Writing Code in the Field: Implications for Robot Software Development

William D. Smart

Department of Computer Science and Engineering
Washington University in St. Louis, United States wds@cse.wustl.edu

1 Introduction

Running robots on a deployment in the “real world” is very different from running them in a laboratory setting. Environmental conditions, such as lighting, are often completely beyond your control. There is often a limited amount of time on-site to test and tune the robot system, and deadlines are generally very inflexible. Finally, the public is notoriously unforgiving of failure. In this chapter, we identify some key design issues that can affect real world robot deployments. We discuss these issues and propose some solutions, based on our own experiences, designed to ease the pain of sending a robot out into the real world.

1.1 The Deployment Scenario

In this chapter we will consider a particular type of real world deployment, characterized by six main features:

Environment: The deployment takes place in an environment where we do not have complete control over the environmental conditions, such as the lighting level. We also assume that we cannot completely replicate the environmental conditions in the laboratory before arriving at the deployment.

We cannot alter the environment in any substantial way for the deployment. This includes removing furniture and obstacles that the robot has difficulty with, adding brightly-colored navigation aids, and forbidding the public from areas of the environment.

Schedule: The robot is not the most important thing in the environment, and existing schedules cannot be altered to accommodate our deployment. For instance, the opening time of a building cannot be delayed if the robot is not ready on time.
We often have a limited amount of time on-site to make sure that the robot behaves as expected, and to tune it to the specifics of the environment. This limitation generally comes from the policies of the deployment site which may, for example, restrict after-hours access to the venue.

**Task:** The robot is performing some autonomous task during the deployment, and is not just a static exhibit. It is expected to interact with the environment, and possibly the general public, in an “intelligent” and purposeful manner.

**Duty Cycle:** The robot must perform its task for a set period of time, often several hours, without intervention. All maintenance and repairs must happen outside of this time period (after the building has closed, for example).

**The Public:** The public will be observing the robot as it performs its task, and might even be interacting with it directly. We cannot assume that the public knows anything about robots, computer science, algorithms, or the like. Although human “interpreters” can be used to explain the robot’s behavior, they should be used sparingly.

**Reputation:** The public has a low tolerance of failure, and will associate poor performance with our lab, institution, and students. Failures in the laboratory, in front of a knowledgeable audience, can be (somewhat) explained away at a technical level. No such easy escape exists in the real world. If the robot seems broken, or is acting erratically, the public will lose interest and move on. More importantly, they will remember that the robot didn’t work as expected, and will tell their friends.

As a concrete example, consider the deployment of a mobile robot in a science museum. The museum has fixed hours, which we have no control over. The robot must be ready when the doors open at 9am, and we cannot perform any obvious maintenance until the museum closes at 5pm. The robot is likely to be one of a large number of exhibits. If someone comes by while the robot is not working, they will simply move on to the next exhibit. This means that it is important to keep the system working for the entire duty cycle. This is especially true if it has been advertised, since people might have made a special trip to see the robot in operation.

It is unlikely that we can replicate the conditions of a typical science museum in our lab, although we might be able to approximate some of them. This means that we will need to spend time tuning our system to operate in the museum. The amount of time we can spend doing this will be limited by museum policies. It also must happen at an inconvenient time, in the evenings and very early mornings.

1.2 Typical Deployment Problems

For any non-trivial deployment of the type described above, we will have to adapt our systems on-site. This adaptation will involve the tuning of software