Assignment P4

Clinton McCardell
cmccardell3@gatech.edu

1 QUESTION 1: GOMS MODELING

1.1 Initial Situation

The initial situation is needing to receive information from a professor about the grade an assignment was given.

1.2 Selection Rules

- If I am located on Georgia Tech's campus and the inquiry is urgent, choose the 'in person meeting' method.
- If the professors offers office hours and I am not located on Georgia Tech's campus, choose the 'office hours' method.
- If the class uses a forum, choose the 'class forum' method.
- If the class does not offer a forum, choose the 'email' method.
- If the professor's email in unavailable, choose the 'Canvas message' method.

1.3 Methods and Operators

The In Person Meeting method

- Open web browser and navigate to Georgia Tech's website. (0-1 minute)
- Find professor in the directory to find their office and schedule. (1-3 minutes)
- Await the time that the professor will be in their office. (0-5 days)
- Travel to office. (5-15 minutes)
- Speak with professor about selected assignment and receive feedback. (5-15 minutes)

The Office Hours method

- Open web browser and navigate to class website. (0-1 minute)
- Navigate to syllabus and locate office hours information. (1-3 minutes)
- Await beginning of office hours. (0-5 days)
- Attend office hours remotely and ask professor for feedback on the assignment. (5-45 minutes)
- Receive feedback about assignment from professor. (1-5 minutes)

The Class Forum method

- Open web browser and navigate to class forum website. (0-1 minute)
- Post private message to professor detailing the solicited explanation. (2-5 minutes)
- Await response from professor. (0-24 hours)
- Receive and read response from professor. (2-5 minutes)
- Ask potential follow-up questions. (1-3 minutes)
- Await potential follow-up answers. (0-24 hours)
- Receive and read potential follow-up answers. (1-3 minutes)

The Email method

- Open web browser and navigate to class website. (0-1 minute)
- Locate professor's email address. (1-3 minutes)
- Open email client. (0-1 minute)
- Compose and send an email to professor detailing the solicited explanation. (2-5 minutes)
- Await response from professor. (0-48 hours)
- Receive and read response from professor. (2-5 minutes)
- Ask potential follow-up questions. (1-3 minutes)
- Await potential follow-up answers. (0-48 hours)
- Receive and read potential follow-up answers. (1-3 minutes)

The Canvas Message method

- Open web browser and navigate to Canvas. (0-1 minute)
- Locate messaging component and begin private message to professor. (2-4 minutes)
- Send private message to professor detailing the solicited explanation. (2-5 minute)
- Await response from professor. (24-144 hours)
- Receive and read response from professor. (2-5 minutes)
- Ask potential follow-up questions. (1-3 minutes)
- Await potential follow-up answers. (24-144 hours)
- Receive and read potential follow-up answers. (1-3 minutes)

1.4 Ultimate Goal

The ultimate goal is to receive an explanation about why the grade was given to the assignment in question.

2 QUESTION 2: COMPLETING AN ASSIGNMENT

Top Level Task: Complete an assignment

2.1 Sub-task 1: Submit Assignment

Sub-sub-task 1.1: Navigate to assignments

- 1.1.1 Open web browser
- 1.1.2 Enter Canvas url (https://gatech.instructure.com/)
- 1.1.3 Enter Georgia Tech credentials
- 1.1.3 Complete 2FA if prompted
- 1.1.4 Click 'Courses' tab
- 1.1.5 Select relevant course from the list
- 1.1.6 Click 'Assignments' link

Sub-sub-task 1.2: Navigate to specific assignment

- 1.2.1 Scroll to locate the assignment
- 1.2.2 Click on that assignment
- 1.2.3 Click 'Start Assignment' link

Sub-sub-task 1.3 Choose file to upload

- 1.3.1 Select the location of the source file (Disk, Box, etc.)
- 1.3.2 Open the file chooser ('Choose File' button)
- 1.3.3 Scroll to find the selected file
- 1.3.4 Choose the selected file

Sub-sub-task 1.4 Upload file

- 1.4.1 Add any comments, if desired
- 1.4.2 Click to agree to EULA
- 1.4.3 Click 'Submit Assignment'

2.2 Sub-task 2: Receive a Grade

Sub-sub-task 2.1: Wait for assignment to be graded

- 2.1.1 Open web browser
- 2.1.2 Enter Outlook url (https://outlook.office.com/mail/inbox)
- 2.1.3 Enter Georgia Tech credentials
- 2.1.3 Complete 2FA if prompted
- 2.1.4 Check for grade notification email
- 2.1.5 Repeat periodically until email comes

Sub-sub-task 2.2: Navigate to grades

- 2.2.1 Enter Canvas url (https://gatech.instructure.com/)
- 2.2.2 Click 'Courses' tab
- 2.2.3 Select relevant course from the list
- 2.2.4 Click 'Grades' link

Sub-sub-task 2.3: View grade

- 2.3.1 Scroll to relevant grade
- 2.3.2 View score
- 2.3.3 View statistics if applicable

2.3 Sub-task 3: Receive Feedback

Sub-sub-task 3.1: Wait for feedback to be given

- 3.1.1 Open web browser
- 3.1.2 Enter Outlook url (https://outlook.office.com/mail/inbox)
- 3.1.3 Enter Georgia Tech credentials
- 3.1.3 Complete 2FA if prompted
- 3.1.4 Check for peer feedback notification emails
- 3.1.5 Repeat periodically until emails come

Sub-sub-task 3.2: Navigate to all feedback

- 3.2.1 Enter Peer Feedback url (https://peerfeedback.gatech.edu/app/dashboard)
- 3.2.2 Click 'View' on relevant course

Sub-sub-task 3.3: Navigate to feedback for assignment

- 3.3.1 Scroll to assignment
- 3.3.2 Click 'View' in assignment
- 3.3.3 Click 'View' under 'Feedback from Peers'

Sub-sub-task 3.4: View feedback

- 3.4.1 Scroll to the bottom of the assignment
- 3.4.2 View individual feedback from peers

3 QUESTION 3: DISTRIBUTED AND SOCIAL COGNITION AND NAVIGATION

Before GPS, the system for navigation comprised a married couple, a map, road signs, and other artifacts. Below, we analyze the role of the driver, passenger, map, and road signs and identify their cognitive activities. We also compare this system to the modern navigation system of a lone driver with a GPS.

3.1 Pre-GPS Analysis

The driver, in our opinion, has the most crucial cognitive role in this distributed system. Their perception activities are to take in input about the car (speed, engine temperature, etc.), the other vehicles around them (lane position, speed), street signs they may pass, and instructions from their spouse. Their memory activities are to maintain in working memory the last instruction given to them by the passenger and information about their surroundings, and to maintain in short-term memory the current speed limit and road conditions. As far as reasoning, the driver must use gathered information to decide when executing maneuvers such as turning is safe, judge proper lane position and position relative to other cars, and cross-reference the received instructions with the surroundings and long-term memory. Their acting activities are to actually maintain the vehicle in a safe lane and car position, execute the received instructions, and assist the passenger with their perceived information.

The passenger is in charge of making navigation judgements and communicating instructions to the driver. In terms of perception, the passenger takes in information about the car's current position, information from street signs or other landmarks, and information provided by the map. The passenger must keep in working memory the information they perceive from their perceptive activities, keep in short-term memory their relative position on the map, and keep in working memory the next instruction to give after it has been reasoned. In terms of reasoning, the passenger must compare the perceived information from the outside with the perceived information from the map and resolve the location of the car. They must also reason about the next proper direction to give so that the driver can execute it. They must determine whether they have arrived at the destination and resolve any feedback from the driver about the validity of the given instructions. Finally, the acting activities are to give the actual directions and communicate to the driver when they have arrived.

The map's role in this system is simpler, as it only performs memory activities. The map itself cannot perceive, reason, or act–it can, however, remember information. The map serves as long-term memory of many different routes to navigate all over a region. Specific sections of the map or specific routes serve as short-term memory that can be "loaded in" when a particular navigation task takes place.

Like the map, the road signs do not act, perceive, or reason. They are long-term memory markers about which other agents can perceive and reason. The signs act as long-term memory by preserving crucial information about the immediate area. This includes, for example, information about the speed limit, cross streets, highway exits, places of interest, and destination checkpoints. By storing all of this information in long-term memory, the signs offload cognition from the passenger who can use the street signs to better navigate the map.

3.2 Comparison to a Lone Driver with GPS

In the case of the lone driver with a GPS, the driver has a less crucial role in the system–they really only need to manage the car and listen for instructions from the GPS. The GPS, on the other hand, becomes very crucial because it takes on the roles of map, passenger, and–even to some extent–the street signs. All of the information that these other artifacts would give are now present in the GPS.

When the GPS is introduced into the system, various social components disappear. For example, because the GPS is constantly recalculating and determining the optimal route, the driver no longer has to judge the quality of the directions they receive, as they do in the case of the human passenger. The driver does not have to worry about the GPS falling asleep or being inattentive and missing a turn or instruction. Additionally, because the GPS is nonliving, the driver does not have to worry about being judged if they misinterpret a direction and act errantly. Finally, because the GPS has no emotions, changes in emotion from either the driver or navigator will not affect the quality of the instructions given.

Social cognition reveals that in the case of the human navigator, the quality of the system's functionality is dependent on conditions that are set before the navigation begins, and that interactions that occur during navigation can impact the relationship between the driver and navigator long after the navigation task is complete. For example, if the navigator has not gotten enough sleep, or the driver is in a testy mood, interaction between the driver and navigator will suffer–the navigator might miss a turn, or the driver might not have confidence in the navigator's instructions. Additionally, the social relationship among the parts can affect the system as a whole. If, for example, the driver misses many instructions and blames the navigator, the navigator might lose confidence in the driver and try to help less. On the other hand, if a really good social relationship exists, the navigator might be able to personalize the instructions in a way that makes the system more effective than the GPS system, such as reminding the driver of instructions less or more than the GPS would.

4 QUESTION 4: DISTRIBUTED COGNITION AND SOFTWARE DEVELOPMENT

I am a professional software developer, and as such spend a considerable amount to time writing code. Below, I describe my task and the system associated with it, and discuss which cognitive tasks are performed by each part of the system.

4.1 Interfaces and System

I develop using Microsoft Visual Studio as my IDE, which contains a text editor and linter. It is the main interface with which I interact. Additional interfaces include input devices such as my mouse and keyboard, as well as display devices such as my three monitors. The parts of the cognitive system we form include myself as the main agent, Visual Studio itself, the compiler, the mouse, the keyboard, and the monitors.

4.2 Cognitive Tasks

As the main agent, I have many cognitive tasks. My perception tasks include reading the code that has already been written by interpreting what is displayed on the screens. I reason about what the written code means and what it does, as well as the proposed additions and changes I will make to the codebase. In working and short-term memory, I store the results of my reasoning tasks. Finally, I act by typing my changes into the keyboard and navigating between files with the mouse.

The mouse really only has two cognitive tasks. It perceives the input I give to it as part of a perception task and performs acting tasks based on that input. By acting, the mouse transfers the actions I perform to other interface components including Visual Studio, the compiler, and source files.

In addition to performing the same tasks as the mouse, the keyboard has the memory task of remembering options I have enabled/disabled such as caps lock and number lock.

The monitor array has two tasks–one acting and the other memory. First, the obvious task of acting is to display to me the contents of the desktop and associated programs. Second, because I use an array of monitors, they perform the memory task of remembering which components of the interface to display on which screen.

Visual Studio, performs perception, memory, reasoning, and acting tasks. First, the text editor perceives the commands received from the keyboard and mouse. It also perceives the text that has been written for use in static code analysis (linting). Visual Studio stores in short-term memory a list of the source files that are currently opened and in which order they are arranged. In long-term memory, the linter holds syntactic and semantic rules from which to judge the quality of the written code before compilation. The linter component reasons by comparing the perceived code to the list of rules it has in long-term memory to see if errors have been written. If errors are present, it acts by reporting the errors to the user, usually in the form of red squiggles and a (sometimes) helpful message describing the error. Visual Studio also acts by editing the text files per the commands it receives from the keyboard and mouse, as well as by running build tasks and subtasks through the use of the compiler.

The compiler is the last piece of this cognitive system. It also performs perception, memory, reasoning, and acting tasks in a form similar to the linter. It perceives input from Visual Studio telling it which files to compile and link. It stores in long-term memory the algorithms for file linking and the rules that need to be followed to compile successfully. Like the linter, the compiler reasons about the rules it has stored in memory to see if the program is formed properly. However, the compiler reasons at a much higher level because C++ is a compiled language. Finally, it acts by building executables and reporting errors back to the user.