Solvelets – Tutors for Learning Programing from Start to Finish

Amruth N. Kumar

Ramapo College of New Jersey, Mahwah, NJ 07430, USA

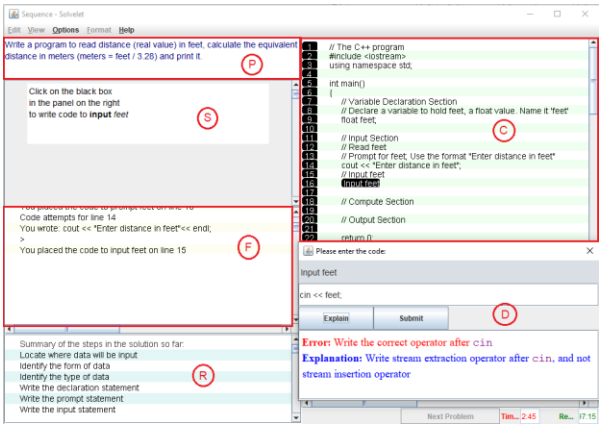
amruth@ramapo.edu

Topic and domain: Solvelets are tutors for learning to program in C++ [1]. They step the learner through all three stages of solving a problem: algorithm development, program design and coding. The learner starts with the statement of a problem and ends with a complete and correct program for the problem statement. So, solvelets promote computational thinking.

Pedagogical activity and feedback: Solvelets elicit the algorithm using selection operation, e.g., selecting the type of control statement to compute output data from input data. They have the learner design the program using selection and location operations, e.g., locate where the output data must be computed from input data in the code canvas. They have the learner use coding operation to write the code for the designed algorithm and program. The tutors interleave algorithm development, program design and coding in an epistemic manner and provide scaffolding through all three stages. They provide immediate feedback at abstract, concrete and bottom-out levels at each step, ensuring that the learner is never trapped at an impasse. During coding, they enforce good programming practices such as writing control statements shell-first and using data types consistently. The outer loop of the tutors steps the students through successive problems and is adaptive.

Instrumentation and evaluation: Each solvelet reports every selection, location and coding action of the learner, along with the feedback provided to the learner. Analysis software codes these actions and counts them to determine whether the learner improves with practice. Solvelets have been evaluated since fall 2019. The results show that practicing with solvelets helped learners solve subsequent problems with fewer erroneous actions and in less time.

Implementation and novel technologies: Model-based reasoning is used for domain modeling, which makes it easy to add new problems and new types of problems. The student model is built as an overlay of the knowledge model and is used for adaptation. The tutor model uses schema to declaratively represent the pedagogy, which can be changed or enhanced with minimal effort. It uses bug libraries to diagnose the learner's code.



Related work: Several programming tutors have been developed that cover all three stages of solving a problem: algorithm development, program design and coding, including LISP Tutor [2], PROUST [3], BRIDGE [4], GPCEditor [5], [6,7] and J-Latte [8]. Unlike the previous tutors, solvelets scaffold rather than diagnose the learner's algorithm and program design. They use comments instead of graphic representation for algorithms. They provide immediate feedback. They are the first to address C++.

Plans for availability: The tutors are currently available for C++, for sequence, selection and post-test loops at the site solvelets.org. They are free for educational use.

References

1. Kumar, A.N. An Epistemic Model-Based Tutor for Imperative Programming. Proc. AI-ED 2021. *To be published.*
2. Reiser, B., Anderson, J., & Farrell, R. (1985). Dynamic Student Modelling in an Intelligent Tutor for Lisp Programming. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 8-14.
3. Johnson, W. L. (1986). Intention-Based Diagnosis of Errors in Novice Programs. Morgan Kaufman, Palo Alto, CA
4. Bonar, J. and Cunningham, R. (1988) BRIDGE: Tutoring the programming process, in Intelligent tutoring systems: Lessons learned. J. Psotka, L. Massey, S. Mutter (Eds.), Lawrence Erlbaum Associates, Hillsdale, NJ.

5. Guzdial, M., Hohmann, L., Konneman, M., Walton, C., & Soloway, E. (1998). Supporting programming and learning-to-program with an integrated CAD and scaffolding workbench. Interactive Learning Environments, 6 (1&2), 143{179.
6. Wei Jin. 2008. Pre-programming analysis tutors help students learn basic programming concepts. In Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08). Association for Computing Machinery, New York, NY, USA, 276–280.
7. Jin W., Corbett A., Lloyd W., Baumstark L., Rolka C. (2014) Evaluation of Guided-Planning and Assisted-Coding with Task Relevant Dynamic Hinting. In: Trausan-Matu S., Boyer K.E., Crosby M., Panourgia K. (eds) Intelligent Tutoring Systems. ITS 2014. Lecture Notes in Computer Science, Vol. 8474. Springer
8. Jay Holland, Antonija Mitrovic, and Brent Martin. 2009. J-LATTE: A constraint-based tutor for Java. In Proceedings of the Conference on Computers in Education. 142–146.