Appendix A7

Appendix for Chapter 7

This chapter appendix provides an introduction to the construction of a dynamic model based on a one-step transition matrix. We generate and display sample one-step transition matrices for the CDF system, and discuss the tradeoffs involved when using dynamic models of higher order.

A7.1 Dynamic Models

In dynamic studies, user behaviour graphs are commonly used to represent sequences of commands that are given by users. Several dynamic workload model studies use user behaviour graphs ([Fer84], [RK85], [CF86], [CMT90], [CS94]). Figure A7.1 is an example of a user behaviour graph that has three nodes [CMT90]. The numbered circles (nodes) represent the states (or command types), and the directional lines connecting the nodes represent the possible transitions from one state to another. Usually the directional lines are labelled with a probability p_{ij}, which indicates the probability of going from node i to node j.

A separate user behaviour graph is needed to show the behaviour of each user in the system. In a system with many users, such as the system studied in this thesis, such a representation may not be manageable. In systems where there are a large number of users, it is more meaningful to look at user behaviour graphs for user classes. For each user class in our model, the state transitions in the user behaviour graph would go from a command class to another command class.

If we consider each set C of distinct user behaviour graphs, then each graph is characterized by a set of command types, such that the number of command types in a user behaviour graph of type c is k_c. When the current command is type i, then the probability

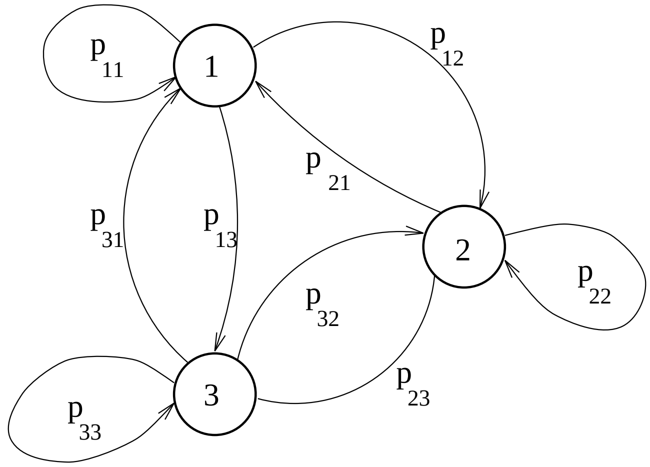

Figure A7.1: Example of a User Behaviour Graph with Three Nodes

that type j will be issued next by a user behaviour graph of type c is denoted by $p_{ij;c}$. Let P_c denote the (k_c x k_c) matrix of one-step transition probabilities $p_{ij;c}$ whose entries are such that

$$0 \leq p_{ij;c} \leq 1, \sum_{j=1}^{k_c} p_{ij;c} = 1, (i = 1, \ldots, k_c), (c = 1, \ldots, C). \quad \text{(A7.1)}$$

The corresponding matrix is

$$P_c = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1k_c} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2k_c} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3k_c} \\ \vdots & \vdots & \vdots & & \vdots \\ p_{k_c1} & p_{k_c2} & p_{k_c3} & \cdots & p_{k_ck_c} \end{bmatrix} = [p_{ij;c}], (i, j = 1, \ldots, k_c), c = 1, \ldots, C. \quad \text{(A7.2)}$$

The p_{ij} values for user graph c can be estimated from the original sequence of commands by counting the total number n_{ij} of transitions from command type i to command type j, and dividing it by the total number of transitions whose initial command type is i, that is

$$p_{ij} = \frac{n_{ij}}{\sum_{q=1}^{k} n_{iq}}, (i, j = 1, \ldots, k). \quad \text{(A7.3)}$$

A user behaviour graph can be modelled as a discrete time Markov chain with the transition probability matrix P_c. This transition matrix is considered a one-step transition matrix, because the probability of the next command is taken into account, but not that of the previous command(s).

The extra overhead that is required to build a model that captures the dynamic components of the workload may not always be worthwhile. This will depend on the goals and the extent of the study. The dynamic components must be studied to determine how much impact they have on the performance measures of concern. [Fer84] states that there are cases in which one should not worry about reproducing workload dynamics faithfully, as the system performance indices in which one is usually interested, do not depend on the order of execution of commands. The question of which transition matrix order should be used is further examined in [CS86] and in [CIS86].

Although it is impossible to determine the suitability of a higher order transition matrix for the model that we have designed in this thesis without running our model, we present the one-step probability transition matrices for eddie and the workstations using the command classes devised in Section 6.5 of Chapter 6. We show the one-step transition matrix for all user classes combined, although in the model a separate transition matrix would be required for each user class.

Notice that in Table A7.1 the column in the transition matrix that has the largest probabilities is column c_4. As the majority of commands (62.2%) were in command class 4, we see higher probabilities for the transitions to this command class. The probability of issuing a class 4 command followed by another class 4 command is the highest probability of all transitions, at 0.714. A similar trend was noticed for the high frequency command class 3 for eddie in Table A7.2.

	p_{ij}							
Clusters	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
c_1	0.235	0.013	0.092	0.465	0.020	0.033	0.096	0.047
c_2	0.141	0.039	0.240	0.401	0.005	0.031	0.115	0.027
c_3	0.193	0.021	0.095	0.471	0.011	0.035	0.145	0.030
c_4	0.104	0.014	0.047	0.714	0.007	0.013	0.081	0.021
c_5	0.156	0.014	0.097	0.380	0.182	0.020	0.105	0.046
c_6	0.250	0.007	0.071	0.474	0.011	0.034	0.116	0.036
c_7	0.123	0.015	0.075	0.553	0.013	0.033	0.141	0.047
c_8	0.212	0.013	0.097	0.440	0.011	0.024	0.132	0.071
Freq	5722	584	2605	25245	507	812	3864	1218

Table A7.1: One-Step Transition Matrix for Workstations

In general, the diagonal of the matrix (i.e., c_{ij}, where $i = j$) often had high probabilities compared to the other probabilities in the same column (i.e., p_{ij} where $i = j$ was usually high relative to probabilities p_{hi} where $h = 1$ to k). This provides some insight into the workload, as it indicates that a command of a certain type was often followed by a command of the same type. Further examination of the transitions for particular commands may provide even more insight into the dynamic properties of the workload.

	p_{ij}			
Clusters	c_1	c_2	c_3	c_4
c_1	0.141	0.198	0.488	0.173
c_2	0.107	0.304	0.353	0.236
c_3	0.046	0.110	0.717	0.127
c_4	0.095	0.180	0.496	0.229
Freq	1269	3067	10626	2936

Table A7.2: One-Step Transition Matrix for Eddie