

James Villemarette  
CISC 489 Homework 1  
Due: March 15, 2022  
Extension: March 20, 2022

## Problem 1

- Nurse Agent
  - Logical Deductive (e.g., Agent0/Concurrent MetateM)
    - Focus: Agent0
    - Strengths
      - Agent0 is good for the “*mentalist* notion” (Wooldridge 55) of making an agent. It is very human like, where its overarching goals are belief, desire, and intent oriented. The belief and desire that the patient should be taken care of, and the intent to keep the patient alive can be a good starting point to write formal rules for the patient.
      - The problem of keeping a patient alive can be easily represent the real-world environment into a symbolic one. For the health sensor data, especially, we can measure the patient’s cholesterol, and ensure it’s in the right range, and represent that as the states NormalCholesterol, HighCholesterol, or LowCholesterol. This same principle can be applied to other vitals.
    - Weaknesses
      - The biggest problem with Agent0 is the “language is essentially a *prototype* not intended for building anything like large-scale production systems” (56). And if there were multiple nurse agents taking care of multiple patients, it would be ineffective. This is because “the inability of agents to plan, and communicate requests for action via high-level goals” (Prof. Decker, Lecture3, page 15).
  - Practical BDI (e.g., PRS)
    - Focus: PRS
    - Strengths
      - Since intentions are much stronger than mere desires, the agent is strongly committed to the patient’s success.
      - Planning is automatic programing, as the design of a course of action that will achieve the desired task of keeping the patient alive.
    - Weaknesses
      - It can be very tricky to setup a PRS agent for nursing given that you must either chose a bold agent, a cautious agent, or dynamism. And each have their own problems.

- A **bold agent** may work well when a patient is stable, but if they have a heart attack, then it becomes problematic.
  - A **cautious agent** may work for an unstable patient, but if the patient stabilizes, then excessive testing may actually worsen a patient's condition.
  - A **dynamic agent** that reconsiders its intentions based on  $\gamma$  rate of environment change is tricky. How do gauge the stability of a patient (a large part of the agent's environment) accurately? Is it blood pressure and heart rate? When it's low, the patient is stable? Then what if the patient is low on sodium or malnutrition? Then how do you weight all these factors to calculate  $\gamma$ ? This math becomes endless.
- Reactive (e.g., Steels' Mars rovers, Brooks' and Maes' work)
  - Strengths
    - It just keeps sensing a patient's vitals, and if the patient needs it, it just provides it
    - Very practical to implement
    - Responds as needed
  - Weaknesses
    - Since the illness is chronic, a reactive agent may not react when there are no changes (no improvement) to the patient
    - If the patient does not improve or change state, then the agent cannot necessarily "react"
- Story-telling NPC Agent
  - Logical Deductive (e.g., Agent0/Concurrent MetateM)
    - Focus: Agent0
    - Strengths
      - The same *mentalist* notion described above still provides a powerful framework for making a good story-telling character.
      - The high level of abstraction makes it easy to understand the story an agent is telling just from its messages in text.
      - Its flexible plan-based approach with different paths can allow for unique story experiences.
    - Weaknesses
      - Since the agent is already in a digital environment (a video game), the transduction (vision) problem is largely or completely reduced.
      - A game world that is a static environment (that deductive reasoning agents typically assume) can be boring from a story telling point of view.

- Practical BDI (e.g., PRS)
  - Focus: PRS
  - Strengths
    - A PRS agent's intention of telling a good story will lead to good actions of specific dialogue. Its intention will persist, and constrain the search space of deliberation
    - A **dynamic agent** could potentially be perfect, where you just update the agent when a story event happens
  - Weaknesses
    - The same problems of a bold, cautious, or dynamic agent persist
    - A **bold agent** may still be telling the player about how bad BossA is when the player has already defeated BossA. This would be because the bold agent has not reconsidered recently.
    - A **cautious agent** may work, but it could slow down the game world with all its constant reconsideration.
- Reactive (e.g., Steels' Mars rovers, Brooks' and Maes' work)
  - Strengths
    - This would be a perfect agent, as the agent would just react to the user's requests and world events each time, in real time
  - Weaknesses
    - This story telling agent could not be a good guide, as it would only react to inputs by the user. A user could request this agent to lead them to a destination, but it could get messy with there not being a distinct state. We would have to implement multiple reactive systems to get states.

## Problem 2

- Utility
  - $Ag1 = .7 * 10 + .2 * 6 + .1 * 5 = 7 + 1.2 + .5 = 8.7$
  - $Ag2 = .6 * 12 + .3 * 3 + .1 * 4 = 7.2 + .9 + .4 = 8.5$
- Conclusion: Agent 2 is preferable for environment 1, as it has a higher expected utility than Agent 1. However, Agent 1 is preferable for u, as it has a higher total utility.

## Problem 3

$[ \text{ask}(\text{dwarf1}, b) \wedge [b > \text{myLastbid}(m)] \wedge [\text{money}(N) \wedge [N \geq b+1]] ] \rightarrow \text{ask}(\text{dwarf2}, b+1)$

If dwarf1 bids b, & b is higher than m, and dwarf2 has money &- , then bid that amount.

^  
enough of it

^(assuming that this function returns an int value, since it seems like it)

#### Problem 4

- What is important
  - Beliefs
    - A user can exercise; It is possible
    - There is such a thing as too much exercise
    - There is such a thing as too little exercise
  - Desires
    - For the user to get x amount of exercise every y days
    - Increase the amount of physical activity
  - Intentions
    - Make sure the user does not over exercise
    - Remind the user to exercise

The biggest problem to tackle first would probably be the reconsideration problem. This is the hardest challenge with BDI agents, is when to reconsider. Too much or too little reconsideration can stall progress on an intention. We would want to reconsider plans maybe on a day-to-day, week-to-week, or month-to-month basis. Deliberation and planning could be problematic because we do not want to interrogate the user too often on their specific desires. A user would maybe set a large desire to just generally improve running speed, and we would have to somehow extrapolate from that one or few desires. The user (human) can only be expected to communicate desires so specifically. There would be substantial work in translating these desires into a format that is processable by deliberation and planning.

#### Problem 5

An [rdfs:range](#) is [used to helpfully specify](#) the intersection of the individual ranges. It is helpful to have to use ranges as an intersection, as a union can be more inclusive than we would need. The intersection is precise.

If someone adds an object to be a subclassOf of an object that we unioned in, and we do not want that sub class object included, then we may switch to `rdfs:range` to limit.

#### Problem 6

- First, we should identify what classes there are
  - Person,
  - Actor,
  - Director,
  - Movie
- Then the triplets that we can infer are
  - < Director, rdfs:subClassOf, rdfs:Class >
  - < Actor, rdfs:subClassOf, rdfs:Class >
  - < freeman, rdf:type, Actor >
  - < mckellan, rdf:type, Actor >
  - < Jackson, rdf:type, Person >