

World Conference on Educational Sciences 2009

Simulink stateflow for algorithm learning

Fecir Duran^{a*}, İsmail Atacak^a, Ömer Faruk Bay^a^a*Faculty of Technical Education, Department of Electronics and Computer Education, Gazi Üniversitesi University, 06500, Ankara, Türkiye*

Received October 10, 2008; revised December 16, 2008; accepted January 04, 2009

Abstract

Systematic solution of a problem is possible with sequential, unambiguous and realizable finite process step set. This set is called “algorithm” and it is one of the most important part of the programming languages. In this study, it is represented how to learn algorithm by using Simulink Stateflow. In flow diagram-based algorithm learning, it is not possible for students to check the solution of a given problem and related parameters without coding the algorithm for the problem. Whereas traditional algorithm learning methods, Simulink Stateflow-based algorithm learning method enables student to check the validity of processing steps without coding. Hence, Simulink Stateflow-based algorithm learning is much more effective than the traditional algorithm learning.

© 2009 Gazi University, Faculty of Technical Education, Department of Electronics and Computer Education

Keywords: Algorithm Learning; flow diagram; simulink stateflow;

1. Introduction

Finding a proper method for a computer application is a result of an individual solution and a general solving methodology. Developing a solution model could be represented by statement or flow diagrams. Independent from the representation way, solution, process steps and orders must be known accurately and concretely without any ambiguity. Solution management, as called as algorithm, is the first and most important milestone in software programming and its learning phase.

Owing to easy-to-use and applicability for solution, designed algorithms are modeled with geometric shape-based flow diagrams. Structure with the diagrams, which are used for the design of the algorithm, enables users to express inputs and outputs clearly for a solution. However, it is needed to code or consider all possible cases to validate if the realized design. Hereby, Simulink Stateflow enable users to check the system algorithm by using diagrams easily

* Fecir Duran. Tel.: +90 312 202-85-86; fax: +90 312 202-00-59.

E-mail address: fduran@gazi.edu.tr.

and quickly. Furthermore, it is possible to run the model built in Stateflow and check/ display the variables and dynamic responses of the model. Hence, it is possible even for students without any programming knowledge to improve problem solving skills.

1. Algorithm

Software is initiated with programming and learning programming languages. Programming is a method applied on computer to solve problems. Though there may be pretty simple problem definitions in numerical processing, problem definitions in industrial applications are achieved with methodical solutions of process and query of large databases.

A kind of a problem could be problem solving, proof problem or developing a methodology for solution. Problems to be handled in computer are usually in the type of developing a methodology. Solution will be sought within computational resources of the computer. Therefore, this methodology requires different steps and order than the traditional ones. Developing a methodology in accordance with programming is the product after the steps of individual approaches on the way of giving a generic solution. Transferring from traditional solution medium to programming, it is needed to have a solution that considers hardware and software capabilities of the computer (Holloway, 2004). It is known that computer has different processing structure than the traditional problem solving such as speed, connection, analog/ digital processing, memory, storage, data retrieve, update and refreshing. It is also possible to mention about some remarkable differences such as input data (log file, data tree, database), data processing (sequential, random, refreshing), storage (sequential, random, related), output formatting (permanent, temporary), user-interaction type (remote, interface). Finding proper solving methodology to the computer processing has been the main approach of programming (Gürkan, 2003). This approach is the generic form of the solving method to solve a quadratic problem easily by Harzemli, a mathematician with Turkish roots (Harel, 2004).

An algorithm is the set of sequential, unambiguous, process able and finite steps to solve a type of problem. It is possible to find very close definitions of the algorithm-based problem solving in different sources. Since it is common, algorithm has significant importance in every programming language (Knuth 1974).

The important feature of an algorithm for both problem and computer is containing the solving methodology for both. Therefore, an algorithm has following properties: It gives results with respect to the inputs or finite number of processing steps and processing starts with a specific step nearby finishing with the step that conducts the result or no feasible solution for the problem.

Starting from late 1800s, algorithm and developing algorithm have evolved and has spread itself from math into many other fields ranging from interdisciplinary sciences to computer. It is not possible to draw a boundary for developing algorithm and algorithmic approaches as an evolving phenomenon. With a very basic definition, it is possible to divide the fields of the algorithm phenomenon into two subfields: numerical and analytical (Harel, 2004).

Mathematicians have proven the feasibility of the solution for a given problem by defining the numerical processing orders, processes, treatments, conditions and systematically defined strict structures for a problem type.

An algorithm does define strictly, concretely, unambiguously and exactly what and how to process further within a solution. These main features make the algorithm a must for numerical processing and analytical solution.

Developing an algorithm has brought a new approach to the traditional problem solving. One of these advantages is the generic solution, and covering all possible cases with concreteness. This feature makes it valid not only in numerical processing but also in analytical solution.

Furthermore, algorithmic system has appeared with coverage of linear algebra, which is used to find the system unknowns with some set of actions. Linear algebra consists of sets objective and constraints on them. Further more, algorithm consists of an input set and processes through which the output should be yielded.

Applicability, process ability and simplicity, which are brought by algorithm into problem solving, make it different from other methods. Design of an algorithm with ease-made process ability and applicability requires much more time high amount of intelligence, knowledge and personal skill. Nonetheless, a bullet-proof algorithm could even be used by a novice user without concrete amount of knowledge.

From another viewpoint, algorithm has also clearly available mathematical consistency and concreteness. Feasibility/ Infeasibility of a problem or a problem type depend on whether or not an algorithm could be developed.

Hence, there is an important relationship between feasibility and the algorithm. Those all properties are rendering the algorithm indispensable.

1.1. Flow Diagrams used for algorithm development

Flow diagrams are geometrical shapes used for designing phase of algorithm development before coding phase. These flow diagrams and their explanations are seen in Figure 1. Composing diagram-based structure, it is possible to have a solution by stating the inputs and outputs clearly. (Vatansever, 2005)

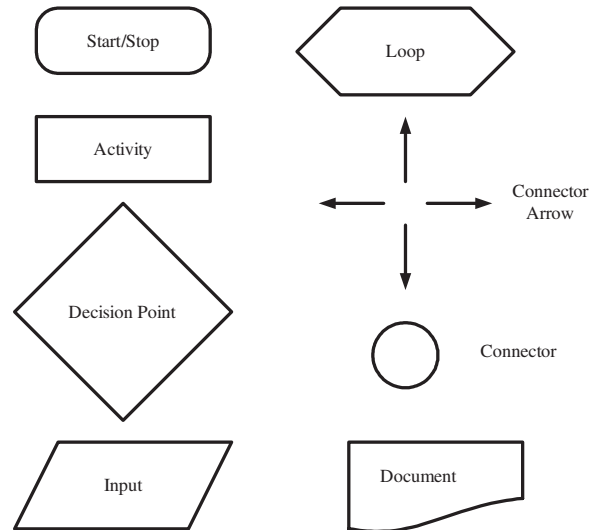


Figure 1. Flow diagrams used in Algorithm Development

Considering a washing machine, the turn on/ off button, the loader cover lock, the water feeding valve, the water pump motor for drainage, the revolving motor are some part of the system. An illustration algorithm for this system represents washing process steps in terms of durations. Figure 2 shows this structure if this definition is considered for the algorithm design.

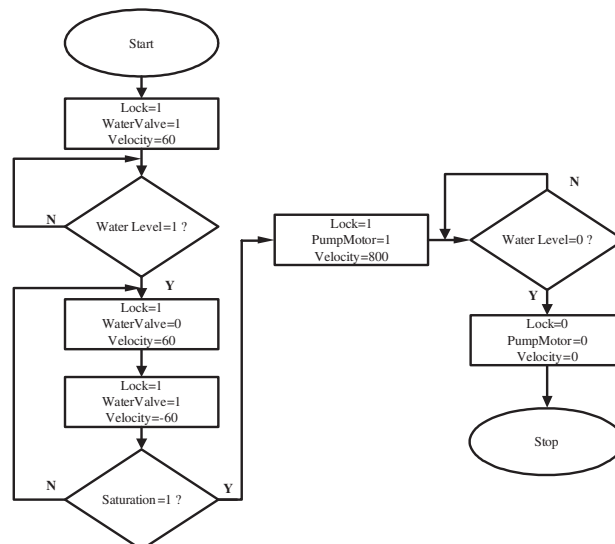


Figure 2. Wash Machines Algorithm Design with Flow Diagrams

Functionality, simplicity, applicability brought by algorithm for problem solving makes it have also accuracy and reproducibility. This feature brings simple mechanical check on the sequential statement of the algorithm. Mechanical behavior is defined as processing without usage of any intelligence and logical resources. As long as the mechanical behavior of the algorithm has been improved, machines and algorithms would have closer similarity. This approach brings new gains for problem solving in terms of tool usage. Three main aspects in problem solving are “logic”, “solution methodology” and “language/ syntax”.

Computer algorithm has difference in terms of process durations definition, syntax that computer can perceive and process, and results achieved with an algorithm run. Therefore, it is not possible to use a syntax without algorithm or an algorithm without a syntax.

Simulink Stateflow–Based Algorithm Development Steps

Simulink Stateflow is a commercial software tool that covers logical modules for solution of every type of problem within graphical user interface platform (GUI). It is possible to design algorithm for every system by using flow diagrams easily and rapidly. By using external switches, it is also possible to observe the behavior of the flow diagrams with respect to changing inputs.

1.2. System Preparation for Algorithm Development in Simulink

Simulink is a model based simulation environment of Mathworks. Role of the simulation in engineering fields have been remarked day by day. It is possible to benefit from computer simulations in system design, and tests of designs if possible. This reduces the need for prototyping and so the risk and costs of possible prototyping.

Models inside Simulink enable systems to be designed easily within the software. It is possible to connect models with simple drag-drop actions. After connecting all sub models, it is possible to get the washing machine algorithm design in Simulink Model as seen in Figure 3.

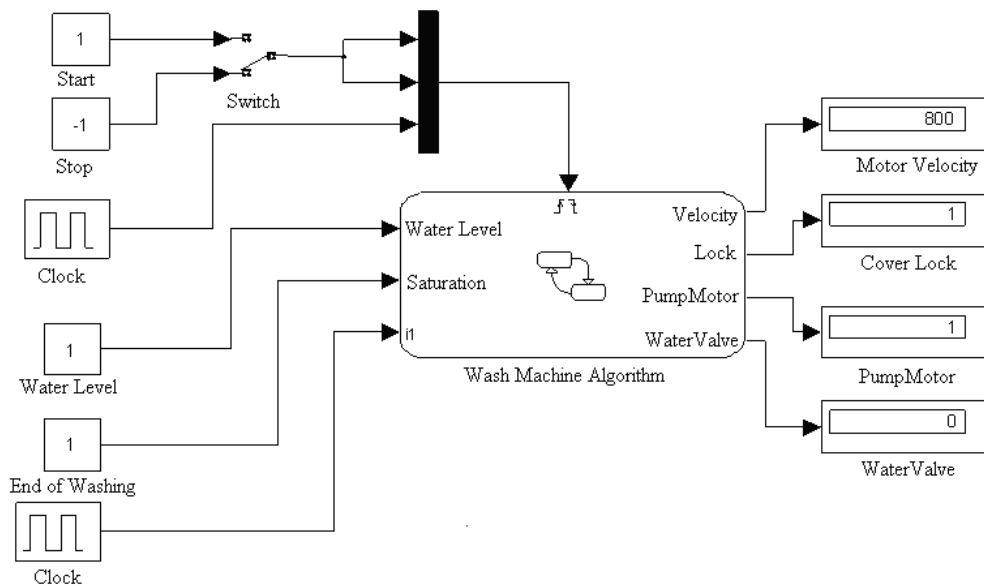


Figure 3. Wash machine block schema in Simulink

All blocks other than washing algorithm represent display and input switches. Core of the model, washing algorithm has been developed by using blocks in the Stateflow.

Simulink enables users to test the algorithm for this system and see the outputs for different inputs. Without coding, Simulink enables check the validity of the algorithm. To test the algorithm represented by the block, Simulink compiles the algorithm in C. It is also possible to reach results in C format.

Blocks and connectors in the flow diagram are used to model algorithm. Figure 4-a represents an illustration washing machine model which characterizes algorithm of the washing duration. Once turn on/ off is pressed, model process functions by visiting subsystems and sub models.

The illustration washing machine algorithm has concatenated blocks. For every set of inputs, active values and blocks could be seen in Stateflow. Thereby, the designer can see the Stateflow also with asked eyes step by step within active blocks. Other durations and processes regarding to the washing machine are seen in Figure 5-a and –b. In case of state change, the flow direction lines –arrows informs users by highlighting itself.

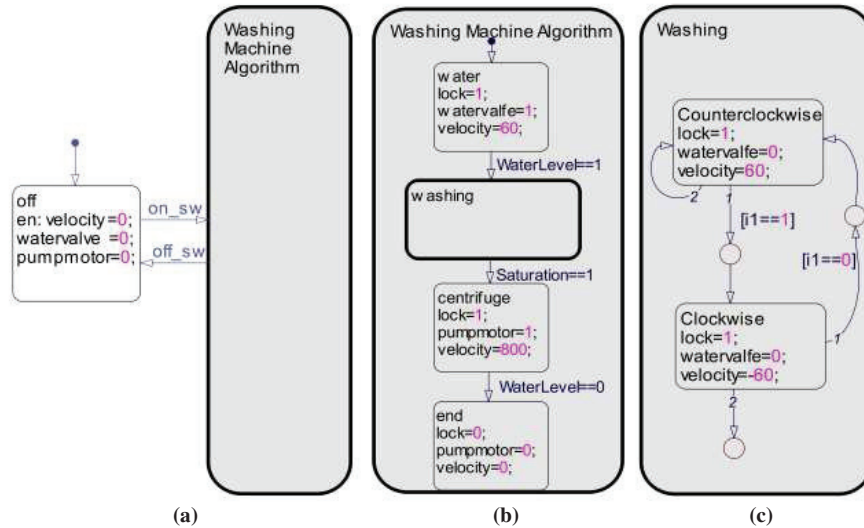


Figure 4 Wash machine in Simulink Stateflow a) main algorithm b) washing algorithm c) function algorithm

Some of the most important features of Simulink Stateflow are that more than one comparison is possible and case adaptive block generation. Subsystem structured calls enable designer to have its desired algorithm structure. Thereby, it is even also possible for students without any prior programming knowledge to improve their problem solving skills. Inputs, outputs, parameters could be seen within model browser.

2. Results

Algorithm learning is the most important part of the studies, where software and programming is conducted. Algorithms designed by students to check system and final software implemented later could be modified many times. Simulink Stateflow enables students to simulate flow diagrams of their projects. Algorithms with flow diagrams could be validated easily.

Whereas traditional algorithm design methods, algorithm design in Simulink Stateflow enables students to test the process steps and problem solving structure and shortens learning period nearby saving money and reducing cost of the final product. By using Simulink Stateflow, designing and developing algorithm can be made easily, more visually, transparently and quickly.

References:

- Gürkan, İ. (2003). Bilgisayar ortamında sorun çözümü. Ankara: Bıçaklar Kitabevi.
- Harel, D. , Feldman, Y. (2004). Algorithmics the spirit of computing. London: Addison-Wesley
- Holloway, J.P. (2004). Introduction to engineering programming solving problems with algorithms. USA: John Wiley & Sons
- Knuth, D.E. (1974). Computer Science and its Relation to Mathematics, American Mathematical, 81:4, 323–343.
- Vatansever, F. (2005). Algoritma geliştirme ve programlamaya giriş. Ankara: Seçkin yayıncılık.
- Mathworks (2008). Design and simulate state machines and control logic. URL <http://www.mathworks.com/products/stateflow>