

CS2310 Software Engineering Final Report

Junhui Chen

Background

This whole project is based on SIS (Slow Intelligent System). SIS is about taking things slow and taking more factors into consideration before trying to make any decisions. There is nothing wrong pursuing performance but it takes time to figure out the best solution based on mutable factors. And that's the reason why it's important to be adapt to the circumstances.

SIS system dedicates to provide comprehensive services for patient, especially elders, on the top of their physical conditions. Since a person in poor health can't take care of himself as normal, it's necessary to aid him in each way possible. But hospitals don't have enough human resources to achieve high standard health care for each individual, so at least some part of the work should be done by machines if possible. Then again, we need to make sure that machines understand how to do the job. For monitoring the condition of a patient, a system needs various sensors, and each sensor may have different features, structure and also limitations. How to efficiently utilize the information from sensors and send out alerts only when necessary is difficult, since an experienced medical worker could make wrong decisions.

But we can keep collecting sensor data from daily use, and recognize patterns for emergency, so accuracy of future predictions can be improved. But for a medical worker to make a decision, he should refer to multiple factors to get the grasp of the whole picture, one-sided view is almost doomed to cause problems, and that's how machines should do. Instead of processing data from each source separately, it is way better to consider them as a whole when we try to find a certain pattern, since aspects of a patient's condition are correlated to each other.

Kinect sensor & Kinect component

As a part of this system, Kinect sensor and Kinect monitor are crucial since they are by far the only way to detect motions of the patient. But the detection of a fall is not predefined with Kinect, so we have to implement our own algorithm.

Fall detection

To achieve fall detection, we should at least know the rough real-time position of the patient, and this is covered by Kinect sensor, Kinect sensor can track a patient's skeleton, and a skeleton consists of more than 20 joints. Once we get a series of skeletons, we can easily extract the coordinates of joints. A frame is generated for each time interval, so we can calculate the difference between 2 frames, even the speed of movement, since we know the time of each frame get generated.

But how to achieve high accuracy? Since Kinect is not specially designed for medical purposes, it has certain limitation such as

- Before a head is detected, Kinect sensor can't detect a person
- Complex environment can generate noise for detection, even non-human objects can be recognized as human beings
- Sufficient indoor lighting is crucial to even be able to detect anything
- If multiple people present in front of Kinect sensor, we need to treat each separately
- A fall should happen during a very short period of time, so we have to differentiate that from a person trying to lay down or squat in a relatively slower speed

I've tried several simple approaches, each has its own deficiency.

I tried to track multiple joints at the same time, and calculate the average difference with respect to time elapsed. But first I need to make sure a person is not running towards the sensor, that will give the sensor the impression that person is falling since technically the skeleton is tumbling down in a fast speed. Then we need to deal with depth information. If a person is running towards the sensor, then the depth will lessen rapidly, so this case should be ignored. But this approach has a major problem, if the head is by chance out of the frame, no matter how hard a person is falling down, the sensor won't be able to detect that.

And then I tried to use an existing algorithm developed by an undergraduate student. He simply solely utilized the position of the head, and it did minimize the possibility of false positive, but also the possibility of true positive too, since he only focused on one joint of the whole skeleton and ignored others completely.

So, I tried to combine these two approaches together, the detection will begin at the moment when a head is tracked, but instead of using only the position of the head, we utilize positions of other joints together to detect the motion of a patient. The position of the head will have the highest priority, but when that is not available, or doesn't give a high credibility, we can still make assumptions based on other information. We currently don't deal with multiple people.

Extract skeletons & communication with monitor

To be able to extract skeletons, we need to utilize Kinect SDK provided by Microsoft. We can hook a listener to Kinect sensor and get notified whenever a skeleton frame is ready, then we can process it. But just for implementation details, since we implement the SIS using Java, but official Kinect SDK doesn't provide Java as an option, we have no choice but to set up a Socket connection between Kinect monitor and Kinect sensor. And the messages being passed between should follow the standard of SIS too. (Message standard and system mechanism will be explained later.)

Kinect monitor communicates with other components

A Kinect monitor is a component that accept a series of messages from Kinect sensor and sending out alerts to certain components when an emergency happens, and this is where the fall detection algorithm lies.

But fall detection is meaningless if it can't be propagated and acknowledged by other components. We can send this to Uploader which is the component that is responsible for collecting information from all other components and informing the medical worker in charge, and possibly building a knowledge base along the way. We can send this to other monitors that each is responsible for the monitoring of a type of sensor (possibly a group of sensors), so that they can make more accurate decisions. We can also receive information from other monitors and work in the similar way as well.

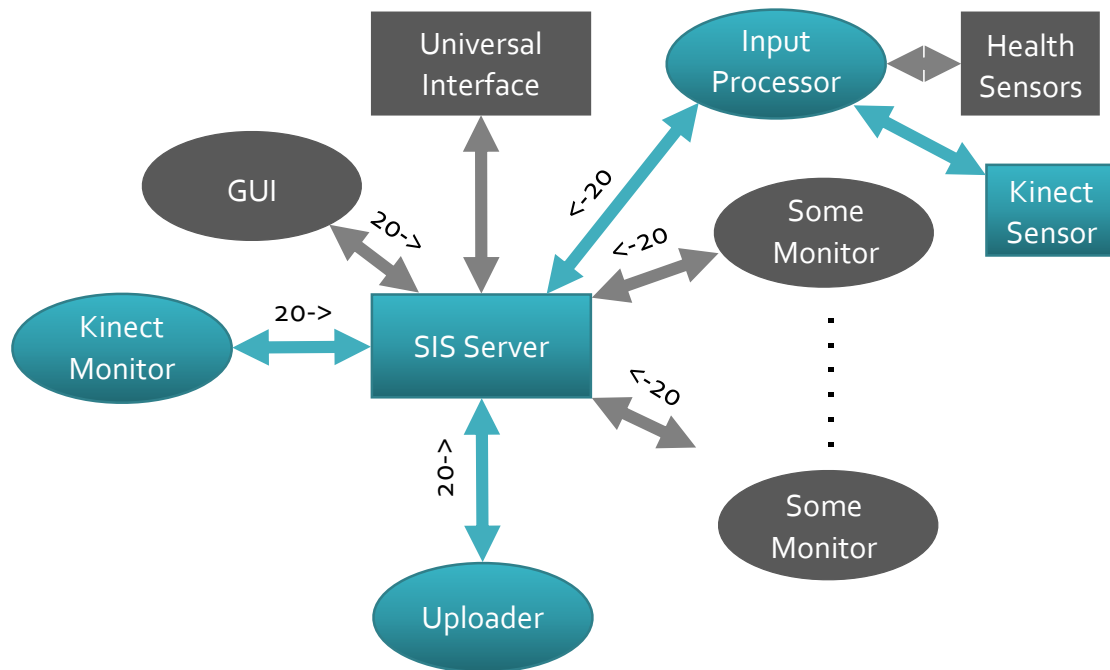
Why do monitors need to communicate with each other? Let's say, if a patient falls onto the ground, chances that he is in dangerous condition, but we can't be totally sure just based on that, but if Kinect monitor also receives alerts from EKG monitor, which means this patient fell onto the ground meanwhile suffering from irregular heartbeats. Now our predication can be way more confident.

Message standard & system mechanism

To make the whole system work, a proper mechanism is crucial. To achieve flexibility, each component should be able to leave and join the system at runtime, and this should be totally controllable by users. Also to achieve security, we need to make sure each message is passed to only target components, and each component can only sends out messages agreed by the system.

Then we will need to set up component step by step by the book, and only in this way that a component can work normally in this environment.

Registration of a component



Before any component starting to work, it must inform its existence to the SIS system by telling all its specification:

- Name of the component
- Input messages (messages that are accepted by this component)
- Output messages (messages that are sent out by this component)
- ...

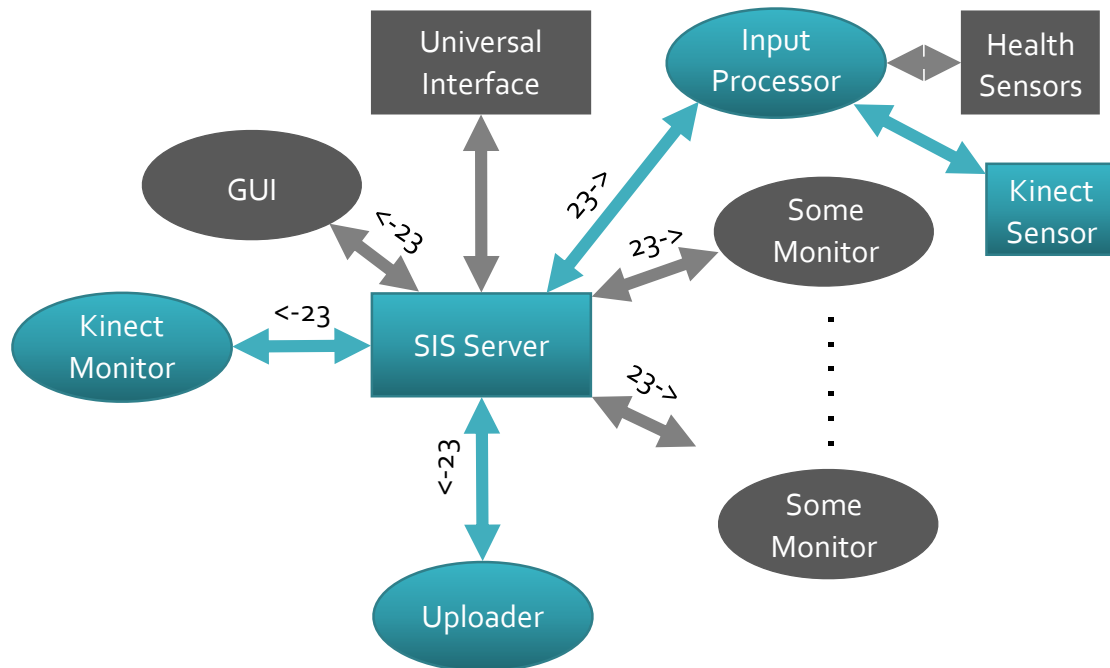
In our case, we need to create 2 components, Kinect Processor (conceptually a part of Input Processor) and Kinect Monitor.

Kinect Processor will take in raw data input from Kinect sensor and generate Kinect Reading message (message 43) as the output.

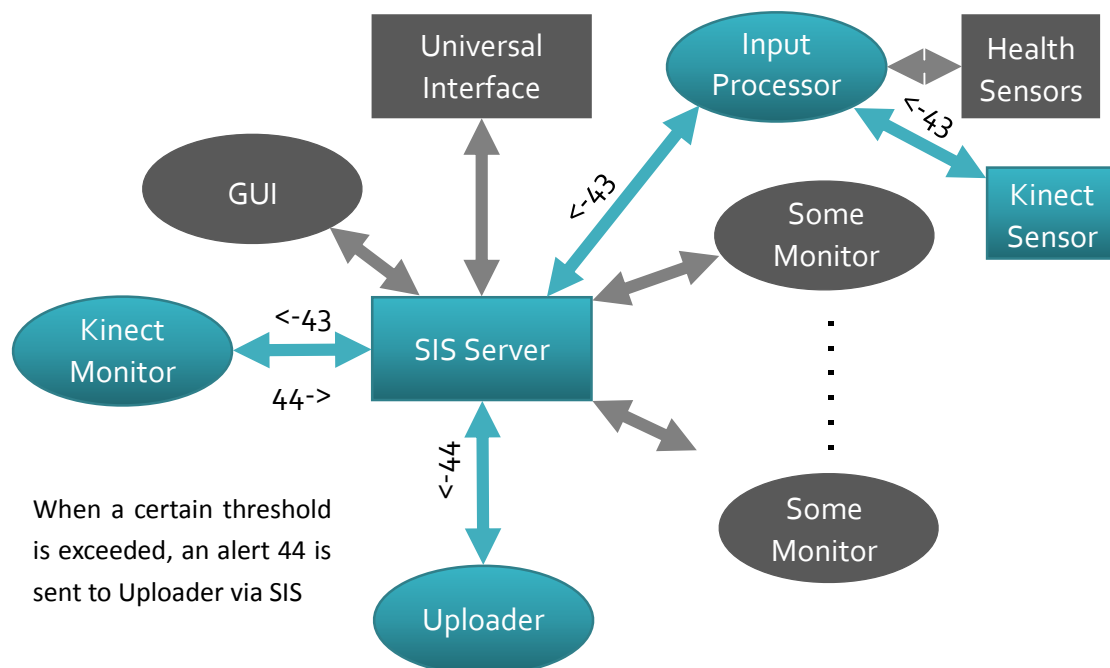
Kinect Monitor will take in Kinect Reading message (message 43) as the input and generate Kinect Alert (message 44) as the output.

Acknowledgement from SIS

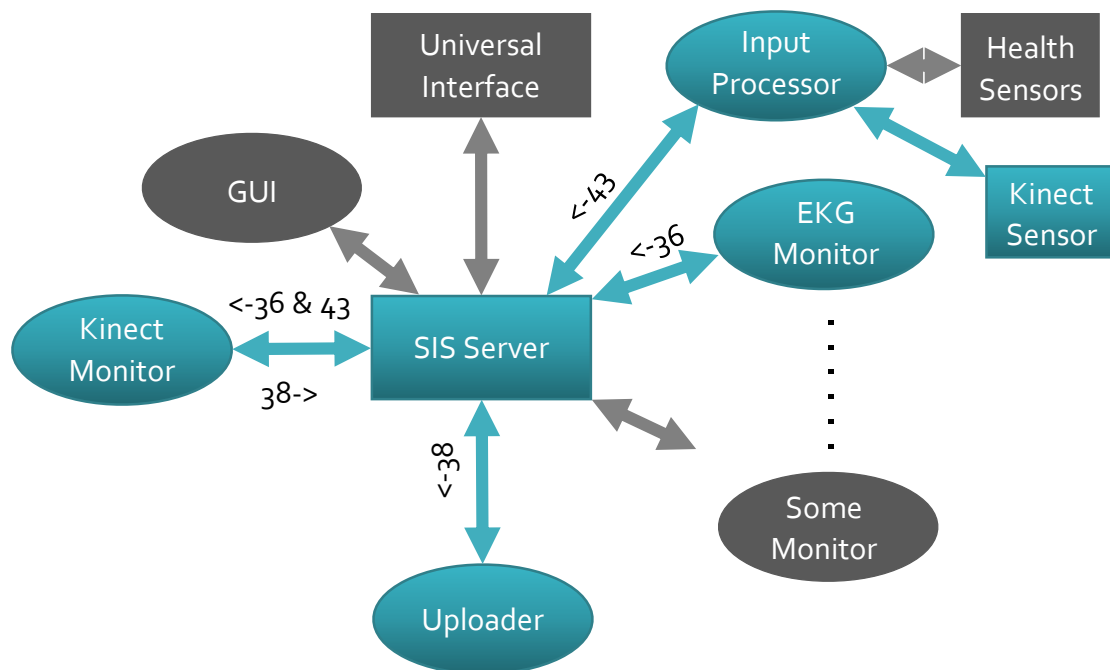
Once there is an agreement, the SIS system will send back acknowledgement (message 23), when a component receive this message, then it will start to function.



Kinect Reading / Kinect Alert



Complex communication



When Kinect Monitor detect alerts from other components, it will send out higher level emergency alerts.