Performance Estimation Assignment

Martin Höst

1. Introduction

In the architecture phase of the development there is a need to estimate the performance of the implemented system. This estimation can be done in a number of ways, and the objective of this study is to evaluate and compare some different estimation methods.

We would be very grateful if you could carry out estimations as described below in this document.

The data that you provide will be treated anonymously. We will not present what you as an individual have answered. We will however present the conclusions on the validity of the estimation approaches and conclusions on what factors affect the validity (experience, etc.). We will take measures to prevent that it is possible for anyone to conclude what any individual has answered.

Your assignment is to estimate a number of execution times in a program. You are given the design of the program through Message Sequence Charts (MSC) and you may use all your knowledge of the program when you do the estimation. However, you should not look at any source code or detailed design.

2. Introduction to the program design and estimation

2.1 Design representation

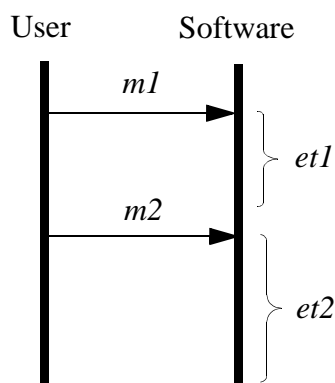The program is presented in MSCs, as the small example in Figure 1.

FIGURE 1. MSC example

In Figure 1 there is one user and one software system. Message *m1* is sent from the user to the software system, resulting in program execution with execution time *et1*. The next signal is *m2,* resulting in execution with execution time *et2*.

2.2 Estimation example

If estimation is to be carried out based on the MSC in Figure 1, the result could be as in Table 1.

TABLE 1. Estimation example

Task	Execution time (Relative to execution time X)
et1	*3.2*
et2	*0.8*

In this case the expert was first presented with function *X*. The expert then estimated that *et1* was 3.2 times longer that function X and that *et2* was 0.8 times as long as function *X*. Notice that you do not have to choose integer values. It is perfectly alright to choose values such as 3.2, 0.8 etc.

2.3 Program implementation

The intention is that it should not make a large difference how the program is implemented, since you should only give relative answers.

You should estimate the execution time of the tasks as if they were executed without any other tasks in the system. That is, you should estimate the execution times of "the LOC written by the programmer" without considering the time it takes to handle signal lists, real time process switching, etc. All estimations concern a program managing a digital telephone switch. In a normal telephone switch the CPU could be involved in transferring the digital representation of the communication. This is however not done by this switch.

3. Estimation

3.1 Introduction

1. Name:_____[your answers will be treated anonymously]

2. Email:_____

3. I have followed the "PUSS-kurs" as a student:
   ( ) Yes, _____years ago
   ( ) No

4. I would rank my knowledge of the exchange in the "PUSS-kurs" as follows:
   ( ) I know every detail of it
   ( ) I know as much as a product responsible[1] student knows after the course
   ( ) I have an overall knowledge of the system
   ( ) I have no knowledge of it

5. I have been teacher in the "PUSS-kurs" ____ times. My roles have been:
   ( ) course responsible, ____ times
   ( ) "design expert", ____ times
   ( ) reviewer, ____ times
   ( ) other:_____

6. How often have you predicted software execution times?
   for the last year:_____
   for the last 5 year:_____
   ( ) it is part of my daily work
   ( ) I am an expert in the area
   ( ) I have only done some work related to this area
   ( ) I have never thought of software performance when I have been programming

---

1. I.e. a student from the "Systemgruppen"

3.2 Part 1

You should carry out estimations for the MSC in Appendix A, i.e. the PUSS-system in its original form. This is the system that is given to the students in the beginning of the course.

In this case we say that the execution time of task 1 is 1 time unit. You should estimate how many time units that are required for the other tasks relative to task 1. For example, if you think that task 2 requires twice as much as task 1, you write "2" for task 2. If you think that it requires half as much, you write "0.5". Fill out the form in Table 2.

TABLE 2. Estimation form, part 1

Task	Relative value (compared to task 1)	Explanation
1	1	Starts after OffHook from User A. Dial tone is sent to User A.
2		User A has pressed the first digit in the phone number of User B. This means that the dial tone should be removed.
3		User A has pressed the second digit in the phone number.
4		User A has pressed the third digit in the phone number.
5		User A has pressed the fourth (and last) digit in the phone number. This means that ringing should be started at User B and that User A should receive a tone that indicates that ringing has started.
6		OffHook from User B, which means that ringing should be stopped at User B and that the tone at User A also should be stopped. The two parties (A and B) are now connected and they can talk to each other. In a normal telephone switch the CPU could be involved in transferring the digital representation of the communication. This is however not done by this switch, and that task should consequently not be considered.
7		OnHook from User B
8		OnHook from User A

3.3 Part 2

In this case you should carry out estimations for the same MSC as in Part 1, i.e. Appendix A. There is one important difference:

- This concerns the system when the following services have been implemented: charging (debitering), call fetching (ta samtal), call forward (medflyttning), and management (underhållsfunktioner).

You should use the same estimation technique as in Part 1. In this case too we say that the execution time of task 1 (of this MSC) is 1 time unit. Fill out the form in Table 3 based on Appendix A.

TABLE 3. Estimation form, part 2

Task	Relative value (compared to task 1)	Explanation
1	1	Starts after OffHook from User A. Dial tone is sent to User A.
2		User A has pressed the first digit in the phone number of User B. This means that the dial tone should be removed.
3		User A has pressed the second digit in the phone number.
4		User A has pressed the third digit in the phone number.
5		User A has pressed the fourth (and last) digit in the phone number. This means that ringing should be started at User B and that User A should receive a tone that indicates that ringing has started.
6		OffHook from User B, which means that ringing should be stopped at User B and that the tone at User A also should be stopped. The two parties (A and B) are now connected and they can talk to each other. In a normal telephone switch the CPU could be involved in transferring the digital representation of the communication. This is however not done by this switch, and that task should consequently not be considered.
7		OnHook from User B
8		OnHook from User A

3.4 Part 3

In this part you should estimate the execution times of the MSC in Appendix B. The MSC describes the ordering of "call forward", i.e. the user requests that any incoming call should be re-routed to a new telephone number. In the system the following services has been implemented: charging (debitering), call fetching (ta samtal), call forward (medflyttning), and management (underhållsfunktioner).

The call forward service is ordered by the following sequence:

```
* 1 * new_telephone_number #
```

You should apply the following technique for every estimation:

1. Determine which execution time from Part 1 (i.e. execution times 1-8 from the "normal telephone call" with the basic system) that is most similar to the execution time that you want to estimate. Write down which similar execution time you have chosen in the form.

2. Estimate the relative execution time compared to the chosen "similar execution time".

For example, if you think that task 1 of this system requires 4.5 times as much execution time as task 2 of the system in part 1, then you write "2" on line 1, column 2, and "4.5" on line 1, column 3.

Fill out the form in Table 4.

TABLE 4. Estimation form, part 3

| Task	Similar execution time (1-8)	Relative value	Explanation
1			Starts after OffHook from the user. Dial tone is sent to the user.
2			The user has pressed "*". Dial tone is stopped.
3			The user has pressed "1".
4			The user has pressed "*".
5			The user has pressed the first digit of the telephone number that incoming calls should be forwarded to.
6			The user has pressed the second digit of the telephone number.
7			The user has pressed the third digit of the telephone number
8			The user has pressed the fourth and last digit of the telephone number.
9			The user has pressed "#". Dial tone is sent to the user.
10			OnHook from the user.

Thank you for your effort!
/Martin

Appendix A, "Normal telephone call"
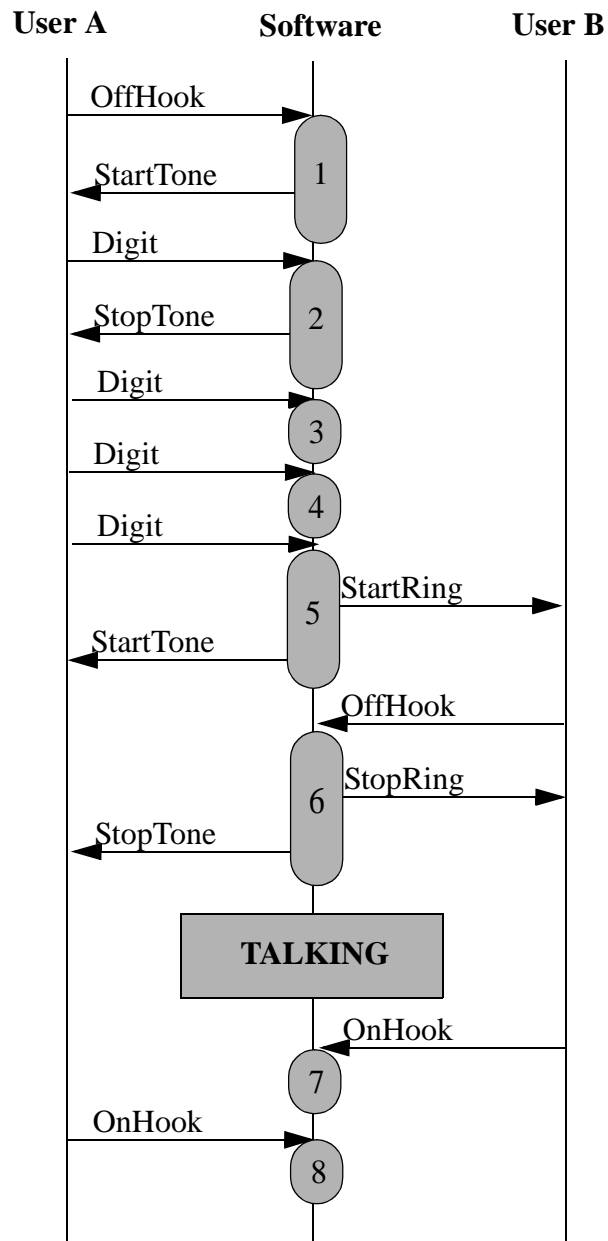
In the MSC the shaded boxes are of different length. Some are "short" (e.g. 3) and some are "long" (e.g. 5). This does not show how long the execution times are. The reason that the lengths are different is that the figure should have a good layout.

Appendix B, "Ordering call forward"
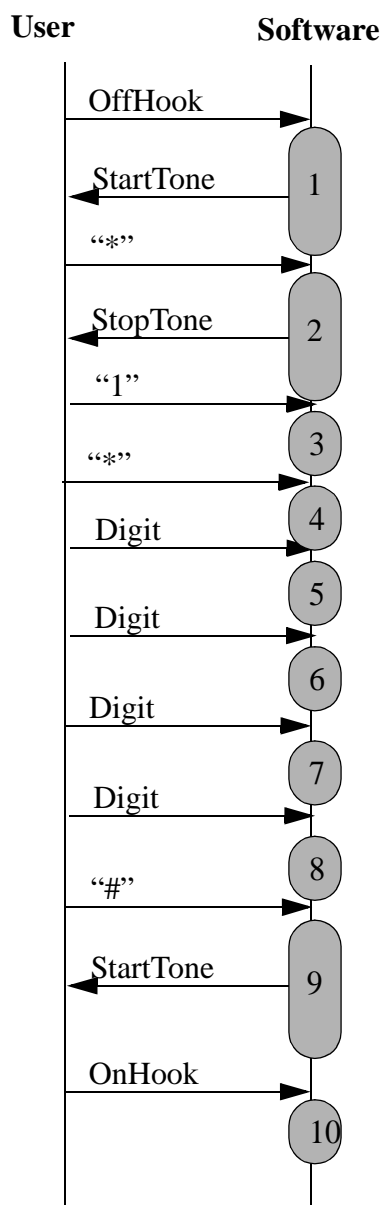
Call forward is ordered by the following sequence:

```
*1* new_telephone_number #
```

In the MSC the shaded boxes are of different length. This does not show how long the execution times are. The reason that the lengths are different is that the figure should have a good layout.