

# 응급 의료 시스템의 자동 제어를 위한 CARDMI 프로세스 알제브라 기반 스케줄러에 관한 연구

우수정<sup>O\*</sup> 온진호\* 최정란\*\* 이문근\*

전북대학교 전자정보공학부\*, 고려대학교 컴퓨터·통신공학부\*\*

{wpig04, jjinghott}@chonbuk.ac.kr, jlchai@chonbuk.ac.kr, moonkun@chonbuk.ac.kr

## A Study on Scheduler Based on CARDMI Process Algebra for Automated Control of Emergency Medical System

Sujeong U<sup>O\*</sup>, Jinho On\*, Jung-Rhan Choi\*\*, Moonkun Lee\*

Chonbuk Univ\*, Korea Univ\*\*

### 요 약

현재 응급 의료 시스템에서는 환자의 소재를 파악하기 힘들 뿐만 아니라, 환자의 이송과정에서도 환자를 치료할 수 없는 병원에 이송하는 등의 문제점을 지닌다. 또한 이송을 위한 대기중인 구급차가 없거나, 환자의 병명에 따른 시한을 지키지 못함으로써 환자를 위급한 상태까지 이르게 하는 경우도 존재한다. 본 논문은 이러한 문제를 해결하기 위해 자동화된 중앙 관리 체계를 기반으로 한 u-EMS의 시공간 환경과 행위를 명세 및 분석하기 위한 CARDMI기반의 HAP 스케줄러를 제안한다. 이를 통해 응급 의료 시스템에서 발생할 수 있는 문제점을 u-EMS를 통해 환자를 적시적소에 EMS서비스를 제공할 수 있도록 스케줄링 하였다.

### 1. 서 론

현재 응급 의료 시스템(EMS : *Emergency Medical System*)[1]에서는 환자의 소재를 파악하기 힘들 뿐만 아니라, 환자의 이송과정에서도 환자를 치료할 수 없는 병원에 이송하는 등의 문제점을 지닌다. 또한 이송을 위한 대기중인 구급차가 없거나, 환자의 병명에 따른 시한을 지키지 못함으로써 환자를 위급한 상태까지 이르게 하는 경우도 존재한다.

본 논문은 이러한 문제를 해결하기 위해 자동화된 중앙 관리 체계를 기반으로 한 u-EMS(*ubiquitous-EMS*)의 시공간 환경과 행위를 명세 및 분석하기 위한 정형기법인 CARDMI(*Calculus of Abstract Real-Time Distribution, Mobility and Interaction*)[2,3,4] 기반의 HAP(*Hospital Ambulance Patient*) 스케줄러를 제안한다.

CARDMI는 분산 이동 실시간 시스템(DMRS: *Distributed Mobile Real-Time Systems*)의 공간과 행위적 특성들을 명세, 분석 및 검증에 위한 정형기법이다. HAP 스케줄러는 CRA(*Carrier Resource Allocation*) Algorithm을 기반으로 하고 있으며 CRA는 발원지(*Source*)에 있는 자원을 이동체 또는 운반체(*Carrier*)에 의해 목적지(*Destination*)인 프로세스에 할당하는 알고리즘이다.

본 논문은 u-EMS의 시공간 환경과 행위를 명세 및 분석하기 위한 CARDMI의 HAP 알고리즘을 통해 EMS에서 발생하는 문제점을 u-EMS를 통해 환자를 적시적소에

EMS서비스를 제공할 수 있도록 스케줄링 하였다.

### 2. CRA (*Carrier Resource Allocation*) Algorithm

EMS에서 발생할 수 있는 문제점을 해결하기 위한 일반적인 스케줄링 알고리즘인 CRA를 정의한다.

CRA는 그림 1과 같이 발원지에 있는 자원을 이동체에 따라 목적지인 프로세스에 할당하는 것으로 이때 이동체와 프로세스는 각각 서로 다른 자원을 수용할 수 있는 능력인 Capacity(용량)를 가지고 있다. 이동체의 종류에 따라 Capacity만큼 자원이 프로세스로 이동 가능하고 프로세스에 대한 자원 할당에는 프로세스의 종류와 각 프로세스의 Capacity만큼 자원을 할당할 수 있다. 이때 발원지에 있는 n개의 자원들과 n개의 이동체, n개의 프로세스들은 서로 병렬적으로 수행된다.

그리고 발원지에 있는 각각의 자원들은 자원 생성시 큐(Queue)에 순서대로 들어가지만 목적지인 프로세스에 할당되는 순간 우선순위가 가장 높은 자원이 이동체에 의해 할당된다. 따라서 발원지에 있는 자원들은 시간에 따라 우선순위가 계속 변하기 때문에 FIFO(First In, First Out) 방식이나 RR(Round-robin) 방식이 아닌 FIPO(First In, Priority Out) 방식을 따른다.

본 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-202-D00451)

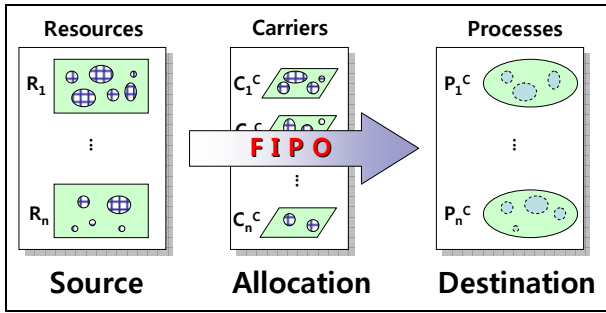


그림 1. CRA (Carrier Resource Allocation) Algorithm

다음은 CRA의 복잡도에 대한 설명이다.

위의 그림 1를  $S(R, C, P)$  으로 표현할 수 있다. 이는

발원지에 있는 자원( $R$ )을 우선순위가 가장 높은 자원부터 이동체( $C$ )를 통해 목적지인 프로세스( $P$ )에 할당( $S$ )하는 문제로 이에 대한 복잡도는 그림 2와 그림 3의 Lattice로 보여준다.

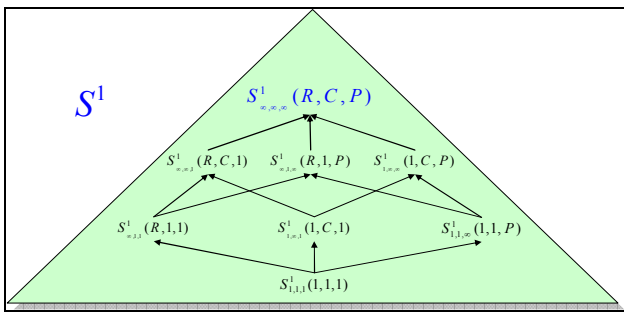


그림 2.  $S^1_{\infty, \infty, \infty}(R, C, P)$  Lattice

그림 2의  $S^1$  은  $S^1_{\infty, \infty, \infty}(R, C, P)$  로  $S^1_{1,1,1}(1,1,1)$  을 바탕으로 그림 2와 같은 Capacity가 1로  $\infty$  (무한대)로 존재하는  $R, C, P$ 의 관계를 Lattice에 의해서 보여준다. 여기서  $S^1_{1,1,1}(1,1,1)$  은  $R:C:P$  가 1:1:1의 관계인 경우로 1개만 존재하는 자원을 이동체가 한번에 1개씩 1개의 자원만을 수용할 수 있는 프로세스에 할당하는 것을 의미한다.  $S^1_{\infty, \infty, \infty}(R, C, P)$  는  $\infty$ 의 자원, 이동체, 프로세스가 존재하는 것으로 우선순위가 높은 자원부터 각 이동체가 한번에 1개의 자원을, 1개의 자원만을 수용하는 각 프로세스들에  $\infty$ 의 이동체가 병렬적으로 할당하는 것을 의미한다.

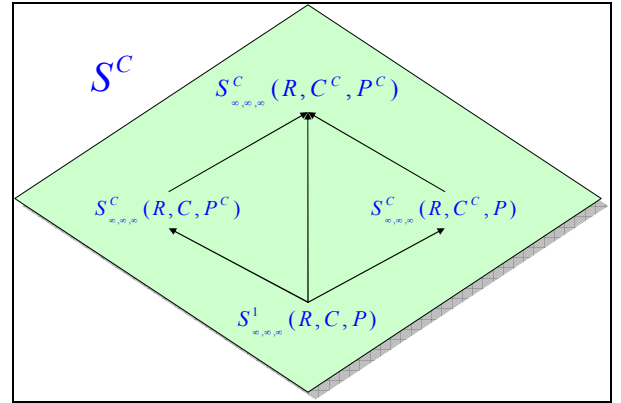


그림 3.  $S^C_{\infty, \infty, \infty}(R, C^C, P^C)$  Lattice

그림 3의  $S^C$  는  $S^1$  과  $S^C_{\infty, \infty, \infty}(R, C, P^C)$  ,  $S^C_{\infty, \infty, \infty}(R, C^C, P)$  를 포함하는 것으로  $S^C_{\infty, \infty, \infty}(R, C^C, P^C)$  를

의미한다. 이때  $S^C$  에서  $C$  는 Capacity, 즉 수용할 수 있는 자원의 수를 의미 하며 이동체와 프로세스에만 Capacity가 존재한다.

$S^1_{\infty, \infty, \infty}(R, C, P)$  는 이동체와 프로세스에 Capacity가 없는 것을 의미한다.

$S^C_{\infty, \infty, \infty}(R, C, P^C)$  는  $P$  에 대한 Capacity가 있는 것으로  $\infty$ 의 자원을 각 이동체가 특정 프로세스에 한번에 1개씩 자원을 병렬적으로 할당 하는 경우로 이때 특정 프로세스는 수용할 수 있는 자원의 수가 Capacity만큼 한정되어 있다.

$S^C_{\infty, \infty, \infty}(R, C^C, P)$  는  $C$  에 대한 Capacity가 있는 것으로  $C$  는 발원지에 있는  $\infty$ 의 자원을 한꺼번에 수용할 수 있는 Capacity만큼 1개의 자원만을 수용하는 프로세스에 병렬적으로 할당 하는 것을 의미한다.

$S^C_{\infty, \infty, \infty}(R, C^C, P^C)$  는  $C$  와  $P$  에 Capacity가 있는 것으로 Source에 있는  $\infty$ 의 자원을 이동체가 한번에 수용할 수 있는 Capacity만큼 이동하며, 프로세스가 수용할 수 있는 Capacity만큼의 자원을 이동체가 할당하는 것을 의미한다.

따라서 발원지에 있는  $\infty$ 의 자원은  $C$  와  $P$  의 Capacity만큼 병렬적으로 할당하는 그림 3과 같은 복잡도를 Lattice에 의해 보여준다.

### 3. CARDMI기반 응급 의료 시스템의 중앙 관리 스케줄러

#### 3.1 CARDMI

CARDMI는 시스템을 행위정보와 공간정보로 구분하여 명세 하는 정형기법이다. 기존의 프로세스 대수의 제약사항들을 해

결하기 위해 공간정보와 에이전트 들의 행위정보(이동, 통신 등)를 구분하였고, 행위는 이동과 인터액션으로 구분하였다. 이러한 구분을 통하여 진화하는 DMRS에서 에이전트의 순차성, 병렬성, 분산성, 이동성 등에 대한 모델링, 추상화, 동일성 검증 등을 제안하였다.[2,3,4]

CARDMI의 기본적인 기본동작은 다음과 같다:

- $x(Z)$ 는 포트  $x$ 를 통하여 임의의  $Z$  (에이전트, 포트, 메시지를) 받는 행위를 의미하며,
- $\bar{x}(M)$ 는 출력포트  $\bar{x}$ 를 통하여  $M$ 을 내보내는 행위를 의미한다.
- $in A$ 는 현재 에이전트가  $A$ 안으로 이동을 의미하며,
- $out A$ 는  $A$ 밖으로 이동을 의미한다.

여기서  $x$ 와  $\bar{x}$ 는 에이전트간의 인터액션이며,  $in$ 과  $out$ 은 이동을 의미한다.

정의 1은 시간 속성이 추가된 t-CARDMI[3, 4]에서 에이전트의 이동 및 인터액션을 정의하기 위해 필요한 문법을 의미이다.

[정의 1] Syntax

1. Primitive action

$$E_T := \frac{t_p x(Z)_{[t_s, t_e]} \bullet E_T \mid \frac{t_p \bar{x}(M)_{[t_s, t_e]} \bullet E_T \mid E_T \setminus (vx)}{!E_T \mid E_{T_1} + E_{T_2} \mid 0 \mid \varepsilon \mid \mathfrak{M} \mid [E_T] \mid \langle E_{T_1}, E_{T_2} \rangle \bullet E_{T_3}}$$

2. Composition

$$\varepsilon := E_T \mid E_T \mid E_T \mid E_T$$

3. Movements

$$\mathfrak{M} := \frac{t_p in A_{[t_s, t_e]} \bullet E_T \mid \frac{t_p out A_{[t_s, t_e]} \bullet E_T$$

4. Communication Entity

$$M := m', M \mid m' \quad m' := A \mid x_i \mid y_i \mid m_i .$$

$$Z := z', Z \mid z' \quad z' := z : agent \mid z : msg \mid z : port$$

정의 2는  $E_T \xrightarrow{\varphi} E_T'$ 로,  $\varphi$ 의 동작을 수행하고  $E_T$ 가  $E_T'$ 으로 전개됨을 의미한다.

전이는 다음과 같은 규칙 형태의 집합으로 정의된다.

전제  
결론 (조건)

“만약 전제(premise)와 측면조건(side condition)이 만족되면 결론(conclusion)을 유도할 수 있다.”

[정의 2] 의미(Semantics)

(1) Tick-time

$$\frac{}{[t_r] E_{T[t_s, t_e]} \xrightarrow{\Delta t} [t_r - t] E_{T[t_s, t_e]}} (0 < t_r, 1 \leq i \leq n) \text{ or}$$

$$\frac{}{[t_r] E_{T[t_s, t_e]} \xrightarrow{\Delta t} [t_r] E_{T[t_s + t_e, t_e]}} (t_s + i \leq t_e, t_r = 0, 1 \leq i \leq n)$$

(2) Immediately tick-time  
-Ready time 감소

$$\frac{t_p E_{T[t_s, t_e]} \xrightarrow{\Delta t} [t_r] E_{T[t_s, t_e]}}$$

-Action의 실행시간

$$\frac{t_p E_{T[t_s, t_e]} \xrightarrow{\Delta t} [t_r] E_{T[t_s - t_p, t_s + t_e - t_p]}}{(0 = t_r, 1 \leq i \leq n) \left( t_s - t_p + i \leq t_e - t_p \right)}$$

(3) time-out

$$\frac{\frac{t_p E_{T[t_s, t_e]} \xrightarrow{\vartheta_{01}} [t_{p_1} + t_{p_2}] E_{T_1} \mid \frac{t_p E_{T[t_s, t_e]} \xrightarrow{\vartheta_{02}} [t_{p_1} + t_{p_2}] E_{T_2}}{\langle E_{T[t_s, t_e]}, E_{T_2} \rangle \bullet E_{T_3}} \xrightarrow{\vartheta_{03}} [t_{p_1} + t_{p_2}] E_{T_3}}{(0 \leq t_{p_1} + t_{p_2} \leq t_s \leq t_e, t_{p_1} \geq 0, t_{p_2} \geq 0, t_{p_3} \geq 0)} \text{ or}$$

$$\frac{\frac{t_p E_{T[t_s, t_e]} \xrightarrow{\vartheta_{01}} [t_{p_1} + t_{p_2}] E_{T_1} \mid \frac{t_p E_{T[t_s, t_e]} \xrightarrow{\vartheta_{02}} [t_{p_1} + t_{p_2}] E_{T_2}}{\langle E_{T[t_s, t_e]}, E_{T_2} \rangle \bullet E_{T_3}} \xrightarrow{\vartheta_{03}} [t_{p_1} + t_{p_2}] E_{T_2}}{(0 \leq t_{p_1} + t_{p_2} \leq t_s \leq t_e, t_{p_1} \geq 0, t_{p_2} \geq 0, t_{p_3} \geq 0)} \left( t_{p_1} < t_{p_2}, t_{p_1} > t_{p_2} \right)$$

(4) input-act

$$\frac{t_p x(Z)_{[t_s, t_e]} \bullet E_T \xrightarrow{x(M)_{01}} [t_p + t_p'] \{M / Z\} E_T}{\left( \begin{array}{l} x \in fn(E_T) \wedge M \notin fn(E_T) \text{ or} \\ x \in bn(E_T) \wedge M \notin bn(E_T) \\ 0 \leq t_p + t_s \leq t_e \leq t_e, t_p' \geq 0 \end{array} \right)}$$

(5) output-act

$$\frac{t_p \bar{x}(M)_{[t_s, t_e]} \bullet E_T \xrightarrow{\bar{x}(M)_{01}} [t_p + t_p'] E_T}{(0 \leq t_p + t_s \leq t_e \leq t_e, t_p' \geq 0)}$$

(6) summation

$$\frac{\frac{t_p E_{T_j[t_s, t_e]} \xrightarrow{\vartheta_{j1}} [t_p + t_p'] E_{T_j}}{\sum_{j=1}^n E_{T_j[t_s, t_e]} \xrightarrow{\vartheta_{01}} [t_p + t_p'] E_T}}{(1 \leq i \leq n, t_p' \geq 0) \left( 0 \leq t_p + t_s \leq t_e \leq t_e, t_p' \geq 0 \right)}$$

(7) parallel

$$\frac{\frac{t_p E_{T_1[t_s, t_e]} \xrightarrow{\vartheta_{11}} [t_p + t_p'] E_{T_1}'}{t_p E_{T_1[t_s, t_e]} \mid E_{T_2} \xrightarrow{\vartheta_{11}} [t_p + t_p'] E_{T_1}' \mid E_{T_2}}}{\left( \begin{array}{l} ns(\varphi) \cap ns(E_2) = \emptyset \\ 0 \leq t_p + t_s \leq t_e \leq t_e \\ t_p' \geq 0 \end{array} \right)} \text{ or}$$

$$\frac{\frac{t_p E_{T_2[t_s, t_e]} \xrightarrow{\vartheta_{21}} [t_p + t_p'] E_{T_2}'}{E_{T_1} \mid t_p E_{T_2[t_s, t_e]} \xrightarrow{\vartheta_{21}} [t_p + t_p'] E_{T_1} \mid E_{T_2}'}}{\left( \begin{array}{l} ns(\varphi) \cap ns(E_1) = \emptyset \\ 0 \leq t_p + t_s \leq t_e \leq t_e \\ t_p' \geq 0 \end{array} \right)}$$

(8) In-movement

$$\frac{t_p in A_{[t_s, t_e]} \bullet E_T \xrightarrow{in A_{01}} [t_p + t_p'] E_T}{(A \sqsupseteq Agt(E_T)) \in \{I_p \text{ of } Agt(E_T) \cap I_p \text{ of } A\} \left( 0 \leq t_p + t_s \leq t_e \leq t_e, t_p' \geq 0 \right)}$$

(9) Out-movement

$$\frac{t_p out A_{[t_s, t_e]} \bullet E_T \xrightarrow{out A_{01}} [t_p + t_p'] E_T}{(\bar{A} \sqsupseteq Agt(E_T)) \in I_p \text{ of } Agt(E_T) \left( 0 \leq t_p + t_s \leq t_e \leq t_e, t_p' \geq 0 \right)}$$

3.2 HAP

HAP은 CARDMI 기반 u-EMS를 위한 스케줄러로 CRA

알고리즘을 기반으로 한다. 즉  $S(R, C, P)$  에서  $R$  은 환자(Patient : P),  $C$  는 구급차(Ambulance : A),  $P$  는 병원(Hospital : H)로  $S(P, A, H)$  로 표현할 수 있으며, 이는 응급환자 발생에 따른 구급차와 병원에 대한 스케줄러로 HAP(Hospital Ambulance Patient) 알고리즘 이라 한다.

```
HAP_Scheduler(G)
H : array [1..1] of Hospital Agent in G; /* Parallel agents */
A : array [1..m] of Ambulance Agent in G; /* Parallel agents */
P : array [1..n] of Patient Agent in G; /* Parallel agents */
i, j, k : integer; /* Will be updated in real -time */
Begin
while ( ∃ patience in P ) /* Wait until ∃ patient */
idle;
i = select_patient(P, Time); /* Select a most urgent patient */
j = select_ambulance(A, P[i], Time); /* Select the closest ambulance */
k = select_hospital(H, P[i], Time); /* Unmark P[i], A[j], H[k] if necessary */

if (! fork() ) { /* Fork the A[j] as a concurrent agent */
A[j] ⇒ P[i]; /* Drive the ambulance to the patient */
P[i] ⇒ A[j]; /* Update P's time properties */
A[j] ⇒ H[k]; /* Get the patient on the ambulance */
H[k] ⇒ A[j]; /* Drive the Ambulance to the hospital */
A[j] ⇒ P[i]; /* Update P's time properties */
}
End;
```

그림 4. 구급차 이동 알고리즘

그림 4는 HAP 알고리즘으로 구급차 이동에 관한 알고리즘이다. HAP의 입력정보인 G는 CARDMI의 그래프 정보로 구급차, 환자, 병원의 공간과 행위, 시간정보를 의미하며 이 정보는 실 시간적으로 갱신 된다.

알고리즘은 응급환자가 존재하면 우선 순위가 가장 높은 환자부터 대기 중이던 최단거리에 있는 구급차를 선택하고 환자를 치료할 수 있는 최단거리에 있는 병원이 존재하면 선택된 구급차가 환자를 태우고 병원으로 이동(⇒) 하는 것으로 m개의 구급차들이 병렬적으로 환자가 존재하지 않을 때까지 임무를 수행한다.

4. CARDMI기반 u-EMS 예제 및 분석

4.1 CARDMI기반 u-EMS

u-EMS 예제는 시간속성이 추가된 t-CARDMI를 통해서 공간, 행위, 시간정보를 명세하고 이를 검증할 수 있다.

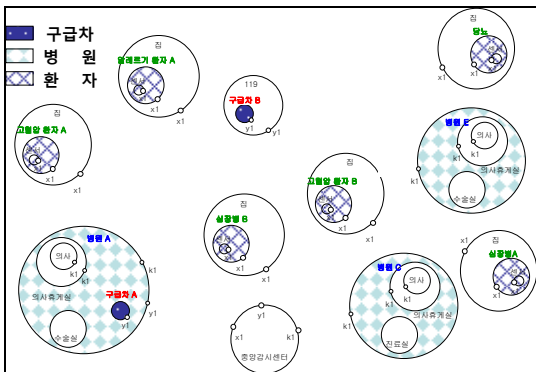


그림 5. u-EMS의 RHG 표현

그림 5는 u-EMS를 RHG(Reconfigurable Hyper Graph)[2]의 공간정보 그래프로 표현한 것으로 환자는 총 6명, 병원은 3곳, 구급차는 2대가 그림 5와 같은 공간에 위치한다.

병원 A, E는 2, 3명의 응급 환자만을 수용 가능하고 병원 C는 5명의 진료환자만을 수용 가능하다. 구급차 A는 병원 A에 위치하고 구급차 B는 병원 A, C, E의 중간 지점에 위치한다. 환자는 응급환자와 진료환자로 나눌 수 있으며 응급환자는 심장병 2명(심장병 A는 0분, 심장병 B는 10분에 발생), 고혈압 2명(고혈압 A는 0분, 고혈압 B는 5분에 발생), 당뇨 1명(0분에 발생)의 환자이며 진료환자로는 알레르기 1명(5분에 발생)의 환자가 존재하고 우선순위는 표 1을 따른다.

표 1. 환자의 우선순위

환자	우선순위	deadline	위험수위
심장병	1	10분	마지막 10분이 남아 있는 상황에서는 우선순위가 1로 된다.
고혈압	2	20분	
당뇨	3	20분	
알레르기	4	30분	

구급차가 환자 집까지 이동과 환자 집에서 병원으로의 이동 경로는 최소한의 시간이 소모되는 경로로 이동하기 때문에 예제에서는 경로를 추상화 하였다.

중앙 관리 센터에서는 환자의 병명에 따른 이동 시한(deadline)을 만족 시키기 위해서 환자, 병원, 구급차의 모든 경로에 대한 이동 시간을 표 2와 같이 알고 있어야 한다.

표 2. HAP의 모든 경로에 대한 이동 시간

	구급차 A ↔ 환자	구급차 B ↔ 환자	환자 ↔ 병원 A	환자 ↔ 병원 C	환자 ↔ 병원 E
심장병A	10분	7분	10분	X	3분
고혈압A	3분	5분	3분	X	8분
고혈압B	6분	3분	6분	X	2분
당뇨	15분	5분	15분	X	1분
알레르기	4분	1분	X	7분	X
심장병B	2분	5분	2분	X	5분

위의 정보를 바탕으로 u-EMS를 구성하는 요소를 CARDMI로 표현하면 그림 6과 같이 정의된다.

심장병환자A-센서 = $SP_1$	심장병환자B-센서 = $SP_2$	
심장병환자A = $P_1$	심장병환자B = $P_2$	
심장병환자A-집 = $HP_1$	심장병환자B-집 = $HP_2$	
고혈압환자A-센서 = $SP_3$	고혈압환자B-센서 = $SP_4$	
고혈압환자A = $P_3$	고혈압환자B = $P_4$	
고혈압환자A-집 = $HP_3$	고혈압환자B-집 = $HP_4$	
당뇨환자-센서 = $SP_5$	알레르기환자-센서 = $SP_6$	
당뇨환자 = $P_5$	알레르기환자 = $P_6$	
당뇨환자-집 = $HP_5$	알레르기환자-집 = $HP_6$	
병원A = $H_1$	병원C = $H_2$	병원E = $H_3$
병원A수술실 = $ORH_1$	병원C진료실 = $MOH_2$	병원E수술실 = $ORH_3$
병원A의사휴게실 = $LH_1$	병원C의사휴게실 = $LH_2$	병원E의사휴게실 = $LH_3$
병원A의사 = $DH_1$	병원C의사 = $DH_2$	병원E의사 = $DH_3$
구급차A = $A_1$		
119센터 = $119_1$		
구급차B = $A_2$		
중앙 관리 센터 = $Center$		

그림 6. u-EMS 구성 요소

그림 7은 u-EMS를 CARDMI로 정의 한 것으로 모든 환자와 구급차, 병원과 중앙 관리 센터는 병렬적으로 동작한다.

$$u-EMS = \left[ \begin{array}{c} \left[ \left[ \left[ \left[ SP_1 \right] \middle| P_1 \right] \middle| HP_1 \right] \middle| \left[ \left[ \left[ SP_2 \right] \middle| P_2 \right] \middle| HP_2 \right] \right] \right] \\ \left[ \left[ \left[ \left[ SP_3 \right] \middle| P_3 \right] \middle| HP_3 \right] \middle| \left[ \left[ \left[ SP_4 \right] \middle| P_4 \right] \middle| HP_4 \right] \right] \right] \\ \left[ \left[ \left[ \left[ SP_5 \right] \middle| P_5 \right] \middle| HP_5 \right] \middle| \left[ \left[ \left[ SP_6 \right] \middle| P_6 \right] \middle| HP_6 \right] \right] \right] \\ Center \\ \left[ 119, \left[ A_2 \right] \right] \\ \left[ \left[ H_1 \middle| A_1 \right] \middle| \left[ ORH_1 \right] \right] \middle| \left[ LH_1 \middle| \left[ DH_1 \right] \right] \right] \\ \left[ \left[ H_2 \middle| \left[ MOH_2 \right] \right] \middle| \left[ LH_2 \middle| \left[ DH_2 \right] \right] \right] \\ \left[ \left[ H_3 \middle| \left[ ORH_3 \right] \right] \middle| \left[ LH_3 \middle| \left[ DH_3 \right] \right] \right] \end{array} \right]$$

그림 7. u-EMS를 CARDMI로 정의

u-EMS에서 중앙 관리 센터는 환자의 발생에 따라 HAP 알고리즘을 통해 환자, 구급차, 병원이 결정되고 이에 대한 행위 정보를 실 시간적으로 스케줄링 한다. 이를 정의하면 아래와 같다.

$$Center = \{ m_{1, \dots, n} / z_{1, \dots, n} \} x_1(z_{1, \dots, n} : msg) \bullet HAP(m_{1, \dots, n}) \bullet \bar{y}_7(m_7) \bullet \bar{k}_7(m_7) \bullet \dots \bullet \bar{y}_7(m_7) \bullet \bar{k}_7(m_7) \bullet Center'$$

$x_1$  이라는 포트를 통해서 응급환자의 정보가 들어 오면 중앙 관리 센터에서는 환자의 정보를 가지고 즉시 HAP 알고리즘을 통해 스케줄링 하고 그 결과에 따라 구급차와 병원에 연락한다. 여기서  $\bar{y}_7(m_7) \bullet \bar{k}_7(m_7)$  은 HAP 알고리즘을

통해 결정된 환자 ( $m_7$ )에 대해서 특정 구급차 ( $\bar{y}_7$ )와 병원 ( $\bar{k}_7$ )에 환자에 대한 정보를 넘겨 구급차와 병원에 대한 행위를 결정한다.

그림 8은 u-EMS 예제의 일부분인 시간 0에서 응급환자 고혈압A, 심장병A, 당뇨환자 발생에 따른 HAP 스케줄링 과정이다.



그림 8. 시간 0에서의 고혈압A, 심장병A, 당뇨환자 발생에 따른 HAP 스케줄링 과정

## 4.2 u-EMS를 위한 HAP의 결과

4.1을 기반으로 응급 환자 발생에 따른 스케줄링의 결과는 그림 9과 그림 10과 같다.

0	→	당뇨(20)	고혈압 A(20)	심장병 A(10)	→	고혈압A(20), 심장병A(10)
5	→	알레르기(30)	당뇨(15)	고혈압 B(20)	→	
6	→		알레르기(29)	당뇨(14)	→	고혈압B(19)
10	→		알레르기(25)	심장병(10)	→	당뇨(10)
11	→			알레르기(24)	→	심장병(9)
15	→			알레르기(24)	→	알레르기(20)

그림 9. 환자 큐(Queue)의 결과

그림 9은 환자의 우선순위가 큐(Queue)에서 시간에 따라 변하는 과정을 나타내고 있으며 그림 9, 10는 표 2를 바탕으로 우선순위가 높은 환자부터 가장 가까운 곳에 위치한 구급차와 해당 환자를 치료할 수 있고 가까운 곳에 위치한 병원을 선택하여 이동하는 과정을 그림으로 나타낸 것이다.

분석 및 검증을 하도록 한다.

참고 문헌

[1] 송현중, "응급의료서비스체계의 현황과 정책과제," 한국보건사회연구원, 보건복지포럼 통권 제90호, pp. 63~73, 2004.  
 [2] J. Choi, "A Calculus for Equivalence Analysis and Verification of Distributed Mobile System Based on Abstraction." PhD Dissertation, Chonbuk national Univ, 2007.  
 [3] 온진호, 공간 프로세스 대수에서 시간속성의 명세와 분석을 위한 정형기법, 석사학위논문, 전북대학교, 2007.  
 [4] 온진호, 우수정, 최정란, 이문근, "공간 프로세스 대수에서 시간 속성의 명세와 분석을 위한 정형기법," 한국통신학회 추계종합학술발표회 논문집 제 36권 제 1호, pp.318-318, 2007.

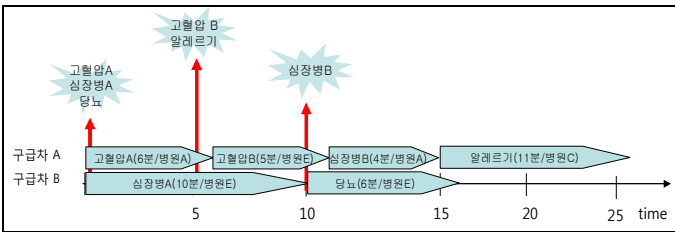


그림 10. u-EMS를 위한 HAP의 진행 과정

결과적으로 u-EMS를 위한 RHG와 CARDMI의 정의, 그리고 HAP 스케줄러를 통해 공간과 행위정보를 명세하고 이를 t-CARDMI를 통해서 그림 11와 같은 분석 및 검증이 가능하다.

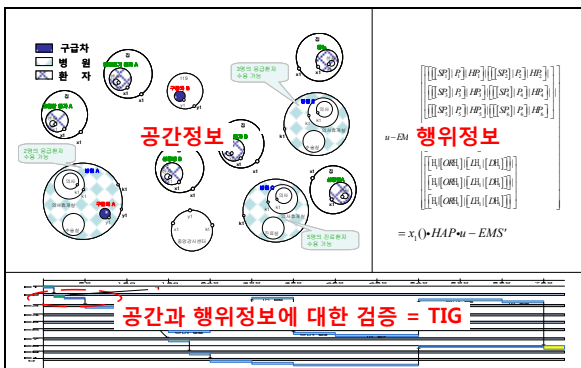


그림 11. u-EMS를 위한 HAP의 결과

5. 결론 및 향후 연구

본 논문에서는 u-EMS에서 발생할 수 있는 여러 가지 문제점을 자동화된 중앙 관리 체계를 기반으로 한 u-EMS의 시공간 환경과 행위를 명세 및 분석하기 위한 CARDMI의 HAP 스케줄러를 통해 해결하였다.

향후 연구에서는 CRA에서 발생할 수 있는 성능 문제를 해결하고 실 시간적인 시뮬레이션을 위한 도구 개발 및 CARDMI에 확률을 도입하여 시스템을 좀더 효과적으로 명세,