

ASADAL : A Realtime System Specification and Analysis Environment

포항공과대학교 고광일* · 김대우** · 이관우* · 이지영** · 김영권** · 이정근* · 강교철***

● 목 차 ●

1. 서 론	2.4 ASADAL/PROVER
2. ASADAL 방법론	2.5 ASADAL/SIM
2.1 기반개념(Underlying Concepts)	3. ASADAL 방법론의 장점
2.2 ASADAL의 요구분석 활동	4. 결 론
2.3 ASADAL/SPEC	

1. 서 론

컴퓨터의 성능이 급속하게 발전함에 따라 컴퓨터의 응용분야 역시 그 영역을 빠르게 넓혀가고 있는데, 이런 응용분야는 일반적으로 대규모의 행동이 복잡한 실시간 컴퓨터 시스템을 요구하고 있다. 실시간 시스템이란 시스템 행동의 올바름이 기능적인 측면뿐만 아니라 시간적인 측면에 의해서도 결정되는 시스템으로 원자력 발전소 차폐기 제어 시스템, 기상 인공위성 제어 시스템, 미사일 제어 시스템, 그리고 교통 정보 시스템 등이 이에 속한다.

실시간 시스템은 자신의 행동에 대해서 엄격한 시간 제약을 갖고 있고, 일반적으로 규모가 방대하고 행동이 복잡하기 때문에 사용자 요구사항을 분석하기가 매우 어렵다. 이런 문제점은 주로 사용자와 분석자 간의 시스템을 바라보는 시각의 차이, 실시간 시스템이 만족해야 할 실시간 성질에 대한 검증의 어려움, 사용자가 실제 원하는 시스템 개발의 어려움 등에 기인한다.

본 논문에서는 위에서 언급한 실시간 시스템의

요구 분석 문제점을 해결하기 위하여 개발된 실시간 시스템 명세 및 분석 환경인 ASADAL을 소개한다.

ASADAL은 실시간 시스템 명세를 위한 ASADAL/SPEC, 실시간 시스템에 요구되는 실시간 성질의 정형적 검증을 위한 ASADAL/PROVER, 명세된 실시간 시스템의 행동과 기능을 시뮬레이션하기 위한 ASADAL/SIM, 그리고 실시간 시스템이 수행되는 환경을 프로토타이핑(prototyping)하기 위한 ASADAL/PROTO 등의 네가지 방법들로 구성되어 있다. 본 논문은 이들 네가지 방법 중에서 ASADAL/PROTO를 제외한 ASADAL/SPEC, ASADAL/PROVER, 그리고 ASADAL/SIM에 대한 내용을 다룬다.

본 논문의 구성은 다음과 같다. 이어지는 제 2절에서는 ASADAL 방법론을 기반개념, 요구 분석 활동, 그리고 ASADAL을 구성하는 (ASADAL/PROTO를 제외한) 세가지 방법을 개략적으로 설명함으로써 소개한다.

제 3절에서는 ASADAL의 장점을 간략하게 기술하고 마지막으로 제 4절에서는 본 논문의 요약과 향후 연구 방향을 소개하면서 결론을 내린다.

본 논문의 이해를 돕기 위한 예로서 Train

*학생회원
*비 회원
**중신회원

Gate System(TGS)을 사용한다. TGS는 건널목의 차단기를 제어하는 시스템으로 특히 안전성이 요구되는 실시간 시스템이다.

2. ASADAL 방법론

본 절에서는 ASADAL의 기반개념, 요구분석 활동, 그리고 ASADAL의 구성요소인 ASADAL/SPEC, ASADAL/PROVER, ASADAL/SIM 등을 차례로 설명함으로써 ASADAL의 개략적인 소개를 한다.

2.1 기반개념(Underlying Concepts)

ASADAL 방법론은 다음과 같은 개념들을 기반으로 하고 있다.

- 관심 분리(separation of concerns),
 - 하향식 명세(top-down specification),
 - 정형적 검증(formal verification),
 - 시뮬레이션(simulation),
 - 신속한 프로토타이핑(rapid prototyping).
- 이들 기반개념에 대한 설명은 다음과 같다.

일반적으로 한 시스템을 개발할 때, 개발하고자 하는 시스템을 다른 관점으로 보는 여러 단체(party)들이 참여하게 된다. 시스템의 요구를 분석할 때는 실제 시스템을 사용할 사용자와 시스템의 기능과 행동을 분석하는 분석자가 관여하게 되는데, 전자는 사용자와 직접적으로 관여하는 시스템의 외부적 행동에 관심이 있고 후자는 시스템의 외부적 행동을 가능케 하는 시스템의 내부적 기능과 행동에 관심이 있다. 이 이후로 사용자 중심의 시스템 관점과 분석자 중심의 시스템 관점을 각각 외부적 관점과 내부적 관점이라 부른다.

ASADAL은 각 관점에 적합한 명세언어를 제공한다. 즉, 외부적 관점에서 시스템을 명세하기 위한 '실시간 사건추적(Real-Time Event Trace)'와 내부적 관점에서 시스템을 명세하기 위한 '시간 강화 Statechart(Time Enriched Statechart)'와 'ASADAL 자료 흐름도(ASADAL Data Flow Diagram)'를 제공한다. 이 후로 실시간 사건추적과 시간 강화 Statechart 그리고 ASADAL 자료 흐름도는 각각 RTET, TES, 그리고 DFD라고 간략하게

부른다.

ASADAL은 대규모 시스템의 복잡성을 처리하기 위해서 모든 명세 언어에 하향식 명세 방법을 제공한다. 즉, RTET의 개체(entity), TES의 상태(state), DFD의 기능자(process)는 계층구조를 형성하면서 하향식 방식으로 상세화(refinement) 될 수 있는데, 이런 하향식 명세는 복잡한 시스템을 명세하는데 반드시 필요한 특징이다.

ASADAL의 두 관점간의 일관성 검증과 실시간 시스템이 만족해야 할 실시간 성질의 검증은 도달가능성 분석 또는 Real-Time Temporal Logic(RTTL) 논리 검증기 사용을 통해 이루어진다. RTTL 논리 검증기 사용을 위하여 ASADAL의 모든 명세언어의 의미는 RTTL의 정형성을 기반으로 설계되었으며, ASADAL 명세언어로 명세된 시스템의 행동은 모두 RTTL로 변환될 수 있다.

ASADAL은 명세된 실시간 시스템의 시뮬레이션을 통해서 시스템을 이해하고 추계적으로 분석한다. 사용자는 시뮬레이션을 통해 시스템을 수월하게 이해하고 분석자에 의해 분석된 시스템이 실제 사용자가 원하는 시스템인가에 대한 정확하고 신속한 검증을 하게 된다. ASADAL은 시뮬레이션 목적과 여건에 따라 대화형과 배치형 시뮬레이션 중에 하나를 선택할 수 있다.

ASADAL은 실시간 시스템이 수행될 환경

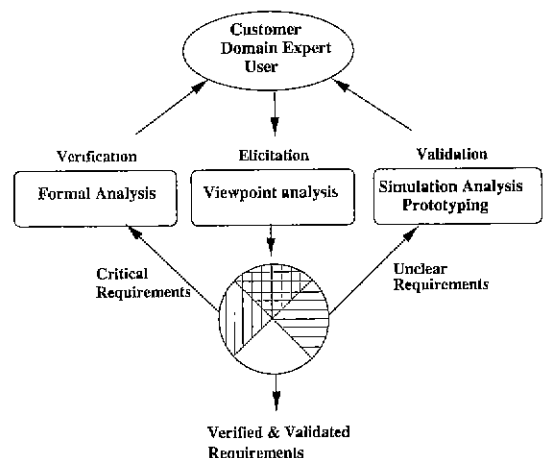


그림 1 ASADAL의 요구분석 활동

을 객체중심으로 모델링하고 그 객체들을 프로토타이핑 한다. 프로토타이핑된 객체들은 ASADAL의 명세언어로 명세된 실시간 시스템과 상호작용하면서 움직인다.

이어지는 절에서는 ASADAL의 요구분석 활동에 대해서 설명한다.

2.2 ASADAL의 요구분석 활동

ASADAL의 요구분석 활동은 그림 1에 나타난 바와 같이 요구사항 추출(elicitation), 정형적인 검증(formal verification), 그리고 유효성 검사(validation)로 나뉘어 지는데, 각각의 설명은 다음과 같다.

- 요구사항 추출 활동에는 의뢰자, 전문가, 그리고 분석자들이 참여하는데 이들 사이의 경험과 문제인식의 차이로 인해 요구사항 명세(specification)가 모호하거나 불완전해 진다. 이러한 문제를 해결하기 위해 ASADAL은 사용자와 분석자의 관점을 분리하여 각 관점에 적합한 방법으로 시스템의 요구사항을 추출한다.
- 정형적 검증 활동에서는 추출된 요구사항 중 반드시 검증되어야 하는 민감한 요구사항(critical requirement)에 대해서 수학적으로 검증한다. ASADAL은 정형적 검증방법으로 두가지 방법을 제공하고 있는데, 하나는 시스템 상태의 도달가능성 분석을 통한 검증이고 다른 하나는 RTTL 논리 검증기를 이용한 검증이다.
- 시뮬레이션과 프로토타이핑 활동에서는 불분명한 요구사항에 대해서 그 유효성을 검사한다. ASADAL은 시스템 행동을 동적으로 시각화함으로써 시스템에 대한 사용자의 유효성 검사를 가능하게 한다.

ASADAL에서 실시간 시스템의 요구사항을 분석하는 과정은 위의 세가지 활동을 의미있는 순서로 반복적으로 수행하는 것이다. 즉, 요구사항 추출 활동에서 얻어진 사용자 요구사항은 민감한 요구사항, 불분명한 요구사항, 그리고 증명, 검증된 요구사항 등으로 분류된다. 그 중에 민감한 요구사항은 정형적 분석 활동에서 수학적으로 검증받게 되고, 불분명한 요구사항은 시뮬레이션이나 프로토타이핑을 통해서 사

용자에게 보여지게 된다. 이 때, 정형적 검증을 통해 오류가 없음이 검증된 요구사항과 시뮬레이션과 프로토타이핑을 통해 사용자의 동의를 얻은 요구사항은 증명, 검증된 요구사항으로 구분되고, 그렇지 못한 요구사항에 대해서는 발견된 오류를 바탕으로 다시 요구사항 추출 활동을 거치게 된다.

이어지는 절들에서는 ASADAL/SPEC, ASADAL/PROVER, 그리고 ASADAL/SIM에 대해서 차례로 설명한다.

2.3 ASADAL/SPEC

ASADAL/SPEC은 실시간 시스템의 명세를 위한 방법으로 사용자 중심의 외부적 관점과 분석자 중심의 내부적 관점 등의 두가지 관점으로 시스템을 명세한다. 효과적인 관점분리를 위하여 ASADAL/SPEC은 각 관점에 적합한 명세언어를 제공하는데, 외부적 관점을 위해서는 시나리오 명세언어인 RTET를 제공하고 내부적 관점을 위해서는 시스템 행동 명세언어인 TES와 시스템 기능 명세언어인 DFD를 제공한다.

2.3.1 실시간 사건추적(Real-Time Event Trace)

RTET는 외부적 관점에서 보여지는 실시간 시스템의 한 행동을 그 실시간 시스템과 외부 환경에 존재하는 개체들 사이에 전달되는 메시지나 사건을 사용하여 시나리오로 표현한다. RTET는 Rumbaugh[7]의 '사건추적(event trace)'을 기반으로 하고 추가적으로 시스템의 복잡성을 다루기 위한 몇 개의 유용한 특징을 갖고 있다.

실시간 시스템은 보통 많은 메세지와 사건들이 외부 개체와 상호전달되는 내장형(embed-

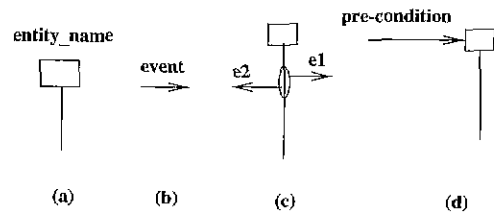


그림 2 RTET의 기본구조

ded) 시스템이기 때문에, 시스템과 외부 개체들 간에 전달되는 일련의 메시지와 사건의 나열의 모습으로 시스템의 외부 행동을 쉽게 표현할 수 있다. 이런 시나리오 기술 방식은 분석자가 사용자의 시스템 요구사항을 표현하고 서로 대화하는 데 매우 효과적이라고 밝혀져 있다[2]. 몇개의 ASADAL의 사례연구를 통해서도 시나리오 기반의 접근방식이 시스템에 대한 사용자의 요구사항 표현과 대화에 효과적이라는 것을 경험할 수 있었다.

그림 2는 RTET의 기본구조를 보여준다. 그림 2의 (a), (b), 그리고 (c)는 각각 개체, 사건 및 메시지 전달자, 동시사건 연결자, 그리고 사전조건(precondition)에 대한 명세를 보여준다. 이들 기본구조에 대한 간략한 설명은 다음과 같다.

- RTET는 추상적 명세의 하향식 상세화(refinement)를 지원한다. RTET 명세의 개체는 그림 3에서 보여지듯이 그 개체를 이루는 여러개의 딸림개체들과 그 딸림개체들 간에 전달되는 사건 및 메시지들로 상세화될 수 있다. 그림 3의 개체 B는 딸림개체 B-1과 B-2로 상세화되었다.
- RTET는 명세를 동시에 발생하는 사건들을 표현하기 위해서 타원 형태의 기호를 제공한다.
- RTET는 시나리오의 초기상태를 표현하는 사전조건(precondition)을 가지고 있다.
- RTET의 두 메시지나 사건 사이의 양적 시간 제약은 RTTL을 사용하여 명세할 수 있는데, 이를 위해서 ASADAL은 함수형태의 '실시간 성질 명세 기본자'들을 제공한다. 그림 4는 RTET 명세와 결합

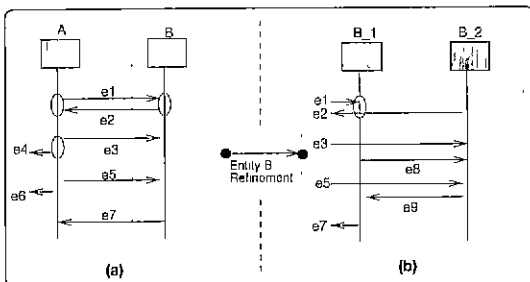


그림 3 RTET의 계층구조를 보여주는 간단한 예

Real-Time Requirement Specification Primitives in Functional Form	Definition in RTTL
jst-dist(e1, e2, tm)	$(e1 \wedge t < T) \wedge \diamond (e2 \wedge t - T < tm)$
max-dist(e1, e2, tm)	$(e1 \wedge t < T) \wedge \diamond (e2 \wedge t < T + tm)$
min-dist(e1, e2, tm)	$(e1 \wedge t < T) \wedge \neg \diamond (e2 \wedge t < T + tm)$

그림 4 RTET의 실시간 성질 명세 기본자들과 각 기본자의 RTTL로 명세된 정의

되어 사용되는 실시간 성질 명세 기본자들과 RTTL로 명세된 정의를 보여준다.

그림 4에 나와있는 실시간 성질 명세 기본자들에 대한 설명은 다음과 같다.

- jst-dist(e1,e2,tm) : 이 기본자는 사건 e1과 e2 사이에 시간적 거리가 정확히 tm임을 나타낸다.
- max-dist(e1,e2,tm) : 이 기본자는 사건 e1과 e2 사이에 시간적 거리가 최대 tm임을 나타낸다.
- min-dist(e1,e2,tm) : 이 기본자는 사건 e1과 e2 사이에 시간적 거리가 적어도 tm임을 나타낸다.

그림 3(a)의 max-dist(e5,e6,3sec)는 메시지 혹은 사건 e5가 개체 A로부터 B로 전달된 후 적어도 3초안에 개체 B가 개체 A로 메시지 혹은 사건 e6를 발생시켜야 한다는 것을 나타내고 있다.

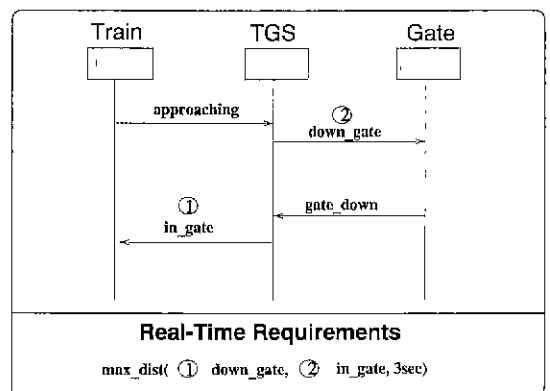


그림 5 TGS의 외부행동에 대한 RTET 명세

그림 5는 RTET를 사용하여 TGS의 한 시나리오를 명세한 것을 보여준다.

2.3.2 시간 강화 Statechart(Time Enriched Statechart)

시간 강화 Statechart(TES)는 내부적 관점 분석을 위한 명세 언어로 실시간 시스템의 내부 행동을 명세한다.

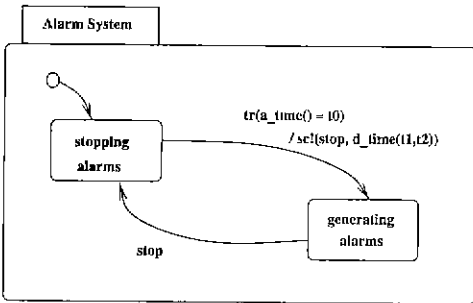


그림 6 절대적 시간과 기간적 시간의 개념을 이용한 자명종 시스템

TES는 STATEMATE의 Statechart[1, 3]를 기반으로 하면서 실시간 시스템의 보다 다양한 시제행동(temporal behavior)을 표현하고 DFD와 강력한 결합을 지원한다.

TES의 주요 특징들은 다음과 같다.

- TES는 실시간 시스템의 다양한 시제행동을 표현하기 위해서 여러가지 시간 기본자들을 제공한다. 예를 들어, 실시간 시스템의 행동을 명세하는 할 때 Statechart는 상대적이고 순간적인 시간만을 제공하는데 반해 TES는 a_time()과 d_time()과 같은 시간 기본자를 제

공함으로써 절대적 시간과 기간적 시간의 개념도 추가적으로 제공한다.

그림 6에 나와있는 TES 다이어그램은 a_time()과 d_time()을 사용함으로써 시스템 행동명세에 절대적 시간과 기간적 시간의 개념을 도입하고 있다. 이 그림의 자명종 시스템은 시스템 시계가 t0의 값을 갖을 때 경종이 시작되는데 경종이 시작한 시각으로부터 t1, t2 사이에 존재하는 어느 한 시간이 경과하면 경종을 멈춘다.

- TES 명세가 만족해야 할 실시간 성질을 RTTL을 사용하여 명세할 수 있다. 분석자가 시스템의 내부행동을 TES로 명세할 때, 요구되는 실시간 성질은 반드시 TES 명세에 반영되어야 한다. 그런데 시스템의 행동이 복잡하면 TES 명세에 요구되는 모든 실시간 성질이 반영되었는지 알아내기가 무척 어렵게 된다. TES는 이런 문제를 해결하기 위해서, TES 명세가 반드시 만족해야 할 실시간 성질을 RTTL로 의식적으로 명세하는 방법을 제공한다. 그리고 TES 명세는 이런 RTTL로 명세된 실시간 성질에 대해서 그 만족성이 검증되어야 하는데 ASADAL은 이를 정형적으로 검증하는 방법을 제공한다. ASADAL은 TES의 실시간 성질을 명세할 때 RTTL 사용을 보다 편리하게 하기 위해서 함수형태의 '실시간 성질 명세 기본자'를 제공한다. 이들 기본자들은 모두 RTTL로 그 의미가 정형적으로 정의되어 있다. 그림 7은 TES 명세와 결합되어 사용되는 실시간 성질 명세 기본자들과 그 기본자들의 RTTL로 명세된 정의를 보여준다.

그림 7의 각 기본자들의 설명은 다음과 같다.

- safety(F1,F2): 이 기본자는 상태공식(state formula) F1이 만족되면 항상 F2가 만족되어야 함을 나타내는 안전성(safety)에 관한 실시간 성질이다.
- liveness(F1,F2): 이 기본자는 상태공식 F1이 만족되면 언젠가는 F2가 반드시 만족되어야 함을 나타내는 생존성(liveness)에 관한 실시간 성질이다.
- RT_rsp(F1,F2,tm): 이 기본자는 상태공식 F1이 만족되면 시간 tm 안에 F2가

Real-Time Requirement Specification Primitives in Functional Form	Definition in RTTL
safety(F1, F2)	$F1 \rightarrow [] r2$
liveness(F1, F2)	$F1 \rightarrow \diamond r2$
RT_rsp(F1, F2, tm)	$(F1 \wedge t = T) \rightarrow \diamond (F2 \wedge t < T + tm)$
persistency(F1, F2, F3)	$F1 \rightarrow (F2 \vee F3)$

그림 7 TES의 실시간 성질 명세 기본자들과 각 기본자의 RTTL로 명세된 정의

만족되어야 함을 나타내는 실시간 반응성 (real-time response)에 관한 실시간 성질이다.

- persistency(F1,F2,F3): 이 기본자는 상태공식 F1이 만족되면 F3가 만족될 때까지 항상 F2가 만족되어야함을 나타내는 지속성에 관한 실시간 성질이다. 이때, 상태공식 F3는 영원히 만족되지 않을 수도 있는데 이런 경우에는 F2가 영원히 만족되어야 한다.

그림 8은 TES를 사용하여 TGS의 내부행동을 명세한 것이다. 그림 하단부에 safe() 기본자로 명세되어 있는 실시간 성질은 기차가 전널목에 있을 때는 차단기가 항상 내려와 있어야 한다는 안전성에 관한 것이다. 기본자의 인자로 사용된 excl(F1,F2)는 두 상태공식 F1과 F2가 동시에 참값을 갖을 수 없다는 것을 나타내는 서술어(predicate)이고 at(S)는 시스템이 상태 S에 존재하면 참값을 갖고 그 이외에는 거짓값을 갖는 서술어이다.

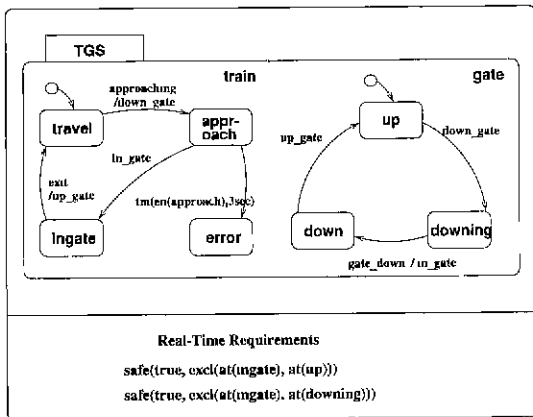


그림 8 TGS의 내부행동에 대한 TES 명세

2.3.3 ASADAL 자료흐름도(Data Flow Diagram)

ASADAL 자료흐름도(DFD)는 STATE-MATE의 Activity-Chart[1]를 기반으로 하는데 의식의 추적성을 높이기 위한 ‘항구연결자(port connector)’를 제공하는 것 이외에는 Activity-Chart와 동일하다. 항구연결자는 기능자(process)가 기능분해(functional decomposition)에 의해서 상세화되어질 때, 기능분해

되어 지는 기능자에 연결된 정보흐름자(information flow)의 위치를 다음 단계의 DFD에 알려주는 역할을 한다. 이런 항구연결자의 존재는 한 기능자와 그 기능자를 기능분해하여 생성된 DFD 사이에 의식의 추적성(traceability)을 높여준다.

그림 9는 ASADAL 자료 흐름도로 TGS의 내부기능을 명세한 것이다.

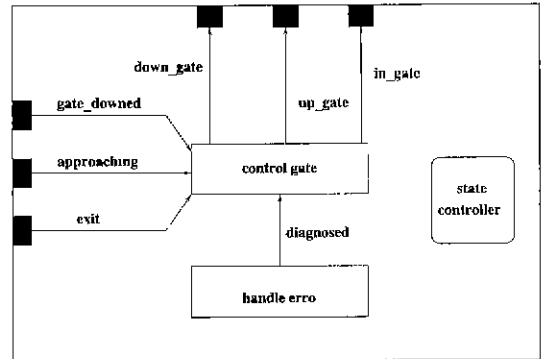


그림 9 TGS의 내부기능에 대한 DFD 명세

2.4 ASADAL/PROVER

ASADAL은 명세분석을 위해서 두가지 행위를 갖는다. 하나는 TES 명세가 자신에게 요구되는 실시간 성질을 만족하는지 검증하는 것이고, 다른 하나는 외부적 관점에서 명세된 시스템의 외부행동과 내부적 관점에서 명세된 시스템의 내부행동 및 기능 간의 일관성을 검증하는 것이다.

ASADAL/PROVER는 명세분석을 시스템 상태의 도달가능성을 분석하는 방법과 RTTL 논리 검증기를 이용한 방법 등의 두가지 방법으로 수행한다.

● 도달가능성 분석방법은 내부적 관점에서 명세된 시스템의 내부행동(TES)과 기능(DFD)을 분석하여 시스템이 도달할 수 있는 모든 상태를 나타내는 도달가능성 그래프를 생성하고 이를 바탕으로 몇가지 정형적으로 검증된 알고리즘을 실행시켜 실시간 성질 및 RTET 간의 일관성을 검증한다. 이 방법은 검증과정에서 오류가 발생하면 그 경우를 분석자에게 이해하기 쉬운 방법으로 정확히 보여줄 수 있으므로 시스템의 오류 이해와 수정에 도움을 준다.

그림 10은 이 방법의 개략적 모습을 보여준다.

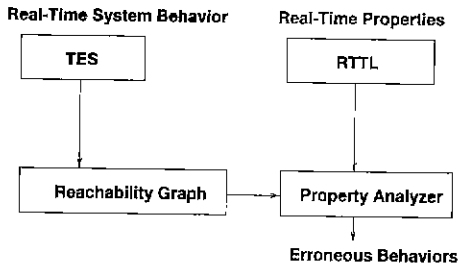


그림 10 도달가능성 분석을 이용한 ASADAL/PROVER의 명세분석 방법

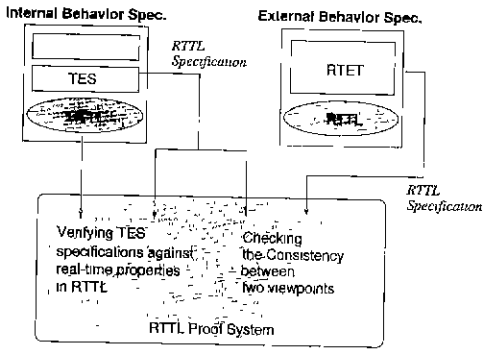


그림 11 RTTL 논리 검증기를 이용한 ASADAL/PROVER의 명세분석 방법

●RTTL 논리 검증기를 이용하는 방법은 그림 11에 보여지듯이 ASADAL/SPEC에 의해서 생성된 RTET, TES, DFD 명세를 모두 RTTL로 변환하여, 이 RTTL 명세를 대상으로 RTTL 논리 검증기를 실행한다. 이 때, RTTL 명세와 TES 명세의 실시간 성질 명세 기본자는 이미 그 의미가 RTTL로 정의되어 있기 때문에 특별히 RTTL로 변환시키는 과정을 거치지 않는다.

시스템의 규모가 너무 크고 복잡한 경우, 위에서 설명한 도달가능성 분석 방법은 공간 폭발(space explosion) 문제를 갖고 있어 적용하기가 어렵다. 이런 경우 RTTL 논리 검증기 이용 방법을 사용하면 공간 폭발 문제없이 명세를 분석할 수 있다.

그림 11은 RTTL 논리 검증기를 이용하여 명세분석을 하는 방법의 개략적 모습을 보여준다.

2.5 ASADAL/SIM

ASADAL의 요구사항 명세는 근원적으로

수행되기에 부적합한 면이 존재한다. 요구사항 명세가 수행되기 힘든 요인은 다음과 같다. 첫째, 요구사항 명세에는 시스템 내부의 행동만이 명세될 뿐 시스템과 상호작용하는 외부 개체의 행동은 명세되지 않는다. 둘째, 시스템의 기능 분해(functional decomposition)서 가장 하위 단계에 존재하는 기본 기능자(primitive process)의 명세는 비정형적인 언어로 된 경우가 많다.

이러한 문제를 해결하기 위한 방안으로 ASADAL은 시뮬레이션 제어 언어(Simulation Control Language)를 제공하고 이 시뮬레이션 제어 언어를 이용하여 외부 개체의 행동과 내부 기본 기능자의 행동을 명세하고 시뮬레이션할 수 있도록한다. 시뮬레이션 제어 언어 및 시뮬레이션 환경과 이어지는 ASADAL/SIM의 특징에 대한 자세한 설명은 [5]를 참조하기 바란다.

다음은 ASADAL/SIM의 특징에 대한 간략한 설명이다.

●시뮬레이션은 대화형 모드 혹은 배치형 모드로 이루어질 수 있다. 대화형 모드 시뮬레이션은 분석자가 모델의 행동을 단계별로 관찰하고자 할 때 유용하다. 대화형 모드 시뮬레이션에서 사용자는 시뮬레이션 진행 명령과 시간 진행 명령을 이용하여 시뮬레이션을 진행시키고 외부 데이터를 발생시키거나 예기치 않은 상황을 맞이 하였을 때 시스템 환경 변수의 값을 바꾸는 등의 활동을 통해 명세의 잘못된 부분을 수정하거나 시스템의 행동을 단계별로 자세히 분석할 수 있다. 배치형 모드 시뮬레이션은 대량의 데이터를 다루는 시나리오 중심의 시뮬레이션이나 추계적인 시뮬레이션(stochastic simulation)을 할 때 유용하다. 배치형 모드 시뮬레이션에서는 사용자가 일일이 시뮬레이션에 관여하지 않고 특정한 상황이 발생하였을 때만 사용자가 개입하고 나머지 부분은 시뮬레이션 드라이버에 그 역할을 넘겨주게 되어 대량의 복잡한 시뮬레이션을 가능케 한다. 시뮬레이션 드라이버는 외부 데이터를 확률적인 분포로 발생시키거나 시뮬레이션 도중에 일어나는 특정한 상황에 시뮬레이션을 멈추고 이를 해결할 수 있도록 멈춤점(breakpoint)을 설정

하고 처리하는 역할을 한다.

● 시뮬레이션을 통해 얻어진 데이터는 확률적인 분석이나 입, 출력 분석[9] 등과 같은 분석에 사용되어 진다. 가장 기본적인 분석 방법은 데이터를 통계 패키지를 사용하여 시각적으로 보여주고 이를 바탕으로 추리하는 것이다 [10]. 가령, 출력 데이터가 시간이 지남에 따라 진동폭이 줄어드는 모습을 보인다면 얼마 진동후에 안정화된 상태(steady state)에 도달할 것이라는 것을 추론할 수 있다.

● 시뮬레이션으로 분석된 결과를 바탕으로 실시간 시뮬레이션을 시각적으로 보여줄 수 있다. 실시간 시뮬레이션은 사용자가 실제 세계에서 실제 시스템과 상호 작용하는 것과 같이 시스템을 운영함으로써 사용자는 시스템의 행동이 그들의 요구사항에 충분히 부합하는가를 평가할 수 있고 최종 시스템의 모습을 보고 느낄 수 있게 된다.

다음에 이어지는 절에서는 지금까지 설명한 ASADAL의 기반개념 및 특징에 의해서 유도되는 ASADAL의 장점에 대해서 간단하게 설명한다.

3. ASADAL 방법론의 장점

ASADAL의 장점은 다음과 같은 범주로 나뉘질 수 있다.

- 사용자 요구의 의도적인 분석과 명세,
- 정형적인 분석,
- 하향식 상세화 방식,
- 빠른 유효성 검사.

이들 장점에 대한 각각의 설명은 다음과 같다.

ASADAL은 RTET를 사용하여 사용자의 요구를 의도적으로 포착한다. 이렇게 의도적으로 명세된 사용자 요구는 시스템의 내부적 행동 및 기능을 명세하는 작업의 기반으로 사용될 수 있으며, 시스템의 ASADAL/PROVER를 이용한 정형적인 유효성 검증의 대상으로도 사용될 수 있다.

ASADAL의 명세분석은 도달가능성 분석과 RTTL 논리 검증기를 이용하여 정형적으로 수행된다. 이는 요즘과 같이 실시간 시스템 분석

시 실시간 성질에 대한 정확한 검증이 절실히 요구되는 상황에 적합하다고 할 수 있다. 또한, ASADAL/PROVER는 도달가능성 분석과 논리 검증기 방법을 둘 다 지원함으로써, 문제의 성격에 따라 적합한 방법을 분석자가 선택할 수 있다.

ASADAL은 대규모의 복잡한 실시간 시스템 분석의 어려움을 처리하기 위해서 모든 명세 언어에 하향식 명세방법을 지원한다. 즉, RTET는 개체의 계층성을, TES는 상태의 계층성을, 그리고 DFD는 기능자의 계층성을 지원한다.

ASADAL은 ASADAL/SIM을 통하여 사용자가 설계나 구현 전에 최종 시스템의 모습을 보고 느낄 수 있으므로 빠른 유효성 검사를 받을 수 있다. 이러한 방법을 사용하면 사용자가 요구사항 명세 단계에 적극적으로 참여할 수 있어 요구사항의 추출(elicitation)과 유효성 검증 사이의 간격을 최소로 줄일 수 있고 많은 선택 사항을 경제적으로 평가할 수 있다.

4. 결 론

본 논문은 대규모의 실시간 시스템의 분석을 목적으로 하는 새로운 실시간 시스템 요구분석 방법론인 ASADAL을 소개하였다. ASADAL의 주요 특징은 다음과 같다.

- 다양한 시간행동을 명세하기 위한 시간 기본자와 편리한 실시간 성질 명세를 위한 실시간 성질 명세 기본자를 제공,
- 사용자 중심의 외부적 관점과 분석자 중심의 내부적 관점의 분리,
- 실시간 시스템의 실시간 성질과 두 관점간의 일관성에 대한 정형적 검증,
- 실시간 시스템의 복잡성을 처리하기 위한 하향식 명세,
- 시스템 개발 초기 단계에서 시스템의 유효성 검사.

ASADAL 방법론은 SafeHome System[6], Cruise Control System[8], 그리고 MICOM application[4] 과 같은 실시간 시스템을 통해서 그 유용성과 효율성을 검사받았다. 이들 사례연구를 통해서, 특히 RTET가 시스템의 외

부행동을 명세하고 다른 사람들과 명세된 시스템의 외부행동에 대해서 대화하는데 매우 효율적이라는 것이 밝혀졌다. 그리고 함수형태의 실시간 성질 명세 기본자가 RTTL에 익숙하지 않는 사람들이 실시간 성질을 명세하는 데 매우 큰 도움을 준다는 것도 알 수 있었다.

현재 ASADAL/SPEC과 ASADAL/SIM, 그리고 ASADAL/PROVER는 상당부분 CASE 도구로 구현이 되어 외부에 여러차례 발표되었다[11, 12].

참고문헌

[1] "Statemate 4.5 User Reference Manual", i-Logix Inc., Burlington. MA, Aug. 1992.

[2] Davis, A., et al., "RLP: An automated tool for the processing of requirements", The Proceedings of IEEE COMPSAC '79, pp. 289-299.

[3] Harel, D., "On visual formalisms", Commun. ACM, vol. 31, no. 5, pp. 514-530, 1980.

[4] Jang, S.C., Kang, M.S., and Kang, K.C., "Application of Software Engineering Techniques for the Improvement of MICOM Software Development Process", POSTECH/CS/SE-93-TR-4, Software Engineering Lab., Computer Science, Postech, 1993.

[5] Lee, K.W. and Kang, K.C., "ASADAL: A Method for Real-time System Behavior Analysis Through Specification Execution", Master Thesis of Computer Science, Postech.

[6] Pressman, S.R., "Software Engineering: A Practitioner's Approach", McGraw-Hill, Inc, 1992.

[7] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W., "Object-Oriented Modeling and Design", Prentice-Hall, Inc, 1991.

[8] Smith, L.S. and Gerhart, L.S., "STATEMATE and cruise control: A case study", The Proceedings of COMPAC

'88, 1988.

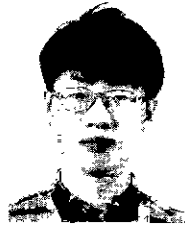
[9] Stevart V. Hoover, Ronald F. Perry, "SIMULATION: A Problem-Solving Approach", Addison Wesley, 1990.

[10] Paul A. Fishwick, "Simulation Model Design and Execution" Building Digital Worlds, Prentice Hall, 1995.

[11] "96 한국형 CASE Workshop", 포항공과대학교 소프트웨어기술연구센터, 1995.

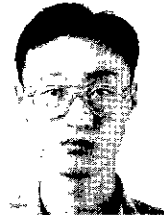
[12] 18th International Conference on Software Engineering, Berlin, Germany, 25-29 Mar. 1996.

고 광 일



1993 포항공과대학교(전자계산학과 학사)
 1995 포항공과대학교(전자계산학과 석사)
 1995~현재 포항공과대학교 전자계산학과 박사과정
 관심분야: 소프트웨어 공학, 실시간 시스템 분석 방법론

김 대 우



1995 포항공과대학교(전자계산학과 학사)
 1995~현재 포항공과대학교 정보통신과 석사과정
 관심분야: 소프트웨어 공학, 실시간 시스템 분석 방법론

이 관 우



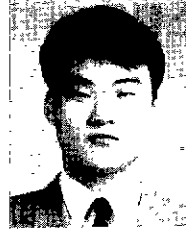
1994 포항공과대학교(전자계산학과 학사)
 1996 포항공과대학교(전자계산학과 석사)
 1996~현재 포항공과대학교 전자계산학과 박사과정
 관심분야: 소프트웨어 공학, 소프트웨어 제사용

이 지 영



1995 포항공과대학교 (전자계산
학과 학사)
1995~현재 포항공과대학교 전
자계산학과 석사과
정
관심분야: 소프트웨어 공학, 실시
간 시스템 분석 방법론

이 정 근



1991 서울대학교 (계산통계학과
학사)
1993 포항공과대학교 (정보통신
과 석사)
1993~현재 포항공과대학교 연
구원
관심분야: 소프트웨어 공학

김 영 권



1996 포항공과대학교 (전자계산
학과 학사)
1996~현재 포항공과대학교 전
자계산학과 석사과
정
관심분야: 소프트웨어 공학, 실시
간 시스템 분석 방법론

강 교 철



1973 고려대학교 (통계학 학사)
1976 Univ. of Colorado (산업공
학과 석사)
1982 Univ. of Michigan (소프
트웨어공학 박사)
1984 Univ. of Michigan (객원
교수)
1987 AT & T Bell Labs (연구
원)
1992 Carnegie Mellon Univ.
(선임 연구원, Project
leader)
1992~현재 포항공과대학교 전자계산학과 부교수 재직
1994~현재 포항공과대학교 전자계산소장
관심분야: 소프트웨어 재사용, 요구분석 공학, CASE, 영역
분석, 소프트웨어 공학 환경

● Call for Papers ●

- 행사명 : High Performance Computing ASIA '97
- 행사일자 : 1997년 4월 21일~25일
- 대회장소 : Hotel Lotte World
- 논문마감 : 1996년 11월 15일
- 주 최 : 한국정보과학회 · 시스템공학연구소
- FOR FURTHER INFORMATION, PLEASE CONTACT :

HEADQUARTERS :

Mr. Joong Kwon Kim
Supercomputer Center
Systems Engineering Research Institute
P.O. Box 1, Yoosung-gu, Taejon 305-600, Korea
Phone : +82-42-869-1997 Fax : +82-42-869-1399
E-mail : hpc97@seri.re.kr
WWW : http://www.seri.re.kr/HPC97.html
FTP : ftp.seri.re.kr(cd/pub/hpc97)

SECRETARIAT :

INTERCOM Convention Services, Inc.
4Fl. Jisung Bldg., #645-20 Yoksam 1-dong
Kangnam-gu Seoul 135-081, Korea
Phone : +82-2-501-7065 / 566-6339
Fax : +82-2-565-2434 / 3452-7292
E-mail : intercom@soback.kornet.nm.kr
intercom@seri.re.kr