2001 년 한국정보처리학회 추계 학술발표논문집 제 8 권 제 2 호

정형명세에 대한 실행코드 생성을 위한 정보 추출

고 현*[1], 이문근[2], 이연식[1]
군산대학교 컴퓨터정보과학과[1], 전북대학교 전자정보공학부[2]
e-mail : kogo@kunsan.ac.kr

Extraction of Information for Executable Code Generation to Formal Specification

Hyun Ko*[1], Moon-Kun Lee[2], Yonsik Lee[1]
Dept. of Computer Information Science, Kunsan National University[1]
Division of Electronics and Information Engineering, Chonbuk National University[2]

요 약

본 논문은 순환공학 환경에서 실시간 시스템 개발 및 검증을 위한 실행코드 생성기 구현과정에서 ATM(Abstract Timed Machine)으로 명세된 실시간 시스템에 대한 정형명세와 명세분석을 통해 생성된 SRL(Software Representation Language) 코드로부터 실행코드로의 변환을 위해 필요한 정보들을 명세하고 이들의 추출 방법을 제시한다. ATM정형기법을 적용하여 실시간 시스템 개발 및 검증을 위해서는 시스템 명세도구나 명세분석을 위한 분석기, 코드 생성기, 검증기 등과 같은 자동화 도구의 개발이 요구된다. 따라서, 본 논문에서는 순환공학 환경에서의 실시간 시스템의 효율적 개발 및 검증을 위하여 사용될 실행코드 생성기 구현을 위하여, 특정 물리적 환경에서의 실시간 시스템을 명세한 DoME/ATM 스크립트 코드에 대한 SRL 코드로의 변환 시 필요한 정보들을 추출하고, 이를 기반으로 SRL 분석기와 SRL 파스트리 생성기를 이용한 Ada 실행코드로의 변환 시 필요한 실행코드 모듈 구조 및 실행구문, 코드 실행 우선순위 결정 등과 같은 관련된 정보들의 추출 방법을 제시한다.

1. 서론

실시간 시스템(Real-Time System)은 임의의 제한된 범위나 시간제약 내에서의 효율적인 작업 수행 및 처리 결과에 대한 높은 신뢰성과 정확성 보장은 물론 외부환경과의 상호작용을 올바르고 신뢰할 수 있도록 제어하는 시스템이다. 이러한 작업 수행 특성 및 제약성들로 인해 실시간 시스템 대다수가 복잡하고 대규모의 시스템으로 구축되어짐에 따라 유지보수의 어려움과 막대한 초기 시스템 개발 비용이 요구됨으로써 재사용 가능하고 신뢰성 있는 시스템을 개발하기 위한 방법으로 정형기법이 필요하게 되었다. 기존의 정형기법들[1, 2, 3, 4, 5, 6]의 경우, 순공학 과정의 제한된 개발 환경 범위에서만 실시간 시스템을 설계, 명세하도록 고안되어 특정 물리적 환경내에서의 실시간 시스템 실행 시 동적 측면에 대한 적절한 표현력을 제공하지 못한다. 반면, ATM은 순공학과 역공학이 하나로 통합된 순환공학 환경내에서 실시간 시스템을 명세, 분석, 검증할 수 있는 정형기법으로서 그래픽 표기를 이용하여 실시간 시스템을 명세함으로써 완전한 정형성에 기반을 두기 보다 직관적인 이해도를 높여 보다 이해하기 쉽고 적용하기 용이하다. 또한, 실시간 시스템의 속성 기술은 물론 순환공학 과정에서 특정 물리적 환경 요인들과 실행 시의 동적 측면을 완전하게 표현할 수 있다. 이러한 ATM정형기법을 적용하여 실시간 시스템 개발 및 검증을 위해서는 시스템 명세도구나 명세분석을 위한 분석기, 코드 생성기, 검증기 등과 같은 자동화 도구의 개발이 요구된다[7,8]. 따라서, 본 논문에서는 순환공학 환경에서의 실시간 시스템의 효율적 개발 및 검증을 위하여 사용될 실행코드 생성기 구현을 위하여, 2장에서 DoME/ATM 명세에 대한 스크립트 코드로부터 SRL/ATM으로의 변환을 위해 필요한 정보를 추출하고, 3장에서 SRL/ATM의 구성요소와 요소간의 관계에 대한 분석, 분석된 정보와 Ada 실행코드 구문과의 관계를 규정한다. 그리고, 4장에서는 이들을 기반으로 SRL/ATM으로부터 Ada 실행코드로의 변환 시 필수적으로 요구되는 모듈 구조 및 구문, 코드의 실행 우선순위와 같은 동적 및 정적 정보들에 대한 추출방법을 제시하고, 마지막 5장에서는 결론 및 향후 연구과제에 대해 기술한다.

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

2. DoME/ATM 스크립트 코드

2.1 ATM 명세에 대한 스크립트 코드의 구성요소

자동화 명세도구인 ATM 명세도구를 이용한 실시간 시스템에 대한 DoME/ATM 명세는 명세모델을 구성하는 각 아이콘의 정보를 기준으로 DoME에서 제공되는 기능에 의해 자동으로 스크립트 코드(텍스트 GUI-ATM 명세)로 저장된다[7]. 텍스트 형태의 스크립트 코드는 명세 모델에서의 전체 시스템을 ProtoDoMEGraph로 표현하고, 머신, 포트, 모드, 포인트를 각각 ProtoDoMENode로 구성하여 ProtoDoMEGraph 내부에 포함한다. 또한 머신에서 포트간, 모드에서 모드간, 모드에서 포인트간 전이들은 각각 ProtoDoMEArc로 표현하며, 각 ProtoDoMEArc는 GrapEArcName으로 구별된다. 이러한 각 구성요소들은 식별을 위한 속성과 다른 요소와의 관계 표현을 위한 속성 등을 내부요소로 가진다[8].

2.2 DoME/ATM으로부터 SRL로의 변환을 위한 정보추출

DoME/ATM 명세에 대한 스크립트 코드로부터의 정보추출은 아이템들간의 포함관계에 따른 구조정보와 각 아이템들을 구분하는 식별정보의 추출로 이루어진다. 텍스트 형태의 스크립트 코드는 DoME/ATM 명세에서 비정형적으로 나타난 아이템간 계층구조 및 통신에 따른 병행관계, 선후시행관계를 표현한다. 머신 간의 활성 통신에 따른 독립적 머신들의 병행관계 및 머신 동작에 있어서의 선후시행관계는 아이템들의 위치정보 속성을 기반으로한 머신에서 포트간 전이와 이에 해당하는 전이 Name 속성에 의해 나타난다. 아이템간 계층구조는 스크립트 코드에서 머신, 포트, 모드, 포인트에 대한 각 ProtoDoMENode 간 포함관계([,]로 표현) 및 들여쓰기(indentation)를 통해 표현된다. ProtoDoMEArc는 머신 동작시의 시간제약이나 이벤트, 조건 등과 같은 제어흐름에 따른 실제 모드간의 연관관계 및 머신의 내부적 계층관계를 명시적으로 표현한다. 이러한 정보들은 실행코드 생성에 있어 모듈 구조와 구문, 코드의 생성위치 및 코드 실행 순위의 결정 등과 관련된 정보들임으로 SRL로의 변환시 반드시 추출되어야 한다. 다음 [표 1]은 ATM 정형명세에 대한 스크립트 코드에서 추출되어지는 정보들을 나타낸다.

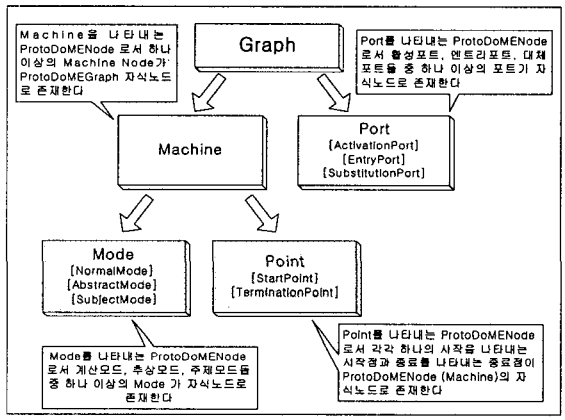
[표 1] ATM 정형명세에 대한 스크립트 코드의 추출정보

구성요소	속 성	추출 정보 내용
ProtoDoMEGraph	ModelTypeName	계층구조상 최상위 루트를 나타내는 Graph이름
	Nodes	구성요소들이 Machine, Mode, Port, Point로 변환
	Arcs	Mode 간의 연결을 나타내는 전이
	Id	자신만의 고유번호(동일 아이템과 구분)
	value:Properties	아이템의 고유번호 (아이템들 간의 식별자)
ProtoDoMENode (Machine)	name	Machine의 이름
	components	자신의 아이템내에 포함된 구성요소를 나타냄 (포함요소로 ProtoDoMENode를 가짐(SRL에서 Mode와 Point로 표현))
	Id	자신만의 고유번호(동일 아이템과 구분)
	value:Properties	아이템의 고유번호(아이템들 간의 식별자)
ProtoDoMENode (Mode)	name	Mode의 이름
	components	자신의 아이템내에 포함된 구성요소를 나타냄 (포함요소로 ProtoDoMEElement를 가짐(SRL에서 Mode의 Code속성으로 표현))
	Id	자신만의 고유번호 (동일 아이템과 구분)
	value:Properties	아이템의 고유번호(아이템들 간의 식별자)
ProtoDoMEArc	indexedOrigin	전이 연결이 시작되는 아이템
	indexedDestination	전이 연결이 끝나는 아이템
	components	자신의 전이내에 포함된 구성요소를 나타냄 (포함요소로 GrapEArcName를 가짐 (SRL에서 전이의 이름을 나타내거나 조건 또는 시간제약 등을 나타냄)
ProtoDoMEElement	name	시간적 관점에서 상태가 확장된 개념인 SRL/ATM에서의 Mode내 실행 구문
	Id	자신만의 고유번호
	value:Properties	아이템의 고유번호(아이템들 간의 식별자)
GrapEArcName	name	전이의 이름
	id	자신의 고유번호

3. SRL/ATM

3.1 SRL/ATM의 구성요소와 요소간 관계 분석

SRL(Software Representation Language)은 시스템을 구성하는 소프트웨어, 하드웨어, 통신망, 형상, 요구사항, 설계, 성능 등에 관한 정보를 저장소에 보관 및 관리하고 이를 재/역공학 과정에서 사용자의 특정 목적에 부합하게 표현하고 분석, 이해 및 평가를 하기 위한 메타언어이다. SRL은 실시간 시스템을 구문노드와 수식노드로 표현하며, 시스템의 중첩구조에 따라 구문노드는 형제노드와 자손노드로 표현되고, 구문의 상세 정보를 표현하기 위해 수식노드를 하위노드로 갖는 추상구문트리(Abstract Syntax Tree) 형태를 갖는다. SRL/ATM은 DoME/ATM 대한 명세분석을 통해 ATM을 SRL 코드로 표현한 형태이다. SRL/ATM은 DoME/ATM 정형명세에 대한 스크립트 코드 내에서의 아이템들 간의 전이와 통신에 따른 관계성 및 포함관계에 따른 구조 정보의 추출을 통해 실행코드로의 변환을 위해 필수적으로 요구되는 정보들을 그대로 표현, 유지한다. SRL/ATM은 스크립트 코드에서의 ProtoDoMENode와 ProtoDoMEArc의 코드 내 위치적 분리 표현을 통합하여 표현한다. 각 ProtoDoMENode로 표현된 아이템들은 식별정보를 통해 각각 Machine, Port, Mode, Point 노드로 구분되고, ProtoDoMEArc는 이러한 노드들의 내부속성인 Out으로 표현함으로써 각 노드간 관계 정보와 제어 흐름 파악에 있어 높은 이해도를 제공한다. 이러한 구분 표현은 각 노드에 대한 식별정보를 기반으로 노드들 간의 포함관계 및 상호작용 관계에 따른 명확한 계층구조를 표현한다. SRL/ATM은 스크립트 코드에서의 계층구조 표현과 동일한 방법인 포함관계와 들여쓰기(indentation)를 통해 계층구조를 나타내며, 동적 및 정적 정보의 표현은 각 노드의 속성을 통해 표현한다. 다음 [그림 1]은 SRL/ATM 구성요소들 간의 계층관계를 나타낸다.

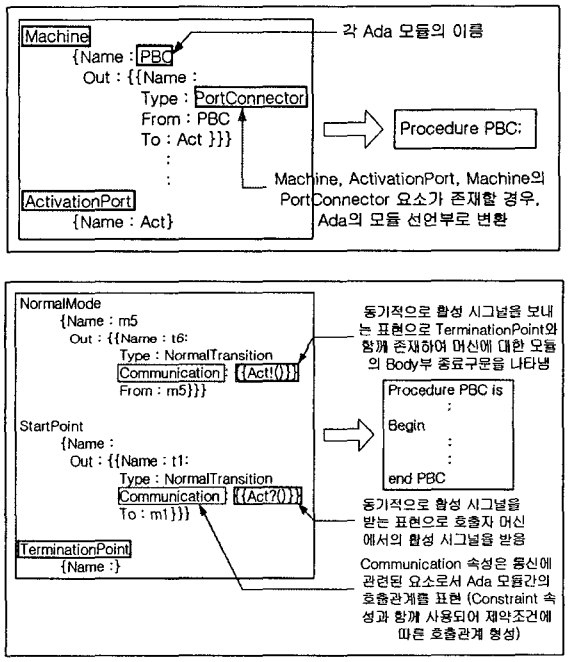


[그림 1] SRL/ATM에서의 구성요소간 계층관계

3.2 SRL/ATM의 구조 및 제어흐름과 Ada와의 관계

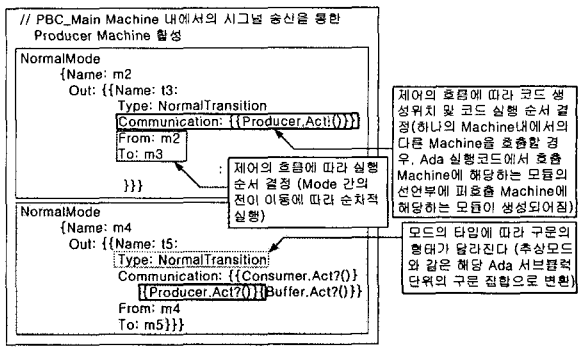
SRL/ATM 구성요소들의 계층구조는 Ada 실행코드의 모듈구조와 대응한다. SRL/ATM 노드 중 Graph는 전체 실시간 시스템에서 하나의 부분적 역할을 수행하는 실제 시스템으로 나타낼 수 있고, Machine은 하드웨어 측면에서의 태스크 단위와 대응되는 것으로 실제 Ada 실행코드에서 프로그램을 구축하는 기본단위 프로그램인 부프로그램(Procedure, Function, Package, etc.)과 Task, Generic, Protected, Entry 등과 같은 모듈구조로 표현할 수 있다. 즉, 각 Machine은 해당 처

리작업의 특성에 따라 적합한 모듈구조로 변환되고, 각 독립된 머신들 간의 활성 통신 및 데이터 통신 관계는 각 모듈간의 호출관계로 표현된다. 또한, SRL/ATM은 실제 실행코드에서의 실행구문이나 수식구문 등을 각 노드의 속성을 통해 표현한다. 각 노드들은 실행코드에서 하나의 개념적 블록단위를 형성하고, 블록 내 구문의 특성에 따라 노드 속성들은 실행문의 집합이나 일반적인 수식계산 구문 등으로 변환된다. 또는 각 노드들 간의 전이관계에 따라 노드에 해당하는 개념적 블록들이 서로 결합하여 실행코드 상에서 하나의 반복, 선택 또는 조건 구문과 같은 서브블록을 이루게 된다. 다음 [그림 2]는 SRL/ATM의 구성요소와 Ada 모듈 구조와의 관계를 나타낸다.



[그림 2] SRL/ATM 요소와 Ada 모듈 구조와의 관계

ATM에서의 각 머신간 통신과 제어흐름은 Ada 실행코드의 우선실행 순위와 관련하여 실제 아이템간의 제어 흐름에 따라 Ada 코드가 실행되는 것과 같다. 이는 머신의 활성 또는 활성종료 등과 같은 동기/비동기화 통신과 대응하여 모듈의 시작과 종료가 결정되며, 모드내의 작업처리에 따른 상태변화는 순차적인 Ada 구문이나 서브블록의 실행 등으로 나타낼 수 있다. 작업처리에 있어 정해진 시간 범위에 따른 제약이나 다음 상태로의 전환을 위한 조건만족, 조건에 따른 선택적 상태로의 전환 등은 각각 Ada 구문의 서브블록에 해당하는 반복구문, 조건구문, 선택구문 등으로 표현할 수 있다. 이러한 Ada 구문들의 코드변환 시의 생성 위치는 SRL/ATM 노드들이 갖는 속성에 따라 결정되거나 노드의 실제 속성값에 대한 구문분석을 통해 생성위치가 결정되어 진다. 즉, Ada 구문에서의 변수 선언 및 모듈 선언 등과 같은 요소들은 실제 Main 모듈의 선언부에 생성되며, 각 머신에 해당하는 모듈에서의 실행부는 모듈 Body부의 구현부분에 생성되어 진다. 다음 [그림 3]은 머신 내 활성 호출에 따른 Ada 코드에서 호출자와 피호출자 모듈의 생성위치 결정 관계를 나타낸다.



[그림 3] 머신 내 활성 호출에 따른 실행코드의 생성위치 결정 관계

4. SRL/ATM 으로부터 실행코드로의 변환을 위한 정보추출

4.1 SRL/ATM 에서의 정보 추출 과정

SRL/ATM에 대한 정보 추출 과정은 먼저 SRL/ATM 분석기를 통해 입력된 SRL 코드를 분석하게 된다. SRL/ATM 분석기는 SRL 코드를 Token 단위로 읽어 정의된 SRL/ATM의 KeywordType과 비교하여 각 구성요소에 대한 정보와 각 구성요소간 전이 이동 시 이벤트 호출이나 시간제약, 조건에 대한 속성 정보를 추출한다. 또한 Token 단위로 SRL 코드를 계속적으로 읽어들일 때, 포함관계를 나타내는 [,] 문자나 다음에 위치한 KeywordType 요소로서의 어휘단위를 구분하여 어휘분석 과정을 통해 각 노드 간의 전이 이동 및 통신관계, 즉 제어의 흐름을 추적하여 실행코드의 생성 위치 및 노드의 탐색순위를 결정하게 된다. 이렇게 추출된 정보는 SRL/ATM 파스트리 생성기를 통해 파스트리를 생성한 후, 트리를 구성하는 노드들의 우선순위에 따른 탐색을 통해 각 해당 Ada 구문과 매핑되어 진다. 다음 [표 2]는 SRL/ATM의 정보 추출 과정에서 제어 흐름을 나타내는 각 요소의 Out 속성에 대한 추출 정보들을 나타낸다.

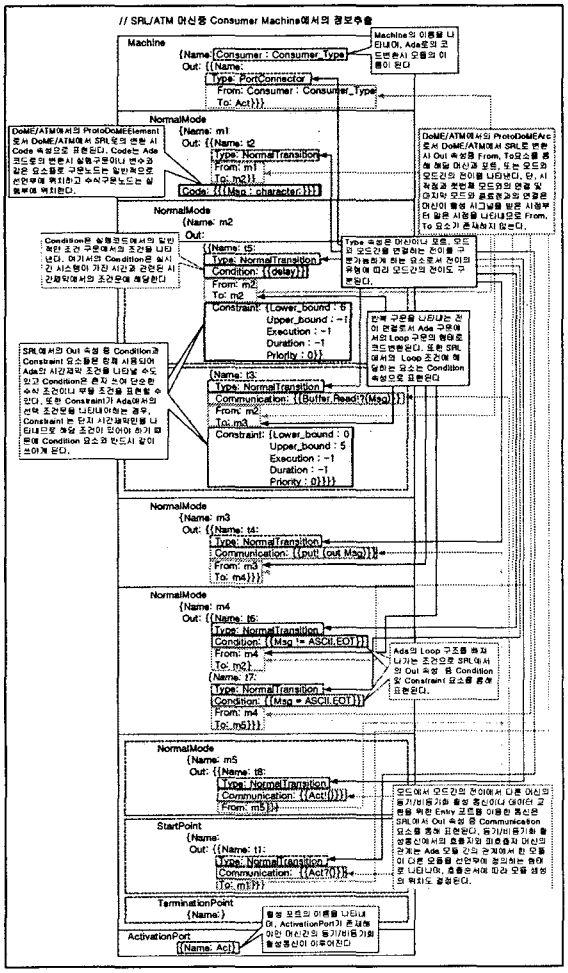
[표 2] 각 요소의 Out 속성에 대한 정보 추출

KeywordType	Ada 실행코드로 변환 시 KeywordType 및 속성 내용
Type	해당 요소의 속성값에 따라 선언구문 및 모듈의 생성 위치 결정 (속성값의 형태로 PortConnector, NormalTransition, AbstractTransition SubjectTransition 포함)
Name	전이의 이름을 나타내는 요소로서 다른 전이와의 구분을 위한 목적으로 사용 (실제 직접적인 Ada 구문으로 변환되지 않고, 해당 요소의 각 구문으로 변환시 구문간의 구별을 위한 참조 목적으로 사용)
From / To	통신 및 전이 이동에 따른 제어 흐름을 나타냄으로써 각 실행코드 구문들의 실행 우선 순위 및 생성 위치 결정
Communication	머신 간의 통신을 나타내는 이벤트 호출로서 모듈 간의 호출관계 표현
Condition	전이 이동에 따른 제어 흐름에 있어서의 조건을 나타내는 요소로서 실행코드에서 조건 또는 선택구문으로 표현 가능
Constraint	통신 및 전이 이동 시의 시간제약을 나타내는 요소로서 코드 상 단독으로 존재할 수 없으며, 반드시 Communication이나 Condition 속성과 함께 사용 (속성값의 형태로 Lower_bound, Upper_bound, Execution, Duration 포함)
Priority	전이의 우선순위를 기록하는 요소로서 실행코드 구문의 실행 순서결정

4.2 Ada 실행코드로의 변환을 위한 정보추출 예

SRL/ATM에서의 정보추출은 각 노드간의 관계 정보 및 노드의 속성 정보의 추출을 통해 이루어진다. SRL/ATM에서 구조 정보와 관계 정보에 대한 추출은 Ada 실행코드의 생성 위치에 있어 Main 모듈의 선언부에 종속적 모듈이 생성되어짐에 따라 SRL에서의 머신간 통신 및 이벤트 호출을 통해 형성되는 병행적 시행관계 및 활성 통신에 따른 개념

적 종속관계에 대한 정보추출이 필요하다. 이러한 관계 정보들은 실제 다른 머신을 호출하는 호출자 머신 내부에서의 전이 이동 시 시간적 흐름에 따른 호출을 통해 이루어진다. 이에 따라 전이 이동 시의 조건이나 시간제약 등과 같은 정보들에 대한 추출이 반드시 필요하고, 전이 이동에 있어서의 해당 두 모드, 즉 전이를 내보내는 모드와 전이가 들어가는 목적모드에 대한 정보 추출도 반드시 이루어져야 한다. 일반적으로 관계 정보 및 구문 정보들은 SRL/ATM에서 주로 전이의 표현과 관련된 노드의 Out 속성 정보의 추출을 통해 대부분 실행코드로 변환되어 진다. Communication 속성은 실제 실행코드에서의 모듈 간 호출과 관련된 요소로서 이에 대한 정보 추출을 통해 실행코드 의 임의의 모듈 내에서 다른 모듈에 대한 호출 구문으로 표현될 수 있다. 또한, Constraint 속성은 Condition 속성이나 Communication 속성에 대한 제약적 요소로서 조건 구문이나 모듈 호출 구문에서 함께 사용되어진다. 실제 실행코드에서의 실행문이나 예외를 제외한 일반적인 구문에 해당하는 모드내 Element의 정적 정보추출은 노드의 Code 속성에 대한 정보추출을 통해 이루어진다. 다음 [그림 5]는 SRL/ATM 에서의 이벤트 흐름에 따른 정보 추출에 대한 예이다.



[그림 5] SRL/ATM에서의 이벤트 흐름에 따른 정보 추출

5. 결론 및 향후 연구과제

본 논문은 순환공학 환경에서 실시간 시스템 개발 및 검증을 위한 실행코드 생성기 구현과정에서 ATM으로 명세된 실시간 시스템에 대한 정형명세와 명세분석을 통해 생성된 SRL/ATM 코드로부터 실행코드로의 변환을 위해 필요한 구조 및 요소간 관계, 작업 수행시의 동적 및 정적 실행 정보 등과 같은 관련된 정보들을 추출하는 방법을 제시하였다. SRL/ATM에서의 정보추출은 SRL 분석기와 SRL 파스트리 생성기를 이용하여 각 구성요소 간 통신이나 이벤트 호출 시 노드간 관계 정보에 따른 구조분석을 통해 구조관계 및 병행관계, 활성 통신의 순서에 따른 개념적 종속관계 등의 정보를 추출하였고, 제어흐름에 따른 실행코드의 실행 순서 및 코드생성 위치에 대한 정보를 추출하였다. 또한 각 구성요소들의 속성정보의 추출을 통해 실제 Ada 실행코드를 이루는 구문적 요소 정보들을 추출하였다. 이러한 추출 정보들을 기반으로 실제 Ada 실행코드와의 구조적 매핑, 즉 SRL/ATM 머신의 작업처리 특성에 따른 Ada 모듈로의 변환, 모드간을 연결하는 전이 형태에 따른 서브블록(반복, 조건, 선택 구문) 또는 일반적인 실행구문으로의 변환이 가능하다.

향후에는 이러한 정보추출 방법을 기반으로 ATM으로부터 구조 및 요소관계 정보, 제어 흐름에 따른 코드실행 우선순위와 코드 생성위치 정보 등을 자동으로 추출할 수 있는 SRL 분석기와 추출된 구문 정보들에 대한 실행코드 구문규칙을 적용할 수 있는 자동화 도구 개발이 요구된다. 또한 이러한 자동화 도구를 이용하여 실시간 시스템 개발 시 요구사항에 맞는 다양한 실행코드를 생성할 수 있는 언어 생성 시스템을 개발하기 위한 연구가 지속되어야 한다.

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] D. Harel, "*Statecharts: A Visual Formalism for Complex System*, *Science of Computer Programming*," Vol. 8, pp. 231-274, 1987

[4] S. jahanian and A. Mok, *Modechart : A Specification Language for Real Time Systems*, IBM Technical Report: RC 15140, November,1989

[5] 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

[6] Stuart Bennett, *Real-Time Computer Control – An introduction*, Prenticce Hall, 1994

[7] 노경주, 박지연, 이문근, "실시간 시스템의 순환공학을 위한 정형기법 : 추상시간기계," 한국정보과학회 학술발표논문집(A), 제27권, 제 1호, pp. 477-479, 2000

[8] 고현, 조상규, 이연식, "순환공학 환경에서의 실시간 시스템 개발 및 검증을 위한 코드 변환기 설계," 한국정보처리학회 학술발표논문집(상), 제8권, 제 1 호, pp. 193-196, 2001