DOI: 10.3745/KIPSTD.2009.16-D.3.339

공간 프로세스 대수를 이용한 정형 명세와 분석에서의 시간속성의 시각화

온 진 호[†] · 최 정 란[††] · 이 문 근[†††]

요 약

유비쿼터스 컴퓨팅 환경에서 분산된 실시간 시스템의 행위와 공간, 시간 속성을 분석하고, 검증하기 위한 다양한 정형기법들이 존재한다. 그러나 대부분의 경우 공간과 행위를 같이 표현하는 구조적, 근본적 한계가 존재한다. 게다가 시간 속성이 포함되는 경우는 더욱 복잡해지게 된다.

이러한 한계를 해결하기 위하여 본 논문은 *Timed Calculus of Abstract Real-Time Distribution, Mobility and Interaction*(t-CARDMI)라는 새로운 정형기법을 제안한다. t-CARDMI는 행위의 표현으로부터 공간정보의 표현을 분리시켜 복잡도를 단순화 시키며, 시간 속성에 대해서 오직 행위적 표현에서만 허용하여 복잡한 명세를 덜 복잡하게 표현한다. t-CARDMI는 대기기간, 실행시작 만족시간, 실행시간, 실행완료 만족시간 등의 특유의 시간속성을 이동과 통신의 행위에서 모두 포함하는 특징을 갖는다. 새롭게 제안된 *Timed Action Graph*(TAG)는 공간과 시간을 포함하는 시스템의 명세를 분석하고 검증하기 위해서 공간과 시간속성을 2차원의 다이어그램으로 표현하며 그 안에서 이동과 통신의 정보를 분산된 그림정보로 표현하는 그래프로 t-CARDMI를 좀더 효율적으로 명세하고 분석할 수 있는 방법을 제공한다.

t-CARDMI는 유비쿼터스 컴퓨팅에서의 분산된 실시간 시스템의 공간적, 행위적, 시간적 속성에 대한 명세, 분석 및 검증에 매우 효율적이고 효과적인 혁신적인 정형기법의 하나로 고려될 수 있다.

본 논문은 t-CARDMI의 문법과 의미, TAG 그리고 Specification, Analysis, Verification, and Evaluation (SAVE)로 명명된 툴을 제안하고 유비쿼터스 헬스케어 시스템 예제를 통해 효율성을 분석한다.

키워드 : 정형기법, 공간프로세스 대수, 시간속성, 카드미, 시간속성 그래프

Visual Representation of Temporal Properties in Formal Specification and Analysis using a Spatial Process Algebra

Jinho On[†] · Jungrhan Choi[††] · Moonkun Lee[†††]

ABSTRACT

There are a number of formal methods for distributed real-time systems in ubiquitous computing to analyze and verify the behavioral, temporal and the spatial properties of the systems. However most of the methods reveal structural and fundamental limitations of complexity due to mixture of spatial and behavioral representations. Further temporal specification makes the complexity more complicate.

In order to overcome the limitations, this paper presents a new formal method, called *Timed Calculus of Abstract Real-Time Distribution, Mobility and Interaction*(t-CARDMI). t-CARDMI separates spatial representation from behavioral representation to simplify the complexity. Further temporal specification is permitted only in the behavioral representation to make the complexity less complicate. The distinctive features of the temporal properties in t-CARDMI include waiting time, execution time, deadline, timeout action, periodic action, etc. both in movement and interaction behaviors. For analysis and verification of spatial and temporal properties of the systems in specification, t-CARDMI presents *Timed Action Graph* (TAG), where the spatial and temporal properties are visually represented in a two-dimensional diagram with the pictorial distribution of movements and interactions.

t-CARDMI can be considered to be one of the most innovative formal methods in distributed real-time systems in ubiquitous computing to specify, analyze and verify the spatial, behavioral and the temporal properties of the systems very efficiently and effectively. The paper presents the formal syntax and semantics of t-CARDMI with a tool, called SAVE, for a ubiquitous healthcare application.

Keywords : Formal Methods, Spatial Process Algebra, Temporal Properties, CARDMI, Timed Graph

1. 서 론

유비쿼터스 시대에는 주변에 있는 다양한 공간에 컴퓨터가 배치되어 각자의 역할을 수행한다. 이러한 환경에서 각각의 센서나 소형 컴퓨터 노드들은 분산되어 병렬적으로 동

※ 본 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-521-D00451).
† 준 회 원 : 전북대학교 컴퓨터공학 박사과정
†† 정 회 원 : 고려대학교 BK21 소프트웨어 산학연 연구교수
††† 정 회 원 : 전북대학교 전자정보공학부(컴퓨터공학과 전공) 교수
논문접수 : 2008년 10월 16일
수 정 일 : 1차 2008년 12월 12일, 2차 2009년 4월 6일, 3차 2009년 5월 19일
심사완료 : 2009년 5월 19일

작될 수 있고, 동적으로 이동하며 각각의 노드들끼리 통신을 수행하며, 전송된 메시지를 통해 제어되고, 재구성 될 수 있는 특징을 지닌다.

이와 같은 유비쿼터스 컴퓨팅 환경은 컴퓨터들의 역할 및 설계된 시스템의 안정성에 대한 명세, 분석, 검증, 구현에 있어 매우 높은 복잡도를 요구하고 있다.

복잡한 시스템의 분석과 검증을 위해 Zed[1], Esterel[2], VDM[3]과 같은 수많은 Real-time 정형기법들이 존재한다. 하지만, (그림 1)에서 보는 바와 같이 기존의 정형기법들은 각 프로세스들의 시간에 따른 행위, 복잡도를 낮추기 위한 다양한 추상화 방법들에 연구의 방향을 맞추고 있으며, 공간 개념 안에서의 프로세스와 에이전트의 이동성을 제공하고 있지는 않다.

유비쿼터스를 위한 정형기법에서는 공간과 이동성의 표현은 필수 요소이며, 이와 같은 분산, 이동, 실시간 시스템(DMRS: *Distributed Mobile Real-time Systems*)의 명세를 위해 제안된 대표적인 정형기법으로는 π-calculus[4-5], mobile ambients[6], bigraph[7-8] 등이 존재한다. 프로세스 대수 기반의 이러한 정형기법을 통해 DMRS의 핵심 모듈의 행위에 대한 요소들을 파악할 수 있고 설계와 검증을 통한 안정성을 보장할 수 있다.

하지만, 실시간 속성이 잘 정의된 기존의 timed CCS[9], timed π-calculus[10]과 같은 정형기법들은 DMRS의 명세에는 한계점들이 존재한다. 또한, πRT-calculus[11], *Discrete time mobile ambient calculus*(DTMA)[12]는 π-calculus와 mobile ambients에 시간 속성을 추가한 정형기법이지만, 정확한 시간속성을 갖는 에이전트의 통신과 이동을 표현하기에는 다음과 같은 부적합성이 존재한다:

1) πRT-calculus는 π-calculus를 기본으로 하고 있어 공간정보를 정의하고 있지 않고, 시간의 흐름을 타임아웃을 통한 시간의 흐름으로만 표현하고 있다.
2) DTMA 또한 bound된 시간구역 e와 e' 사이에 존재하는 특정 시간에 행위가 이뤄짐을 표현하고 있을 뿐 다양한 시간속성에 대해서는 정의하고 있지 않다.
3) 앞에서 제시된 프로세스 대수들은 실행시간을 즉각적인 시간으로 정의하였다. 하지만, 이동의 행위는 즉각적인 실행이 어렵고 시간의 소모를 명시해야 할 필요성이 존재한다. 특히 물리적 공간에서의 즉각적인 이동은 실제와는 맞지 않다.

본 논문에서는 이동하며 동일한 시간을 공유하는 시스템의 명세를 위한 시간 속성을 기존의 *Calculus of Abstract Real-Time Distribution, Mobility and Interaction*(CARDMI)[13]에 추가하였다. tick-time을 기반으로 하며, 시간의 속성을 이동과 통신을 수행하는 에이전트에 맞도록 대기시간, 실행시작 만족시간, 실행시간, 실행완료 만족시간으로 분류하고, 주기/비주기적 시간을 지니는 속성을 추가하였다. 이러한 세부 시간속성을 통해 다른 정형기법에서는 제공하지 못하는 즉각적인 실행과 실행시간이 있는 실행을 명세할 수 있으며, 시간에 따른 반복적인 행위를 표현할 수 있는 장점을 지닌다.

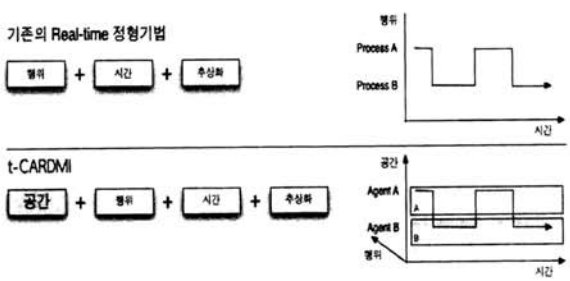
DMRS를 명세하기 위한 언어들은 공간, 행위, 시간이 혼재되어 있어 이해와 분석이 매우 어려운 단점이 존재한다.

CARDMI는 위와 같은 문제를 해결하기 위하여 시스템을 행위정보와 공간정보로 구분하여 명세 하는 정형기법으로 공간정보와 에이전트들의 행위정보(이동, 통신 등)를 구분하였고, 행위는 이동과 인터액션으로 구분하였다. 진화하는 DMRS에서 에이전트의 순차성, 병렬성, 분산성, 이동성 등에 대한 모델링, 추상화, 동일성 검증 등을 제안하였다.

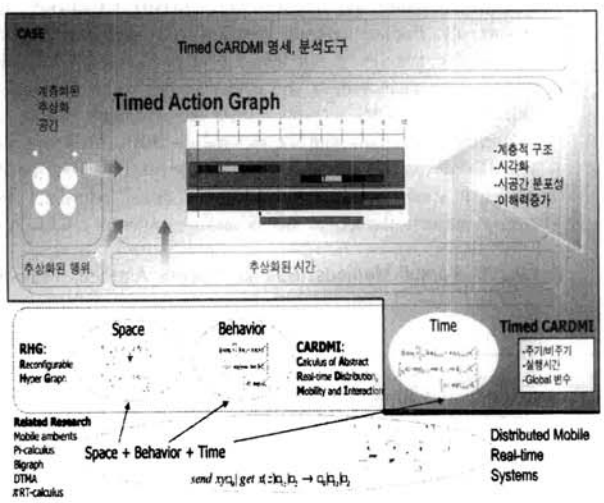
본 논문은 (그림 2)와 같이 DMRS의 공간과 행위적 특성들을 명세하기 위해 제안된 CARDMI에 시간속성을 추가한 프로세스 대수 기반의 정형기법(t-CARDMI)과 이를 분석하기 위한 시간 그래프(TAG), 그리고 이를 지원하기 위한 CASE 도구인 *Specification, Analysis, Verification, and Evaluation* (SAVE)를 제안한다.

TAG 는 시간 흐름기반의 에이전트의 행위를 가시적으로 표현할 수 있으며, 행위에 대한 공간, 시간을 동시에 표현할 수 있는 장점을 지닌다. 또한 계층적 공간에 대한 시간명세 방법을 제공하며, 시간/공간/행위를 추상화할 수 있도록 정의되었다. SAVE는 공간정보와, TAG, 텍스트 기반 명세를 모두 포함하며, 명세, 수정, 분석, 검증에서의 일관성을 유지할 수 있는 장점을 지닌다. 또한, t-CARDMI에 최적화된 GUI 명세방법을 제안한다.

이러한 방법은 공간과 시간, 그리고 행위로 구분되어 있었던 명세에 대해 통합된 명세환경을 제공하여 복잡한 DMRS에 대한 분석 및 검증을 용이하게 할 수 있다.



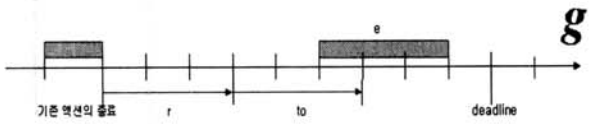
(그림 1) 기존 Real-time 정형기법과의 비교



(그림 2) 논문의 개요

2. Timed CARDMI (t-CARDMI)

CARDMI에 구간(interval) 개념의 시간 속성을 추가하여 확장하였다. CARDMI는 기존의 프로세스 대수[4-5]의 전통적인 명세 방법을 그대로 따르고 있으며, 이러한 이유로 문법과 각 명세가 지니는 의미로 표현된다. Act는 에이전트의 인터액션과 이동 행위를 의미하며, 시간 속성이 추가되어 Act_T로 표현된다. 이러한 시간의 특성은 (그림 3)과 같다. t-CARDMI에서의 시간은 이산적(Discrete)이며, 양의 정수로 표현된다.



(그림 3) 일반적인 시간의 흐름

2.1 시간의 정의

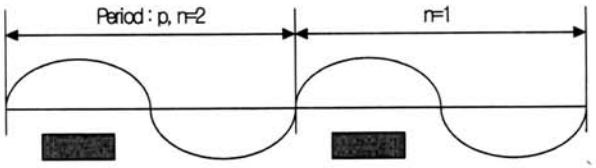
[정의 2.1] Timed Action

$$Act_T := {}_{[g]}Act^{p,n}_{[r,to,e,d]}$$

[정의 2.1]에서 Act_T에 대한 각각의 설명은 다음과 같다:

- 시간 g는 글로벌 타이머의 시간을 나타내며, 명세에서는 행위가 어떠한 시점에서 시작되어야 하는지를 명시하며, 실행의 순간에는 시작 이후에 변경되는 타이머의 시간을 나타낸다. g가 생략되면 행위의 시작 시점이 모든 시간에 대해 오픈 되어 있음을 나타내며, 실행의 순간에 결정되어 짐을 나타낸다.
- r은 액션이 실행되기 위해 최소한으로 만족되어야 하는 준비/대기 시간을 나타낸다.
- to는 타임아웃(timeout)의 줄임 표현으로 준비시간 r이 지난 후 액션이 실행되기 위해 대기할 수 있는 최대 시간을 나타낸다. Tick이 한번 씩 흐를 때마다 to은 0부터 증가하며, 최대 timeout 시간을 초과하면, 액션은 실패한다.
- e는 행위가 실행되는데 소모되는 시간을 나타낸다. 실행시간이 0으로 표현되면, 행위의 실행이 순간적으로 실행됨을 의미한다.
- d는 데드라인(deadline)을 나타내며 액션이 실행된 시점부터 d까지의 시간을 말하며, 타임아웃 안에 시작된 행위가 실행 시간 e만큼을 소모한 후 d안에 완료되어야 함을 의미한다.
- p와 n은 주기적인 행위를 위한 식별자로 그 행위는 그림 4와 같고, 특성은 다음과 같다. p는 Act의 총 실행 주기를 말하며, $p=0$이면 생략 가능하다. 또한, n은 p의 주기를 n번 반복함을 나타나며, $n=0$이면 생략 가능하다.

${}_{[g]}Act_{[r,to,e,d]}$은 행위가 g시간에 동작하기 시작하여 r의 시간을 준비, 대기한 후 to안에 동작이 시작되어 e만큼의 실행시간 최소 실행시간 동안 실행을 한 후 d안에 완료해야 함을 나타낸다.



(그림 4) 주기적 실행의 의미

[정의 2.2] 시간의 간격

$[r,to,e,d]$: 실행시간 g이상의 시간에 실행되어 d시간 이하에 실행이 종료됨.

$(r,to,e,d]$: 실행시간 g초과의 시간에 실행되어 d시간 이하에 실행이 종료됨.

$[r,to,e,d)$: 실행시간 g이상의 시간에 실행되어 d시간 미만에 실행이 종료됨.

(r,to,e,d) : 실행시간 g초과의 시간에 실행되어 d시간 미만에 실행이 종료됨.

2.2 문법(Syntax)

[정의 2.3]은 t-CARDMI에서 에이전트의 이동 및 인터액션을 정의하기 위해 필요한 문법을 의미한다. 기본적인 문법은 CARDMI[13]의 문법을 따르며, 시간 속성이 추가된 행위에 대해서만 기술하였다.

[정의 2.3] Syntax

(1) Primitive action:

$$E_T := {}_{[g]}x(Z)^{p,n}_{[r,to,e,d]} \bullet E_T \mid {}_{[g]}\overline{x}(M)^{p,n}_{[r,to,e,d]} \bullet E_T \mid E_T \setminus (vx) \mid E_{T_1} + E_{T_2} \mid 0 \mid \varepsilon \mid \mathfrak{M} \mid_{[g]} [E_T]_{[r,d]} \mid < E_{T_1}, E_{T_2} > \bullet E_T,$$

(2) Composition: $\varepsilon := E_T \mid E_T \mid E_T \mid E_T$

(3) Movement:

$$\mathfrak{M} := {}_{[g]}in\ A^{p,n}_{[r,to,e,d]} \bullet E_T \mid {}_{[g]}out\ A^{p,n}_{[r,to,e,d]} \bullet E_T$$

(4) Communication Entity:

Send: $M := m', M \mid m'$

$m' := A \mid x_i \mid y_i \mid m_i$

Receive: $Z := z', Z \mid z'$

$z' := z : agent \mid z : msg \mid z : port$

에이전트의 동작 수식에 대한 표기의 의미는 다음과 같다: '0'은 동작의 종료나 아무 동작을 수행하지 않는 상태를 나타내며, 선행 동작을 위한 '•', 선택을 위한 '+', 제한을 위한 '\'. 이들의 의미적 해석은 다음과 같다.

(1) Primitive action

• 선행 형식,

$$_{[g]}x(Z)^{p,n}_{[r,to,e,d]} \bullet E_T, \quad _{[g]}\overline{x}(M)^{p,n}_{[r,to,e,d]} \bullet E_T$$

시간정의에 따라 선행형식은 다음과 같은 행위를 수행한다. 선행 형식에서의 행위는 g의 시간에 $x(Z)$, $\overline{x}(M)$의 동작을 $[r,to,e,d]$를 만족하도록 수행하고, E_T의 행위를 한다. 이 때 E_T의 시간은 선행행위의 시간보다 앞이 될 수 없다.

• 선택, $E_{T_1} + E_{T_2}$

기본 CARDMI가 행위의 실행만을 생각하였다면, t-CARDMI에서는 같거나 혹은 다른 시간대에 존재하는 여러 개의 행위 중 한 개의 비결정적 행위에 대한 선택을 의미한다. 선택 행위에서의 시간은 동일한 동작을 수행하는 행위에 대해서도 시간에 따라 다른 행위로 인식된다. 예를 들면 $_{[1]}x(Z)_{[1,2.0.3]}$, $_{[2]}x(Z)_{[1,2.0.3]}$와 같이 동일한 행위를 수행하는 두 개의 행위도 $_{[1]}x(Z)_{[1,2,0,3]} +_{[2]} x(Z)_{[1,2,0,3]}$와 같이 선택행위로 동작할 수 있다.

• Global 시간속성, $_{[g]}[E_T]_{r,d}$

행위들의 집합에 대한 전체적인 시간속성을 표시, g는 전체 행위가 시작되어야 하는 시간, r은 행위집합의 첫 번째 행위의 준비시간, d는 행위집합 안의 마지막 행위가 만족해야 하는 완료 만족시간을 나타낸다. 즉, 연속적으로 구성된 행위들에 대한 시간 속성을 명세 한다.

• Timeout (Alternative), $< E_{T_1}, E_{T_2} > \bullet E_{T_3}$

$< E_{T_1}, E_{T_2} > \bullet E_{T_3}$은 E_{T_1}의 행위가 정의된 시간을 만족하는 경우 E_{T_3}의 상태로 전이되고, E_{T_1}의 시간속성을 만족하지 못하는 경우, E_{T_2}의 행위를 수행함을 의미한다.

• Restriction, $E_T \setminus (vx)$

E와 같이 동작을 하는 에이전트는 x와 같은 포트 이름들로의 통신이 외부로부터 제한되었음을 의미한다.

(2) Composition: 병렬 결합, $E_T \mid E_T \mid E_T \mid E_T$

"|"는 병렬적으로 돌아가는 에이전트를 나타낸다. 싱크 통신을 수행하는 에이전트들은 "$\dot{|}$" 로 표시하며, 통신이 수행되기 위해서는 두 개의 에이전트가 최소한 to시간 중 한 부분이 겹쳐 있어야 한다. 실행시간이 0이 아닌 경우의 동기 통신은 두 에이전트의 실행시간 e는 동일한 시간이어야 한다.

(3) Movement: 이동($\mathfrak{M}$)

$$_{[g]}in\ A^{p,n}_{[r,to,e,d]} \bullet E_T, _{[g]}out\ A^{p,n}_{[r,to,e,d]} \bullet E_T$$

이동행위는 특정 시간 g에 시작하여 r의 시간을 대기한 후 to의 시간 안에 e의 실행시간을 갖는 행위를 시작하여 d안에 이동 동작을 완료한다.

주기적인 이동행위 p에서는 이동 동작을 수행하면서 최초의 공간으로 돌아올 수 있는 행위만 적용될 수 있다.

(4) Communication Entity

M은 에이전트가 보낼 수 있는 통신 엔티티로서, 에이전트(A), 포트(x_i, y_i), 메시지(m_i) 등이다. 이들은 동시에 혹은 각각 여러 개의 메시지를 보낼 수 있다. Z는 입력으로 받을 수 있는 엔티티로 종류가 3가지로 분리되어 있다. 이들은 각각 에이전트를 받을 수 있는 $z:agent$, 포트 이름을 받을 수 있는 $z:msg$, 메시지를 받을 수 있는 $z:port$로써 정의된다.

2.3 t-CARDMI의 의미 (Semantics)

$$E_T \xrightarrow{\varphi_T} E_T^{'}$$

φ_T의 동작을 수행하고 E_T가 E_T'으로 전개됨을 의미한다. 전이 시스템은 상태와 전이의 두 가지 기본 개념만으로 구성되며, 병행 시스템의 명세와 검증을 위해 광범위하고 전통적으로 사용된 방법이다. t-CARDMI는 전이 시스템으로 의미론을 표현함으로써 시스템을 이루는 하나의 에이전트가 행위를 실행하기 위해 어떠한 전제조건상에서 어떠한 조건을 만족하며 행위를 실행하고 어떠한 상태로 전이되는지에 대한 각 문법의 의미를 설명하고 있다. 또한, 전이 시스템을 사용하여 각 상태의 전이 과정에서의 나타나는 상태의 동일성[4-5]을 Bisimulation[4-5]을 사용하여 검증할 수 있다. 기본 의미론은 CARDMI[13]의 의미론을 따르며, 본 논문에서는 시간 속성이 추가된 행위에 대해서 설명한다.

전이는 다음과 같은 규칙 형태의 집합으로 정의된다.

$$\frac{\text{전제}}{\text{결론}}(\text{조건})$$

이의 의미는 다음과 같다:

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

[정의 2.4] 기본 동작의 전이 규칙들

에이전트 A의 의미는 네 개의 튜플로 구성된다. $< E_T, A, \varphi_T, \rightarrow >$. 여기에서 E_T는 정의한 문법을 통해 표현된 에이전트의 이동 규칙 명세와 관한 수식이고, 전이관계 $\rightarrow \subseteq E_T \times \varphi_T \times E_T$는 아래 규칙들을 만족한다:

(1) Tick-time:

$$\frac{}{_{[g]}E_{T[r,to,e,d]} \xrightarrow{\triangleright_i} {}_{[g+i]}E_{T[r-i,to,e,d-i]}}(r-i \geq 0)\ or$$

$$\frac{}{_{[g]}E_{T[r,to,e,d]} \xrightarrow{\triangleright_i} {}_{[g+i]}E_{T[0,to-i,e,d-i]}}(to-i \geq 0)$$

(2) Timeout:

$$\frac{{}_{[g]}\varphi_{[r,to,e,d]} \bullet E_{T1} \xrightarrow{|g-i|\varphi} E_{T1}}{< {}_{[g]}\varphi_{[0,to,e,d]} \bullet E_{T1}, E_{T2} > \bullet E_{T3} \xrightarrow{|g-i|\varphi} E_{T3}}\begin{pmatrix} g' = g+e, \\ to \geq 0 \end{pmatrix}\ or$$

$$\frac{{}_{[g_2]}\varphi_{[r_2,to_2,e_2,d_2]} \bullet E_{T2} \xrightarrow{|g_2-g_1|\varphi} E_{T2}}{< {}_{[g_1]}E_{T1[r_1,to_1,e_1,d_1]}, {}_{[g_2]}\varphi_{[r_2,to_2,e_2,d_2]} \bullet E_{T2} > \bullet E_{T3} \xrightarrow{|g_2-g_1|\varphi} E_{T2}}\begin{pmatrix} g_2' = g_2+e_2, \\ to_1 < 0, to_2 \geq 0 \end{pmatrix}$$

(3) T-Repetition:

$$\frac{}{{}_{[g]}\varphi^{p,n}_{[0,to,e,d]} \bullet E_T \xrightarrow{[g+e]\varphi} {}_{[g+e]}\varphi^{p,n-1}_{[r,to,e,d]} E_T} \begin{pmatrix} n \geq 1, p > 0 \ \ or \\ to \geq 0 \end{pmatrix}$$

(4) T-Input-act:

$$\frac{}{{}_{[g]}x(Z)_{[r,to,e,d]} \bullet E_T \xrightarrow{[g+e]x(M)} (\{M/Z\}E_T')} \begin{pmatrix} x \in fn(E_T) \wedge M \notin fn(E_T) \ or \\ x \in bn(E_T) \wedge M \notin bn(E_T), \\ to \geq 0 \end{pmatrix}$$

(5) Input-ported act:

$$\frac{}{{}_{[g]}x(z:port)_{[r,to,e,d]} \bullet E_T \xrightarrow{[g+e]x(y)} \{y'/y\}\{y/z\}E_T'} \begin{pmatrix} x, y \in fn(E_T) \ or \\ x, y \in bn(E_T) \notin bn(E_T) \\ to \geq 0 \end{pmatrix}$$

(6) Output-act:

$$\frac{}{{}_{[g]}\overline{x}(M)_{[r,to,e,d]} \bullet E_T \xrightarrow{[g+e]\overline{x}(M)} E_T} (to \geq 0)$$

(7) Summation (choice):

$$\frac{{}_{[g_j]}E_{Tj[r_j,to_j,e_j,d_j]} \xrightarrow{[g_j+e_j]\varphi_j} E'_{Tj}}{\sum_{i\in I} {}_{[g_i]}E_{Ti[r_i,to_i,e_i,d_i]} \xrightarrow{[g_j+e_j]\varphi_j} E'_{Tj}} \begin{pmatrix} 1 \leq i \leq n, \\ to_j \geq 0 \end{pmatrix}$$

(8) Restriction:

$$\frac{E_T \xrightarrow{\alpha} E'_T}{E_T \setminus x \xrightarrow{\alpha} E'_T \setminus x} (x \notin ns(\alpha))$$

(9) Parallel:

$$\frac{{}_{[g]}E_{T1[r,to,e,d]} \xrightarrow{[g+e]\varphi} E'_{T1}}{{}_{[g]}E_{T1[r,to,e,d]} \mid E_{T2} \xrightarrow{[g+e]\varphi} E'_{T1} \mid E_{T2}} \begin{pmatrix} ns(\varphi) \cap ns(E_2) = \varnothing \\ to \geq 0 \end{pmatrix}$$

$$\frac{{}_{[g]}E_{T2[r,to,e,d]} \xrightarrow{[g+e]\varphi} E'_{T2}}{E_{T1} \mid {}_{[g]}E_{T2[r,to,e,d]} \xrightarrow{[g+e]\varphi} E_{T1} \mid E'_{T2}} \begin{pmatrix} ns(\varphi) \cap ns(E_1) = \varnothing \\ to \geq 0 \end{pmatrix}$$

(10) Close sync-communication:

$$\frac{{}_{[g_1]}E_{T1[r_1,to_1,e_1,d_1]} \xrightarrow{[g_1+e_1]\overline{x}(M)} E'_{T1}, {}_{[g_2]}E_{T2[r_2,to_2,e_2,d_2]} \xrightarrow{[g_2+e_2]x(Z)} E'_{T2}}{{}_{[g_1]}E_{T1[r_1,to_1,e_1,d_1]} \mid {}_{[g_2]}E_{T2[r_2,to_2,e_2,d_2]} \xrightarrow{[g]\tau} (E'_{T1} \mid \{M/Z\}E'_{T2}) \setminus x} \begin{pmatrix} x \in \{bn(E_{T1}) \cap bn(E_{T2})\} \\ to_1 \geq 0, to_2 \geq 0 \\ g_1 = g_2 \end{pmatrix}$$

(11) Open sync-communication:

$$\frac{{}_{[g_1]}E_{T1[r_1,to_1,e_1,d_1]} \xrightarrow{[g_1+e_1]\overline{x}(M)} E'_{T1}, {}_{[g_2]}E_{T2[r_2,to_2,e_2,d_2]} \xrightarrow{[g_2+e_2]x(Z)} E'_{T2}}{{}_{[g_1]}E_{T1[r_1,to_1,e_1,d_1]} \mid {}_{[g_2]}E_{T2[r_2,to_2,e_2,d_2]} \xrightarrow{[g]\tau(y)} (E'_{T1} \mid \{M/Z\}E'_{T2}) \setminus x} \begin{pmatrix} x \in \{fn(E_1) \cap fn(E_2)\} \\ to_1 \geq 0, to_2 \geq 0 \\ g_1 = g_2 \end{pmatrix}$$

(12) In-movement:

$$\frac{}{{}_{[g]}in\ A_{[r,to,e,d]} \bullet E_T \xrightarrow{[g+e]in\ A} E_T} \begin{pmatrix} (A \sqsupseteq Agt(E_T)) \in \{I_p of\ Agt(E_T) \cap I_p\ of\ A\}, \\ to \geq 0 \end{pmatrix}$$

(13) Out-movement:

$$\frac{}{{}_{[g]}out\ A_{[r,to,e,d]} \bullet E_T \xrightarrow{[g+e]out\ A} E_T} \begin{pmatrix} (\overline{A} \sqsupseteq Agt(E_T)) \in I_p\ of\ Agt(E_T), \\ to \geq 0 \end{pmatrix}$$

(14) Nested in-movement:

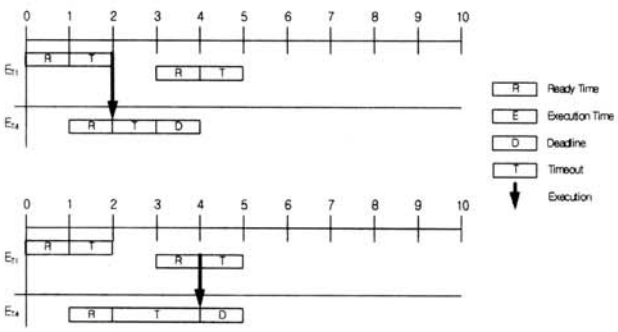
$$\frac{\left[{}_{[g]}in\ A_{[r,to,e,d]} \bullet E_{T2}\right] \xrightarrow{[g+e]in\ A} [E_{T2}]}{[E_{T1}] \mid \left[{}_{[g]}in\ A_{[r,to,e,d]} \bullet E_{T2}\right] \xrightarrow{[g+e]in\ A} [E_{T1}] \mid [E_{T2}]} \begin{pmatrix} Agt(E_{T1}) = A \\ (A \sqsupseteq Agt(E_{T2})) \in \{I_p of\ Agt(E_{T1}) \cap I_p\ of\ A\}, \\ to \geq 0 \end{pmatrix}$$

(15) Nested out-movement:

$$\frac{\left[{}_{[g]}out\ A_{[r,to,e,d]} \bullet E_{T2}\right] \xrightarrow{[g+e]out\ A} [E_{T2}]}{\left[E_{T1} \mid \left[{}_{[g]}out\ A_{[r,to,e,d]} \bullet E_{T2}\right]\right] \xrightarrow{[g+e]in\ A} [E_{T1}] \mid [E_{T2}]} \begin{pmatrix} Agt(E_{T1}) = A \\ (\overline{A} \sqsupseteq Agt(E_{T2})) \in I_p\ of\ Agt(E_{T2}) \\ to \geq 0 \end{pmatrix}$$

- 모든 action은 r시간 이후 $to \geq 0$일 때 가능하다.
- tick-time은 $r - i \geq 0$인 경우 대기시간 안에서 $\triangleright i$만큼의 tick이 발생되어 현재시간 g가 $g+i$로 증가되고, 대기시간이 i만큼 감소된 것을 의미하고, 이때 액션의 완료가 만족해야 하는 시간 d도 $d-i$로 변하는 것을 의미한다. 또한 액션 시작 시간 to에서 발생한 tick의 의미는 행위의 실행 타임아웃 시간에서 i만큼의 tick이 흐른 것을 의미한다($to - i \geq 0$). 다른 모든 행위들은 tick-time에서의 시간의 흐름을 전제로 한다. 행위가 실행되었다는 것은 tick-time에 의해 시간이 정상적인 실행시간 안에서 동작된 것을 의미한다.
- Timeout은 $to \geq 0$인 경우의 정상실행과 $to_1 < 0$, $to_2 \geq 0$일 때 실행 실패의 복구 동작으로 나눠진다. 정상적인 실행은 E_{T1}에서의 시간 속성을 만족하여 정상적으로 실행된 것을 의미하고, 실패의 복구 동작은 E_{T1}의 시간 속성을 만족하지 못하여, 복구 행위인 E_{T2}가 실행된 것을 나타낸다.

$< E_{T_1[1,1,0,2]}, E_{T_2[1,0,0,1]} > \bullet E_{T_3[1,1,0,2]}$와 같은 경우 (그림 5)와 같이 정상적인 실행 E_{T1}의 타임아웃 시간 안에 행위가 시작



(그림 5) Timeout 시간분석(위: 정상실행, 아래: 실패)

되지 않으면, 오류처리 행위 E_{T2}가 실행된다. 이때는 E_{T1}의 타임아웃 시간 이후 E_{T2}의 g가 0이면 즉시 실행되며, 그렇지 않은 경우는 E_{T2}의 g시간에 행위가 시작된다.

- Repetition은 p의 시간 주기로 n만큼의 행위를 반복함을 나타내며, 다음 실행은 갱신된 g에 기존의 시간 정의에 따라 반복된다. n이 1이면 일반적인 실행을 의미한다.
- Summation(Choice)는 가정에서 특정 동작의 전이가 발생하였다면 선택전이는 가정에서 발생된 $_{[g_i+e_i]}\varphi_j$가 수행한 후 전개된 행위 수식으로 전이한다.

$$_{[2]}E_{[1,1,0,3]} +_{[7]} E_{[1,1,0,2]}$$

(그림 6)과 같이 어떤 행위에 대한 선택은 해당 행위들 중에 하나가 실행되는 경우를 나타낸다. 둘 중의 하나의 행위가 실행되도록 명세 되지만, 시간적으로 각 행위들은 다른 시간대에 존재할 수 있다. 예제는 $_{[2]}E_{[1,1,0,3]}$와 $_{[7]}E_{[1,1,0,2]}$의 선택 행위 중 4의 단위 시간에 $_{[2]}E_{[1,1,0,3]}$의 행위가 실행된 것이다.

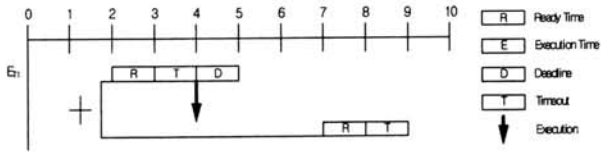
- Parallel은 병렬적으로 돌아가는 에이전트들의 행위를 말한다.

$$_{[2]}E_{[1,1,0,3]} \mid_{[7]} E_{[1,1,0,2]}$$

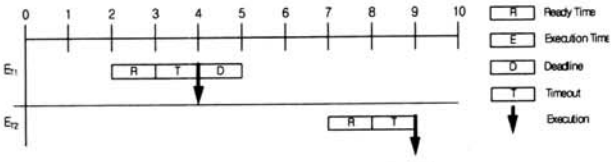
선택과 마찬가지로 여러 개의 에이전트들이 수행되는 시간은 동일한 시간대일 수 있고, 다른 시간대일 수도 있다. Parallel의 의미는 동일한 시간에 동시에 실행된다는 의미이지만, 시간속성이 추가되면서 동시에 실행될 수 있는 특성을 의미하는 것이지, 동시에 2개의 에이전트가 실행되고 있다라고는 말할 수 없다. (그림 7)의 예제는 병렬적으로 실행되는 2개의 에이전트들이 시간의 흐름에 따라 4의 단위 시간에는 E_{T1}이 실행되었고, 9의 단위시간에 E_{T2}가 실행되었다.

- In-movement와 Out-movement는 이동 동작으로 tick에 의해서 변경된 시간 g와 e시간 후에 완료하게 된다.

$$_{[1]}in\ A_{[1,2,2,5]}$$



(그림 6) Summation 시간분석



(그림 7) Parallel 시간분석

이동은 실행시간이 완료되는 시점에서 다른 에이전트로의 이동이 완료된다. 이 완료되는 시점은 데드라인을 넘지 않아야 한다. (그림 8)은 timeout 안에서 정상적으로 실행이 시작되고, 실행시간 2동안 실행된 뒤 데드라인 안에서 정상적으로 종료된 예제이다.

- Synch-Communication (Case 1)

$$_{[2]}E_{[1,1,0,3]} \mid_{[3]} E_{[1,1,0,3]}$$

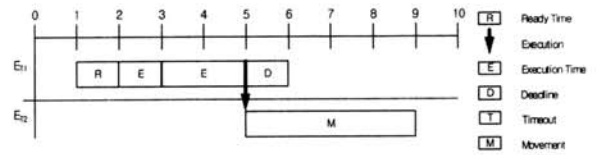
통신에서의 시간은 모호하지 않고, 정확하다. 싱크통신이 실행되는 경우는 하나의 에이전트가 싱크통신을 위해 준비를 완료하고, 대기 중이어야 하며, 상대 에이전트가 통신을 위한 준비가 완료되는 시점에서 즉시 이루어진다. (그림 9)는 E_{T1}과 E_{T2}의 즉각적인 동기적 통신을 의미하며, E_{T1}이 먼저 대기 중이며 E_{T2}가 실행 가능한 상태가 되는 시점에 통신이 수행된다. 이때 E_{T1}이 단위시간 4를 지나면 통신은 실패하게 된다.

- Synch-Communication (Case 2)

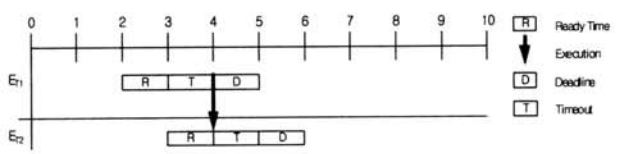
$$_{[2]}E_{[1,1,1,3]} \mid_{[3]} E_{[1,1,1,2]}$$

실행시간을 갖는 행위의 경우, 실행시간의 전제는 통신을 수행하는 2개의 에이전트에서의 실행시간은 서로 같아야 한다는 것이다. 이때는 행위 시작을 위한 타임아웃 시간에 실행이 시작되어야 하며, 양쪽에서 실행시간만큼 실행된 후 실행의 완료는 데드라인 시간 안에 완료되어야 한다. (그림 10)은 실행시간이 존재하는 통신을 표현한 예제이다. 즉각적인 통신과 같이 통신을 위한 에이전트는 통신을 수행할 수 있는 즉시 실행된다.

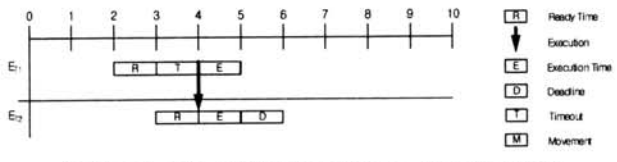
- Close synch-communication은 동기적인 액션을 수행할 때 하나의 포트에 대해서 감춰진 동작으로 τ액션을 수행한다.



(그림 8) 이동 시간분석



(그림 9) 통신 시간분석(즉각실행)



(그림 10) 통신 시간분석(실행시간이 소모되는 경우)

• Open synch-communication은 Close synch-communication과 동일하지만, 통신을 수행하는 포트가 공개되어 있다. τ액션과 구분하기 위해 $\tau(y)$로 표기한다.

2.4 통신 (인터액션) 의 표현

간단한 통신을 수행하는 에이전트의 명세는 다음과 같다. 시스템은 에이전트 A에서 B로 x포트를 통해 메시지를 전송한 후, B에서 A로 전송한 메시지를 다시 수신 받는 예제이다. *Reconfigurable Hyper Graph*(RHG)[13]로는 (그림 11)과 같이 나타낼 수 있다.

시간에 따른 행위정보는 다음과 같다(단, $A \overset{def}{=} E_{T1}$, $B \overset{def}{=} E_{T2}$):

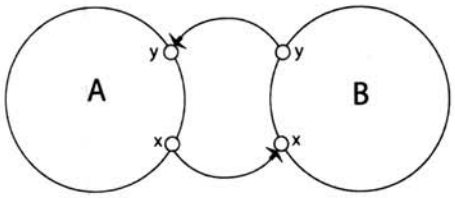
$$system_1 \overset{def}{=} E_{T1} \mid E_{T2}$$

$$=_{[1]} \overline{x}(m_1)_{[1,1,0,5]} \bullet y(z_1 : msg)_{[1,1,0,3]} \bullet E_{T1}' \mid_{[2]} x(z_1 : msg)_{[1,1,0,3]} \bullet \overline{y}(z_1)_{[1,1,0,3]} \bullet E_{T2}'$$

$$\xrightarrow{[5]\tau(x)}{}_{[7]} y(z_1 : msg)_{[1,1,0,3]} \bullet E_{T1}' \mid_{[6]} \overline{y}(z_1)_{[1,1,0,3]} \bullet E_{T2}'$$

$$\xrightarrow{[8]\tau(y)} E_{T1}' \mid E_{T2}'$$

x포트를 통한 τ액션 후에 $\tau(y)$액션을 실행해 $E_{T1}' \mid E_{T2}'$의 상태로 변하게 된다. 처음 $_{[1]}\overline{x}(m_1)_{[1,1,0,5]}$와 $_{[2]}x(z_1 : msg)_{[1,1,0,3]}$의 통신이 3의 시간에 수행되면 각각의 다음 액션의 전역시간은 7, 6이 된다. 만약, 2번째 통신이 8의 시간에 수행된다면 모든 액션이 정상적으로 수행된다.



(그림 11) 메시지를 주고받는 2개의 에이전트

3. Timed Action Graph (TAG)

TAG 는 t-CARDMI의 시간속성을 분석하기 위해 정의된 그래프로 시간의 공간정보와 행위의 시간정보를 동시에 표현가능하며 각 행위들의 시간속성의 충돌을 쉽게 파악할 수 있다.

3.1 TAG의 표기법

• 단일 노드

하나의 노드는 사각형의 박스로 나타낸다. 노드는 정적, 동적 노드가 존재한다. 정적 노드는 전체시간에 걸쳐 사각형이 그려진다. 동적 노드는 생성과 소멸이 자유로우며, 특정시간에 생성되어 특정시간에 소멸될 수 있다. 또한 동적 노드 안에는 정적 노드가 포함될 수 없다.

• 포함된 노드

포함된 노드는 하나의 노드 안에 또 다른 노드가 포함되어 나타낸다.

• 시간의 표현

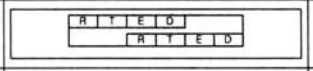
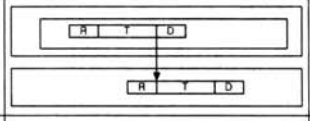
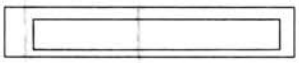
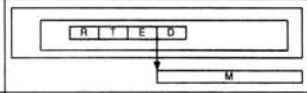
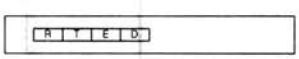
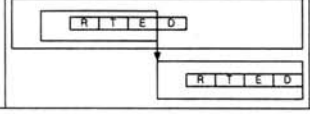
-대기시간 (Ready Time)

준비/대기 시간 r은 'R'이 표기된 파란색의 박스로 나타내며, r의 시작부분은 g에 해당되는 부분이다. 모든 행위는 r을 모두 소모해야 실행될 수 있다.

-실행시작 만족시간 (Timeout)

Timeout to는 'T'가 표기된 노란색의 박스로 나타내며, 대기시간이 r이 끝나는 시점부터 행위가 시작될

〈표 1〉 TAG 문법

Notation	Semantic	Notation	Semantic
	Time line		Choice
 (파랑)	Ready Time		Time out
 (노랑)	Time out		Group
 (빨강)	Execution Time		Interaction
(녹색)	Deadline		Movement
	Agent		Interaction
	Nested Agent		Movement (Spec Mode)
	Agent execution		Movement (Run Mode)

수 있는 시간이다.

-실행시간(Execution Time)

실행시간은 'E'가 표기된 붉은색의 박스로 표현되며, 실행시간이 0인 경우는 즉시 실행을 나타낸다.

-실행완료만족시간 (Deadline)

실행 완료만족시간은 'D'가 표기된 녹색의 박스로 나타내며, 실행시작 시간에 시작된 행위의 실행이 실행시간만큼 실행되고 완료되는 시점에서 만족해야 하는 시간이다.

• 이벤트

이동과 통신을 위한 이벤트는 화살표로 표현된다. 화살표는 이동과 통신에 따라 다음과 같은 의미를 갖는다:

· 이동: 화살표의 시작은 이동하는 주체를 나타내며, 화살표의 끝은 이동을 하고자 하는 목적 노드를 나타낸다.

· 통신: 화살표의 시작은 통신에서 메시지를 전송하는 노드의 포트를 나타내며, 화살표의 끝은 통신에서 메시지를 전송 받을 노드의 포트를 가리킨다.

-통신의 명세

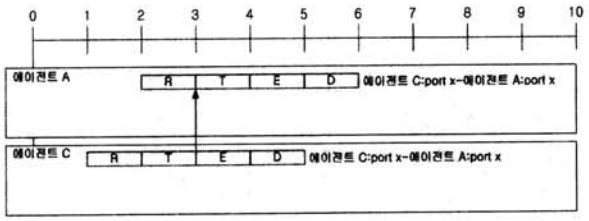
(그림 12)는 에이전트 C에서 준비를 마치고 대기하고 있던 행위가 에이전트 A가 활성화되면서 싱크통신을 수행하는 그림이다.

-명세에서의 이동

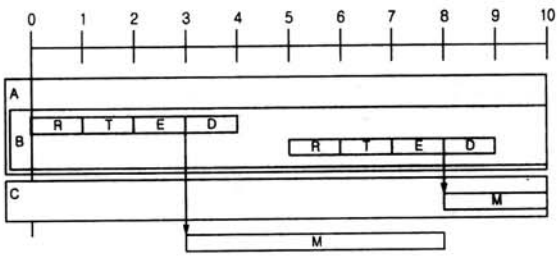
명세과정에서의 이동의 표현은 이동하고자 하는 노드에 포함된 이동 이벤트의 실행시간이 완료되는 시점에서 다른 노드로 노드가 포함된다. 이때 이동됨을 표현하기 위해 목적노드에 박스 형태의 이동된 에이전트가 나타난다. 그림 13은 에이전트 B가 *out A*명령에 의해서 A를 빠져나온 후 다시 *in C*에 의해서 C로 포함되는 것을 명세하고 있다.

-실행에서의 이동

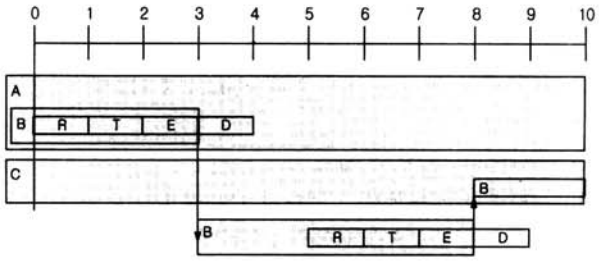
실행에서의 이동은 (그림 14)와 같이 노드가 실제로 이동되는 과정을 보여준다. 이때 실행된 명령을 제외한 나머지



(그림 12) 통신의 표현



(그림 13) 명세에서의 이동 표현

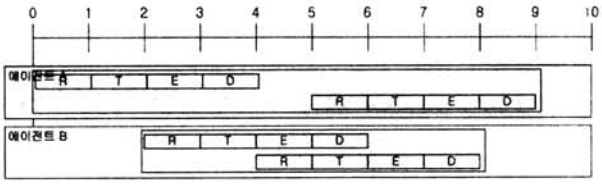


(그림 14) 실행에서의 이동의 표현

행위는 노드 안에 계속 포함된다.

• 그 룹

그룹은 TAG에서 명세를 위해 새롭게 정의된 개념이며, 일반적인 그룹은 동작의 연속을 표현한다. 이는 어떠한 행위를 한 후 연결된 다음 행위를 명세하기 위해 필요하다. (그림 15)의 경우 에이전트 B의 그룹의 시간명세는 잘못된 표현이다. 그룹 안에서의 에이전트들의 행위는 중복되지 않아야 한다.



(그림 15) 그룹의 표현

• 선 택

선택 하나의 행위나 그룹, 타임아웃이 복합적으로 명세될 수 있다.

• 오류복구

오류복구는 정상적인 실행에 실패하였을 경우 실행되는 행위로, 하나의 행위나 그룹, 선택이 복합적으로 적용될 수 있다.

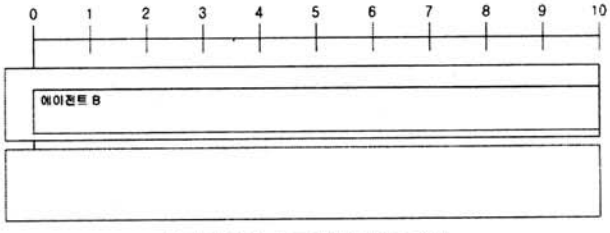
3.2 TAG의 의미

그래픽 기반의 표기법을 따르고 있는 TAG는 다음과 같은 의미를 내포하고 있다.

• 시공간정보의 표현

RHG와 같이 노드의 포함관계를 나타낸다. (그림 16)은 B를 포함하고 있는 A와 동급의 C를 나타낸 그림이다.

RHG와 같이 공간의 계층적 관계를 표현하지만, TAG는 각 에이전트가 가지는 모든 행위와 시간에 따른 행위들의



(그림 16) 시공간정보의 표현

반응을 같이 표현할 수 있다.

• 프로세스의 상태에 대한 표현

특정시간에 프로세스가 생성되고 액션 수행 후 소멸되는 것을 나타낸다.

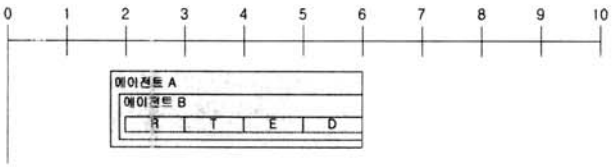
그림 17과 같이 동적인 에이전트가 언제 실행되어 어떠한 행위를 한 뒤 어느 시점에서 사라지는지를 정확히 판단할 수 있다.

TAG는 시간 및 공간 정보와 시스템 및 각 에이전트가 지니는 행위 및 생성/소멸에 대한 정보를 포함하고 있는 그래프이다.

• TAG를 통해 분석할 수 있는 시간속성

다양한 시간속성과 TAG를 통해 분석할 수 있는 시간에 대한 다양한 속성들은 다음과 같다:

1) 시간의 속성에 따른 각 에이전트들의 위치정보.
2) 특정 통신이 발생했을 때의 에이전트의 위치 정보.
3) 시간에 따른 에이전트의 행위에 대한 Sync 발생 여부.
4) 통신이 발생되지 못했을 경우 이를 복구할 수 있는 시간에 대한 정보.
5) 특정 시간에 공간상에서의 통신권한 획득 여부.



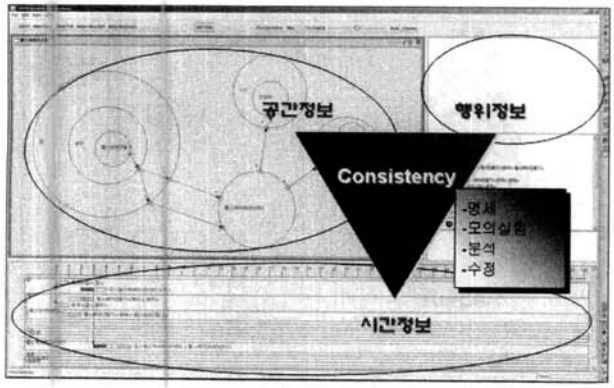
(그림 17) 프로세스 상태표현

4. t-CARDMI 에디터 및 검증기(SAVE)

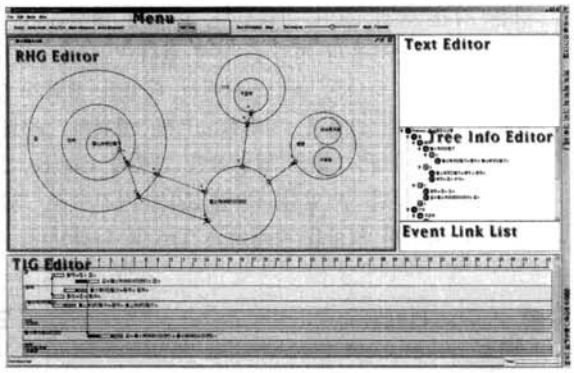
t-CARDMI에서 시간을 명세하고 분석하기 위해 개발된 SAVE는 공간에 대한 명세와 행위의 명세를 수행하고, 시간에 따른 행위를 분석하기 위한 응용프로그램이다. (그림 18)과 같이 SAVE는 공강정보와 행위정보, 시간정보에 대해 명세, 분석, 수정에서 내용의 일치를 제공하는 장점을 갖는다.

4.1 화면구성

SAVE의 화면은 (그림 19)와 같이 구성되어 있다.



(그림 18) SAVE의 구현 접근 방법

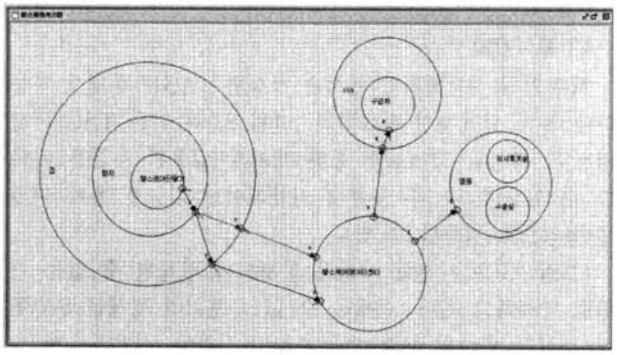


(그림 19) SAVE의 인터페이스

4.2 명세과정

명세의 방법은 기존의 프로세스 대수의 명세 방법과는 많은 차이를 보인다. 하나의 시스템을 명세하고 분석하기 위한 순서는 다음과 같다:

1) 공간정보의 명세: 시간 속성을 제외한 순수한 공간정보만을 구성한다. 이때 노드와 포트가 구성된다.
2) 행위의 명세: 시간을 포함하지 않은 이동과 통신에 대해서 명세를 한다. (그림 20)은 행위정보를 명세한 이후의 그림이다.
3) 시간의 명세: 인터액션의 경우 Add Time 버튼으로 시간을 명세하고자 하는 링크를 선택하면 (그림 21)과

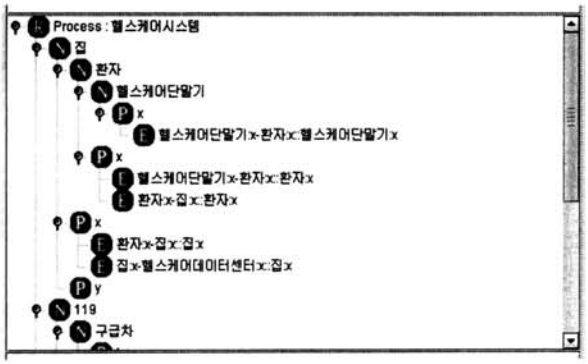


(그림 20) RHG 공간 행위 정보 명세

(그림 21) 시간의 명세

같은 시간 속성 창이 나타난다. 이곳에 행위를 하는 양쪽의 에이전트들의 시간을 명세 한다.

4) 그룹의 생성: 하나의 에이전트에서 다양한 행위가 발생하기 때문에 그룹을 정의함으로써 행위의 순서를 명세 한다. 이때 그룹은 일반적인 그룹과 Choice, 오류복구를 생성할 수 있다. (그림 22)는 시간수정, 그룹 생성 등 다양한 역할을 수행하는 Tree를 나타낸다.



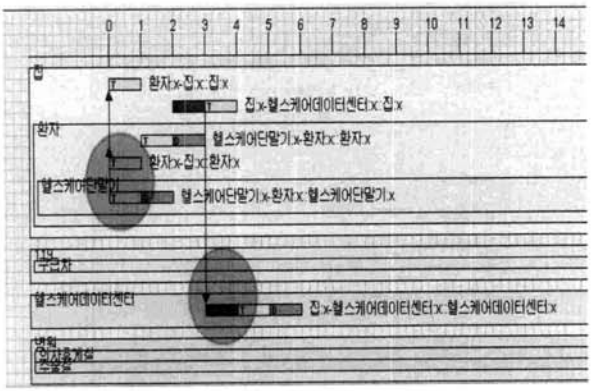
(그림 22) 그룹, 초이스, 오류복구, 시간수정을 수행하는 Tree

5) 최종 시간 수정: TAG 그래프에 나타난 각 행위의 시간에 대해 분석하여, 잘못된 곳을 수정한다. 이때는 Tree Info Editor에서 해당 행위를 선택한 뒤 팝업 메뉴로 시간을 수정할 수 있다.

4.3 분석방법

행위의 분석: 행위의 분석은 생성된 TAG를 통해서 가능하다. 4)의 단계까지 완료하면 (그림 23)과 같은 TAG가 생성된다. 생성된 그래프를 통해 명세의 잘못된 부분을 파악할 수 있으며, 추 후 포함될 시뮬레이션 기능을 통해 명세를 동적으로 검증할 수 있다.

(그림 23)은 시간의 정의가 잘못되어 실행될 수 없는 행위를 명세하고 있다. 실행될 수 없는 행위의 명세는 SAVE에서 시간 속성의 오류가 있음을 표시하고, 사용자에게 충돌이 발생하는 액션을 표시해준다. 명세 시점에서의 시간은 범위로 표현되지만, 추후 구현될 시뮬레이션 및 검증기는 각 액션에서 동적으로 변하는 시간들에 따른 액션의 도달성 그래프를 사용하여 명세한 시간 범위 안에서의 실행 가능한 시간을 분석하게 된다.



(그림 23) 시간의 정의에 따른 TAG 표현

5. 예제를 통한 분석 및 활용: 헬스케어 시스템

본 예제는 Body Sensor Network System의 주도적인 연구를 진행하고 있는 [14]에서 제안된 UbiMon System이며, (그림 24)는 UbiMon 헬스케어 시스템의 아키텍처를 나타내는 그림이다. 본 시스템은 환자의 몸에 센서를 장착하여 환자가 위급할 때 가까운 병원이나 의사의 직접적인 도움을 받도록 구현된 시스템이며, 본 예제에서는 기본 시스템을 본 논문에 맞도록 수정하였고, (그림 25)는 본 예제를 RHG의 시간의 흐름에 따라 표현한 것이다.

$$\text{센서} \overset{def}{=} E_{T1},\ \text{환자} \overset{def}{=} E_{T2},\ \text{집} \overset{def}{=} E_{T3},\ \text{중앙감시센터} \overset{def}{=} E_{T4},$$
$$119\text{센터} \overset{def}{=} E_{T5},\ \text{구급차} \overset{def}{=} E_{T6},\ \text{병원} \overset{def}{=} E_{T7},\ \text{수술실} \overset{def}{=} E_{T8},$$
$$\text{의사휴게실} \overset{def}{=} E_{T9},\ \text{의사} \overset{def}{=} E_{T10}$$

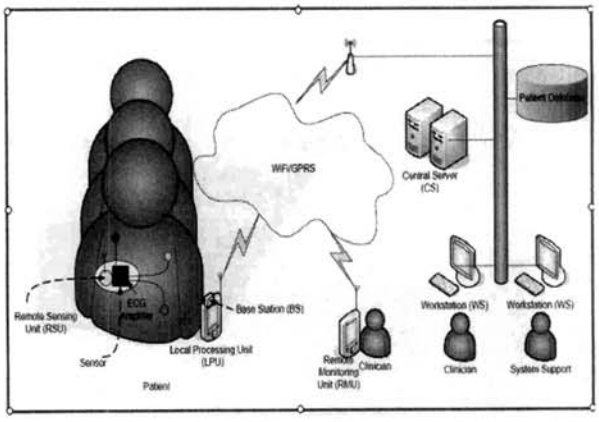
$$HealthcareSystem \overset{def}{=} \left[\left[\left[E_{T1}\right] \mid E_{T2}\right] \mid E_{T3}\right] \mid E_{T4} \mid \left[E_{T5} \mid \left[E_{T6}\right]\right] \mid \left[E_{T7} \mid \left[E_{T8}\right] \mid \left[E_{T9} \mid \left[E_{T10}\right]\right]\right]$$

$$= \left[\begin{array}{l} \left[\begin{array}{l} \left[\overline{x}_1(m_1)_{[5,9,0,14]} \bullet E'_{T1}\right] \mid \\ \{m_2 / z_1\} x_1(z_1 : msg)_{[5,9,0,14]} \bullet <_{[15]} \overline{x}_1(m_2)_{[5,9,0,14]}, \{\overline{x}_2 / \overline{x}_1\}_{[30]} \overline{x}_2(m_2)_{[5,39,0,44]} > \\ \bullet_{[240]} in\ E_{T6[5,15,39,59]} \bullet_{[480]} out\ E_{T6[10,30,59,119]} \bullet_{[600]} in\ E_{T8[5,15,29,59]} \bullet E'_{T2} \end{array}\right] \mid \\ \left[\begin{array}{l} \{m_3 / z_2\}_{[15]} x_1(z_2 : msg)_{[5,9,0,14]} \bullet \overline{x}_1(m_3)_{[5,24,0,29]} \\ \bullet E'_{T3} + \{m_4 / z_2\}_{[45]} x_2(z_2 : msg)_{[5,69,0,74]} \bullet \overline{x}_2(m_4)_{[5,24,0,29]} \bullet E'_{T3} \end{array}\right] \end{array}\right] \mid$$

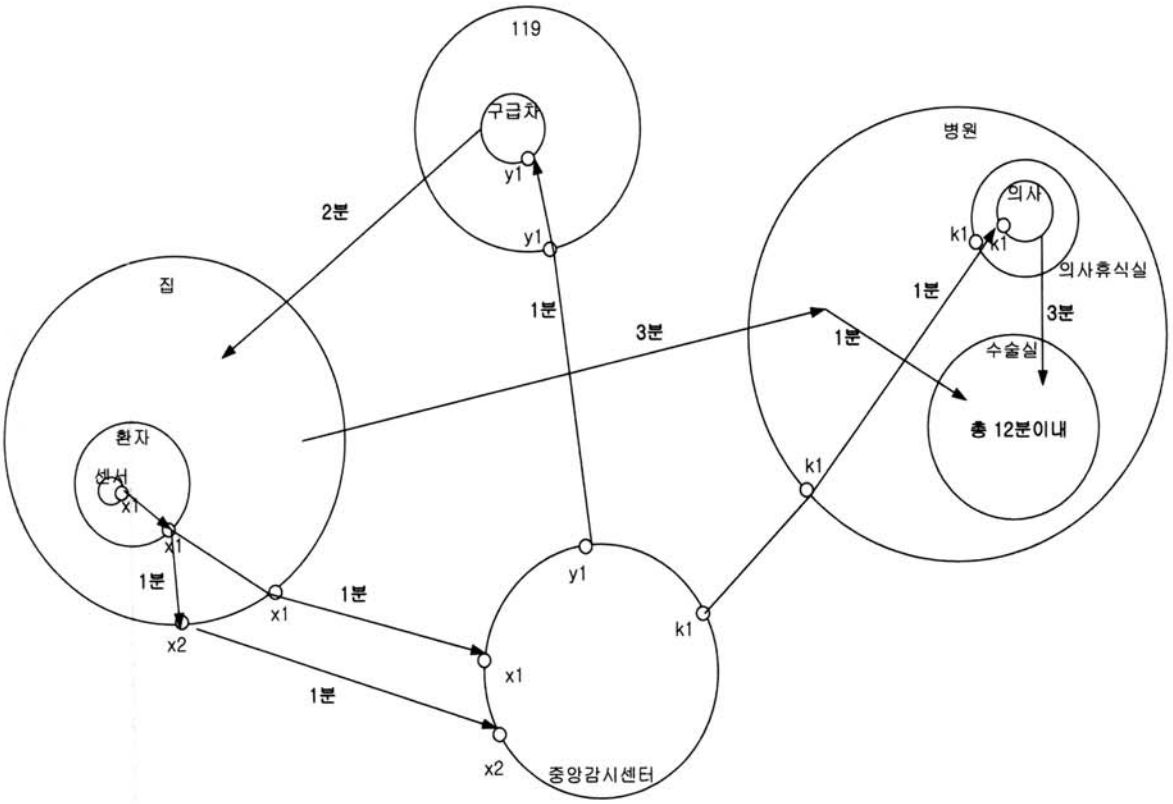
$$\left[\begin{array}{l} \{m_5 / z_3\}_{[40]} x_1(z_3 : msg)_{[5,24,0,29]} \bullet \overline{y}_1(m_5)_{[5,14,0,19]} \bullet \overline{k}_1(m_5)_{[5,24,0,29]} \bullet E'_{T4} \\ + \{m_5 / z_4\}_{[100]} x_1(z_4 : msg)_{[5,24,0,29]} \bullet \overline{y}_1(m_5)_{[5,14,0,19]} \bullet \overline{k}_1(m_5)_{[5,24,0,29]} \bullet E'_{T4} \end{array}\right] \mid$$

$$\left[\begin{array}{l} y_1(z_5 : msg)_{[5,134,0,139]} \bullet \overline{y}_1(m_5)_{[5,24,0,29]} \bullet E'_{T5} \mid \\ [_{[120]} y_1(z_5 : msg)_{[5,24,0,29]} \bullet out\ E_{T5[5,15,10,29]} \bullet in\ E'_{T3[5,15,29,59]} \\ \bullet out\ E'_{T3[5,15,29,59]} \bullet in\ E_{T7[5,15,159,179]} \bullet E'_{T6}] \end{array}\right] \mid$$

$$\left[\begin{array}{l} k_1(z_5 : msg)_{[5,164,0,169]} \bullet \overline{k}_1(m_5)_{[5,14,0,19]} \bullet E'_{T7} \mid \\ [E'_{T8}] \mid \\ \left[\begin{array}{l} \{m_6 / z_5\} k_1(z_5 : msg)_{[5,174,0,179]} \bullet \overline{k}_1(m_6)_{[5,24,0,29]} \bullet E'_{T9} \mid \\ [_{[130]} k_1(z_6 : msg)_{[5,54,0,59]} \bullet out\ E'_{T9[5,15,99,119]} \bullet in\ E'_{T8[5,15,159,179]} \bullet E'_{T10}] \end{array}\right] \end{array}\right]$$



(그림 24) 헬스케어 시스템



(그림 25) RHG의 표현

• 정상실행 결과

$$\xrightarrow{[65]\,in\ E_8}\left[\left[[E'_{T1}]\,|\,E'_{T2}\right]\,|\left[\begin{matrix}E'_{T3}+\{m_4/z_2\}_{[45]}x_2(z_2:msg)_{[5,69,0,74)}\\ \bullet\,\overline{x}_2(m_4)_{[5,24,0,29]}\bullet E^{*}_{T3}\end{matrix}\right]\right]\,|$$
$$\left[\begin{matrix}E'_{T4}+\{m_5/z_4\}_{[100]}x_1(z_4:msg)_{[5,24,0,29]}\bullet\overline{y}_1(m_5)_{[5,14,0,19]}\\ \bullet\overline{k}_1(m_5)_{[5,24,0,29]}\bullet E^{*}_{T4}\end{matrix}\right]\,|$$
$$\left[E'_{T5}\,|\,[E'_{T6}]\right]\,|\left[E'_{T7}\,|\,[E'_{T8}]\,|\left[E'_{T9}\,|\,[E'_{T10}]\right]\right]$$

$<_{[15]}\overline{x}_1(m_2)_{[5,9,0,14]},\{\overline{x}_2/\overline{x}_1\}_{[30]}\overline{x}_2(m_2)_{[5,39,0,44]}>$에서 $_{[15]}\overline{x}_1(m_2)_{[5,9,0,14]}$가 정상적으로 실행된 경우를 나타낸다. 이러한 경우는 통신을 수행하기 위한 상대 에이전트가 정상적인 실행 시간을 만족해야 한다. 예제에서는 $\{m_3/z_2\}_{[15]}x_1(z_2:msg)_{[5,9,0,14}$이 해당된다.

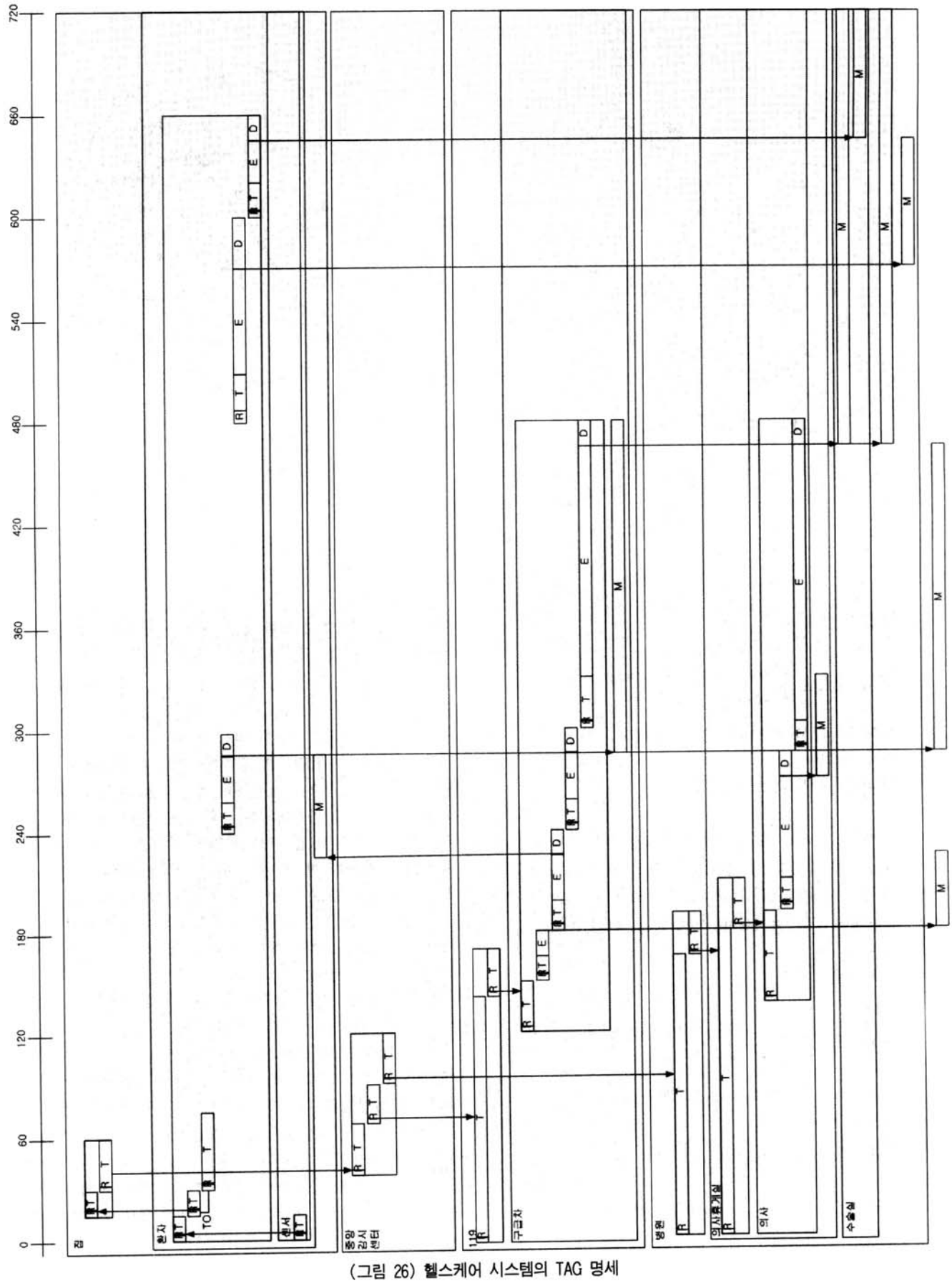
• 오류복구실행 결과

$$\xrightarrow{[65]\,in\ E_8}\left[\left[[E'_{T1}]\,|\,E'_{T2}\right]\,|\left[\begin{matrix}E'_{T3}+\{m_4/z_2\}_{[45]}x_2(z_2:msg)_{[5,69,0,74)}\\ \bullet\,\overline{x}_2(m_4)_{[5,24,0,29]}\bullet E^{*}_{T3}\end{matrix}\right]\right]\,|$$
$$\left[\begin{matrix}E'_{T4}+\{m_5/z_4\}_{[100]}x_1(z_4:msg)_{[5,24,0,29]}\bullet\overline{y}_1(m_5)_{[5,14,0,19]}\\ \bullet\overline{k}_1(m_5)_{[5,24,0,29]}\bullet E^{*}_{T4}\end{matrix}\right]\,|$$
$$\left[E'_{T5}\,|\,[E'_{T6}]\right]\,|\left[E'_{T7}\,|\,[E'_{T8}]\,|\left[E'_{T9}\,|\,[E'_{T10}]\right]\right]$$

$<_{[15]}\overline{x}_1(m_2)_{[5,9,0,14]},\{\overline{x}_2/\overline{x}_1\}_{[30]}\overline{x}_2(m_2)_{[5,39,0,44]}>$에서 $_{[15]}\overline{x}_1(m_2)_{[5,9,0,14]}$의 timeout에 정상적으로 실행되지 못해 $\{\overline{x}_2/\overline{x}_1\}_{[30]}\overline{x}_2(m_2)_{[5,39,0,44]}$이 실행된 경우를 나타낸다. 정상적인 실행 시간을 만족하지 못한 경우, 오류를 복구하기 위한 대체 액션을 명시한 것으로, 대처 가능한 시간을 만족하는 행위가 대신 수행된다. 이러한 경우 $\{m_4/z_2\}_{[45]}x_2(z_2:msg)_{[5,69,0,74)}$과 같이 상대 에이전트는 오류를 대처할 수 있는 시간을 모두 만족하거나, 대체 액션이 명시되어야 한다.

TAG 그래프는 RHG 의 공간정보와 t-CARDMI의 시간, 행위정보를 종합적으로 표현하여, (그림 26)과 같이 헬스케어 시스템 예제를 가시화하여 나타낼 수 있다. 시각화와 인식성이 뛰어난 TAG를 통해 시간에 기반을 둔 에이전트들의 행위를 가시화 할 수 있으며, 시간속성에서의 타임아웃, 지연 등의 분석/판별을 통해 시간의 요구사항 즉, 에이전트들의 행위의 발생시간과 각 에이전트들간의 통신 발생 시간, 통신이 발생했을 때의 에이전트의 공간정보, 각 행위들의 복구를 위해 허용되는 시간 등에 대한 시간 정보를 정확히 파악할 수 있다. 또한, 공간적인 오류에 대한 오류를 보안/수정 할 수 있는 큰 장점을 지닌다.

t-CARDMI는 다양한 시간 속성을 명세함으로써 기존의 시간속성을 갖는 프로세스 대수에 비해 복잡한 행위를 명세할 수 있는 장점을 지닌다. 하지만, 도입된 시간 속성이 복잡해짐에 따라 명세에 있어서 다양한 속성들을 모두 분석해야 된다는 문제점을 지닌다. 또한, 에이전트의 이전 행위에 대한 시간 정보를 유지하기 위한 명세 방법이 존재하지 않으므로 행위가 이전 행위의 어떠한 시간 흐름 속에서 실행되었는지에 대한 정보를 알 수가 없다.



(그림 26) 헬스케어 시스템의 TAG 명세

6. 결 론

본 논문은 CARDMI에 시간 속성을 확장하고, 시간의 분석을 위한 TAG를 제안하였다. 또한, 명세와 분석을 위한 에디터와 검증기를 구현하였다.

시간의 속성은 병렬, 이동, 동기적인 에이전트들의 통신과 이동의 특성에 맞도록 정의하였으며, 실행을 위한 준비시간, 실행시간, 데드라인의 개념을 적용하여 정확한 시간의 명세를 가능하도록 하였다. 또한 오류의 복구를 수행할 수 있도록 Timeout 오퍼레이터를 추가하였다.

t-CARDMI의 시간에 따른 행위를 분석하기 위해 공간정보 및 행위정보, 그리고 시간정보를 표현할 수 있으며, 이를 이해하기 쉬운 형태로 표현한 TAG를 제안하였다. TAG의 각각의 정보는 명세와 실행의 단계를 가지며, 명세의 단계는 공간과 행위, 시간의 정적인 상태를 기술하고, 실행의 단계는 동적인 상태를 확인할 수 있는 특징을 보였다.

재구성 하이퍼그래프와 TAG를 정의하고 검증할 수 있는 에디터 및 검증기를 구현하였다. 구현된 명세 툴은 기존의 텍스트 방식의 명세가 아닌 GUI기반의 명세를 지원하고, t-CARDMI와 TAG 명세의 편리한 사용자 인터페이스를 제공한다. 또한 정적, 동적인 분석 기능을 지원하여, 이를 통해 효율적인 정형명세를 수행할 수 있도록 하였다.

향후 연구에서는 즉시 실행에 대한 정의와 즉시 실행에 따른 시간의 보상에 대해 연구와 RHG 안에서의 시간의 흐름에 대한 표현 방법, 그리고 정의된 행위의 정확한 시간 분석 방법에 대한 연구가 진행되어야 할 것이다. 또한, 구현된 에디터 및 검증기에 시뮬레이션 기능을 추가하여 다양한 속성에 대한 검증을 수행할 수 있도록 기능을 추가하고, Zed/VDM과 같은 실행모델 언어와 ITS 명세 언어에 대해서도 연구가 진행되어야 할 것이다.

참 고 문 헌

[1] Jonathan Jacky (1997). The Way of Z: Practical Programming with Formal Methods. Cambridge University Press. ISBN 0-521-55976-6.

[2] Gérard Berry. To appear in Proof, Language and Interaction: Essays in Honour of Robin Milner, G. Plotkin, C. Stirling and M. Tofte, editors, MIT Press, 1998.

[3] J. Dawes, The VDM-SL Reference Guide, Pitman 1991. ISBN 0-273-03151-1.

[4] R. Miler, J. Parrow and D.J. Walker. A Calculus of Mobile Processes, Part I. Report ECSLFCS8985, Laboratory for foundations of Computer Science, Computer Science Dep., Edinburgh Univ., 1989.

[5] R. Miler, J. Parrow and D.J. Walker. A Calculus of Mobile Processes, Part II. Report ECSLFCS8986, Laboratory for foundations of Computer Science, Computer Science Dep., Edinburgh Univ., 1989.

[6] 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.140-155, 1998.

[7] R. Miler. Bigraphical Reactive Systems: Basic Theory. Technical Report 503, University of Cambridge Computer Laboratory, 2001.

[8] R. Milner. Bigraphical Reactive Systems, In Proc. of the 12Ih International Conference on Concurrency Theory, No. 2154 in Lecture Note in Computer Science, 2001. pp.1635.

[9] Wang Yi. CCS+Time = an Interleaving Model for Real-Time System. In Proc. of the Eighteenth Colloquium on Automata Languages and Programming, volume 510 of LNCS. Springer Verlag, 1991.

[10] Xu K, Liu L.C., Wu C. Time Pi Calculus and Weak-timed Bisimulation Analysis, Computer Integrated Manufacturing Systems, 2005, In Press.

[11] J. Y. Lee and J. Zic. On modeling real-time mobile processes. Australian Computer Science Communications archive Vol.24, pp.139-147, 2002.

[12] Gao and Li and Chen, Services Composition Modeling with Mobility and Time, services, pp.316-323, 2007 IEEE Congress on Services (Services 2007), 2007.

[13] J. Choi A Calculus for Equivalence Analysis and Verification of Distributed Mobile System Based on Abstraction. PhD Dissertation, Chonbuk national Univ. 2007.

[14] UbiMon Systems http://www.ubimon.org/.

온 진 호

e-mail : jjinghott@gmail.com

2006년 원광대학교 컴퓨터정보통신공학(학사)

2008년 전북대학교 컴퓨터공학(석사)

2008년~현 재 전북대학교 컴퓨터공학 박사과정

관심분야 : 실시간시스템, 정형기법, 분산 이동 명세언어, 소프트웨어 공학, 분산/병렬시스템

최 정 란

e-mail : seohyun@formal.korea.ac.kr

1999년 전북대학교 컴퓨터과학과(이학사)

2001년 전북대학교 컴퓨터통계정보(이학석사)

2007년 전북대학교 컴퓨터통계정보(박사)

2008년~현 재 고려대학교 BK21 소프트웨어 산학연 연구교수

관심분야 : 정형기법, 소프트웨어공학, RFID, etc.

이 문 근

e-mail : moonkun@chonbuk.ac.kr

1989년 The Pennsylvania State University, Computer Science 학과(이학사)

1992년 The University of Pennsylvania, Computer and Information Science 학과(이공학석사)

1995년 The University of Pennsylvania, Computer and Information Science 학과(이공학박사)

1992년 5월~1996년 1월 미국, Computer Command and Control Company, Computer Scientist로 근무

1996년~1998년 전북대학교 컴퓨터과학과 전임강사

1998년~1998년 전북대학교 컴퓨터과학과 조교수

1999년~2002년 전북대학교 전자정보공학부(컴퓨터과학과 전공) 조교수

2002년~2007년 전북대학교 전자정보공학부(컴퓨터과학과 전공) 부교수

2008년~현 재 전북대학교 전자정보공학부(컴퓨터공학과 전공) 교수

관심분야 : 정형기법, 소프트웨어 재·역공학, 실시간 시스템, 운영체제, 형식언어, 병렬함수 언어, 컴파일러 등