RAFS - Robot Aided Feng Shui

風
水

Problem Definition Document (PDD)

Peter Motykowski-|-Bradley White-|-J.D. Pohlman-|-Matt Allen

1.0 Project Summary

The RAFS project will consist of a hardware/software solution to a problem set of physical objects. In our case, these objects are rolling chairs that are often scattered around the SIUE senior project laboratory (EB2029). Our solution will be comprised of one or more robots controlled by custom software developed by the RAFS team members. Ideally, the robot(s) will be able to identify unorganized chair placement and move the chairs to open computer desks. While doing this, the robot(s) may be required to deal with unexpected objects or situations. These exceptional conditions will be discussed later in this document.

2.0 Project Overview

The RAFS team was requested to design this software package for the customer Dr. William White. While not explicitly stated, this software may be used by anyone who is familiar with its operation. A considerable amount of effort will be exerted to accommodate computer users of various skill levels.

Throughout the document terms are used to refer to specific situations, items and concepts. The following section clarifies the precise meaning of such dictionary items.

- ***Robot*** – The use of this word throughout the document will serve the same purpose as the notation *robot(s)*. This distinction is made to avoid awkward subject verb agreement in sentences. We are unclear at this time whether the project will include one or more robots; therefore this redefinition is absolutely necessary.
- ***Ordinary operation cycle*** – This term shall represent the following cycle of operation.
  1. Robot executes startup procedure, default chair arrangement procedure selected.
  2. Robot identifies misplaced chairs and empty desks.
  3. Robot places each chair, one at a time, into empty desks.
  4. Robot executes shutdown procedure.
- ***Desk*** – An empty or chair occupied computer station.

3.0 Project Objective

- ***3.1 Functional Requirements***
  - Enable robot to recognize an object and determine appropriate behavior
  - Advanced object analysis leading to the recognition of specific objects such as chairs, tables and desks.
  - An appropriate level of error handling is needed to cope with unexpected objects or situations.
  - Facilities for variable levels of accuracy and speed of chair organization.
- ***3.2 Nonfunctional Requirements***
  - Development will take place in the Linux environment.
  - At this time we are aware of two robotics APIs that are available for this project. We will be reviewing each to determine the appropriate API for our project.

4.0 Project Scope

The RAFS project is primarily concerned with the core functions of object recognition and placement. Although we aim to handle as many exceptional conditions as possible, only a limited subset of error states will earn our attention.

- ***4.1 In Scope***

  Within this section we will define goals, deliverables and progress milestones. We plan to structure our work in such a way that each milestone will yield a fully functional, and to some degree, tested, software module.

  - *4.1.1 Goals*

    - *Recognizing an object and being able to determine whether or not it is a chair or desk.*

    - *Maneuvering in an area and being able to avoid objects.*

    - *Identifying an empty desk.*

    - *Enabling the robot to grasp and move chairs.*

  - *4.1.2 Deliverables*

    Each goal will produce a completely functional standalone software module. Using this approach will ensure quality-tested platforms to begin each next phase of development. Using this development approach will help expand the SIUE repository of example robotics source code.

  - *4.1.3 Exceptional Conditions*

    Inappropriate behavior can result when the robot encounters objects and/or situations it is not programmed to deal with. We plan to tend to certain situations, which may interfere within an ordinary operation cycle. The objects/situations causing this behavior can be sorted into two categories, dynamic and static objects/situations.

    - 4.1.3.1 Dynamic Objects/Situations

      Objects that fall into this category are those that do not belong to the original layout of EB2029. The room is equipped with collapsing wall dividers and rolling tables. At any time these objects may be placed in various locations throughout the room. As a result, we may add facilities to determine room layout during each operation cycle. Several objects may not necessarily move, however, they may produce a situation the robot needs to deal with. Loose cabling or carelessly placed objects, i.e. temporarily stored boxes, could become obstacles for the robot. Facilities must be in place to avoid such objects and/or correct situations caused by them, i.e. wheel caught in wires. In worst case scenarios it would be appropriate to power down the robot and terminate the operation cycle.

    - 4.1.3.2 Static Objects/Situations

      Objects that fall into this category are those that belong to the original layout of EB2029. The room is equipped with several computer stations that we will refer to as desks. Supporting the ceiling are poles that reach from floor to ceiling. Two large tables reside in the center of the room. Although these tables may be moved, we will assume they are returned to their proper place. We make this assumption based on the markings on the floor that indicate the appropriate placement of the tables. It is fairly clear that the *tables belong in this position and should not be moved for any extended period of time. At certain parts of the floor are small mound like wire covers. The robot alone may roll over the covers with ease, but there may be a concern while the robot is moving a chair. We will determine at a later time if the covers are something that must be avoided. If confronted with an unrecognizable object, the robot will either avoid the object or enter the idle state until assisted by user.*

- 4.1.3.3 Random Objects/Situations

  In some instances the robot may encounter foreign matter while performing an ordinary operation cycle. In event that these objects/situations are unpredictable we must not be responsible for the behavior of the robot. An example of this would be water or some other foreign matter either on the floor or in the path of the robot. These conditions are beyond the scope of arbitrary collision detection.

- 4.1.3.4 Environment Issues

  There exist several variables within EB2029 that may play a role in the ordinary operation cycle. Room lighting may influence the accuracy of sonar and/or laser guided object recognition elements. In addition, room temperature may affect robot operation, however this case is much less likely. We must also consider the possibility of the floor being waxed on occasion. This could pose a problem with the steering and wheel elements.

- ***4.2 Out of Scope***

  To precisely define what we are trying to achieve we must sometimes define what we are not trying to achieve.

  - ***4.2.1 Exceptional Conditions***

    While it would be ideal to accommodate all exceptional conditions, it would be overly optimistic to strive to do so. We will only be able to handle conditions that are of the highest priority. High priority situations are those in which the robot, SIUE equipment, and/or spectators are in immediate danger.

5.0 Project Estimated Cost/Effort/Duration

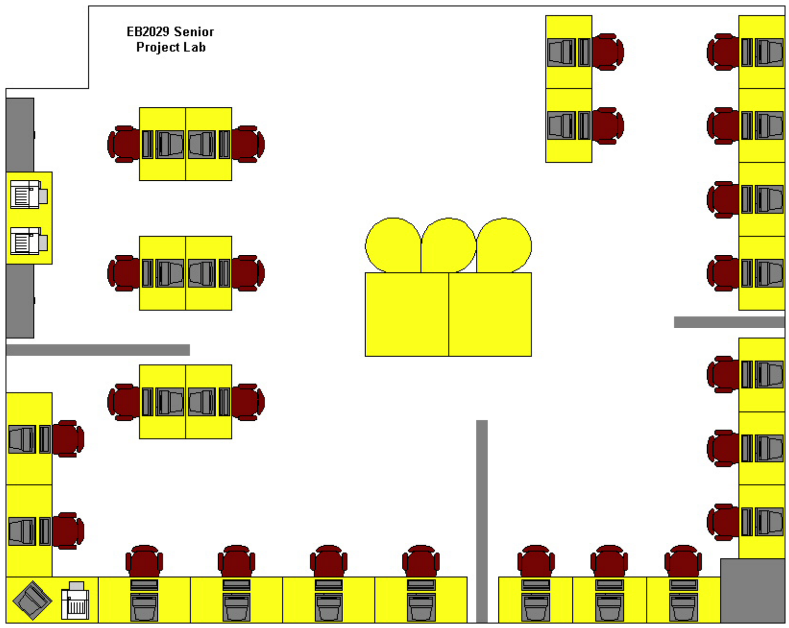
- ***5.1 Estimated Cost*** – All hardware for this project has already been purchased. A visual development environment may need to be purchased to simplify development in the Linux environment.

- ***5.2 Estimated Effort Hour –*** *Estimated time given to us at the beginning of the CS425 semester was 10 hours per person per week. Given this, probably 4 hours (10 hours –3*2) per person per week for the project. Therefore, (4 hours * 4 people * 14 weeks/semester * 2 semesters) = 448 hours. This is a low estimate. We will add 25% for a total of 560 hours.*

- ***5.3 Estimated Duration –*** *2 semesters and some possible work over summer semester yields 8 months of duration.*

6.0 Project Assumptions

- The project will be performed in the Engineering Building (EB2029).



- The door to EB2029 will be closed during the ordinary operational cycle. This assumption is in agreement with the Department of Engineering's policy regarding the secure status of this room.

- There will be a limited amount of people in the room during the ordinary operation cycle. We do not anticipate more than Dr. White, the RAFS group members and a few spectators.

- The square tables in the middle of the room will remain stationary.

- The computer desks will remain stationary.

7.0 Project Risks

- The project may be too complicated to be successfully completed in the time allotted. However, there appears to be ample time (in the neighborhood of 10 months) indicating this risk may not be a significant issue if consistent progress is made. ***(Risk assessment: HIGH)***

- The robots are shared resources in the School of Engineering and this brings up several issues. First, currently any student in the school of engineering with an electrical engineering account can access the robot. Secondly, the robot can be controlled via a LAN/WAN and is susceptible to viruses and other problems that plague networked computers. ***(Risk assessment: HIGH)***

- The project may not be able to achieve all goals as set out by Dr. White or those stated in the PDD. Although the PDD process provides the team an opportunity to identify which goals are achievable, necessary, or unattainable. ***(Risk assessment: MEDIUM)***

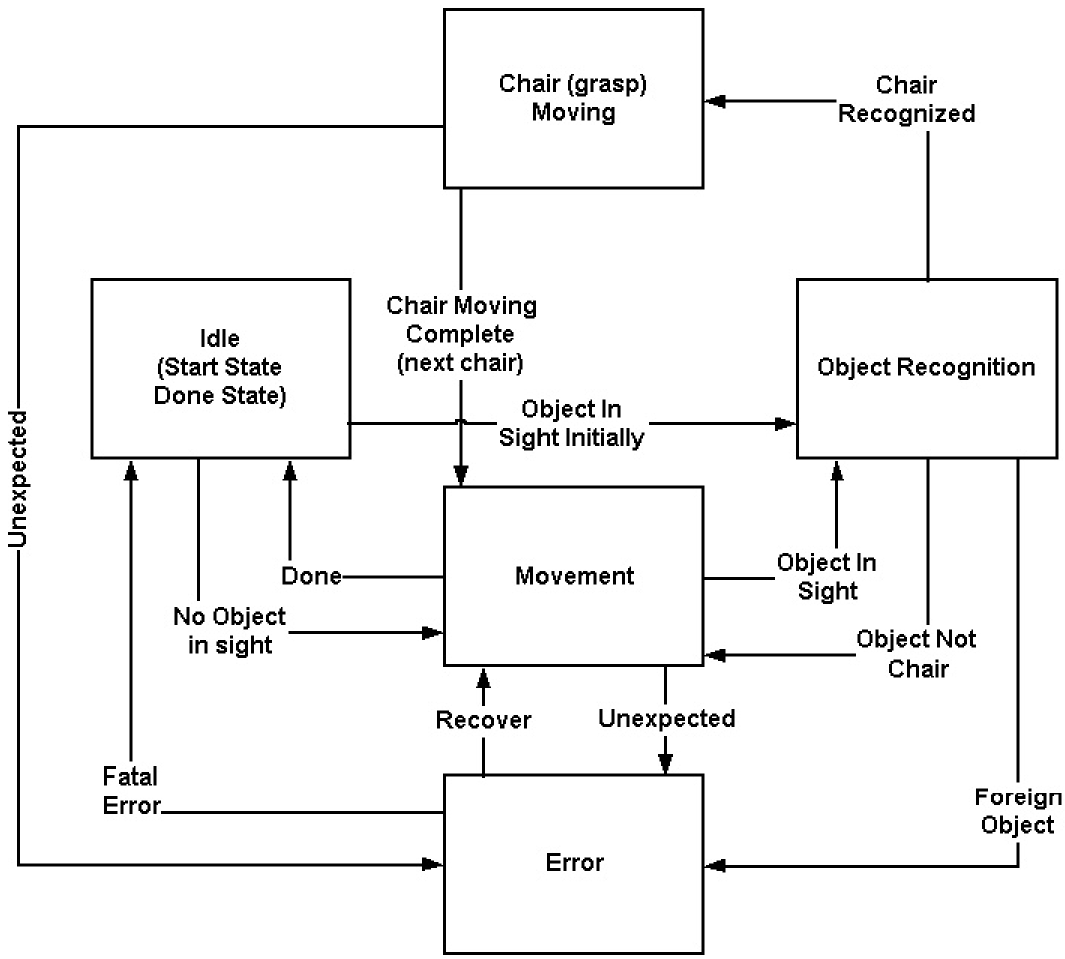
- The robot may pose a damage risk to property in the room such as tables, computers, printers, chairs, or walls. However, the robots have an emergency "OFF" switch. In the event that the robots damage SIUE property, they could always be quickly powered down. The robots also have image processing and object detection capabilities that could be used to avoid damage to objects in the room. ***(Risk assessment: MEDIUM)***

- The robot may present a physical threat to people who are in the room at the time of robot operation. On the other hand, the robot will most likely be used after hours, to put chairs back when students have left the room. When the robot is run under these conditions, only the team members and faculty may be present. Also, most operators could get out of the danger's path long before any injury could happen to him or her. Finally, the image processing and object recognition capabilities of the robot could be used in this instance as well. ***(Risk assessment: LOW)***

8.0 Project Approach

- ***8.1 State Diagram***

  In this section we will highlight a state diagram that describes in detail the ordinary operation cycle. Each state will be detailed with conditions responsible for its invocation.



Idle (Start/Done State)

- Robot is powered up and initialized with startup data.
- The task is complete when no remaining chairs require placement.
- Robot may experience a fatal error causing it to go idle.

- If the object is in sight initially, the robot will go directly to recognizing the object.
- If no object is in sight, the robot will move and scan the area to try to find an object to recognize.

Movement

- If the robot does not initially have an object in sight, seek an object.
- If the robot finds an object, it tries to recognize the object.
- If the robot encounters an error, it tries to recover from the error and resume movements seeking a new object.
- When the chair movement is complete, the robot seeks a new object for recognition.
- If the object in sight needs not to be considered, the robot begins seeking a new object.

Object Recognition

- Robot approaches an object and attempts to determine the objects type.
- Robot initially has an object in sight and attempts to determine object type.
- Robot must be able to distinguish the difference between a chair, static objects, and dynamic objects.
- If the robot recognizes an object as a chair, it grasps the chair and moves with the chair grasped.

Chair (grasp) Movement

- Robot recognizes an object as a chair, grasps it, and then begins the movement process.
- Once the chair is put in the proper place, the robot will move and scan the area to try to find an object to recognize.

Error

- Robot encounters a problem during the chair movement state.
- Robot encounters a problem during the movement state.
- Object recognition fails due to unidentifiable foreign object.
- If the error is one where the robot can recover, the robot will resume the movement process.
- If a fatal error is present, the robot will halt and go into the idle state.

The robot we will be using has a variety of different subsystems we could use for different applications. It has the capability of sonar, which could possibly be our "eyes" for this project. The robot could scan the room with this capability. The camera that is on the robot is also a possibility for being our “eyes”. The robot also has a set of bumpers, which could be used in some conjunction with the "eyes" as collision detection. Also, the robot has a gripper, which can be used to grasp the chair and maneuver it. All of these tools are here for our benefit, but we do not know to what extent we will use them.

We have decided to read up on the documentation on the website for the robots (robots.activmedia.com). We have also decided we are probably going to want to interview a few people; the NORRT project, Andrew Lamonica, and Dr. Umbaugh are a few examples. We will use Dr. Umbaugh's book as a reference. We have also decided we will observe how the robot functions, how it accomplishes basic tasks to get a better understanding of how it works, and how to program it.

9.0 Project Organization

- ***9.1 Team Roles***
  - ***Peter Motykowski***

    Primary Role: *Team Leader*

    Secondary Role: *Website Designer/Analyst*
  - ***Matt Allen***

    Primary Role: *Unassigned*

    Secondary Role: *Analyst*
  - ***Bradley White***

    Primary Role: *Unassigned*

    Secondary Role: *Diagram artist/Analyst*
  - ***J.D. Pohlman***

    Primary Role: *Unassigned*

    Secondary Role: *Documentation/Analyst*
- ***9.2 Team Website***

  Most current and up to date information can always be found at the RAFS website located at http://cfs2.siue.edu/sp/s02g2 or an easier the easier URL http://tofu.cs.siue.edu/rafs