A. Questions regarding the theory in the textbook

A.1 Study the figure 5.3 and its description as a horizontally layered agent architecture in chapter 5.2.1. Try to adapt the final state machine (FSM) for the citizen agents from DiC_chp4b/C.1 in a way that it still is a finite state machine but adheres to this horizontally layered agent architecture. Draw by hand on paper.

Horizontal

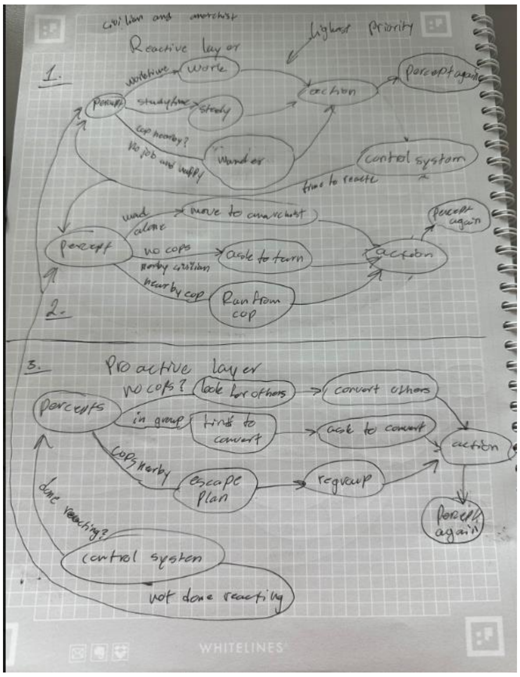
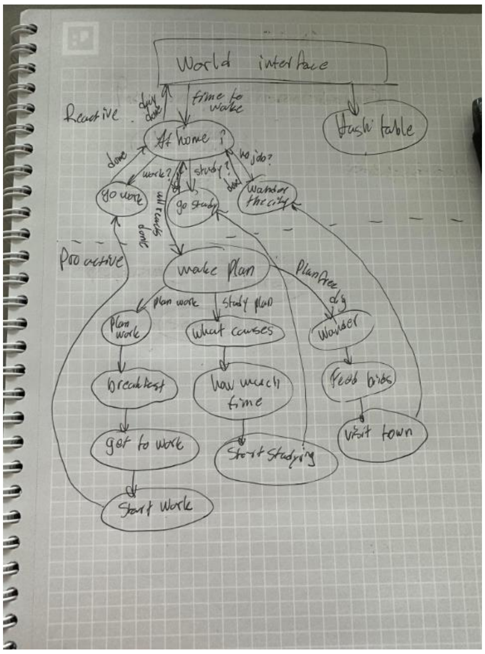


Figure 1 shows an example of a normal civilian who has work or studies or has no job and likes to wander, it reacts to what time it is and does an action accordingly. Figure 2 shows the same reactive FTM however for an anarchist who has different reactions to what happens around it, so it wants to group up with others, asks people to convert to anarchists and runs from nearby cops. Once the reactive layer is completed the control system instead of pointing us back to the reactive state machine it points us to the proactive state machine and we instead of reacting to things and instantly do something we make a plan for how to achieve our goals and then step by step do these actions, then we go back to the control system and check if we need to react to something again or if we can keep on planning and make up steps to complete our goals. Important to note is that reactive layer always takes priority since those reactions are time sensitive and these actions are critical processes needed to be complete before other things can be done, like a self-driving car that needs to avoid pedestrians can't instead plan for the most efficient rout to the destination it needs to prioritize stopping the vehicle and saving a life. A more relevant example is that a anarchist in our project cant make a plan of how to get to another activist if there is a cop nearby, it needs to first save its skin and run from the cop before it can be proactive.

A.2 Study the figure 5.4 and its description as a vertically layered two-pass agent architecture in chapter 5.2.2. Try to adapt the final state machine for the citizen agents from DiC_chp4b/C.1 in a way that it still is a finite state machine but adheres to this vertically layered agent architecture. Draw by hand on paper.

Vertical



The proactive layer of this civil agent uses the reactive layer however there is no way for the reactive layer to affect the proactive layer. The plan gets made for how to use the hours of the day and how to achieve our goals and once the plan aligns with the reactive layer the plan uses the react layer to execute the actions required to do the actions. The world interface sets th beliefs based on perceptions from the world into the hashtabel and these beliefs gives the perception to the rest of the layers.

A.3 According to the theory, the information - and control flow differs between these types of architectures. How is the information - and control flow in your FSM from A.1? How are they in your FSM from A.2? What is the difference regarding the information- and control flow between the two FSM?

The two FSM differ in control flow in the way that in a vertical architecture everything flows through one layer to another and the different layers are aware of the layer underneath. Once a plan is made we can go back down a layer into the reactive layer and execute the plan using this layer. There is no control circuit for this type of machine and everything travels through the hierarchy from the top priority to the lowest propriety (react to social). The horizontal architecture uses a control system to determine what state machine to enter and what state to start in. It has a priority order that it uses to determine what has to be done first (always reactive layer) and designates when to do what, once all the reactions have been complete the control system then tells the system to instead enter the proactive FTM and create and execute the created plan.

B. Questions regarding research articles/the other research material

Referring to the articles about implementing BDI in Netlogo (available on Canvas):

B.1 How is BDI being implemented in NetLogo? How are "beliefs", "desires" and "intentions" implemented within computer science? (Meaning that you describe the implementation with terms from computer science).

BDI in netlogo gets implemented by a stack for the intentions and a hash table for the beliefs, in the programming world there is no real way to implement desires in a separate way, usually these desires just get treated as weaker intentions

B.2 Regarding the hybrid agent architectures, why is the way "intentions" are implemented a good way?

The hybrid architecture uses different ways to implement intentions giving the agent the best solution to the problem of how to use intentions, this hybrid architecture gives the agent the adaptability needed to adapt to changing environments and processing of dynamic situations. Basically by using both reactive architectures and using BDI we mitigate the disadvantages of these methods individually and strengthens the strengths of these methods.

C. Questions regarding the implementation in the project

Referring to the citizen agents:

C.1 What are possible "desires" for the citizen-agents and how could they be implemented with the BDI for NetLogo? How could they be implemented in NetLogo?

Possible desires for the citizen agent could be getting to work or mindlessly wandering about the town for their own pleasure. This could be implemented by using a variable that slowly decreases and once it goes under a threshold this desire becomes a intention and the agent will try doing this action to bump up its value to simulate satisfying this belief, if it does not get to refill this vale and it gets way to low this could make the citizen angry and increase the odds of it turning again the government and rebelling.

C.2 How can the finite state machine from A.2 or A.3 above be implemented with help of the BDI-extension of NetLogo?

The BDI extension for netlogo allows you to set beliefs and intentions for the agents and this means that all actions and criterias can be replaced with checking intentions and beliefs of agents, if an agent has a particular belief and something happens to modify that belief due to it getting mad that means we can modify its beliefs and perhaps if needed change the current intentions to new intentions that agree with the new beliefs. This can give much needed individuality to the agents and the agents can communicate their own beliefs and this will in turn affect nearby agents. The intentions will become actions and the beliefs will become the new parameters for when to do what.