Usability of Policy Authoring Tools: a Layered Approach

Stephanie Weinhardt[1] and Olamide Omolola[2]
[1]*Institute for Human Factors Engineering, University of Stuttgart, Nobelstrasse 12, Stuttgart, Germany*
[2]*Institute of Applied Information Processing and Communications, Graz University of Technology, Inffeldgasse 16a, Graz, Austria*
stephanie.weinhardt@iat.uni-stuttgart.de, olamide.omolola@iaik.tugraz.at

Keywords: Policy Authoring, Usability, Layered Approach, Trust Policy,

Abstract: Many policy authoring tools lack usability, and this deficiency often deters new users from using the tools. In this paper, we propose an approach to make policy authoring more usable and enable novice users to create policies. The process of creating a trust policy using a trust policy language has different levels of complexity for different users. This paper identifies three categories of such users and introduces a three-layered approach to cater to each user group. The approach intuitively reduces the functionalities available based on the capability of each group of users and therefore making policy creation more usable.

1 INTRODUCTION

Although a lot of work on usability in IT-Security has been conducted [1]–[8], policy authoring tools still lack usability [9]–[13]. We believe that this is a multi-layered problem. On the one hand, the tools themselves lack usability, on the other hand, the policy languages are too complex and too hard to understand for people with little or no knowledge in policy authoring [9], [10]. Especially for novice users policy authoring is a complex and complicated topic. The need for usable security has been addressed quite extensively, but within that usable policy authoring is still unaddressed [11] and solutions for usable policy authoring need to be developed. Especially because Reeder et al. [12] discovered that there is an uprising need for all kinds of users to formulate different types of policies. This need is increasing because policy authoring is not limited to administrators anymore [7], [13], [14]. Recent developments, such as the Internet of Things and automation, increases the need for non-programmers and novice users to formulate policies. But end users will not purchase and will not use security software they cannot understand [8].

This paper aims to show an approach to designing usable policy authoring. In the field of policy authoring different user groups with varying levels of knowledge and different goals, need different sets of functionalities and a tailored user interface to not overstrain novice users and still provide full functionality to expert users [13]–[15]. So far, there was no possibility for novice and expert users to work within the same tool. Beyond that, Clare-Marie Karat et al. [16] found in their evaluation of the SPARCLE tool that it is beneficial to let users choose the way they want to create their policy, no matter to which user group they belong.

In this paper, we introduce a 3-layer approach. The 3-layers combine several methods and principles to meet all the identified requirements, plus increasing usability and user adoption. Each layer has a specific target user group for which it is designed. They also address user friendliness on their own by implementing well-known usability and usable security design principles. Moreover, these different layers gradually introduce users to the topic and ideally enable them to become experts in it and create policies that are more complex.

We implemented the 3-layer-approach in a high-fidelity prototype created in the framework of the LIGHTest. This European funded research project implements a global trust infrastructure that can be used by anyone. The challenge of enabling all kinds of users to formulate trust policies with a tool and enhancing its usability is the focus of this contribution. Although the approach is shown within the European project using trust policies, we strongly believe that the method can be adapted to any policy authoring tool.

The structure of the paper is as follows: first, we give a short outline of related work as well as on literature that deals with usable security design principles. Then, we introduce the approach and show how it is implemented in the high-fidelity prototype. Finally, we provide an outlook on the following evaluation and future work.

2 RELATED WORK

Policy authoring is the process of creating a set of rules combined either with the goal to regulate access, make systems more secure or in our case to define trust relationships. There are several existing contributions on the topic of usable policy authoring from the last decade. Most of them focus on authoring access or security policies [10], [12], [17]–[21]. We know of no significant research that has been done on usable trust policy authoring tools.

No matter what kind of policy users create, they are all formulated using a programming policy language. One effort in making policy authoring more user-friendly was to implement it with a natural language. Natural language is believed to shorten the "distance" [22] between the user's real-world needs and its expression within a computer. Furthermore, some contributions suggest natural language should be controlled, i.e. its semantics should be limited because it is less ambiguous in comparison to free natural language [19]. Although the controlled natural language introduced in [9], [18], [23]–[26] makes a big step towards usability, it still requires some coding knowledge that makes it hard for users without any programming experience to use it.

Most research focuses on a specific tool, an area of expertise or a particular experienced user group [27] that cannot be adapted to the European project topic because it addresses a different issue as well as a diverse user group and cannot utilise the substantive, limited knowledge of the problem. A lot of contributions focus on a specific set of users and try to make policy authoring more usable by adding usability on top of an already existing policy language or tool. Little contributions consider a user-centred design from the very beginning. With the approach presented here, we incorporate user-centred design from the very beginning.

Also, the authors of [9], [10] mention that most of the policy tools and languages are too complex to be understood by novice users – therefore, complexity should be reduced.
Despite rare contributions that address usable policy authoring on the next level, there have been two significant contributions that form the basis of our approach.

The authors of [16], [25] focused on usable (security) policy authoring. They evaluated three different possibilities to formulate security policies: a guided natural language, a structured format and an unconstrained natural language. Their evaluation showed that the guided natural language and the structured format helped users formulate policies. The quality of the policies with the unconstrained natural language was not as high as with the guided natural language and the structured format. There was no significant difference between the guided natural language and the structured format considering usability and user satisfaction. They also indicate that it adds value to the user interface if the users can select in which form they create their policy in.

In [14], the authors identified three user groups: novice users, intermediate users and expert users. They stated the necessity of designing the tool or interaction differently for each user group. They also identified three different possibilities (levels) of how the user could formulate an access control (policy). The first possibility guides the user systematically through the creation process of a policy; the user only needs to select from the given choices. The second possibility provides the users with the alternative in formulating their access control policy based on their mental model. Although this provides the user with more functionality/possibilities, it also leaves the user without guidance and introduction.
The last possibility provides the users, which are most likely system administrators, with the full functionality that also includes managing multiple systems, not hindering their work.

Other contributions are design guidelines for designing usable policy authoring tools [12], [18], [28]–[31]. Ever since usable security became a topic in security research, new design principles have been introduced by various authors. A lot of them are basic usability guidelines that have been adapted to the topic[2]. As the design of our approach considers common usability guidelines, we only mention the ones that we took under special consideration:

- A professional look and feel are beneficial in establishing trust [30], [31], [28]

- Transparency to an extent aids trust and understand ability [30]
- Attractive design helps in building trust [28]
- Support object grouping [18], [12]
- Communicate and enforce rule structure [18], [12]
- Support appropriate limitation of expressivity [18]

However, we firmly believe that all of those principles still will not make trust policy authoring usable for a large group of users, instead of a specific set of users. Therefore, we see the need for an abstract interaction concept that allows all kinds of users to formulate policies matching their needs.

Although all the contributions are essential for usable policy authoring, none of them explicitly deals with trust policy authoring as well designing for a diverse user group.

Taking all these aspects under consideration, we designed a new approach, i.e., the 3-layer approach that allows us to address different users' needs.

3 PROTOTYPE ANALYSIS

The first interaction concept was designed based on related work and further analysis in the field of trust policy authoring. This first concept, shown in Figure 1, was implemented in a low-fidelity prototype for visualisation and evaluation purposes.

With the prototype, a usability evaluation was conducted. The main goal of the evaluation was to get a first impression on the overall usability, as well as the mental model of the users while interacting with such a tool.
In the description below, we briefly go into this evaluation to explain the key findings. Ten participants took part in the evaluation that were acquired randomly from a pool of participants. The only criteria they had to meet was to be over 18. Eight of the participants stated that they had primary to good programming skills with mostly, one participant did not have experience in programming at all, and one had advanced knowledge. Considering creation of policies, only one participant ever created a policy before.
The participants had to create a new trust policy using a controlled natural language approach with a combination of typing and dynamic dropdowns with suggestions.

Results from the evaluation show that most of the users were able to create a trust policy, as the task completion rate was 94% overall tasks. Lowest task completion rate was 90%. Although this rates hint towards a success, most of the participants showed signs of mental stress and indisposition. Although they succeeded in creating a trust policy, they did not feel good about it and were unsure about the quality of the trust policy they created and its impact.

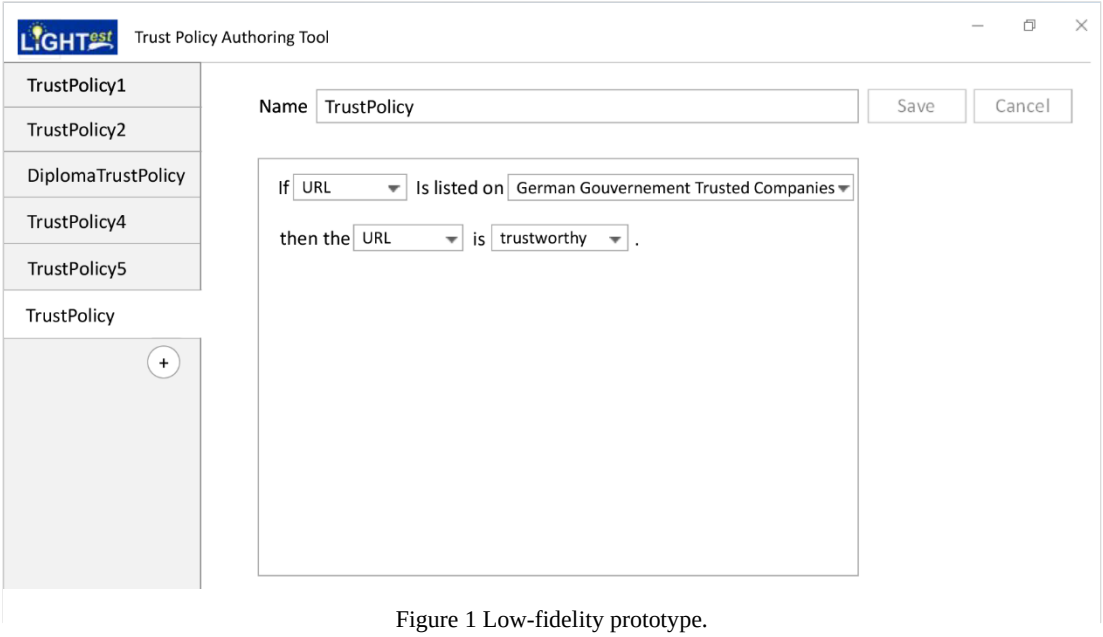

Figure 1 Low-fidelity prototype.

Therefore we believe that the high success rates is because of the guidance given through the whole creation process, rather than good usability of the tool. In the follow-up interview participants mentioned, their discomfort while creating the policy because they were unsure what was actually asked and how the systems functions. Considering those results, we concluded that the approach of the low-fidelity prototype is not satisfactory. Tools like these need to have high usability, as well as meeting the knowledge level of the users to make them feel good while interacting with the tool and not overstraining them.

During the follow-up interview, participants also expressed their need for either more complex or simpler interaction possibilities. These results are backed up by the findings from [14], [17]. Both suggest putting users into three categories.

1. Novice users with no to little knowledge
2. Intermediate users with low to intermediate knowledge
3. Expert users with intermediate to expert knowledge

These different aspects led us to the conclusion that we not only have differently skilled user groups, but we also need a different set of functionalities and guidance for each of them as they also have different expectations, mental models and goals.

Novice users that were recently introduced to policy authoring and trust policies are assumed to state entities they trust or do not trust. Anything beyond basic functionality would probably result in cognitive overload and therefore would reduce usability and understanding of the tool.

Intermediate users that either already know how other policy authoring tools work, have some programming skills or have already created some simple trust policies, might want to integrate some conditions into their Trust Policies. They should have more functions to create policies that are more complex.

Expert users that have profound programming skills or good knowledge in (trust) policy authoring want to create complex policies with several conditions and operations. The whole set of functionalities should be available to them. In addition, they do feel comfortable using a programming editor to create their policies.

However, trying to incorporate all these different requirements into one user interface always led to low usability, too much content and a bad structure. It became clear that designing for different user groups would not only mean reducing complexity but also decreasing functionality and providing the various user groups with different user interfaces, which lead us to the development of the 3-layer approach.

Figure 2 High Fidelity Prototype.

4 THE THREE LAYER APPROACH

The 3-layer approach is based on previous contributions [16], [17] and contributions that stated that most policy authoring tools are too complicated for novice users and that the complexity of those tools should be reduced to improve usability [9]–[11]. To achieve a holistic approach to the problem of usable policy authoring for all user groups we combined several methods, guidelines and principles. The overall concept and each layer within that concept was developed by considering Usability Design Principles [32]–[34], Usable Security Design Principles [4], [29], [30], [35]–[38] and Usable Policy Authoring Design Principles (see chapter 2 Related Work).
The central aspect of the concept is that it offers the user the possibility to create a trust policy in three different ways:

- using a graphical editor
- create it in a controlled natural language
- and programming a policy using a trust policy language

The most basic layer is the Graphical Layer (GL) providing just basic functionality, designed for novice users. The Natural Language Layer (NLL) has more features and is intended for intermediate users. The Trust Policy Language Layer (TPLL) has full functionality and is designed for experts. When creating a new trust policy, the tool provides the users with the three different possibilities and explains each.

By providing the users with three different possibilities to create a trust policy, the tool becomes more usable for each user group because each layer is customised for their needs. As a result, the needs established in chapter 3 are met. Also, novice users can learn and maybe create a policy in the programming language at some point [10], [11], [29], [39].

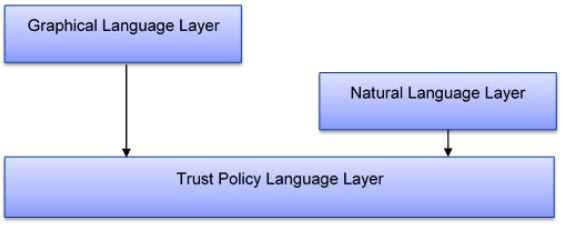

Figure 3: The Three-layer Approach.

Figure 3 shows the concept of the 3-layer approach. It shows that the users always can change from either the GL or the NLL to the TPLL, giving them the possibility to get full functionality for an already existing policy.

Each layer implements design principles in its context by considering the user group and the user goals. The following sections will discuss each layer separately and emphasize the conceptual aspects that have been developed explicitly to make the layers more usable.

4.1 Graphical Layer

The Graphical Layer is designed for novice users and offers basic functionality. Users are only able to state entities they trust or do not trust. This concept is commonly referred to as blacklisting and whitelisting. In this layer, the user is given the possibility to select entities from a searchable list and either add it to the trusted list (whitelist) or the not trusted list (blacklist). The user can also create entities and assign them to the list it deems correct.

The design is aided by giving the user visual feedback through drag and drop colours, icons and text to highlight which is the trusted list (whitelist) and which is the not trusted list (blacklist). The trusted list has a white background colour and a closed padlock icon. The untrusted list has a grey background and an opened padlock icon. For some users it might be more intuitive to edit just one side or the other, others may also want to edit both lists. This capacity could cause some problems when users are putting the same entity on both sides. In that case, it is unclear whether the user wants to trust the entity – or not. The tool detects such inconsistencies, and the user is informed about it by a dialog box.

Despite limited functionality, this layer covers the main feature of the tool and is probably sufficient for a large set of users.

4.2 Natural Language Layer

The Natural Language Layer is designed for users with intermediate knowledge of policy creation or basic programming knowledge. It gives the users the possibility to create policies more naturally, by using a controlled natural language rather than a programming language.

Previous research has shown that giving users the possibility to create policies in the natural language improves usability. We believe that there is potential to this approach and it can be made even more user-friendly. Thereby providing different user groups that have varying needs with varying possibilities while using it.

To create the policy, the users can structure their policy by creating several statements within the policy. A policy consists of at least one statement. Each statement is displayed as a separate design element.
Each statement starts with the words "if input" to hint a possible way of formulating a proper statement. This snippet can be deleted if the users choose to begin their sentence differently. In addition, the tool provides different ways to add elements to a statement.

One way to add content to a statement in the NLL is to write plain text within the statement. The tool will support the user by highlighting words it cannot interpret and by giving suggestions on how and what can be formulated. This interaction concept was adopted from the first usability study described in section 3. Another way for the user to develop the policy is to use predefined building blocks that the tool provides in this layer. After defining the set of functionality for the Natural Language Layer, possible building blocks and their parameters were analysed and determined. Those are provided on top of the editing area of the tool for the ease of access. By clicking on a building block the parameters and options that are available for this block are shown on a sidebar at the right. Therefore, the users can see what types of building blocks are available to them and what content they have. The parameters of the building blocks are displayed with different UI elements to fit the content.

There are two possible interactions with these building blocks. One is to click on a building block, set the parameters on the right sidebar and add them to the statement by clicking the “select”-button that is provided within the sidebar. The other possibility is to drag and drop a building block into the statement. The drag and drop action causes a placeholder to appear as well as the parameters in the right sidebar. The users can then set the parameters and add them to the statement by clicking the "select"-button. These two possibilities enable the user to create their statements with the building blocks systematically or defining the statement first and then fill it with the parameters. The building blocks improve usability as they give the users more guidance on what they can formulate but still provide them with the freedom on how they want to create their statement. The building blocks take away some of the cognitive load for intermediate users.

Users can choose anytime what type of interaction they prefer and can seamlessly switch between all the before mentioned interactions. Therefore, this layer provides freedom in formulating the policy supporting different mental models and expectations towards this layer, resulting in better usability.

A similar approach was evaluated by [18], with the result that users produced higher-quality policies in comparison to an unguided approach.

Figure 2 shows an example of a Trust Policy created in the Natural Language Layer.

4.3 Trust Policy Language Layer

The Trust Policy Language Layer is aimed for expert users with intermediate to high level of knowledge in programming and creating trust policies. The layer provides the full set of functionalities to the users by letting them create the policies in the programming language (Trust Policy Language). Creation of a policy is done in a programming editor that has the standard functionality as well as look and feel of common programming tools.

We decided not to change anything from the standard editor look and functionalities because we aimed at giving the users a familiar programming interface. There is no drag and drop function in this layer. However, the tool evaluates the statements in the editor and provides feedback while the users are writing statements. In this layer, the users have total freedom to define and create different constructs. However, the expert users are expected to know how to write policies directly in the Trust Policy Language.

5 DISCUSSION

Although this is not the first contribution to the topic, it is the first contribution that deals with usable trust policy authoring tools from the very beginning of the development. The 3-layer concept is based on previous works, and a usability evaluation carried out initially when the project began. This contribution not only augments principles and approaches from earlier contributions to the topic of trust policy authoring but also combines several mechanisms to give a wide range of possible users the possibility in creating their own (trust) policies. As a result, it contributes to the

issue of explainable security and gives more research perspective on how to make such security mechanisms usable for users. We see potential of such a tool especially in the area of IoT as well as in automation. In those domains (and possibly others), users have to make a lot of (trust) decisions but cannot always decide on the spot.
If the results from future usability evaluations are positive, this approach and its successful implementation into products can contribute to the research on information system trust, security and privacy.

6 CONCLUSIONS

In this paper, we have shown an approach to make policy authoring usable for a diverse user group. Key aspects that we have been demonstrated are a reduction in complexity by reducing functionality and offering three different layers to create a trust policy. Those are also meeting different mental models and user goals. By implementing usability design principles and security design principles in each of the various layers, we designed for improved usability, too.

The Graphical Layer and Trust Policy Layer have a more straightforward design as the user groups can be better defined. The user group for the Natural Language Layer is assumed to be more diverse with different levels of experience with policies creation and programming knowledge. Therefore, we implemented different interaction concepts and different guidance models in the Natural Language Layer, giving the users various possibilities for policy creation.

The presented mechanism and concepts can be implemented in any policy authoring tool.
As part of the future work, we will evaluate the 3-layer approach in an extensive usability evaluation and optimise it based on the results. Furthermore, we will research and design a wizard that introduces the user to the topic and the tool. Visualisation of the authored policies and their outcome will be integrated too.

Recent research shows that user acceptance of a tool does not only have to do with good usability but also with a good user experience design. Therefore, future work aims at not only investigating, how to make such policy authoring tools user-friendly but also how to improve the underlying user experience in these tools.

ACKNOWLEDGEMENTS

This research is part of the LIGHT^{est} project funded from the European Union's Horizon 2020 research and innovation programme under G.A. No 700321

REFERENCES

[1] I. Kirlappos and M. A. Sasse, "What Usable Security Really Means : Trusting and Engaging Users," *Hum. Asp. Inf. Secur. Privacy, Trust HAS. Lect. Notes Comput. Sci.*, no. 8533, p. 11, 2014.

[2] L. Lo Iacono, M. Smith, E. Von Zezschwitz, P. L. Gorski, and P. Nehren, "Consolidating Principles and Patterns for Human-centred Usable Security Research and Development," in *European Workshop on Usable Security*, 2018, no. April.

[3] L. F. Cranor and S. Garfinkel, "Secure or usable?," *IEEE Secur. Priv.*, vol. 2, no. 5, pp. 16–18, 2004.

[4] L. P. Prieto, M. J. Rodríguez-Triana, M. Kusmin, and M. Laanpere, "Maybe poor Jhonny Really Cannot Encrypt - The Case for a Complexity Theory for Usable Security," *CEUR Workshop Proc.*, vol. 1828, pp. 53–59, 2017.

[5] A. Ferreira, C. Rusu, and S. Roncagliolo, "Usability and security patterns," *Proc. 2nd Int. Conf. Adv. Comput. Interact. ACHI 2009*, pp. 301–305, 2009.

[6] P. H. Meland and J. Jensen, "Secure Software Design in Practice," *2008 Third Int. Conf. Availability, Reliab. Secur.*, pp. 1164–1171, 2008.

[7] P. A. Bonatti, "Flexible and Usable Policies," *W3C Work. Lang. Priv. Policy Negot. Semant. Driven Enforc. REWERSE*, pp. 1–5, 2006.

[8] M. E. Zurko, R. T. Simon, and S. Street, "User-Centered Security," vol. 1, no. 212, pp. 1–9, 1996.

[9] J. L. de Coi, P. Fankhauser, T. Kuhn, W. Neijdl, and D. Olmedilla, "Controlled Natural Language Policies," *Seventh Framew. Program.*, pp. 9–11, 2011.

[10] M. Rudolph, "User-friendly and Tailored Policy Administration Points," no. 076, 2014.

[11] D. D. Caputo, S. L. Pfleeger, M. A. Sasse, P. Ammann, J. Offutt, and L. Deng, "Barriers to Usable Security? Three Organizational Case Studies," *IEEE Secur. Priv.*, vol. 14, no. 5, pp. 22–32, 2016.

[12] R. W. Reeder, C.-M. Karat, J. Karat, and C. Brodie, "Usability Challenges in Security and Privacy Policy-Authoring Interfaces," *IFIP Conf. Human-Computer Interact.*, pp. 141–155, 2007.

[13] S. Fischer-Hübner, L. Iacono, and S. Möller, "Usable Security und Privacy," *Datenschutz und Datensicherheit - DuD*, vol. 34, pp. 773–782, 2010.

[14] X. Cao and L. Iverson, "Intentional access management: making access control usable for end-

users," *SOUPS Proc. Second Symp. Usable Priv. Secur.*, vol. 149, no. 06, pp. 20–31, 2006.
[15] M. Bishop, "Psychological Acceptability Revisited," in *Security and Usability: Designing Secure Systems That People Can Use*, 2005, pp. 1–11.
[16] J. K. Clare-Marie Karat, Carolyn Brodie, "Usability Design and Evaluation for Privacy and Security Solutions," in *Security and Usability: Designing Secure Systems That People Can Use*, 2005, pp. 47–74.
[17] K. Beznosov, P. Inglesant, J. Lobo, R. Reeder, and M. Zurko, "Usability meets access control: challenges and research opportunities," *SACMAT Proc. 14th ACM Symp. Access Control Model. Technol.*, vol. 09, pp. 73–74, 2009.
[18] M. Johnson, J. Karat, C.-M. Karat, and K. Grueneberg, "Optimizing a policy authoring framework for security and privacy policies," *Proc. Sixth Symp. Usable Priv. Secur. - SOUPS '10*, p. 1, 2010.
[19] D. Chadwick and A. Sasse, "The virtuous circle of expressing authorization policies," *CEUR Workshop Proc.*, vol. 207, 2006.
[20] L. Bauer, L. F. Cranor, R. W. Reeder, M. K. Reiter, and K. Vaniea, "A user study of policy creation in a flexible access-control system," *Proceeding twenty-sixth Annu. CHI Conf. Hum. factors Comput. Syst. - CHI '08*, p. 543, 2008.
[21] C.-M. Karat, J. Karat, C. Brodie, and J. Feng, "Evaluating interfaces for privacy policy rule authoring," *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. - CHI '06*, p. 83, 2006.
[22] J. F. Pane, C. A. Ratanamahatana, and B. A. Myers, "Studying the language and structure in non-programmers' solutions to programming problems," *Int. J. Hum. Comput. Stud.*, vol. 54, no. 2, pp. 237–264, 2001.
[23] K. Vaniea, C.-M. Karat, J. B. Gross, J. Karat, and C. Brodie, "Evaluating assistance of natural language policy authoring," *Proc. 4th Symp. Usable Priv. Secur. - SOUPS '08*, p. 65, 2008.
[24] L. Shi and D. W. Chadwick, "A controlled natural language interface for authoring access control policies," *Proc. 2011 ACM Symp. Appl. Comput. - SAC '11*, p. 1524, 2011.
[25] C. A. Brodie, "An Empirical Study of Natural Language Parsing of Privacy Policy Rules Using the SPARCLE Policy Workbench," *Public Policy*, pp. 8–19, 2006.
[26] J. Karat, C. Karat, C. Brodie, and J. Feng, "Designing Natural Language and Structured Entry Methods for Privacy Policy Authoring," pp. 671–684, 2005.
[27] C. C. Machado, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "Policy Authoring for Software-Defined Networking Management," *IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, 2015.
[28] A. Patrick, S. Marsh, and P. Briggs, "Designing systems that people will trust," *Secur. Usability Des. Secur. Syst. That People Can Use*, no. January, pp. 75–100, 2005.
[29] K.-P. Yee, "User Interaction Design for Secure Systems," *Proc. 4th Int. Conf. Inf. Commun. Secur.*, pp. 278–290, 2002.
[30] E. Atwater, C. Bocovich, U. Hengartner, E. Lank, and I. Goldberg, "Leading Johnny to Water: Designing for Usability and Trust," *Elev. Symp. Usable Priv. Secur. (SOUPS 2015)*, pp. 69–88, 2015.
[31] P. Lanford, "E-Commerce: A Trust Perspective," *Int. Conf. Internet Comput.*, pp. 64–70, 2006.
[32] D. A. Norman, *The Design of Everyday Things*, 2nd ed. New York: Perseus Books Group, 2013.
[33] ISO9241-110:2006(en), "Ergonomics of human-system interaction - Part 110: Dialogue principles," 2006.
[34] B. Shneiderman, *Designing the User Interface*, 6th ed. Edinburgh: Pearson Education Limited, 2016.
[35] C. Braz and J. Robert, "Security and Usability : The Case of the User Authentication Methods," *Proc. 18th Int. Conf. Assoc. - IHM '06*, pp. 199–203, 2006.
[36] A. Whitten and D. Tygar, "Why Johnny Can ' t Encrypt," *Security*, vol. 1999, no. October, pp. 679–702, 2005.
[37] S. L. Garfinkel, "Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable by," *Gene*, vol. 31, no. 1987, pp. 234–239, 2005.
[38] M. A. Sasse, S. Brostoff, and D. Weirich, "Transforming the 'Weakest Link': A Human-Computer Interaction Approach for Usable and Effective Security," 2001.
[39] T. Trojer, B. Katt, F. Wozak, and T. Schabetsberger, "An authoring framework for security policies: A use-case within the healthcare domain," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng.*, vol. 69 LNICST, pp. 1–9, 2011.