RAFS - Robot Aided Feng Shui

Post-Mortem Document

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

Table of Contents

1.0 Introduction

This document contains a general description and analysis of not only the problems but also what worked for the Robot Aided Feng Shui project. Some suggestions are made for better solutions if this project were to be done again and also suggestions for future enhancements and growth of the system.

2.0 Robot Limitations & API Limitations

First, it should be noted that the Robot, at best, is a blunt instrument that has extremely finicky tendencies. Some of the problems encountered in development with the robot were its constant snagging of wire covers causing the robot's motors to stall and its localization (guidance system) to become inaccurate.

Second, a great deal of the system's original API (ARIA) is an open source code. While this does have some positive implications as far as development goes, there is little or no reason for the original manufacturer (ActivMedia Robotics) to update or maintain the ARIA API. The latest version of this has been absorbed into their commercial proprietary API; however, it is not available for free. Also, it contains bugs and errors, and new versions are being released at a dizzying rate. (Three separate builds this year alone were made available to consumers.)

Third, Saphira API is closed source. This posed a problem when it came to dealing with and fully understanding this code. We would have liked to modify the code to use Saphira's API functions for gradient path finding, but couldn't due to the code not being open.

3.0 What we would have done the same

Most of this project was successful in achieving its goals. One of the things we would have done in the same manner is our regular meeting schedule. During normal times, the team would meet every Tuesday, Thursday, and Saturday. This led to each team member making sure things were done so the rest of the team could review them at the next meeting. Also, members could show their progress and talk about the next steps in the project.

Second, if possible we would also have a resident "local-expert" who was not a member of the team. Fortunately for our project Andrew Lamonica was helpful in answering questions about the ActivMedia robots and the Saphira software. His answers to some of our more simple questions saved us hours of valuable research time and days of coding test modules to try to find out how pieces of the software worked.

Third, we would have kept our filmed progress. This made good material for our website, so that people could see what we had working. Also, it proved invaluable in both semesters of our project. It showed our up-to-date progress in action and we had to fall back on it several times due to robot malfunctions.

Fourth, we would have switched roles in our group. JD proved to be an invaluable leader. The group needed Peter to bring us all up to speed and forge ahead in new areas.

If he had been leader, he would have been overwhelmed with too many leadership issues to do such a good job as lead programmer. This was a major switching of roles in our group, but this change was definitely for the best.

4.0 What we would have done differently

There are a few items we would have done differently that still would have enabled us to complete a successful project. First of these items would have been setting more achievable goals for each of the three successive releases. This is especially true of the third release where all of the previous knowledge had to be pulled together to form a completed project.

Second, we would have started code integration a bit earlier or at least researched this item before release 3. In the midst of release 3, we had a great deal of trouble integrating four existing modules. Unfortunately, although all of our individual code worked well, it had problems working together due to the fact that all modules had shared resources (laser, gripper, and camera).

5.0 Robot Bugs

This section contains bugs for all parts of the robot, including the chassis, gripper, camera, and laser. First the chassis: As mentioned earlier, sometimes either the left or right tire would attempt to go over the electrical outlet in the floor and would stall. This became a very large problem as the project went on. Towards the end of the project, though, we found a workable solution to this. We placed shoeboxes on top of the outlets and the outlets would appear as obstructions that could be avoided on the robot map. The shoeboxes were also small enough not to be major obstructions.

Second, the robot's gripper facility had some bugs. The first problem we had was in adjusting the gripper height. The height could only be adjusted when moving up. The problem came in moving the gripper down. The gripper had to be lowered all the way to the bottom of its range and then raised to the desired height. While this was a hardware problem, it was also an API problem as it was controlled by software.

Third, the camera had some issues as well. The camera can only handle approximately 30 frames per second, and the code on the robot runs much faster than that. Also, due to varying lighting conditions in the room, it would not pick up the markers on the chair or it would pick up reflections from the lights on the floor.

Fourth, the laser also had some problems. The laser has a wide range (180 degrees, 90 left of center and 90 right of center), but no objects can be seen behind the robot. This can be a problem because nothing directly behind the robot or to the rear quarters of the robot is seen.

6.0 API bugs

In the development process, our team found many subtle bugs in the API not only for the robot chassis, but also for the subsystems of the laser, the camera, and the gripper.

First, the laser software bugs. The included programs such as the Markov Localization Module and Gradient Path Finding worked very well. Unfortunately, we were unable to use these included modules when it came to the point of pushing a chair. The Localization and Gradient Modules considered the chair in the robot's gripper an obstacle that had to be avoided. Extensive research and development on this point yielded no major success. Due to this problem, we developed our own obstacle avoidance using the getClosestBox function along with our own algorithm. (Please see Coding Algorithm Design Document for details). This function returns an integer although all of the ActivMedia documentation and compilers claim it returns a double. This led to many problems in initial testing and, in the end, led to many erroneous warnings being generated at compile time.

Second, the gripper software has bugs. The gripper software has a feature to allow it to tighten its grip on an object. The more common way of doing this in robotics involves adding more pressure to the paddles of the gripper. Not so with ActivMedia. To do so with the robot's gripper requires a function to be called and the parameter to that function is the number of milliseconds for the gripper to keep pressing inward. Unfortunately, there is not a constant for it to keep pressing.

Third, the camera had a few bugs as well. For the most part, the camera served the project very well. However, there were some problems with the pan and tilt features of the camera. In many cases, it would seem to randomly change these values.

7.0 Things that went well

On the whole, this project seemed to go very well from start to finish as far as development and meeting deadlines are concerned. Some of this can be attributed to our regular meeting schedule.

Second, the fact that the SIUE School of Engineering now has two Pioneer robots helped greatly. During the first semester of senior project, our robot "broke down" and had to be sent back to the manufacturer in Connecticut. Having two robots this semester avoided that calamity.

8.0 Things that went bad

This is a difficult section to write. Most of the problems that were encountered have been covered already. One more can be added though. When we began this project, we had intentions of developing and testing our code in Windows with the robot simulator (both software packages that we all had on our home computers), however, the most recent distribution of Saphira at that time would not compile on a Windows machine. Most of this problem has been fixed now.

So in retrospect, some of our code could have been more refined if we had a Saphira distribution that could compile code to be tested on a Windows Simulator.

9.0 Future Enhancements

No major enhancements to the project are planned right now. However, this project could be used as a starting point for future projects. This project does, however, have room for growth and could be improved. It could be further refined to place all chairs in the room and have a "better" placement of chairs rather than just letting go of the chair when it approaches "close" to a desk. Our approach works fine in respect to how close the chair is, but a more accurate way could be used. Our approach lets go of the chair when the robot is within a certain radius of the point where the chair should be placed. A better approach would be to actually figure out where the chair is, and as long as it is within a smaller radius, let go of the chair. But this approach was figured out too late to implement.

10.0 Dr. White as Instructor, Manager, and Customer

Instructor

The role of instructor played a very minor part in both CS 425 and 499. In CS 425, the instructor led the class, chose teams, and guided the projects. As an instructor, Dr. White was very good. Organization proved to be Dr. White's strong point as an instructor. Because of this, schedules were prepared early throughout the semester. Dr. White made himself very available for our groups to ask questions and discuss our projects both during and outside class. He was more than willing to look over any work before submission in order for corrections to be made prior to grading.

Manager

As manager, Dr. White played a minor role in CS 425, but played an integral role in CS 499. In CS 425, Dr White only made sure that we kept on schedule in our documentation. However, In CS 499 management took on a vital role. Team conflict became an issue that had to be dealt with by management. Dr. White was fair and dealt with the situation promptly. Because of Dr. White's prompt and fair managerial decisions, our group overcame our hardships. After our team conflicts were resolved, we moved on with no hard feelings and did quite well on our project. The most significant downfall as manager was Dr. White's little knowledge of the robot and its API. Due to his lack of knowledge, we had to look to other sources when questions arose.

Customer

As customer, requirements for the project were made and adjusted. Early in our project, we knew we would not be able to complete the original requirement of finding and placing all misplaced chairs. We narrowed the scope of our project to find one chair in the northwest corner of EB2029 and place it in a pre-determined desk. This refinement of scope was the major role of Dr. White as customer. Along with determining project requirements, judging our progress on the project was also a customer task.

All Roles

All roles were hard to distinguish. Because the same person served as instructor, manager, and customer, it is hard to distinguish what Dr. White did as it pertains to each role. This was the biggest problem with a single person holding multiple roles in our project. It was confusing on what role the instructor was performing when changes

were being made to the project's specification. For example, when we changed our specifications to include the Northwest corner of the room as the bare minimum, were we talking to the customer or the manager of the project?

11.0 Personal Critiques of Team Members

Brad – I have little to add to what has already been said. I would reiterate the comments about integration and testing on Windows with a simulator. Also I would suggest the following for SIUE Computer Science Department Curriculum: More UNIX classes, & more real time programming.

Third, there were some positive and some negative aspects to Dr. White's role in the project. Positively, he was flexible with us changing the contents of our release and also answered our questions regarding documents, etc. Negatively, Dr. White is not a robot expert and we could not benefit from Dr. White's extensive graphics background. Secondly, the daily status report became cumbersome.

Finally, the leadership structure works OK. I believe that the titles we had were titles only, as the team members are really peers all programming together. Each member found his own niche, but the team did all work together.

JD – First, I would like to see a mandatory UNIX class added to the curriculum. I think it would be very useful for a job. Everywhere I look seems to use UNIX as their platform. Plus, this senior project would have been so much easier if I had some UNIX experience.

As far as preparing for these robots, I don't think there is much the school could have done. They were brand new, and only one group has used any of them before. We were the first to use the little robot, so not much preparation for that is even possible. Regarding the coordination of everybody's efforts on this project, I think it turned out okay. At first, I think the learning curves hurt some of our team, but it worked out really well in the second and third releases. I think our personnel problems got a little blown out of proportion (with the threat of kicking half our team off the project), when all I wanted was some advice on how to handle the situation. But it worked, nonetheless, and we had no problems after that point. I think the division of labor was pretty even for the most part, but we played to everybody's strengths too. I think there is something I could have done differently to handle the personnel problems at the beginning of the semester, but I could not think of anything else to try. I think that comes with experience as being a team leader. I think that our team gelled into the second release, and everybody did a considerable amount of work to get the project done.

I think the senior project type should be changed in order to get people into the roles they could perform the best. I think that it should have several smaller projects at the beginning of CS 425, and then the instructor could figure out who fits where for one larger project in CS 499.

I think that Dr. White did a nice job of letting the teams handle how the project was developed. He gave a general overview of what he wanted done, and we had to come up with how to do it. He did handle our personnel problem differently than I would have, but it worked so I can't complain too much about it.

I found out that I never want to work with robots again after this project, because they are too temperamental. I am glad that I got to work with the teammates I did, and I learned an awful lot working on this project. I hope we can still communicate after this project is over, because I really enjoyed my team. This has been a real learning experience for me, and I'm sure it will help me once I find a real job.

Peter - I am quite pleased with the outcome of the RAFS project. The CS425 phase proved useful in getting team members acquainted with each other and familiar with each other's skill sets. Despite my satisfaction with the outcome, I do have some issues with several aspects of the CS senior project curriculum. I would have liked to have been able to choose the project I participated in. I had and still do have several ideas for senior projects and feel students should have some input on this matter. I do understand that professors overseeing the project may be uncomfortable with projects outside their realm of expertise, but this should not hinder students from exploring topics in which they are most interested. In some respects, this could prohibit students from perusing topics in which they may be more experienced, losing valuable research and development time.

In many respects, I didn't feel prepared to work on a robotics project. First, I have not had a formal introduction to robotics concepts or programming. This put me at a real disadvantage when trying to apply my software development skills to this project. My performance was also limited by my mathematics skills. I chose the Bachelors of Arts route in the Computer Science Department; therefore my advanced number crunching skills are not very well developed. There were several research topics (localization, for example) which I found difficult to conceptualize with my limited math background. While I found participating on this project challenging and satisfying, I would have felt more prepared working on a less mathematically oriented project.

The CS curriculum at SIUE could use some additions to make it more rounded and robust. As my teammates have suggested, a required course covering basic UNIX and operating systems skills would be very helpful. While I have spent a considerable amount of time learning these concepts myself, I would have liked to receive college credit for these efforts. In addition, the department should consider extending the CS111 curriculum to include advanced computer usage for software development. I have encountered many students who seemed ill prepared for effective use of system accounts and development tool usage. If a period of time was spent introducing students to the actual development environments and tools they will be using throughout their career, we could assume a standard skill set among SIUE students. Concepts that needs covering are FTP, file management, version control, text editors (vi, emacs), command line shell (win32 and UNIX), command line compiling, basic HTML… etc…

Matt - Suggested Curriculum Changes:

- Internships should be credited as an elective or do a specified number of hours to substitute for senior project.
- An introductory course to UNIX should be added.
- Languages, other than C++ based, should be offered more frequently.
    - Java
    - Advanced Visual Basic
- Offer electives during times most core classes not offered (excluding pre-requisite core of course).

Instructor's Role as Manager/Customer in Project:

GOOD

- Dealt with group problems very promptly and efficiently.
- Understood difficulties arise and adjusted the project accordingly.

BAD

- Customer needed to be more actively involved.

I feel our group worked very well together. We had our share of rough times in the beginning, but they were surpassed by our desire to graduate. JD proved to be a good leader. He was organized, enthusiastic, and very involved in all aspects of the project. Peter, being the lead programmer, helped the rest of us pick up areas where we were not skilled. He made himself very available to help the rest of the group and did a lot of outside research to keep everyone up to speed on what topics needed to be addressed. Brad was our Visio wiz and did most of the graphics and charts. He also played a very large role in coding. He had many ideas that worked out nicely and survived into our final release.

12.0 Suggestions from Others

The group gave a demonstration to the CS 490 robotics class and its instructor, Dr. Weinberg. The results and analysis of this data are contained in the **Customer Feedback Document.** We also had some personal feedback from our resident expert, Andrew Lamonica. He suggested some ways we could improve our chair placement algorithm to make it more robust (such as ideas for approaching from any angle).