

Weekly Meeting, April 6th-13th, 2011

Multiple Mobile Social Robots

I. THOUGHTS ON AUTONOMY VS MANUAL TELEOPERATION

What is the relationship between the operator the GUI and the rest of the system (collection of robots).

How is the GUI mediating between both sides? It helps to “balance” or to “compensate” the lack of automation of the robot(s) and helps the performance of the operator.

More autonomy is required to be able to manipulate more robots. However, this increased level of automation per individual robot only allows to create better groups of robots that need to be guided appropriately. Thus, sending the problem of automation one layer higher, e.g. from individual robots to groups of robots.

It is then important to also increase the level of automation to this higher layer. This automation regarding this later must include methods that allow to convey the existence of errors to the operator.

The operator can then have manual monitoring or operation of the collection of robots. An initial advantage to this method is that the operator can decide the frequency at which he/she can check the status/activities of groups of robots or individual robots. One downside is the possible routine the operator may fall in: keep checking the same groups/robots and neglecting others that may require attention for the cases on which the system cannot detect a “system” (hardware/software) error.

Another method would be automatic monitoring. This methodology would rely on the system to indicate the operator which group/robot should be monitored/checked/paid attention to. This method has the advantage of going through all entities within a given frequency and under certain “rules” or constraints. The problem with this method is that the operator may become frustrated by not having control on the decision to monitor a given group and how often to do so. This will fall into a similar problem of the “operator-out-of-the-loop”.

Similar to the “sliding autonomy” principle, the operator could choose between these two options/methods. This is the “sliding autonomy switching”, which may also be called the “collaborative operator-system” method. This method allows the operator to take decisions on which method to use and when.

II. TYPES OF OPERATION

A. Direct Operation

The operator focuses on a single robot and controls it, like we do with the current version of MRC.

B. “Shallow supervision”

The operator checks all robots regularly by glancing at a summarized display. This might be in a grid-type interface, or represented on a map. The point is that by checking very quickly the operator can decide whether or not the robot needs more attention. In current MRC, this is like looking at the robot status panels to decide what to do.

C. “Deep supervision”

The operator needs to focus on a single robot to examine its situation, e.g. switching to the robot to listen to its conversation or look through its video display. In current the MRC, this is like switching to a robot to see if anything is needed.

III. SWITCHING BETWEEN ROBOTS

There may be 3 different ways of switching among robots.

A. Detection-based system

The overall system can automatically identify when intervention is necessary and alerts the operator through the GUI.

B. Risk-based system

The overall system cannot detect problems but can identify when supervision is necessary, e.g. the risk of error is high, so the operator should monitor this robot more frequently. Either of these can be on/off detections or gradual levels of risk.

C. Auto-switching vs Manual

The operator can select between an automatic and a manual switch. The automatic switch can be configured to be any of the two possible systems proposed above and the manual switch would be based on the operator’s preference of which robot to monitor and when.

IV. GUI DESIGN CONSIDERATIONS

Fig. 1 and Fig. 2. present two possible GUI designs for the operator’s view. In both cases, circles represent robots and their different color represent to which group they belong to. In both cases, stars represent a person. Robots and persons would have their ID and current status represented in the map. Other simple status may be represented in later designs.

A. Grouping of robots in interface

- 1) Grouping might allow more robots to be managed, but also adds clicks and complexity to the interface, as seen on Fig. 1. This might be good for large number of robots (≥ 20) and robots may be grouped based on task not geographical location.
- 2) A GUI design that presents all robots present at one time in a given environment can be useful for a small collection of robots (≤ 20). Less clicks would be necessary to reach a single robot but more memory would be required by the operator to remember each robot's location.
- 3) Hiding groups could be dangerous because it takes more effort to get to the robots that need help.
- 4) Grouping can be based on an automatic switching which may consider the level of risk of failure of each robot.
- 5) Grouping can be based on manual recognition of a problem provided by the system as an alarm, or by allowing the operator to simply choose a personal monitoring routine.

B. Display of robots

- 1) Map makes sense if geographical information is important (e.g. if knowing what shop the robot is in will help the operator's situation awareness, or if visualization how crowded is around the robot will help).
- 2) Grid has the advantage that the operator can learn where the different robots are on the display, and thus needs less context to be explained for each robot.
- 3) Dynamic list has the disadvantage that the operator needs to gain SA about each robot as soon as it appears, but has the advantage that many robots can be filtered out, e.g. *geq* 20-robot case.
- 4) A combination of 1 and 2 is shown in Fig. 1 and Fig. 2.

C. Types of unexpected situations

- 1) Detectable situations - the system can alert the operator. If there are many false alarms this is annoying, and if there are many missed alarms this is ineffective.
- 2) Undetectable situations - operator must actively monitor the robots

- d) Experimenter view: other info like duration of interaction, list of recently completed tasks and durations, number of sales that day, etc will be presented in this type of view

V. OTHER ISSUES

- 1) How do we simulate errors?
- 2) What kinds of information are needed by the operator?
 - a) spatial info like how many people are in the environment or location of the robot can be shown at-a-glance on a map.
 - b) contextual info like current topic of conversation, conversation partner's expressions, etc. (can also be shown at the map, as shown in Fig. 1 and Fig. 2.)
 - c) operator may want to zoom into a robot specifically

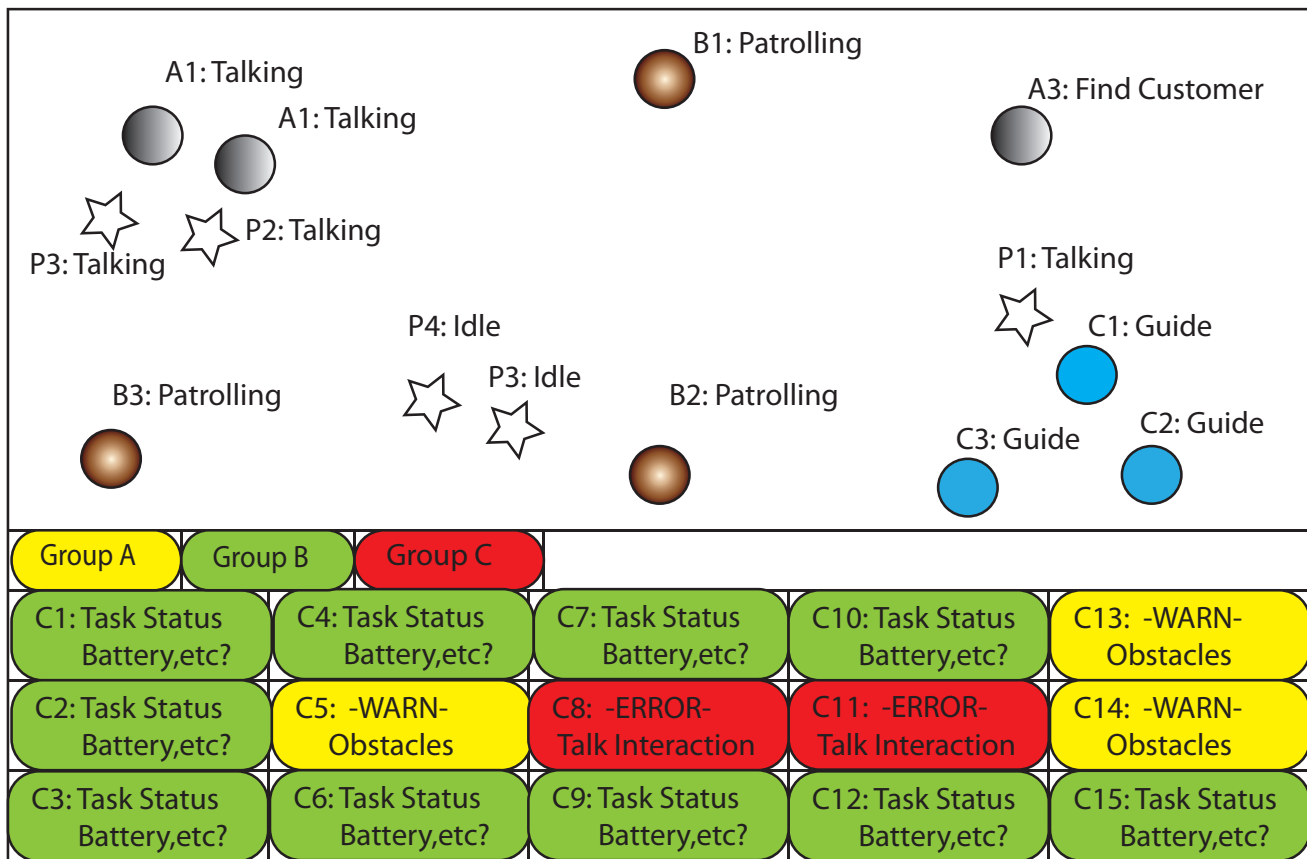


Fig. 1. Grouped operator view

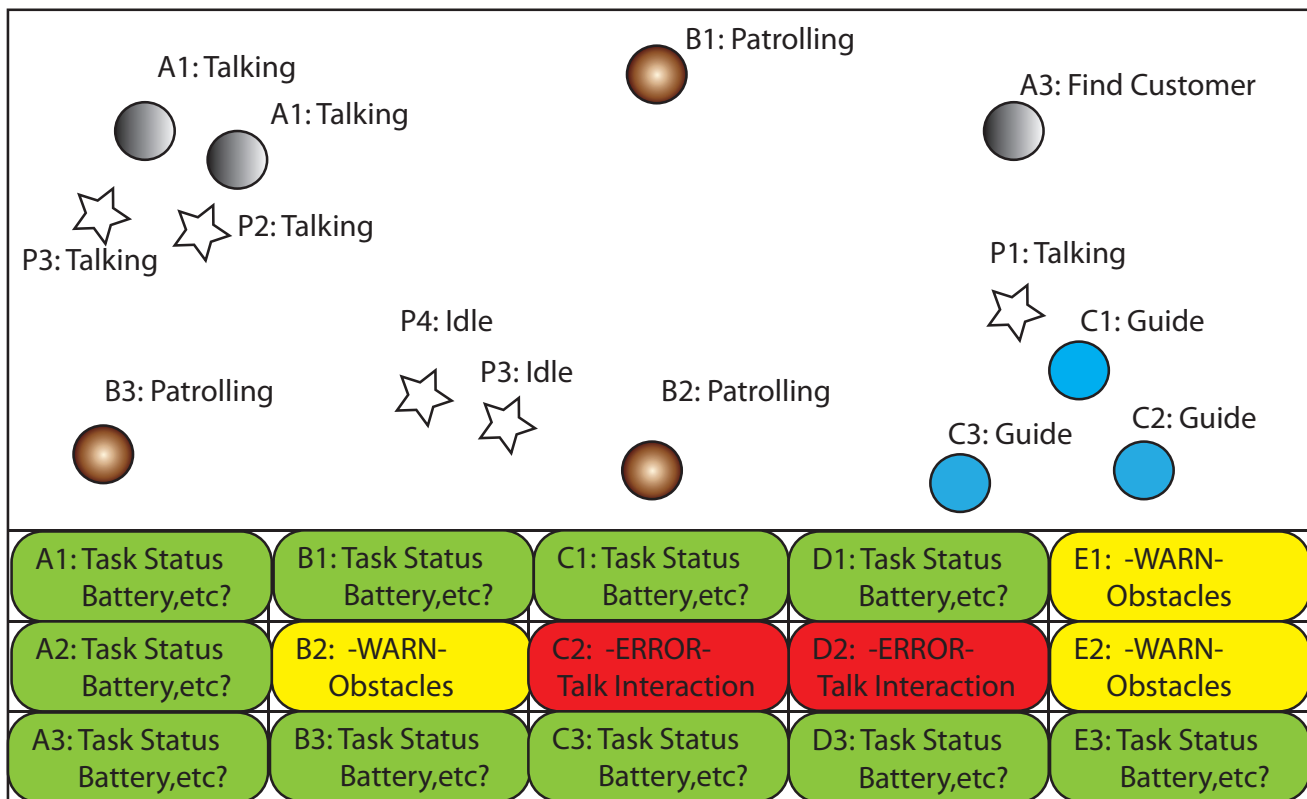


Fig. 2. Non-grouped operator view