

# 프로세스 대수에서 이동성과 상호작용을 분석하기 위한 행위 온톨로지를 이용한 추상화 방법

## Abstraction Method for Analysis of Mobility and Interaction in Process Algebra Using Behavioral Ontology

우수정, 온진호, 이문근  
전북대학교 컴퓨터공학과

Su-jeong Woo(wpig04@gmail.com), Jin-ho On(jjinhott@gmail.com),  
Moon-kun Lee(moonkun@jbnu.ac.kr)

### 요약

분산 이동 실시간 시스템을 개발하기 위해 pi-Calculus, Mobile Ambients Calculus, Bigraph 등의 수많은 프로세스 대수가 존재한다. 하지만 시스템이 방대해지고 복잡해질수록 시스템을 구성하는 프로세스들의 통신과 이동 역시 방대해지고 복잡해지므로 프로세스 대수로 이를 이해하는데 어려움이 존재한다. 그러므로 방대하고 복잡한 시스템을 체계적으로 이해할 수 있는 방법이 필요하다. 본 논문에서는 방대하고 복잡한 시스템을 프로세스들의 통신과 이동의 순서화된 추상화 방법 즉, 계층적으로 구조화된 격자(Lattice)들의 형태인 프리즘(Prism)으로 다룬다. 이는 액티브 온톨로지(Active Ontology)에서 확장된 새로운 개념인 행위 온톨로지(Behavior Ontology)를 기반으로 한다. 프리즘은 시스템을 체계적으로 이해하기 위해 시스템을 계층적으로 구성된 행위적 속성을 지닌 격자들 관점에서 분석하는 것을 허용한다. 이러한 방법은 통신과 이동의 복잡함을 의미적이고 계층적으로 구성된 행위의 구조로 체계적으로 이해할 수 있게 한다.

■ 중심어 : | 프로세스대수 | 추상화 | 프리즘 분석기(Prism Analyzer) | 행위 온톨로지(Behavioral Ontology) | 태그(Timed Action Graph) |

### Abstract

A number of process algebras have been proposed to develop distributed mobile real-time systems: pi-Calculus, Mobile Ambients Calculus, Bigraph, etc. However, as the systems get large and complex, the algebras become less suitable for understanding the interactions and mobility of the processes of the systems due to the size and complexity. Therefore it is necessary to handle the size and complexity for systematic understanding of the systems. This paper handles the size and complexity with a method of abstraction on sequences of interactions and movements of processes in the systems, which can be further organized in the form of hierarchically structured lattices, namely, Prism. The theoretical principle of the abstraction is based on a new concept of Behavior Ontology, which is extended from Active ontology. Prism allows the systems to be analyzed in the perspective of the lattices in Prism, which are characterized by the hierarchically organized behavioral properties of the developing systems, for systematic understanding the systems. In this way, the complexity of the interactions and the movements can be handled systematically in the semantically and hierarchically organized structure of the behavior.

■ keyword : | Algebra | Abstraction | Prism Analyzer | Behavioral Ontology | TAG(Timed Action Graph) |

\* 본 연구는 2007년도 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-521-D00451)

\* 본 연구는 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(No.2010-0023787)

\* 본 연구는 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 2단계BK21사업 지원을 받아 수행된 것임

접수번호 : #110628-001

심사완료일 : 2011년 08월 29일

접수일자 : 2011년 06월 28일

교신저자 : 우수정, e-mail : wpig04@gmail.com

## 1. 서론

분산 이동 실시간 시스템(Distributed Mobile Real-time Systems: DMRS)을 명세, 분석 및 검증하기 위한 방법들로 pi-Calculus[1][2], Mobile Ambients Calculus[3], Bigraph[4][5] 등과 같은 다양한 프로세스 대수가 존재한다. 하지만 시스템이 방대해지고 복잡해질수록 시스템을 구성하는 프로세스들의 통신(Interactions)과 이동(Movements) 역시 방대해지고 매우 복잡해지므로 명세, 분석, 검증과정에서 프로세스 대수로 이를 이해하는데 어려움이 존재한다. 그러므로 방대하고 복잡한 시스템을 효율적이고 체계적으로 이해할 수 있는 방법이 필요하다.

본 논문에서는 방대하고 복잡한 시스템을 프로세스들의 이동과 통신의 순서화된 추상화 방법으로 효율적이고 체계적으로 분석한다. 이는 더 나아가 프리즘(Prism)으로 불리는 계층적이며 구조화된 격자들(Lattices)의 형태로 구성된다. 추상화의 이론적 원리는 액티브 온톨로지(Active Ontology)[18]에서 확장된 새로운 개념인 행위 온톨로지(Behavior Ontology)를 기반으로 한다.

본 논문의 접근법은 [그림 1]과 같으며, 다음과 같이 정리될 수 있다:

- 1) 온톨로지(Ontology): 첫 번째로, 구축된 시스템 온톨로지는 개발되는 시스템의 기본 프로세스들 또는 컴포넌트들을 결정하기 위해 구축된다.
- 2) 행동(Actions): 두 번째로, 프로세스들의 기본 행동 즉, 통신(Interactions)과 이동(Movements)은 각 프로세스들과 프로세스들 사이에서 정의된다.
- 3) 행위(Behaviors): 명확성을 위해 행위는 유한한 것으로 취급; 세 번째로, 프로세스들의 기본 행위는 각 프로세스들과 프로세스들 사이에서 결정되며, 이 행위들은 이전 단계에서 결정된 행동들의 순서화에 의해 정의된다.
- 4) 격자(Lattice): 네 번째로, 행위들의 격자들은 이전 단계의 행위들의 추상화에 의해 결정되며, 각 프로세스들과 프로세스들 사이에서 결정된다.

마지막 단계에서 구축된 격자들의 집합을 프리즘(Prism)이라 한다. [그림 1]에서 보여진 바와 같이, 프리즘은 시스템 관점에서 체계적으로 구축된 격자들에 개발되는 시스템의 대수적 명세에 의해 생성된 실행 또는 시뮬레이션 데이터를 투영하여 각 격자 관점의 다양한 분석 결과를 생성한다. 이는 프로세스들의 공간적 시간적 차원에서의 즉각적이고 체계적이지 않은 일차원적인 데이터를 분석하는 기존의 방식을 개선하여 공간상의 통신들과 이동들을 격자들에 의해 추상화된 행위들로 계층적으로 구성하여 체계적이고 구조적으로 분석할 수 있음을 의미한다. 이와 같은 방법은 시스템을 좀 더 의미적으로 분석하고 이해하는데 도움을 준다.

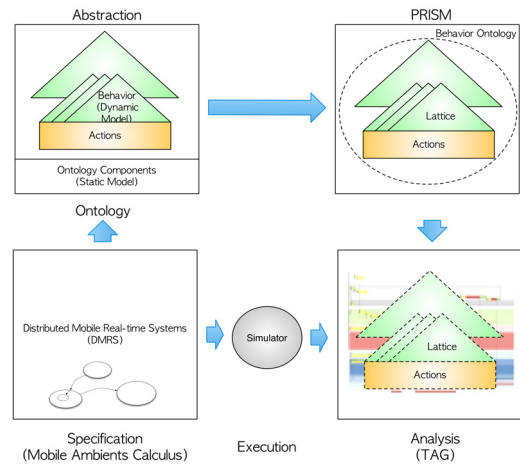


그림 1. Prism을 통한 시스템 분석의 구조적 관점

본 논문에서는 강성 실시간 시스템(hard real-time system)의 전형적인 예제로 응급의료시스템의 최적화된 환자 이송인 EMS(Emergency medical services) System을 예를 들어 논문의 접근법을 아래와 같이 검증한다:

- 1) 명세(Specification): EMS 예제는 프로세스 대수 중 Mobile Ambients Calculus로 시스템을 명세한다.
- 2) 시뮬레이션(Simulation): 예제의 실행데이터는 TAG (Timed Action Graph)[6,7,8]로 생성된다.
- 3) 프리즘(Prism): 행위 온톨로지를 정의하기 위해

액티브 온톨로지가 사용되며, 이는 프리즘(Prism)을 위한 액션(Actions)과 행위(Behaviors) 그리고 격자(Lattices)들을 정의한다.

시각적 이해를 위해, 명세와 시뮬레이션은 SAVE (Specification, Analysis, Verification and Evaluation) [6-8]에서 시각적으로 표현되었으며, 프리즘의 실행 가능성을 증명하기 위해 부분적으로 구현되었다.

본 논문의 구성은 다음과 같다. 2장에서는 프리즘의 기본 원리에 대해 정의하고, 3장에서는 EMS 예제를 Mobile Ambient Calculus와 TAG로 정의 하였다. 4장에서는 EMS의 관점에서 프리즘 접근법의 실 예를 보여준다. 5장에서는 프로세스 대수에서의 다른 정형 기법의 분석 방법들을 비교한다. 마지막으로 6장에서는 결론 및 향후 연구를 다룬다.

## II. Prism 구축 방법

### 1. Ontology

본 논문의 접근법은 온톨로지를 기반으로 하며, 온톨로지는 시스템의 프로세스들이나 컴포넌트들의 범주를 정의하기 위해 사용된다. [그림 2]는 간단한 EMS 예제의 온톨로지를 구축한 것으로 isa와 같은 관계정보로 정적 정보만을 포함한다[18].

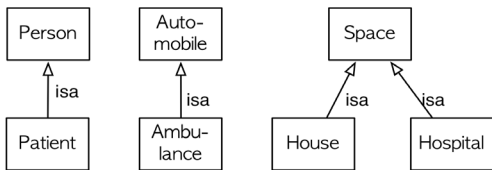


그림 2. 간단한 온톨로지 예

### 2. Behavior Ontology

행위 온톨로지(Behavior Ontology)는 아래와 같이 행동(Actions), 행위(Behaviors) 그리고 격자(Lattices)에 의해 시스템의 동적인 측면을 보여준다.

### [정의 1. Action]

행동(Action)은  $a = \langle O_i, O_j \rangle$  으로,  $O_i$ 와  $O_j$ 는 온톨로지의 클래스를 의미한다. □

행동은 온톨로지 상에서 두 개의 클래스들 간의 순서화된 관계정보를 나타내는 것으로 액티브 온톨로지와 유사하다. 즉, 지리학적 공간상에서 프로세스들의 통신(Interactions)이나 이동(Movements)를 의미하는 것으로 행동  $A = \{a_1, \dots, a_i, \dots, a_n\}$  이다.

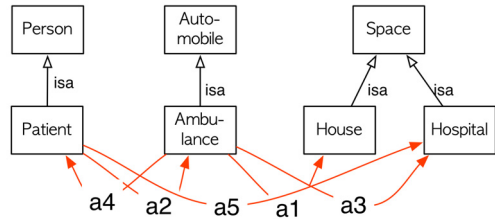


그림 3. EMS 예제의 행동들

[그림 3]은 EMS 예제의 행동들을 보여주는 것으로, 행동은  $a_1 = \langle A, House \rangle$ ,  $a_2 = \langle P, A \rangle$ ,  $a_3 = \langle A, H \rangle$ ,  $a_4 = \langle A, P \rangle$ ,  $a_5 = \langle P, H \rangle$ 가 있다. 여기서  $A$ ,  $P$ ,  $House$  그리고  $H$ 는 구급차, 환자, 환자의 집 그리고 병원을 의미 하고, 각 행동은 아래와 같이 해석 된다:

- 1)  $a_1 = \langle A, House \rangle$ : 구급차  $A$ 가 환자 집인  $House$ 로 이동한다.
- 2)  $a_2 = \langle P, A \rangle$ : 환자  $P$ 가 구급차  $A$ 에 탄다.
- 3)  $a_3 = \langle A, H \rangle$ : 구급차  $A$ 가 병원  $H$ 로 이동한다.
- 4)  $a_4 = \langle A, P \rangle$ : 환자  $P$ 가 구급차  $A$ 에서 내린다.
- 5)  $a_5 = \langle P, H \rangle$ : 환자  $P$ 가 병원  $H$ 로 이동한다.

### [정의 2] Behavior

행위(Behavior)는  $b = \langle a_1, \dots, a_n \rangle$ 으로,  $a_1, \dots, a_n$ 는 정의 1의 행동들을 의미하는 것으로,  $a_i = \langle O_j, O_k \rangle$ 와  $a_{i+1} = \langle O_x, O_y \rangle$ 에서  $O_j = O_x$  또는  $O_j = O_y, O_k = O_x$  또

는  $O_k = O_y$  이다. □

행위는 정의 1에서 정의한 순서화된 일련의 행동들이 연속적으로 나타나는 것을 의미한다. 즉, 프로세스들의 통신 또는 이동이 순서화 된 것을 의미하는 것으로 행위  $B = \{b_1, \dots, b_n\}$  이다. 그리고 명확성을 위해 행위는 유한한 것으로 정의한다.

예를 들어, [그림 3]의 행동들로부터 아래와 같이 행위들을 정의할 수 있다:

- 1)  $b_1 = \langle a_1, a_2, a_3, a_4, a_5 \rangle$ : 이 일련의 행동들은 하나의 구급차가 한 명의 환자가 있는 한 곳의 집에 가서 환자를 태우고 한 곳의 병원으로 이송하는 행위이다.
- 2)  $b_2 = \langle a_1, a_2, a_3, a_4, a_5 \rangle^+$ :  $b_1$  행위의 반복을 의미한다.
- 3)  $b_3 = \left( \langle a_1, a_2 \rangle^+, a_3, \langle a_4, a_5 \rangle^+ \right)$ : 이 일련의 행동들은 하나의 구급차가 여러곳 즉, 환자의 집에 한 명씩 존재하는 환자를 모두 태우고, 한 곳의 병원에 모든 환자를 이송하는 행위를 의미한다.
- 4)  $b_4 = \left( \langle a_1, a_2 \rangle^+, \langle a_3, a_4, a_5 \rangle^+ \right)$ : 이 일련의 행동들은 하나의 구급차가 여러 곳 즉, 환자의 집에 한 명씩 존재하는 환자를 모두 태우고, 여러 병원에 한 명씩 모든 환자를 이송하는 행위를 의미한다.
- 5)  $b_5 = \left( \left( \langle a_1, a_2, a_3, a_4, a_5 \rangle \middle| \langle a_1, a_2 \rangle^+ \right)^+, \left( a_3, \langle a_4, a_5 \rangle^+ \middle| \langle a_3, a_4, a_5 \rangle^+ \right)^+ \right)$ : 이 행위는  $b_1$

부터  $b_4$ 의 모든 행위들을 포함하는 행위를 의미한다.

이와 같은 개념은 정규 표현인 속성 문법(context-free grammar)[17]을 기본으로 한다.

**[정의 3] 행위들 사이의 부분수열 관계 (Subsequence Relation, | )**

두 행위  $b_i$ 와  $b_j$  사이의 부분수열 관계( $|$ )는  $b_i | b_j$

로 표현되며,  $b_i$ 는  $b_j$ 의 일부분의 연속적 행위들을 지닌다. □

예를 들어, 위의 예제에서  $b_1 | b_3$ 의 부분수열 관계가 존재한다.

**[정의 4] Abstract Behavior**

추상화 행위(Abstract Behavior)  $B(O_1, \dots, O_n)$ 는  $AB(b)$ 로,  $b$ 는 정의 2에 의해  $b = \langle a_1, \dots, a_n \rangle$ 이며,  $b$ 의 주체들은 하나의 집합으로 각  $O_i$ 는  $O_{\{a_1, \dots, a_n\}}^x$ 로 표현된다. 여기서  $x$ 는  $O_i$ 's의 cardinality로 표현이 되며 이는  $O_i$ 's의 수를 의미한다. 그리고 각  $c_j$ 는  $O_i$ 's 중  $j$ 번째  $O_i$ 의 capacity로  $O_i$ 의 취급 가능한 수를 의미한다. 여기서  $AB$ 는 추상화 프로세스를 의미한다. □

Cardinality와 capacity의 추가적인 개념을 설명하면 다음과 같다. Capacity는 적용될 수 없을 경우 표시하지 않으며, 더군다나  $O_i$ 's가 뚜렷하게 구별되면  $O$ 에 대한  $i$ 의 정의는 필요하지 않다.

예를 들어, 행위의 주체  $P, A$ 와  $H$ 에 대하여 행위  $b_i$ 부터  $b_4$ 는 아래와 같이 추상화 된다:

$$1) B_1 = B(P^1, A_{(1)}^1, H_{(1)}^1) = AB(b_1),$$

$$b_1 = \langle a1, a2, a3, a4, a5 \rangle;$$

capacity가 1인 하나의 구급차가 한명의 환자를 capacity가 1인 한 곳의 병원에 이송하는 것을 의미한다.

$$2) B_2 = B(P^1, A_{(1)}^1, H_{(1, \dots, 1)}^1) = AB(b_2),$$

$$b_2 = \langle a1, a2, a3, a4, a5 \rangle^+;$$

capacity가 1인 하나의 구급차가 환자들 중 한 명의 환자를 capacity가 1인 병원들 중 한 곳의 병원에 이송하는 것으로 한 번에 한 번씩 반복적으로 수행하는 것을 의미한다.

$$3) B_3 = B(P^i, A_{(i)}^1, H_{(i)}^1) = AB(b_3),$$

$$b_3 = \langle \langle a1, a2 \rangle^+, a3, \langle a4, a5 \rangle^+ \rangle;$$

capacity가  $i$  인 하나의 구급차가  $i$  명의 환자를 capacity가  $i$  인 한 곳의 병원에 한 번에 이송하는 것을 의미한다.

$$4) B_4 = B(P^i, A_{(i)}^1, H_{(1, \dots, 1)}^i) = AB(b_4),$$

$$b_4 = \langle \langle a1, a2 \rangle^+, \langle a3, a4, a5 \rangle^+ \rangle;$$

capacity가  $i$  인 하나의 구급차가 한번에  $i$  명의 환자를 capacity가 1인 여러 병원에 한명씩 모든 환자를 한 번에 이송하는 것을 의미한다.

$$5) B_5 = B(P^i, A_{(j)}^1, H_{(z_1, \dots, z_k)}^k) = AB(b_5),$$

$$B_5 = \left\langle a1, a2, a3, a4, a5 \mid \left( \begin{array}{l} \langle a1, a2 \rangle^+, \\ \langle a3, \langle a4, a5 \rangle^+ \mid \rangle \\ \langle a3, a4, a5 \rangle^+ \end{array} \right) \right\rangle;$$

이는  $B_1$  부터  $B_4$  까지 모든 추상화 행위를 포함한다.

**[정의 5] 추상화 행위들 사이의 부분수열 관계( $\mid$ )**

두 행위  $B_i$  와  $B_j$  의 부분수열관계( $\mid$ )는  $B_i = AB(b_i)$  와  $B_j = AB(b_j)$  로, 만일  $b_i \hat{=} b_j$  라면  $B_i \hat{=} B_j$  로 정의된다.

예를 들어, 위의 예제에서  $B_1 \hat{=} B_3$  가 존재한다.

**[정의 6] Behavior Lattice**

행위 격자(Behavior Lattice)  $L(O_1, \dots, O_l) = BL(B)$  으로,  $B = \{B_1, \dots, B_n\}$  이다.  $B$  의 모든  $B_i$  's 는 동일한  $(O_1, \dots, O_l)$  를 지니지만, 다른 cardinalities와 capabilities를 지니고,  $O_i$  's 의 capabilities의 cardinalities를 지닌 순서화된 추상화에 대해서 하나의 격자 구조를 구성한다. 즉,  $B_i$  의  $O_{(c_1, \dots, c_k)}^x$  는  $B_j$  의

$O_{(c_1, \dots, c_k)}^x$  의 추상화로  $B_i$  는  $B_j$  의 추상화이다. 즉,  $B_i \hat{=} B_j$  가 성립된다. 여기서 BL은 행위 격자를 구축하는 과정을 의미한다. □

격자는 행위들의 추상화를 기본으로 하며, 하나의 행위는 다른 행위들의 추상화를 의미한다. 이와 같이 추상화는 자연스럽게 하나의 격자를 구축하며  $L = \{L_1, \dots, L_n\}$  이다.

예를 들어 그림 4에서는 예제에 대한 두 개의 격자를 다음과 같이 보여 준다:

1)  $L(P^i, A_{(j)}^1, H_{(z_1, \dots, z_k)}^k)$ : 이 격자는  $B_1$  부터  $B_3$  로 구축된 격자로,  $B_i$  's 사이에서 추상화 될 수 있는 주체들의 cardinalitically로 연결된다. 이는 capacity가  $j$  인 하나의 구급차가 환자들을 다양한 방법으로 병원에 이송하는 추상화를 의미한다.

2)  $L(P^i, A_{(j_1, \dots, j_j)}^j, H_{(z_1, \dots, z_k)}^k)$ : 이 격자는  $A_{(j)}^1$  의 추상화에 대해서  $A_{(j_1, \dots, j_j)}^j$  의 행위들을 포함하는 것이다. 이는 capacity가  $j_j$  인  $j$  개의 구급차들이 환자들을 다양한 방법으로 병원에 이송하는 추상화를 의미한다.

표기의 편의성을 위해, 상관없거나 의도되지 않은 행위들은 격자나 프리즘에서 생략될 수 있다. 예를 들어,  $L(P^i, A_{(j)}^1, H_{(z_1, \dots, z_k)}^k)$  에서  $A_{(j)}^1$  는 본 격자의 행위의 주체이고,  $P^i$  와  $H_{(z_1, \dots, z_k)}^k$  는 본 격자의 행위의 관심 밖으로 생략되어  $L(A_{(j)}^1)$  로 정의된다.

격자들은 격자들 사이에 존재하는 일반 추상화 행위들에 의해 서로 연결된다. 이와 같이 격자들의 집합을 아래와 같이 행위 프리즘(Behavior Prism)이라 정의한다.

**[정의 7] Behavior Prism**

행위 프리즘(Behavior Prism)은  $P(O_1, \dots, O_n) = BP(L)$  으로,  $B = \{L_1, \dots, L_n\}$  이다. 여기서  $L$  의  $L_i$  와  $L_j$  에 대한

$B_i$  's 는  $L_i = \{B_1, \dots, B_i, \dots, B_n\}$  과  $L_j = \{B_1, \dots, B_j, \dots, B_m\}$  일 때,  $B_i = B_j$  이다. □

다시 말해, 모든  $L_i$  's 은 일반 추상화 행위들에 의해서로 연결되어 있고, 이를 행위 프리즘이라 한다. 프리즘은 정의된 시스템의 행위들의 주체에 대하여 시스템의 통합적 뷰(View)를 보여준다.

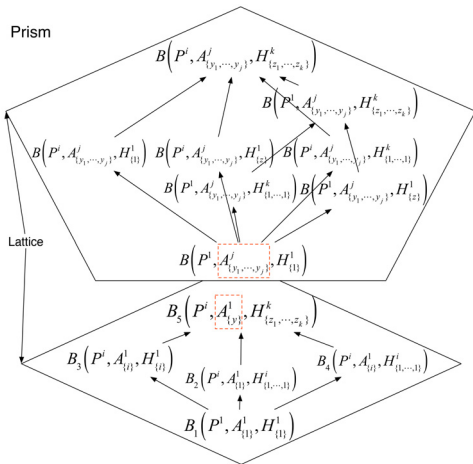


그림 4. EMS 예제의 격자와 프리즘

예를 들어, [그림 4]에서  $P(P, A, H)$  의 프리즘은  $L(P^i, A^j_{\{x_1, \dots, x_j\}}, H^k_{\{z_1, \dots, z_k\}})$  와  $L(P^j, A^i_{\{y_1, \dots, y_j\}}, H^k_{\{z_1, \dots, z_k\}})$  로 구성되며 이들은  $B(P^i, A^j_{\{y\}}, H^k_{\{z_1, \dots, z_k\}})$  에 의해 서로 연결된다. 그리고 이는 EMS 예제에서 환자들을 집에서 병원으로 다양한 경우로 이송하는 것을 보여주는 것으로, 이는 계층적으로 구조화된 추상화 방법을 기반으로 한다.

### III. 시간과 공간 속성이 복잡한 예제

분산 이동 실시간 시스템을 분석하는데 있어 EMS 예제를 들어 프리즘의 효용성을 검증한다. EMS 예제는 매우 복잡한 환경에서 프로세스들의 엄격한 상호작용의 행위적 요구사항을 명세하고 분석하는데 좋은 예

제이다. 이는 8명의 응급 환자 즉, 심장병 환자와 당뇨병 환자, 식중독 환자 그리고 고혈압 환자가 위급한 상황에 따라 119에 존재하는 2개의 구급차와 병원에 존재하는 하나의 구급차에 의해 병원에 이송되는 예제로 각 환자들은 중앙감시센터로 위급상황을 알림으로써 병원까지 이송되는 데드라인을 지니게 된다.

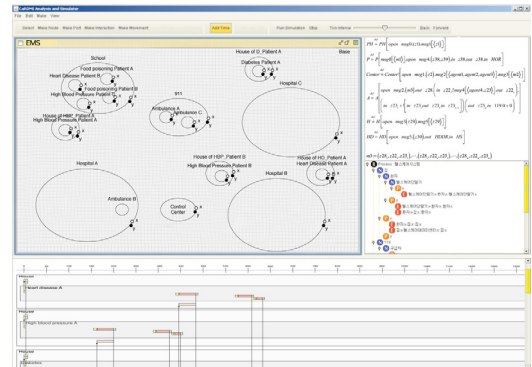


그림 5. EMS 예제를 SAVE로 명세

[그림 5]는 EMS 예제를 명세하고 실행할 수 있는 도구인 SAVE(Specification, Analysis, Verification and Evaluation)를 보여준다.

첫 번째로, 시스템은 RHG(Real-time Hyper-Graph) 에디터로 프로세스는 노드로 표현하고, 각 프로세스간 통신(interaction)은 포트를 통해 채널을 생성하여 수행한다. 프로세스 또는 에이전트들의 이동(movement)은 in/out 행위이며 서로 다른 프로세스로 이동하는 것을 의미한다. 이와 같은 그래픽 명세는 [그림 5]의 맨 위 왼쪽 창과 같이 명세 된다:

- 심장병 환자  $P_H^A$ , 고혈압환자  $P_B^A$  와  $P_B^B$ , 그리고 당뇨병 환자  $P_D^A$  가 각 환자의 집에 위치하며, 심장병 환자  $P_H^B$ , 고혈압 환자  $P_B^C$  그리고 식중독 환자  $P_F^A$  와  $P_F^B$  가 학교에 위치하는 것으로 총 8명의 환자가 존재한다.
- 119는 3개의 병원  $H_3^A$ ,  $H_3^B$  그리고  $H_3^C$  의 중앙에 위치하며, 각 병원은 3명, 3명, 그리고 5명의 환자를 개별적으로 수용한다.

$$\begin{aligned}
 & \text{Heart Disease } A = P_H^A, \text{ Heart Disease } B = P_H^B, \text{ Diabetes } A = P_D^A, \\
 & \text{High Blood pressure } A = P_P^A, \text{ High Blood pressure } B = P_P^B, \text{ High Blood pressure } C = P_P^C, \\
 & \text{Food poisoning } A = P_F^A, \text{ Food poisoning } B = P_F^B, \\
 & \text{House of Patient } = PH, \text{ School } = PS, \text{ Center } = Center \\
 & 119 = 119, \text{ Ambulance } = A_1^A, \text{ Ambulance } = A_2^A, \text{ Ambulance } = A_3^C, \\
 & \text{Hospital } A = H_3^A, \text{ Hospital } B = H_3^B, \text{ Hospital } C = H_3^C, \\
 & \text{Rest Room of Doctors } = HRRD, \text{ Doctor } = HD, \text{ Surgery } = HS
 \end{aligned}$$

$$\begin{aligned}
 & PH = PH \left[ \text{open msg0}(z1).\text{msg1}[(z1)] \right] \\
 & P = P \left[ \text{msg0}[(m1)].\text{open msg4}(z38, z39).\text{in } z38.\text{out } z38.\text{in } HOR \right] \\
 & Center = Center! \left[ \text{open msg1}(z2).\text{msg2}[(agent1, agent2, agent3)].\text{msg3}[(m2)] \right] \\
 & A = A \left[ \text{open msg2}(m3).\text{out } z28. \left( \text{in } z22, \text{msg4}[(agent4, z23)].\text{out } z22, \right) \right. \\
 & \quad \left. \left( \text{in } z23, +! \left[ \text{in } z23.\text{out } z23.\text{in } z23, +! \right] \right) \left( \text{out } z23.\text{in } 119.0 + 0 \right) \right] \\
 & H = H \left[ \text{open msg3}(z29).\text{msg5}[(z29)] \right] \\
 & HD = HD \left[ \text{open msg5}(z30).\text{out } HDDR.\text{in } HS \right] \\
 & m3 := (z28, z22, z23), \dots, (z28, z22, z23), \dots, (z28, z22, z23), \dots
 \end{aligned}$$

$$\begin{aligned}
 & EMS = EMS \left[ \begin{array}{l} PH [P_H^A] | PH [P_H^B] | PH [P_H^A] | PH [P_H^B] | PS [P_P^C | P_P^C | P_P^A | P_P^B] | \\ Center | \\ 119 [A_1^A | A_2^A] | \\ H_3^A [A_1^A | HRRD | HD] | HS | \\ H_3^B [HRRD | HD] | HS | \\ H_3^C [HRRD | HD] | HS | \end{array} \right]
 \end{aligned}$$

그림 6. EMS 예제를 Mobile Ambients Calculus로 명세

- 119안에 2개의 구급차  $A_1^B$  그리고  $A_3^C$  가 존재하며 이들은 1명 그리고 3명의 환자를 수용 가능하다. 그리고 병원  $H_3^A$  에는 1명의 환자를 수용 가능한 구급차  $A_1^A$  가 존재한다.

두 번째로, 그래픽적으로 명세한 시스템에 대한 텍스트 명세는 프로세스 대수로 명세 하는 것으로 [그림 5]의 오른쪽 가장 위쪽 부분에 존재한다. 본 예제에서는 Mobile Ambients Calculus를 사용하였으며 [그림 6]과 같이 명세 된다.

세 번째로, SAVE의 시뮬레이터에 의해 생성된 실행 결과는 [그림 7]과 같이 TAG(Timed Action Graph)로 보여 진다. TAG는 시간의 흐름에 따라 구급차들에 의한 이동 결과를 보여주는 것으로 프로세스들 그리고 에이전트들 사이의 잦은 이동과 엄격한 통신으로 예제와 같이 시스템이 복잡하면 복잡할수록 예제를 분석하는데 있어 매우 복잡하고 어렵게 만든다. 따라서 시스템을 분석하는 방법으로는 잦은 이동과 통신을 간단화 또는 추상화 방법 등의 새로운 분석 방법이 필요하다.

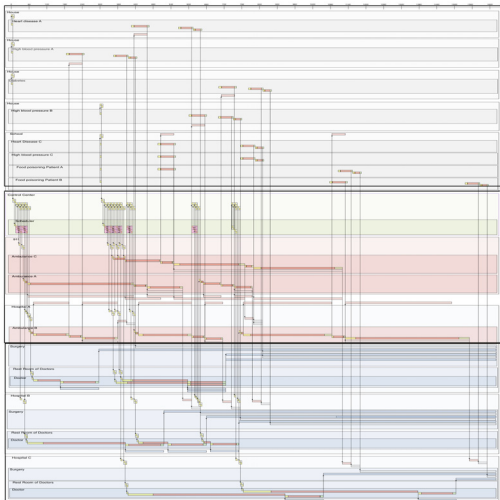


그림 7. EMS 예제를 TAG 생성 결과

#### IV. 행위 온톨로지를 기반한 분석

시스템을 효과적으로 분석하기 위해서는 시스템을 분석할 수 있는 특정 관점 즉, 특정 분석 관점으로 계층적으로 추상화할 수 있는 의미적인 구조의 정의가 필요하다. 예를 들어, EMS 예제는 “구급차가 환자를 병원에 이송 한다”라는 관점으로 해석이 될 수 있다. 다시 말해, 이는 자원(Resources)을 이동체(Carrier)에 의해 핸들러(Handlers)에 할당하는 문제로 분석될 수 있다. 따라서, 본 논문에서는 이를 자원 할당 문제(Resource Allocation Problem, RAP)로 정의한다.

##### 1. RAP(Resource Allocation Problem) 정의

자원 할당 문제의 주체는 자원(Resource, R), 이동체

(Carrier, C) 그리고 핸들러 (Handler, H)로 이동체에 의해 자원이 핸들러에 할당되는 문제를 다루고 있다.

주체의 cardinality와 capacity는 이전에 정의한 것과 같이  $O_{(c_1, \dots, c_n)}^n$ 로 정의되며,  $n$ 은  $1 \leq n \leq \text{sum}(c_1 + \dots + c_n)$ 를 만족해야 한다.

[정의 8] RAP

자원 할당 문제(RAP)는  $B(R_{(x)}^i, C_{(y)}^j, H_{(z)}^k)$ 로 자원은  $R$ , 이동체는  $C$ , 핸들러는  $H$ 로 정의된다. □

이는 정의 4를 기반으로 하며, 예를 들어 설명하면,

$R_{(1,3)}^2$ ,  $C_{(1,3,1)}^3$  그리고  $H_{(4)}^1$ 의 의미는 다음과 같다:

- $R_{(1,3)}^2$ : 1명의 환자, 3명의 환자가 서로 다른 2공간에 존재한다.
- $C_{(1,3,1)}^3$ : 3개의 구급차가 있으며, 1,3,1의 환자를 수용하는 capacity를 가진다.
- $H_{(4)}^1$ : 1개의 병원이 존재하며, 4명의 자원을 수용한다.

이 정의에 의해  $B(R_{(1,3)}^2, C_{(1,3,1)}^3, H_{(4)}^1)$ 는 추상화된 행위로 1명, 3명의 환자들을 capacity가 1, 3 그리고 1인 3개의 구급차에 의해 capacity가 4인(즉, 4명의 환자를 수용 가능한) 병원에 이송되는 것으로 해석이 된다. 따라서 시스템은 위의 주어진 조건들을 만족해야 한다.

2. EMS 예제의 행위 온톨로지

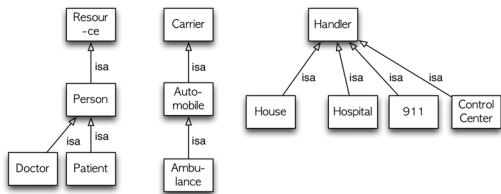


그림 8. EMS 예제 온톨로지

[그림 8]은 EMS 예제의 온톨로지를 보여주고 있다.

3. RAP를 위한 행위 온톨로지

3.1 RAP의 Action

정의 1에서 정의한 바와 같이, RAP의 행동은 다음과 같다. 여기서 구급차, 환자, 환자의 집, 그리고 병원은  $A$ ,  $P$ ,  $House$ ,  $H$ 로 정의 된다:

- $a_1 = \langle A, House \rangle$
- $a_2 = \langle P, A \rangle$
- $a_3 = \langle A, H \rangle$
- $a_4 = \langle A, P \rangle$
- $a_5 = \langle P, H \rangle$

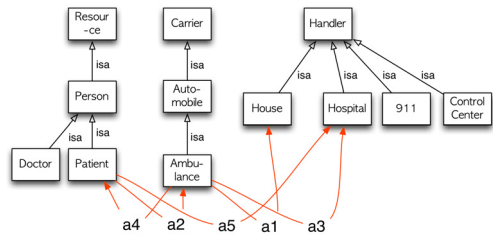


그림 9. RAP의 Action

[그림 9]는 EMS 온톨로지 상에서 위에서 정의한 행동들을 보여준다.

3.2 RAP의 행위

정의 2에서 정의한 바와 같이, RAP의 행위들은 아래와 같다:

- $B_1 = \langle a1, a2, a3, a4, a5 \rangle$
- $B_2 = \langle a1, a2, a3, a4, a5 \rangle^+$
- $B_3 = \langle a1, \langle a2 \rangle^+, a3, \langle a4, a5 \rangle^+ \rangle$
- $B_4 = \langle a1, \langle a2 \rangle^+, \langle a3, a4, a5 \rangle^+ \rangle$
- $B_5 = \langle a1, \langle a2 \rangle^+, \langle a3, \langle a4, a5 \rangle^+ | a3, a4, a5 \rangle^+ \rangle$
- $B_6 = \langle \langle a1, a2 \rangle^+, a3, \langle a4, a5 \rangle^+ \rangle$



- $$B_7 = \langle \langle a1, a2 \rangle^+, \langle a3, a4, a5 \rangle^+ \rangle = AB \left( \left\langle \langle a1, a2 \rangle^+, \langle a3, \langle a4, a5 \rangle^+ | \rangle^+ \right\rangle \right)$$
- $$B_8 = \langle \langle a1, a2 \rangle^+, \langle a3, \langle a4, a5 \rangle^+ | a3, a4, a5 \rangle^+ \rangle$$
- $$B_9 = B \left( R_{\{x_1, \dots, x_j\}}^i, C_{\{y\}}^1, H_{\{z_1, \dots, z_k\}}^k \right)$$
- $$B_{10} = \left\langle \left\langle \left\langle a1, a2, a3, a4, a5 \right\rangle | \left\langle a1, \langle a2 \rangle^+, \langle a3, \langle a4, a5 \rangle^+ | \right\rangle^+ \right\rangle | \left\langle \langle a1, a2 \rangle^+, \langle a3, \langle a4, a5 \rangle^+ | \right\rangle^+ \right\rangle$$

### 3.3 RAP 추상화 행위

정의 4에서 정의한 바와 같이 RAP의 추상화 행위는 아래와 같이 정의 된다:

- $$B_1 = B \left( R_{\{1\}}^1, C_{\{1\}}^1, H_{\{1\}}^1 \right) = AB \left( \langle a1, a2, a3, a4, a5 \rangle \right)$$
- $$B_2 = B \left( R_{\{x_1, \dots, x_j\}}^i, C_{\{y\}}^1, H_{\{z_1, \dots, z_k\}}^k \right) = AB \left( \langle a1, a2, a3, a4, a5 \rangle^+ \right)$$
- $$B_3 = B \left( R_{\{1\}}^1, C_{\{1\}}^1, H_{\{z\}}^1 \right) = AB \left( \langle a1, \langle a2 \rangle^+, a3, \langle a4, a5 \rangle^+ \rangle \right)$$
- $$B_4 = B \left( R_{\{1\}}^1, C_{\{1\}}^1, H_{\{1, \dots, 1\}}^k \right) = AB \left( \langle a1, \langle a2 \rangle^+, \langle a3, a4, a5 \rangle^+ \rangle \right)$$
- $$B_5 = B \left( R_{\{1\}}^1, C_{\{1\}}^1, H_{\{z_1, \dots, z_k\}}^k \right) = AB \left( \left\langle a1, \langle a2 \rangle^+, \langle a3, \langle a4, a5 \rangle^+ | \right\rangle^+ \right)$$
- $$B_6 = B \left( R_{\{1, \dots, 1\}}^i, C_{\{1\}}^1, H_{\{z\}}^1 \right) = AB \left( \langle \langle a1, a2 \rangle^+, a3, \langle a4, a5 \rangle^+ \rangle \right)$$
- $$B_7 = B \left( R_{\{1, \dots, 1\}}^i, C_{\{1\}}^1, H_{\{1, \dots, 1\}}^k \right) = AB \left( \langle \langle a1, a2 \rangle^+, \langle a3, a4, a5 \rangle^+ \rangle \right)$$
- $$B_8 = B \left( R_{\{1, \dots, 1\}}^i, C_{\{1\}}^1, H_{\{z_1, \dots, z_k\}}^k \right)$$

### 3.4 RAP의 Behavior Lattices

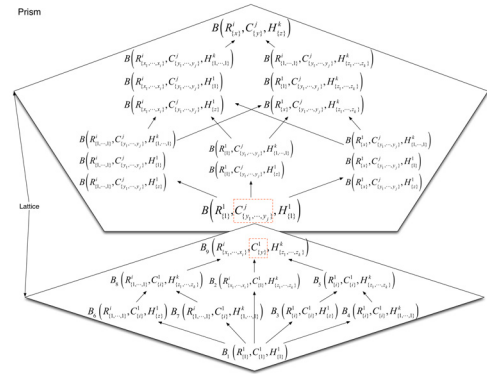


그림 10. RAP관점을 적용한 EMS System의 Prism

위의 정의 6을 바탕으로 RAP의 행위 격자는 그림 10과 같다:

- $$\bullet$$
 Lattice  $L_{\{y\}}^1 = L \left( P^i, C_{\{y\}}^1, H_{\{z_1, \dots, z_k\}}^k \right)$ : 이는 capacity가  $y$  인 구급차의  $B(R, C, H)$  행위를 보여준다.
- $$\bullet$$
 Lattice  $L_{\{y_1, \dots, y_j\}}^j = L \left( P^i, C_{\{y_1, \dots, y_j\}}^j, H_{\{z_1, \dots, z_k\}}^k \right)$ : 이는 각각  $y_1, \dots, y_j$  의 capacity를 지닌  $j$  개의 구급차의  $B(R, C, H)$  행위를 보여준다.

RAP 격자들은 위의 정의 3과 5를 바탕으로 부분수열 관계에 의해 [그림 6]과 같이 구축이 된다.

RAP의 행위 프리즘 즉,  $P(R,C,H)$ 는 [그림 10]과 같고, 이는  $L_{(y)}$  과  $L'_{(y_1, \dots, y_j)}$  가  $B(R_{(1)}, C'_{(y)}, H'_{(1)})$ 에 의해 연결구조를 지닌다. 그리고 이 프리즘은 방대하고 복잡한 실시간 시스템에서의 실행 또는 시뮬레이션 데이터를 추상화 관점으로 분석하는데 사용된다.

### 3.4 행위 프리즘을 바탕으로 분석 결과

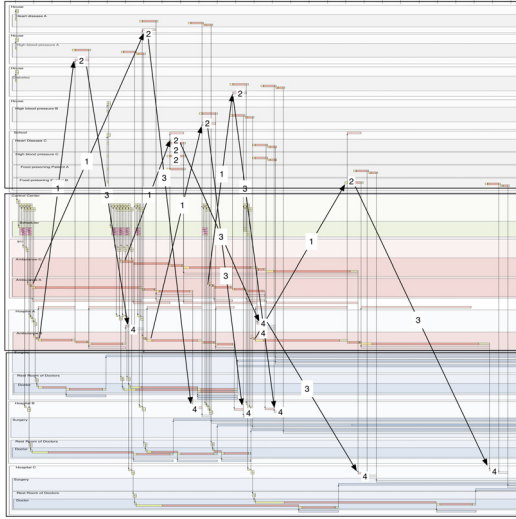


그림 11. TAG에 RAP Behavior Ontology를 적용하여 분석된 결과

[그림 11]은 [그림 7]의 실행 결과인 TAG에 RAP의 행동을 보여주는 것으로 구급차 관점에서 행위를 정의하면 아래와 같다:

$$\begin{aligned}
 \bullet B_1^A &= \left\{ \begin{aligned} B_{1,1}(R^1, C_{(1)}^A, H_{(1)}^1) &= AB \langle a1, a2, a3, a4, a5 \rangle, \\ B_{1,2}(R^1, C_{(1)}^A, H_{(1)}^1) &= AB \langle a1, a2, a3, a4, a5 \rangle \end{aligned} \right\} \\
 \bullet B_1^B &= \left\{ \begin{aligned} B_{1,1}(R^1, C_{(1)}^B, H_{(1)}^1) &= AB \langle a1, a2, a3, a4, a5 \rangle, \\ B_{1,2}(R^1, C_{(1)}^B, H_{(1)}^1) &= AB \langle a1, a2, a3, a4, a5 \rangle, \\ B_{1,3}(R^1, C_{(1)}^B, H_{(1)}^1) &= AB \langle a1, a2, a3, a4, a5 \rangle \end{aligned} \right\} \\
 \bullet B_5^C &= \left\{ \begin{aligned} B_{5,1}(R^1_{(3)}, C_{(3)}^C, H^k_{\{z_1, \dots, z_k\}}) &= \\ AB \langle a1, a2, a2, a2, a3, a4, a5, a4, a5, a3, a4, a5 \rangle & \end{aligned} \right\}
 \end{aligned}$$

구급차 A, B 그리고 C는  $B_1^A$ ,  $B_1^B$  그리고  $B_5^C$ 로 [그

림 12]와 같이 각각 추상화 레벨 1, 1 그리고 5로 분석된다. 그리고 이 행위들은 다음과 같은 부분수열 관계를 지닌 행위 프리즘을 통해 추상화 된다:

- $B_{1,1}(R^1, C_{(1)}^A, H_{(1)}^1) \hat{\circ} B_2(R_{(1,1)}^2, C_{(1)}^A, H^k_{\{z_1, \dots, z_k\}})$   
 $\hat{\circ} B_9(R^1_{\{x_1, \dots, x_j\}}, C_{(1)}^A, H^k_{\{z_1, \dots, z_k\}})$
- $B_{1,2}(R^1, C_{(1)}^A, H_{(1)}^1) \hat{\circ} B_2(R_{(1,1)}^2, C_{(1)}^A, H^k_{\{z_1, \dots, z_k\}})$   
 $\hat{\circ} B_9(R^1_{\{x_1, \dots, x_j\}}, C_{(1)}^A, H^k_{\{z_1, \dots, z_k\}})$
- $B_{1,1}(R^1, C_{(1)}^B, H_{(1)}^1) \hat{\circ} B_2(R_{(1,1)}^3, C_{(1)}^B, H^k_{\{z_1, \dots, z_k\}})$   
 $\hat{\circ} B_9(R^1_{\{x_1, \dots, x_j\}}, C_{(1)}^B, H^k_{\{z_1, \dots, z_k\}})$
- $B_{1,2}(R^1, C_{(1)}^B, H_{(1)}^1) \hat{\circ} B_2(R_{(1,1)}^3, C_{(1)}^B, H^k_{\{z_1, \dots, z_k\}})$   
 $\hat{\circ} B_9(R^1_{\{x_1, \dots, x_j\}}, C_{(1)}^B, H^k_{\{z_1, \dots, z_k\}})$
- $B_{1,3}(R^1, C_{(1)}^B, H_{(1)}^1) \hat{\circ} B_2(R_{(1,1)}^3, C_{(1)}^B, H^k_{\{z_1, \dots, z_k\}})$   
 $\hat{\circ} B_9(R^1_{\{x_1, \dots, x_j\}}, C_{(1)}^B, H^k_{\{z_1, \dots, z_k\}})$
- $B_{5,1}(R^1_{(3)}, C_{(3)}^C, H^k_{\{z_1, \dots, z_k\}}) \hat{\circ} B_9(R^1_{\{x_1, \dots, x_j\}}, C_{(3)}^C, H^k_{\{z_1, \dots, z_k\}})$

이 행위들의 모든 개념들은 행위(behaviors), 격자(lattices), 프리즘(prism)의 의미적 개념을 따르며, 결론적으로 예제를 쉽고 직관적으로 분석할 수 있다.

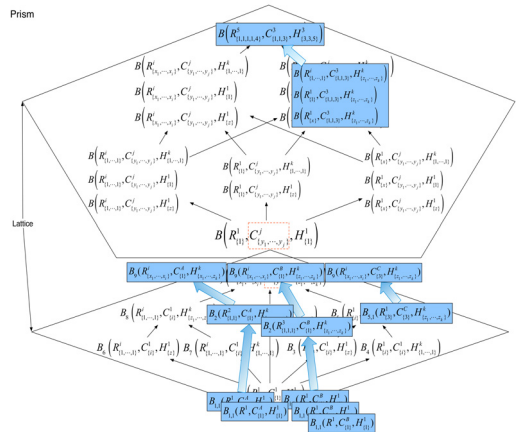


그림 12.  $B(R_{1,1,1,1,4}^5, C_{(1,3,1)}^3, H_{(3,3,5)}^3)$ 를 RAP Behavior Ontology로 분석한 결과

표 1. Process Algebra Formalisms

Criteria	CSP[9,10]	CCS[11,12]	ACP[13,14]	LOTOS[15,16]	TCSP[17]
Temporal model	None	None	None	None	Dense
Automated tools	Some	None	Good	Some	None
Reliability	Good	Good	Good	Good	Good
Proof system	Axiomatic	Bisimulation	Bisimulation	Bisimulation	Axiomatic
Industrial strength	Some	Some	Some	Great	Some
Methods of verification	Both	Both	Both	Model-checking	Both
Concurrency	Interleaved	Interleaved	Interleaved	Interleaved	Both
Communication	Synchronous/ asynchronous	Synchronous	Synchronous	Synchronous	Synchronous
Reverse engineering	No	No	No	No	No

### V. 비교 연구

분산 이동 실시간 시스템을 명세, 분석 및 검증하기 위한 방법들은 pi-calculus, Mobile Ambients 등과 같은 프로세스 대수 기반 정형기법 방법들이 사용된다. 이와 같은 정형기법으로 명세한 시스템은 시스템이 크고 복잡할수록 시스템이 갖는 시간과 공간 속성에 의해 그 시뮬레이션 또는 실행 데이터 역시 크고 복잡하여 이를 이해하고 분석하는데 상당한 어려움이 존재한다.

[표 1]은 기존에 제안되었던 다양한 프로세스 대수 기반 정형기법의 분석 및 검증 방법에 대한 분석 자료이다.

위와 같이 기존 프로세스 대수 기반 정형기법들은 프로세스의 상태를 트랜지션 시스템에 대한 Bisimulation[19], Axiomatic[20], Model Checking[21] 등을 사용하여 시스템의 안전성 및 구조만을 분석 및 검증하기 위한 방법들이다. 위와 같은 방법들은 시스템의 행위에 따른 의미적 분석이 아닌 시뮬레이션의 최종 결과 분석만을 위하여 제안되었다.

본 논문의 제안 방법은 기존 방법에서 비중있게 사용되지 않았던 시뮬레이션 또는 실행 데이터에 의미를 부여하여 체계적이고 구조적인 의미적 추상화를 통해 전체 시스템을 일괄적이고 체계적으로 분석할 수 있는 방법을 제공한다. 이는 기존 정형기법들의 확장된 분석방법으로 적용될 수 있다.

### VI. 결론

본 논문에서는 프로세스 대수로 매우 방대하고 복잡한 분산 이동 실시간 시스템을 명세하고, 실행된 데이터를 체계적이면서 효과적인 접근 방법으로 분석할 수 있는 새로운 방법을 제안하였다.

본 논문의 접근법은 행위 온톨로지라는 새로운 개념을 기반으로 하며, 이는 행동(Actions), 행위(Behaviors), 격자(Lattices) 그리고 프리즘(Prism)으로 구성된다. 이들의 의미적 개념은 액션(Actions)과 행위(Behaviors)들의 추상화된 기본 구조로 구성되며, 이는 격자들의 프리즘으로 구축된다.

시스템을 위한 다양한 격자들의 프리즘이 구성이 되면, 시스템에 대한 분석은 프리즘에서 나타나는 추상화를 기반으로 체계적이면서 의미적으로 분석이 수행된다. 이는 시스템을 쉽고 직관적으로 분석할 수 있게끔 한다.

향후 연구에서는 부분적으로 구현된 SAVE를 매우 방대하고 복잡한 실 산업의 어플리케이션을 분석 가능한 도구로 확장할 것이다.

### 참고 문헌

[1] R. Miler, J. Parrow and D. J. Walker, *A Calculus of Mobile Processes, Part I*, Report ECS-LFCS-89-85, Laboratory of foundations of Computer

- Science, Computer Science Dep., Edinburgh Univ, 1989.
- [2] R. Miler, J. Parrow, and D. J. Walker, *A Calculus of Mobile Processes, Part II*, Report ECS- LFCS-89-86, Laboratory for foundations of Computer Science, Computer Science Dep., Edinburgh Univ, 1989.
- [3] K. Adi, L. Hamza, L. Pene, "Formal Modeling for Security Behavior Analysis of Computer Systems," MCETECH, pp.39-59, 2008.
- [4] R. Miler, *Biographical Reactive Systems: Basic Theory*, Technical Report 503, University of Cambridge Computer Laboratory, 2001.
- [5] R. Milner, "Biographical Reactive Systems," In Proc of the 12th International Conference on Concurrency Theory, Lecture Node in Computer Science, No.2154, pp.16-35, 2001.
- [6] J. Choi, *A Calculus for Equivalence Analysis and Verification of Distributed Mobile System Based on Abstraction*, PhD Dissertation, Chonbuk National Univ, 2007.
- [7] J. H. On, *A Formal Method for Specification and Analysis of Timing Properties in a Process Algebra*, Master Thesis, Chonbuk National Univ, 2007.
- [8] J. H. On, "Visual Representation of Temporal Properties in Formal Specification and Analysis using a Spatial Process Algebra," KIPS Transactions. Vol.16-D, No.3, 2009.
- [9] Y. Zhu, Z. Q. Huang, Guangquan Zhang, "Modeling and Analysis of Real-Time Software Based on Resource Communication Sequential Process," ICIECS, pp.1-5, 2009.
- [10] R. Milner, *A Calculus of Communicating Systems*, LNCS 90, 1980.
- [11] R. Milner, *Communication and Concurrency*, Prentice-Hall International Series in Computer Science, 1989.
- [12] Li Bao, Weishi Zhang, Xiuguo Zhang, "Describing and Verifying Web Service by CCS on a Case Study," ICNSC, pp.1571-1576, 2008.
- [13] J. A. Bergstra, and J. W. Klop, "Process Algebra for Synchronous Communication," Information and Control, Vol.60, pp.109-137, 1984.
- [14] F. S. De Boer, J. W. Klop, C. Palamidessi, "Asynchronous communication in process algebra," LICS, pp.137-147, 1992.
- [15] L. Logrippo, T. Melanchuck, and R. J. D. Wors, "The Algebraic Specification Language LOTOS: An Industrial Experience," ACM SIGSOFT Software Engineering Notes, Vol.15, pp.59-66, 1990.
- [16] P. L. Takecian, J. E. Ferreira, S. Malkowski, C. Pu, "Using LOTOS for rigorous specifications of workflow patterns," CollaborateCom, pp.1-7, 2010.
- [17] Chomsky, Noam, *Three models for the description of language*, Information Theory, IEEE Transactions Vo.1.2, No.3, 1956.
- [18] W. Xing, O. Corcho, C. Goble, and M. Dikaiakos, "Active Ontology: An Information Integration Approach for Highly Dynamic Information Sources," in Europe Semantic Web Conference, pp.1-15, 2007.
- [19] Guang Zheng, Miao Jiang, Gao Chen, "Process Algebra with Goal Based Bisimulation," ISECS, pp.89-95, 2010.
- [20] L. Spenner, P. Krier, M. Thornton, "Large system decomposition and simulation methodology using axiomatic analysis," SYSTEMS, pp.223-227, 2010.
- [21] G. Norman, C. Palamidessi, D. Parker, "Model checking the probabilistic pi-calculus," QEST, pp.169-178, 2007.

저 자 소 개

우 수 정(Su-jeong Woo)

정회원



- 2006년 8월 : 원광대학교 컴퓨터 공학과(공학사)
- 2008년 8월 : 전북대학교 컴퓨터 공학과(공학석사)
- 2008년 8월 ~ 현재 : 전북대학교 컴퓨터공학과 박사수료 후

과정중

<관심분야> : 소프트웨어공학, 정형기법

과 조교수

- 1999년 3월 ~ 2007년 2월 : 전북대학교 컴퓨터학과 부교수
- 2007년 3월 ~ 현재 : 전북대학교 컴퓨터공학부 교수  
<관심분야> : 정형기법, 소프트웨어 재/역공학, 실시간 시스템, 운영체제, 형식언어, 병렬함수언어, 컴파일러 등

온 진 호(Jin-ho On)

정회원



- 2006년 2월 : 원광대학교 컴퓨터 공학과(공학사)
- 2008년 2월 : 전북대학교 컴퓨터 공학과(공학석사)
- 2008년 8월 ~ 현재 : 전북대학교 컴퓨터공학과 박사수료 후 과

정중

<관심분야> : 소프트웨어공학, 정형기법

이 문 근(Moon-kun Lee)

정회원



- 1989년 : The Pennsylvania State University, Computer Science 학과 졸업(이학사)
- 1992년 : The University of Pennsylvania, Computer and Information Science 학과 졸업

(이공학석사)

- 1995년 : The University of Pennsylvania, Computer and Information Science 학과 졸업 (이공학박사)
- 1992년 5월 ~ 1996년 1월 : 미국, Computer Command and Control Company, Computer Scientist로 근무
- 1996년 4월 ~ 1998년 3월 : 전북대학교 컴퓨터학과 전임강사
- 1998년 4월 ~ 1999년 2월 : 전북대학교 컴퓨터학과