view size?	No	Sometimes	Often	Always

Basic checklists include questions about the following:

- Situations of end users' disabilities
- Computer environments and configurations that end users use
- Support tools and devices that end users use, etc.

Table 2. Example of problem checklists

No.	Question	Selections of response			
Problems of current usages					
1	Do you have any problems using computers?			Yes -> to 2.	No
2	What kinds of problems do you experience?	Difficult to watch display -> to 3.	Difficult to hear voices and sounds -> to 4.	Difficult to operate keyboards and mouse -> to 5.	Other -> 6.
3	Please response the following questions about difficulties of watching displays.				
	X-1	Font size configurations of the display	Too big	Just right	small	Too small
	X-1-1	If you responded “a little small” or “too small”, please indicate the display size, resolution, and font size that you normally use.			
	X-2	Color usages	Excellent	Good	Poor	Very poor
	X-2-1	If you responded “poor” or “very poor”, please indicate the background and foreground colors that you feel difficult to watch.			
	X-3	Vision of display	Very clear	Clear	Slightly blurry	Blurry

Problem checklists include questions about current usage problems. Table 2 shows an example of problem checklists.

Questions of checklists are associated to guideline elements, and there are various accessibility guidelines by governments, companies, and organizations. Additionally, important guideline elements may be different from software characteristics. Thus, templates for these checklists based on the policies of governments, companies, and organizations as well as currently realized support tools and devices were prepared. Because the actual templates can be customized, elements of various guidelines can be associated to checklist questions, and checklists can reflect the intended software characteristics.

Table 3. Examples of accessibility guideline elements

No.	Guideline element
Usages of support tools and devices	
1-1	All operations must performed by a keyboard.
1-2	All contents must be able to be read by a screen reader.
1-3	All contents must be able to be shown by a braille display.
1-4	Software to adjust view size must be available.
Color usages	
2-1	Contents must not be denoted solely by only colors.
2-2	Brightness contrasts of background and foreground colors must be sufficient.
2-3	System configurations must be applicable (e.g., color and font).

Accessibility Guidelines. In the proposed method, elements in the guidelines are elicited as accessibility requirements. Thus, the checklists and accessibility guidelines must be associated. We prepared our checklists based on the existing guidelines, such as WCAG 2.0 [3] and JIS X 8341-3 [2]. Table 3 shows examples of the guideline elements. If necessary, extra elements can be added.

5.2 Association of Guideline Elements and Checklist Questions

Questions in a checklist are associated with specific guideline elements. The association strength is identified as "Strong", "Medium", "Weak", and "None". The results indicate how each element should be realized during software development.

Table 4 shows an example of associations between checklist questions and guideline elements. Table 5 also shows an example of the association strengths.

The question numbers (e.g., "A-1" and "X-1") are from Tables 1 and 2, while the element numbers (e.g., "1-1" and "2-1") are from Table 3. Both the associations and association strengths can be customized.

Table 4. Example of associations between questions and elements

No.	Situations of disabilities
	Basic checklists
A-1	Determination of blind and levels of weak eyesight
A-2	Determination of braille display usages
B-1	Determination of blind, weak eyesight
B-2	Determination of blind, weak eyesight
	Problem checklists
X-1	Determination of levels of weak eyesight
X-2	Determination of types and levels of color-impairments
X-3	Determination of levels of weak eyesight

Table 5. An example of strength of associations

	A-1	A-2	B-1	B-2	B-3	B-4	X-1	X-2	X-3
1-1	Strong	Strong	Strong	Strong	Weak	Medium	Weak	Weak	Weak
1-2	Strong	Strong	Strong	Weak	Weak	Medium	Medium	Weak	Medium
1-3	Strong	Strong	Strong	Weak	Weak	Medium	Medium	Strong	Medium
1-4	Strong	Strong	Medium	Medium	Weak	Strong	Strong	Weak	Strong
2-1	Strong	Weak	Medium	Weak	Weak	Medium	Medium	Strong	Medium
2-2	Strong	Weak	Medium	Weak	Weak	Medium	Medium	Strong	Medium
2-3	Strong	Weak	Medium	Weak	Weak	Strong	Strong	Strong	Strong

5.3 Relevance of Assigned Values between Guideline Elements and Users' Responses

To analyze the situations and levels of users' difficulties, the strength between difficulties of end users and guideline elements must be calculated. Thus, the numerical values are assigned to the users' responses to checklists. Currently each response is assigned a value from 0 to 3, where 0 indicates that the support described by the question does not need to be considered, while 3 indicates that the support must be fully considered. Table 6 shows examples of assigned numerical values. The question numbers (e.g., "A-1" and "X-1") are from Tables 1 and 2.

Table 6. Example of numerical value assignments

Numerical values for responses		**0**	**1**	**2**	**3**
Basic checklists					
A-1	What is your eyesight?	Over 1.0	Below 1.0 but above 0.3	Under 0.1	None
A-2	Can you use braille?	No	Hardly	Somewhat	Can
B-1	Do you use a screen reader?	No	Sometimes	Often	Always
B-2	Do you use a braille display?	No	Sometimes	Often	Always
B-3	Do you use a voice input system?	No	Sometimes	Often	Always
B-4	Do you use software of adjusting view size?	No	Sometimes	Often	Always
Problem checklists					
X-1	Font size on display	Too big	Just right	Small	Too small
X-2	Color usages	Excellent	Good	Poor	Very poor
X-3	Vision of display	Very clear	Clear	Slightly Blurry	Blurry

In addition, numerical values are assigned to the levels of strength in Table 5. These values are used to calculate the necessity of applying guideline elements to the target software. Currently, 0, 1, 2, and 3 are assigned as "Strong", "Medium", "Weak", and "None", respectively. Table 7 shows the numerical value assignments to Table 5.

5.4 Elicitation of Guideline Elements as Accessibility Requirements

Based on the numerical values in Tables 6 and 7, guideline elements to be applied to the target software are specified and prioritized. The priorities are calculated in three steps.

Step 1. Based on the association strengths between basic checklists and guideline elements, the importance of each guideline element ($impB_{i-j,m-n}$) is calculated using the end users' responses via formula (1). $i - j$ indicates the number

Table 7. Example of association strenghs

	A-1	A-2	B-1	B-2	B-3	B-4	X-1	X-2	X-3
1-1	3	3	3	3	1	2	1	1	1
1-2	3	3	3	1	1	2	2	1	2
1-3	3	3	3	1	1	2	2	3	2
1-4	3	3	2	2	1	3	3	1	3
2-1	3	1	2	1	1	2	2	3	2
2-2	3	1	2	1	1	2	2	3	2
2-3	3	1	2	1	1	3	3	3	3

of a guideline element (e.g., "1-1" in Table 3), whereas $m-n$ indicates the number of a question (e.g., "A-1" in Table 1). $S_{i-j,m-n}$ indicates the value of association strength between guideline element $i-j$ and question $m-n$ in the basic checklists, and R_{m-n} indicates the value of users' responses to the question $m-n$.

$$impB_{i-j,m-n} = S_{i-j,m-n} \times R_{m-n} \tag{1}$$

Step 2. Similar to Step 1, the importance of each guideline element ($impP_{i-j,p-q}$) is calculated using the end users' responses via formula (2). $p-q$ indicates the number of a question (e.g., "X-1" in Table 2). $S_{i-j,p-q}$ indicates the value of association strength between guideline element $i-j$ and question $p-q$ in the problem checklists, while R_{p-q} indicates the value of users' responses to the question $p-q$.

$$impP_{i-j,p-q} = S_{i-j,p-q} \times R_{p-q} \tag{2}$$

Step 3. Finally, the priority of guideline elements is calculated by integrating the results of Step 1 and Step 2. Guideline elements with a higher priority can be specified as accessibility requirements of the target software. The priority is calculated using formula (3). $priority_{i-j}$ indicates the priority value of a guideline element $i-j$. $M-N$ and $P-Q$ indicate the maximum question numbers of basic and problem checklists, respectively.

$$priority_{i-j} = \sum_{m-n=A-1}^{M-N} impB_{i-j,m-n} + \sum_{p-q=X-1}^{P-Q} impP_{i-j,p-q} \tag{3}$$

After the priority values are calculated, the specified guideline elements are validated by generating prototypes. Previously, we have proposed methods to generate GUI prototypes from scenarios [14][15]. Then end users validate the methods to implement the guideline elements.

6 Simulation

A simulation was conducted to confirm whether the specified guideline elements are valid as accessibility requirements. Below is a summary of the simulated end user's situation. Table 8 shows select responses to the checklist where the question numbers (e.g., "A-1" and "X-1") are from Tables 1 and 2.

Basic checklist

- Have weak eyesight
- Use zoom software
- Sometimes use screen readers

Problem checklist

- Sometimes difficultly recognizing the display colors

Table 8. Select responses of the simulated end user

Basic checklists						
	A-1	A-2	B-1	B-2	B-3	B-4
Response	Under 0.3	No	Sometimes	No	Sometimes	Always
Value	2	0	1	0	1	3
problem checklists						
	X-1		X-2		X-3	
Response	Small		Poor		Slightly blurry	
Value	2		2		2	

Based on responses in Table 8, $impB_{i-j,m-n}$ and $impP_{i-j,p-q}$ are calculated using formulas (1) and (2). Tables 9 and 10 show the results.

Table 9. Calculation of $impB_{i-j,m-n}$

Guideline element No.	Question No.					
	A-1	A-2	B-1	B-2	B-3	B-4
1-1	6	0	3	0	1	6
1-2	6	0	3	0	1	6
1-3	6	0	3	0	1	6
1-4	6	0	2	0	2	9
2-1	6	0	2	0	1	6
2-2	6	0	2	0	1	6
2-3	6	0	2	0	1	9

Table 10. Calculation of $impP_{i-j,p-q}$

Guideline element No.	Question No.		
	X-1	X-2	X-3
1-1	2	2	2
1-2	4	2	4
1-3	4	2	4
1-4	6	2	6
2-1	4	6	4
2-2	4	6	4
2-3	6	6	6

Using the values in these tables, the priority values of the guideline elements are calculated by formula (3). Figure 2 shows the priority values of the guideline elements.

Using the values in these tables, the priority values of guideline elements were calculated by the formula (3) in 5.4. Figure 2 shows the priority values of guideline elements.

In this simulation, the average value of priority values of all guideline elements were calculated. The average value was 28.71 and shown in Fig. 2. Guideline elements were classified into two groups of higher and lower values than the average value. According to this classification, the following guideline elements have high priority values.

1-4: Software to adjust view size must be available.
2-1: Contents must not be denoted solely by colors.
2-2: Brightness contrasts of background and foreground colors must be sufficient.
2-3: System configurations must be applicable (e.g., color and font).

According to the checklist responses, the simulated end user has weak eyesight, uses zoom software, and occasionally experiences difficulty recognizing the display colors. Thus, the simulation specified appropriate guideline elements.

The following guideline elements have low priority values.

1-1: All operations must performed by a keyboard.
1-2: All contents must be able to be read by a screen reader.
1-3: All contents must be able to be shown by a braille display.

The simulated end user does not operate only by a keyboard and does not use braille displays. Thus, the priority values of guideline elements 1-1 and 1-3 are appropriately calculated. Although the simulated end user occasionally used a screen reader, the priority value of guideline element 1-2 is low due to the end user's response about the frequency of using a screen reader. The checklist questions result in a subjective gauge. However to specify appropriately guideline elements for the target software, the checklist questions must be improved so that

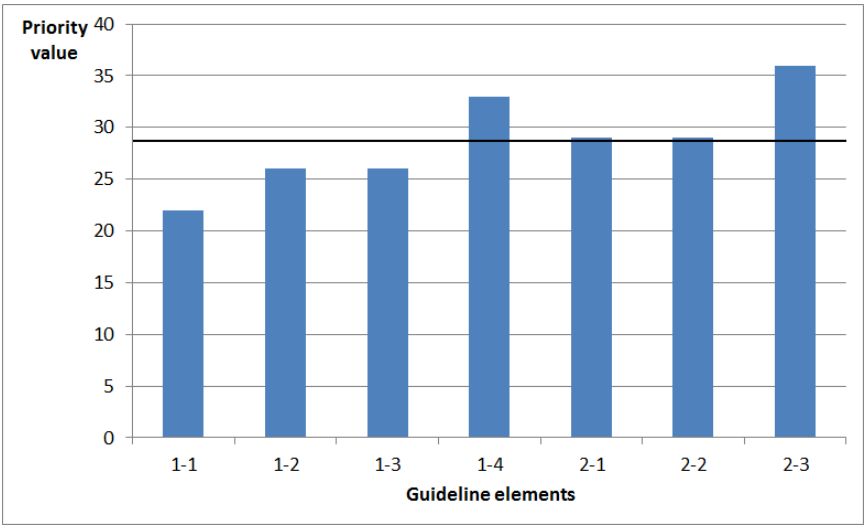

Fig. 2. Simulation results

end users' responses provide an objective gauge. In addition, the validity of the specified guidelines elements must be confirmed by end users.

However, according to this simulation results, almost all appropriate guideline elements are specified. Thus, this proposed method can appropriately specify the accessibility requirements.

7 Conclusion

In this paper, a method to elicit accessibility requirements is proposed by associating end users' disability situations and guideline elements. The calculated priority values are used to determine the priorities of the guideline elements. Although it is difficult to apply all guideline elements to the target software, the proposed method elicits the accessibility requirements in the early development phase and allows software to be appropriately developed based on end users' characteristics, reducing the cost and burden of software development. In addition, the appropriateness of the specified guideline elements is confirmed.

Future work includes:

- Confirming the numerical value appropriateness by simulating various end users' characteristics
- Designing checklist questions that objectively gauge end users' responses
- Evaluating the proposed method with actual challenged users
- Addressing challenges associated with the implemented guideline elements and automatically applying GUI prototypes

References

1. Section 508: http://www.section508.gov/u
2. Japanese Industrial Standards Committee: Guidelines for older persons and persons with disabilities-Information and communications equipment, software and services-Part 3: Web content, JIS X 8341-3 (2010)
3. Web Content Accessibility Guidelines (WCAG) 2.0: http://www.w3.org/TR/WCAG20/
4. Alexander, I.: Misuse Cases: Use Cases with Hostile Intent. IEEE Software 20(1) (2003)
5. Beckers, K., Heisel, M., Cote, I., Goeke, L., Guler, S.: Structured Pattern-Based Security Requirements Elicitation for Clouds. In: Proc. of 2013 Eighth International Conference on Availability, Reliability and Security, ARES (2013)
6. Baguma, R., Stone, R.G., Lubega, J.T., van der Weide, T.P.: Integrating Accessibility and Functional Requirements. In: Stephanidis, C. (ed.) UAHCI 2009, Part III. LNCS, vol. 5616, pp. 635–644. Springer, Heidelberg (2009)
7. Henry, S.L.: Just Ask: Integrating Accessibility Throughout Design, Lulu. Com (2007)
8. Cysneiros, L.M., Leite, J.C.S.P.: Nonfunctional Requirements: From Elicitation to Conceptual Models. IEEE Transactions on Software Engineering 30(5) (2004)
9. Masuwa-Morgan, K.R.: Introducing AccessOnto: Ontology for Accessibility Requirements Specification. In: Proc. of First International Workshop on using Ontologies in Interactive Systems, ONTORACT 2008 (2008)
10. Minona, R., Morenob, L., Martinezb, P., Abascal, J.: An approach to the integration of accessibility requirements into a user interface development method. Science of Computer Programming (April 29, 2013)
11. International Organization for Standardization: Ergonomics of human-system interaction – Part 171: Guidance on software accessibility (ISO 9241-171: 2008) (2008)
12. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Lopez-Jaquero, V.: USIXML: A language supporting multi-path development of user interfaces. In: Proc. of the 2004 International Conference on Engineering Human Computer Interaction and Interactive Systems (2004)
13. Guerrero-Garcia, J., Gonzalez-Calleros, J.M., Vanderdonckt, J., Muoz-Arteaga, J.: A Theoretical Survey of User Interface Description Languages: Preliminary Results. In: Proc. of Web Congress, LA-WEB 2009, Latin American (2009)
14. Shirogane, J., Fukazawa, Y.: A Method of Scenario-based GUI Prototype Generation and its Evaluation. ACIS International Journal of Computer & Information Science (IJCIS) 4(1) (2003)
15. Shirogane, J., Shibata, H., Iwata, H., Fukazawa, Y.: GUI Prototype Generation from Scenarios in the Requirements Elicitation Phase. In: Proc. of the 13th IASTED International Conference on Software Engineering, SE 2014 (2014)