

EE 5903 – CA 1 Exercise

Report

The software design for the apple packaging process is divided into three major threads, which are shown in figure 1. First of all, there is the camera thread which is responsible for taking a picture of every apple which is passing by the camera. Second, the image processing thread which is represented by the image processing unit, which evaluates apples based on the taken image and returns GOOD, BAD or UNKNOWN. Finally, the actuator thread discards apples, which are labelled as bad. Each process is running as long as there are apples on the processing line.

In order to ensure interprocess communication among the different threads there is a message queue in between the camera and photo processing threads and the photo processing and the actuator threads. The first message queue gives the photo processing unit access to the starting time of the photo process, which is repeated in every step for accuracy reasons, and the image data of the photo. The second message queue contains the apple quality and also the starting time.

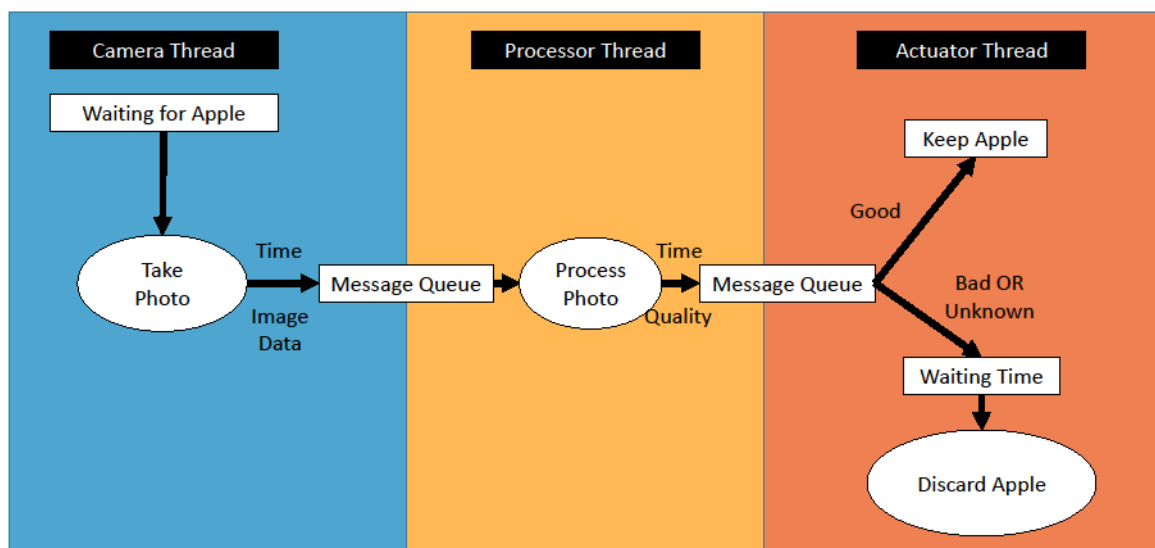


Figure 1: Software design (DFD) – Apple packaging

The camera thread starts by waiting for an apple to reach the tolerated area for a picture. Once an apple is in the necessary position the time (including milliseconds) is saved and a photo is taken. The photo and time are put on the message queue and the camera is set to wait, so that it does not take multiple pictures of the same apple.

Once the image data is read from the queue by the image processing thread, it starts to identify the quality of the apple on the picture. As soon as the quality is determined the time and quality data are put on the second queue.

This queue is read by the actuator thread. The starting time of the whole process is used by this thread to calculate how long the process has already taken. This information is very important since this enables the actuator to decide whether it is able to react to the processed information or not. Because of the position of the actuator in relation to the camera an apple needs five seconds to travel from the camera to the actuator. This means if the processing time takes over five seconds the actuator is not able to react to the following apples (dependent on how many more seconds the process actually takes).

There are two things to do when the process time is greater than five: First, delete messages from the queue about apples which can't be processed. Second, discard apples which weren't processed. I

discard these apples because I rather sacrifice a few good apples than risking one bad apple turning many good apples into bad ones. This also seemed to be the better way to handle it while testing the system. To be able to discard these apples the system has to know when the next apples are arriving which is calculated using the starting time of the process. I also wanted to implement that apples will be discarded as long as the message queue is empty, because this would mean that there are apples which won't get processed in time. Unfortunately, my implementation had errors which is why I left out this part of the code, but this is definitely another improvement for the current system.

If the process time is smaller or equal to five, the actuator can react to the arriving apple by waiting five seconds minus the processing time. Using these two cases I get better results than just applying the handling of apples which don't take as long to process.