

순환공학 환경을 위한 ATM 명세

코드 변환기 설계 및 구현

고 현*, 조상규, 이연식
군산대학교 컴퓨터정보학과
e-mail : kogo@cs.kunsan.ac.kr

Design and Implementation of ATM Specification Code Converter for Software Round-Trip Engineering Environment

Hyun Ko*, Sangkyu Joe, Yonsik Lee
Dept. of Computer Information Science, Kunsan National University

요 약

본 논문은 ATM(Abstract Timed Machine)으로 명세된 실시간 시스템에 대한 재/역공학 측면에서의 검증을 위한 Ada 코드로의 변환 과정에서 요구되는 DoME/ATM 파스트리 생성과정과 이를 통해 추출되는 정보를 이용하여 ATM을 SRL/ATM 으로의 변환방법을 설계 구현한다. 임무 위급 시스템 같은 실시간 시스템을 명세, 분석, 검증하기 위한 정형기법인 ATM은 순환공학에서의 실시간 시스템 속성은 물론 특정 환경과 동적 정보 등을 명세하기 위한 정형기법이다. 본 논문에서는 DoME를 이용하여 실시간 시스템에서의 특정 요구사항에 대한 DoME/ATM을 생성하고, DoME의 기능을 이용하여 ATM을 스크립트 코드로 저장한 후, DoME/ATM 파스트리 생성기를 통해 명세 정보를 추출하여 SRL/ATM으로의 변환기를 설계 구현한다. 변환기에 의해 생성된 매개언어인 SRL은 명시적인 정의와 효율적 분석 정보를 제공함으로써 Ada, COBOL, C 등과 같은 실행코드를 생성할 수 있도록 하는 기반을 제공한다.

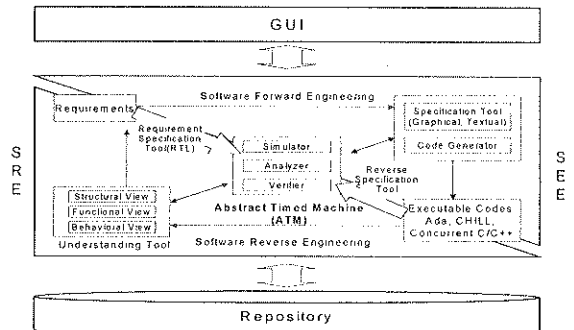
1. 서론

임무위급 시스템(mission-critical real-time system)은 특정 시간 간격 내에 특정 입력에 대한 결과를 발생시켜야 하는 시스템으로 실시간적 발생 상황에 대해 정확한 치리와 예외상황에 대한 신속하고 적절한 조치를 요하는 특성을 가진다. 따라서 특정 상황에 대한 시스템의 여러 동적 행위를 표현하고 다양한 속성을 명세할 수 있는 정형기법이 필요하게 되었다. 그러나 실시간 시스템을 명세하는 기존의 정형 기법들[1, 2, 5, 7, 8, 9]을 통해 역공학 과정에서 소프트웨어의 동적 행위를 이해하는 것은 매우 어렵다. 이는 기존의 정형기법들이 순공학 과정에서 실시간 시스템을 설계, 명세하기 위해 고안되어 실제 시스템이 특정 물리적 환경에서 실행될 때의 동적 행위를 완전하게 표현하지 못하기 때문이다. 따라서 순공학 과정에서만 적용가능한 제한된 정형기법으로 역공학에서 수백만 줄의 코드를 가진 소프트웨어를 표현하기에는 부적합하다.

ATM은 [그림 1]에서 보여주는 마와 같이 SEE (Software Engineering Environment)/SRE (Software Reverse Engineering Environment), 즉 순공학과 역공학의 통합환경인 순환 공학 환경(Software Round-Trip Engineering Environment)에서 실시간 시스템을 명세, 분석하기 위한 정형 기법이다[10].

본 논문에서는 실시간 시스템을 명세, 분석 및 검증할 수 있는 정형기법인 ATM을 이용한 명세에 대한 Ada와 같은 실행코드를 생성하기 위한 연구과정 중, DoME/ATM과 그에 대한 파스트리를 생성하고 트리를 통해 추출된 정보를 이용하여

실행코드 생성을 위한 매개언어인 SRL/ATM으로 변환하는 변환기를 설계 구현한다.



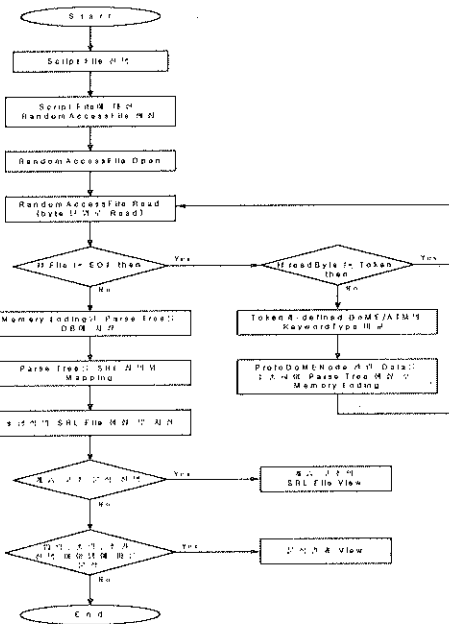
[그림 1] 순공학과 역공학의 통합 순환 환경

2. ATM 명세

2.1 ATM 명세모델

다음 [그림 2]는 DoME를 이용해 개발한 ATM 명세모델이다. DoME 명세 파일에서는 ATM에서 필요로 하는 여러 아이콘과 각 아이콘의 상속관계 및 연관관계, 각 아이콘의 모양, 메소드 등과 같은 아이콘이 포함하는 요소들을 명세한다. 이러한 DoME 명세 파일을 통해 ATM 명세모델이 생성된다.

본 연구는 한국과학재단 특정기초연구(과제번호 1999-2-303-003-3) 지원으로 수행되었음

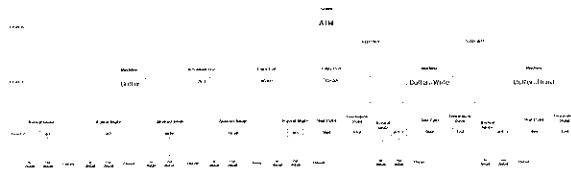


[그림 5] DoME/ATM to SRL/ATM 변환 시스템의 동작 흐름도

3.2.1 ATM 파스트리 생성기

ATM 파스트리 생성기는 그래픽 명세 내용으로부터 명세의 각 요소에 대한 정보를 추출하여 파스트리로 구성하는 도구이다. ATM 트리에서 각 노드는 명세 단위인 그래프, 머신, 모드, 포인트, 포트가 된다. 이는 ATM의 그래프는 머신과 포트의 집합이고, 머신은 모드와 포인트의 집합이라는 ATM의 기본 정의에서 유도된다. 즉 ATM의 그래프, 머신, 모드, 포인트, 포트는 내부에 서브 명세를 포함할 수 있는 블록단위이고, ATM 트리의 각 노드도 상위노드와 하위노드의 관계를 같은 블록구조에 기반한 트리이기 때문이다. 이러한 ATM 트리를 통하여 ATM 명세의 블록구조, 계층구조, 명세에서 나타난 전이에 따른 동작의 흐름 등을 분석할 수 있다.

다음 [그림 6]은 PBC 예제에 대한 ATM 명세를 ATM 파스트리 생성기를 통해 ATM 각 구성요소에 관한 정보를 추출하여 생성한 파스트리의 개층구조를 보인다.



[그림 6] Buffer 머신에 대한 파스트리

다음 [그림 7]에서는 ATM 명세도구에 의해 명세된 프로토타입 노드의 클래스 스키마를 보인다.

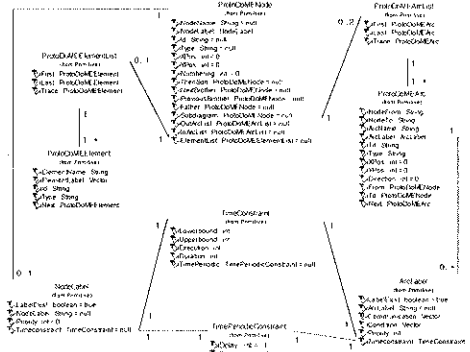
3.2.2 ATM 파스트리 Unloader

ATM 파스트리 Unloader는 ATM 파스트리 생성기를 통해 생성된 ATM 파스트리로부터 시스템의 계층구조와 추출된 ATM 명세의 각 요소에 대한 정보를 .s와 같은 파일형태의 SRL 파일로 언로드(Unload)한다.

3.2.3 데이터베이스 Unloader

데이터베이스 Unloader는 ATM 명세로부터 생성된 ATM 트리의 정보를 데이터베이스로 저장하기 위한 도구이다. 각 분석

기의 효율적인 탐색을 가능하게 하는 테이블과 뷰가 결정되어 데이터베이스에 저장된 정보는 개발자나 사용자에 의한 다양한 명세 분석 및 자동화 도구의 개발을 용이하게 한다.



[그림 7] ATM 명세 프로토타입 노드 클래스 스키마

4. 결론 및 향후과제

본 논문에서는 순공학 과정에서 ATM으로 실시간 시스템을 명세하고, 명세분석을 통해 이의 실행코드 변환 시 효율적 코드 생성을 위한 매개언어인 SRL로의 변환을 수행하는 SRL/ATM 코드 변환기를 설계 구현하였다. 또한, 이 과정에서 실시간 시스템의 명세를 위하여 모델링 도구인 DoME를 이용하여 ATM 명세도구를 개발하고, 명세에 대한 분석을 통해 명세 정보를 추출하는 DoME/ATM 파스트리 생성기를 구현하였다. 구현된 DoME/ATM 파스트리 생성기는 이해지원 도구와 같은 분석도구나 효율적 코드 생성을 위한 매개언어인 SRL로의 변환에 필요한 정보들을 제공한다. 또한, 변환기에 의해 생성된 매개언어인 SRL은 명시적인 정의와 효율적 분석 정보를 제공함으로써 Ada, COBOL, C 등과 같은 실행코드를 생성할 수 있도록 하는 기반을 제공한다.

향후 연구과제로는 시스템 순공학 과정에서의 요구명세 도구, 명세로부터 코드를 생성하기 위한 코드 생성기 및 자동화 도구들의 개발이 요구되며, 매개언어인 SRL에 대한 분석을 통해 정적 정보와 동적 정보 추출하여 다양한 실행코드를 생성할 수 있는 분석도구와 실행코드 생성기 개발이 필요하다.

참고문헌

- [1] A. Shaw, *Communicating Real-Time State Machines*, IEEE Transactions on Software Engineering, Vol. 18, No. 9, pp. 805-816, September, 1992
- [2] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, 1985
- [3] Honeywell, DoME Guide, Honeywell, Inc., 1999
- [4] C. Heitmeyer and D. Mandrioli, *Formal Methods for Real-Time Comouting: An Overview*, John Wiley and Sons Ltd., 1995
- [5] D. Harel, *Statecharts: A Visual Formalism for Complex System*, *Science of Computer Programming*, Vol. 8, pp. 231-274, 1987
- [6] Feldman and Koffman, *Ada95*, Addison-Wesley, 1996.
- [7] I. Kang and I. Lee, *State Minimization for Concurrent System Analysis Based on State Space Exploration*, *Proceedings of Conference on Computer Assurance*, Gaithersburg, June, 1994
- [8] S. Jahanian and A. Mok, *Modechart: A Specification Language for Real Time Systems*, IBM Technical Report: RC 15140, November, 1989
- [9] Sitaram C. V. Raju, *An Automatic Verification Technique for Communicating Real-Time State Machines*, Dept. of CS&E at University of Washington, TR 93-04-08, 1993
- [10] 노경주, 박지연, 이문근, 실시간 시스템의 순환공학을 위한 정형기법: 주상시간기계, 한국정보과학회 학술발표논문집(A), 27권 1호, pp. 477-479, 2000