Final Report Context Project S&R

Group 2

June 20, 2014

1 Evaluation & analysis functional modules and product

1.1 Functional modules

User tests The functional modules, those being the environment store, the bot store, the scenario editor and the EPartner store are analyzed and evaluated here. A small user test with a researcher has actually been executed with the map editor. This gave the following results: it is hard for a user that has not developed the system to actually understand how to add blocks to a certain room. There is no hint that the white box at the bottom side of the room has to be clicked to add blocks to the room. A solution would be to add a small label or tooltip explaining what the white box at the bottom of the room or map is for. The task to create your own map and save it, however, went very well. As the researcher was very experienced with the BW4T environment, she was impressed by what is possible now using the map maker. There was an issue with there not always being a charging zone in the map, but this is fixed now in a small fix. The other issues are discussed in the Outlook part of the report.

Demo feedback There was some feedback on the environment store during demo's where the customer could use the software. The original warnings for saving a map that wasn't solvable were simple and did not evaluate on what was the unsolvable part of the map. This was fixed later and the warnings are far more descriptive. They still don't show where in the map the problem is, because that was quite hard to do.

1.2 Product

As the functional modules are mainly evaluated through usability, the product in general is going to be evaluated through structure and architecture of the components. Also, our part focuses on the new feature aspects of the software, so only the structure and architecture of the code forming that part of the system will be discussed here. The analysis is as follows:

Decorator pattern Unlike most of the other groups, we started with pretty much no starting code at all. There was no previous support for robots with handicaps, let alone a bot store where one could create a robot with certain handicaps. The handicap functionality makes use of a decorator pattern to be able to put multiple handicaps on a single robot. At first, the decorator pattern was not correct, so the code was rewritten to be able to function as a decorator pattern. This required some minor changes to the standard robot code. The decorator pattern is correctly implemented now, as stated by the project supervisors, and also allows for a single robot to have multiple handicaps active at the same time. The code used is high quality code now.

MVC pattern