Team Organization
&
Task Allocation Proposal

Group 9 – Project 2 – 07 March 2003

Troy Humphrey, Joshua Page, Tim Stevens, Jangho Yoon

Original Team Organization:

The organization of Group 9 has changed substantially between Project 1 and Project 2. Originally, the model followed a professional approach where each member was given individual tasks, executed them independently, then the pieces were all brought together for integration during testing. The original four roles decided upon was a *group leader*, which was the person in charge of organization, insuring tasks were being executed, and paperwork/documentation. The second role was that of the *tester*. Working with both the hardware and the software, he would be the person in charge of figuring out how to best integrate the two and successfully test the performance of the robot. The third role was that of the *coder.* Taking the rough software design worked out in the meetings, the *coder* would then implement the pseudo-code into usable code and begin debugging and rudimentary testing. Finally, the last role was that of the *builder*. The *builder* was in charge of the design and construction of the physical characteristics of the robot. The group would meet 4 times total; once for hardware design, once for software design, once for integration, and once for testing.

This original configuration had several appealing qualities. First of all, it is modeled off of the way that professionals and academics in the real world work together and complete projects though they may be separated by vast physical distances. Secondly, it minimized the amount of interaction needed, and also the amount of interference and second guessing. Finally, this configuration allowed for each individual to become completely involved in the four big aspects of this course; building your robot, coding your robot, testing your robot, and writing about your robot.

Unfortunately, this configuration suffered from several fatal flaws. First of all, none of us were professionals. Though this kind of environment sounded very appealing, we lacked the experience and discipline to properly execute. Secondly, since this project had all of the group members in tasks they were weak on, and we were so independent, the tasks were not accomplished as well as they could have. As a result, project 1, while not a complete failure, was not nearly as well executed as it could have been. In the review process that was recently completed, it became apparent that a re-organization was needed, a in looking at the models provided by the other *more successful* groups, a master/apprentice system seemed to be the best fit. This is the system that will be implemented for Project 2.

New Team Organization:

In reviewing the organization of the other, more successful groups, many of them chose a simple master/apprentice type system that was divided into two equal parts, hardware and software. After some discussion during the last couple of weeks, it was decided that this was the kind of organization that could seriously benefit this group. The organization is as follows:

Hardware Team: This team is composed of a senior member and a junior member. The roles are slightly reversed, in that the junior member is the one who will be **building** the robot, while the senior member is there for technical advice, design issues, and experience. This is very similar to a master/apprentice system, where the junior member learns by actually doing. The senior member of the *hardware team* for Project 2 is the *builder* from project 1, while the junior member of the *hardware team* is the *tester* from Project 1.

Software Team: This team is also composed of a senior member and a junior member. The roles again are slightly reversed, with the junior member being the primary **coder**, while the senior member is more of an advisor for technical advice, design issues, and debugging. In this way, the junior member is able to learn much more than he would by merely watching the senior member code away. Most people learn more by doing than by watching, and this is the idea behind this system. The senior member of the *software team* for Project 2 is the *coder* from Project 1, while the junior member of the *software team* is the *group leader* from Project 1.

It is important to note that while the Hardware and Software *implementation* will be performed separately by the two teams, the *design* will be agreed upon by the entire group in group meetings. This allows the maximum amount of knowledge and experience to be brought to bear upon the situation.

Tasks:

The tasks are as follows *(in rough chronological order):*

- Team Organization and Task Allocation Document*
- Timeline with Milestones and Fallback Plan*
- Hardware Design *(group meeting)*
- Chassis Construction
- Software Design *(group meeting)*
- Software Construction
- Testing *(group meeting)*
- Demonstration
- Presentation
- Robot Code & Documentation*
- Robot Design Documentation*
- Team Organization Evaluation*

**Also includes electronic submissions of these documents*

- Final Report*
- Task Allocation Proposal Review *(individual)*
- Time Line, Milestones, Fallback Plan Review *(individual)*
- Peer Reviews *(individual)*
- Team to team review forms *(group meeting)*
- Project 1 presentation review form *(individual)*
- Project 1 demo review forms *(individual)*
- Technical Reviews *(individual)*

Task Allocation:

The tasks will be assigned to group members as follows:

Troy Humphrey: *responsible for coding* - primary coding design, construction, and documentation.
Joshua Page: *hardware assistance* - Team Organization and Task Allocation Document, Timeline with Milestones and Fallback Plan, Robot Design Documentation, Presentation
Tim Stevens: *responsible for hardware construction* - primary hardware design and chassis construction
Jangho Yoon: *coding assistance* - Robot Code & Documentation, Team Organization Evaluation, Final Report, Demonstration
Group Work: Hardware design, Software Design, Testing, Team to team review forms
Individual Responsibilities: All review forms except the Team to team review form.

The reason for this allocation is as follows:

Troy is the *coder*. His main focus should be on writing the code and initial testing of the code to insure it works. He does not need to be bogged down with paperwork. This is the reason that coding is his only responsibility.

Tim is the hardware *builder*. His main focus should be on coming up with the most efficient, simple, and effective design possible. He does not need to be bogged down with excessive paperwork and reports.

Josh has the two initial reports (team organization & the milestones) and the presentation because these reports are on a rotating basis, everyone will do them. The Robot Design Document is assigned to Josh because he is the *assistant* member of the hardware team and has the time and experience for this document while Tim is concerned with the actual robot itself.

Jangho has the Team Organization Evaluation, the Final Report and the Demonstration because these are assigned on a rotating basis and everyone will get a turn at them. Robot Code & Documentation is Jangho's job because as the *assistant* member of the software team, he has the time and experience to complete this document while Troy is concerned with finishing the software.