

논문 2004-41CI-3-1

# 이동물체 데이터베이스의 페트리 넷 모형

## (Petri Net Model for Moving Objects Database)

임 재 결\*, 이 계 영\*

(Jaegool Yim and Gyeyoung Lee)

### 요 약

현재 이동물체 데이터베이스에 관한 연구가 활발히 진행되고 있다. 그러나 기존 연구의 대부분은 갱신 전략의 효율성 증대에 초점이 맞추어져 있고, 이동물체 데이터베이스 시스템의 사용자 요구사항 검증에 대한 연구 논문은 전무하다. 아무리 효율성이 좋은 갱신 전략이 적용되었다고, 시스템 설계자는 시스템이 사용자 요구사항을 만족하는지에 대해 개발초기에 반드시 검증을 해 보아야 한다. 예를 들어, 이동물체가  $n$ 개이면,  $n$ 개에 대한 데이터베이스 정보 갱신 작업과 새로운 정보를 사용자에게 제공하여주는 작업을 정해진 시간 내에 완료할 수 있는지 개발 초기에 검증해야 하는 것이다. 본 논문은 이동물체 데이터베이스의 요구사항 분석을 위하여, Design/CPN이라는 시뮬레이션 도구를 이용하여 페트리 넷 모형을 구축하고, 시뮬레이션을 통하여 사용자 요구사항을 검증하는 방법을 소개한다. 본 논문은 이동물체 데이터베이스의 사용자 요구 분석을 위한 최초의 시뮬레이션 모델을 제공하는데 의의가 있다. 이 시뮬레이션 모델은 본 저자가 이미 발표한 퍼지시간 페트리 넷 모형에 대해 거리기반 갱신전략과 분산 데이터베이스 모형을 첨가 확장한 것이며,  $n$  개의 이동물체를 고려한 분석 결과를 소개한다. 또한, 본 모형은 Design/CPN이라는 시뮬레이션 도구로 구축되었기 때문에 시스템에 대한 자동분석이 가능하다. 본 모형의 응용 범위는 사용자 요구사항 분석에 국한된 것이 아니라, 갱신 비용과 정보의 정확도 간, 혹은 갱신에 소요되는 시간과 자원 간의 상관관계를 분석하는 데에도 유용하게 쓰인다.

### Abstract

Quite many papers about moving objects database (MOD) have been published. Most of them are concerning of improving efficiency of updating policy, but none of them talks about verification of system's requirements. No matter how efficient updating policy is employed, a system designer still has to verify if the MOD satisfies user's requirement at the beginning of system lifecycle. For example, if a MOD serves  $n$  moving objects, then the designer must verify if it can update information for  $n$  moving objects and provide new information to  $n$  moving objects within specified time limit. For the purpose of requirement analysis of MODs, we build a Petri net model for MOD using Design/CPN, and then we show how to verify whether the MOD satisfies user's requirements by simulation. The contribution of this paper is providing a simulation model for requirement analysis of MOD for the first time. The model is expansion of our previous fuzzy-timing Petri net model. The distance-based updating policy, and distributed database management system are reflected in this model and system analysis for  $n$  moving objects is considered. It is built in Design/CPN so that the simulation can be automatically performed. The application of our model is not limited to requirement analysis, it is useful to study other MOD design issues, such as the trade-offs between update cost and information accuracy, and between the time interval needed for updating database and MOD system resources, etc.

**Keywords:** Moving objects database, Petri net model, updating time, Design/CPN, and wireless communication

## I. 서 론

\* 정희원, 동국대학교 컴퓨터·멀티미디어학과  
(Dept. of Computer & multimedia, Dongguk Univ.)  
※ 이 논문은 동국대학교 교원해외연수 지원에 의하여 이루어졌음.  
접수일자: 2003년10월15일, 수정완료일: 2004년4월14일

GPS(Global Positioning System)는 물체의 위치를 파악하는 시스템으로서, Global의 의미는 서비스 범위가 지구 전체임을 뜻한다. GPS의 핵심은 24 개의 GPS 인공위성으로, 이들이 발사하는 특수 신호를 GPS 수신

기가 받아서 처리하여 자기의 위치가 지구상의 어디인지를 정확히 계산해 낸다<sup>[1]</sup>. GPS를 이용하는 방법은 매우 다양하다. 예를 들면, 헬리콥터, 전투기, 폭격기, 탱크, 병사 등 모든 이동 물체에 GPS 수신기를 장착하고, 각 이동물체의 위치 정보를 데이터베이스에 저장하면 “1분 이내에 x, y 좌표에 도달하여 적을 제압할 수 있는 헬리콥터나 전투기 혹은 탱크는 어느 것인가?”와 같은 질문에 데이터베이스가 즉시 대답하여 줄 수 있다.

미국 국방성이 GPS 인공위성의 신호를 민간에게 무료로 제공하여 줌에 따라, 이를 이용한 각종 서비스가 등장하게 되었다. 예를 들어 택시회사는 소속 택시에 GPS 수신기를 장착하고, 무선통신으로 본사 데이터베이스와 항상 교신하게 함으로써 ‘대한빌딩’에 위치한 손님에게 ‘1분 이내에 택시를 보내드리겠습니다.’와 같은 서비스를 제공할 수 있게 되었으며, 자가용 운전자에게 길을 안내하는 서비스, PDA 사용자에게 주변 약도와 저명한 지형 지물에 대한 정보를 제공하는 서비스 등이 모두 GPS를 이용한 서비스의 예이다.

기존의 데이터베이스는 정보 갱신이 드물게 발생하는 정적 데이터베이스인데 반하여, GPS 서비스에 사용되는 데이터베이스는 위치 정보가 시시각각으로 변화하는 동적 데이터베이스이다. 따라서 GPS 서비스에 사용되는 데이터베이스는 기존의 데이터베이스와 현저하게 다르며, 이를 이동물체 데이터베이스 (Moving Objects Database)<sup>[2]</sup>라고 지칭한다.

이동물체 데이터베이스에서는 이동물체가 항상 이동하고 있기 때문에 이동물체의 실제 위치와 데이터베이스에 저장된 이동물체의 위치 사이의 차이를 줄이는 것이 가장 중요한 과제이다. 이를 해결하기 위해서는 이동물체와 데이터베이스가 항상 교신하고 있으면 되겠지만, 이것은 무선통신 비용과 데이터베이스가 서비스하는 이동물체가 다수라는 사실 때문에 실현 불가능하다. 따라서, 이동물체 데이터베이스 분야에서는 이동물체의 위치 정보 갱신 횟수를 줄이면서 실제 위치와 데이터베이스 상의 위치의 차이를 극소화하기 위한 갱신 전략에 대한 연구가 활발히 진행되어 왔다.

Bar-Noy<sup>[3]</sup> 등은 갱신 전략을 시간기반 갱신 전략 (time-based updating policy), 이동기반 갱신 전략 (movement-based updating policy) 그리고 거리기반 갱신 전략 (distance-based updating policy)의 세 가지로 분류하였다. 시간기반 갱신 전략에서는 모든 이동물

체가 주기적으로 자기의 현재 위치를 데이터베이스에 통고하는데 반하여, 이동기반 갱신 전략에서는 이동물체가 주어진 범위를 벗어났을 때 위치를 갱신한다. 한편, 거리기반 갱신 전략에서는 전번 갱신 위치로부터 현재 위치가 제한거리 이상 멀어졌을 때 이동물체가 현재 위치를 데이터베이스에 통보한다.

이동물체의 이동경로가 이미 설정되어 있는 경우에는 주어진 시각의 이동물체의 위치를 추정하는 것이 비교적 용이하다. 이런 경우에는 이동물체의 실제 위치가 추정 위치와 상이할 때에만 데이터베이스를 갱신하므로 추정 방법이 정교할 경우에는 갱신 비용을 크게 경감할 수 있다. 이와 같이 이동경로가 이미 설정된 경우에 적용할 수 있는 갱신 전략이 [4]에 소개되었다. [5]에서는 이동경로를 알 수 없을 때, 경로를 추정하는 방법과 함께 이때 적용할 수 있는 갱신 전략 두 가지가 소개되었다.

이와 같이, 기존 연구는 갱신 비용을 줄임으로써 이동물체 데이터베이스의 효율성을 높이는 데에 초점을 맞추고 있다. 그러나 갱신 효율성이 높다고 해서 이동물체 데이터베이스가 적절한 서비스를 제공할 수 있는 것은 아니다. 시스템 설계자의 입장에서는 설계중인 이동물체 데이터베이스가 과연 정확한 서비스를 제공할 것이라는 것을 개발 초기에 검증해야 한다. 예를 들어 이동거리가 10m를 넘을 때에 갱신하는 거리기반 갱신 전략을 채용하는 n 개의 이동물체를 서비스하는 데이터베이스의 경우, “이동물체 m의 현재 위치는 (x, y) 지점의 반경 10 m 이내이다.”라고 대답하는 것이 과연 옳을 지 검증해야 하며, 이를 위하여 이동물체의 평균 이동속도가 초속 10m라면 n개의 이동물체가 데이터베이스를 갱신하는 작업이 1초 이내에 완료될 수 있음을 보여야 한다.

본 논문은 이동물체 데이터베이스의 페트리 넷 모형과 이동물체 데이터베이스의 사용자 요구사항을 검증하기 위한 페트리 넷의 분석 방법을 소개한다. 나아가서, Design/CPN을 이용한 이동물체 데이터베이스의 페트리 넷 모형을 구축하고 시뮬레이션을 통하여 사용자 요구사항을 분석하는 방법을 소개한다. 본 논문의 나머지 부분은 다음과 같이 구성된다. II장은 본 논문과 연관된 기존의 연구를 소개하고, III장은 본 논문의 핵심 부분으로서 Design/CPN<sup>[6]</sup>으로 구축한 이동물체 데이터베이스를 소개하며, IV장은 시뮬레이션을 통한 사용자 요구사항 검증 방법을 소개한다.

## II. 관련 연구

본 논문에서는 Design/CPN<sup>[6]</sup>을 이용하여 페트리 넷 모형을 구축하고 시뮬레이션을 통하여 이동물체 데이터베이스 시스템의 요구사항을 검증하는 방법을 소개한다. 본 절에서는 페트리 넷 분야와 이동물체 데이터베이스 분야에서 관련 연구를 살펴본다. 요구사항 분석은 전통적인 소프트웨어 공학 라이프 사이클의 첫 단계라 할 수 있다. 즉, 첫 단계에서 잠정적인 문제점들을 발견함으로써 유지보수 단계의 거대한 비용을 절감할 수 있다. 이와 관련하여 페트리 넷 분야에서는 요구사항 분석의 중요성을 인식하고 시스템 설계와 요구사항 검증에 대하여 많은 연구 결과를 발표하였다.

[7]은 요구사항 분석을 위하여 페트리 넷 모형을 구축하고 분석하는 방법을 소개하고, [8]은 컬러 페트리 넷 (colored Petri net)의 정의와, 컬러 페트리 넷으로 소프트웨어 시스템의 모형을 구축하는 방법과 분석하는 방법을 소개하였다. 또한, 페트리 넷으로 분산 실시간 시스템의 요구사항을 명시하고 분석하는 방법이 [9, 10]에 소개되었다. 업무 처리의 모형에서 가장 널리 사용되는 방법으로 사건 주도 과정 연결(Event-driven Process Chains: EPCs) 방식을 들 수 있는데, [11]에서는 EPCs를 컬러 페트리 넷으로 변형하는 방법이 소개되었다. [12, 13, 14]에서는 시간 페트리 넷(Time Petri Nets: TPNs)을 이용한 실시간 시스템의 요구사항 명세와 검증 방법이 소개되었다.

서론에서 언급한 바와 같이 이동물체 데이터베이스

분야의 경우, 갱신 비용 절감을 위한 연구가 활발히 진행되고 있으며, 이에 대한 적용 방법으로 이동물체의 미래 위치를 예측하는 기술이 주로 사용되고 있다. 그러나, 갱신 비용을 아무리 절감한다 하더라도 시스템 설계자는 본인이 설계 중인 시스템이 요구사항을 만족시키는지를 시스템 개발 초기에 반드시 검증할 필요가 있다. 그럼에도 불구하고, 이동물체 데이터베이스를 위한 사용자 요구사항을 검증하는 방법에 대한 연구는 아직 전무한 편이며, 다만 Fuzzy-Timing Petri Net (FTPN)<sup>[15]</sup>을 이용한 이동물체 데이터베이스 모형과 분석 방법을 소개한 [16]이 이동물체 데이터베이스의 분석을 위한 유일한 연구 결과이다. 본 논문의 내용은 [16]의 연장이므로, 본 절에서는 [16]의 내용을 비교적 자세히 설명한다.

### 2.1 FTPN 모형

Fuzzy-Timing Petri Net (FTPN)이 처음으로 소개된 논문은 [17]이다. 그 이후로 FTPN은 실로 다양한 문제 해결에 적용되었다<sup>[18][19]</sup>. (그림 1)은 이동물체 데이터베이스의 FTPN 모형을 보인 것으로 페트리 넷 관련 기본 용어는 참고문헌<sup>[20]</sup>을 참조하기 바란다. 이동물체 데이터베이스 시스템은 다음과 같이 구성된다.

- 1) GPS 수신기, 처리장치, 무선통신 장치 등을 장착한 다수의 이동물체
- 2) 데이터베이스 시스템
- 3) 무선통신 에이전트 등

이동물체는 주기적으로 특수하게 코드화 된 신호를 인공위성으로부터 받아, 이를 처리하여 시각과 위치 정보를 추출한다. 인공위성 신호는 미국 국방부로부터 무료로 제공되어 브로드캐스트 되기 때문에 모든 GPS 수신기가 동시에 수신할 수 있다. 따라서, 현재의 실제 위치를 파악한 이동물체는 무선통신 에이전트를 통하여 데이터베이스 관리시스템 (DBMS)에 이를 전송할 수 있다.

시스템이 사용하는 에이전트의 수가  $r$ 이라면 한 순간에 동시에 통신 할 수 있는 이동물체의 수는  $r$ 로 제한된다. 그리고 DBMS에 도착한 정보는 저장장치에 트랜잭션으로 보관되며, DBMS는 트랜잭션을 하나씩 차례로 가져다가 수행한다. DBMS는 갱신 작업을 하기 위하여 반드시 보조기억장치를 접근해야 하기 때문에, 갱

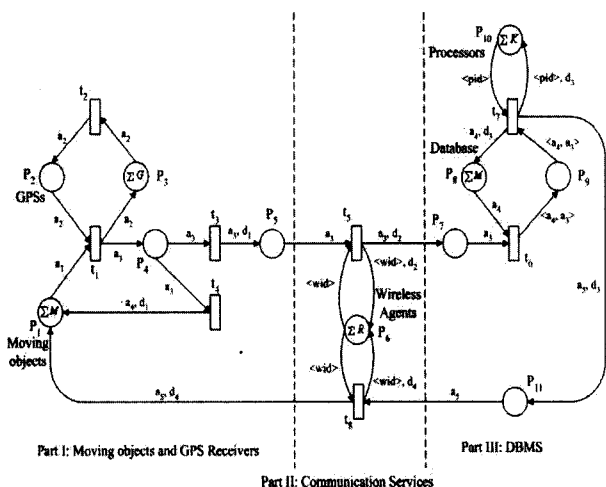


그림 1. 이동물체 데이터베이스 시스템의 FTPN 모형  
Fig. 1. Fuzzy-Timing Petri Net Model for Moving Objects Database System.

신 작업은 비교적 긴 시간을 필요로 한다.

(그림 1)에 보이는 FTPN은 이러한 이동물체 데이터베이스 시스템을 표현하는 모형을 보인 것으로, 그림에 보이는 원과 사각형을 각각 플레이스와 트랜지션이라고 일컫는다. 이 모형은 이동물체와 GPS 수신기, 통신서비스, 데이터베이스 관리 시스템 등 세부부분으로 구성되며, 각각의 하는 일은 바로 위에서 언급한 바와 같다. 각 플

표 1. 각 플레이스의 토큰  
Table 1. Tokens of Each Place.

플레이스	토큰 유형
P1	<mid, th>, 현재의 실제 위치와 시간 신호를 기다리는 이동물체들. 초기에는 모든 이동물체가 이 상태에 있음.
P2	<gid, gcl, gct>, 위성 신호를 수신한 GPS 수신기.
P3	{<gid>}, 위성 신호를 기다리는 GPS 수신기들. 초기에는 모든 GPS 수신기들이 이 상태임.
P4	<mid, mcl, mct, th>, 현재의 실제 (위치, 시각) 정보를 가지고 있는 이동물체들.
P5	<mid, mcl, mct, th>, 현재의 실제 (위치, 시각) 정보를 데이터베이스로 보내고자 하는 이동물체들.
P6	{<wid>}, 이동물체를 서비스하기 위하여 기다리고 있는 무선통신 에이전트들. 초기에는 모든 에이전트가 이 상태임.
P7	<mid, mcl, mct, th>, 데이터베이스 갱신을 위하여 기다리는 이동물체의 메시지들.
P8	<db, mid>, 이동물체로부터 정보를 기다리는 트랜잭션.
P9	<db, mid, mcl, mct, th>, 수행을 기다리는 트랜잭션.
P10	{<pid>}, DBMS 처리장치.
P11	<mid, mreinfo>, return 값을 가지고 있는 이동물체들.

표 2. 트랜지션 설명  
Table 2. Operation of Each Transition

트랜지션	역할
t1	이동물체가 현재의 실제 위치와 시각을 얻는 작업.
t2	GPS 수신기가 현재의 실제 위치와 시각의 신호를 위성으로부터 수신하는 작업
t3	이동물체가 갱신 정책을 적용하여 갱신을 해야 하는지 판단하는 작업
t4	이동물체가 t3의 반대 경우를 판단함.
t5	무선통신 에이전트가 갱신 메시지를 전달함
t6	이동물체가 데이터베이스 갱신 트랜잭션을 생성하는 작업
t7	데이터베이스 처리 장치가 갱신 작업을 수행
t8	이동통신 에이전트가 리턴 정보를 전송함

레이스와 트랜지션, 그리고 간선에 대한 설명은 각각 <표 1>, <표 2> 그리고 <표 3>에 요약된 바와 같다. (그림 1)에 보이는 페트리 넷의 초기 마킹, M0은 다음과 같다:

$$M_0(P_1) = M_0(P_8) = \sum M = n \text{ 이동물체} = \{mid \mid mid = 1, 2, \dots, n\}$$

$$M_0(P_3) = \sum G = n \text{ GPS 수신장치} = \{gid \mid gid = 1, 2, \dots, n\}$$

$$M_0(P_6) = \sum R = r \text{ 무선 에이전트} = \{wid \mid wid = 1, 2, \dots, r\}$$

$$M_0(P_{10}) = \sum K = \text{DBMS의 } k \text{ 처리장치} = \{pid \mid pid = 1, 2, \dots, k\}$$

나머지 플레이스 P<sub>i</sub> 에 대해서는 M<sub>0</sub>(P<sub>i</sub>) = Φ .

표 3. 간선과 간선의 레이블 설명  
Table 3. Explanation of Each Arc Inscription.

간선	레이블과 설명
a1	<mid, th>, mid는 이동물체의 고유 번호, th는 한계값
a2	<gid, gcl, gct>, gid는 GPS 수신기 고유번호, gcl은 현재의 실제 위치, gct는 현재 시각
a3	<mid, mcl, mct, th>. 고유번호 mid인 이동물체가 현재 위치와 시각 (mcl, mct), 그리고 한계값 (th)을 가지고 있음.
a3, d1	d1은 갱신 정책을 적용하여 갱신 여부를 판단하는 작업 소요 시간
a3, d2	d2는 무선통신 에이전트의 통신에 소요되는 시간.
a4	<db, mid>, db는 이동물체 mid의 정보를 다루는 데이터베이스.
<a4, a3>	<db, mid, mcl, mct, th>, 이동물체 mid, 관련 정보, 한계값, 이 물체의 정보를 관리하는 데이터베이스 db
a4, d1	d1은 갱신 정책에 따라 갱신을 하지 않아도 됨을 판단하는데 걸리는 시간
a4, d3	d3는 하나의 이동물체가 데이터베이스 갱신을 일회하는 데 소요되는 시간
a5	<mid, mreinfo>, mreinfo는 이동물체로 리턴되는 정보.
a5, d3	d3는 데이터베이스 db가 하나의 트랜잭션 처리에 소요되는 시간.
<wid>	wid는 무선통신 에이전트의 고유번호
<wid>, d2	d2는 갱신 메시지를 전송하기 위하여 소요되는 통신 시간
<wid>, d4	d4는 리턴 정보를 전송하기 위하여 소요되는 통신 시간
<pid>	pid는 DBMS 처리장치의 고유번호
<pid>, d3	d3는 처리장치 pid의 갱신 작업에 소요되는 시간

2.2 분석 방법

이동물체 데이터베이스의 목적은 이동물체의 위치와 관련된 각종 정보를 사용자에게 정확하게 제공하는 것이다. 그러므로, 이동물체 데이터베이스 시스템 분석에서 가장 중요한 것은 이동물체들이 과연 주어진 시간 내에 데이터베이스를 모두 성공적으로 갱신할 수 있는가 하는 것이다. 예를 들어, 이동물체의 수가 n이고, 이동물체가 초속 10m의 속도로 움직이고 있는 이동물체 데이터베이스 시스템이 '이동물체 m의 위치는 좌표 (x,y) 지점 반경 10 m 이내이다'라는 대답을 제공하려면, 본 시스템은 n 개의 갱신 트랜잭션을 1초 내에 처리할 수 있어야 한다.

시간 페트리 넷(Timed Petri Net)에서 초기 마킹으로부터 시작하여 모든 트랜지션을 최소 한번 이상 격발하고 초기 마킹으로 다시 돌아오는데 소요되는 최소시간을 최소 사이클 시간(Minimum Cycle Time)이라고 하며, 이것을 찾기 위한 수식이<sup>[20]</sup> 소개되었다. 최소 사이클 시간이 바로 n개의 이동물체를 한번 서비스하는데 소요되는 시간에 해당하므로, 이 방법을 적용하여 (그림 1)의 최소 사이클 시간을 찾고자 한다. 이 방법은 시간 페트리 넷에 적용되는 방법이므로 (그림 1)을 우선 (그림 2)와 같이 시간 페트리 넷으로 변형한다.

(그림 2)의 페트리 넷을 행렬로 표기하면 A와 같다.  $A^T x = 0$ 을 만족하는 벡터 x를 T-불변(T-invariant)이라 하며, 각 트랜지션을 x의 해당 항만큼 격발하면 본래의 마킹으로 돌아오는 성질이 있다. (그림 2)의 경우에는 초기 마킹에서 다음과 같은 순서로 격발하면 원래의 초기마킹으로 되돌아옴을 알 수 있다:  $\langle t_2, t_1, t_4, t_2, t_1, t_3, t_9, t_5, t_6, t_7, t_{10}, t_9, t_8 \rangle$ .

	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$	$P_{11}$	$P_{12}$	$P_{13}$
$t_1$	-1	-1	1	1	0	0	0	0	0	0	0	0	0
$t_2$	0	1	-1	0	0	0	0	0	0	0	0	0	0
$t_3$	0	0	0	-1	1	0	0	0	0	0	0	0	0
$t_4$	1	0	0	-1	0	0	0	0	0	0	0	0	0
$t_5$	0	0	0	0	-1	1	1	0	0	0	0	-1	0
$t_6$	0	0	0	0	0	0	-1	-1	1	0	0	0	0
$t_7$	0	0	0	0	0	0	0	1	-1	-1	1	0	1
$t_8$	1	0	0	0	0	1	0	0	0	0	-1	-1	0
$t_9$	0	0	0	0	0	-1	0	0	0	0	0	1	0
$t_{10}$	0	0	0	0	0	0	0	0	0	1	0	0	-1

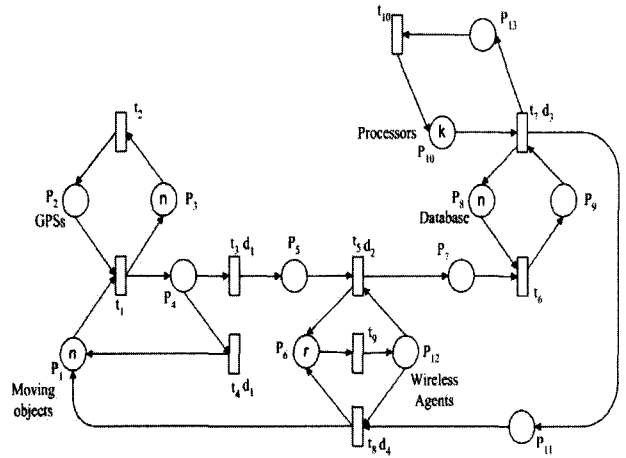


그림 2. 그림 1을 변형하여 얻은 시간 페트리 넷  
Fig. 2. Timed Net Model Obtained From Fig. 1.

따라서 다음과 같은 T-불변이 있음을 알 수 있다.

$$(2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 1)^T \tag{1}$$

한편,  $Ay=0$ 을 만족하는 벡터 y를 S-불변(S-invariant)이라고 한다. 즉, 초기 마킹  $M_0$ 으로부터 도달 가능한 임의의 마킹  $M_i$ 에 대하여 다음 성질이 만족한다.

$$y^T M_i = y^T M_0 \tag{2}$$

(그림 2)에 대한 S-불변은 다음과 같다:

$$\begin{aligned} y_1^T &= (0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0), \\ y_2^T &= (0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0), \\ y_3^T &= (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0), \\ y_4^T &= (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1), \\ y_5^T &= (1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0). \end{aligned} \tag{3}$$

따라서 최소 사이클 시간은 다음 식을 만족한다.

$$\text{최소 사이클 시간} = \text{Max} \{y_k^T (A^{-T} D x / y_k^T M_0)\} \tag{4}$$

(그림 2)의 경우, x는 (1)에 주어진 T-불변,  $y_k$ 는 (3)에 주어진 다섯 개의 S-불변,  $A^{-1} = [a_{ij}]_{n \times m}$ , with  $a_{ij} = w(P_j, t_i)$ , 대각행렬  $D = d_i, i = 1, 2, \dots, n$ , 그리고 초기 마킹  $M_0$ 은 다음과 같다:

$$M_0 = (n \ 0 \ n \ 0 \ 0 \ r \ 0 \ n \ 0 \ k \ 0 \ 0 \ 0 \ 0),$$

n은 이동물체의 수, r은 무선통신 에이전트의 수, k는 DBMS의 처리장치 수를 의미한다. (그림 2)의 다섯개 S-불변 각각에 대한 식 (4)의 분모를 구하면 다음과 같다:

$$y_1^T M_0 = n, \quad y_2^T M_0 = r, \quad y_3^T M_0 = n, \quad y_4^T M_0 = k, \\ y_5^T M_0 = n.$$

(그림 2)의 행렬  $A^-$  는 다음과 같다:

$$A^- = \begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & R_{10} & R_{11} & R_{12} & R_{13} \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \\ t_{10} \end{matrix} & \begin{bmatrix} +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 \end{bmatrix} \end{matrix}$$

또한 지속시간 행렬  $D$ 는 다음과 같다:

$$D = \begin{matrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \\ t_{10} \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

따라서,  $(A^-)^T D x = (0 \ 0 \ 0 \ 2d_1 \ d_2 \ 0 \ 0 \ 0 \ 0 \ d_3 \ d_3 \ d_4 \ d_2 + d_4 \ 0)^T$ 이며, 식(4)의 값은 다음과 같다:

$$y_1^T (A^-)^T D x / y_1^T M_0 = 0 \\ y_2^T (A^-)^T D x / y_2^T M_0 = (d_2 + d_4) / r \\ y_3^T (A^-)^T D x / y_3^T M_0 = d_3 / n \\ y_4^T (A^-)^T D x / y_4^T M_0 = d_3 / k \\ y_5^T (A^-)^T D x / y_5^T M_0 = (2d_1 + d_2 + d_3 + d_4) / n$$

그리고, (그림 2)에 대한 최소 사이클 시간은 다음과 같다:

$$\text{Max} \{0, (d_2 + d_4) / r, d_3 / n, d_3 / k, (2d_1 + d_2 + d_3 + d_4) / n\} \quad (5)$$

지속시간  $d_1$ 은 간단한 계산과 비교 연산 수행 시간이므로 700 microsecond,  $d_2$ 는 통신 시간이므로 1,000 mi

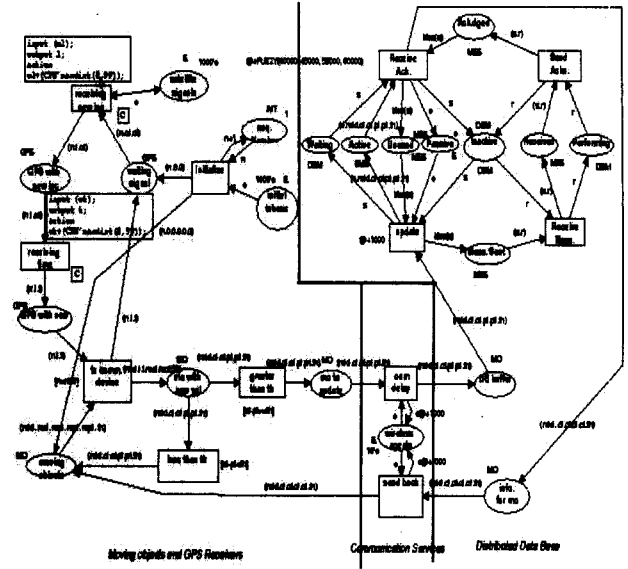


그림 3. Design/CPN으로 구축한 이동물체 데이터베이스의 페트리 넷 모형

Fig. 3 Petri net model for MOD built in Design/CPN.

-crosecond,  $d_3$ 는 데이터베이스 갱신 시간이므로 50,000 microsecond,  $d_4$ 는 통신 시간이므로 1,000 microsecond,  $r$ 은 통신 에이전트의 수이므로 10,  $n$ 은 이동물체의 수이므로 1,000,  $k$ 는 처리장치의 수이므로 1 이라 하면, 최소 사이클 시간은 50,000 마이크로초이며, 데이터베이스 갱신 작업 수행 속도에 의하여 결정됨을 알 수 있다.

$$d_1 = 700, \quad d_2 = 1,000 \quad d_3 = 50,000 \quad d_4 = 1,000 \quad r = 10 \quad n = 1,000 \quad k = 1$$

### III. 시뮬레이션 모형

(그림 1)에 보이는 이동물체 데이터베이스의 페트리 넷 모형은 갱신 정책과 데이터베이스 관리 시스템 부분이 너무 추상적이다. 따라서, 갱신 정책으로 '거리기반 갱신 정책'을 반영하고, 데이터베이스 관리 시스템으로 분산 데이터베이스의 처리 과정을 반영한 이동물체 데이터베이스 시스템의 페트리 넷 모형을 (그림 3)에 보였다.

이 모형은 페트리 넷 시뮬레이션 도구인 Design/CPN<sup>[6]</sup>으로 구축한 것이며, 트랜지션 'Receiving new loc.'에 연동된 다음과 같은 코드가 현재의 실제 위치를 구하는 GPS 수신기의 작업을 흉내낸다:

```
input (ol);
output l;
```

```
action ol+(CPN'randint(0,99));
```

'input(ol);' 은 입력 간선의 레이블을 구성하는 변수 중 'ol' 이 코드의 출력값 계산에 사용된다는 것을 의미하며, 'output l' 은 코드 수행 결과가 출력 간선의 레이블을 구성하는 변수 'l' 로 출력된다는 것을 의미한다. 트랜지션 'greater than th' 에 연관된 guard, 즉 [cl-pl>=th]는 현재 위치와 지난번 위치와의 차이가 한계값 (th)를 넘는지 비교하여, 넘을 때에만 갱신 정보를 데이터베이스로 전송한다.

이동물체 데이터베이스는 일반적으로 분산 데이터베이스이다. 분산 데이터베이스에서는 데이터 항목이 여러 장소에 복사되어 있기 때문에 갱신 트랜잭션이 도착하면 자기 사이트에 소속된 데이터 항목 카피의 내용을 갱신함과 동시에 나머지 모든 사이트에 갱신 메시지를 전송한 다음, 이들로부터 갱신 완료 Acknowledge가 모두 도착해야 갱신 작업이 완료된다.

무선통신 지속 시간이 1 ms이라고 가정하여 '@+1,000' 을 트랜지션 'update'에 연관시켰다. HP97560의 접근시간이 평균 18.2 ms임<sup>[21]</sup>을 감안하여, 데이터베이스 갱신을 위한 지속시간으로 '@+FUZZY (40,000,

```
fun FUZZY (a: int, b:int, c:int, k:int)
let
val atime = CPN'randint(a, k);
in
(if atime >= anadalso atime <= c
then atime
else if (atime < b)
then if (a<b)
then if (10*(atime-a) div (b-a) >=
CPN'randint(0,10)
then atime
else FUZZY(a, b, c,
k)
else if (atime = a) then a
else FUZZY (a, b, c, k)
else
if (k>c)
then if (10*(atime-c) div (c-k)+10 >= CPN'randint(0,
10))
then atime
else FUZZY (a, b, c, k)
else if (atime = c) then c
else FUZZY(a, b, c, k)
)
end;
```

그림 4. FUZZY 함수  
Fig. 4. Function FUZZY.

45,000, 55,000, 60,000)' 을 'Receive Ack.' 에 연관시킨다. FUZZY(40,000, 45,000, 55,000, 60,000) 값을 구할 함수 FUZZY는 (그림 4)와 같이 정의된다.

40,000과 60,000 사이의 임의의 수 x를 추출하여, (그림 5)에 보이는 가능성 값에 따라 x를 함수 값으로 반환한다. 즉, x가 45,000과 55,000 사이이면 x가 함수 값이 될 가능성은 1이고, 아니면 x가 함수 값이 될 가능성은 (그림 5)의 x에 연관된 값에 따라 결정된다. 예를 들어, x가 42,500이면 42,500이 함수 값이 될 가능성은 그림 5에 보이는 바와 같이 50%이다. 또 다른 예로, x가 44,000일 때, 44,000이 함수 값이 될 가능성은 80%이다.

2절에서 살펴 본 분석 방법을 (그림 3)에 그대로 적용하는 것도 이론적으로는 가능하지만 계산이 너무 복잡하여 실제적으로는 불가능하다. 이러한 경우에는 시뮬레이션이 적당한 분석 방법이다. (그림 3)에 대해 시뮬레이션을 수행하면 무한히 종료하지 않으므로, (그림 6)과 같은 시뮬레이션을 위한 모델로 변형한다.

(그림 6)이 (그림 3)과 다른 점은 시뮬레이션을 유한 시간 내에 종료할 수 있도록 플레이스 'Sink' 를 추가한 것과, 시뮬레이션 시간을 기록하기 위한 코드가 트랜지션 'Recording' 에 추가된 것이다. 'receivetime' 은 임의의 변수 이름이고, 'time()' 은 현재 시뮬레이션 시각을 생성하는 함수이다. 'delaytime' 은 임의의 파일 이름이며, 다음 문장은 'delaytime' 이라는 파일에 mid와 receivetime을 출력한다.

```
(output (worldresponse, makestring mid ^ "\t"
^ Real.toString (IntInfToReal 6 receivetime) ^ "\n" );
```

#### IV. 시뮬레이션 결과

시뮬레이션의 결과로 (그림 7)과 같은 갱신에 소요되

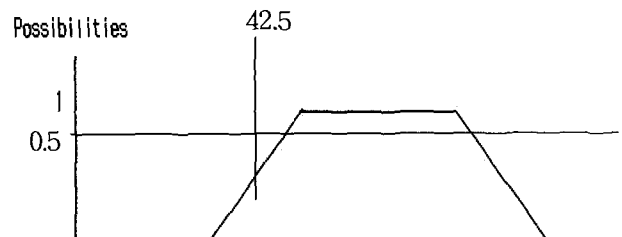


그림 5. FUZZY(40,000, 45,000, 55,000, 60,000)의 가능성 분포  
Fig. 5. Possibility distribution of FUZZY(40,000, 45,000, 55,000, 60,000).

는 시간을 얻었다. 100 개의 이동물체 중 약 50%는 소요시간이 0이다. 이것은 서론에서 언급한 바와 같은 갱신 정책을 사용하여 이동물체 중 50%만 갱신을 요구한다고 가정하여, 트랜지션 'less than th' 에 사용된 한계값이 50이기 때문이다. 가장 큰 소요시간은 2.685325 초이다. 이것은 이동물체가 100 개일 때, 이동물체 모두가 갱신 정책에 의하여 데이터베이스 갱신을 1회씩 하는데 소요되는 시간을 말한다. 위에서 언급한 바와 같이, 이동물체의 수가 100이고, 이동물체가 초속 10m의 속도로 움직이고 있는 이동물체 데이터베이스 시스템이 '이동물체 m의 위치는 좌표 (x,y) 지점 반경 10 m 이내이다' 라는 대답을 제공하려면, 본 시스템은 100 개의 갱신 트랜잭션을 1초 내에 처리할 수 있어야 한다.

시물레이션 결과 현 상황에서는 2.685325 초를 소요함으로 사용자 요구사항을 충족하지 못함을 알 수 있다. 이러한 경우에는 서비스 받는 이동물체의 수를 줄이든지, 더욱 효율적인 갱신 정책을 구현한 소프트웨어를 구입하든지, 아니면 더욱 성능이 좋은 데이터베이스 시스템으로 교체해야 한다.

똑같은 시스템에서 이동물체의 이동 속도가 초속 3.3 m라면 데이터베이스 갱신 소요 시간 2.685325 초는 시스템의 요구사항을 만족한다. 이러한 경우에는 더 많은 사용자를 서비스하는 경우를 고려하여 이동물체의 수를 늘리든가, 저렴한 갱신 정책 소프트웨어를 사용하는 경우를 고려하여 한계값, th를 감소하든가, 통신비용을 절

약하기 위하여 통신 에이전트 수를 감소하든가, 아니면 저렴한 데이터베이스 관리 시스템을 사용하는 경우를 고려하여 'Receive Ack.' 에 연합된 시간을 증가시켜 볼 수 있을 것이다.

### V. 결 론

이동물체 데이터베이스의 사용이 급격히 증가하고 있다. 따라서, 이동물체 데이터베이스에 대한 연구도 활발히 진행 중이다. 이 분야에서 가장 중점적으로 연구하는 주제는 이동물체의 과거 경로를 분석하여 미래의 위치를 정확히 추정함으로써 데이터베이스 갱신 비용을 절감하는 것이다. 그러나, 아무리 효율적인 갱신 정책을 개발하더라도 시스템 설계자의 입장에서는 시스템의 궁극적인 설치 목적인 사용자 요구사항을 설계 중인 시스템이 충족하는지를 시스템 설계 초기에 검증하여 보아야 한다.

본 논문은 이동물체 데이터베이스에 대한 페트리 넷 모형과 분석 방법을 소개하고, Design/CPN을 이용한 시물레이션 모형 및 시물레이션 방법을 소개하였으며, 이를 이용한 이동물체 데이터베이스 사용자 요구사항 분석 방법을 소개하였다. 즉, 시스템 구성 요소들의 값을 (그림 6)과 같은 모형에 대입하고 Design/CPN을 이용한 시물레이션을 통해 요구 분석 사항에 대한 결과를 얻을 수 있음을 보였다.

시물레이션 결과가 요구사항을 만족할 때에는 비용 요소를 절감하기 위하여 구성 요소들의 값을 변경하고, 요구사항을 만족하지 못하는 경우에는 비용을 증가하면서 구성 요소들의 값을 변경함으로써 최소의 경비로 사용자 요구 사항을 만족하는 최적의 시스템을 설계할 수 있다. 본 논문은 이동 물체 데이터베이스에 대한 시물

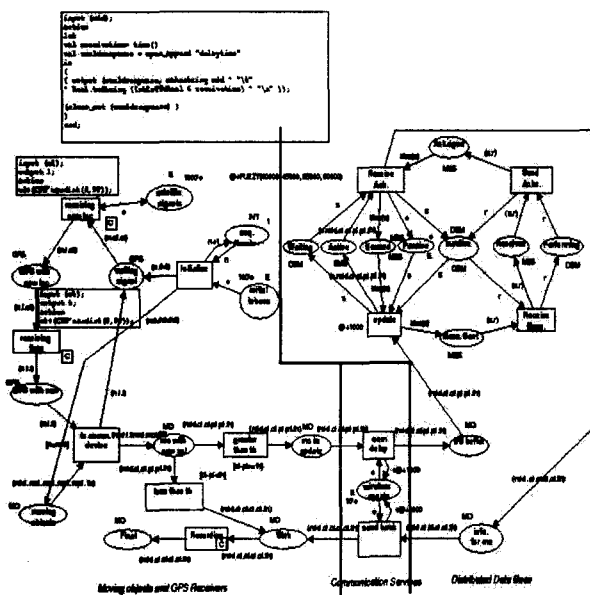


그림 6. 시물레이션을 위한 페트리 넷 모형  
Fig. 6. Petri net model for simulation.

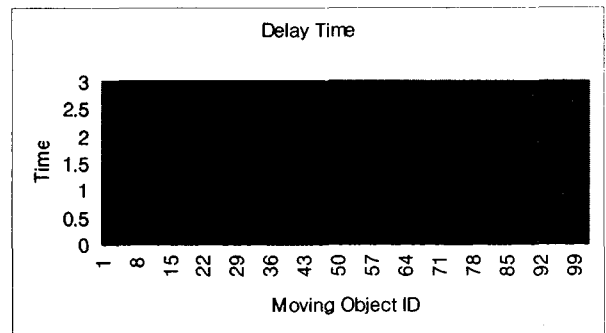


그림 7. 시물레이션 결과 얻은 소요 시간  
Fig. 7. Delay time obtained from simulation.



레이션 모형을 최초로 소개하였다는 점에 의의가 있으며, 본 모형은 Design/CPN을 이용하여 구축하였으므로 시뮬레이션을 자동으로 수행할 수 있다는 장점이 있다.

### 참 고 문 헌

- [1] [http://www.colorado.edu/geography/gcraft/notes/gps/gps\\_f.html](http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html)
- [2] O. Wolfson, "Moving objects information management: The database challenge", in the Proceedings of the 5th Workshop on Next Generation Information Technologies and Systems (NGITS'2002), Israel June 2002.
- [3] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: To update or not to update?" ACM/Baltzer Wireless Networks Journal, 1(2).1999.
- [4] O. Wolfson, A.P. Sistla, S. Chamberlain, and Y. Yesha, "Updating and Querying Databases that Track Mobile Units". Special issue of the Distributed and Parallel Databases Journal on Mobile Data Management and Applications, 7(3), 1999.
- [5] O. Wolfson and H. Yin, "Accuracy and Resource Consumption in Tracking and Location Prediction", Proceedings of the 8th International Symposium on Spatial and Temporal Databases, Santorini Island, Greece, July 24-27.
- [6] K. Jensen, Design/CPN [Online]. Dept. Computer Science, Univ. Aarhus, Denmark. Available: <http://www.daimi.au.dk/designCPN/>.
- [7] L. Anneberg and H. Singh, "Petri Net Approach for Requirement Analysis Phase of Software Engineering," Proceedings of the Twenty-First Annual Pittsburgh Conference, 1990, Pittsburgh, PA, USA; Part 3, pages 1015-1019, 1990.
- [8] Jensen Kurt, "Coloured Petri Nets: A High Level Language for System Design and Analysis," Lecture Notes in Computer Science, Vol. 483; Advances in Petri Nets 1990, Berlin, Germany: Springer-Verlag, pages 342-416, 1991.
- [9] M. Bax and N. Giambiasi, "Modeling and simulating temporal constraints in OpenRT," Proc. 1996 Simulation Multiconference - Simulators International XIII, 8-11 April 1996, New Orleans, LA, pages 128-133. 1996.
- [10] A. Alekseev, A. Bystrov, S. Kurtov, S. Mylnikov, V. Nepomnyashchy, E. Okunishnikova, P. Chubarev, and T. Churina, "Application of Petri nets in verification of distributed systems represented in the Estelle language," Journal of Computer and Systems Sciences, Vol. 38, No. 5, pages 771-781. 1999.
- [11] P.Langner, C. Schneider and J. Wehler, "Petri Net Based Certification of Event-driven Process Chains," Lecture Notes in Computer Science, Vol. 1420, Berlin: Springer-Verlag, 19th Int. Conf. on Application and Theory of Petri Nets, ICATPN'98, Lisbon, Portugal, June 1998, pages 286-305, 1998.
- [12] D. Gurovic, W. Fengler and J. Nutzel, "Development of real-time system specifications through the refinement of duration interval Petri nets," Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'2000), 8-11 October 2000, Nashville, TN, Vol. 4, pages 3093-3098. 2000.
- [13] O. Roux, D. Delfieu, and P. Molinaro, "Discrete Time Approach of Time Petri Nets for Real-Time Systems Analysis," ETFA2001 proceedings of the 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation, Volume 2, IEEE Catalog number : 01TH8597, pages 197-204. October 2001.
- [14] J. Wang, Y. Deng, and G. Xu, "Reachability analysis of real-time systems using time Petri nets," IEEE Trans. on Systems, Man, and Cybernetics; B: Cybernetics, Vol. 30, No. 5, pages 725-736. 2000.
- [15] T. Murata, "Temporal Uncertainty and Fuzzy-Timing High-Level Petri Nets," in Application and Theory of Petri Nets 1996, Lecture Notes in Computer Science, Vol. 1091, Springer, New York, pp. 11-28.
- [16] T. Murata, J. Yim, H. Yin, O. Wolfson, "Fuzzy-Timing Petri-Net Model for Updating Moving Objects Database," to appear in IPSI 2003.
- [17] T. Murata, "Temporal Uncertainty and Fuzzy-Timing High-Level Petri Nets," in Application and Theory of Petri Nets 1996, Lecture Notes in Computer Science, Vol. 1091, Springer, New York, pp. 11-28.
- [18] T. Murata, T. Suzuki and S. Shatz, "Fuzzy-Timing High-Level Petri Nets (FTHNs) for Time-Critical Systems," in J. Candoso and H. Camargo (editors) "Fuzziness in Petri Nets" Vol. 22 in the series "Studies in Fuzziness and Soft Computing" by Springer Verlag, New York, pp. 88-114, 1999.
- [19] Y. Zhou, T. Murata, and T. DeFanti, "Modeling and Performance Analysis Using Extended Fuzzy-Timing Petri Nets for Networked Virtual Environments," IEEE Transactions on Systems,

- Man, and Cybernetics Part B: Cybernetics, Vol. 30, No. 5, October 2000, pp.737-756.
- [20] T. Murata, "Petri Nets: Properties, Analysis and Applications," Proceedings of the IEEE, Vol. 77, No. 4, April, 1989, pp.541-580.
- [21] T. Murata and C. Chen, "Fuzzy-Timing Petri-Net Modeling and Analysis on Video-On-Demand System Response Times,"

---

저자 소개

임재걸(정회원)  
제38권 CI편 제5호 참조.

이계영(정회원)  
제38권 CI편 제5호 참조.