

모델 기반 자동코드 생성기 개발

이정우*, 최경희**, 정기현*
*아주대학교 컴퓨터공학과
**아주대학교 일반대학원
e-mail : ahn2828@ajou.ac.kr

A Development of Model Based Automatic Code Generation

Jeong-Woo Lee, Kyung-Hee Choi**, Ki-Hyun Jung*
*Devision of Computer Engineering, Ajou University
**Graduate School of Ajou University

요 약

본 연구에서는 Simulink Model(이하 Model)을 C Code로 변환시켜주는 자동 코드 생성기, SimAutoCodeGenerator를 제안하였다. Model의 테스트케이스를 생성하기 위한 작업에서 반복적인 프로그램의 실행을 통한 테스터와의 다양한 피드백이 필요하다. 이때, Simulink는 이런 작업을 하는 데 있어서 적절한 정보를 제공하기 어려울 뿐만 아니라 테스트 케이스를 생성하는데 필요한, 실행 시 특정 시점으로 돌아가는 기능을 수행하기 위해서는 프로그램을 정지한 후 다시 실행해야하는 문제점을 가지고 있다. 본 연구에서는 이와같은 단점을 보완하기 위하여 Model을 C 코드로 변환하였다. 생성된 C 코드는 Model과의 일치성을 증명한 후 자동 테스트케이스 생성을 위한 프로그램에 쓰이게 될 것이다. 먼저 C 코드의 생성 메커니즘을 알아보고 생성된 C 코드와 Model의 일치성을 증명하였다.

1. 서론

소프트웨어의 테스트케이스는 요구사항을 만족하는 테스트 입력값을 의미한다. 하지만 요구사항에 테스트 입력값이 명시되어 있지 않은 경우, 주어진 시스템으로부터 도출해야 할 경우도 생기기 마련이다.

Matlab®은 알고리즘을 검증하는데 유용한 도구로서 그 중에서 Simulink는 IP(Intellectual Property) 형태의 수많은 컴포넌트를 가지고 블록 다이어그램 형태의 설계가 가능하며, 높은 추상화를 지원한다[1]. 그러한 장점으로 인하여 Matlab/Simulink는 시스템 모델링에 많은 부분에서 사용되고 있다. 최근의 추세에 따르면 알고리즘 검증단계에서의 자원을 그대로 HW 설계 및 SW 설계에서 사용하고 최종적으로 검증에까지 사용되는 것이 바람직하다[2]. 이 때 검증 단계의 자원은 Simulink 모델을 뜻하며, 검증 단계에서 사용할 실제 동작 가능한 다양한 테스트케이스를 도출해 내는 것은 중요한 작업이다[3].

본 연구에서는 테스트 케이스 도출을 위한 정보(State, Transition 등)를 얻는 방법을 Model을 C Code로 자동으로 변환함으로써 얻고자 한다. 기존의 Matlab/Simulink에서 제공하는 Real-Time Workshop(이하 RTW)을 사용하여 얻은 C Code는 이런 작업을 하는 데 있어서 사용자에게 적절한 정보를 제공하기 어려울 뿐만 아니라 테스트 케이스를 생성하는데 필요한, 실행시 특정 시점으로 돌아가는 기능을 수행하기 위해서는 프로그램을 정지한 후 다시 실행해야 하는

문제점을 가지고 있다. 이 페이퍼의 나머지 부분들은 다음과 같은 내용을 포함하고 있다. (2) 관련연구에서는 Model을 C Code로 변환하는 상용 툴인 RTW를 소개하고, (3) SimAutoCodeGenerator 실행 방법에서는 C Code 생성작업에 필요한 파일을 알아보고 생성 방법을 설명한다. (4) Simulation에서는 간단한 Model인 sf_if Model을 사용하여 Model과 해당모델을 C Code로 변환한 후 그 둘의 실행하여 일치성을 증명한다. (5) 결론 및 향후과제에서는 전체 내용 정리 및 Model의 C Code 변환이 가지는 의미를 간략하게 설명하면서 끝을 내도록 하겠다.

2. 관련연구

Model을 Code로 변환시켜주는 툴에는 Matworks사의 RTW가 있다. RTW는 다양한 운영환경 및 하드웨어 종류에 상관없이 사용될 수 있도록 설계된 개방형 시스템이다.

RTW는 사용자 코드와의 통합을 위해서 C, C++코드를 생성을 지원해 준다[1]. RTW 기능이 포함되어 있는 Matlab®모델링 제어 알고리즘을 사용하는 업계에서 사실표준으로 인정되고 있다. 많은 자동차 회사들(Ford, GM, Toyota)이 Matlab®을 사용하고 있거나 사용하려고 하고 있다. 그리고 항공 회사들에서도 사용하고 있다[4].

(1) Real-Time Workshop 실행 방법

(표 1) 은 RTW 의 사용에 쓰이는 구성 요소를 보여준다.

표 1 RTW 구성 요소

이름	내용
Model.mdl	VHLL(Very High-Level Language)로 이루어진 최상위 추상층, 각 Entity 가 그림으로 표현되어 있다.
Model.rtw	모델의 Description 파일로서 Model 의 각 Entity 간 수직관계가 표현되어 있다.
Target Language Compiler	모델의 Description 파일을 읽어 Model.c 코드를 생성해 낸다. 생성된 코드는 하나의 독립된 프로그램이다.

(그림 1)은 RTW 의 실행 모습을 보여주고 있다.

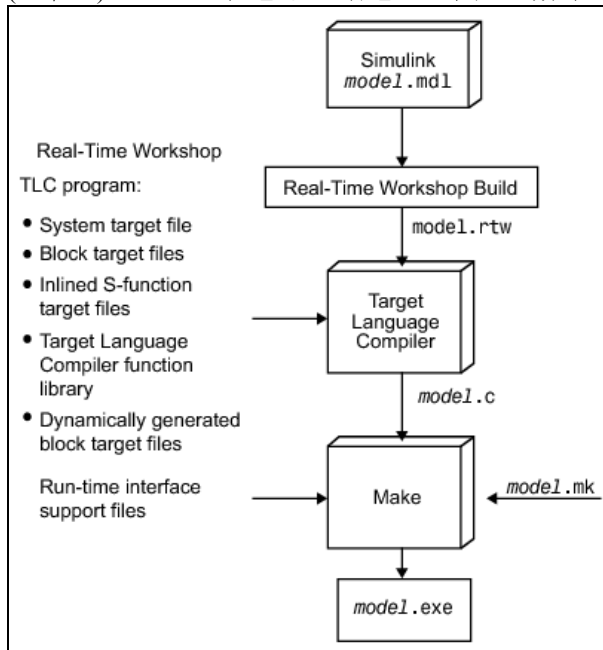


그림 1 RTW 의 실행 과정

이때 코드 생성 부분은 Simulink Model.mdl 파일이 빌드되어서 model description 파일인 model.rtw 로 변환되고 이 model.rtw 파일이 TLC 에 의하여 model.c 파일로 변환되는 과정 까지이다.

실행 과정을 각 step 별로 나누어 설명하면

1) Step 1

- Model.mdl 파일에서 code 생성을 위한 중간 단계인 model.rtw 으로 변호나한다. 이 rtw 파일에는 시스템 아래에 존재하는 block 들의 수직적 연관관계가 나타나 있다.

2) Step 2

- TLC 는 Step 1 의 결과물인 model.rtw 를 C Code 로 변환하기 위하여 존재한다. 이때 생성되는 코드는 전체 시스템을 하나의 코드로 만들거나, 각 블럭별로 생성하는 두 방식중에 선택할 수 있다.

3) Step 3

- Step 2 가 완료되면 C Code 를 결과물로 가질 수 있게 된다. 생성된 C Code 를 makefile 인 model.mk 로 실행시키면 model.exe 가 생성되고, 시뮬레이션을 시작할 수 있다.

(2) Testcase 생성을 위한 Simulation 도구로서의 한계

RTW 를 통하여 생성된 C Code 는 Simulink 의 실행에 최적화된 코드이다. 하지만 테스트케이스 생성을위한 Simulation 에 사용하기엔 많은 제약이 따른다. 테스트케이스 생성 작업 시에 많은 Simulation 작업을 통해 어떠한 테스트케이스로 어떠한 스테이트, 트랜지션들을 방문하였는지, 그때의 통과 조건들은 무엇인지와 같은 정보들이 필요하게 되는데 Simulink 를 사용하게되면 이런 정보들을 수집하는데 있어 제약이 있게된다. RTW 를 사용한 C Code 생성시에는 생성되는 코드를 변형하는 것이 가능하지 않기 때문이다.

만약에 C Code 가 아닌 Matlab/Simulink 를 직접 돌리는 것을 생각 할 수도 있다 하지만 RTW 의 한계에 더하여 Simulink 를 직접 돌리는 작업도 많은 어려움이 있다. 그 중 하나가 실행 시 특정 시점으로 Simulink 를 되돌릴 수 없다는 것이다.

테스트케이스를 생성하는 작업은 반복적인 Simulation 작업을 필요로 하게된다. 왜냐하면 다양한 커버리지를 만족하기 위하여 가능한 모든 조합을 고려하여야 하기 때문이다. 이때 Simulink 항상 다시 시작해야 한다면 성능에 치명적인 영향을 줄 수 있다.

3. SimAutoCodeGenerator 실행 방법

다음 (그림 2)는 C Code 생성 프로세스를 나타낸다.

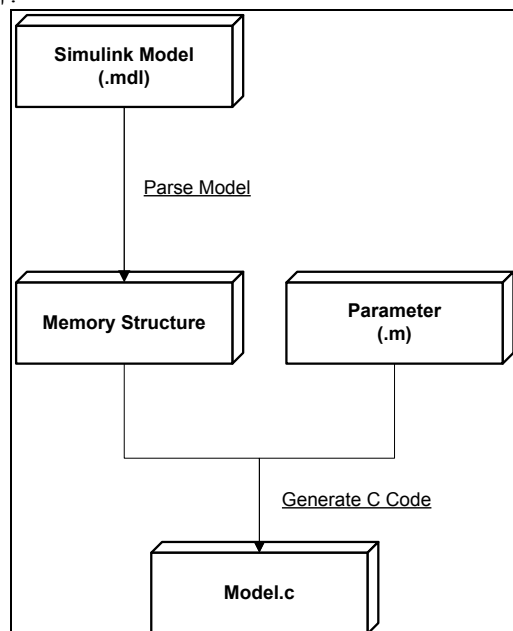


그림 2 SimAutoCodeGenerator C Code 생성 프로세스

상위에 있는 2 개의 데이터는 C Code 생성을 위해 필요한 파일들이다. Simulink Model 과 그 모델속의 데이터를 초기화 시키는데 필요한 .m 형식의 Parameter 파일을 나타낸다.

표 2 SimAutoCodeGenerator 입력파일의 구성

SimAutoCodeGenerator 입력파일	
1. Model(.mdl)	: Model 의 구성
2. Parameter(.m)	: Model 내 변수의 초기값

(그림 2)의 왼쪽 상단부터 내려가면 Simulink Model 은 .mdl 파일형식으로 구성되어 있기 때문에 이 정보를 사용하기 위해서는 mdl 파일을 분석해야 한다. 이는 mdl 파일을 텍스트 형식으로 읽어서 데이터로 저장한다. 이렇게 분석된 데이터의 구조와 Parameter 파일, 의 정보를 조합하여 C Code 를 생성시킨다.

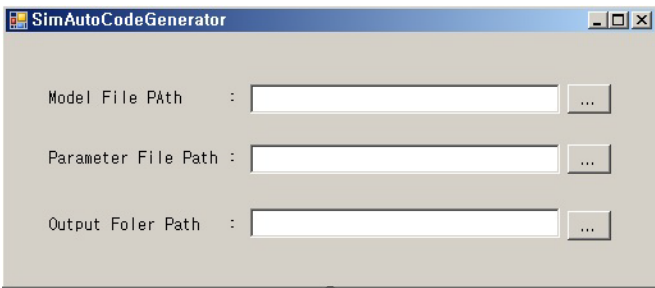


그림 3 SimAutoCodeGenerator UI

3.1 Code Generation Steps

(1) Step 1

소스 파일인 Model.mdl 파일을 읽어들이 각 Entity 별로 Memory 에 정리한다.

(표 3)은 해당 Model.mdl 파일로 부터 얻은 Entity 의 종류를 보여준다.

표 3 Model File 로부터 얻은 Entity 종류

Entity	역할
블럭	Simulink 상에 존재하는 Inport, Output, Subsystem 등의 블럭을 의미한다. 내부에 서브 블럭을 가질 수 있다.
포트	각 블럭에서 다른 Entity 와의 연결하기 위한 Entity 이다. 이 연결은 라인을 통해서 이루어진다.
라인	포트와 포트사이를 연결하는 역할을 한다.
스테이트	Stateflow 상에 존재하는 AND, OR 스테이트를 의미한다. 내부에 자식 스테이트를 가질 수 있다.
트랜지션	스테이트간의 이동을 의미하는 Entity 이다. 트랜지션은 구성요소로서 Condition 을 가지고 있기 때문에 해당 Condition 을 만족하여야만 트랜지션이 실행될 수 있다.

정션	상태를 가지고 있지 않은 스테이트이다.
데이터	숫자와 이벤트로 나뉜다. 숫자 는 int, float 등의 타입을 가지고 있고 이벤트는 bool 타입으로서 해당 이벤트의 종류에 따라서 특정 Action 을 실행하게 된다.

(2) Step 2

Step 1 에서 생성된 Data 와 각 Entity 의 값을 초기화한다. 이때 데이터에 존재하는 Initfrom Workspace 의 필드의 값에 따라서 Parameter 파일에 존재하는 값으로 초기화 할지 결정한다.

4. Simulation

생성된 C Code 는 Simulink/Stateflow Model 과의 일치성이 검증되어야 한다. 이는 C Code, Simulink/Stateflow 에 같은 Testcase 를 입력하여 동일한 output 을 생성함으로써 검증할 수 있다.

(1) Simulation 에 사용된 Simulink/Stateflow Model

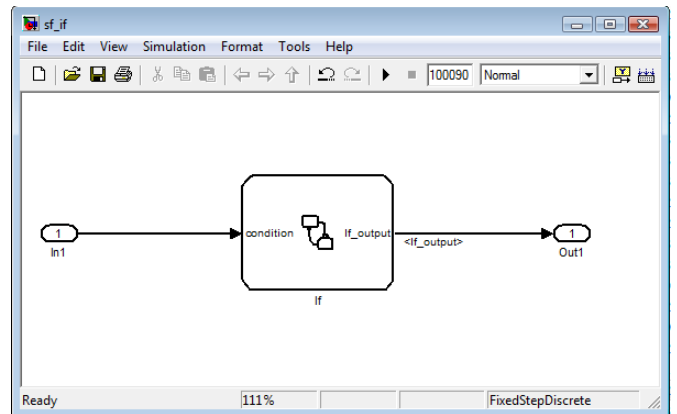


그림 4 sf_if Model

(그림 4)는 하나의 스테이트와 3 개의 정션, 그리고 6 개의 트랜지션으로 이루어진 간단한 Model 이다.

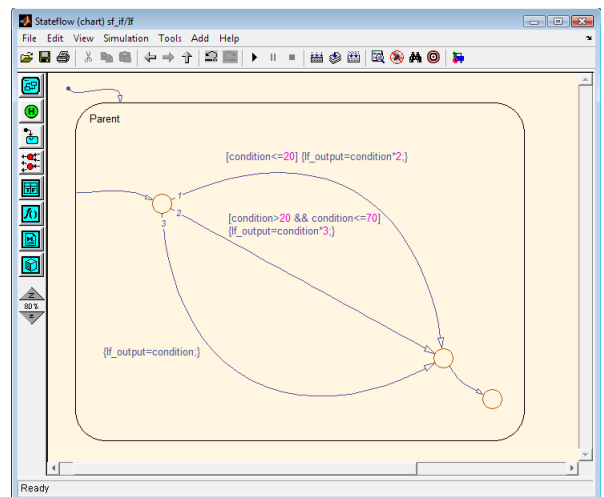


그림 5 sf_if 모델의 내부

SimAutoCodeGenerator 는 Simulink/Statflow Model 의 기능을 그대로 실현할 수 있는 C Code 를 생성한 것이기 때문에 정확하게 C Code 가 생성되었는지 검증하는 작업이 필요하다. 테스트에 사용될 테스트케이스를 결정할 때는 고려해야 할 사항이 있다. 정확한 테스트를 위해서 테스트케이스의 커버리지가 높아야 한다는 것이다.

아래 (그림 5) sf_if 모델을 보면 스테이트 Parent 속 에 첫번째 정선으로 부터 세개의 트랜지션이 발생하는 것을 알 수 있다. 이때 condition 변수의 값에 따라서 Action 이 달라지므로 테스트케이스의 범위는 최소 20 ~ 70 이라는 것을 알 수 있다.

Sf_if 모델의 슈도코드는 아래와 같다.

표 3 sf_if Model 의 슈도코드

1. While (until program halt)
1.1 if condition <= 20
1.1.1 if_output = condition * 2
1.2 if condition > 20 && condition <= 70
1.2.1 if_output = condition * 3
1.3 Else
1.3.1 if_output = condition

(2) 3 개의 테스트케이스 적용 및 결과 비교

모든 Transtion 을 만족하도록 하기 위하여 테스트케이스는 컨디션 범위 20 보다 작은 10, 20 과 70 사이에 존재하는 50, 70 보다 큰 100 을 입력하였다. 이 경우에 모든 스테이트 뿐만 아니라 정선, 트랜지션까지도 만족하는 결과가 나온다.

1 개의 인풋, 1 개의 아웃풋을 가진 모델을 총 3 가지 테스트케이스를 입력하여 아웃풋을 비교하였다. 스테이트와 트랜지션의 커버리지는 100%를 만족 하였다.

표 4 테스트케이스 아웃풋 검증

Model		C Code	
Input	Output	Input	Output
10	20	10	20
50	150	50	150
100	100	100	100

5. 결론 및 향후 과제

이 논문에서는 업계에서 표준으로 받아들여지고 있는 모델링 언어인 Simulink/Stateflow[4]의 Model 을 C Code 로 변환하는 프로그램인 SimAutoCodeGenerator 를 제안하였다. MathWorks 사에서 Real-Time Workshop 이라고 하는 다양한 운영환경 및 하드웨어 종류에 상관없이 사용될 수 있도록 설계된 개방형 시스템을 지원해 주고 있지만 Simulink/Stateflow 는 Simulation 중 에 다양한 테스트 커버리지를 만족하기 위하여 필요한 정보를 제공하기에는 사용자에게 단혀있는 Tool 이다. SimAutoCodeGenerator 를 이용하여 생성된 C Code

는 테스터에게 기존의 Simulink 보다 테스트 커버리지에 관련된 정보를 다양하게 제공해 줄 수 있음을 이해하고 이를 통하여 완벽한 테스트를 할 수 있는 테스트케이스 생성을 기대해 본다.

참고문헌

[1] The MathWorks, <http://www.mathworks.com>
 [2] 송문빈, 송태훈, 오재곤, 정연모, “효율적인 통합 시뮬레이션에 의한 스피커 연결 시스템의 SoC 설계”, 2006 년 10 월 전자공학회 논문지 제 43 권 SD 편 제 10 호
 [3] 김성조, 최경희, 정기현, “Simulink 에 적용되는 Testing Framework”, 2008 년 11 월 한국정보처리학회 추계학술대회 논문집 제 15 권 제 2 호
 [4] Ranville S., Black P. Automated Testing Requirments-Automotive Perspective. // The Second Int. Workshop on Automated Program Analysis, Testing and Verification. 2001(<http://hissa.nist.gov/~black/Papers/autoTestReqsW/APATV.rtf>)