

# VISUAL ROBOT PROGRAMMING

*Laura Diao*

*Advisor: Professor Franklyn Turbak*

*Computer Science Department*

*Wellesley College*

*Wellesley, MA, 02181*

*(617) 283-1747*

*ldiao@wellesley.edu*

Robot behaviors can be difficult to express in traditional text-based procedural programming languages. For example, students in Wellesley College's robot course found that even simple behaviors could be tricky to express in Handy Logo, a version of the Logo programming language extended with primitives for robot control. There are two main difficulties:

- (1) Handy Logo is a mostly sequential, procedural language, in which a sequence of commands are executed in a step-by-step manner. But in many cases, a more natural approach is to express robot behaviors as rules (if `<condition>`, then `<action>`).
- (2) Handy Logo has no built-in debugging interface. The only way to monitor data flow and state transitions during a program execution is by putting numerous `<print>` statements within the program.

Our goal is to address these problems and simplify the specification of robot behaviors. The design has been guided by our study of programs written by students from the robot course. We explored ways of expressing these programs in several other programming models: state-transition diagrams, data-flow diagrams, and rule-based models.

Based on our studies, the rule-based paradigm seems have important advantages over the other models:

- (1) Intuitively, when a robot performs its tasks, it follows some set of "rules" giving by the programmer.
- (2) Rules are easier to represent visually than state-transition diagrams and data-flow diagrams, which become very complicated and confusing in the presence of numerous nodes and edges.
- (3) Rules are a natural way to capture the inherent concurrency of many robot controllers.

Based on our experience, students have an easier time understanding program execution when state and event information is explicitly shown. The purpose of our visual approach for the language design is to make this information visible in the robot programming language development and execution environment.

We have designed a preliminary visual representation for this rule-based paradigm. This design offers a number of plausible advantages over Handy Logo and provides a framework for future research. We are in the process of implementing this robot programming environment in Java and hope to have a prototype version working by summer of '97.