Improving Link Analysis Method in User Interface Design Using A New Computational Optimization Algorithm

Cheng-Jhe Lin, Wanyan Yu, Changxu Wu

Department of Industrial and Systems Engineering at the University at Buffalo

Link analysis (LA) method is one of major user interface analysis methods in HCI. However, it seemed that it has several limitations, including negligence of high-frequency, low-link elements in a control interface and underestimation of the total moving distance in real task settings. To improve LA in these limitations, this study developed an algorithm based on the concept of optimizing the total moving distance and the outcome of the algorithm can be quantitatively validated. The findings along with the difficulties faced in this study might provide beneficial reference as well as challenges worth for future studying.

INTRODUCTION

Background

Link analysis. Link analysis is a research method which aims to improving design by examining the relationships and associations between objects and was originally used to investigate the social relationships between people in group situations. (Sandom et al., 2004) Later interests in applying this method have been spawned to information system and Human factors. Numerous researches have been done to employ computer-assisted or fully automatic computer-based link analysis for search engine optimization. (Wasserman et al., 1994) In terms of use of interfaces, link analysis is specifically used to study task sequences or communication links between system components, as a form of task analysis.

Stanton and Young (1999) summarized the procedure of link analysis in ergonomics and suggested that after HTA has been done, the data on components links in simple task movement should be recorded through a link diagram or link table to analyze and manually design a revised layout which minimize the length of links. Wesley et al. (1992) proposed a more detailed five-step link-analysis procedure. Different from Stanton and Young, a “link value” was assigned to each component based on the importance and frequency of the link. Then layout rearrangement followed the principle to minimize link crossings and shorten links.

A different objective of link analysis is proposed by Andrew Sears (1993). He developed a metric, layout appropriateness (LA), to evaluate user interface widget layout. The metric is computed by weighting the cost of each sequence of actions by how frequently the sequence is performed. The optimal LA layout is found by a simple branch-and-bound algorithm with several enhancements. The algorithm does not aim to minimize the moving distance but facilitate searching the LA optimal. Whether such algorithm works for link distance minimization in terms of HCI consideration, however, is unknown.

Weakness of link analysis. According to Stanton et al. (2005), one of disadvantages of link analysis is that the output is not easily quantifiable. In addition, the inter-rater reliability is relatively low. Besides these deficiencies, link analysis only considers minimizing the distance between elements on the target interface. However, the initial distance from original hand position to the target control element should be considered so that the total movement of the control operation can be minimized. To achieve this goal, the frequency and the priority of control position can play vital roles. If we consider only link value, possibly a frequently used elements with relatively fewer links with other elements might be ignored in the beginning of rearrangement and put into marginal space. This will unnecessarily increase the long-term, holistic moving distance of users and deteriorate the time performance.

Objective

The aim of this study is to develop an algorithm to rearrange the interface and validate the proposed algorithm by a case study. A specific case of interface design is chosen for modification by link analysis. Furthermore, the original interface is compared with the modified one through quantitative evaluation method. A step-by-step modification based on the algorithm is demonstrated to show how to quantitatively and objectively rearrange the elements on the interface. At the same time, optimization-based rearrangement will validate modified interfaces in terms of our quantitative evaluators, which reflects the long-term, holistic moving distance during the operation.

METHOD

Essentials of an Optimized Link Analysis

Analysis of the Interface. Before a link analysis could be conducted, firstly, it was assumed that a target interface could be divided into several equal square units, which were called basic units, and each of them had identical area. Secondly, certain Control Elements (CEs), such as buttons or knobs, had similar functions, and they should be assigned into the same Functional Group (FGs) to facilitate the analysis. Therefore, a target interface could be viewed as an area consisted of certain rectangular units, and each CGs could occupy one or several of these areas. In this study, it is assumed that all control elements (CEs) and functional groups (FGs) could fit into either one unit or an area that was composed of even units. Consequently, rectangular shape and total area of a CE or FG could be decided by several units they fit into.
Links and frequency. For link analysis, the link table was the key to get mutual linking relationship between CEs and FGs. However, to minimize the long-term, holistic moving distance while operating an interface, the frequency of using each CEs and FGs should be taken into consideration as well because the moving distance associated with reaching frequent used CE/FGs is responsible for a considerable part of the grand sum of the moving distance in a long-term. Therefore, a task analysis based on the official instruction of the target interface should be conducted so that a general idea of how CEs and FGs were linked with one another could be acquired. Then a survey was designed to obtain frequency of using each CE/FG on the target interface. After the link table and the frequency were obtained, the appropriate priorities to arrange CEs/FGs should be determined as well as the priorities of available positions.

Priority. Once the link values as well as the frequency data of each button were obtained, the question becomes: which CEs/FGs should be arranged into the new interface first, and into which unit the CEs/FGs with the highest priority should be fitted. Considering both of these two perspectives, it was intuitive to arrange the CEs/FGs with the highest priority into the most accessible unit, which has the highest priority of position. In this study, PFG(i) and PPO(i) were defined to represent the priority of the CE/FG(i) and the priority of the position to fit CE/FG(i), respectively. For PFG(i), the higher the using frequency of CE/FG(i), the larger the value of PFG(i). For PPO(i), the higher the accessibility of the unit i, the smaller the value of PPO(i).

Rearranging. Consider a general operation including n buttons. The total moving distance would be:

\[ D(\text{origin}, \text{button}_i) + \sum_{i=2}^{n} D(\text{button}_j, \text{button}_i) \]  

where \( i \) is the operational sequence of each button in the operation and \( D(x,y) \) stands for distance traveled between element \( x \) and element \( y \). The first part in equation (1) represents the moving distance between the origin of movement and the interface, while the second part accounts for the moving distance within the interface. Chances are that if the most frequently used button is arranged into the most accessible unit on the interface, the first part of total moving distance can be minimized. Then the second part can be minimized if the buttons linking to the first button can be arranged into a neighboring area of the first button.

Implementation of Rearrangement Algorithm

Before implementing the initial rearrangement algorithm, our objectives should be restated mathematically. To quantify the objectives, two mathematical objective functions, distance evaluator and accessibility evaluator, were developed to evaluate distance and accessibility, respectively.

The distance evaluator (Z1) was formulated as:

\[ Z1 = \sum_{i,j} L(i,j) \times D(i,j) \]  

where \( L(i,j) \) is the number of links between FG(i) and FG(j); \( D(i,j) \) is the Euclidean distance between the geometric center of FG(i) and FG(j). The accessibility evaluator could be represented as:

\[ Z2 = \sum_i PPO(i) \times PFG(i) \]  

where PPO(i) is the priority of position of CE/FG(i), by taking the average of the position priority values which FG(i) covers, and PFG(i) is the priority of CE/FG(i)

In Z1, the number of links between any two FGs is obtained from the link table. Therefore to minimize Z1 implied to minimize the distance between any two linked CE/FGs, which was the second part in equation (1). Z2 accounted for the first part in equation (1). Since the higher the accessibility of the unit i, the smaller the value of PPO(i), Z2 could be minimized if CE/FGs with higher PFG(i) could be arranged into area with smaller PPO(i), which was more accessible area. To achieve the general goal of minimizing the total moving distance with in an operation, the rearrangement algorithm should incorporate the notion of minimizing two evaluators.

It can be reasonably conjectured that the first part in equation (1) would account for more moving distance because, in general, the distance from origin to the control interface is far greater than the separation between two buttons. Hence an appropriate sequence of rearrangement should be, firstly, putting the most frequently used CE/FGs to the most accessible place to reduce the value of accessibility evaluator, then utilize the link table to find a CE/FG linking to the precisely arranged one most, which potentially contribute to distance evaluator more, and fitting it close to the previously arranged one to minimize the distance evaluator. The detailed procedure of the optimization-based algorithm used in this study was described in the following:

1. If not all CEs/FGs is arranged, find the CEs/FGs with the highest PFG value; fit it into the available area with highest PPO and name the FG as NUI1 (New Interface Unit 1).
2. Find a CEs/FGs that has the highest link value with previously arranged CEs/FGs, assign its NUI number according to the sequence it is picked, and fit it into the unit close to previously arranged CEs/FGs so that the distance between them would be minimized. If two CEs/FGs are found having the same number of links, fit the FG with higher PFG. The position where the FG is place into the new interface is determined by trying all possible positions it can be fit into and select the position, which minimizes Z1 in equation (2). If two positions have the same Z1 value, fit the FG to the position with lower PPO.
3. Repeat step 2. If there is no other CEs/FGs linked with previously arranged CEs/FGs, go back to step 1, until all FGs are fit into the new UI.
4. During step 1 and step 2, if one FG can not be fit into any available position in the new UI, take out the UI around the empty position which has the highest NUI value (the latest arranged one) and fit the current FG into that position. Then continue step 2 or 3. The reason to take the FG which is fit later (with higher NUI value) is because the later a FG fit, either the less impact it has to the other FGs or the less PPO value of its linked FGs. Changing the po-
position of such FG will have the smallest impact to our objective.

RESULTS

Case Study
The control interface on Emerson MW8117W microwave oven (Figure 1a) was selected as our target of link analysis to exemplify the algorithm. At the beginning, the interface was analyzed in accordance with assumptions and principles stated in Analysis of Interface. The outcome was illustrated in Figure 1b. Each unit is separated by tiny lines in Figure 1a could be easily mapped onto a cell or a group of cells in Figure 1b. For example, the number pad should be mapped to the group numbered 17, and the POWER button should be mapped to the group numbered 12. The number in each cell or group of cells was the PFG(i) of corresponding CE/FGs, which was derived from a survey aiming at obtaining the using frequency of each button on the interface. In this case, the accessibility of units is determined according to a top-down, left to right principle. This rule was adopted because in the real environment, the microwave oven was placed about waist height on the right hand side. Accordingly, we assume that the upper area is generally more convenient for users to reach while the left side of the control panel is more approachable than the right side. Consequently the upper-left corner becomes the area with the highest priority while the least priority is assigned to the bottom-right. The priority of units (positions) is shown in Figure 2. This easily accessible area (e.g., upper-left corner) on a UI can be accommodated in other circumstances.

In addition, a task analysis was conducted based on operation instruction of the target microwave oven (Emerson Radio Corp.) It was a good technique to obtain links between CEs/FGs so that we can construct link table accordingly. The link table obtained through this task analysis was shown in Figure 3.

Step-by-step Demonstration

Rearrangement. Given the information of this case, the algorithm suggested in this study was demonstrated in the following by applying the optimization-based algorithm to the microwave case. In the simulation, the number of each button was assigned according to the outcome of sorting the original reference number.

1. FG 14 (+30s) had the highest PFG value - 18 and therefore it was fitted first into the new UI. The position into which it was placed is the left top where indicated the lowest PPO value. According to this step, assign NFG1=14 (Figure 4).

2. FG16 was the only FG which had link with FG14 therefore it was fitted next. Z1 was calculated and the position shown in Figure 6 gave the lowest Z1 value. Assign NFG2=16. (Figure 5)

3. Among the FGs linked with FG14 and FG16, FG1 had the highest number of links and was fitted in the 2nd run of step 2. Because FG1 was linked with FG16, it was placed to the available position where it was most close to FG16. Note that there were two positions where gave
FG1 the same Z1 (the position shown in Figure 6 and the next row of the same columns), but the one shown in Figure 6 had lower PPO therefore it was selected. Assign NFG3=1.

Figure 6. Rearrangement Step 2 – 2nd Run

4. Figure 7-9 showed the 3rd to 5th runs in the second step. When 2 positions to be fitted into have the same Z1, the algorithm fitted the current item into the position with higher priority. Assign NFG4=17, NFG5=13, NFG6=2 in their corresponding runs.

Figure 7. Rearrangement Step 2 – 3rd Run

Figure 8. Rearrangement Step 2 – 4th Run

Figure 9. Rearrangement Step 2 – 5th Run

5. In the 6th run of step 2, FG12, FG3 and FG15 had the same number of links with the previous FG. According to the algorithm to break the tie, FG12 was fitted because it had the highest PFG value among the three. The same strategy could be used to determine which FG to be fit in the 7th-9th run. (see Figure 10-13) Assign NFG7=12, NFG8=18, NFG9=3, NFG10=15 in their corresponding runs.

Figure 10. Rearrangement Step 2 – 6th Run

Figure 11. Rearrangement Step 2 – 7th Run

Figure 12. Rearrangement Step 2 – 8th Run

Figure 13. Rearrangement Step 2 – 9th Run

6. Repeat step 1 and find the unplaced FG with the highest PFG, fit it to the place with lowest PPO. Assign NFG11=6. (Figure 14)

Since all the FGs with links have been fitted, then in the rest steps, the rest FG were fitted in the descending order of their PFG. The fitting sequence of each of them was the following:
A problem occurred when the last FG needs to be fit in because it could not be fitted in any available position in the new UI. Based the algorithm, the positions around the empty position were considered and either FG15 or FG2 should be taken out. Because FG15 had a NFG value of 10 and FG2 had a NFG value of 6, which meant FG15 was fitted after FG2. Therefore FG15 was taken out and FG11 was fitted in. FG15 could be fitted in the last spot in the new interface to complete the whole task. (Figure 15)

Validation. According to the result, the new control interface would be like what is shown in Figure 16b. Consequently, the Z1 values of the old and new interface were 147.27 and 72.32, respectively. And the Z2 values of our old and new interface were 3004 and 3073, respectively. That Z1 value of our new interface was greatly less than that of the old one could be attributed to the algorithm that considered minimizing $Z_1$ the rule of thumb. However, this priority excluded some possible optimization of $Z_2$, so that the $Z_2$ value of the new interface was slightly greater than that of the old one. Proper weightings could be assigned to these two evaluators through considering the range of $PFG(i)$ and $L(i,j)$. In our case, a proper weight for each of them to balance the multiplier could be 3 and 2 to $Z_1$ and $Z_2$, respectively. But even though conservatively the equal-weighting was adopted, the sum of $Z_1$ and $Z_2$ of the new interface was still less than that of the old one.

DISCUSSION

In the new control interface, generally the most frequent used buttons were fitted into the top-left corner, which was the most accessible part in the interface. Some grouped buttons, for example, the stop and the start button, were separated in the new interface due to their difference in using frequency. The reason why the certain buttons were elevated to higher position than other previous grouped buttons, such as popcorn, potato, etc., was similar.

There are some limitations in the study. Firstly, it would be better to develop a computer program to simulate the algorithm. Secondly, whether the algorithm could be generalized to other case should be further tested. Thirdly, it is noticeable that in our new interface, an infrequent button (frozen vegetable) is arranged into a relatively “good” position. This phenomenon happens whenever there is a CEs/FGs cannot be fitted into available space and the algorithm is forced to take out some other prearranged items near the position with high priority. This can be improved by adding some “fine tune” routines into the algorithm.

Based on the experience of conducting this study, software implementation is suggested for future work firstly. This study provides some beneficial ideas to develop such a software package, especially in aspects of how to consider frequency along with the link value to optimize the total moving distance. Also combining additional information into the analysis, for example, which button tends to be the first button in a task, might be valuable information to decide the initial moving distance. Also a frequency column can be integrated into the link table so that the frequency of usage can be well considered. Finally Generalization of algorithm should be improved by taking wider variability of situations into consideration. There are still many simplifications in the assumption of the algorithm waiting for releasing. And this predicts considerable future work to do for improving the link analysis method.

REFERENCES

Carl Sandom and Roger S. Harvey (2004), Human Factors for Engineers, IET Publisher, pp84