https://doi.org/ 10.70985/xbmpaz29
Vol. 2024, No. 10, pp. 41–51, 2024 ISSN 1353-4858 print, ISSN 1872-9371 online
Network Security

Improving the User-friendliness and Ease of Use of Network Security Software through Visual Communication Design

Wei Sun[1], Bo Zhang[1], Liwei Li[2], Xiaocen Guo[2*]

[1] School of Art Design, Shandong Youth University of Political Science, Jinan, China
[2*] College of Design, Hanyang University, Ansan, Korea

Abstract

Existing network security software generally has problems such as overly complex interface design, unreasonable navigation structure and imperfect error handling mechanism. This paper uses visual communication design methods to simplify the software interface, optimize the navigation structure and improve the error handling mechanism. In the interface simplification design, combined with the characteristics of network security software, a flat layout is adopted to reduce redundant security configuration items, optimize the layout of security functions, and highlight key security functions to reduce the user's learning cost and risk of misoperation in complex security operations. When optimizing navigation, breadcrumb navigation and global search functions are added to facilitate users to quickly locate security settings and monitoring functions, and quick entrances to important security functions are added to the interface to improve the user's response efficiency in emergency security incidents. By optimizing the display of security prompt information, the accuracy and guidance of error prompts are enhanced, and automatic recovery mechanisms and security log recording functions are applied, improving the system's ability to handle security incidents and user response speed. After the interface is simplified, the time to update the software is reduced from 7 minutes to 4.5 minutes, and the navigation operation steps for professionals to handle system alarms are reduced by 50%. The approach adopted effectively reduces the interface complexity of network security software, optimizes the problem of improper navigation error handling, and significantly improves user experience and usage efficiency.

Keywords: Visual Communication Design, User Interface Optimization, Navigation Enhancement, Error Handling Improvement, Software Usability

1. Introduction

With the rapid development of information technology, network security software has played an important role in protecting user information and maintaining system security. However, many current network security [1-2] software faces problems such as complex interface design [3-4], inconvenient function navigation, and imperfect user error feedback mechanism [5]. These problems affect the user experience [6] and also lead to the emergence of security vulnerabilities [7-8]. Complex user interfaces (UI) [9-10] often make it difficult for users to quickly find the required functions, resulting in a longer time for users to open the software, browse the interface, find the required function options, make settings and adjustments, and perform operations [11]. Unreasonable navigation structure requires users to go through multiple steps when performing tasks, increasing the burden of operation and the probability of errors. The imperfect user error feedback mechanism means that users lack clear guidance when they encounter problems, which makes them feel discouraged. As network threats become increasingly complex [12], users' expectations for security software are not limited to comprehensive functions, but also emphasize ease of use and efficiency. Therefore, it is particularly important to improve the user-friendliness and ease of use of network security software. Effective interface design and intuitive operation processes improve the interaction efficiency between users and software [13-14], enhance users' trust in software, and thus promote the widespread application of software. Many network security softwares have complex interface designs and unreasonable function distribution, which affect user experience and make it difficult to meet security needs in a timely manner. Studies have shown that users often increase the risk of errors and delay the handling of security incidents due to cumbersome navigation and too many operation steps during use. At the same time, the existing error feedback mechanism lacks clear guidance, and users find it difficult to respond in time when faced with operation failures or warning messages. These problems not only reduce the efficiency of software use, but also increase system security risks, so they need to be optimized through more efficient design methods.

In recent years, user experience research on software has gradually attracted attention. Before designing an interface, the principles of interface design must be studied first [15]. Kamaruddin, Norfadilah [16] and others studied the interface design principles of multimedia teaching aids in Malaysian higher education through content analysis, and evaluated the application of principles such as consistency, hierarchy, contrast, balance and harmony in interface design to improve the effectiveness of multimedia teaching and students' learning participation. Sharma, Vatsal [17] and others discussed the design and tools of UI and user experience (UX), studied the current status and development trends of UI/UX technology, and covered design principles, interaction methods and the application of commonly used design tools. Rembulan, Glisina Dwinoor [18] and others evaluated the user interface of an e-grocery application through human-centered design (HCD) and usability testing, put forward suggestions for interface improvement, and achieved significant improvements in usability, including effectiveness, efficiency and user satisfaction. Although these studies provided methods and ideas for UI design [19-20], they mostly focused on the application evaluation of interface design principles and tools, and lacked in-depth analysis of the causal relationship between interface design principles and user experience improvement, and the consideration of diverse user needs was still not comprehensive enough.

The theory of visual communication design has become the main research direction of UI design. As an important field of modern design, visual communication design involves the effective transmission and communication of visual elements such as images, text and color. Ilma, N. [21] et al. used qualitative research methods to analyze the application of visual communication design as an information and educational media during the epidemic. Through data collection, data simplification, display and summary, the study showed how visual design can effectively convey information in society, especially in response to the new crown epidemic, providing important educational and entertainment functions for the public. Wang, Ruoyao [22] used deep learning technology to study the interaction between computer-assisted visual communication technology and art in new media scenarios, and verified the performance of the deep learning network model SGRU (Stacked Gated Recurrent Unit) in image coloring by conducting experiments on automatic coloring of line drawings. The results showed that the SGRU network model was superior to other similar methods in terms of coloring accuracy, image quality and creation efficiency. Yoo, Yoon Seok [23] analyzed the structure and content of 15 online graduation exhibition websites of visual communication design majors, evaluated the usability of the exhibition, dividing into four stages: arrival, search, appreciation and interaction, and put forward suggestions for improving future online exhibitions. Although these studies explored the application of visual communication design [24-25] in different scenarios, most of them focused on the analysis of specific cases, lacked a broader and deeper systematic study of the relationship between visual communication design theory and practical effects, and did not take into account the diverse needs of users in different fields.

Based on the above research, this paper adopts the method of visual communication design to systematically improve the user experience of network security software. The study analyzes the pain points and needs of users during use and collects relevant data to guide design decisions. Then the interface is simplified, and a flat layout is adopted. The function distribution is optimized, and important functions are highlighted to reduce the user's learning cost. In the process of navigation optimization, breadcrumb navigation and global search functions are added, and shortcuts to commonly used functions are added to improve the user's navigation efficiency. Finally, by optimizing error prompt information, the automatic recovery mechanism and logging function are applied to improve the error handling ability of the system. The research in this paper not only integrates the concept of visual communication design into network security software, but also through systematic analysis and design methods, it is expected to provide users with a more friendly operation experience and provide new ideas for the design of future network security software.

2. Methods

This paper combines visual communication design methods to improve the user-friendliness and ease of use of network security software. The first step is to collect the pain points and needs of users during use, analyze the problems of existing software in terms of interface complexity, navigation difficulties and insufficient error feedback, and use this as the basis for design improvement. On this basis, the interface is optimized [26], combining flat design and functional layout adjustment to highlight commonly used functions and reduce the learning cost of operation. The efficiency of user search and operation is improved by applying breadcrumb navigation [27], global search and shortcut entry in the UI. When the user makes an operation error, the error handling mechanism enhances the system's error response capability by improving the clarity of the prompt information and adding automatic recovery and logging functions [28]. The overall framework aims to build a more efficient and friendly network security software user experience through multi-level optimization strategies. The research framework is shown in Figure 1.

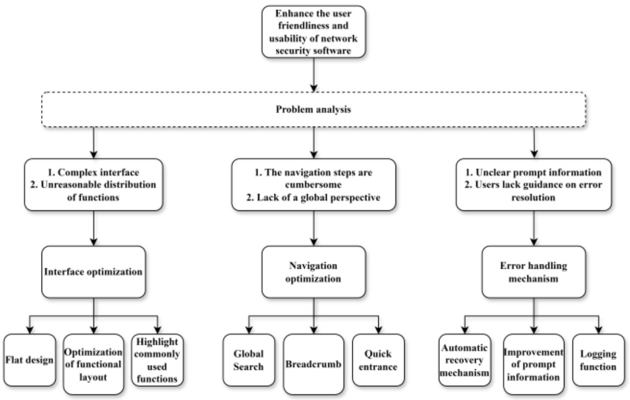

Figure 1 Research framework

2.1 Data Collection and User Needs

Data collection adopts a combination of structured questionnaires, user interviews and log data analysis. To ensure the representativeness of the results, the survey covers 200 users from different industries, including information technology, finance, education, public services and manufacturing. 40 users were randomly selected from each industry, and the questionnaire content was designed around key indicators such as interface simplicity, navigation efficiency and error handling mechanism. User feedback from each industry is strictly screened and classified to ensure that the collected data is representative in both breadth and depth. In terms of demand analysis [29-30], multiple user needs for the software are obtained through user behavior analysis, questionnaire surveys and usability testing. The user interview selected 30 representative users from the questionnaire participants, and conducted semi-structured interviews to gain an in-depth understanding of the actual pain points and needs of users during operation. The interview content covered opinions on functional layout, common usage obstacles, and suggestions for optimization. All interviews were recorded and organized by two researchers to ensure the accuracy of the information.

The Likert scale is used in the scoring process. Respondents score dimensions such as interface simplicity [31-32], navigation efficiency, and error handling based on their own usage experience. The scoring criteria for each dimension are jointly developed by design experts and user experience experts to ensure that all aspects of user experience are covered. The scores range from 1 to 10, and the higher the value, the stronger the user's demand for that dimension. The scores in Table 1 are derived from the questionnaire feedback of 200 users, and the average scores of different user groups are calculated to show the differences in demand.

Table 1 shows that the interface simplicity score of information technology [33] practitioners is 8.5, indicating that this group has a high demand for functional simplification, and the navigation efficiency score is 9.0, showing that they have a high expectation for the quick search function. Users in the education industry and public service field have relatively high scores on error handling instructions, reflecting that these users hope that the system can provide more operation instructions.

After analyzing user needs, user needs are mainly reflected in the simplicity of interface design [34-35], the efficiency of the navigation system, and the effectiveness of the error handling mechanism. The expectation for the interface is to be able to quickly find the required functions to reduce the complexity of operations and improve the overall user experience. In terms of navigation, users hope to quickly access various functions through clear paths and shortcuts to enhance operational efficiency. Regarding the error handling mechanism, users hope that the system provides clear error feedback and operation instructions to help them quickly solve problems. These needs provide an important basis for optimizing the user experience of network security software.

Table 1 Demand scores of different user groups

User Group	Interface Simplicity Demand Score	Navigation Efficiency Demand Score	Error Handling Guidance Demand Score	Overall Satisfaction Score
IT Professionals	8.5	9	7.8	8.4
Finance Professionals	8	8.8	7.5	8.1
Education Professionals	8.2	8.5	8.6	8.3
Public Service Professionals	8.4	8.7	8.9	8.4
Manufacturing Professionals	7.9	8.4	7.6	8

2.2 Interface Design and Navigation Optimization

The optimization of interface design is based on the user's strong demand for simplicity and efficiency. Flat design is chosen among many design schemes because it has the advantages of reducing cognitive burden, improving operational efficiency and enhancing interface uniformity. Compared with traditional multi-level design, flat design conveys information with simple graphics and clear levels, eliminates redundant decorations and complex interactions, significantly reduces the user's learning cost, and helps users quickly master the functional layout. Flat design is visually intuitive and clear, which can highlight the core functions of the interface and prevent users from ignoring key operations due to visual interference. In addition, it also adapts to the screen specifications of different devices to ensure the consistency of multi-terminal interface experience, thereby meeting the needs of multi-scenario use. In navigation optimization, the usability of the interface is further enhanced through the combination of breadcrumb navigation and global search functions. Users can quickly complete tasks in a clear path, reduce the possibility of misoperation, and gain a clear sense of direction and control in complex operation processes. This overall optimization effectively improves the efficiency of software use and user satisfaction.

In the flat design of network security software, the color matching adopts a high-contrast color scheme, blue and green are used to indicate the security status, and red is used for warning information to ensure that important prompts are clearly visible. Icon design follows the principle of simplicity and intuition, using linear icons to reduce details and enhance recognition. Sans-serif fonts are used in a moderate size to ensure the readability of the text while keeping the interface neat. The overall layout removes redundant decorations and uses ample blank areas to separate different functional modules, allowing users to quickly focus on core operations, reduce cognitive burden, and improve operational efficiency.

This paper optimizes the interface design based on the results of user needs analysis. In terms of interface layout [36-37], important functions should be highlighted, and interface space should be reasonably allocated to ensure that visual hierarchy is clear. Based on the user's demand for simplicity of interface, the flat design is adopted to reduce multi-level menus and make commonly used functions easily accessible. Flat design is a design method that enhances user experience by reducing the visual hierarchy and complex elements in the interface. It emphasizes the use of simple graphics, bright colors and clear typography to highlight important functions and avoid excessive decorative elements [38]. This design can help users quickly understand the interface structure, reduce learning costs, and make them more fluent when using the software. Flat design also promotes responsive design, making the interface consistent on different devices.

Based on the user's expectations for navigation efficiency, intuitive breadcrumb navigation and global search functions are applied to help users quickly locate the required functions. Breadcrumb navigation is a navigation tool that provides users with their current location and access path, usually located at the top or bottom of the page. It displays the hierarchical relationship from the homepage to the current page through a series of links, and users can easily return to the previous level or higher level based on this design. This navigation method can reduce the user's sense of loss in complex operations, improve task execution efficiency, and enhance the user's understanding of the entire interface structure. The global search function greatly reduces the time and effort users spend searching in multi-level menus by providing a direct way to locate target content. This is especially suitable for scenarios where the needs are clear but the functional structure is unfamiliar. These designs enhance the intuitiveness of operations and the efficiency of task completion, making the user experience smoother and more convenient. The combination of flat design and breadcrumb navigation significantly improves the usability and user experience of the software. An example of interface optimization is shown in Figure 2.

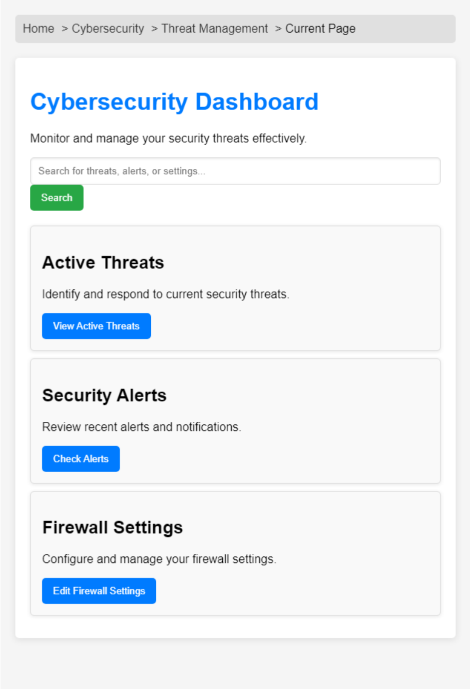

Figure 2 Interface optimization

At the top of Figure 2 is a breadcrumb navigation design that guides users to the target page. If users want to quickly find the content they want, they can also use the global search function to find the corresponding content. The flat design below allows users to grasp the key points of the page at a glance.

When optimizing the interface of a certain enterprise-level network security software, the application of flat design significantly improved the user experience. The software originally used multi-level menus and complex icons. Users had to go through multiple steps when searching and operating functions, which increased the learning cost and the risk of misoperation. By introducing flat design, the interface layout was greatly simplified, unnecessary decorative elements were removed, and simple lines and geometric shapes were used to represent icons to ensure visual clarity. In terms of color scheme, a design dominated by cold colors such as dark blue and light gray was used, combined with high-contrast text colors to make important information more prominent. The fonts were unified in a sans serif style, and the font size was reasonably adjusted according to the importance to ensure the readability and consistency of the text. The blank area in the interface was fully utilized, which not only separated different functional areas, but also guided the user's attention to key operations. The navigation bar was redesigned, and breadcrumb navigation and global search functions were added, allowing users to quickly locate the required security settings and monitoring functions. The display of security prompt information has also been optimized, error prompts have become more accurate and guiding, and the introduction of automatic recovery mechanisms and logging functions has further enhanced the stability and reliability of the system. These improvements have significantly improved users' operational efficiency when handling complex security tasks, reduced the probability of errors, and enhanced users' trust in the software.

2.3 Improvement of Error Handling Mechanism

In network security software, the improvement of error handling mechanism is crucial to optimizing user experience. Users encounter a variety of errors such as incorrect input format, network connection failure, insufficient permissions, software crash or functional abnormality during use. These different types of errors are solved by taking different processing solutions.

When handling user exceptions, for errors with incorrect input format, the system should provide clear error prompts, point out the requirements of the input format, and give examples of the correct input format. The design of error prompts should be based on the core principles of clarity, simplicity, friendliness, and guidance. The prompt content should accurately point out the problem and provide an intuitive solution path. Prompt information should avoid technical terms, use language that is easy for users to understand, and combine visual elements such as icons and colors to enhance the recognition of the prompt. When an input error occurs, the prompt box must directly indicate the correct format requirements and provide real-time feedback on the input status. When the system is abnormal, the error message must include a description of the problem, solution steps, and operation options, while maintaining a unified design style to ensure the readability and consistency of the prompt content. This is achieved by displaying real-time feedback information near the input box, dynamically displaying error information when the user enters, and helping users to correct it in time. In the case of network connection failure, the system needs to detect the network status and display clear connection error information to the user, while providing a retry button and a quick entry to the network settings so that the user can quickly restore the connection. The recommended solution should include prompts to check the network settings or try other network connection methods. For the problem of insufficient permissions, users should receive a specific permission error prompt and specify the details of the required permissions. The system guides users to the permission management interface, allowing users to apply for permissions or provide relevant help documents to ensure that users can understand and solve permission issues. When the software crashes or functions abnormally, the automatic recovery mechanism should restore the user's last operation status as much as possible after the system restarts. This mechanism is implemented by regularly saving the user's progress to reduce the risk of user data loss. The automatic recovery mechanism saves the user's operation status regularly, and records the user's work progress and key system parameters as recovery points in the local cache or cloud storage. When the software crashes or closes abnormally, the system will first detect whether there is an available recovery point when restarting. The selection of recovery points is based on the user's operation frequency and mission criticality to ensure that important data is saved and restored first. The recovery strategy adopts a differentiated storage method, triggering full state saving when the user's operation changes significantly, and supplemented by lightweight incremental updates to balance storage efficiency and data integrity. During the recovery process, the system automatically loads the user's last operating environment and related function configurations through the background to minimize the user's repeated operations. In addition, after the crash, the system guides the user to submit an automatically generated error report, and through the detailed operation data and abnormal environment analysis recorded in the log, it helps the development team diagnose the source of the problem and optimize system performance. The introduction of the automatic recovery mechanism not only shortens the user's recovery time after a failure, but also reduces the frustration caused by data loss, so that users have a higher sense of trust in the stability and reliability of the software. The system should also provide a crash report function to allow users to feedback specific error information so that the development team can make subsequent improvements. The above exception handling settings are shown in Figure 3.

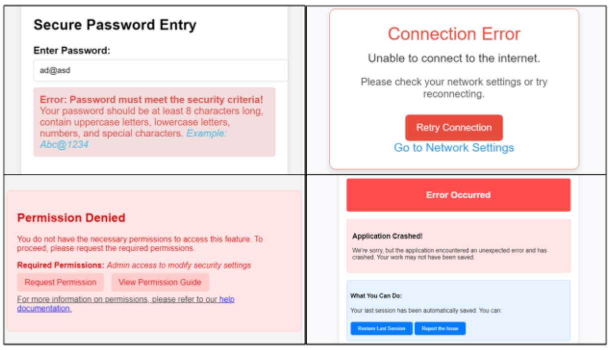

Figure 3 User experience exception handling

For functional abnormalities, the system should have real-time monitoring functions to detect common functional errors and provide prompts to users in a timely manner. The relevant solution provides multiple measures for functional abnormalities, as shown in Figure 4. When a functional abnormality occurs, the automatic retry mechanism automatically tries to perform the operation again after the user-triggered operation fails. If successful, the user is prompted to perform the successful operation. If it still fails, a detailed error prompt pops up to guide the user to the next step. The troubleshooting guide uses concise text and diagrams to help users quickly identify and solve problems. The content covers common causes of failures, corresponding solution steps and inspection items. The system guides users to check relevant settings, restart the program or make necessary environmental adjustments to resolve common functional disorders. In addition, the

software version update suggestion automatically pops up when the user encounters a functional abnormality. The system prompts the user to check whether there is an available update and provides an update log, listing the problems fixed and new features added by the new version, to encourage users to update in time and ensure the use of the latest stable version, thereby improving the overall performance and user experience of the software.

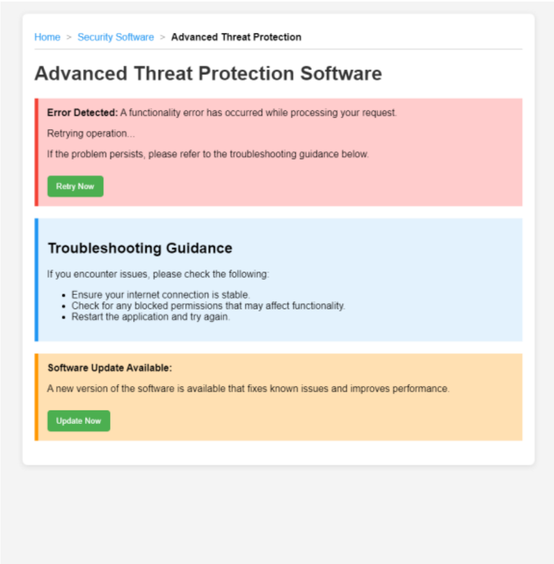

Figure 4 Functional abnormality handling

The logging function is an important means to improve the error handling mechanism. Its content covers key information such as the user's operation path, input parameters, system status, error code and timestamp. The log adopts a structured format and is divided into two forms: plain text and JSON data. Plain text is convenient for quick viewing, and the JSON format provides standardized support for system analysis. The storage method adopts a combination of local and cloud modes. Short-term data is stored locally to improve access efficiency, and long-term data is stored in an encrypted cloud server to ensure data security and traceability. In the event of a failure, the log fully presents the interaction record between the user and the system through a timeline, providing an accurate basis for problem diagnosis and performance optimization, while supporting filtering and keyword retrieval functions to improve analysis efficiency. Through complete records and scientific storage strategies, the log function significantly improves the accuracy of system maintenance and user trust.

3. Interface Design Experiment

3.1 Experimental Setting and Participants

The experiment is conducted in a controlled environment to ensure that the experimental process is not interfered with by the outside world. The hardware configuration used in the experiment is shown in Table 2.

Table 2 Experimental hardware configuration

Hardware	Model	Quantity	Description
CPU	Intel Core i7-10700	1	8 cores, 16 threads, 2.90 GHz
GPU	NVIDIA GTX 1660 Super	1	6GB GDDR6
RAM	16GB DDR4 3200MHz	2	8GB each, total 16GB
Storage	512GB NVMe SSD	1	High-speed SSD
Power Supply	500W 80+ Bronze	1	500W, stable and efficient
Monitor	Dell U2419H	1	24-inch, 1080p, IPS

In terms of software, participants use cybersecurity software optimized by visual communication design and compare it with the unoptimized version. The experimental site is quiet and well-lit to reduce the impact of external factors on participants' operations. The experiment designs several tasks to simulate the scenarios that users may encounter when using cybersecurity software in daily life. The experiment records in detail the completion time, operation steps, user error rate and task completion rate of each task to evaluate the actual effect of the optimized design.

The experiment recruits 50 participants from different industries to ensure the wide applicability of the experimental results. The age range of the participants is 25 to 50 years old. They are all intermediate and advanced users who frequently use computers and have experience in using cybersecurity software. The professional backgrounds of the participants cover fields such as information technology, finance, education and public services. Participants in the information technology field have in-depth experience in using network security software and are able to skillfully operate a variety of tools such as firewalls and intrusion detection systems; participants in the financial industry rely more on security software to prevent data leakage and online fraud, and their experience is focused on protecting sensitive data and financial transaction security; participants in the education field are relatively basic in using network security software, mainly focusing on daily virus protection and improving network security awareness; participants in the public service field have strong compliance requirements when using network security software, and their experience is more focused on ensuring that network systems meet national and regional security standards. To ensure the representativeness of the results, the participants are divided into two groups: the experimental group and the control group. The selection criteria for the control group were consistent with those for the experimental group, i.e., they were randomly selected from the same occupational background, age range, and usage experience to reflect the general characteristics of users in different industries. The experimental group uses the cybersecurity software optimized by visual communication design, while the control group uses the unoptimized version. Participants in the experimental group are able to use simplified interfaces, flat designs and optimized navigation functions during task execution, while the control group uses traditional interfaces and navigation structures. Each participant's experience is evaluated through questionnaires and interview feedback. The questionnaires cover satisfaction with aspects such as interface simplicity, navigation efficiency, and error handling mechanisms. After completing the task, participants are also required to answer open-ended questions to further describe their feelings about the software's ease of use and user-friendliness.

3.2 Experimental Evaluation Indicators

The design of the experimental evaluation indicators includes the following key aspects, aiming to measure the improvement of the user experience of network security software by visual communication design optimization. The user's experience of the software is analyzed by the frequency of function use.

User task completion time measures the impact of interface design and navigation system on operational efficiency. The following formula is used to calculate the average task completion time of users:

$$T_{avg} = \sum_{i=1}^{n} T_i \Big/ n \quad (1)$$

Among them, T_{avg} represents the average completion time; T_i is the task completion time of the i-th user; n is the total number of users.

The number of navigation operation steps is used to evaluate the average number of operation steps taken by users to complete a task, reflecting the simplicity of the navigation system. The formula is as follows:

$$S_{avg} = \sum_{i=1}^{n} S_i \Big/ n \quad (2)$$

Among them, S_{avg} represents the average number of operation steps, and S_i is the number of operation steps of the i-th user.

The user operation error rate is a measure of the number of errors encountered by users when using the software. The formula is as follows:

$$E_{rate} = \sum_{i=1}^{n} E_i \Big/ n \times T \quad (3)$$

Among them, E_{rate} represents the operation error rate, and E_i represents the number of errors encountered by the i-th user within time T.

The speed at which the system responds during the user's operation directly affects the user experience. The average response time of the system can be calculated using the following formula:

$$R_{avg} = \sum_{i=1}^{n} R_i \Big/ n \quad (4)$$

Among them, R_{avg} represents the average response time, and R_i represents the system response time after the i-th user's operation.

Each indicator is calculated and analyzed using actual data from the experiment to help determine the effect of visual communication design optimization, thereby deriving the degree of improvement in user friendliness and ease of use of the design solution.

4. Experimental Results

4.1 Task Completion Time

With the continuous development of network security software, users have put forward higher requirements for the operating efficiency and user experience of the software. The design of the optimization interface affects the time users spend on completing specific tasks, and is also related to the frequency of users' use of software functions. This paper records the time users spend on setting security rules, checking network security status, handling system warnings, updating software, backing up security configurations, and scanning system vulnerabilities before and after optimization, to show the difference before and after interface optimization. The comparison results are shown in Figure 5.

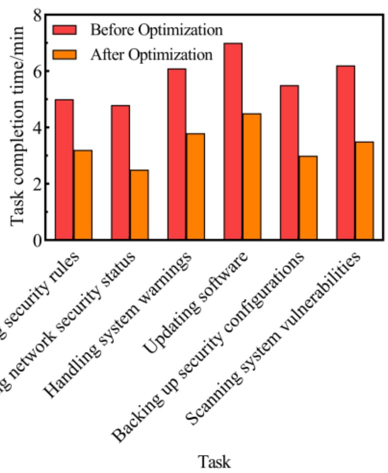

Figure 5 Completion time of different tasks

In the comparison results of Figure 5, the average completion time of all task operations has decreased, among which the average completion time of setting security rules has decreased from 5 minutes before optimization to 3.2

minutes, a decrease of only 1.8 minutes. The task of scanning system vulnerabilities has been reduced by 2.7 minutes, which is the largest reduction among all tasks. This shows that after optimization, the efficiency of users in performing various tasks has been improved, and the user experience has been effectively improved.

4.2 Reduction of Navigation Operation Steps

In modern network security software, users in different industries need different number of operation steps to perform various tasks. This paper selects two industries, IT industry network security engineers and financial industry risk control specialists, to represent professionals and non-professionals respectively. The effectiveness of interface design and navigation structure optimization is revealed by comparing the operation steps of personnel in the two industries before and after optimization in 6 scenarios. The comparison results are shown in Figure 6, where scenarios 1 to 6 are setting security rules, checking network status, handling system alarms, updating software, backing up security configurations, and scanning system vulnerabilities.

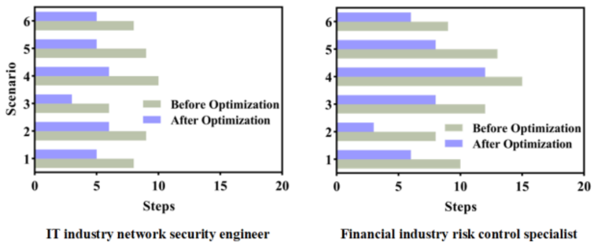

Figure 6 Navigation operation steps

After analyzing the specific data in Figure 6, it is found that for IT industry network security engineers, 3 steps are reduced in setting security rules, checking network status, handling system alarms, and scanning system vulnerabilities; 4 steps are reduced in updating software and backing up security configuration tasks. In the task of handling system alarms, the operation steps decrease by 50%, which shows that the burden of IT industry network security engineers in performing these tasks has been greatly reduced. When checking the network status, the risk control specialists in the financial industry decrease by 62.5% from 8 steps before optimization to 3 steps after optimization. These results show that the optimized interface design can significantly improve non-professionals, reflecting the effectiveness of visual communication design.

The reason for the reduction in operation steps is the simplification of task flow, optimization of functional layout and improvement of interaction methods. For professionals, by reducing repetitive clicks and simplifying the entrance to complex functions, the optimization process is more in line with the needs of high-frequency tasks. For non-professionals, the cognitive burden of tasks is reduced by reducing the complexity of operations and improving guidance. The reduction in task steps also reflects the differences in the optimization of user behavior paths in different scenarios, which is related to the user's familiarity with the function and the difficulty of task operations. The optimization design focuses on user experience, integrates intuitiveness and efficiency into the interface layout and navigation logic, and significantly improves the efficiency of task completion.

4.3 Changes in the Frequency of Function Usage

This paper compares the user's behavior before and after optimization to gain a deeper understanding of the impact of improvements in software interface and operation process on user usage habits. Table 3 shows the changes in the frequency of use of each major function before and after optimization.

Table 3 Frequency of use of each function of network security software

Function Item	Usage Frequency Before Optimization	Usage Frequency After Optimization	Increased in Frequency (%)
Setting Security Rules	150	300	100%
Checking Network Status	120	250	108.33%
Handling System Alarms	90	200	122.22%
Updating Software	80	180	125%
Backing Up Security Configurations	60	150	150%
Scanning System Vulnerabilities	40	100	150%

Table 3 shows that the number of times setting security rules and checking network status are used is the highest before and after optimization, reaching 300 times and 250 times respectively after optimization, which shows that users can find and use these functions more easily. The usage frequency of backing up security configuration and scanning system vulnerabilities increases the most, both increasing by 150%. These data show that the optimization measures have significantly improved the user's utilization of software functions and enhanced the overall user experience.

4.4 User Operation Error Rate

The error rate of different user groups has changed before and after the software interface optimization. Especially in complex tasks, simplifying the interface and optimizing navigation have effectively reduced operation errors. By applying intuitive design, users can find functions faster and the system feedback speed is also improved. To evaluate these optimization effects, this paper compares the changes in error rates of different user groups in tasks. The relevant data are shown in Table 4.

According to the data analysis in Table 4, the error rate of optimized operations has decreased in all user groups and task scenarios. Among them, the error rate of users in the education industry in the task of checking network status has dropped from 13% to 4%, which is the most obvious. Although the error rate of IT practitioners in the task of setting security rules has also decreased, the change before and after optimization is small, only decreasing by 5%. These results show that the optimization of visual communication design significantly improves the accuracy and efficiency of user operations when there are differences in the needs and usage habits of different user groups.

Table 4 Comparison of operation error rates of different user groups before and after optimization

User Group	Task Scenario	Error Rate Before Optimization (%)	Error Rate After Optimization (%)
IT Professionals	Setting Security Rules	10	5
IT Professionals	Checking Network Status	12	6
Finance Professionals	Setting Security Rules	15	7
Finance Professionals	Checking Network Status	13	5
Education Professionals	Setting Security Rules	14	6
Education Professionals	Checking Network Status	13	4

4.5 Error Handling Response Time

The response time of users when performing tasks is an important indicator for evaluating software performance. This paper compares the response time of users when using the software before and after optimization for tasks such as input format checking, network connection, permission checking, and function failure. Figure 7 shows the comparison of response time of task scenarios before and after optimization.

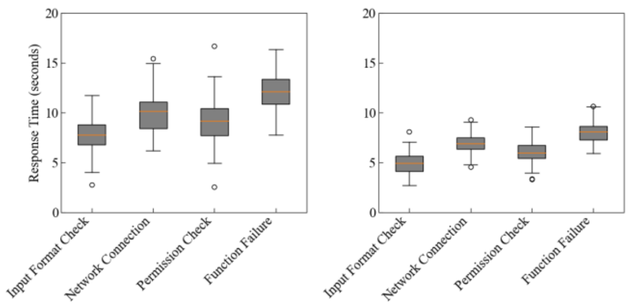

Figure 7 Error handling response time

The left sub-graph in Figure 7 is before optimization, and the right sub-graph is after optimization. Before optimization, the average response time of function failure is 12.2 seconds, which is significantly higher than other tasks, reflecting the complexity and difficulty of processing this link. The response time of each task scenario after optimization is significantly reduced, among which the average response time of input format check is reduced to 4.9 seconds after optimization. Through analysis, it can be seen that the optimized network security software has achieved significant results in improving response speed and enhancing the fluency of users in actual operations.

5. Conclusions

This paper combines visual communication design methods to propose multi-dimensional optimization strategies for the problems of user interface complexity, inconvenient navigation and imperfect error handling mechanism in network security software. In terms of interface simplification, a flat layout is adopted to streamline elements and optimize functional layout, making user operation smoother. Navigation optimization improves user navigation efficiency by adding global search and breadcrumb navigation functions. For the improvement of error handling mechanism, the clarity of error prompts is enhanced, and automatic recovery and logging functions are applied, which significantly improves the error response speed and user experience of the system. The experimental results show that the optimized software has been significantly improved in many aspects. The task time for setting security rules was reduced by 1.8 minutes, and the task time for scanning system vulnerabilities was reduced by 2.7 minutes. In terms of function usage frequency, backup security configuration and vulnerability scanning increased by 150%. The operation error rate among IT professional users was reduced by 50%. However, this study has not yet fully covered the special needs of all user groups. In the future, the personalized needs and behavior patterns of different user groups can be further combined to continuously optimize the user experience design of network security software.

References

[1] Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., & Kambhampati, S. (2020). A survey of moving target defenses for network security. IEEE Communications Surveys & Tutorials, 22(3), 1909-1941.
[2] Peng, Y. (2023). Research on the Technology of Computer Network Security Protection. Journal of Applied Data Sciences, 4(1), 22-29.
[3] Oulasvirta, A., Dayama, N. R., Shiripour, M., John, M., & Karrenbauer, A. (2020). Combinatorial optimization of graphical user interface designs. Proceedings of the IEEE, 108(3), 434-464.
[4] C. Lopes, P., L. Rangel, R., & Martha, L. F. (2021). An interactive user interface for a structural analysis software using computer graphics techniques in MATLAB. Computer Applications in Engineering Education, 29(6), 1505-1525.
[5] Rubins, D., McCoy, A. B., Dutta, S., McEvoy, D. S., Patterson, L., Miller, A., ... & Wright, A. (2022). Real-time user feedback to support clinical decision support system improvement. Applied Clinical Informatics, 13(05), 1024-1032.
[6] Namayala, P., Kondo, T., Manyilizu, M., & Hassan, K. (2023). Practical applicability of user experience capability/maturity models in the development processes of free and open-source software: a systematic literature review. Behaviour & Information Technology, 42(5), 596-623.
[7] Fei, S., Yan, Z., Ding, W., & Xie, H. (2021). Security vulnerabilities of SGX and countermeasures: A survey. ACM Computing Surveys (CSUR), 54(6), 1-36.
[8] Chen, Z., Kommrusch, S., & Monperrus, M. (2022). Neural transfer learning for repairing security vulnerabilities in c code. IEEE Transactions on Software Engineering, 49(1), 147-165.
[9] Samrgandi, N. (2021). User interface design & evaluation of mobile applications. International Journal of Computer Science & Network Security, 21(1), 55-63.
[10] Kobayashi, K., & Suzuki, T. S. (2020). Development of impedance analysis software implementing a support function to find good initial guess using an interactive graphical user interface. Electrochemistry, 88(1), 39-44.
[11] Chen, Q., Chen, C., Hassan, S., Xing, Z., Xia, X., & Hassan, A. E. (2021). How should i improve the ui of my app? a study of user reviews of popular apps in the google play. ACM Transactions on Software Engineering and Methodology (TOSEM), 30(3), 1-38.
[12] Bouchama, F., & Kamal, M. (2021). Enhancing cyber threat detection through machine learning-based behavioral modeling of network traffic patterns. International Journal of Business Intelligence and Big Data Analytics, 4(9), 1-9.
[13] Diederich, S., Brendel, A. B., Morana, S., & Kolbe, L. (2022). On the design of and interaction with conversational agents: An organizing and assessing review of human-computer interaction research. Journal of the Association for Information Systems, 23(1), 96-138.
[14] Li, W. (2024). The Impact of Apple's Digital Design on Its Success: An Analysis of Interaction and Interface Design. Academic Journal of Sociology and Management, 2(4), 14-19.
[15] Ruiz, J., Serral, E., & Snoeck, M. (2021). Unifying functional user interface design principles. International Journal of Human–Computer Interaction, 37(1), 47-67.
[16] Kamaruddin, N., & Sulaiman, S. (2020). A Content Analysis on Interface Design Principles and Characteristics of Tertiary Teaching Aids. Environment-Behaviour Proceedings Journal, 5(SI1), 145-149.
[17] Sharma, V., & Tiwari, A. K. (2021). A study on user interface and user experience designs and its tools. World Journal of Research and Review (WJRR), 12(6), 41-45.
[18] Rembulan, G. D., Akhirianto, P. M., Priyono, D., Pramudito, D. K., & Irwan, D. (2023). Evaluation and Improvement of E-Grocery Mobile Application User Interface Design Using Usability Testing and Human Centered Design Approach. Jurnal Sistim Informasi dan Teknologi, 41-45.
[19] Chen, J., Chen, C., Xing, Z., Xia, X., Zhu, L., Grundy, J., & Wang, J. (2020). Wireframe-based UI design search through image autoencoder. ACM Transactions on Software Engineering and Methodology (TOSEM), 29(3), 1-31.
[20] Xu, Y., Liu, Y., Xu, H., & Tan, H. (2024). AI-driven UX/UI design: Empirical research and applications in FinTech. Academia Nexus Journal, 3(1).
[21] Ilma, N., Hidayat, P. A. W., Setiono, D., & Prilosadoso, B. H. (2021). Role of visual communication design as effective COVID-19 information and education media for community. International Journal of Social Sciences, 4(1), 148-152.
[22] Wang, R. (2021). Computer-aided interaction of visual communication technology and art in new media scenes. Computer-Aided Design and Applications, 19(S3), 75-84.
[23] Yoo, Y. S. (2021). A study on the website structures and content types of online graduation exhibitions for 2020 domestic visual communication design major cases. Journal of the Korea Convergence Society, 12(3), 99-106.
[24] Zhao, Y., & Luo, Y. (2020). Autonomous learning mode based on a four-element teaching design for visual communication course. International Journal of Emerging Technologies in Learning (iJET), 15(19), 66-82.
[25] Andriyan, W., & Anesti, V. (2020). Visual audio communication design on the role of information technology on student life style of universitas raharja. ADI J. Recent Innov, 2(1), 15-24.
[26] Tatineni, S., & Allam, K. (2024). AI-Driven Continuous Feedback Mechanisms in DevOps for Proactive Performance Optimization and User Experience Enhancement in Software Development. Journal of AI in Healthcare and Medicine, 4(1), 114-151.
[27] Hassan, J. (2020). Usability analysis of university websites in Pakistan. Lahore Garrison University Research Journal of Computer Science and Information Technology, 4(2), 13-29.
[28] Goyal, S. (2022). Handling class-imbalance with KNN (neighbourhood) under-sampling for software defect prediction. Artificial Intelligence Review, 55(3), 2023-2064.
[29] Chazette, L., Brunotte, W., & Speith, T. (2022). Explainable software systems: from requirements analysis to system evaluation. Requirements Engineering, 27(4), 457-487.
[30] Akbar, M. A., Khan, A. A., Mahmood, S., & Mishra, A. (2023). SRCMIMM: the software requirements change management and implementation maturity model in the domain of global software development industry. Information Technology and Management, 24(3), 195-219.
[31] Campos, J. C., Fayollas, C., Harrison, M. D., Martinie, C., Masci, P., & Palanque, P. (2020). Supporting the analysis of safety critical user interfaces: an exploration of three formal tools. ACM Transactions on Computer-Human Interaction (TOCHI), 27(5), 1-48.
[32] Datseris, G., Vahdati, A. R., & DuBois, T. C. (2024). Agents. jl: a performant and feature-full agent-based modeling software of minimal code complexity. Simulation, 100(10), 1019-1031.

[33] Jasim, Y. A., & Raewf, M. B. (2020). Impact of the information technology on the accounting system. Cihan University-Erbil Journal of Humanities and Social Sciences, 4(1), 50-57.

[34] Li, X., Li, Y., & Jae, M. H. (2021). Neural network's selection of color in UI design of social software. Neural Computing and Applications, 33(3), 1017-1027.

[35] Deng, X. (2021). The Impact of Good Ui Design. Journal of Frontiers in Art Research, 1, 5-9.

[36] Brie, P., Burny, N., Sluÿters, A., & Vanderdonckt, J. (2023). Evaluating a large language model on searching for gui layouts. Proceedings of the ACM on Human-Computer Interaction, 7(EICS), 1-37.

[37] Jiang, Z., Gao, Q., Wei, Y., & Yin, H. (2021). Mobile application user interface layout features for data-driven design. International Journal of Embedded Systems, 14(5), 421-431.

[38] Potts, C. (2020). Supporting software design: integrating design methods and design rationale. In Design Rationale (pp. 295-321). CRC Press.