Programming Robots for Activities of Everyday Life

Name: Swaib Dragule

REG. NO.: 2025/HD05/1394U, STUDENT NO. 2500701394

Abstract

Robots are increasingly being used across various domains to assist humans with repetitive, hazardous, and everyday tasks in environments such as homes, hotels, airports, and museums. However, specifying robotic missions remains a challenge, especially for end-users who lack programming expertise.

Traditionally, robotic missions have been programmed using low-level text-based languages, which are error-prone and difficult for non-programmers to use. To fully harness the potential of robots in everyday tasks, mission specification must become more accessible, unambiguous, and flexible for diverse end-users.

This research aims to empirically understand the current state-of-the-art in languages and tools used for specifying robot missions by end-users. It focuses on evaluating domain-specific languages (DSLs) designed to simplify robot programming, particularly behavior tree and state machine paradigms.

Through a detailed analysis of 30 mission specification environments, 133 features were identified and mapped into a feature model, providing insights into the tools' capabilities and their use in educational and professional settings. During the empirical study, DSLs for robot mission specification were analyzed through published literature, their websites, user manuals, sample missions, and using the languages to specify missions for supported robots.

From 30 environments, 133 features were identified. The study further involved an empirical evaluation of two prominent DSLs: Groot, representing behavior trees, and FlexBE, for state machines. A controlled experiment measured their effectiveness and efficiency in terms of abstraction level, code comprehension, mission correctness, and overall usability. The results showed that while both paradigms ranked above average in comprehension, FlexBE performed better in usability. However, improvements in user interfaces, tutorials, and basic training for end-users were identified as essential for both tools.

Our results show that most end-user environments are used in education to teach novice programmers and STEM subjects, often built using Blockly and Scratch libraries, although behavior trees and state machines are gaining dominance. However, these tools focus on robotic and programming abstractions rather than end-user-specific concepts, highlighting a gap that needs improvement. Future work should develop reusable libraries centered on end-user concepts and adapt these environments for novice programmers in low-resource settings like Uganda. While DSLs offer more abstraction than text-based languages, further refinement is needed to improve their usability and accessibility for non-programmers. This study provides insights for enhancing user-friendly DSLs, focusing on better educational resources and user interfaces to support end-user robot programming in real-world contexts.

Keywords: Mobile robots, Mission specifications, Robot missions, Robotic IDEs, Domain specific languages, End-user programming, Behavior trees, State machines, empirical evaluation, Control experiment.