Analyzing Procedural Behaviors of Human-Machine Systems using Sequence Alignment Techniques

Yochan Kim
1Korea Atomic Energy Research Institute 150, Dukjin, Yuseong, Daejeon, Korea
yochankim@kaeri.re.kr

Abstract
In what order operators or users use and interact with a system is quite related with the performance of human-machine systems. For example, a system that has complex interfaces or information structures possibly makes the way the users use the system difficult. Thus, scrutinizing the action sequences of the users provides important evidence to estimate usability or reliability of the system. In this paper, we introduce some techniques of sequence alignment to analyze several procedural behavior records. Using these techniques, VPP and PCL measures have been developed to quantitatively estimate the characteristics of the procedural behaviors of operators in a nuclear power plant. We claim that it is possible to apply the techniques to analyze interactions between human and other computer systems. In human-computer interactions, there can be three types of action sequences in users and designers: prescriptively designed action sequences to be followed, actually followed action sequences, and action sequences in the user’s mental model. This paper explains methods to evaluate the complexity of the action sequences or compare the different types of action sequences. It is expected that these methods will support systematically analyzing the usability or reliability of human-computer systems.

Keywords: Procedural behavior analysis, Sequence alignment algorithm, Action sequence, Variability of procedure progressions, Procedural compliance level

Introduction
In what orders operator or users use and interact with a system is quite related with the performance of human-machine systems [1]. For example, a system that has complex interfaces or information structures possibly makes the way the users use the system difficult. Thus, scrutinizing the action sequences of the users provides important evidence to estimate the usability or reliability of the system.

However, many kinds of investigations about the procedural behaviors have usually been conducted qualitatively. In this paper, some techniques of sequence alignment were introduced to quantitatively analyze several procedural behavior records. A sequence alignment, one of the important subjects in Bioinformatics, arranges sequences of DNA, RNA, or protein to identify the relationships between the sequences. Some algorithms of sequence alignment have been employed to analyze the behaviors of operators in nuclear power plants. For example, how compliantly operators follow a procedure was estimated using the Smith-Waterman algorithm, and how variably operators follow a procedure was estimated using the partial order alignment (POA) algorithm.

We claim that it is also possible to apply the techniques to analyze interactions between humans and other interactive systems. In human-computer interactions, there can be three types of action sequences for users and designers: prescriptively designed action sequences, actually executed action sequences, and action sequences in the user’s mental model. This paper explains methods used to evaluate the complexity of action sequences or compare the different types of action sequences with a case study.

This paper is organized as follows. First, sequence alignment techniques are briefly explained. Two algorithms that were employed in analyzing the operator’s behaviors in nuclear power plants are also introduced. VPP (variability of procedure progression) and PCL (procedure compliance level) measures are then introduced in the next section. In addition, how these techniques can be used to analyze interactions between humans and other computer systems is shown. Finally, in the last section, the contributions and related issues of this study are drawn with some discussions.

Sequence Alignment Techniques
Sequence alignment is an important subject in the field of Bioinformatics to identify the relationships among sequences of DNA, RNA or proteins. The arrangement can be depicted by placing sequences in a rectangle and inserting some blanks at each column appropriately. The algorithms of sequence alignment arrange two different sequences and calculate how they are similar. The Needleman–Wunsch and Smith-Waterman algorithms have been commonly used for a sequence comparison [2, 3]. The Smith-Waterman algorithm is a dynamic programming tool that finds a maximum total score based on a substitution matrix and the gap-scoring scheme. The algorithm is as follows:
Procedure compliance Level (PCL)

The PCL measure estimates how a procedure progression is similar to the standard progressions [10]. To evaluate the PCL measure, a procedure progression that an operator performs and another procedure progression that should be progressed in typical situations are extracted. When $S_{\text{operate}}$ and $S_{\text{standard}}$ are the two kinds of procedure progressions, a PCL score is calculated by this equation:

$$PCL = \frac{\text{Similarity}(S_{\text{operate}}, S_{\text{standard}})}{\text{Similarity}(S_{\text{standard}}, S_{\text{standard}})}.$$

To calculate the similarity, the Smith-Waterman algorithm is employed for this measure [3]. [10] gave an example of how a PCL score provides an insight into operator behaviors during a diagnosis procedure.

Variability of procedure progressions (VPP)

The VPP measure evaluates how variously the operators follow the steps of the procedures using step sequences that the operator performs in human performance data [1]. This measure generates a directed acyclic graph using the POA algorithm and calculates the complexities of each node using an entropy-based graph complexity method. The VPP score of a given task is obtained by summing the complexities of nodes relevant to the task in the generated graph. The equation that calculates a VPP score is as follows. In a generated graph by POA, there are $V$ vertices, $\{v_1, v_2, \ldots, v_k\}$, which are procedure progressions. When an edge $E(i,j)$ is the link from vertex $i$ to vertex $j$, it is possible to assign a certain weight, $w_{ij}$, to each of the edges. In this method, $w_{ij}$ is defined as the number of progressions that have a path from vertex $i$ to vertex $j$.

$$\text{Complexity}_i = -\sum_{j=1}^{k} \frac{w_{ij} \log w_{ij}}{W}$$

$$\text{VPP}_r = \sum_{r \in T} \text{Complexity}_r$$

where $k$ is the number of edges that emanate from vertex $i$, $W$ is $\sum_{j \neq i} w_{ij}$, and $T$ is the set of steps in the selected task.

[1] and [11] showed that the VPP measure is related with the operator’s performance in nuclear power plants.

Analyzing Interactions between Human and Computer

Three types of action sequences

It is possible to apply these techniques to analyze interactions between humans and various interactive systems. To do so, we can use three different types of action sequences.

The first is a prescriptive designed action sequence to be followed. This kind of action sequence is determined by a system designer. To obtain the action sequences, a sort of task analysis to analyze the formal behaviors of users should be conducted or a state diagram of the system should be converted into prescriptive action sequences.

The second is an action sequence in the user’s mental model. This kind of action sequence is determined by

Applications in Nuclear Safety Analysis

Recently, two measures have been developed to quantitatively estimate the characteristics of procedural behaviors of operators in a nuclear power plant. Because the operators cope with abnormal or emergency situations using relevant procedures, the sequences of steps in a procedure that the operators follow are used for the measures. Kim et al. explained several features that affect step-by-step following behaviors of a procedure [1].

$$H(i, j) = \max \begin{cases} 
0, & \text{if } a_i = b_j \wedge w(a_i, b_j) = w(\text{match}) \\
H(i-1, j-1) + w(a_i, b_j), & \text{if } a_i = b_j \\
H(i-1, j) + w(a_i, \_), & \text{if } a_i = b_j \\
H(i, j-1) + w(\_), & \text{if } a_i = b_j 
\end{cases}$$

where $a, b = \text{Strings over the Alphabet } \Sigma, m = \text{length (a)},$ $n = \text{length (b)}, H(i, j)$ is the maximum similarity-score between a suffix of $a[1...i]$ and a suffix of $b[1...j]$, $w(c, d)$,

A multiple sequence alignment arranges three or more sequences to identify regions of similarity. When we attempted to obtain an optimal solution for any multiple sequence alignment by dynamic programming, this process becomes an NP-complete problem. Therefore, several heuristics are presented: POA [4], ClustalW [5], T-Coffee [6], DIALIGN [7], MUSCLE [8], and SAGA [9].

The POA algorithm generates a directed acyclic graph (DAG) of all sequences; hence, it reveals the branches and joints hidden among the sequences. Fig. 1 shows the strategy of POA. POA redraws each sequence as a linear series of nodes connected at single incoming and outgoing edges (Fig. 1a). Using a score matrix that contains similarity values of letters, POA aligns two sequences using dynamic programming that finds maximum similarity (Fig. 1b). The aligned identical letters are then combined to form a single node, whereas the others are represented as separate nodes.

Fig. 1. Representation of multiple-sequence alignment using the POA algorithm [3]. (a) Single representation of POA, (b) two sequences aligned using POA, and (c) alignment results of POA.
Y. Kim

prior experience with existing devices or user’s intuition of the given system. To obtain the action sequences, a sort of task analysis for prior experience should be conducted.

The last is an action sequence that a user actually conducted to use the given system. This kind of action sequence is determined by interactions between the system interfaces and user’s mental model. Action sequence can also be obtained by observations or experiments.

![Comparison](image)

*Fig. 2. Analysis of various types of action sequences*

Fig. 2 shows the three types of action sequences and their relationships. There are many ways to analyze the procedural behaviors of a user. By comparing prescriptive action sequences and expected action sequences, how coincidently the system was designed to the user’s expectations can be predicted. By comparing prescriptive action sequences and expected action sequences, how appropriately the user used the system based on the designed orders can be estimated. By comparing the actual action sequences and expected action sequences, how much the user employed his (her) prior knowledge when (s)he was using the system can be determined. Furthermore, the variability of actual action sequences can be compared in terms of the structure of information in the system or the variability of prescriptive action sequences. Many methodologies beyond the examples mentioned above will provide quantitative indices about the important dimensions of usability.

**Case study**

A case study comparing a prescriptive action sequence and an actual action sequence was conducted to show the feasibility of the proposed approach [12]. The target task was forwarding a text message to another friend using a smart phone. The participant had no experience of any smart phone but only experience of some feature phones. The participant was instructed the usage of the five basic tasks and used the smart phone for forwarding a message. The prescriptive action sequence was obtained through a task analysis using OCD [13]. The actual action sequence was also observed by the experimenter. The obtained sequences are described in Table 1.

![Unlock the Screen](image)

*Fig. 3. Directed acyclic graph of the prescriptive and actual action sequences*

To compare the two action sequences, we employed a method of PCL measure. The weights of the Smith-waterman algorithm were assumed as follows: 2 for every matching state \( w(\text{match}) \), -1 for every mismatching state \( w(\text{mismatch}) \), and 0 for other states.

As a result, the PCL score of the action sequence was 0.767. Fig. 3 also shows a directed acyclic graph of the two sequences. The reason that two action sequences are not same was that the participant did not know that the smart phone provides a small contact list when the participant types the name of the recipient.

<table>
<thead>
<tr>
<th>Table 1. Prescriptive and actual action sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prescriptive</td>
</tr>
<tr>
<td>Actual</td>
</tr>
</tbody>
</table>

**Conclusion**

In this paper, a new approach to analyze the procedural behaviors of users was proposed. The described algorithms and applications to nuclear power plants showed the usefulness of this approach. The extensibility of this approach to other applications was also demonstrated with a case study. Sequence alignment techniques facilitate a quantitative or graphical analysis of procedural behaviors. It is expected that the usability or reliability of interactive systems can be analyzed more intuitively or statistically with this approach. For example, the resultant score of the case study can be compared with a score measured
with another smart phone. To do so, more investigations with large samples will be conducted continuously.

Acknowledgements
This research was supported by a Nuclear Research & Development Program of the National Research Foundation (NRF) grant funded by the Korean government, Ministry of Science, Ict & future Planning (MSIP). (Grant code: 2013M2A8A4025991).

References