

이동 프로세스 대수에서 행위의 의미 분석을 위한 정형기법

우수정^o 온진호 이문근

전북대학교 컴퓨터공학과

wpig04@gmail.com, jjinghott@gmail.com, moonkun@jbnu.ac.kr

Formal Method for Meaning Analysis of Behavior on Mobile Process Algebra

Sujeong U^o Jinho On Moonkun Lee

Department of Computer Engineering, Chonbuk University

요 약

분산 이동 실시간 시스템의 명세, 개발 및 검증을 위해 π -calculus, bigraph, Mobile Ambient, CARDMI 등의 정형기법이 존재한다. 이러한 정형기법은 이동하는 에이전트 혹은 프로세스에 대한 명세 및 시스템의 안전성과 검증에 대한 분석을 지원하지만, 행위의 의미적 관점에서 분석 및 검증 방법을 제시하고 있지 않다. 본 논문에서는 정형기법으로 명세 된 시스템의 실행 데이터인 원시 데이터를 행위의 의미적 관점에서 시스템을 분석 및 검증이 가능한 Prism Analyzer를 제안한다. 제안된 Prism Analyzer는 특정 시스템에 대해 발생할 수 있는 다양한 행위를 온톨로지와 속성문법으로 정규화한 다양한 행위모델을 지닌다. 이러한 Prism Analyzer는 원시 데이터를 행위의 의미적 측면에서 개별적, 연속적, 복합적으로 분석 및 검증이 가능하고, Prism Analyzer에 정의된 다양한 행위 모델을 바탕으로 동일한 원시 데이터에 대해 행위 모델에 따른 다양한 분석 결과를 도출해 낼 수 있다.

1. 서론

분산 이동 실시간 시스템의 명세, 개발 및 검증을 위해 π -calculus[1], Mobile UNITY[2], bigraph[3], Mobile Ambient[4], CARDMI[5,6] 등의 정형기법이 존재한다. 이 중 CARDMI는 시스템을 이동, 통신과 같은 행위에 대해 구조적으로 시간과 공간 차원의 정보로 구분하여 시스템에 대한 복잡도를 줄임으로써 시스템에 대한 명세, 분석 및 검증에 대한 이해를 높이기 위해 새로 고안된 프로세스 대수 기반 정형기법이다. 이러한 정형기법은 이동하는 에이전트 혹은 프로세스에 대한 명세 및 시스템의 안전성과 검증에 대한 분석을 지원하지만, 행위의 의미적 관점에서의 분석 및 검증 방법을 제시하고 있지 않다. 본 논문에서는 원시데이터로 CARDMI로 명세 된 시스템이 실행단계에서 공간과 시간의 흐름에 따라 행위를 분석 및 검증할 수 있는 *Timed Action Graph*(TAG)[6]를 사용하여 행위의 의미적 관점에서 시스템을 분석 및 검증이 가능한 *Prism Analyzer*를 제안한다.

TAG와 같이 원시 데이터로 명세 된 시스템은 방대하고 복잡할수록 이를 분석하고 검증하는데 어려움이 있다. 이러한 원시 데이터를 의미적 행위

데이터로 분석 및 검증하는데 있어 *Prism Analyzer*는 특정 시스템에 대한 행위를 기본 행위부터 복합적 행위까지 발생 가능한 행위들을 추상화하여 정의한 행위모델인 Lattice를 가지고 있다. 본 논문에서는 행위 모델로 이동체가 자원을 핸들러에 할당하는 행위를 지닌 *Carrier Resource Allocation*(CRA)[7]를 사용한다. 이와 같은 행위 모델은 *Prism Analyzer*에서 온톨로지와 행위 모델을 속성문법[8]으로 정의한 *Attributed Prism Analyzer*(APA)로 정의된다. 온톨로지는 TAG의 원시 데이터를 바탕으로 시스템의 구조적 관계를 알아보기 위함이고, APA는 행위 모델을 APA Syntax에 맞게 속성문법으로 정규화하여 정의한 것이다. 따라서 *Prism Analyzer*는 TAG의 원시 데이터를 행위의 의미적 측면에서 개별적, 연속적, 복합적으로 분석 및 검증이 가능하고, Prism Analyzer에 정의된 다양한 행위 모델을 바탕으로 동일한 TAG에 대해 행위 모델에 따른 다양한 분석 결과를 도출해 낼 수 있다.

2. 관련연구

분산 이동 실시간 시스템에서 명세, 개발 및 검증을 위해 π -calculus, Mobile UNITY, bigraph, Mobile Ambient 등의 정형기법이 존재한다. 이러한 정형기법은 검증을 위해 *Model checking*이나 *Automated theorem proving* 등의 방법을 사용하고 있다. *Model checking*은 특정 시스템의 안전성이나 설계상 오류가 없음을 검증하는 것으로 시스템의 흐름이 설계한대로 올바르게

본 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-521-D00451)
이 연구에 참여한 연구자는 2단계 BK21사업의 지원비를 받았음

작동하는지를 논리적으로 검사하는 것이다. 대표적인 틀로 SPIN[9]이 존재하며, 이는 데이터 통신 프로토콜과 같은 동시성 시스템의 논리적인 일관성을 분석하는 도구이다. *Automated theorem proving*은 *automated reasoning*(AR)로 잘 알려져 있으며, 컴퓨터 프로그램에서 수학 이론을 바탕으로 검증하는 방법이다. 위와 같이 정형기법에서는 명세 된 시스템을 분석하기 위한 방법으로 행위에 대해 의미를 적용하여 분석하는 방법은 제시되지 않고 있다.

그 외 본 논문에서 제시한 분석방법과 비슷한 방법으로 데이터마이닝 분야에서 자주 활용되고 있다. 예를 들어 [10]에서는 사용자의 웹 로그를 가지고 사용자의 행위를 분석하여 각 사용자들의 웹 사용 패턴을 자동으로 개인화 할 수 있는 방법을 제안하였다. 이를 위해 데이터 분석 기술로 순서화된 Lattice 이론을 바탕으로 하며, 개개의 도메인에서 관계와 속성을 재구성한 concept lattice를 사용하였다. 이는 인식을 위한 데이터가 데이터 자체로 각 데이터들이 지니는 패턴을 통해 데이터들이 내포하고 있는 지식을 추론하기 위해 사용된다. 따라서 본 논문과 같이 행위의 의미를 추론하여 다양한 분석결과를 도출해내는 방법으로 적용될 수 없다.

3. Carrier Resource Allocation (CRA)

본 논문에서 제안하고 있는 Prism Analyzer의 효율성을 보이기 위해 CRA 알고리즘을 예로 들면서 사용한다.

CRA란, Prism Analyzer에서 사용되는 행위 모델 중 하나로 행위의 의미적 측면에서 시스템을 분석 및 검증하는 것으로 *Carrier*(이동체)가 공간상에 있는 *Resource*(자원)를 소스에서 목적지까지 *Handler*(핸들러)에 할당하는 행위를 지닌다. 각 이동체, 자원, 핸들러에는 Capacity가 적용되며, 적용된 Capacity는 이동체, 자원 그리고 핸들러에 따라 그 의미가 다르다. 이동체의 경우 Capacity가 없는 것은 S^1 에 해당하는 것으로 자원을 1개만 수용한다는 것을 의미하고, Capacity가 있다는 것은 S^C 에 해당하는 것으로 이동체가 자원을 n개 만큼 수용한다는 것을 의미한다. 자원에 Capacity가 없는 것은 S^1 에 해당하는 것으로 특정 공간에 자원 1개만 존재하는 것을 의미하고, 특정 공간에 자원이 n개 이상 있는 것은 S^C 로 Capacity가 있는 것을 의미한다. 핸들러는 자원, 이동체와 다르게 Capacity가 있는 것으로 Capacity 값 만큼 자원을 수용한다.

위와 같이 자원, 이동체 그리고 핸들러에 대한 할당을 행위에 대해 Lattice화 하면, 그림 1과 같다. S 는 스케줄링에 의해 이동체가 자원을 핸들러에 병렬로 할당하는 것을 의미하고, 그 상황에 맞게 다양한 스케줄링 알고리즘을 사용한다. R, C, H 는 자원, 이동체,

핸들러에 대한 수를 정수로 표현하고, x, y, z 는 R, C, H 의 Capacity로 x 는 특정 공간에 존재하는 자원의 개수를 의미하고, y 와 z 는 이동체와 핸들러의 Capacity 값을 의미한다. S^1 은 핸들러에 Capacity가 있고 자원과 이동체에 Capacity가 없는 것을 의미하고, S^C 는 S^1 을 포함하는 것으로 자원과 이동체 그리고 핸들러에 Capacity가 있는 것을 의미한다. Lattice에서 화살표는 상위관계를 나타내는 것으로 S^C 의 $S^C(R_{\{x\}}, C_{\{y\}}, H_{\{z\}})$ 로 갈수록 복잡도는 커진다.

예를들어 $S^1(2,3,1_{\{\infty\}})$ 는 $S^1(R, C, 1_{\{\infty\}})$ 로 서로 다른 특정 공간에 자원이 1개씩 존재하고, 존재하는 이동체 3개는 1개의 자원만을 수용할 수 있으며, 존재하는 하나의 핸들러는 무한의 자원을 수용할 수 있다. 이는 $S^1(1,1,1_{\{\infty\}})$ 보다 복잡한 것으로 이들의 스케줄링은 스케줄러에 의해 결정되어 이동체가 자원을 핸들러에 병렬로 할당한다. $S^C(4_{\{1,2,1\}}, 2_{\{3,1\}}, 1_{\{10\}})$ 은 $S^C(R_{\{x\}}, C_{\{y\}}, H_{\{z\}})$ 로 서로 다른 공간에 자원이 1개, 2개, 1개가 존재하고, 3개의 자원을 수용할 수 있는 이동체 1개와 1개의 자원을 수용할 수 있는 이동체 1개가 있고, 자원을 10개 수용할 수 있는 핸들러가 1개 있다. 이는 S^1 에 Capacity가 적용된 복잡한 스케줄링으로 이들에 대한 스케줄링은 스케줄러에 의해 결정되어 이동체가 자원을 핸들러에 병렬로 할당한다.

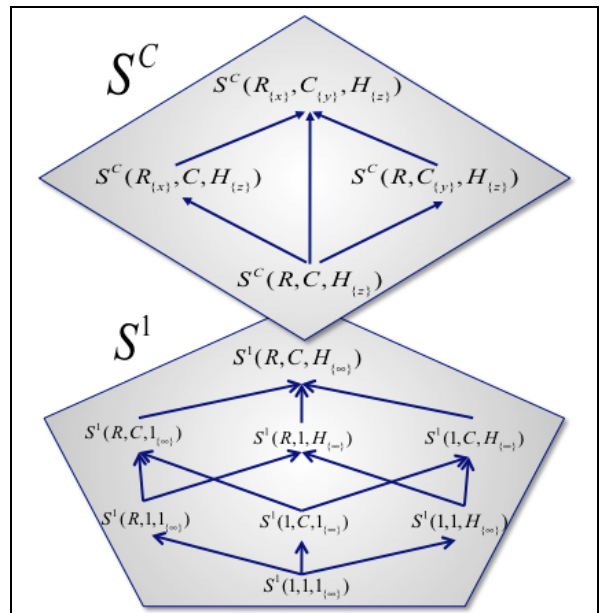


그림 1. CRA Lattice

이러한 CRA는 구급차(이동체)가 환자(자원)을 병원(핸들러)에 이송하는 시스템 등에 행위의 의미적 관점에서 분석 및 검증하는데 적용될 수 있다.

4. TAG

행위에 대한 초기 데이터는 시스템을 명세, 분석 및 검증을 위한 CARDMI의 틀인 *Specification, Analysis*

and Verification Environment (SAVE)[6] 에서 생성되는 특별한 데이터 형식을 따르는 것으로 이러한 데이터는 *Timed Action Graph*(TAG)의 형태로 표시된다. 실행모형을 표현하기 위한 특별한 그래프로 TAG가 있다. 다른 정형기법들은 시간과 공간에서의 행위들에 대해서 동시에 표현하지 못하는 문제점이 존재하지만, TAG는 시간 흐름 기반의 에이전트 행위를 가시적으로 표현할 수 있으며, 행위에 대한 공간과 시간을 동시에 표현할 수 있는 장점을 지닌다. 또한 계층적 공간에 대한 시간 명세 방법을 제공하며, 시간과 공간 및 행위를 추상화할 수 있도록 정의되어 있다. 이를 통해 시간 차원에서의 에이전트의 이동과 통신의 분포에 대한 분석 및 검증 방법을 제공한다.

그림 2는 명세의 이동을 TAG로 표현한 것이다. A, B, C는 노드로 에이전트를 의미하고 B는 A에 포함되어 있다. B는 특정 시점에서 액션에 의해 A를 빠져 나온 후, 특정 시점에 액션에 의해 C 영역으로 들어간 것을 보여주고 있다.

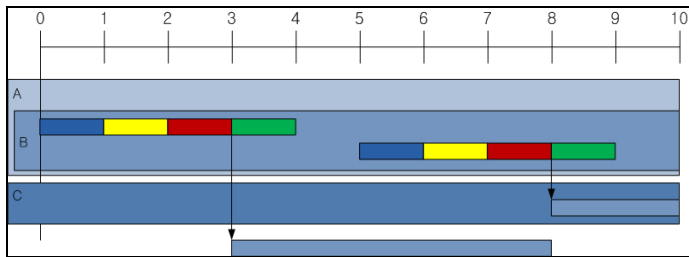


그림 2. 명세의 이동을 TAG로 표현

위와 같이 TAG는 명세에 대한 실행 결과로 모든 원시 데이터를 표현하기 때문에 시스템이 방대해 졌을 경우 복잡도가 증가하여 행위의 분석과 검증이 어려운 문제점을 가지고 있다.

5. Prism Analyzer

Prism Analyzer는 특정 시스템에 대한 행위를 기본 행위부터 복합적 행위까지 모든 행위를 발생 가능한 행위들을 추상화하여 정의한 행위모델인 Lattice를 가지고 있다. 이 Lattice를 Prism Analyzer에서는 온톨로지와 속성문법(*Attributed Prism Analyzer*, APA)으로 정의한 것이다. 온톨로지는 TAG의 원시 데이터를 바탕으로 시스템의 구조적 관계를 알아보기 위함이고, APA는 행위모델을 APA Syntax에 맞게 속성문법으로 정규화하여 정의한 것이다. 따라서 Prism Analyzer는 TAG의 원시 데이터를 의미적 행위 데이터로 행위의 의미적 측면에서 개별적, 연속적, 복합적으로 분석 및 검증이 가능하고, Prism Analyzer에 정의된 다양한 행위 모델을 바탕으로 동일한 TAG에 대해 다양한 분석결과를 도출해 낼 수 있다.

5.1 Prism Analyzer 온톨로지

Prism Analyzer 온톨로지는 행위 모델을 구성하는 모든 원소들에 대한 관계를 정의해 놓은 것으로 적용 가능한 다양한 시스템을 구성하는 모든 원소들에 대한 관계를 포함하고 있다. 예를 들어 CRA가 구급차가 환자를 병원에 이송하는 *Emergency Medical Service*(EMS) System과 같은 시스템에 적용이 된다면, 이를 분석 가능하도록 EMS System에 대한 모든 원소들의 관계를 행위 모델에 맞게 온톨로지에 정의되어야 한다. 이러한 온톨로지는 행위 모델의 분석 방법에 따라 동일한 온톨로지가 사용될 수 있다.

그림 3은 CRA에 EMS System를 적용한 온톨로지를 보여주고 있다. 이 온톨로지를 바탕으로 구급차는 이동체에 속하고, 이동체는 공간에 포함될 수 있음을 확인할 수 있다. 이와 같이 온톨로지를 통해 잘못된 포함관계에 있는 원소들을 분석 및 검증할 수 있다.

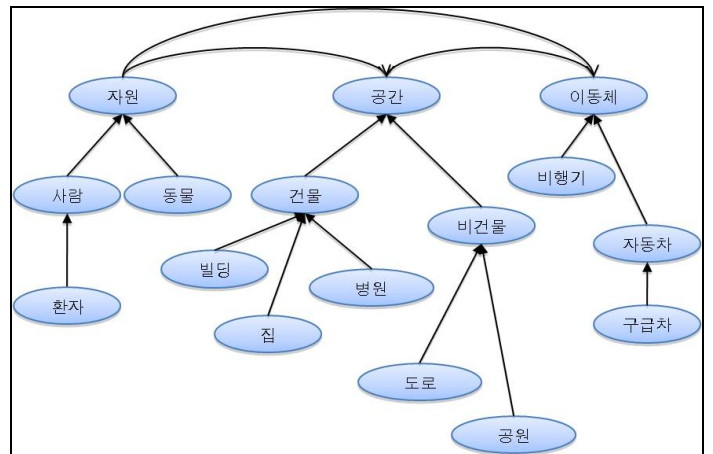


그림 3. EMS System을 적용한 CRA 온톨로지

5.2 Attributed Prism Analyzer(APA) Syntax

APA는 Lattice를 APA Syntax에 맞게 정규화하여 정의해 놓은 것으로, 서로 동일한 행위라도 속성이 다를 경우 이를 서로 다른 행위로 보는 것에 그 의미를 두고 있다. 예를 들어 '이동'이라는 행위가 존재할 때, 행위의 주체가 자원일 경우 자원이 공간상에서 '걸어간다'라는 의미로 해석이 될 수 있지만, 주체가 이동체일 경우 '(다른 공간으로)이동한다'라는 의미로 해석될 수 있다.

다음은 APA Syntax를 살펴본다:

- 1) 행위 : *Behavior(Subject, to)*

$$\{Behavior = in, out, send, receiver\}$$
- 2) Time : *Behavior(Subject, to, time)*
- 3) 통신
 - A. 비동기 : *send(Subject, channel, msg, time)*
 - B. 동기:

$send(Subject, channel, msg, time)$
 $\circ receive(Subject, channel, msg, time)$
 $\{send\ time = receive\ time\}$

- 4) 행위의 연속 : \circ
- 5) 행위의 반복 :

$\{Behavior(Subject, to)\}^+ | \{Behavior(Subject, to)\}^*$
 $(+ := 1 \cdots n, * := 0 \cdots n, n \geq 1)$

$Behavior$ (행위)는 공간상의 이동인 in , out 과 통신시 사용되는 $send$, $receive$ 로 구분이 되며, 이러한 행위는 하나의 액션으로 볼 수 있다. 행위의 주체는 $Subject$ 라 하며 행위의 $target$ (대상)은 to 이다. 예를 들어 $in(ambulance, Hospital)$ 은 ‘구급차가 병원이라는 공간에 이동한다’로 해석이 된다. 하지만 속성이 다음과 같이 $in(Person, Ambulance)$ 라면, ‘사람이 구급차에 탄다’라는 의미로 in 이라는 행위가 ‘이동하다’와 ‘탄다’라는 행위로 의미가 달라 질 수 있다.

$Time$ 은 각 행위의 소비시간을 의미하는 것으로 $time$ 이 0일 경우 즉시 실행을 의미한다.

통신은 동기와 비동기 통신이 존재하며, 통신의 주체인 $Subject$ 가 $channel$ 을 통해서 메시지나 포함하는 노드를 $time$ 동안 $send$ (전송)를 의미한다. 이때 동기 통신에서는 $send$ 와 $receive$ 의 $time$ 이 같아야 한다.

행위의 연속은 ‘ \circ ’으로 연속되며, 행위의 반복에서 $+$, $*$ 은 APA 행위 문법 정의시 사용되는 것으로 $+$ 는 1이상 행위의 주체가 동일한 행위를 명시한 숫자만큼 반복하는 것을 의미하고, $*$ 은 0이상 행위의 주체가 동일한 행위를 명시한 숫자만큼 반복하는 것을 의미한다.

5.3 CRA Lattice의 Attributed Prism Analyzer화

CRA Lattice를 APA로 정의하기 위해서는 S^I 과 S^C 의 모든 행위를 APA Syntax로 정규화하여 정의해야 한다.

CRA Lattice의 가장 기본 행위는 $S^I(1,1,1_{\{\infty\}})$ 로 이는 아래와 같이 정의된다.:

$S^I(1,1,1_{\{\infty\}}) := out(Carrier, Space)^+$
 $\circ in(Carrier, Space)^+ \circ in(Resource, Carrier)$
 $\circ out(Carrier, Space)^+ \circ in(carrier, Handler)$
 $\circ out(Resource, Handler) \circ out(carrier, Space)^*$

여기서 $out(Carrier, Space)^+ \circ in(Carrier, Space)^+$ 는 Carrier가 Resource의 공간까지 공간상의 이동을 보여준다.

S^I Lattice는 Capacity가 없는 것으로 자원과 이동체, 그리고 핸들러의 개수에 따라 Lattice가 달라진다. 즉, $S^I(3_{\{1,1,1\}}, 1, 1_{\{\infty\}})$ 은 하나의 이동체가 존재하는 자원 3개를 연속적으로 핸들러에 할당하는 것으로 $S^I(1,1,1_{\{\infty\}})$ 의 행위가 연속적으로 3번 이뤄짐을 의미한다. 따라서 S^I Lattice의 정규화된 행위 문법은 정의 1과 같다.

정의1. $S := out(Ambulance, Base)$
 $\circ in(Ambulance, Building) \circ in(Person, Ambulance)$
 $\circ out(Ambulance, Base) \circ in(Ambulance, Hospital)$
 $\circ out(Person, Hospital)$

S^C Lattice는 Capacity가 있는 것으로 이동체의 Capacity 여부에 따라 행위 문법이 달라진다:

- 1) $S^C(R, C, H_{\{z\}})$ 일 때, H에 할당 받을 수 있는 자원이 한정된 것으로 자원과 이동체에 Capacity가 없기 때문에 $S^I(1,1,1_{\{\infty\}})$ 의 행위 문법이다.
- 2) $S^C(R_{\{x\}}, C, H_{\{z\}})$ 일 때, 같은 공간에 여러 자원이 있을 수 있는 것으로 이동체에 Capacity가 없기 때문에 $S^I(1,1,1_{\{\infty\}})$ 의 행위 문법이다.
- 3) $S^C(R, C_{\{y\}}, H_{\{z\}})$ 일 때, 이동체에 Capacity가 있는 것으로 이동체가 자원을 y 만큼 수용하여 핸들러에 할당하는 것으로 아래와 같이 행위 문법이 정의된다 :

A. $S^C(R, C_{\{y\}}, H_{\{z\}}) := out(Carrier, Space)^+$
 $\left\{ \begin{array}{l} in(Carrier, Space)^+ \\ \circ in(Resource, Carrier) \\ \circ out(Carrier, Space)^+ \end{array} \right\}^+ \circ in(carrier, Handler)$
 $\circ out(Resource, Handler)^+ \circ out(carrier, Space)^*$

이동체가 y 만큼 서로 다른 공간에 하나씩 존재하는 자원을 연속적으로 수용하여 하나의 핸들러에 연속적으로 할당하는 것이다.

B. $S^C(R, C_{\{y\}}, H_{\{z\}}) := out(Carrier, Space)^+$
 $\left\{ \begin{array}{l} in(Carrier, Space)^+ \\ \circ in(Resource, Carrier) \\ \circ out(Carrier, Space)^+ \end{array} \right\}^+$
 $\left\{ \begin{array}{l} in(carrier, Handler) \\ \circ out(Resource, Handler)^+ \\ \circ out(carrier, Space)^* \end{array} \right\}^+$

이동체가 y 만큼 서로 다른 공간에 하나씩 존재하는 자원을 연속적으로 수용하여 서로 다른 핸들러에 연속적으로 할당하는 것이다.

4) $S^1(R_{\{x\}}, C_{\{y\}}, H_{\{z\}})$ 일 때, 자원과 이동체, 그리고 핸들러에 Capacity/가 있는 것으로 아래와 같이 행위 문법이 정의된다.

$$A. S^C(R_{\{x\}}, C_{\{y\}}, H_{\{z\}}) := out(Carrier, Space)^+ \\ \left\{ \begin{array}{l} in(Carrier, Space)^+ \\ \circ in(Resource, Carrier)^+ \\ \circ out(Carrier, Space)^+ \end{array} \right\}^+ \circ in(carrier, Handler) \\ \circ out(Resource, Handler)^+ \circ out(carrier, Space)^*$$

이동체가 y 만큼 서로 다른 공간에 하나 또는 그 이상 존재하는 자원을 연속적으로 수용하여 하나의 핸들러에 연속적으로 할당하는 것이다.

$$B. S^C(R_{\{x\}}, C_{\{y\}}, H_{\{z\}}) := out(Carrier, Space)^+ \\ \left\{ \begin{array}{l} in(Carrier, Space)^+ \\ \circ in(Resource, Carrier)^+ \\ \circ out(Carrier, Space)^+ \end{array} \right\}^+ \\ \left\{ \begin{array}{l} in(carrier, Handler) \\ \circ out(Resource, Handler)^+ \\ \circ out(carrier, Space)^* \end{array} \right\}^+$$

이동체가 y 만큼 서로 다른 공간에 하나 또는 그 이상 존재하는 자원을 연속적으로 수용하여 서로 다른 핸들러에 연속적으로 할당하는 것이다.

5.4 CRA Lattice를 바탕으로 한 Prism Analyzer를 통한 행위 분석 방법

예를 들어 $time$ 이 $t1$ 에 119 에 있는 구급차가 $Base$ 로 나와서 $time t2$ 에 구급차가 빌딩으로 들어가고, $time t3$ 에 사람이 구급차에 타고, $time t4$ 에 구급차가 $Base$ 로 나와서 $time t5$ 에 구급차가 병원으로 들어가고, $time t6$ 에 환자가 병원으로 들어가는 시스템 S 가 있다. 이를 정의하면 아래와 같다:

$$S := out(Ambulance, Base, t1) \\ \circ in(Ambulance, Building, t2) \circ in(Person, Ambulance, t3) \\ \circ out(Ambulance, Base, t4) \circ in(Ambulance, Hospital, t5) \\ \circ out(Person, Hospital, t6)$$

시스템 S 에 대한 분석 방법은 적용된 Lattice 의 온톨로지에서 노드들의 관계가 어긋나지 않는지와 정의된 행위와 일치하는지 등을 통해 시스템을 분석 및 검증할 수 있다. 또한 각 노드들을 시간의 흐름에 따라 행위를 분석할 수 있다.

CRA 온톨로지에서 이동체인 구급차는 건물인 *Building* 과 *Hospital*, 그리고 *Base* 에 포함될 수 있다. 사람은 이동체인 구급차와 공간상 *Building* 과 *Hospital* 에 포함될 수 있다. 이를 바탕으로 시스템 S 는 CRA 온톨로지 상에서 어긋남이 없는 것을 확인할 수 있다. 그리고 위 시스템 S 는 CRA 에서 이동체, 자원, 핸들러가 각각 1 개인 $S^1(1,1,1_{\{\infty\}})$ 임을 확인할 수 있다. 이를 바탕으로 CRA 행위 문법인 APA 중 $I^1(1,1,1_{\{\infty\}})$ 에 맞는 행위인지를 검증할 수 있고, 위 행위는 CRA APA 를 따른다. 마지막으로 시스템의 각 행위를 하나의 이벤트로 분석이 가능하고, 이에 따라 각 노드의 행위도 아래와 같이 분석이 가능하다 :

$$S := out(e_1) \circ in(e_2) \circ in(e_3) \circ out(e_4) \circ in(e_5) \circ out(e_6) \\ Ambulance := e_1 \circ e_2 \circ e_4 \circ e_5 \\ Person := e_3 \circ e_5$$

이를 통해 Ambulance 와 Person 의 공간상의 이동을 분석할 수 있으며, Ambulance 의 e_4 라는 액션을 수행하기 전에는 반드시 Person 의 e_3 의 액션이 있어야지만 Ambulance 의 연속적인 행위가 발생할 수 있음을 확인할 수 있다.

6. 예제

그림 4는 EMS System의 $S^C(3_{\{2,1\}}, 2_{\{1,2\}}, 1_{\{10\}})$ 로 SAVE로 명세한 것이다. 환자는 총 3명으로 2명은 같은 공간에 존재한다. 구급차는 119에 2대 존재하며, 1대는 2명의 환자를 수용한다. 병원은 총 10명의 환자를 수용할 수 있다. 중앙감시센터는 환자의 위급상황을 감지하여 환자가 병원에서 안전하게 치료를 받을 수 있도록 각 구급차와 병원에 연락을 취한다.

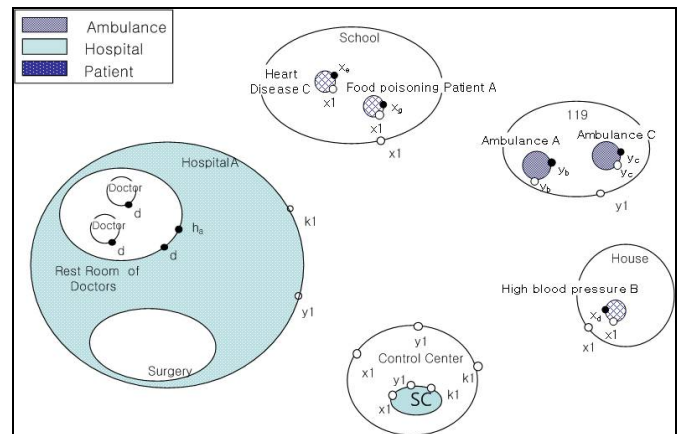


그림 4. $S^C(3_{\{2,1\}}, 2_{\{1,2\}}, 1_{\{10\}})$ 명세

그림 4와 같이 복잡한 시스템을 명세하면 실행결과인

TAG는 그림 5의 왼쪽 그림과 같이 복잡해서 시스템을 분석 및 검증하는데 어려움이 있다. 따라서 Prism중 CRA Prism Analyzer를 적용하면 그림 5의 가운데와 오른쪽 그림과 같이 구급차나 환자, 병원의 개별적 행위뿐만 아니라 연속적, 복합적 행위, 그리고 시스템의 전체 행위에 대해서 분석 및 검증을 할 수 있다. 그림 5의 가운데 그림은 시간의 흐름에 따라 구급차의 개별적 행위에 대해 분석 및 검증한 것이고, 그림 5의 오른쪽 그림은 시스템 전체 행위에 대해서 분석 및 검증한 것이다.

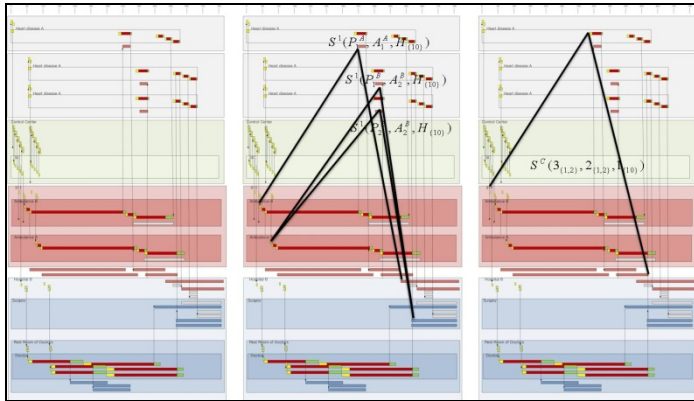


그림 5. 왼쪽 TAG의 결과를 Prism Analyzer로 분석한 결과(가운데, 오른쪽)

7. 결론

분산 이동 실시간 시스템의 명세, 분석 및 검증을 위해 다양한 정형기법이 존재한다. 이들은 이동하는 에이전트 혹은 프로세스에 대한 명세 및 시스템의 안전성과 검증에 대한 분석은 지원하지만 행위의 의미적 관점에서의 분석 및 검증 방법을 제시하고 있지 않다.

본 논문에서는 원시 데이터로 CARDMI로 명세 된 시스템이 실행단계에서 공간과 시간의 흐름에 따라 행위를 분석 및 검증할 수 있는 TAG를 사용하여 행위의 의미적 관점에서 시스템을 분석 및 검증이 가능한 Prism Analyzer를 제안하였다.

제안된 Prism Analyzer는 특정 시스템에 대한 행위를 기본 행위부터 복합적 행위까지 발생 가능한 행위들을 추상화하여 정의한 행위모델인 Lattice를 가지고 있고, 이를 온톨로지와 속성문법으로 정의한 Attributed Prism Analyzer로 정의된다. 이러한 Prism Analyzer는 기존 정형기법에서의 시스템 분석 및 검증 방법과 다른 것으로 시스템 분석 및 검증시 적용 가능한 행위모델을 통해 시스템을 구성하는 각 노드들의 구조적 관계와 각 노드들의 행위를 의미적 관점에서 시간의 흐름에 따라 분석 및 검증이 가능하다. 또한 적용 가능한 다양한 행위모델을 적용하여 동일한 시스템에 대한 서로 다른 다양한 분석 및 검증 결과를 도출해 낼 수 있다.

향후 연구에서는 다양한 시스템을 분석할 수 있도록

좀 더 다양한 행위 모델을 정의하고, Prism Analyzer를 일반화 한다. 또한 이를 구현하여 TAG와 함께 시스템 분석 및 검증에 대한 편의성을 제공한다.

참고문헌

- [1] 저자, 제목, 학술지명, 권, 호, 쪽수, 발행년도 순으로 작성.
- [1] Sangiorgi, D. and Walker, D., "The π -calculus: A Theory of Mobile Processes," Cambridge University Press, 2001.
- [2] G.-C. Roman, P.J. McCann, and J.Y. Plun., "Mobile UNITY: Reasoning and specification in mobile computing," ACM Transactions on Software Engineering and Methodology, pp.250-282, 1997.
- [3] O.H. Jensen, R. Milner, "Bigraphs and mobile processes (revised)," Technical Report UCAM-CL-TR-580, University of Cambridge Computer Laboratory, 2004.
- [4] L. Cardelli, and A. D. Gordon, "Mobile ambients, In Foundations of Software Science and Computational Structures," Maurice Nivat(Ed), No. 1378 in Lecture Note in Computer Science, Springer, pp.140, 1998.
- [5] J. Choi, "A Calculus for Equivalence Analysis and Verification of Distributed Mobile System Based on Abstraction," PhD Dissertation, Chonbuk national Univ, 2007.
- [6] 온진호, 최정란, 이문근, "공간 프로세스 대수를 이용한 정형 명세와 분석에서의 시간속성의 시각화," 정보처리학회논문지, 제16-D권, 제 3호, pp.339-352, 2009.
- [7] Sujeong U, "Design and Implementation of a Scheduler for CARDMI Execution Model," Master Thesis, Chonbuk National University, Korea, 2007.
- [8] Kenneth C. Loudon. Compiler Construction Principles and Practice. PWS publishing Company. 1997.
- [9] Gerard J. Holzmann. The Model Checker SPIN. IEEE Transactions on Software Engineering, v.23 n.5. pp.279-295. May 1997.
- [10] D. Vasumathi. Dr. A Govardhan. Efficient Web Usage Mining Based On Formal Concept Analysis. Journal of Theoretical and Applied Information Technology(JATIT). pp.99-109. 2005.