Towards User-friendly E-Government Solutions: Usability Evaluation of Austrian Smart-Card Integration Techniques

Thomas Zefferer and Vesna Krnjic

E-Government Innovation Center Austria,
Inffeldgasse 16a, 8010 Graz, Austria
{thomas.zefferer,vesna.krnjic}@egiz.gv.at
http://www.egiz.gv.at

Abstract. Security and usability are key requirements of e-Government solutions. Security requirements are often met by reliance on smart card technology. In Austria, the open source components *MOCCA Local* and *MOCCA Online* facilitate the integration of smart cards into national e-Government applications. While MOCCA Local and MOCCA Online guarantee security, their capability to meet given usability requirements has not been assessed so far.

To bridge this gap, the usability of MOCCA Local and MOCCA Online has been evaluated by means of a usability test. The obtained results show that MOCCA Local and MOCCA Online basically meet given usability requirements. Still, some minor issues have been identified that threaten to reduce the usability of these components.

In this paper we introduce the basic architecture of MOCCA Local and MOCCA Online and present results of the conducted usability test. Based on these results we identify existing usability issues and derive possible improvements.

Keywords: E-Government, Smart cards, Usability, MOCCA

1 Introduction

Secure authentication of citizens and creation of electronic signatures are common requirements of e-Government applications. These requirements are perfectly met by smart cards as they support the secure storage of authentication information and can be used as *Secure Signature Creation Devices (SSCD)*. This way, smart cards fulfill the requirements of qualified electronic signatures according to the Directive 1999/93/EC of the European Parliament and of the Council on a Community framework for electronic signatures [1].

Besides their security enhancing features, their wide acceptance is another key advantage of smart cards. Bank institutes have recognized the potential of smart card technology early and nowadays provide customers with multi-functional bank account cards. In various countries, smart cards are also used in the health sector. For instance, in Austria citizens are supplied with health insurance cards that facilitate use and charging of public health services. Due to their various fields of application, smart cards can nowadays be regarded as well accepted and approved technology.

Its wide acceptance and its capability to fulfill given security requirements make smart card technology perfectly suitable for e-Government solutions. It is thus less astonishing that e-Government strategies of various European countries foresee the use of smart cards. In Austria, smart cards are used in e-Government applications as they are able to meet given legal requirements such as the Austrian Signature Act [2] and the Austrian e-Government Act [3]. Also in numerous other European countries smart cards are an integral part of current e-Government solutions. A comprehensive overview of national eID solutions is given in [13].

Unfortunately, the use of smart cards in e-Government applications also raises various challenges for citizens, governments, and application developers. For governments, the implementation of appropriate card roll-out and personalization processes is a serious challenge as this usually involves significant organizational and financial efforts. For citizens, the need for appropriate card reader devices often represents a serious barrier as off-the-shelf PCs and laptops do usually not support this functionality by default. The integration of smart card technology into e-Government applications also raises several technological challenges for application developers. For instance, in[4], Orthacker et al. have discussed accessibility challenges that arise with the use of smart cards in e-Government applications. So far, less attention has actually been paid to usability aspects of smart card based solutions. Although smart card vendors often advertise the usability of their products, there is still a lack of scientific research on this topic.

Nevertheless, usability is a crucial factor that heavily influences user acceptance. Since e-Government applications allow for more efficient administrative and governmental proceedings, a high degree of user acceptance is desirable to improve efficiency and to save costs. The requirement for user acceptance directly leads to the demand for an appropriate level of usability in e-Government solutions. Significant effort has already been made to optimize the usability of Web based e-Government solutions. Related work has been discussed in [5] and [6]. At the same time, less attention has been paid to the usability of different approaches to integrate smart cards into these applications. We filled this gap by conducting a comprehensive usability analysis of established smart card integration methods of the Austrian e-Government infrastructure. In this paper we present results of the conducted usability analysis and propose several enhancements that can help to improve the usability of existing smart card integration approaches.

This paper is structured as follows. In Section 2 we discuss relevant concepts of the Austrian e-Government infrastructure and introduce MOCCA Local and MOCCA Online in more detail. We provide details of the methodology that has been followed to evaluate the usability of these components in Section 3. Results of the conducted usability test are presented in Section 4 and discussed in Section 5. Finally, conclusions are drawn.

2 Smart-card Integration: The Austrian Approach

The integration of smart cards is a serious technological challenge for developers of e-Government applications. In this section we introduce different approaches that are followed in Austria to overcome this challenge. Usability aspects of these approaches will be discussed later in this paper.

Austrian e-Government solutions are based on the so called *Citizen Card* concept. The Citizen Card represents a token that allows citizens to authenticate at remote services and to create qualified electronic signatures according to the EU Signature Directive. Although the term Citizen Card might suggest the use of smart cards, the Citizen Card concept is actually technology-independent and can also be applied to e.g. mobile phones[1]. Despite of its technology-neutral nature, smart cards still play an important role in the Austrian Citizen Card concept. Currently, Citizen Card implementations that rely on health insurance cards, bank account cards, or special signature cards are available in Austria.

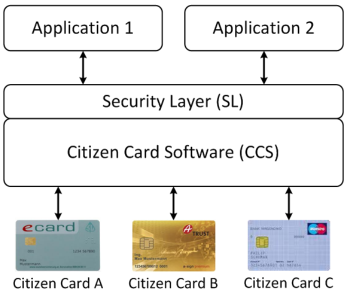

Fig. 1. The Security Layer provides e-Government applications a common interface to different Citizen Card implementations.

The support of different cards is beneficial for citizens as they can use their preferred card type. However, this flexibility significantly complicates the integration of Citizen Card functionality into e-Government applications, as support for each card type has to be implemented separately. With a growing number of Citizen Card implementations (i.e. smart card types), development of new and maintenance of existing applications increase complexity and cause additional costs.

To overcome this problem, the Austrian Citizen Card concept follows a middleware approach and defines the so called *Security Layer (SL)* interface as shown in Fig. 1. The Security Layer has been introduced in [7] and represents a common XML based interface between e-Government applications and different Citizen Card implementations. The Security Layer interface is implemented by the so called *Citizen Card Software (CCS)*. This software provides access to all Citizen

[1] A Citizen Card implementation using mobile phones has been introduced in [8].

Card implementations and makes their functionality available to e-Government applications through the common SL interface. This way, application developers are released from integrating and maintaining different smart card types, as this task is outsourced to and implemented by the CCS.

The middleware approach shown in Fig. 2 raises the question about possible implementation variants for the CCS. Following the most obvious approach, the CCS is often implemented as software running on the user's local system (cf. Fig. 2). This way, the CCS has access to locally connected smart cards through the system's PC/SC interface. Following this approach, the SL interface is provided by the CCS through a local network port. This way, also Web based e-Government applications can easily access the SL interface through the user's Web browser. Since all specifications of the SL interface are open and publicly available, there are already various CCS implementations from different vendors available on the market. Some of these solutions such as the A-Trust a-sign client [9] are for free, others charge a license fee. The only open source CCS available in Austria so far is called *MOCCA Local* and has been introduced in [10]. MOCCA Local represents one of the main outcomes of the MOCCA (Modular Open Citizen Card Architecture) project [11] that aims to provide Austrian citizens with Java based open source CCS implementations. Fig. 3 shows MOCCA Local's GUI that allows users to review data to be signed and to confirm it by pressing the *Sign* button.

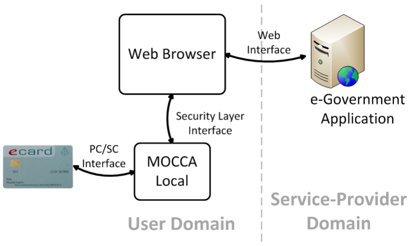

Fig. 2. MOCCA Local is a software that runs on the user's local system and acts as intermediary between the Web browser and locally connected smart cards.

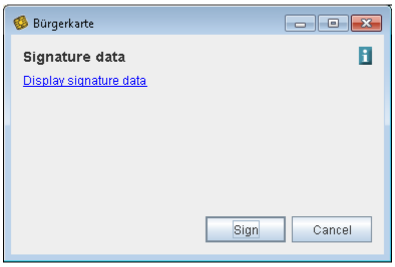

Fig. 3. The GUI of MOCCA Local allows users to review data to be signed and to start the signature creation process.

All solutions mentioned above including MOCCA Local follow the approach shown in Fig. 2 and make use of software running on the user's local system. While this approach works fine from a functional point of view, it raises several problems for citizens. Especially inexperienced users sometimes have problems to carry out software installations on their own. In some cases, users do not even have the privileges to install software on the computer they are currently using. To overcome these issues, a minimal footprint CCS implementation has been developed in the course of the MOCCA project. According to this minimal footprint approach, users are not required to install software on their local system. To underline the installation-free nature of this approach, this solution has been named *MOCCA Online*. Similar to MOCCA Local, MOCCA Online is based on the Java framework. Java has been chosen as underlying technology for all MOCCA components in order to achieve platform independence. Reliance on Java requires a current Java Runtime Environment (JRE) to be installed on the user's local system.

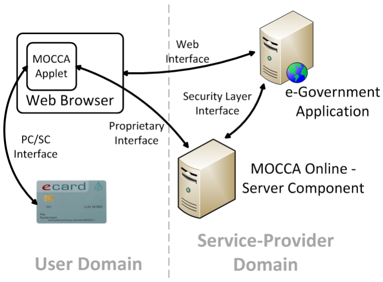

Fig. 4. MOCCA Online follows a distributed approach and consists of a server component and a Java Applet.

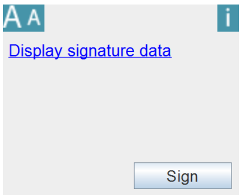

Fig. 5. The GUI provided by MOCCA Online's Java Applet allows users to review data to be signed and to start the signature creation process.

MOCCA Online has been introduced and discussed by Centner et al. in [10]. The basic architecture of MOCCA Online is shown in Fig. 4. The main idea behind MOCCA Online is to split the entire CCS functionality into two core components. The first component runs on a server and implements the SL interface. E-Government applications communicate with this server component to access smart card functionality. A Java Applet represents the second core component of MOCCA Online. The Java Applet runs in the user's Web browser and implements smart card communication based on the PC/SC protocol. Furthermore, the Applet provides a GUI through which users can review and confirm the data to be signed. The Applet's GUI is shown in Fig. 5 and has been designed similar to the GUI provided by MOCCA Local (cf. Fig. 3). This way, a similar user experience is achieved irrespective of the used MOCCA variant. Typically, the Applet is integrated into Web based e-Government applications by means of a HTML IFRAME tag. The two components of MOCCA Online, i.e. the server component and the Java Applet, communicate with each other over a proprietary interface.

Currently, MOCCA Local and MOCCA Online are among the predominating CCS implementations in Austria. During the past few years, much effort has been invested to assure and improve the security of these components. Less attention has been paid to usability aspects so far. To bridge this gap, we have conducted an extensive usability analysis of MOCCA Local and MOCCA Online in order to identify usability problems and to further improve the user acceptance of these components. In the following section we discuss the applied methodology of the conducted usability test.

3 Methodology

Approved usability evaluation methods have been applied to analyze the usability of MOCCA Local and MOCCA Online. This section defines a set of research questions and discusses the followed methodology to answer them by means of the conducted usability test.

3.1 Research Questions

The following research questions have been defined in order to cover all relevant usability aspects of the evaluated components.

Q1 Does reliance on Java technology cause additional usability problems?
Q2 What are the main usability problems that have been encountered during the installation of MOCCA Local and which user groups are especially affected?
Q3 Once MOCCA Local is correctly installed, what are the main usability problems that have been encountered during the usage of MOCCA Local and which user groups are especially affected?
Q4 What are the main usability problems that have been encountered during the usage of MOCCA Online and which user groups are especially affected?
Q5 Which MOCCA variant appears more secure and trustworthy to users and are there significant differences between different user groups?

3.2 Test Method and Setup

To find answers to the predefined research questions, a thinking-aloud test has been conducted. A Thinking-aloud test is an approved method to evaluate the usability of software products or websites and has been discussed by Nielsen in [12]. In a thinking-aloud test, test users are asked to carry out a set of well-defined tasks with the software to be evaluated and to articulate their thoughts during the test run. This way, users' interactions with the software under test can be observed and valuable user feedback can be collected. Since the users' emotional state can also be of interest, test users are usually recorded with a camera during the test.

Thinking-aloud tests typically consist of two phases. During the *test phase*, test users are asked to carry out predefined tasks. In the subsequent *analysis phase*, data recorded and collected during the test phase is analyzed in order to identify common usability problems and to find answers to predefined research questions.

We have used special software during both phases. During the test phase, the used software assisted in recording and collecting relevant data by tracking test users' mouse movements and keyboard inputs. Furthermore, the used software has automatically related additionally recorded video and sound data to the tracked user inputs. During the test phase, additional user feedback has been collected by means of different questionnaires and a conclusive interview. Recorded user sessions and collected user feedback have been analyzed in the subsequent analysis phase. Again, the used software has facilitated an efficient analysis of the collected data sets and the application of statistical methods.

All tests have been carried out on an off-the-shelf PC with Microsoft Windows 7 operating system. Test users were asked to use the Microsoft Internet Explorer 8 Web browser. We have chosen this system configuration as it represents a common OS/Web browser combination. To facilitate a systematic analysis of the collected data and to ease comparisons between different test users, we did not give test users the opportunity to choose their preferred operating system or Web browser configuration.

3.3 Test Users and User Groups

In total, 20 test users participated in the conducted usability test. In order to achieve convincing and sound results, test users have been chosen in a way that they approximately form a representative sample of the Austrian society. All test users have been asked to carry out the following three tasks using their personal smart card based Citizen Card.

T1 Install MOCCA Local using a provided Java Web Start based installing routine.

T2 Use MOCCA Local to carry out a given e-Government procedure including identification and signature creation.

T3 Use MOCCA Online to carry out a given e-Government procedure including identification and signature creation.

To avoid the influencing of results by learning effects, test users were split into two groups. Group A was asked to carry out the tasks in the order given above and hence started with installation and usage of MOCCA Local. Contrary, Group B was asked to carry out task T3 first, followed by task T1 and task T2. Thus, test users of Group B started with an evaluation of MOCCA Online first.

Both MOCCA Local and MOCCA Online require a current Java Runtime Environment (JRE) to be installed on the client system. As we were also interested in the usability of the Java installation process and its integration in the evaluated MOCCA components, the test system was provided without an installed JRE. Test users were requested to install the required JRE during the test. Depending on their assigned group, Java had to be installed either during the installation of MOCCA Local or during the first usage of MOCCA Online. This way, we were able to compare the usability of the Java installation processes of MOCCA Local and MOCCA Online.

The assignment of test users to Group A and Group B was completely random. Additionally, all test users have been subdivided into different user groups according to different characteristics. This way, we were able to assess the impact of given usability flaws on different user groups. The following table summarizes user groups that have been analyzed separately. Details of the obtained results are discussed in the next section.

Table 1. Test users have been classified according to four different characteristics.

ID	Description	Users
Group ALL	This group comprises all test users.	20
Group A	Test users of this group started with the evaluation of MOCCA Local.	10
Group B	Test users of this group started with the evaluation of MOCCA Online.	10
Group 30+	Test users of this group are more than 30 years old.	8
Group 30-	Test users of this group are 30 or less years old.	12
Group U	Test users of this group have a university degree.	12
Group NU	Test users of this group have no university degree.	8
Group T	Test users of this group are technicians.	7
Group NT	Test users of this group are no technicians.	13

4 Results

All results provided in this section have been obtained by analyzing data collected during the usability test. The obtained results are presented in the following subsections.

4.1 Installation of the Java Runtime Environment

Both MOCCA Local and MOCCA Online represent Java based solutions. To answer research question Q1, we assessed whether the given dependency on Java raises additional usability issues. Both MOCCA Online and MOCCA Local automatically check for an installed JRE upon start-up. If no JRE can be detected, MOCCA Local automatically redirects the user to the Java installation page provided by Oracle[2]. Similarly, MOCCA Online provides users an appropriate error message containing a link to the Java installation page. Users have to follow this link manually.

Surprisingly, it turned out that without exception all test users were basically aware of Java. When being requested by MOCCA Local or MOCCA Online to

[2] http://www.java.com/en/download/

install a JRE, 95% of the test users started the Java installation process using the provided link or button. Only 5% did not know what to do in this situation. After starting the Java installation process, 20% of all test users had problems to successfully complete it. These users mostly successfully downloaded Java but did not realize that the downloaded installer file had to be executed afterwards. User group specific results are illustrated in Fig. 6 and show that the Java installation process was especially problematic for older and non-graduate users. As expected, also test users without technical background were more prone to problems during the Java installation process. Interestingly, users of Group B, i.e. users who had to install Java in the course of using MOCCA Online, had more problems with the Java installation compared to users of Group A.

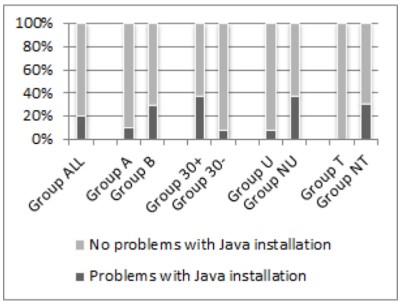

Fig. 6. The installation of Java was problematic especially for users of Group 30+, Group NU, and Group NT.

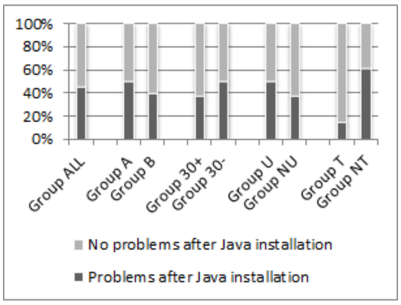

Fig. 7. Especially users of Group NT had problems to proceed after the Java installation process.

After completion of the Java installation process, users had to manually restart the Java Webstart based installation process of MOCCA Local or to manually reload the Java Applet of MOCCA Online. It turned out that this was problematic for 45% of all test users. The group specific results illustrated in Figure 7 show that especially technically inexperienced users had problems in this situation. This time, users of Group A were slightly more prone to usability problems. Obviously, after completion of the Java installation process it was more intuitive for users to manually reload the website containing the Java Applet of MOCCA Online than to manually restart the installation procedure of MOCCA Local.

4.2 Installation of MOCCA Local

The installation of MOCCA Local is based on Java Webstart technology. Hence, test users simply had to click a provided button on a website to start the installation process. This was intuitive for 95% of all test users. After completion of the Java Webstart based installation process, a new Web-browser tab was opened automatically. The website shown in this tab asked test users to install a certificate into their Web browser[3]. The certificate to be installed was provided via a link. 20% of all test users just ignored this message and didn't install the certificate at all. Fig. 8 shows that this affected all user groups. 15% of all test users downloaded the certificate but did not install it. Another 10% were not able to complete the certificate installation process on their own.

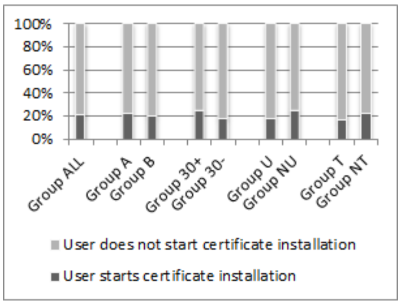

Fig. 8. No significant differences between different user groups could be observed regarding the installation of certificates.

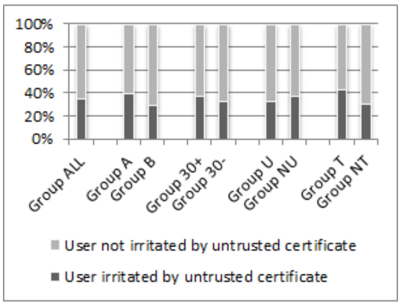

Fig. 9. Users of all user groups were irritated by untrusted certificates.

The certificate to be installed was not recognized as trusted by the used Web browser. Hence, a security warning was shown during the installation process. 35% of all test users felt irritated by this security warning and were not sure whether to proceed with the installation process or not. Fig. 9 shows that again there were only marginal differences between different user groups.

4.3 Card Reader Interaction

MOCCA Local and MOCCA Online support both smart card reader devices with and without integrated PIN pads. Experience has shown that usually devices with integrated PIN pad cause more usability problems. Thus, we have used a Reiner SCT card reader device with integrated PIN pad during the conducted usability test.

It turned out that 25% of all test users had problems to enter the PIN through the integrated PIN pad correctly. In most cases, users didn't realize that entered PINs had to be confirmed using a green OK button. Fig. 10 shows that especially graduated users had problems to enter the PIN correctly. Interestingly, test users starting with the evaluation of MOCCA Local (Group A) were also more prone to problems during the entering of PINs.

[3] This certificate is required to establish an appropriate trust status between the Web browser and MOCCA Local

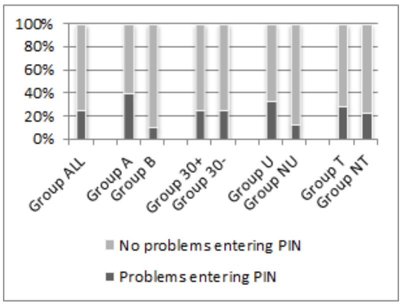

Fig. 10. Users of Group A had significantly more problems to enter the PIN correctly.

A significant learning effect could be observed. Test users, who ran into an timeout because of not confirming the PIN entry by pressing the OK button, did not make the same mistake twice. Already the second PIN entry could be completed successfully by all test users.

4.4 Usage of MOCCA Local

In order to test the usability of MOCCA Local, test users were asked to carry out a typical e-Government procedure using their Citizen Card and MOCCA Local. This procedure comprised the reading of identification data from the user's smart card and the electronic signing of an application form.

Usually, when MOCCA Local is requested to access the locally connected smart card, there is a short delay until MOCCA Local starts up its GUI. However, it turned out that only 5% of all test users were irritated by this delay.

The GUI basically serves two purposes. First, it allows users to enter secret PINs if no card reader with integrated PIN pad is used. Furthermore, it allows users to review the data that is about to be signed during a signature creation process. Users can follow a link labeled "Signature Data" in order to open a separate window that finally contains the data to be signed. Interestingly, the conducted usability test revealed that only 40% of all test users were interested in the provided signature data and followed the shown link to inspect them. All other test users just completed the signature process without reviewing the data to be signed. Fig. 11 shows that this affected all user groups. Interestingly, test users with technical background showed most interest in the provided signature data.

In general, all test users were able to carry out the e-Government procedure using MOCCA Local. Severe usability issues did not arise. The usability, security, and trustworthiness of MOCCA Local has also been attested by the test users. Most of them perceived MOCCA Local as secure and trustworthy. Fig. 12 illustrates group specific results.

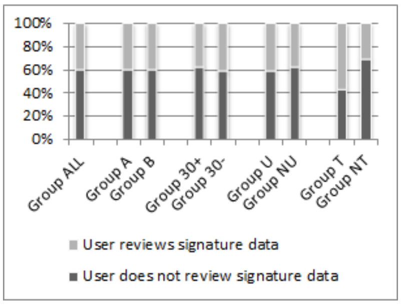

Fig. 11. The majority of all test users was not interested in the data to be signed.

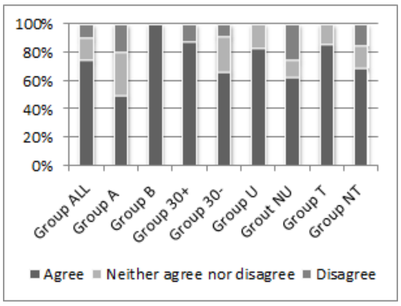

Fig. 12. Most users perceived MOCCA Local as secure and trustworthy.

4.5 Usage of MOCCA Online

Similar to MOCCA Local, the usability of MOCCA Online has been evaluated by requesting test users to carry out a typical e-Government procedure. Again, this procedure comprised the reading of identity data from the user's smart card and the electronic signing of an application form.

On the client system, a Java Applet represents the key component of MOCCA Online. The Applet implements access to the locally connected smart card and offers the user a GUI. Again, this GUI can be used to enter PINs (if a smart card reader without integrated PIN pad is used) and to access and review data to be signed. Compared to MOCCA Local, the Java Applet usually takes more time to load and to provide the user with the GUI. In total, 20% of all test users were irritated by the delay caused by the Applet loading process.

Since the used Java Applet requires access to local resources (i.e. the smart card), the Applet needs to be signed. For the conducted usability test we used a test instance of MOCCA Online that was signed with a test certificate only. This certificate was not recognized to be trusted by the used Web browser. Hence, during the loading of the Java Applet a security warning was shown. 35% of all test users were irritated by this security warning and considered to cancel the loading process. Fig. 13 illustrates group specific results and shows that especially non-graduated users were irritated by the displayed security warning.

Similar to MOCCA Local, only a small percentage of all test users showed interest in the provided signature data. 80% completed the electronic signing process without verifying the data to be signed. Fig. 14 shows that escpecially non-graduated test users were not interested in the data to be signed.

As shown in Fig. 15, the majority of all test users perceived MOCCA Online as secure and trustworthy. Especially older and well-educated test users rated the security and trustworthiness of MOCCA Online positively.

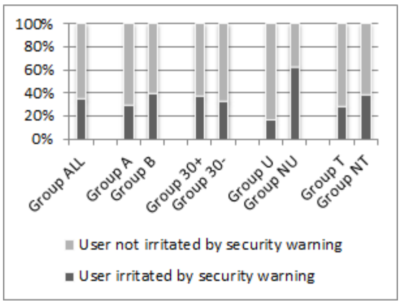

Fig. 13. Especially users of Group NU were irritated by the shown security warning.

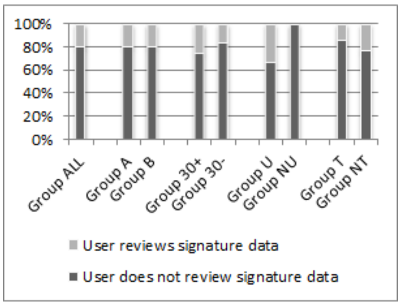

Fig. 14. The majority of all users was not interested in the data to be signed.

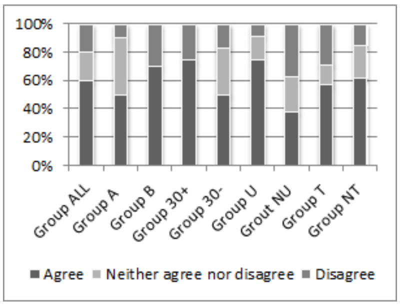

Fig. 15. Most users perceived MOCCA Online as secure and trustworthy.

5 Discussion

The conducted usability test yielded various interesting results. In this section we interpret these results to answer the five predefined research questions. Furthermore, we sum up the most relevant lessons learned and derive possible improvements for the evaluated CCS implementations.

To answer research question Q1, we can conclude that reliance on Java technology does not automatically lead to usability problems. All test users were aware of Java and most of them were able to complete the Java installation process successfully. Still, usability problems could be identified regarding the integration of the Java installation process. After completion of the Java installation process, some users did not know how to proceed. This especially applied to technically inexperienced users. In order to overcome this problem and to improve the integration of the Java installation process, users should be provided with more information and guidance during the installation process.

Regarding research question Q2, the conducted usability test revealed that the Java Webstart based installation process of MOCCA Local does not cause severe usability problems. Most users were able to install MOCCA Local without assistance. However, several users had problems with the subsequent certificate installation that had to be carried out in the used Web browser. Again, this problem can be overcome by providing users with more information and guidance during the installation process. Additionally, used certificates should always be chosen such that their trust status is recognized by common Web browsers. Otherwise, displayed security warnings might irritate users and lead to an abort of the certificate installation process.

The use of MOCCA Local turned out to be unproblematic for users. Minor problems occurred only during the first PIN entry, when users did not know that entered PINs had to be confirmed using the green OK button on the card reader device. To avoid possible errors already during the first use of MOCCA Local, users should be informed appropriately if a PIN confirmation is required. It also turned out that most users did not verify provided signature data before electronically signing them. To solve this issue, the link that has to be followed in order to display the signature data should be placed more prominently in the shown GUI window (cf. Fig.3). Despite these minor issues, we can answer research question Q3 by concluding that MOCCA Local is usable for most users without problems.

Similar results have been obtained for research question Q4. Evaluation of MOCCA Online has shown that additional information about an expected confirmation of PIN entries could improve usability. Similar to MOCCA Local, signature data to be signed was hardly ever reviewed by test users. A more prominent placement and design of the shown link that leads to the signature data (cf. Fig. 5) thus seems reasonable. The conducted usability test has also shown that users are often irritated by displayed security warnings. Hence, it should be guaranteed that the trust status of used signing certificate of the MOCCA Applet is recognized by common Web browsers.

To answer research question Q5, we can conclude that both security and trustworthiness of MOCCA Local and MOCCA Online have been rated positively by most test users. A direct comparison of the results obtained for MOCCA Local and MOCCA Online shows that MOCCA Local has been rated slightly better than MOCCA Online. Interestingly, older and graduated users rated both evaluated CSS implementations better than younger and non-graduated users. According to the obtained results, technicians rated the security and trustworthiness of MOCCA Local higher. For users without technical background MOCCA Online appeared to be more secure and trustworthy. For both CCS implementations, it turned out that the use of untrusted certificates significantly reduces the perceived security and trustworthiness. Hence, it is crucial that CCS implementations rely on certificates that are recognized as trusted by common Web browsers.

6 Conclusions

The conducted usability evaluation of MOCCA Local and MOCCA Online has led to valuable findings. The obtained results show that both MOCCA Local and MOCCA Online basically fulfill given usability requirements. Most test users were able to successfully install and use the evaluated components without assistance.

Still, some minor usability problems could be identified. Provision of more detailed information and improved guidance through installation routines will probably solve most of the identified issues. Additionally, reliance on certificate being recognized as trusted by common Web browsers is crucial for the perceived security and trustworthiness of MOCCA Local and MOCCA Online.

All obtained results and findings will be incorporated in future releases of the evaluated CCS implementations. This way, the conducted usability test will contribute to the usability of MOCCA Local and MOCCA Online and will help to improve the user acceptance of e-Government applications that rely on these components.

References

1. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. Offcial Journal of the European Communities L 013, 12-20 (2000).
2. Austrian Federal Act on Electronic Signatures. Federal Law Gazette, part I, Nr. 137/2000, last amended by Nr. 59/2008, (2000).
3. Austrian Federal Act on Provisions Facilitating Electronic Communications with Public Bodies. Federal Law Gazette, part I, Nr. 10/2004, (2004).
4. Orthacker,C., Zefferer, T.: Accessibility Challenges in e-Government: an Austrian Experience. Proceedings of the Forth International Conference on Internet Technologies and Applications (ITA 11), 221–228, (2011)
5. Garcia, A. C. B., Maciel, C., Pinto, F. B.: A Quality Inspection Method to Evaluate E-Government Sites. Electronic Government, 3591, 198-209, Springer, (2005).
6. Sørum, H.: An empirical investigation of user involvement, website quality and perceived user satisfaction in eGovernment environments. Proceedings of the Second international conference on Electronic government and the information systems perspective, EGOVIS'11 Toulouse, France, Springer-Verlag Berlin, Heidelberg, 122–134, (2011).
7. Leitold, H., Hollosi, A., Posch, R.: Security Architecture of the Austrian Citizen Card Concept. Proceedings of the 18th Annual Computer Security Applications Conference, Washington, DC, USA, IEEE Computer Society, 391–400, (2002).
8. Orthacker, C., Centner, M., Kittl, C.: Qualified Mobile Server Signature. Proceedings of the 25th TC 11 International Information Security Conference, SEC 2010, (2010).
9. a.sign client, `https://www.a-trust.at/default.aspx?lang=GE&ch=1&node=765`, (2012).
10. Centner, M., Orthacker C., Bauer, W.: Minimal-Footprint Middleware for the Creation of Qualified Signatures. Proceedings of the 6th International Conference on Web Information Systems and Technologies, Portugal, 64–69, (2010).
11. Modular Open Citizen Card Architecture, `http://mocca.egovlabs.gv.at/`, (2012).
12. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers, (1993).
13. Siddhartha, A.: National e-ID card schemes: A European overview. Information Security Technical Report, Volume 13, 46–53, (2008).