Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

NOTE: Dr. Chang suggested that the best way to start this process was to have a mixed state IC card at the top, which represents the overall task of organization of the picnic, which is then decomposed into numerous sub-tasks below. As such, I will follow the paradigm that he suggested here. Thus, the first IC card will be the mixed state card, which represents the overall action of organizing the picnic. As there are many sub-actions in its children, it is a complex action, and is therefore a mixed state.

Since numerous people will be coming to the picnic, we assume that certain people invited to the picnic (invitees) will help organize it. Thus, invitees will be delegated certain tasks to do. One person will be responsible for one task, another person is responsible for another, then another person is responsible for telling everybody once the date / time of picnic is decided, etc. So, the basic hierarchy is shown below.

IC Card IC Name: Organize Picnic

Description: Organize a Picnic
Interaction Pattern:

Mixed

Time Critical Condition: Must take place before the scheduled picnic date & time
Name of Other IC: Invitees
Message to Other IC: Invitees please begin organizing the picnic
Other IC's Task: Numerous sub-tasks
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

We begin by determining which of our friends will actually be coming to the picnic (we call the people that are actually coming Invitees). Once we know who is coming, we will be able to coordinate a location for the picnic (which is close enough to everybody).

(see next page)

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

IC Card

IC Name: Determine Invitees

Description: Determine all the friends which will actually be coming to the picnic (invitees)
Interaction Pattern:

By Myself with Interaction

Time Critical Condition: None (picnic date & time is currently unknown)
Name of Other IC: Friends
Message to Other IC: Do you want to go to a picnic that I am holding?
Other IC's Task: None (besides responding yes or no)
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

We now know who is actually coming to the picnic (we have the list of invitees). The assumption here is that we know where all the invitees live. Because of that, this IC will be able to determine the picnic location autonomously simply knowing the locations of all the invitees (we assume all the invitees are a reasonable distance apart).

IC Card

IC Name: Select Picnic Venue

Description: Choose a picnic venue reasonably close to all the invitees locations
Interaction Pattern:

By Myself no Interaction

Time Critical Condition: None (picnic date & time is currently unknown)
Name of Other IC: None
Message to Other IC: None
Other IC's Task: None
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

Once, we have decided on a venue, we must determine the availability of a pavilion in this facility to reserve (i.e. in case it rains we will have a pavilion to go to)

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

IC Card IC Name: Determine Venue Availability

Description: Find out numerous venue available slots and times which the venue can be reserved
Interaction Pattern:

By Myself with Interaction

Time Critical Condition: None (picnic date & time unknown)
Name of Other IC: Venue Manager
Message to Other IC: Please provide list of available times and dates which this venue can be reserved
Other IC's Task: None (besides providing information)
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

We now know a list of available times for which the venue can be reserved. We must now contact our invitees (which we know are coming) and determine which date & time works best for everybody.

IC Card IC Name: Set Picnic Time (and Date)

Description: Determine the picnic time (and date) which fits into everyones schedule
Interaction Pattern:

By Myself with Interaction

Time Critical Condition: The picnic time (and date) which fits into everyones schedule must be the same for everybody
Name of Other IC: Invitees
Message to Other IC: Given all these possible times and dates, please tell me which times and dates you are available (may be more than one)?
Other IC's Task: None (respond with list of times and dates which work)
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

We now know which date and time works for everybody.

Contact the park manager and reserve the pavilion for that time.

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

Here, we contact the venue manager and request him to reserve the venue in the venue's computer system. Note, we cannot do this ourselves, so the park manager must do this action himself in the computer system.

IC Card IC Name: Reserve Venue

Description: Contact Venue Manager and Make Reservations For Venue
Interaction Pattern:

By Others with Interaction

Time Critical Condition: Before agreed upon venue date & time
Name of Other IC: Venue Manager
Message to Other IC: Please reserve XXXX for date and time XXXXX for our picnic.
Other IC's Task: Reserve the venue in the system and mark it unavailable for others
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

Venue Manager now receives request to reserve venue. He now reserves it.

IC Card IC Name: Venue_manager_reserve_venue

Description: Reserve the venue in the system and mark it unavailable for others
Interaction Pattern:

By Myself no Interaction

Time Critical Condition: Before venue date & time and before another reservation for that venue location comes in
Name of Other IC: None
Message to Other IC: None
Other IC's Task: None
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

The venue date and time is now reserved in the venue manager's system. We now have a venue (which we assume is a covered pavilion or similar covered structure in case of bad weather).

We must now tell everyone about which date and time has been decided for our picnic and the venue.

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

IC Card

IC Name: Update Invitees

Description: Tell invitees which date, time, and venue has been decided on (which we know fits in everybodys schedule)
Interaction Pattern:

By Myself with Interaction

Time Critical Condition: Before picnic date and time
Name of Other IC: Invitees (all invitees)
Message to Other IC: The picnic date, time, and venue is XXXXXXX
Other IC's Task: None
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

All invitees now know the date, time, and venue of the picnic.

Dr. Zhang suggested we also make IC cards covering the organization of games and who will bring the foods. We will now delegate certain tasks to certain Invitees. We will say Invitee #1, Invitee#2, etc. to refer to the people coming and the tasks they need to do.

We now will delegate the bringing of food to certain invitees.

IC Card

IC Name: Bring Utensils, Cups, Plates, Co

Description: Have an invitee bring Utensils, Cups, Plates, Cooler, and Ice
Interaction Pattern:

By Others with Interaction

Time Critical Condition: Before Picnic Date and Time
Name of Other IC: Invitee #1
Message to Other IC: Please bring utensils, cups, plates, a cooler, and ice to the picnic enough for X many people.
Other IC's Task: Bring the supplies to the picnic with them.
Card 1 of 3 (If necessary please use several IC cards to describe an IC)

Invitee #1 is bringing the utensils, cups, plates, cooler, and ice.

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

IC Card

IC Name: Bring Snacks and Drinks

Description: Have an invitee bring the snacks and drinks to the picnic
Interaction Pattern:

By Others with Interaction

Time Critical Condition: Before picnic date and time
Name of Other IC: Invitee # 2
Message to Other IC: Please bring some snacks and drinks to the picnic, enough for everybody
Other IC's Task: Bring the snacks and drinks and buy some if necessary
Card 2 of 3 (If necessary please use several IC cards to describe an IC)

Invitee # 2 has been delegated the task of bringing all the snacks and drinks

IC Card

IC Name: Bring Main Course

Description: All invitees (except the organizer) are asked to bring a different main course to the picnic
Interaction Pattern:

By Others with Interaction

Time Critical Condition: Before picnic date and time
Name of Other IC: Invitees
Message to Other IC: Please bring food XXXX to the picnic as it is a potluck picnic
Other IC's Task: Bring the delegated food item
Card 3 of 3 (If necessary please use several IC cards to describe an IC)

Everybody except the organizer is asked to bring a main course item (potluck) to the picnic.

We will also need the radio for the picnic, so invitee #3 is asked to bring it.

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

IC Card IC Name: Bring Radio

Description: Have an invitee bring the radio
Interaction Pattern:

By Others with Interaction

Time Critical Condition: Before picnic date and time
Name of Other IC: Invitee # 3
Message to Other IC: Please bring a radio to the picnic with you when you come.
Other IC's Task: Bring the radio to the picnic.
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

Since the organizer isn't bringing any food, he will also bring some games to the picnic.

IC Card IC Name: Bring Games

Description: Bring some games and activities to the picnic so everyone has something to do
Interaction Pattern:

By Myself no Interaction

Time Critical Condition: Within 2 hours before scheduled picnic start date and time
Name of Other IC: None
Message to Other IC: None
Other IC's Task: None
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

Dr. Chang suggested we also need to drive certain friends to the picnic if they don't have a car.

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

IC Card

IC Name: Determine who Needs a Ride

Description: Poll invitees to determine who needs a ride
Interaction Pattern:

By Myself with Interaction

Time Critical Condition: 5 hours before picnic start time
Name of Other IC: Invitees
Message to Other IC: Do you have a ride or do you need picked up by the organizer?
Other IC's Task: None
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

We know who needs a ride. The organizer will pick them up and take them to the picnic

IC Card

IC Name: Drive Invitees

Description: Pick up the invitees that need a ride and take them
Interaction Pattern:

By Myself no Interaction

Time Critical Condition: 1-2 hours before picnic start time
Name of Other IC: None
Message to Other IC: None
Other IC's Task: None
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

Finally, everyone relaxes at the picnic after all the work of planning it.

Christopher Thomas
Software Engineering – CS 2310
CLT29@PITT.EDU

IC Card

IC Name: Relax

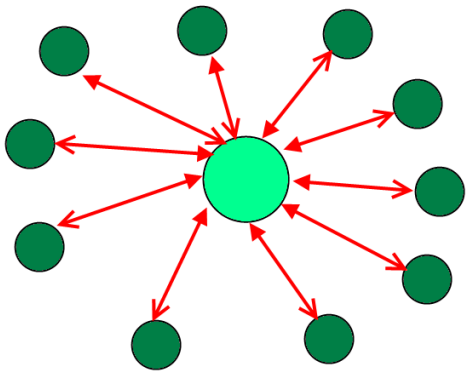
Description: Relax at picnic
Interaction Pattern:

Quiet State

Time Critical Condition: At picnic time
Name of Other IC: None
Message to Other IC: None
Other IC's Task: None
Card 1 of 1 (If necessary please use several IC cards to describe an IC)

PART B.

In the past, I have taken the distributed operating systems course in this department. Thinking about how the picnic was organized makes me think about a particularly important pattern: centralized command and control. In the case of the picnic, the organizer was responsible for polling invitees, deciding who brought what, telling people the decision, etc. So, this is the structure of a centralized architecture, whereby a central coordinator decides on what will happen. This is a well-studied development pattern in operating systems and software architecture (centralized vs decentralized systems). I have illustrated this below:



Thus, we see as the number of attendees grows, the number of messages also grows for the coordinator. For a small number of friends, this strategy works, but for a larger system, we need to divide the tasks up into sub-jobs, and appoint sub-coordinators, i.e. one person is in charge of arranging the date and time, one person is in charge of organizing food, etc. So in that case, we would have multiple mixed cards, and multiple sub-coordinators. For a huge system, it

may need to be done in a decentralized manner. Thus, one of the core patterns of this exercise is organizational architecture.

Another pattern I see here is the pattern of coming to a decision which suites everybody's needs and schedules. In computer science, these kinds of problems are called optimization problems, i.e. trying to determine the decision which fits everybody's needs. In this simple example, what would happen if no date and time fit everybody's schedule? In that case, the coordinator would either have to try to get more dates and times or simply decide some people cannot come to the picnic. In this pattern, a list of "constraints" for each invitee can be devised, the goal of the problem is to find a time which minimizes the number of conflicts. For a few friends, the problem is trivial, but when more complicated conditions occur (somebody needs to leave early, arrive late, etc.) and these conditions want to be considered, robust techniques exist, such as integer linear programming to solve these problems. This pattern of optimizing over a number of constraints is common in computer science and in many other domains.

One final pattern which I see is the concept of tasks and subtasks which must be completed before we move on to other tasks. A task can be considered completed when all it's subtasks are finished. In this way, we can view the problem in a tree structure, where tasks such as the following are represented in a tree structure **(note this illustration is NOT exhaustive, just an illustration)**. We can see that the "food" subtask is only considered complete when BOTH invitee #1's task and invitee #2's tasks are done. Note each of those tasks may also have subtasks which are not shown. If failure occurs in any one of those subtasks, our task will fail. However, using a tree pattern such as this can allow a system to quickly respond to task failures and model the overall process the system is taking in a way a system designer can quickly use to identify patterns and weak points. Thus, tree-like dependencies are an important pattern. Note, these kinds of graphs can also be used to detect CYCLES where certain objects or people are waiting for each other and in a state of deadlock. Thus, this pattern crops up in many areas of computer science and life.