FLOGO: A ROBOTICS PROGRAMMING LANGUAGE FOR LEARNERS

Chris Hancock, MIT Media Laboratory
ch@media.mit.edu

Project Goals

Robotics is a relatively new domain for programming by children, but a domain that is quickly gaining interest among educators, researchers, and especially children. Many children love to build robots and create behaviors for them. In doing so, they can learn on many levels – about science, engineering, design, and artistic expression, as well as programming (Druin, 2000; Resnick, Berg, & Eisenberg, 2000). In this rich mixture, however, programming may not always figure prominently. Children may focus their efforts on imaginative or elaborate physical forms, or on mechanical engineering problems. Robot behaviors may be pre-packaged, or provided by an expert mentor; or, the child may construct extremely simple behaviors. Programming a robot to behave in interesting or intelligent ways – in ways that have some apparent autonomy – can be surprisingly hard. Like generations of AI researchers before them, children often find that goals must be repeatedly scaled back as the complexity of seemingly simple behaviors is revealed.

The Flogo project attempts to empower children to construct more sophisticated behaviors than they have so far been able to do. In doing so, we hope also to contribute to progress in more general problems in novice programming. The long-term target is a "low threshold, high ceiling" environment making optimal use of visual and textual representations to provide a smooth ramp for progress from novice to advanced levels of robotics programming. The project works within the context of an established program of research and development in robotics for children at the MIT media Laboratory (e.g. Martin, Mikhak, et al., 2000; Resnick, Berg, & Eisenberg, 2000).

Language Characteristics

One key characteristic of Flogo programming is "liveness" (Tanimoto, 1990). In the typical setup, children use Flogo to drive robots through a constant communications link between a desktop computer and the robot's microcontroller, so that the operation of the program can be seen and adjusted in real time. Current sensor data, computed values, and other aspects of program state are constantly visible. The user can intervene directly in the running program, or change the program structure itself as it runs.

Within a single integrated environment, Flogo attempts to combine the advantages of two programming paradigms. The first of these is a visual dataflow model (the "flow" part of Flogo) using a hierarchical variant of the wiring diagrams found in such languages as LabView and MindRover. The second is a textual, procedural paradigm (the "Logo" part). For different parts of the same programming task, one paradigm or the other may be more expressive and intuitive (see e.g. Green&Petre, 1992, Whitley, 2000). We are working to allow users to combine these paradigms as fluidly as possible.

One additional property is essential to the Flogo vision. Flogo is an extensible language, supporting the construction of re-usable, recombinable, modifiable, higher-level tools, complete with their own user interfaces. In this respect Flogo follows in the footsteps of Boxer and the "open toolsets" philosophy (diSessa, 1997).

Preliminary Findings

The language has been tested with 10- and 11-year-olds with varied previous experience of Logo Microworlds programming. Eight children participated in a two-day "robotic arts" workshop. Eight more, plus one of the first group, participated in a four-day (20 hour) workshop which was preceded and followed by informal clinical sessions in which children, in pairs, were introduced to new language features and took on a variety of programming and problem-solving challenges.

These workshops exemplified our vision of robotics as a medium for personal and artistic expression of many kinds. In addition to the more conventional humanoids and vehicles, children built such things as a model fairground ride, a "gardener's helper" which spreads seeds by spinning, and assorted kinetic sculptures with interesting movement and light patterns. In the second workshop, children also collaborated to construct mechanized versions of scenes from a storybook; these were used to make a video of the story.

This exploratory research has helped to surface three themes of interest:

- Conceptual issues in the temporal structure of data. Some information is represented as discrete events, other as quantities that vary over time. Novices ancountered some interesting difficulties in applying and coordinating these two kinds of information.
- The language's real-time visual feedback allows children to apply a kind of hand-eye learning, not unlike video gameplay, in tandem with more theory-based thinking.
- Children made surprisingly effective use of language components that had not been explained to them. Certain features of the language appear to favor programming by tinkering. Dynamic feedback, spatial organization, and low fragility all help to increase the rewards and reduce the risks of experimentation. On reflection, we consider "tinkering-friendliness" a very desirable language property and are working to better understand and cultivate it.

These issues are discussed in a paper which was presented at the meeting of the American Educational Research Association in April (Hancock, 2001). The project is now back in software development mode (focusing on the textual-procedural side of the language), with further testing with children planned for the coming fall.

Focus of Presentation

Drawing on examples of children's work with Flogo, I will discuss some advantages and challenges of supporting programming-by-tinkering. Time permitting, I may also touch on some design issues in integrating visual dataflow and procedural paradigms.

Further Information

More information, including illustrations, may be found in (Hancock, 2001), which is accessible via the Flogo home page, at: http://llk.media.mit.edu/projects/flogo

References

diSessa, A. A. (1997). Open toolsets: New ends and new means in learning mathematics and science with computers. Paper presented at the 21st Conference of the International Group for the Psychology of Mathematics Education, Lahti, Finland.

Druin, A., Ed. (2000). *Robots for kids: Exploring new technologies for learning experiences*. San Francisco: Morgan Kaufman / Academic Press.

Green, T. R. G., & Petre, M. (1992). When visual programs are harder to read than textual programs. Proc. Sixth European Conference on Cognitive Ergonomics (ECCE 6), pp. 167-180.

Hancock, C. (2001). Children's Understanding of Process in the Construction of Robot Behaviors. Paper presented at the symposium "Varieties of Programming Experience," AERA 2001, Seattle.

Martin, F., Mikhak, B., Resnick, M., Silverman, B., & Berg, R. (2000). To Mindstorms and beyond: Evolution of a construction kit for magical machines. In A. Druin (Ed.) *Robots for kids: Exploring new technologies for learning experiences*. San Francisco: Morgan Kaufman / Academic Press.

Resnick, M., Berg, R., & Eisenberg, M. (2000). Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *Journal of the Learning Sciences*, 9(1) pp.7-30.

Tanimoto, S. (1990). VIVA: A Visual Language for Image Processing. *Journal of Visual Languages and Computing,* 2(2), 127-139.

Whitley, K.N. (2000). Empirical research of visual programming languages: An experiment testing the comprehensibility of LabView. Unpublished doctoral dissertation. Computer Science dept., Vanderbilt University.