Constructing User Interaction Behaviors Net from System Log
Hua-Qiang SUN\textsuperscript{1,a}, Shu-Leng DONG\textsuperscript{2,b}, Bing-Xian MA\textsuperscript{1,c,*}
\textsuperscript{1}Shandong Provincial Key Lab of Network Based Intelligent Computing, University of Jinan, Jinan, Shandong, China
\textsuperscript{2}Colleges of Information & Mechanical Engineering, Shanghai Normal University, Shanghai, China
\textsuperscript{a}ise_sunhq@163.com, \textsuperscript{b}dsl@shnu.edu.cn, \textsuperscript{c}ise_mabx@ujn.edu.cn
\textsuperscript{*Corresponding author

Keywords: Interaction Behavior, Statistical Analysis, User Interaction Behaviors Net, System Log.

Abstract. Traditional modeling system methods are mainly on system mechanism and structures. For many applications interaction between user and system is important. How to model and analyze user behaviors is becoming more and more important and it is applied to solve problems of user interest, information service recommendation, network security and so on. This paper introduces an algorithm to extract and model user interaction behaviors from system logs and represent them in the form of user interaction behaviors net. By combining user interaction behaviors net with user log datasets stored in system log, it is possible to describe user behaviors in a much more comprehensive and effective manner. Results show that the algorithm can successfully model user behaviors and supply a new description basis for user behavior feature extraction and identification.

Introduction

With the rapid development of computer and network technology, Internet has been playing an increasingly important role in people's life. How to acquire valuable data and obtain the whole interactive view between user and system, so that companies can benefit from translating data into commercial value. e.g., personalization recommendation, behavior prediction, anomaly detection, etc. Which is a hot potato that every internet enterprise has to face.

Thus far, there has appeared plenty of models that modeling user behavior from different views. The approach described in [1] modeling interaction behavior by combining Petri net with UML. But it is limited to the theoretical analysis, and lacks of practice with use of tools. [2] proposed an automatically learning model of user behavior based on extending existing clickthrough strategies with richer browsing and interaction features, which could adapt to changes in conditions, user behavior patterns and different search settings for general real-world web search. [3] proposed a method that used interaction trace to model system-user dialog, and a prototype tool named LeNDI was set up for reverse engineering. [4] presented $\alpha$-algorithm to extract a process model from so-called “workflow log” and represented it as a Petri net. [5] constructed a framework for lightweight interacting workflow processes. By promoting interactions to support communication and collaboration it was possible to model complex workflows in a more natural manner and expression and flexibility were improved compared to traditional workflow modeling language. [6] described tool-supported techniques that generated LTS(Labeled Transition System) by using of interaction scenarios that developers supported. This technique maybe generate redundant models and need many manual inputs.

From the perspective of modeling composition web services, a new model based on complex network was proposed by analyzing characteristics and evolution process of service software system from the complex system perspective in [7], which provided meaningful means for understanding the behavior of the service system and the evolution law. [8] proposed a framework that uniformly modeled service behavior and QoS (quality of service) by using Petri net and was used to judge the uniformity between service behaviors and QoS when searching and replacing services.
However, most modeling methods pay more attention to analyze system mechanism and do not consider interactive processes between user and system in running process sufficiently. As a result, making those models difficult to understand and analyze for non-professional users. Hence, in order to make behavior model more automatic, intelligent and visualization, we propose a new approach that uses user interaction behaviors net to model and describe user behaviors more efficiently.

The remainder of this paper is organized as follows. First, we introduce some concepts that have relevant with our analysis and present the framework. Then we discuss methods of data acquisition and preprocessing. In Section 4, we elaborate on how user interaction behaviors net is generated. And model user interaction behaviors through the trading process of pharmaceutical E-commerce. Finally, we give a conclusion and conclude with our plans for further research work.

User Behavior Analysis Model

Definition 1. (User Activity Set) Let T be a triple T = (I, O, S), where:
- I defines the set of possible input bindings per activity.
- O defines the set of possible output bindings per activity.
- S records scene in which the activity firing.

Definition 2. (User Trace) Let \( \sigma \) be a triple \( \sigma = (T, F, W) \) where:
- \( T \) is a finite set of activities.
- \( F \subseteq T \times T \) represents the dependency relation. Representing the order of user’s behaviors.
- \( W(f) \) is a function and its range is a finite set of integer, \( f \in F \). \( W(t_i, t_j) = n \) means activity \( t_i \) occurs directly followed by \( t_j \) and this relation appears \( n \) times, \( t_i \in T, t_j \in T \).

Definition 3. (Interaction Incidence Matrix) The casual relations between two activities can be presented in the form of \( n \)-square matrix \( A = [a_{ij}] \), \( i \in [1,n], j \in [1,n] \). Where:
- \( a_{ij} = n \), if \( t_i \) occurs directly followed by \( t_j \) and \( W(t_i, t_j) = n \), \( t_i \in T, t_j \in T \).
- \( a_{ij} = -n \), if \( t_j \) occurs directly followed by \( t_i \) and \( W(t_j, t_i) = n \), \( t_i \in T, t_j \in T \).
- \( a_{ij} = 0 \), otherwise.

Definition 4. (User Interaction Behaviors Net) Let an activity be a vertex, the order between two activities be a directed edge, and each directed edge has a positive integer label meaning weight. Let \( G \) be a weighted digraph, \( G = (V, E, W) \), where:
- \( V \) is a finite set of vertexes.
- \( E \) is a finite set of directed edges.
- \( W \) is a weight label defined in definition 2.

Design Model Structure

We propose a model architecture for user interaction behaviors net generation (Fig 1). First, We get system logs from service host, then information about T can be obtained through data preprocessing. User log classifier then divides preprocessing data into several parts referring to the user ID. To avoid invalid user info, such as too short user trace, user logs are processed by user log filter, and validated data is stored in database. User interaction net builder then provides the processed data to the next step: displaying user interaction behaviors net.

Figure 1. Overview of the model architecture.
Data Acquisition and Preprocessing

Access to raw data is a foundation of the whole system. In general, system administrator can provide user logs, e.g. We can use Log4j component to customize and record log whatever format and information needed. It is important to ensure order between user activities pulled from system logs coincide exactly with order in original system logs. We treat anonymous activities (i.e. activities have no remarkable user label) as solely file to process.

In order to make data structured, facilitate processing as well as display user interaction behaviors net in a clearer and more concise manner, scene variable Time is mapped into successive integer identifier. According to the characteristics of 24 hours one day variable Time is discretization handled, and divided into four periods with different width: morning, noon, afternoon, evening respectively sections. Then the four periods mapping of 0 to 3 consecutive integers. The string type attribute value also needs to be translated into a continuous integer identifier as well. e.g. Operation type can be translated into continuous integer identifier due to the number of operation types in one system is usually fixed. Take E-commerce for example, it generally includes operations such as login, product browse, add to wishlist, add to cart, submit orders, payment, etc. So login can be mapped into integer 0 and product browse can be mapped into integer 1, the rest can be done in the same manner, so payment can be mapped into 5 according to the rule.

Generation of User Interaction Behaviors Net

Algorithm Description

Input: system log
Output: user interaction behaviors net G
Step1: system log preprocessing. System log is divided into several user logs referring to user ID. Anonymous activities are treated as a solely file to process.
Step2: user logs preprocessing. Based on user logs invalid user info is discarded according to the filter rule and validated data is stored in database. Such as table 1.

Table 1. User activity table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>11</td>
<td>User ID</td>
</tr>
<tr>
<td>tid</td>
<td>2</td>
<td>Activity ID</td>
</tr>
<tr>
<td>ti</td>
<td>20</td>
<td>Input parameter</td>
</tr>
<tr>
<td>to</td>
<td>20</td>
<td>Output parameter</td>
</tr>
<tr>
<td>s</td>
<td>20</td>
<td>Scene</td>
</tr>
</tbody>
</table>

Step3: interaction incidence matrix A generating. According to the valid data gained in previous steps, adjacency relations and weight can be presented in the form of interaction incidence matrix.
Step4: user interaction behaviors net G generating. On the basis of interaction incidence matrix A user interaction behaviors net G is generated by utilizing mxGraph technology [10].

Example

Based on the algorithm we have proposed in the previous subsection. We apply it to pharmaceutical E-commerce, and further illustrate its validity and feasibility in modeling user behavior.

The main purpose of pharmaceutical E-commerce is to facilitate users according to their inputs and supply them with medicine. It mainly contains activities such as login, advertisement browse, pharmacy consultant, family medicine-chest, medicine search, add to cart and submit orders, etc.

Consider the user log shown in table 2. It records five activity sequences, and note that each activity has been translated already.
Table 2. User activity sequences.

<table>
<thead>
<tr>
<th>Number</th>
<th>Activity sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;0, 1, 2, 3, 4&gt;</td>
</tr>
<tr>
<td>2</td>
<td>&lt;0, 2, 4, 5, 6&gt;</td>
</tr>
<tr>
<td>3</td>
<td>&lt;1, 4, 5, 0, 6&gt;</td>
</tr>
<tr>
<td>4</td>
<td>&lt;1, 2, 0, 3, 5, 6&gt;</td>
</tr>
<tr>
<td>5</td>
<td>&lt;0, 1, 4, 5, 4, 5, 6&gt;</td>
</tr>
</tbody>
</table>

Combining with this sample, user interaction behaviors net can be generated as follows.

Step1: Classify and store different user logs referring to user ID from system log.

Step2: Identify valid data info by filter rules such as behavioral sequence length or core activity, and store trading information such as user identification, activity name, input parameters, output parameters, time as well as its adjacency activity identification in responding datatable.

Step3: Construct interaction incidence matrix A of user behavior. Matrix A mainly records activity identification, adjacency relationship and weight. Among which, we define the right column data as start point of directed edge and the top row data as end point of directed edge as shown in fig 2.

\[
A = \begin{bmatrix}
0 & 1 & 2 & 3 & 4 & 5 & 6 \\
0 & 2 & 1 & 1 & 0 & -1 & 1 \\
-2 & 0 & 2 & 0 & 2 & 0 & 0 \\
1 & -2 & 0 & 1 & 1 & 0 & 0 \\
-1 & 0 & -1 & 0 & 1 & 1 & 0 \\
0 & -2 & -1 & -1 & 0 & 4 & 0 \\
1 & 0 & 0 & -1 & 1 & 0 & 3 \\
-1 & 0 & 0 & 0 & -3 & 0 & 6
\end{bmatrix}
\]

Step4: Based on matrix A gained in the previous step, user interaction behaviors net is generated as shown in figure 2. As defined in definition 4, each vertex represents an activity, each directed edge represents order between two activities, and an integer value on directed edge represents weight.

Figure 2. User interaction behaviors net G.
Comparing G with matrix A, we can find that the user interaction behaviors net can model user behavior adequately. And habitual operation behavior of an user can be analyzed intuitively through the analysis of net. Additionally, user interaction net G contains other activity sequences besides example shows. For example, G contains activity sequence <0, 1, 2, 3, 5, 6> while matrix A does not contain. However, system process may define that sequence.

Conclusion and Further work

In this paper we propose an user interaction behaviors net generation algorithm. First, we detach and filter logs that belong to different users from system log. Then we store these data in database and present user behavior info in the form of interaction behaviors net. Finally, the effectiveness and feasibility of algorithm is verified by taking a drug trade in electronic commerce platform as an example. By integrating user interaction behaviors net with data sets stored in database, we can carry out intuitive analysis of user behaviors in system running process and get formal analysis basis for further user identification and anomaly detection.

The further work of this paper includes:

1) On the basis of testing algorithm in different practical scenarios, further optimize algorithm and improve its accuracy.

2) From the point of view of actual system operation, combine with user interaction behaviors net and datasets to conduct user behavior prediction, anomaly detection, and enhance system credibility.

Acknowledgements

This work was financially supported by Shanghai Normal University Research Program (SK201317) and A Project of Shandong Province Higher Educational Science and Technology Program (J13LN21) and Young Teacher Domestic Visiting Scholar Project of Shandong Province China.

References


