Assignment 3
Adaptation Strategy - analysis, design & implementation

Fundamentals of Adaptive Software 2023

Group: BSN 1-1
Member names: [redacted]
Emails: [redacted]
Student numbers: [redacted]

December 24, 2023

Contents

1 Introduction

In the health domain, the implementation of self-adaptive systems becomes increasingly rewarding to handle vast amounts of data. Introducing a self-adaptive system in healthcare has the potential to improve the quality of health services in terms of efficiency and personalisation, while being more economical to develop at a large scale. For this project a self-adaptive system for wearable technology is explored using the Self-Adaptive Body Sensor Network (SA-BSN) by Gil et al. [1].

The BSN collects data on the state of the patient, the environment and its goals at run-time to resolve uncertainties and satisfy its goals with minimal human interaction. BSN simulates a patient, six wearable sensors as well as a noise injector using ROS [2].

In the context of the wearable health domain, adaptive systems can play a significant role in optimising resource allocation. They dynamically distribute resources based on both the patient's needs and the environment, maximising its utility as well as battery life.

2 Overall System Description

The BSN exemplar in an adaptive system that is deployed in the domain of healthcare, to enhance an effective patient monitoring and dynamically adjusting its parameter based on patient conditions and varying environment, eliminating noise from sensor data, and handling unforeseen situations. While this system proffers valuable insight into the patient's health status, challenges are faced in managing energy consumption and reliability in an uncertain environment, as such this system was introduced to help manage these uncertainties. The architecture of the BSN design can be classified into 3 layers.

1. *Sensor layer* - This layer uses the sensors to monitor in real-time the health situation of a patient and feed the same back to the central hub through the communication layer.

2. *Communication layer* - This layer relays the monitored data to the central hub nodes and the decision layer.

3. *Decision layer* - This layer is tasked with the responsibility of dynamically adjusting the sensor sampling rates and data transmission parameter based on real-time data available and the current patient's requirements needs.

2.1 Noise injection mechanism

The system introduces a noise injection to simulate the real-world scenario, by randomly introducing noise to fluctuate the sensor measurements.

2.2 Sensor simulation

The BSN currently deployed six sensors to provide near world realistic data for testing and evaluation of the system, these sensors monitor various vital signs such as blood pressure, heart rates etc.

The managing system comprises strategy manager and strategy enactor, collaboratively they are responsible for implementing controllers to deal with adaptation issues. The strategy manager estimates the reliability and cost. The strategy enactor implements the controls, it is responsible for applying the adaptation strategies to achieve the desired quality of service. While the managed system uses the sensors to monitor the patient's health status and reports the vital signs reading to the central hub. The knowledge-based system houses the necessary data needed to perform

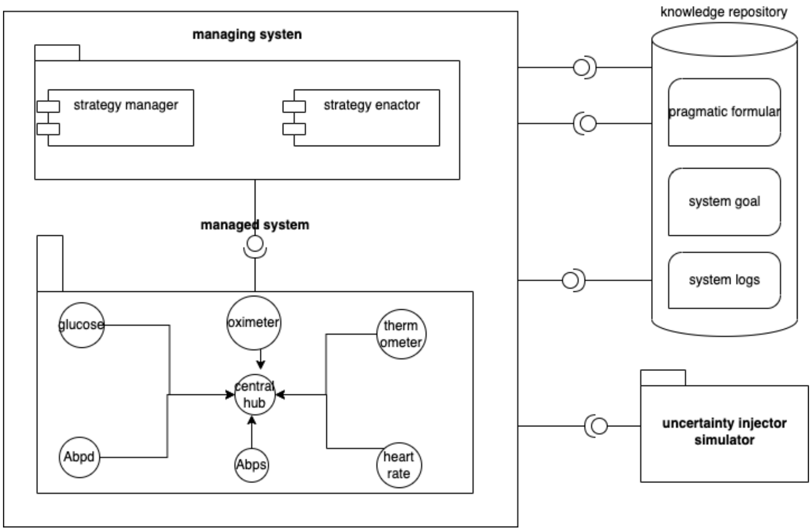


Figure 1: Mape-K reference architecture of BSN

adaptation and the simulator module randomly injects noises to the sensor's data simulating real world scenarios.

3 Analysis of Uncertainties

We analyse the uncertainties we identified in the system using the uncertainty template, as follows:

Uncertainty 1	
Field	**Description**
Name	Dynamic Health Condition
Classification	Run-time/execution phase
Context	If a sensor has a high polling rate it uses more energy. During normal operation, a high polling rate is not desirable. Instead, the self-adaptive system could temporarily increase the polling rate of specific sensors to improve accuracy and lower the polling rate again when the scenario passes. One such scenario is related to the temperature data from the patient. If the thermometer reports a temperature indicative of a fever (38°C and above), the aim is to increase the polling rate of the thermometer to improve accuracy. When the patient recovers, the polling rate can be reduced to improve battery life.
Impact	If the polling rate is not dynamically controlled, energy may be wasted that could have used when it is critically needed. With lower energy consumption the battery life can be prolonged.
Degree of severity	Medium, because this uncertainty will not immediately negatively impact the system.
Sample illustration	A patient has a fever and uses a SA-BSN. The BSN chooses an adaptation which increases the polling rate of the thermometer. This increases battery consumption, but considering the current condition of the patient it is justifiable. When the fever eventually resides, the polling rate comes down with it.

Table 1: Uncertainty 1: Offline and Online Functionality

Uncertainty 2	
Field	**Description**
Name	Sensor Reliability
Classification	Monitoring phase
Context	This uncertainty arises because of the sensors' unreliability. A sensor could report incorrect data about the patient's status or no data at all.
Impact	if a sensor publishes incorrect blood pressure data from a patient with high blood pressure, returning a low blood reading will result in suspension of medication administration, this single act can cause the patient's system to shut down when the pressure gets too high without any medication being administered because of unreliable sensor reading.
Degree of severity	High degree of severity, as error sensor data may lead to wrong decision making that can adversely affect the patient's health.
Sample illustration	A patient is being monitored by a SA-BSN. The BSN finds irregular data during its monitor phase. The analyser chooses to turn off (set the polling rate to 0 Hz) the affected sensor as to avoid making decisions based on incorrect data.
Evaluation	Evaluate the sensor reading by comparing it's readings to values in the past as well as from other sensors that are closely related to the condition monitored to make inference.

Table 2: Uncertainty 2: Sensor Reliability

Uncertainty 3	
Field	**Description**
Name	Varying Patient Status
Classification	Monitoring phase
Context	The system might need to account for health conditions which cannot be categorised using the six sensors that are currently available.
Impact	An undetected health condition could have severe consequences, but this highly dependent on its nature. An additional sensor can give more information about patients that cannot be identified from the current sensor suite.
Degree of severity	High, because incorrect classification of health status will result in wrong diagnosis.
Sample illustration	A patient is being monitored by a SA-BSN. The analyser phase of the BSN finds nothing remarkable. If a patient feels affected by something that isn't detected currently, a new sensor can be added and used for analysis.
Evaluation	Adding a new sensor should lead to a noticeably different adaptation.

Table 3: Uncertainty 3: Varying Patient Status

4 Requirement Analysis

The traditional requirements of the adaptation strategy are as follows:

Monitoring

1. The sensor SHALL monitor the patient's data once an emergency is detected and every 5 minutes interval after data.
2. The sensor data SHALL be published to the analyser components at intervals.

Analyser

1. The analyser SHALL be allowed to analyse the published sensor data at any time.
2. The analyser SHALL published the sensor data to the planner immediately it received them.

Planning

1. The planner SHALL be allowed to plan the adaptation strategy at any time.
2. The SHALL enact the adaptation strategy at any time, once it receives updates from the analyser components.

Execute

1. The strategy enactor shall be allowed to enact adaptation strategy at any time.

The RELAX-ation requirements of the strategies is as follows: *Monitoring*

1. The sensors SHALL monitor the patient's health conditions AS EARLY AS POSSIBLE when an emergency is detected.
2. The monitored patient's data SHALL be forwarded AS MANY AS POSSIBLE to the analyser component.

Analyser

1. The analyser component SHALL analyse the published sensor data AS EARLY AS POSSIBLE, after they are published.
2. The analysed sensor data SHALL be published to the planner AS EARLY AS POSSIBLE, for effective planning.

Planning

1. The planner module SHALL make adaptation plan AS SOON POSSIBLE to mitigate against any unforeseen circumstance.
2. The planner SHALL enact the adaptation as EARLY AS POSSIBLE.

Execute

1. The strategy enactor SHALL enact the adaptation as EARLY AS POSSIBLE, EVENTUALLY implement the new adaptation.

5 Analysis of potential solution

5.1 Adaptation 1

5.1.1 Description

During normal operations, high data publishing rate is not desirable, if a sensor publishes data at higher frequency, more energy will be consumed. Here, we implement our sensors to classify scenarios as high, medium, or low and publish data. If a scenario is classified as high, we increase the data publishing rate for that specific sensor based on what is being sensed. Once the situation that warrants the increase in data publishing rate is under control, the sensor dynamically goes back to normal data publishing rate.

5.1.2 Potential Solution

Using high fever detection as a use case we temperature reading of the patient is desire to identify the next course of action based on the data available. If the thermometer data indicates high temperature reading indicative of fever, the polling rate of thermometer data is increased to confirm the earlier published data and improve recommendation accuracy.

5.1.3 Advantages

The solution will minimise the energy consumed during sensor data publishing by reducing the rate at which other sensors publish their data. It will also lead to improved recommendation accuracy as decisions will be made based on sufficient data.

5.1.4 Limitations

We will miss information from other sensors that can support or indicate other possible assessments to be made before deciding on the treatments and recommendations to administer.

5.1.5 Motivation

Increasing battery life by limiting the amount of energy consumed.

5.2 Adaptation 2

5.2.1 Description

We plan to implement a rule-based algorithm to decide whether to turn on or off certain sensors. Based on a sliding window we can detect significant spikes in sensor readings and activate new sensors to verify this behaviour and new sensors can classify health conditions that are not classified by the existing sensors.

5.2.2 Potential Solution

This adaptation extends the existing system with new sensors without significantly increasing energy consumption. We added new sensors to cater for health conditions that the existing system cannot classified. Also, this rule-based algorithm decides whether to turn on or off certain sensors. Based on a sliding window we can detect significant spikes in sensor readings and activate new sensors to verify this behaviour. This adaptation will improve the reliability of the system. If the sensors disagree on the readings, we can either turn them both off or turn the old sensor off based on readings from the new sensor.

5.2.3 Advantages

Rhe reliability of the system is guaranteed by comparing the sensor data published using new sensors and the system scales to handle other unidentified health conditions.

5.2.4 Limitations

Comparing this sensor reading leads to high expenses on energy.

5.2.5 Motivation

The solution is motivated by the need for reliability and scalability in the system.

5.3 Adaptation 3

5.3.1 Description

We implement a new sensor node, this adaptation will enable us to monitor patients' health conditions that the existing sensors implemented by the current BSN cannot monitor. This sensor is meant to replicate an EDA (electrodermal activity) sensor, which can measure the amount a person is sweating using skin conductance. We plan on turning on this sensor depending on the data from the other sensors to support findings from other sensor reading.

5.3.2 Potential Solution

we implement a new node that provides different information about the patient's current health. If the patient's health cannot be classified based on the existing sensors, we activate this new sensor to provide indications on what might be going on with the patient, we only activate this sensor if the other sensors cannot provide insight on the patient's health status, or when we need more information to support our decision.

5.3.3 Advantages

This sensor can give insight into health conditions that existing sensors cannot classified.

5.3.4 Motivation

The need for handling unidentified health conditions.

5.4 Adaptation 4

5.4.1 Description

In this strategy we improve the battery consumption rate by reducing the frequency in which sensors publish their data, as well as safely guiding patients by having sensors publish their data at a higher frequency in case of emergencies.

5.4.2 Potential Solution

Throttling the sensor publishing rate depending on the circumstance monitored.

5.4.3 Advantages

If all the sensors publish data at a higher frequency, we end up using energy that could have been preserved to collect data that is not needed, also resources like storage can run out because of the high rate of data published to them. If the update frequency of the sensors is not what we expect, we might be using more power than necessary or not get enough data about the patient's health and high data rates leads to computation complexity.

5.4.4 Motivation

Publishing data based on requirements again, improves battery life.

6 Design of the Proposed Adaptation Strategy

The adaptation phase starts by identifying uncertainty, checks the sensor data knowledge based to ascertain the degree of disparity from the expected behaviour, a response is sent back to the strategy manager who places a self-call to analyse the possible the impact of the uncertainty on the system, based on it discoveries the strategy enactor is invoked, to propose the a plan for adaptation, the enactor again confirms the current situation from the knowledge repository, is adaptation is still required the effector ids Invoked to implement the adaptation plan on the target system, while the probing component continually access the knowledge base to see if there are some adaptations that needs to be made and the current state of the system is persisted is no changes are required.

The adaptation cycle of the BSN system is described using the sequence diagram shown in Figure 2. This shows the behavioural interaction between the components and how they collaborate to implement adaptation.

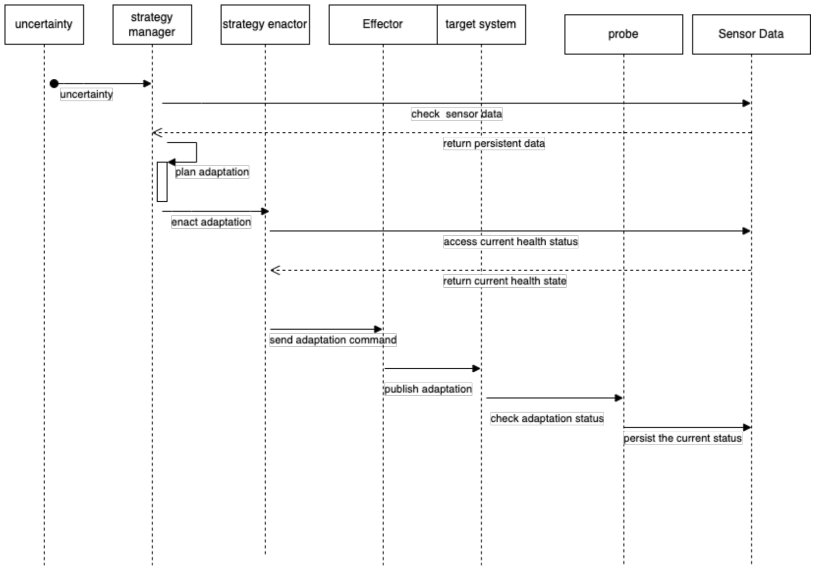


Figure 2: Sequence diagram of the Self-Adaptive Body Sensor Network, from Gil et al. [1]

7 Implementation

To implement the SAS, the waterfall method was used, where the assignment was worked on component wise, in sequence. The following subsections go over the design process in a similar sequence.

7.1 UPISAS and Docker

The UPISAS connection had to be established and a docker image had to be build. To build this image, we made a Dockerfile which constructs an image based on the 'egalberts/bsn:latest' docker image provided to us, and layered some instructions on top. These instructions i.a. perform updates, installs dependencies and clones our BSN fork. We did not find a clean solution to cloning a private repository from any machine during the build process, so an access key is present in this Dockerfile. It only has read access to that specific private repository.

To have a correct environment for ROS commands inside the container, the BSN fork had to be updated as well to run the 'source' commands from '.bashrc' in the 'run.sh' script. The UPISAS tests do not pass completely, because the json schemas from the BSN HTTP-endpoint fail verification. My hypothesis is that the schema is not identical to what UPISAS tests for because the schema builder we used adds a link to the json schema website by default.

Whenever a container is run, it executes the run.sh script in the container, which also starts the HTTP-endpoint. Once the container is built, the UPISAS fork cloned and the requirements installed, running 'python3 run.py' should be all that's necessary to test the system.

7.2 Monitoring Phase

This phase was left untouched, apart from calling the monitor method without verification as explained above.

7.3 Analysing Phase

During the analysing phase, the SAS only takes the last 5 values from each sensor into account — if available. This pseudo sliding-window is used to average the sensor readings out, to filter out a part of the noise. The built-in python dictionaries were used to store this filtered data, as they allow for indexing using strings.

7.3.1 Battery Drain

Design Phase An important step during the analysis is calculating *battery drain*. After we picked PID controllers for our managing system, we had to choose what these controllers would interact with. During our work on assignment 1, we were only ever able to confirm publishing to sensor nodes in ROS. Therefore we chose to let the PID controllers update the polling rate of each of the sensors, based on the current battery drain. The analysis decides whether continue to planning, based on the temperature, risk and battery level of each sensor.
Downside Late in the implementation process we discovered flaw in this design decision. Since the SA-BSN automatically recharges the simulated batteries in ROS, there needs to be extra logic in place to handle recharging for the PID controllers, since drain is not affected by charge level, only polling rate.

7.4 Planning Phase

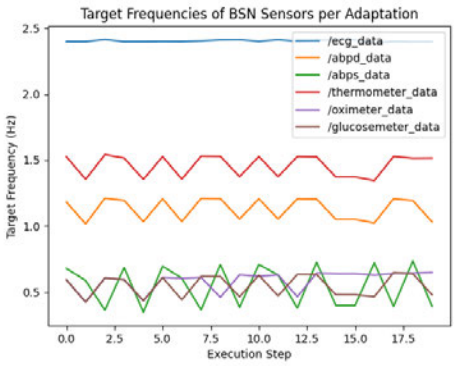
The planning phase triggers the next step of the PID controllers. The PID controllers take the battery drain level as their input, which is calculated as the battery level of the first and last element of the sliding window. The resulting target frequencies for each of the sensors is what is passed on to the execution phase, based on a few environmental conditions such as risk, battery level and body temperature.

7.5 Execution Phase

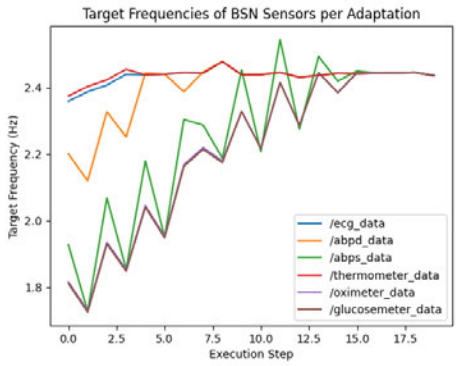
The execution phase is left untouched, apart from disabling schema verification again.

8 Evaluation

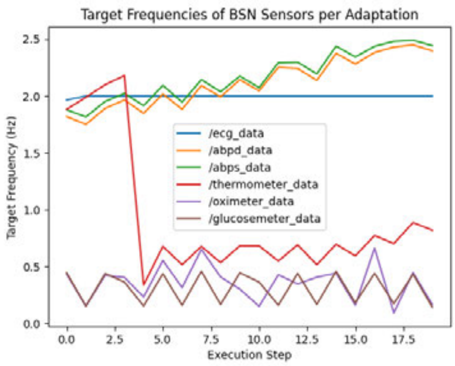
To evaluate our strategy, we used dynamic plots that we added to the system as seen in Figure 3. Figure 3c shows the system adapting to a low battery state of the thermometer. Similarly, Figure 3d shows the polling rate of sensors increase because of the available battery capacity.



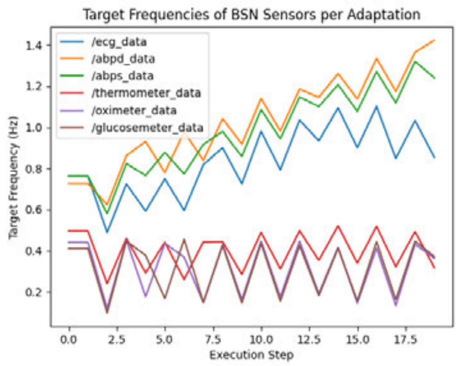
(a)



(b)



(c)



(d)

Figure 3: Dynamic Plotting

9 Reflections

The most challenging part of the task was getting to understand how the components of the BSN works, and creating an external node to interact with the system and having to learn the technicality surrounding ROS in a short period of time. At the early stage of the project we encounter challenges in getting the provided docker image to work on our machines. Another major challenge was getting the team members to contribute meaningfully.

10 Division of labour

Team collaboration was mostly during the lab session where we brainstorm and deliberate on issues, we encounter personally will working on the project trying to help each other figure out the way forward. We supported this we virtual team meetings and collaborative coding. We jointly deliberate on the proposed adaptation strategy to implement, suggesting possible implementation principles and design.

[redacted]

[redacted]

[redacted]

References

[1] Eric Bernd Gil, Ricardo Caldas, Arthur Rodrigues, Gabriel Levi Gomes da Silva, Genaina Nunes Rodrigues, and Patrizio Pelliccione. Body sensor network: A self-adaptive system exemplar in the healthcare domain. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, May 2021.

[2] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, number 3.2, page 5. Kobe, Japan, 2009.