RAFS - Robot Aided Feng Shui

風

水

Project Plan Document (PPD)

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

Table of Contents

1. Project Overview

The RAFS team was requested to design a 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.

1.1 Purpose, Scope, and Objectives

Purpose

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 will be able to identify unorganized chair placement and move the chairs to open computer desks. While doing this, the robot may be required to deal with unexpected objects or situations. These exceptional conditions will be discussed later in this document.

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.

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.

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.

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.

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.

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 which needs to be dealt with by the robot. 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.

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.

Random Objects/Situations

In some instances the robot may encounter foreign matter while performing an ordinary operation cycle. In the 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.

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.

Out of Scope

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

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.

Objectives

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.

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, ARIA and Saphira. We will be reviewing each to determine the appropriate API for our project.

1.2 Assumptions, Constraints, and Risks

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 Computer Science'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.

Constraints

- The project will be performed in the Engineering Building (EB2029).
- We are limited to robots owned by SIUE. This may include one or both (the acquisition of a third robot is currently pending).

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 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)

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. However, since this risk was initially realized, Dr. Weinberg has been gracious in allocating the smaller robot specifically for our group's senior project use. (Risk assessment: LOW)

1.3 Project Deliverables

Paper Deliverables

- Project Definition Document due on 2/06/02
- Project Plan Documentation due on 2/27/02
- Project Design Documentation due on 4/03/02
- Project Prototype due on 4/8/02
- Final Documentation due on 4/17/02

Presentation Deliverables

- Project Definition Slides due on 2/13/02
- Project Definition Documentation Presentation on 2/20/02
- Project Plan Slides due on 3/06/02
- Project Plan Documentation Presentation on 3/18/02
- Project Design Slides due on 4/10/02
- Project Design Documentation Presentation on 4/17/02
- Project Final Documentation on 4/22/02

Goal 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.

1.4 Schedule and Budget Summary

- Project Definition Document due on 2/06/02
- Project Definition Slides due on 2/13/02
- Project Definition Documentation Presentation on 2/20/02
- Project Plan Documentation due on 2/27/02

- Project Plan Slides due on 3/06/02
- Project Plan Documentation Presentation on 3/18/02
- Project Design Documentation due on 4/03/02
- Project Prototype due on 4/8/02
- Project Design Slides due on 4/10/02
- Final Documentation due on 4/17/02
- Project Design Documentation Presentation on 4/17/02
- Project Final Documentation on 4/22/02

1.5 References

- The NORRT project team
- http://robots.activemedia.com
- Dr. White
- Dr. Weinberg

1.6 Definitions and Acronyms

- 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
- IDE – Integrated Development Environment
- GNU – GNU is Not UNIX, open source development project
- CVS – Concurrent Version System
- GPL – GNU Public License
- HTML – Hyper-Text Markup Language
- Cygwin – A UNIX API system implementation for Win32. http://cygwin.com
- Frontier2 – The smaller of the two robots
- XWindows – A Window-like GUI environment for UNIX platforms
- XFree86 – A free XWindow implementation that will be used in our project
- SSH – Secure Shell.
- XDMCP – XWindow Display Manager Control Protocol
- R&D – Research and Development
- PHP – PHP Hypertext Preprocessor
- UNIX/Linux – These terms are used interchangeably throughout this document, dealing with the platform we will be working with
- NIS – Network Information Systems
- SOENT – School Of Engineering Windows NT network

2. Project Organization

2.1 External Interfaces

The software that came with the robot is designed to run on a UNIX-based platform. It does come with a version for Windows that is not very reliable at this point, so we are using the UNIX version. We are getting some information from ActivMedia's website to help with the learning curve with the robot. It explains how to initially set up the robot, and provides demo programs to look at.

Our project depends heavily on the new release of ARIA. ActivMedia is scheduled to have a new release out some time in the middle of March. This new release is supposed to have more high-tech features, especially in dealing with the camera. The newer version is supposedly easier to work with, but will not be as “tried and true” as Saphira.

CodeWarrior and C*Forge are other products we would like to use. These are IDEs for Linux that are similar to Visual C++ for Windows. We would need to purchase these through the university and install them on the computer(s) we will be using for this project. Any documentation that comes with these products will be helpful to us to learn the program, and our progress depends upon the documentation being well written.

The customer is Dr. White. He wants us to put all of the chairs that are scattered across the lab in room 2029 in the Engineering Building in the correct places. The correct places are where a desk with a computer station is located. He specified that he wants us to do this procedure with one or both of the robots. He does not care how it gets done. He did not specify how long the procedure should take. He said it could be left overnight as long as the robot will not run into a wall or anything.

The NORRT team will interact with the project for the simple fact that they are using a robot for a project right now. They could be of some assistance in what works and what does not. We could learn different aspects of how the robots work and pass that along to them, and vice versa. We do not know to what extent we will interact with them because we might even use a different API. A corrected version of a supposedly better API is supposed to be released in time for our project.

2.2 Internal Structure

Within our team there are several methods and standards we have set to ensure the steady flow of ideas and information. At the heart of the organization is the project management website. We are running **moreGroupWare** as a means of communication for the group. Peter has altered the source code of this GPL product to work in special ways for our group. Anyone in the group can post to the main project website via the project management site. This avoids having to update files and awkward HTML. The site has calendar features, forums, link management and several other areas, which aid project management.

File naming and document management

To aid in the management of files making up our project we created a few rules for file naming. We are storing documents in ftp://csfs2.siue.edu/sp/s02g2. We create a directory for each work item such as pdd/ and ppd/. As progress is made on each document, older versions of work products will be placed in an archive folder, i.e., /pdd/archives. The ftp directory structure is in the following format:

```
/sp/s02g2 > ls
```

```
ppd/        index.html      Minutes.doc
pdd/        Remote robot.bat
```

- *Index.html* is a html file containing a pointer to our main website located at http://tofu.cs.siue.edu/rafs
- *Minutes.doc* is a Word document containing the minutes from RAFS team meetings. This file is maintained by J.D. Pohlman.
- *Remote Robot.bat* is the file that starts the cygwin xfree86 server and opens an xwindow session on the smaller frontier2 robot. Requires cygwin and xfree86 to be installed on the client computer.

pdd_Document_1.3.1.rtf

Documents are started at version 1.0 and the minor section is incremented when sections are completed. If a release is made with grammar and spelling corrections and no additional sections, a release field is added, i.e., 1.3.1.

Inter-group communication.

News is posted on the project management website. Team members are expected to check this site for updates daily. Team meeting information is posted here along with discussions regarding the current phase of software development. Email is also utilized to quicken communication and to ensure everyone will see a piece of information by a certain time.

Documentation Management

J.D. Pohlman is responsible for assembling documents from team member work submissions. Work is submitted to the ftp site in the folder of the current document. Upon completion of a document, a team member will convert it to HTML for the main project website. HTML versions will be constructed on a later schedule than the rich text format.

2.3 Roles and Responsibilities

Introduction

There are many roles and responsibilities for the Robot Aided Feng Shui project. Some of these responsibilities are minor, while others are vital to the success or failure of the project. The team members will fulfill these listed requirements as a minimum contribution. There are many other tasks, items, and responsibilities not listed that have not been discovered yet. The current roles and responsibilities for the Robot Aided Feng Shui group are as follows:

Peter Motykowski is the group's appointed leader. Peter's primary responsibilities include general leadership tasks, such as leading team meetings and keeping all members motivated and productive. Peter will also take the lead in introducing the group to new technologies and software and will give the group direction in what tools to use in working on this project. Peter will also contribute to the team's project management website, along with the team's website to the world showing the project's up-to-date progress. This site can be seen at http://tofu.cs.siue.edu/rafs.

J.D. Pohlman is the group's head documenter. Some of J.D.'s responsibilities include maintaining the most recent version of documents (whether they are actual paper documents or file management in the software configuration management side of our project). J.D. is in charge of combining all documents into one, final document. J.D. is also a contributing analyst and programmer to the RAFS project and is doing a great deal of research work with the hardware (robots) to be used.

Matt Allen has the role of “publicist” for the group’s internal documentation. The responsibilities of this position include establishing threads for conversational topics on the group’s management website (http:// tofu.cs.siue.edu) and also creating and updating links on the management website. Matt will also be a contributing analyst and programmer to the project.

Brad White is the group’s head of the graphics portion of the project. Brad’s responsibilities include (but not limited to) drawings, general charts and graphs, Gantt charts, room layouts, and organizational hierarchies. Brad is also a contributing analyst and programmer for this project. Brad will take care of items that need to be drawn and will take a large part in measuring and recognizing milestones and making sure the project stays on a timely track.

All Team Members have some responsibilities that are identical. Some of these include:

1. Document writing contribution
2. Source code writing contribution
3. Meeting attendance
4. Taking part in the discussion forums on the project management website
5. Hardware research
6. Software research
7. Learning new API’s and programming languages

And others yet to be developed

3. Managerial Process Plans

3.1 Start-up Plan

We decided that we are going to start research and development sessions. The group agreed that late Friday afternoons would now be spent covering a research topic that will be a core part of the Project Design Document. This activity will be complemented with meetings on Monday to wrap up issues covered on Friday and document those activities. Wednesday will be spent working on the Project Design Document. More detailed scheduling information can be found later in this document.

3.1.1 Estimates

We are estimating that research topics will take a fixed amount of time. We have allocated one full Friday (approximately 8 hours max) and some of Monday to each topic. We are basing our estimation solely the number of topics and the amount of time to cover them. Dividing the research topics in this fashion ensures each will earn our attention during the construction of the Project Design Document.

Starting 02-27-02, our weekly schedule will follow this pattern:

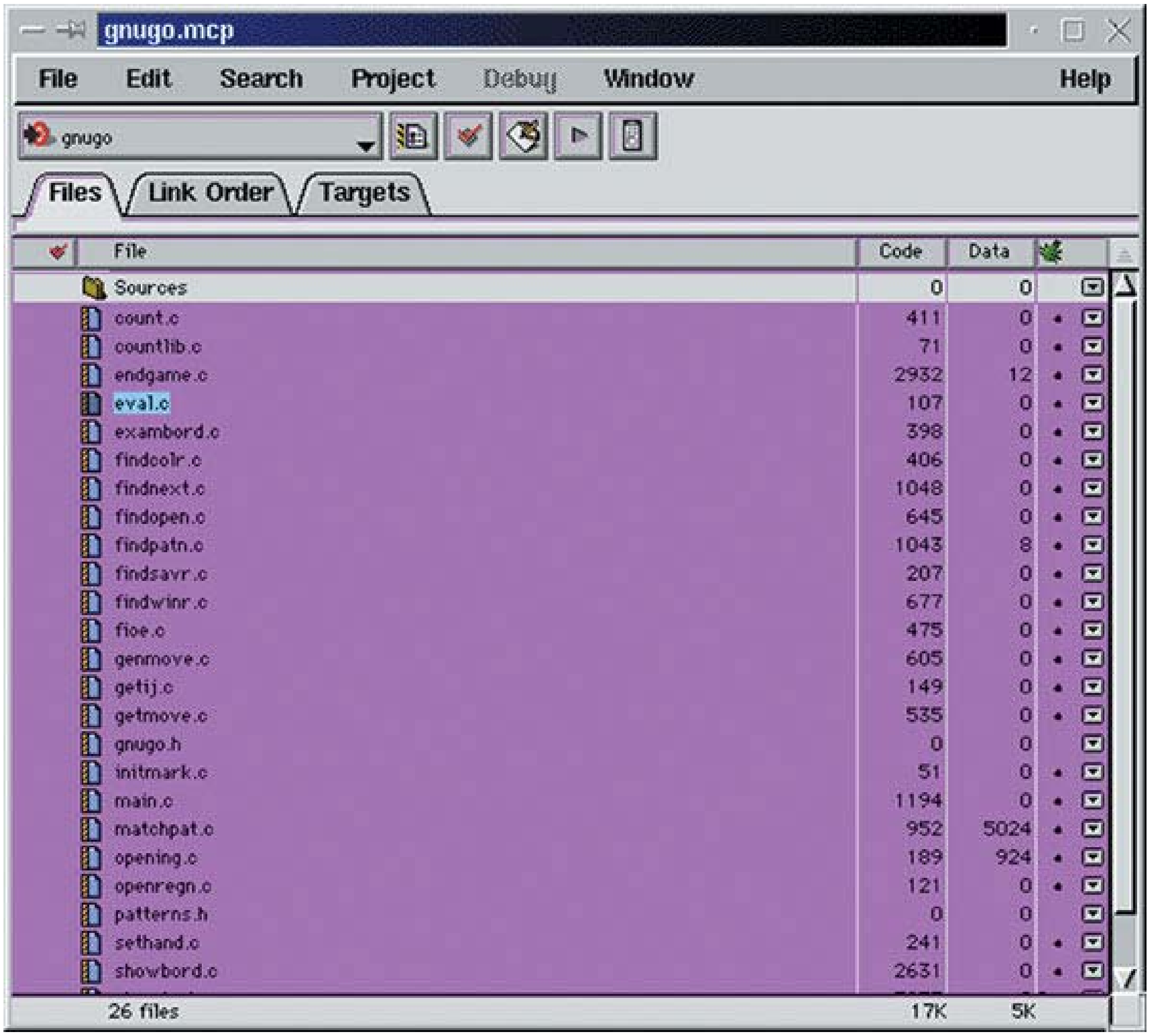
Friday - Research meetings
Monday - Research/Document meetings
Wednesday - Document meetings

3.1.2 Staffing

As our research and development matures, we will be expressing interest in different areas. The individual will expand upon the areas of most interest to him. For instance, if John really takes charge during the laser R&D day, and he is interested in being in charge of that part of the project, John will be labelled as the “laser expert.” If anyone has a question about the laser, John should be the person to go and talk to.

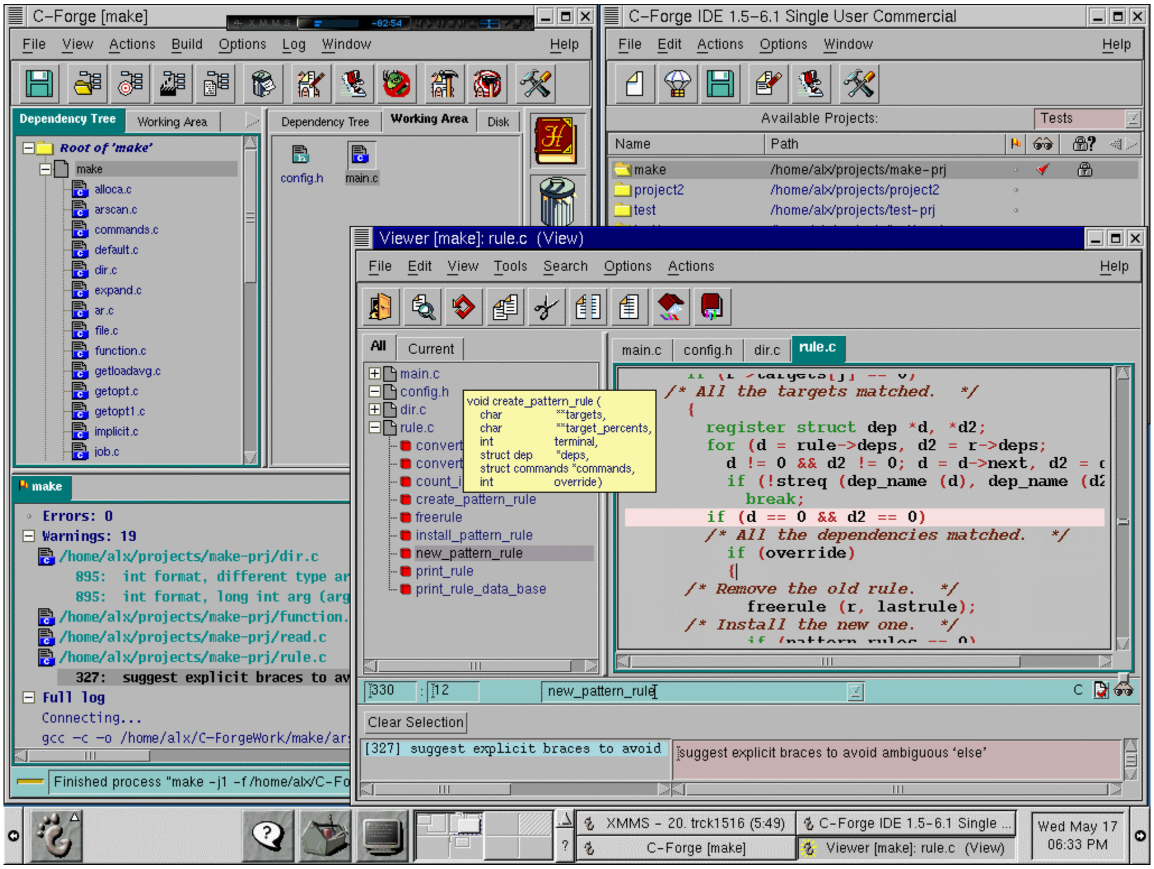
3.1.3 Resource Acquisition

We cannot underestimate the value of a visual development environment. While it would be staying true to the UNIX purist style of development, text editors and command line compilers will not suit our development time schedule. We have referenced an article published by Linux Magazine to help us narrow down our choices of IDEs (http://www.linux-mag.com/2001-04/IDE_01.html). From what the article says, CodeWarrior will be a top choice for our project. It has been around for several years and is part of the curriculum at many universities. We have been unable to obtain an evaluation copy for preview purposes. However, the screenshots and reviews provide adequate information regarding the software suite.



File	Code	Data	
Sources	0	0	
count.c	411	0	•
countlib.c	71	0	•
endgame.c	2932	12	•
eval.c	107	0	•
exambord.c	398	0	•
findcolr.c	406	0	•
findnext.c	1048	0	•
findopen.c	645	0	•
findpatn.c	1043	8	•
findsavr.c	207	0	•
findwinr.c	677	0	•
fioe.c	475	0	•
genmove.c	605	0	•
getij.c	149	0	•
getmove.c	535	0	•
gnugo.h	0	0	
initmark.c	51	0	•
main.c	1194	0	•
matchpat.c	952	5024	•
opening.c	189	924	•
openregn.c	121	0	•
patterns.h	0	0	
sethand.c	241	0	•
showbord.c	2631	0	•
26 files	17K	5K	

Another top choice is C*Forge by CodeForge. This IDE seems to be more of a front end to the common set of Linux development tools. This suite was given rave reviews and is said to be an extremely stable development environment. It possesses features such as a visual debugger, *MakeFile* generator and automated CVS support. At this time in our project it is safe to say we like to use this environment. Although it carries a higher price tag than CodeWarrior, the additional features and stability would be invaluable to us.



Name	Path
make	/home/alx/projects/make-prj
project2	/home/alx/projects/project2
test	/home/alx/projects/test-prj

```
Errors: 0
Warnings: 19
  /home/alx/projects/make-prj/dir.c
     895:  int format, different type ar
     895:  int format, long int arg (arg
  /home/alx/projects/make-prj/function.
  /home/alx/projects/make-prj/read.c
  /home/alx/projects/make-prj/rule.c
     327:  suggest explicit braces to av
Full log
  Connecting...
  gcc -c -o /home/alx/C-ForgeWork/make/ar
```

Finished process "make -j1 -f /home/alx/C-Fo

```
void create_pattern_rule (
    char                **targets,
    char                **target_percents,
    int                 terminal,
    struct dep          *deps,
    struct commands *commands,
    int                 override)
```

```
      /* All the targets matched.   */
        {
          register struct dep *d, *d2;
          for (d = rule->deps, d2 = r->deps;
            d != 0 && d2 != 0; d = d->next, d2 = c
            if (!streq (dep_name (d), dep_name (d2
              break;
          if (d == 0 && d2 == 0)
          /* All the dependencies matched.  */
            if (override)
            {
    /* Remove the old rule.   */
              freerule (r, lastrule);
    /* Install the new one.   */
```

[327] suggest explicit braces to avoid

suggest explicit braces to avoid ambiguous 'else'

We are not interested in dealing with flawed beta testing IDEs for our project, therefore it is likely we will choose one of the top choices we have listed.

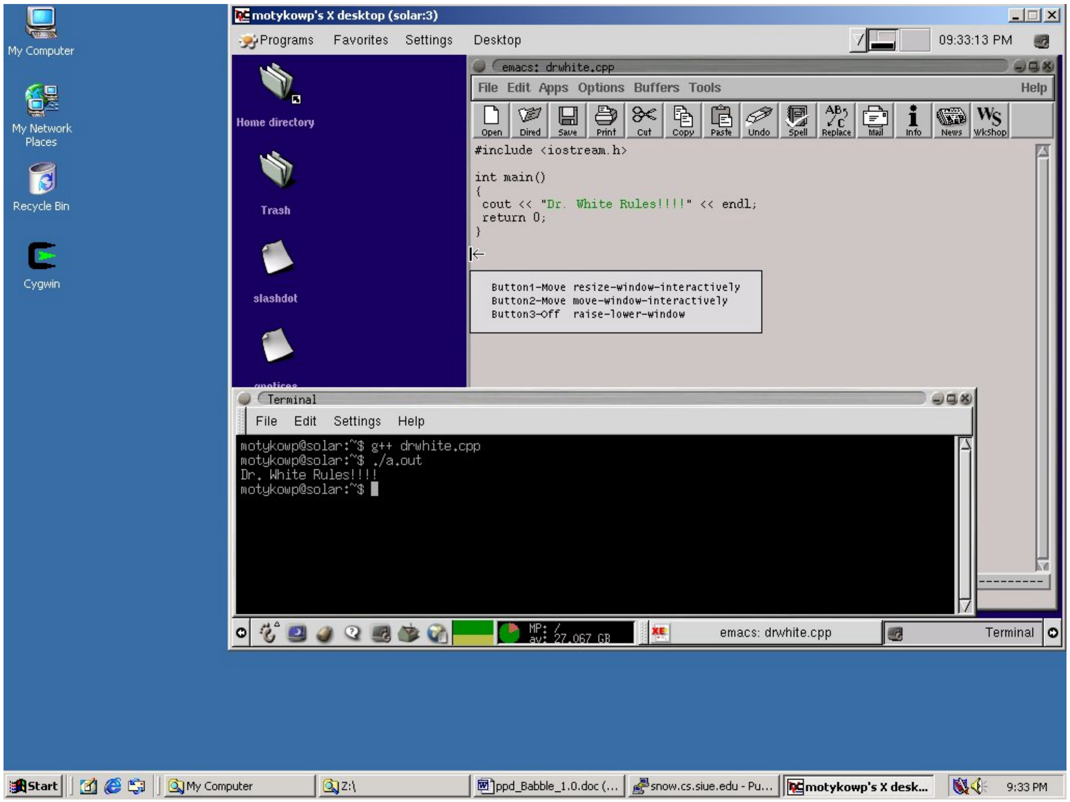
Although we do have some hardware and software needed for this project (namely the Robot and XWindows), we plan on purchasing a few things for this project through the university. First off, we would like to purchase CodeWarrior or C*Forge (maybe both). We are also aware of a second (and maybe third) wireless Ethernet that could be used in EB 2029. The one the university already has is now downstairs, because of the Engineering Building open house. The NORRT team is debating on using their robot downstairs until they receive the extra wireless Ethernet(s) to put upstairs. We are also requesting Visio 2000 Professional Edition. We are planning on asking Dr. Weinberg to purchase the products we will need with the money for the robots. We plan to ask Dr. Weinberg toward the end of this semester for the products we will need next semester.

3.1.4 Project Staff Training

With a new software development platform comes training needs. We will all need to spend a considerable amount of time learning the chosen IDE. We hope that our advanced skills in computer programming will make this training quick and painless. We are all competent developers and do not foresee any huge problems within this training phase.

In general, the Linux environment is new to most of the team members. Peter is considerably experienced in Linux and will help the group get up to speed in a number of ways. Peter organizes a Linux/UNIX users group on the campus of SIUE and will encourage all team members to attend the meetings. These meetings will cover both advanced and beginning topics of UNIX/Linux. Overall, the meetings should help the other members of the RAFS group become more comfortable working in Linux. In addition, a collection of Linux documentation will be posted in the links area of the project management website. This collection will cover the basic usage of popular text editors (vi, emacs) and the use of command line development tools (gcc,g++,ld). Along with these links will be special group meetings that will focus purely on working in Linux. We find it imperative not only to be able to program in this environment, but to be comfortable working with files, network protocols and command shells. To ensure this project's success, each team member will need to exert a considerable amount of effort into learning the Linux environment. The Linux How-to project will be an invaluable resource during this period of training. This is a collection of documentation on how to do "just about anything" in Linux. Although there is a large amount of How-To documentation that does not apply to our project, there is a healthy amount that does. We will review documents regarding application development, in particular the use of threads, CVS and GNU gcc.

Although the majority of our work will be in the Linux environment, there are several Win32 applications that will play an important role in our project. To connect to the robots remotely, team members will use the puTTy SSH client. This client enables the team members to interact with the robot via a UNIX shell. Second, we will be using the cygwin UNIX environment for Win32. This suite provides remote XWindow access to the robots via XDMCP, enabling the user with full GUI control of the robot. Furthermore, we will utilize VNC for a second means of remote GUI access to the robot. This method provides a means for low bandwidth network connections to utilize the robot effectively. All team members will to some degree utilize the above software during the software development phase.



```
#include <iostream.h>

int main()
{
 cout << "Dr. White Rules!!!!" << endl;
 return 0;
}
```

```
Button1-Move resize-window-interactively
Button2-Move move-window-interactively
Button3-Off  raise-lower-window
```

```
motykowp@solar:~$ g++ drwhite.cpp
motykowp@solar:~$ ./a.out
Dr. White Rules!!!!
motykowp@solar:~$
```

VNC connected to solar.cs.siue.edu

3.2 Work Plan

3.2.1 Work breakdown structure

Our team will be taking a modular approach to our software development. Utilizing this method will enable team members to complete work units individually unless the organization of the modules states otherwise. This will make for neat and timely work unit assignments and avoid unnecessary group work that may slow down the development cycle. Although it is too early to give precise assignments of who will be working on each module, we could, for estimation purposes, construct a sample work breakdown structure.

3.2.1.1 Module Development

After our R&D sessions, we hope to have a clear idea of the module structure that will be used in the development of that particular subsystem/package. For example, this coming Friday (3/1/02) we plan to hold an R&D session regarding general robot movement. After this session we may have a clearer picture of what will entail the robot movement subsystem. When these modules are identified, a team member will become responsible for documenting and further developing them. When the module specification is completed, the rest of the team will review it, ensuring it follows module communication standards.

3.2.1.2 Documentation Development

So far in our project we have been both dividing up documentation work and also working together. Some parts of the PDD and PPD where general enough to be written by one team member and later evaluated for correctness by the rest of the team. This approach has been working well so far but has resulted in minor inconsistencies or wording and term usage. Over all, this has been an effective way of dispersing documentation work and later merging units into a polished complete document.

3.2.1.3 Website Maintenance

The RAFS website has been engineered to be a minimal maintenance website. Although some file editing will be required to keep document links up to date, it can be updated using web forms. The site is coded using PHP and HTML providing dynamically generated web content as pages are loaded. It is the responsibility of team member Matt Allen to keep the news area up to date. Here will be posted current information regarding the status or our project. Peter Motykowski will be maintaining the links and other HTML code related segments of the website.

3.2.2 Schedule Allocation

We plan on holding an R&D session for each item classified as a subsystem or package. We have a loosely planned structure so far for this approach and plan on further defining our subsystems and packages at a later time. The divisions of research topics concerning these subsystems are as follows:

- Robot Movement (with and without a chair)
- Robot to robot communication
- Object Recognition
- Image processing
- Logging Facilities

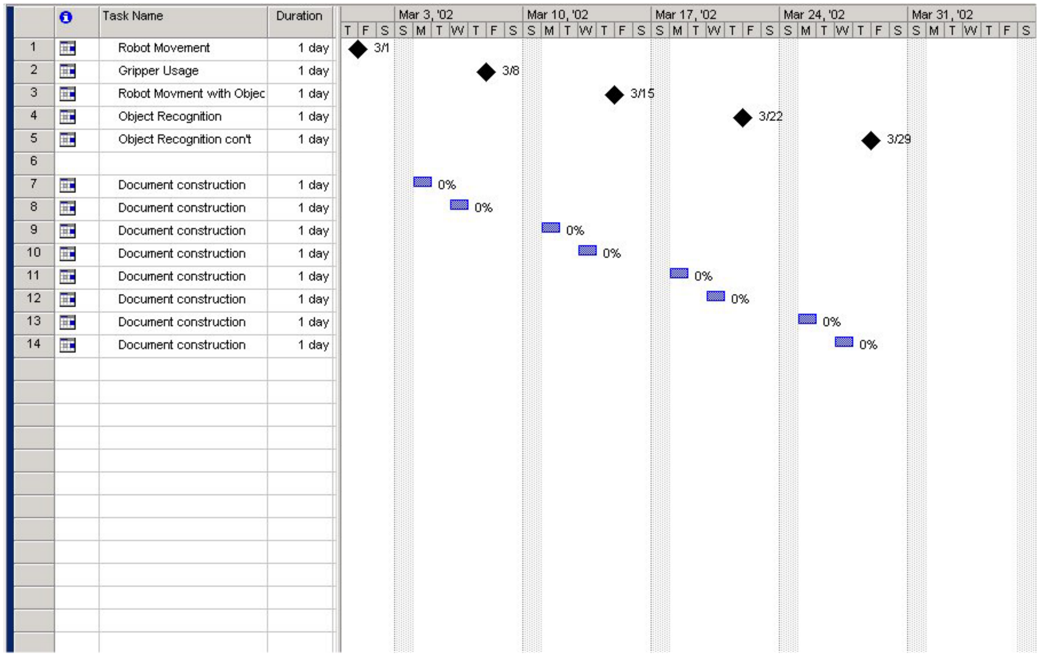
This list is incomplete and will be expanded as research topics surface from R&D meetings.

R&D meetings are considered core-required meetings that will be attended by all group members. In the unfortunate event that someone is unable to attend a meeting, the following actions will be taken. R&D meetings will yield a considerable amount of notes that will be organized into a usable resource to team members. If needed, a team member will review these notes with the absent member. All team members present the day of R&D sessions could be expected to provide review sessions for an absent team member.

In the event that the entire R&D session must be cancelled, an alternative date will be scheduled. A considerable amount of effort will be exerted to not have to push back other R&D meetings. The worst case would result in pushing back R&D session topics causing a shift in our schedule.

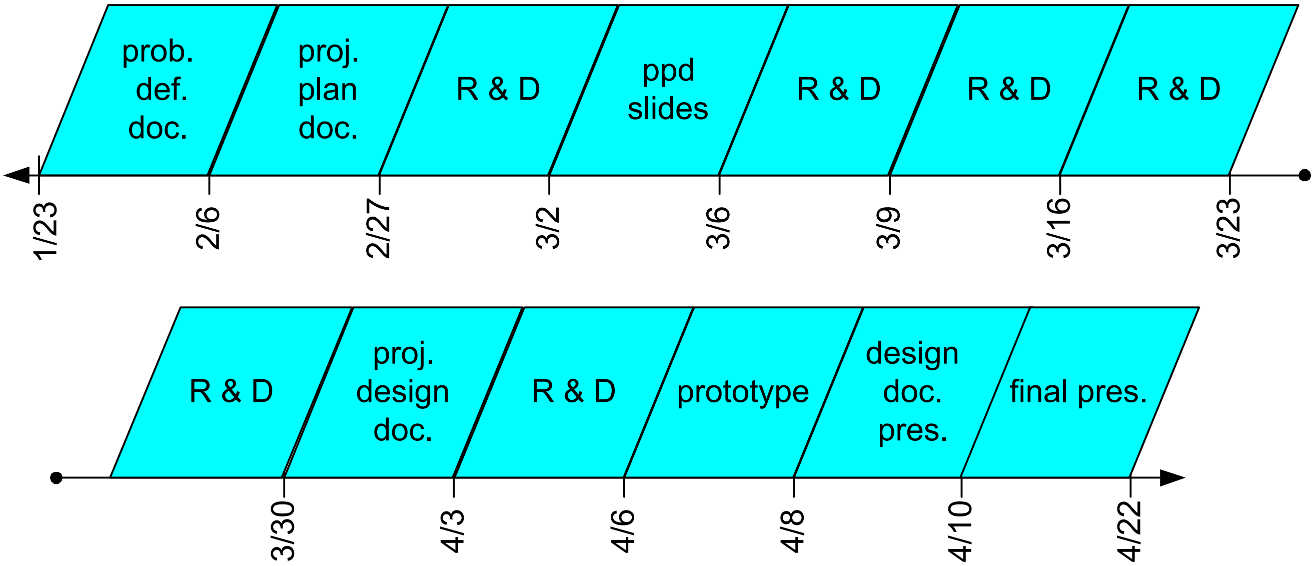
To an extent, each team member is encouraged to work on his own with regards to each R&D topic. To ensure the complete understanding of these concepts, individual work will be imperative. This scheduling decision will be in the hands of each individual team member. With regards to this, the project manager will assume everyone is doing his fair share of topical research. At any point during the project the team leader may ask to see notes and or code samples regarding R&D. If nothing can be presented it will be assumed that person is not conducting individual research. If this is the case, this team member will be assigned some individual research in which they will be expected to study, and present findings to the rest of the group. Such tactics will ensure we sufficiently cover the broad range of topics and concepts present in our project.

Our R&D topics will define planning milestones. When we have studied, evaluated and to some extent coded examples resulting in planning decisions, we will have reached a planning milestone. To be more specific, we will use the robot movement R&D as an example. During this R&D we will study selections from repositories of documentation provided by ActivMedia. With careful analysis we will filter out irrelevant material to our project. After evaluating approaches robot movement we will begin coding examples in an attempt to further our understanding of the concepts learned. Wrapping up this R&D would be several planning decisions reflecting our research of the material considered. After R&D sessions take place, team members will be expected to participate in discussion in the forums area of the project management website.



	Task Name	Duration
1	Robot Movement	1 day
2	Gripper Usage	1 day
3	Robot Movment with Objec	1 day
4	Object Recognition	1 day
5	Object Recognition con't	1 day
6		
7	Document construction	1 day
8	Document construction	1 day
9	Document construction	1 day
10	Document construction	1 day
11	Document construction	1 day
12	Document construction	1 day
13	Document construction	1 day
14	Document construction	1 day

The above Gantt chart shows our current R&D and documentation schedule for the project design phase. We will be utilizing several one-day sessions to conduct R&D sessions and constructing the project plan document as work and research progresses.



Introduction

The above diagram is a timeline for our project. Everything that takes place on the above two lines is part of the "Software Concept" and "Requirements Analysis" sections of our lifecycle model taken from the Design to Schedule Model (McConnel, 150)

Important Dates

1/23/2002 – Inception of project.

2/6/2002 – Problem Definition Document handed in.

2/27/2002 – Project Plan Document handed in.

3/2/2002 – Research and Development activity on the robot navigation system. The test will use the Colbert exercise. (http://www.ai.sri.com/~konolige/cs225b/materials.html)

3/6/2002 – Slides for Project Plan Document are due. Also on this day there will be some consideration of the logging facilities of the robot available and needed for the project to take place (what kind of statistics can the robot generate).

3/9/2002 – Research and Development activity for the gripper system of the robot. Also during this activity there will be research done on the robots collision detection system. This will be used with the information collected on 3/2 and 3/6.

3/23/2002 - Research and Development activity for Object Recognition. There are several people who we can discuss this topic with (Dr. Umbaugh, Dr. Weinberg, Dr. Noble) in addition to library and text research.

3/30/2002 - Research and Development activity for Object Recognition (continued).

4/3/2002 – Project Design Document handed in.

4/6/2002 – Prepare a Prototype for handing in.

4/8/2002 – Project Prototype due.

4/10/2002 – Project Design Slides due.

4/22/2002 – Final Presentation to Faculty.

3.2.3 Resource Allocation

Since no other groups are using the smaller robot, it has been allocated exclusively to our project. This gives us the freedom to do as we wish with the robot. We do not have to worry about another group using the robot when we are able to meet and work with it as well. This eliminates some problems we could possibly run into. If we decide to use the large robot, we would encounter these problems though. We also have access to the Senior Project Lab, which usually has enough empty computers, and enough space to complete the work for the robot.

3.3 Project Tracking Plan

3.3.1 Requirements Management

Our requirements are pretty well established already, but a small possibility exists of the requirements changing. If they would change, we would update our documents and post a thread saying the documents have changed. The website for such changes would be updated at http://tofu.cs.siue.edu/rafs/News.php.

3.3.2 Schedule Control

We will have R&D days on Fridays, with any research needed for that day being done between the Wednesday and Friday of that week. The group is encouraged to come up with some clear-cut ideas to bring for Friday, so the work will get done quicker. We must know beforehand what documents will be used for that day, so we do not spend a majority of the R&D day looking for some document or a clear topic. We can acquire these documents from ActivMedia's website, or anywhere else we can find information regarding the robots. Each individual is encouraged to bring somewhat of an outline to the meeting on Friday. This way we will have some kind of a lead for Friday's work. Monday will be more of a documentation day as opposed to an R&D day if Friday goes as planned.

3.3.3 Quality Control

When we write a section of a document, certain points must be met. These points will be discussed in a previous meeting, and if they are not met, the quality suffers. When a section is finished, it is supposed to have correct grammar, as well as content. It will be revised when combined with the rest of the document. After all the documents are fully combined into one document, a final revision will be made to check for grammar and content of the document.

3.3.4 Reporting

Some key decisions made thus far in the project include narrowing down the API to ARIA and Saphira, and using CodeWarrior or C*Forge. Most of our decisions cannot be made yet because of the nature of the project. We need to get more familiar with the robot and start using the different products. The new release of ARIA is due out in the middle of March, and we are relying on that date to start working with that API to narrow that area down to one. We need to purchase CodeWarrior and/or C*Forge (see section 3.1.3 Resource Acquisition) to learn more in depth about each IDE. We need to get through a few R&D sessions before a clear understanding is met. More ideas will also surface during the R&D sessions.

3.3.5 Project Metrics

Friday's work will tentatively be measured in modules of code that have been completed, or that at least have a good start. We will measure the progress of the implementation area of the project as modules of code completed. As far as the plan and design areas go, we will use the completed sections, along with charts for documents and updates to the team website as a means of measuring how the project is coming.

3.4 Priority Management Plan

The factors with the highest priorities will be the four main points we have decided as obstacles we need to address for this project. Recognizing a chair is the first of the high priorities. We need some way for the robot to be able to tell that an object in front of it is a chair as opposed to any other object. Desk recognition is also an issue we will have to address. These two objects are different from any other object in the room, whether it is dynamic or static. The other two high priorities deal with movement: with and without a chair. Without a chair will not be as large of a task as with a chair. Movement with a chair requires some way of measuring what is in front of the chair, and avoiding that obstacle. Other priorities will develop during the R&D phases. These priorities will be lower, namely medium and low.

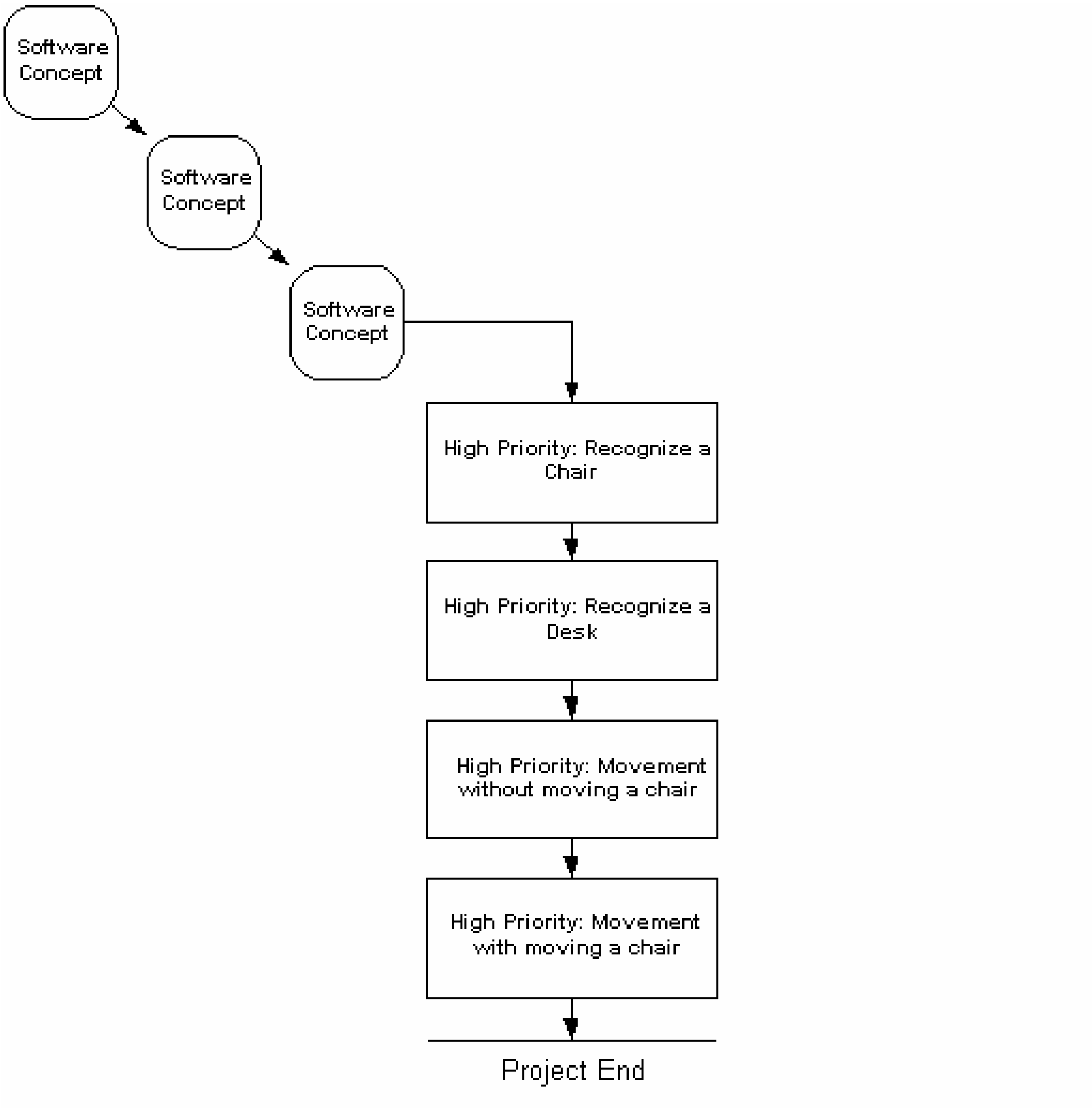
4. Technical Process Plans

4.1 Process Model

The model we have chosen for the Robot Aided Feng Shui project is the "Design to Schedule" model taken from the McConnell text book. We chose this model for several reasons. First we plan to develop our project in successive stages. This allows us to have something to release by the end of the CS499 class (although it may not be all we were hoping for). The development of successive stages guarantees that a product can be released by a particular date. The only disadvantage to this system is that we may lose some time in developing plans for features that we may not get to implement. Also this model enables us to stop when we run out of time.

At this time however, we only have high priority features listed (features that are critical to the success of the project). This model keeps us from wasting or losing time on non-critical features that would only get left out later in the process.

Design to Schedule Software Model


Software Concept
Software Concept
Software Concept
High Priority: Recognize a Chair
High Priority: Recognize a Desk
High Priority: Movement without moving a chair
High Priority: Movement with moving a chair
Project End

4.2 Methods, Tools, and Techniques

For this project, we will be using several tools:

- XWindow is the interface we will be using for development.
- The API(s) used for programming is/are Saphira and ARIA. At this time, we have not limited this to one or the other. In fact, we may need a hybrid of both because Saphira 8.0 is based on ARIA. So, to some extent we will have to be familiar with both. Our decision will be made shortly after the anticipated March release of the new ARIA.
- The actual coding will be done in C++ in the Linux environment. We are currently looking into CodeWarrior and C*Forge to add the feel of using Visual C++, in which most of the group is familiar with developing.

The project will be completed through the completion of each milestone and milestones will be completed as standalone modules. This will ensure that each milestone is fully complete and will work with all other following modules. By having standalone milestones, we are hoping that others will be able to benefit from our labors (i.e., future senior projects or robotics classes will have some good solid code that can be easily reused).

The process of development will be to design a milestone, build it, test it, and integrate it if need be. Documentation will be done as coded and then reviewed by the lead documenter.

Programming in the UNIX will require certain tools and techniques that deserve mention. Several tools were highlighted previously in the document; therefore we will talk about techniques. A publication has been circulating the Internet for some time now with regards to development in UNIX. Unlike other documents, this document approaches the mindset of UNIX development. UNIX lacks a lot of features that modern developers depend on, often leaving them to complain about inadequacies of the system. This document approaches UNIX from a different perspective. Overall, this document will be exactly what the RAFS group needs to get in the UNIX train of thought. The following is a quote from this publication (http://www.tuxedo.org/~esr/writings/taoup/), highlighting its approach to warming up to UNIX:

> This is a book about UNIX programming, but in it we're going to toss around the words `culture', `art' and `philosophy' a lot. If you are not a programmer, or you are a programmer that has had little contact with the UNIX world, this may seem strange. But UNIX has a culture; it has a distinctive art of programming; and it carries with it a powerful design philosophy. Understanding these traditions will help you build better software, even if you're developing for a non-UNIX platform.

4.3 Infrastructure

In order to ensure a stable development environment, several configuration settings will be considered. First, the operating system of the robot is a RedHat Linux 7.0 installation. This specific version of RedHat has known flaws, mostly concerning system security. Since the robot will be connected to the campus network, tending to these issues will be high priority. Second, Peter Motykowski will be the systems administrator of the Linux environments the RAFS project will be using. There are several computers that will be used for this project: First are the workstations in the senior project laboratories, which require little to no configuration. Second come the two Linux computers embedded in the robots. Additionally, there is a third system that is being used for Linux Java development. If at any point the RAFS project requires Java, this last system will be utilized.

In order to make user account management simple, the smaller robot’s Linux computer will be bound to the computer science NIS domain. This domain is an account database of all students in the SOENT domain that are computer science students. The system administrator will restrict access only to the students involved in the RAFS project or any other projects utilizing the smaller robot. This authentication scheme synchronizes the robot users’ accounts with the valid SOENT user accounts. The larger robot is under the administration of Adam Gregory of Electrical Engineering and is configured differently.

The system administrator will request a static IP address for the robot, avoiding the problems of dynamic IP addresses. In addition, a hostname entry will be requested, something symbolic such as http://rafs.cs.siue.edu would be nice for easier accessibility. This issue will be taken up with Earl Herman, the current administrator of siue.edu name servers.

A limited amount of networking hardware will be required to provide remote accessibility to the robot. Currently SIUE possess a single wireless Ethernet entry point for the robot to utilize. This equipment requires us to use the robot within a certain distance of this entry point. Talk has been going on with regards to the purchase of additional wireless network points. This would enable a robotics project using wireless Ethernet to work elsewhere in the building. Currently, we are restricted to operating the robot in limited areas of the Engineering Building.

4.4 Product Acceptance

Acceptance of deliverables will first go through the RAFS team, then to Dr. White, and then to the client. After this process, the deliverable will have been reviewed by at least two groups before the client reviews the deliverable. This should ensure the deliverable viewed by the client will have minimal problems.

- Documents - Will be reviewed for adequate and accurate content, grammar, spelling, and consistency.
- Code – Will be reviewed for adequate, consistent, accurate coding, and to ensure modules are standalone.

All deliverables are to be uploaded to the team directory, uploaded to each member's directory, posted on the website, and handed in to Dr. White. Periodic evaluations will be done by presentations of the delivered material on days set by the client.

5. Supporting Process Plans

5.1 Configuration Management

We will be using CVS for controlling source code. CVS is traditional for UNIX. We will adhere to the rules built into CVS by default (merge of code, etc.). We will also recycle the make files that are already in use by ActivMedia. The make files are on ActivMedia's website, and we will use them accordingly. We will also review minutes and make sure that older documents are kept up-to-date. Some things that could change would be: definitions to PDD, finding better times to meet for R&D days other than Friday, etc. We need to keep updating the old documents.

5.2 Verification and Validation

We will use several tools to verify and validate our projects. First, Dr. White will give us valuable feedback in regards to our project's progress. The RAFS team in general will be verifying everything is correct as the work progresses. J.D. will check over the segments while he is combining everything to one document, and if time permits, someone will check over the final document to make final revisions before the document is turned in.

5.3 Documentation

Documentation will be taken care of with the current version of the document (or visibility tool) being placed into the correct place on the ftp site. For example, our drawing of the room in which our project is to take place should be kept in the PDD folder on the site. If more than one version applies, the older version(s) should be placed in an archive folder. In this way, the files in the correct folder will be up-to-date and easy to find. We will also zip the archive folders to conserve space. As far as documentation for the code goes, we will use standard Hungarian Notation, and document thoroughly at the beginning of each module what that module does. We will also be keeping track of minutes taken during meetings.

5.4 Quality Assurance

We will follow the schedule given to us by Dr. White at the beginning of each semester. We will also follow a test plan that will be based on a module-by-module basis. Each module will undergo the process of design, coding, and testing individually. This way the testing will be done sequentially from module to module. It will be easier to test one module thoroughly before putting it together with other modules, and then test the coupling between the modules. We want the modules to be strong stand-alone units that accomplish a specific task, and are not interdependent on each other.

5.5 Reviews

We put a review plan into effect, which will basically review our Resource and Development stage from the Friday before to anybody that could not be at the meeting. We will hold reviews after hearing back from each milestone set by Dr. White. We will look at the reviews posted by Dr. White and accommodate changes accordingly. The client will approve any product deliverable we come across. These will be milestones set by the RAFS team.

5.6 Installation

We have the luxury of being able to design in the target environment, which makes installation a minimal problem. We will be writing the code on the machine itself, so no installation will be necessary for this project.

5.7 Problem Resolution

Problems with the website and the project management site will be taken care of by Peter. If problems during the R&D days should occur, we would talk to Dr. Weinberg and/or Dr. Umbaugh about them. Any general problems with the project (team member problems, etc.) would be discussed within the group, and if no decisions are made, we will climb the ladder to Dr. White. We hope the problems do not go this far, but it is there if necessary. Anything that is beyond our control will be taken care of by Dr. White.

5.8 Process Improvement

In order to improve the process of work getting done, we will try to ensure that Friday's work gets done on Friday, leaving Monday open to working on documents due. This is to ensure that we do not get wrapped up in the research and not get the documents done by the due date. We will stay on a better track. If someone has some free time, they can do some research on an individual basis. If they wanted to look into some aspect of the robot in greater detail, that would help improve the project.

6.0 Project Summary Status

At this point, the RAFS team has made initial design decisions, identified initial problems, and have begun extensive R&D on the robots and related systems. We have narrowed down the number of APIs being considered to two, and the final API will be selected after the newest release of ARIA is made public. Preliminary evaluation of IDEs has taken place and evaluation copies will be tested as they are obtained. We have considered the wide array of room configurations in which the robot must operate. These processes will continue throughout the development of the project. Other documentation of these processes will be made available as the milestones are met.