The FORCE

The Flow of Regulated C Expressions

A graphical programming environment for beginners learning the C language.

Authors: Jeff Alexander, Leonard Kirschen, Michael Pellauer

Introduction

• What is The FORCE?

The FORCE is a tool intended for novice programmers that aids in the process of developing simple C programs by allowing users to manipulate graphical objects which correspond to C semantics. The FORCE easily expresses powerful concepts such as flow of control and parameter passing. Programs generated by The FORCE are always syntactically correct, allowing beginners to focus on implementation rather than sifting through endless obscure compiler errors.

• What isn't The FORCE?

While The FORCE includes a GUI builder, this is by no means its primary purpose. Graphical flow-based programming represents a complete paradigm shift from traditional text-based coding. The FORCE is not intended for experienced programmers to build large scale software. Rather, its unique flow-based "diagrams" allow novices and intermediate programmers to easily move from the design process to implementation.

Requirements Overview

• **Program Builder**: This is the environment used to "write" the program. It involves creating a flow chart-like graph of regulated C expressions. This *program graph* represents the operators, conditionals, loops, functions, and data. The traversal of this graph creates a working program which is completely independent of The FORCE and all but indistinguishable from a hand written program. (Priority: 1)

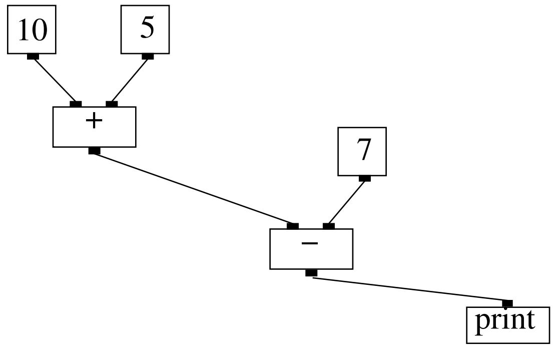
• **Interface Builder**: This is a traditional GUI builder that allows for easy layout of basic control types. It must be seamlessly integrated with the Program Builder so that when a new instance of a control type is created, a program graph customized for it is made available. The user must be able to control non-visual interface events. (Priority: 2)

• **Code Builder**: The Code Builder generates C code from the contents of the Program and Interface Builders. The code must be readable, well structured, and commented. The final program must be completely independent of The FORCE. (Priority: 1)

Program Builder

The Program Graph: The program graph is the fundamental core of The FORCE. It allows the user to build software without writing a single line of code. Instead, the user programs by placing graphical elements which represent basic C operations, functions, and data. These elements are then connected to form a series of operations, conditions, and loops that make up a complete program.

Each element of the program should be represented as a box with input and output "plugs." These plugs are strongly typed and correspond to parameters and return values. By connecting output plugs to input plugs, the user can pass data from one operation to the next. For example, the following diagram illustrates simple addition and subtraction.



This snippet would output “8”.

The types of parameters and return values must be checked as the user creates connections between elements. This will ensure that the code created by The FORCE constitutes a syntactically valid C program.

The user need not need to be aware that his diagrams are being mapped to C code. Because many users will want to view the code after its creation, it must be well commented and feature understandable naming conventions. However, many users will not even be aware that their code is being compiled to C specifically. It schould be emphasized that no knowledge of C is required to use The FORCE.

Some language features that must be supported are base types (priority 1), basic function calls (1), constants (1), literals (1), loops (1), conditionals (1), operators (1), arrays (2), structures (5), and recursion (68). There must be three types of subroutines. The first type is a call to a library function, such as strcmp() or exp(). The program must include the most common standard ANSI C functions. Secondly, the user must be able to define subroutines using the program graph exclusively, effectively creating “sub-graphs.” Lastly, an advanced user must be able to insert actual C code directly into special code elements of the graph.

Code Builder:

The code builder generates the C code for the program graph and the code from the interface builder. The code must be commented, and the names should be consistent with those given by the user. In addition, the code should be well organized and formatted in a clear fashion so that the code can be easily explored.

A low priority requirement is the generation of a Makefile to accompany the user’s code.

Interface Builder:

The interface builder is a tool for attaching graphical interfaces to projects being developed with The FORCE. The builder must be versatile and robust and must allow the user complete control over layout. High priority controls are push buttons, radio buttons, labels, text fields, and sliders. Lower priority controls are scroll bars, multiple windows and/or dialog boxes, and menus.

The interface builder must allow the user to construct programs using an event driven model. Program graphs from the program builder can be associated with different events, such as the click of a button. In addition, the user should be given access to non-visual events such as key presses and application start.

Additional Requirements:

The user must be able to save a project and to reload it later. Special attention should be given to creating a user friendly application, as this system is intended for novice users.