

실시간 시스템 모형

건국대학교 김문희*
 한국외국어대학교 김정국*
 시립인천대학교 민병준*
 숭실대학교 양승민*

● 목 차 ●

- | | |
|-------------------|---------------------|
| 1. 서 론 | 2.4 RTTL |
| 2. 기존의 실시간 시스템 모형 | 3. 실시간 객체 모형: RTO.k |
| 2.1 DARTS | 4. IVS 개발 |
| 2.2 TPN | 5. 결 론 |
| 2.3 RTL | |

1. 서 론

컴퓨터 시스템이 실시간 시스템, 특히 경성 실시간 시스템의 제어 및 운용 장치로 많이 이용되고 있으며 그 운용 분야는 점점 더 확산되고 있다. 따라서 그러한 시스템의 개발 방법, 특히 실시간 소프트웨어의 체계적인 개발 방법이 절실히 요구된다. 이제까지의 실시간 소프트웨어들은 체계적인 방법론에 의해서라기 보다 그때 그때 시스템의 특성에 따라 설계되어 왔다. 많은 시스템들이 여러 종류의 센서나 터미널로부터 들어오는 입력을 인터럽트 방식에 의해 처리하는 형태로 설계되었고 어셈블리어가 주된 구현 언어로 사용되어 왔다. 그러나 대상 실시간 시스템의 규모와 복잡도가 커짐에 따라 이제 까지의 개발 방법은 적합하지 못하다. 즉, 소프트웨어 모듈간의 상호 작용과 동기화 그리고 많은 경우 분산/병렬처리가 필수적인데 이때 설계된 시스템의 올바른 수행(즉 logical correctness)과 실시간성(즉, time cor-

rectness)보장이 어렵게 된다. 따라서 경성 실시간 시스템들의 특성을 분석한 후, 보다 체계적인 설계 방법에 의한 시스템 개발이 필요하다. 그럼으로써 개발 기간과 개발 경비를 줄이고 신뢰성 높은 시스템을 개발할 수 있다.

경성 실시간 시스템의 설계시 다음의 세가지 사항을 고려하여야 한다. 첫째, 시스템이 점점 대형화되고 복잡해지므로 소프트웨어의 모듈화와 자동화가 이루어져야 하는데 이를 위한 보다 체계적인 방법이 필요하다. 둘째, 실시간성이 보장되어야 한다. 구현된 시스템의 실시간 응답 보장을 위해서는 설계시 그에 대한 검증이 가능하여야 한다. 특히, 많은 경우 지리적인 이유 또는 연산 능력이나 신뢰도를 높이기 위해 분산 병렬처리를 하게 되는데 각 노드간의 통신 지연과 동기화 문제를 고려해야 하기 때문에 실시간성 보장 검증이 어렵게 된다. 셋째, 시스템이 수행하는 업무의 중요성(예를 들어, 비행기나 무기 시스템 제어) 때문에 고신뢰도가 보장되어야 한다. 시스템 설계나 구현시 결함을 줄이기 위한 방법의 필요성은 물론 결함 발생시 그것을 감지하고 복구하는 기능, 그리

고 필요한 경우 시스템 재구성까지 할 수 있어야 한다. 따라서 경성 실시간 시스템 개발을 위해서는 위의 세 가지 특성을 고려하여 설계 방법이 개발되어야 한다.

특히, 실시간 시스템의 기능적 요구사항뿐만 아니라 시간적 요구사항을 명확하고 효율적으로 모형화할 수 있는 실시간 시스템의 모형의 역할은 매우 중요하다. 실시간 시스템 개발을 위한 모형이 갖추어져야 할 조건은 다음과 같다.

- 시스템 추상화가 용이하여야 한다.
- 시간요구 사항을 명확히 표현할 수 있어야 하며 검증이 가능하여야 한다.
- 소프트웨어 요소뿐만 아니라 하드웨어 요소에 대한 명세가 가능하여야 한다.
- 예외처리나 결함허용기능을 표현할 수 있어야 한다.
- 모형으로부터 구현이 자동 또는 반자동으로 이루어져야 한다.
- 시스템 설계작업과 분리되어 이루어져서는 안된다.

본 고에서는 기존의 실시간 시스템의 모형화 기법인 Timed Petri-Net(TPN), Real-Time Logic(RTL), Real-Time Temporal Logic(RTTL) 방식 등을 소개하고 그들의 장점 및 한계를 논하고 이들의 한계를 극복하기 위한 시도를 현재 저자들이 공동으로 연구하고 있는 실시간 객체모형에 관하여 소개한다. 또한 실시간 객체 모형의 유효성을 보여주기 위해 부인 자동차 시스템(IVS)의 설계를 실시간 객체 모형으로 기술한다.

2. 기존의 실시간 시스템 모형

실시간 시스템의 설계 단계에서는 시스템 기능의 논리적 특성 뿐 아니라 시간 제약 사항이 반영된 시간적 특성이 기술되고 검증되어야 한다. 일반 시스템의 모형화 방법에서 발전하여 시간적 특성을 모형화하려는 노력이 지난 십여년간 경주되었다. 이 연구 결과를 유형별로 나누어 보면 크게 세가지로 구분할 수 있다. 첫째는 시스템의 구조화 설계 방법을 토대로 한 것이다. 시스템을 계위적 모듈들로 분해하고

이들 간의 데이터 흐름을 분석하는 방법이다. 프로세스 기반 방식과 그래프를 이용하는 방법이 있다. 그래프에 의한 방법은 보다 정확한 시스템 검증 가능성을 제시하고 있지만 동시에 프로세스 스케줄 가능성 분석에 대한 일반적인 해답이 존재하지 않는 문제가 있다. 2.1절에서는 프로세스 기반 구조화 설계 방법의 확장으로 GE에서 개발한 DARTS(Design Approach for Real-Time Systems)를 소개한다. 두번째는 페트리 네트의 확장에 의한 방법이다. 동시성(concurrency)과 동기화(synchronization) 문제를 엄밀하게 표현할 수 있고 계위적 분해가 적용될 수 있다는 페트리 네트의 장점으로 인하여 여러 형태의 페트리 네트 확장 모형이 개발되었다. 이들중에서 대표적인 TPN(Timed Petri Net)에 대하여 2.2절에서 논한다. 세번째 유형은 시스템 설계의 분석에 정리 증명(theorem-proving) 기술을 적용한 것이다. 시간적 특성을 표현하고 검증하는 방법으로 RTL(Real-Time Logic)과 RTTL(Real-Time Temporal Logic)을 각각 2.3절과 2.4절에서 간략히 소개한다.

2.1 DARTS

이 모형은 시스템내의 스케줄할 수 있는 개체인 프로세스를 모형의 요소로 사용한다. 사건 발생에 대하여 출력을 제공하는 상호작용 실시간 시스템의 경우 데이터 흐름 특성이 시스템의 논리적 시간적 행위를 결정하는 중요한 수단을 제공한다. DARTS는 구조화 설계 방법의 확장으로 GE에서 개발한 것이며, 최근 Ada 기반 설계로 발전되었다. DARTS의 설계는 요구사항 규격으로부터 시작된다. 데이터 흐름 분석은 필요한 주요 기능을 결정하는 도구를 제공한다. 데이터 흐름 분석은 변환(transform), 데이터 저장, 그리고 변환간의 데이터 흐름으로 구성된다.

데이터 흐름 분석이 완료되면 설계자는 확인된 변환을 기반으로 시스템 프로세스를 분해하는 방법을 결정한다. 데이터 흐름 분석을 통하여 각 변환의 동시성을 유출해 낼 수 있다. 변환간의 비동기적 성질을 분해 과정의 주안점으로 이용할 수 있다. 기능적 또는 시간적으로

결합된 변환들은 하나의 프로세스로 합치고 모듈 차원에서 이들을 구분할 수 있도록 한다. 위급성이나 특별한 요구가 주어진 변환은 독립된 프로세스로 한다. DARTS는 데이터 전송을 두 개의 모듈, MCM(Message Communication Module)과 IHM(Information Hiding Module)을 이용한다. 하나는 메시지 통신 방법이고 다른 하나는 공유 데이터에 의한 것이다.

결론적으로, DARTS는 시스템을 계위적 모듈들로 분해하고 이들 간의 데이터 흐름을 분석하는 방법으로 데이터 흐름 기반의 단일 응용 시스템 설계에 적합하다. 이 경우, 주요 관심사는 분해와 시스템 성능상의 비용을 최적화하는 문제이다.

2.2 TPN

페트리 넷트는 인과관계와 독립성을 명백하게 표현하고 시스템에 대한 여러 차원의 추상화를 가능케 하는 등의 장점을 갖고 있다. 전통적인 페트리 넷트는 $PN=(P, T, A, M)$ 으로 표현된다. 여기서, P 는 장소(place)의 집합이고, T 는 전이(transition)의 집합이며, A 는 $\{P \times T\} \cup \{T \times P\}$ 입출력이며, 마지막으로, M 은 초기점을 나타낸다. 토큰들이 전이를 통하여 장소들 간을 이동하는 동적인 모형이다. 이동은 화이어링(firing)에 의하여 이루어진다.

페트리 넷트 모형에서 어떤 전이가 기준으로 선택되면 다른 전이는 기준 전이에 대해 상대적인 화이어링 비율(firing ratio)을 갖는다. 많은 경우에 전이의 화이어링 비율은 각 전이에 변수를 주어서 각 장소의 입력 출력을 나타내는 연립 방정식을 풀어서 계산할 수 있다. 계산이 되지 않는 경우는 화이어링 비율의 범위를 제한한다.

어떤 장소의 입력 전이가 화이어링된 시간으로부터 T_i 단위 시간 후에 출력 전이를 할 수 있는 준비가 되었다면, 이 때 T_i 는 그 장소의 수행 시간이 된다. 출력 전이가 준비된 시간부터 출력 전이의 화이어링까지의 대기 시간을 W_i 라고 하면, $(T_i + W_i)$ 는 장소 p_i 의 입력 전이와 출력 전이 화이어링 사이 시간이다.

TPN에서는 장소와 전이에 두가지 중요한

특성을 추가하는데, 그것은 MRFF(Maximum Relative Firing Frequency)와 MTIAT(Mimum Token Inter-Arrival Time)이다. MTIAT는 한 장소에 연속적으로 토큰이 도착할 때, 최소한의 시간 간격이다. 이들 간의 관계는 다음과 같다.

$$MTIAT(p_i) = T_i / MRFF(T_i),$$

여기서 t_i 는 장소 p_i 의 입력 전이이고 T_i 은 "driving cycle period"이다. 최악의 경우에도 한 장소가 한 개 이상의 토큰을 갖지 않는다면 그 장소는 "time-safe"하다고 한다.

결론적으로, TPN은 설계 단계에서 시간 특성을 검증할 수 있는 수단을 제공한다. 위에서 언급한 장점이 있는 반면에 TPN은 초기 상태에서 도달할 수 있는 상태들의 집합이 폭발적으로 증가할 수 있으며, 전이가 비결정적으로 행해지고 특정 시간에 연관 지을 수 있는 방법이 없는 등의 약점이 있어서, 경성 실시간 시스템에 적용하기 위해서는 효과적인 확장이 더 필요하다.

2.3 RTL

RTL의 연산 모형은 사건(event), 동작(action), 인과관계(causality relations), 시간 제약(timing constraints)으로 구성된다. 이 모형은 시스템 특성과 외부사건에 대한 시스템 영향을 묘사하는 1차 논리식들로 표현된다. RTL 시스템은 사건 발생을 나타내는 @로 시간을 표현한다. 즉, 사건 e 의 i 번째 발생은 $@(e, i)$ 에 이루어진다. RTL은 다음 세가지 유형의 상수를 이용한다.

- 동작 상수: 프리미티브 혹은 합성의 형태를 갖는다. 합성 상수의 경우, 동작들간의 순차 혹은 병렬 관계를 이용한 사건-동작 모형에 의해 선행 관계가 주어진다.
- 사건 상수: 세 가지 형태로 구분된다. 시작/종료 사건은 동작의 시작과 마감을 나타낸다. 전이 사건은 상태 속성을 바꾸는 것을 말한다. 외부 사건은 시스템 행위에 영향을 미칠 수 있는 사건으로 시스템에 의해서 생성되지 않는다.
- 정수: 사건 발생 시간과 연속적으로 발생

한 사건 발생 횟수를 나타낸다.

RTL은 시스템의 물리적인 상태에 대한 “assertion”을 대수 관계로 바꾸어준다. 이 관계식은 등호나 부등호로 적절한 전이 사건의 발생을 묘사한다. 다음의 특성들을 이용하여 시스템의 사건-동작 모형으로부터 공리들을 도출할 수 있다.

- 동작과 시작/종료 사건들간의 관계
- 주기적인, 신발적인 사건 제약 사항
- 전이 사건을 유발시키는 인과관계
- 불필요한 동작 수행을 방지하기 위한 인위적인 제약 사항의 부가

시스템의 시간적 특성(RTL assertion)은 논박에 의하여 유추되고 증명된다. RTL은 규격이 올바르게 정의되어 있고 시스템의 안전성이 보장되는가를 확인하는데 사용된다. 우선 RTL로 표현된 시스템 규격을 그래프로 만든다. 그래프의 각 노드는 동작 발생을 나타낸다. 이들 노드는 가중치가 주어진 화살표로 연결된다. 화살표에 주어진 숫자는 화살표가 출발한 노드에 더해는 상수이다. 화살표의 숫자를 계산하면서 노드를 줄여나가는 방법으로 규격이 올바르게 주어졌는가를 확인할 수 있다. 이 방법은 매우 엄밀하게 시간 특성을 검증할 수 있는 가능성을 제공하는 반면 복잡한 연산 시스템의 실시간 특성을 검증하기에는 한계가 있다.

2.4 RTTL

RTTL은 다음과 같은 연산자들을 갖는다.

- until(\cap)
- next(\odot)
- 궁극적으로(\diamond), $\diamond w$ 는 “true \cap w”이다. 즉 결국 w는 어느 상태에서 참값을 갖는다.
- 이제부터는(\square), $\square w$ 는 $\neg(\diamond(\neg w))$ 이다. 즉, 이제부터 w는 모든 상태에서 참값을 갖는다.
- unless(U)
- precede(P)

예를 들면,

① $w_1 \wedge (t=T) \rightarrow \diamond(w_2 \wedge (t \leq T+5)) : w_1$ 이 참이고 클럭 값이 T이면 5시간단위 내에 w_2 가 참이 되어야 한다.

② $w_1 \rightarrow w_2 P w_3 : w_1$ 이 참이면 w_2 는 w_3 보다 선행한다.

③ $\square \diamond (n=\text{tick})$: 클럭이 계속 움직인다.

RTTL에 의한 모형의 장점은 시간 특성을 엄밀하게 검증할 수 있다는 점이다. 반면에 실제 대규모 실시간 시스템에 적용하기에는 매우 복잡하다.

지금까지 실시간 시스템의 시간 특성을 표현하고 검증하는 기존의 방법들에 대하여 간략히 소개하였다. 각 방법들은 제각기 장점과 동시에 한계점을 갖고 있다.

3. 실시간 객체 모형 : RTO.k

일반적 병행 프로그램에서 부터 경성 실시간 응용에 이르기까지 프로세스 또는 thread 들이 가져야 할 특성과 객체지향 프로그래밍 기법의 장점을 결합한 실시간 프로그램 패러다임으로서의 실시간 객체 모델로는 RTO.k, RTC++ 등이 있다. 이들 중 주목 받는 모델의 하나인 RTO.k는 다음과 같은 목표를 추구하는 모델이다.

- 경성 실시간 응용에서 비실시간 응용 까지 적용할 수 있는 유연한 형태의 객체 모델
- 실시간 커널 서비스와 실시간 객체 모델을 빌딩 블럭화 하여 구성되는 실시간 서비스의 설계시 실시간 보장(design-time guarantee of timely service)

일반적 객체 지향 프로그램의 객체가 객체 데이터 공간과 member 함수들로 구성되는데 반해 RTO.k는 객체의 특성을 유지하면서 그 구성요소가 데이터 공간과 동적인 member thread(method)로 구성되는 점이 다르다. RTO.k의 구성 및 특징을 요약하면 다음과 같다.

- 두가지 기능 그룹의 method 들로 구성된다 : 첫번째 그룹은 실시간 클럭에 의해 주기적으로 수행되는 시간 구동 method (time-triggered method)들로 Spontaneous Method(SpM)이라고도 하며 시스템 응용 설계시 그 작동이 결정되는 Method 그룹이다. 작동 기간과 주기에 의해 수행

되며 각 주기적 작동 시기에 대한 오차 허용시간과 1회 작동 완료까지의 보장성 deadline이 주어진다. SpM의 이러한 시간 명세를 Autonomous Activation Condition(AAC)라 하며 다음과 같은 표현 방식을 갖는다.

```
“for t=from 10am to 10 : 50am every
    2sec start-during t ± 10msec
    finish-by t + 0.5sec”
```

두번째 그룹은 client로 부터의 message 수신 event에 의해 구동되는 메세지 구동 method(message triggered method)로 Service Method(SvM)으로도 불린다. message가 전달된 후 작동완료까지의 보장성 deadline이 주어진다.

- SvM과 SpM의 구동이 공유되는 데이터 부분에 대해 충돌될 때에는 SpM에 우선권을 준다. 이는 객체 데이터 공간을 segment로 나누고 각 segment에 대한 method 들의 access list를 선언함으로써 가능한데 이러한 선언은 Basic Concurrency Constraint(BCC)라 한다.
- 실시간 객체내의 각 자료구조에는 그 자료가 의미를 갖는 시간인 Maximun Validity Duration(MVD)가 주어진다.

RTO.k의 구성은 그림 1과 같다. 이상과 같은 RTO.k의 설계 개념은 실시간 시스템이 항상 RTO.k(SpM 과 SvM의 조합)의 network로 표현될 수 있음에서 출발한다. RTO.k의 설

계시 시간 보장의 개념을 구현하려면 RTO.k 모델의 수행과 시간 보장 커널 서비스를 제공하는 실시간 커널이 필요하다. UCI DREAM Lab.에서 개발된 DREAM(Distributed Ever-available Micro Computing) kernel은 이러한 RTO.k의 수행 엔진으로 개발되었고 현재 기능 보장 중에 있다. 그 특징을 요약하면 다음과 같다.

- RTO.k의 SpM, SvM에 대한 실시간 스케줄링을 제공한다.
- 분산 IPC와 클럭 동기화에 의한 분산 환경용 커널이다.
- 커널 서비스 시간의 예측을 위해 시간 점유의 우선 순위에 의한 5단계 계층으로 커널을 구성하였다.
- 커널 자신도 RTO.k로 설계되었다.

이상과 같은 RTO.k 모델은 PC를 바탕으로 한 분산환경에서 경성 실시간 시스템 개발용으로 주로 사용되고 있는데 다음은 이의 응용에 관한 예이다.

4. IVS 개발

IVS(Intelligent Vehicle System)는 필자들이 시험적으로 개발 중인 무인 자동차 시스템으로 lane sensor, 초음파 sensor, mark sensor 등에 의해 출발, 정지 및 감속을 자동으로 하는 소형 전기 자동차 시스템이다. IVS는 여러가지 형태의 주기적 탐지 및 비동기적 사건 처리등을 해야하는 전형적인 실시간 시스템으로 RTO.k의 SpM과 SvM들을 method로 갖는 몇개의 실시간 객체들로 모델링할 수 있다. 다음은 그 구성의 요약이다.

- Lane-sensor : 진행해야 할 lane을 주기적으로 sense 하는 SpM과 lane 정보에 대한 타 객체의 문의에 답하는 SvM으로 구성된다.
- Sonic-sensor : 장애물 탐지를 위해 주기적으로 세방향의 초음파 scan 입력을 수정하는 SpM과 이의 내용에 대한 타 객체의 문의에 응답하는 SvM으로 구성된다.
- Marker-sensor : 진행로 상의 정지점 등 특수 표식을 주기적으로 인식하는 SpM과

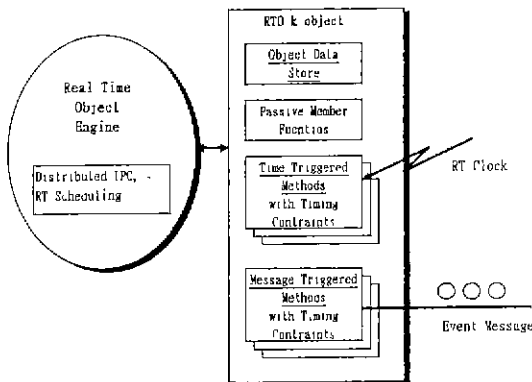


그림 1 RTO.k의 구성도

이의 결과에 대한 타 객체의 문의에 답하는 SvM으로 구성된다.

- Speed-controller : 주기적으로 자동차의 속도를 check하는 SpM과 외부 객체의 감속 명령을 수행하는 SvM으로 구성된다.
- Direction-controller : 주기적으로 진행 방향을 check하는 SpM과 외부 객체의 조향 명령에 해당 동작을 수행하는 SvM으로 구성된다.
- IVS : 각 sensor로부터 자료를 수집 이를 바탕으로 Speed-controller와 Direction-controller에 작동을 명령하는 SpM으로 구성된다.

위와 같은 실시간 객체에 의한 IVS의 구성은 다음장에 기술될 확장된 RTO.k 모델과 1차적으로 DREAM 커널을 이용, 구현의 완성단계에 있다.

5. 결 론

우리의 삶이, 그리고 우리가 하는 모든 작업이 시간의 흐름 속에 이루어지기 때문에 엄밀히 말해 실시간 시스템이 아닌 시스템은 없다. 다만 작업의 종료시간이나 자료의 유효시간 등에 대한 엄격성의 차이를 실시간 시스템과 비실시간 시스템을 구별할 뿐이다. 이러한 측면에서 본다면 실시간 시스템이 비실시간 시스템의 서브셋(subset)이 아니라 슈퍼셋(superset)이란 개념으로 접근하여야 한다. 즉, 지금까지의 비실시간 시스템(또는 비실시간 프로그래밍)은 시간이란 변수를 명시적으로 고려하지 않고, 논리적인 정확성만을 고려하였다. 같은 맥락에서 파스칼 언어를 개발한 니콜라우스 위스는 실시간 소프트웨어를 4차원 프로그래밍이라 하였다. 그는 순차(sequential) 프로그램이 2차원이던 병렬(parallel) 프로그램을 3차원, 그리고 실시간 프로그램을 4차원 프로그램이라 정의하였다. 2차원에서 3차원으로 가기 위한 많은 연구가 있었다. 멀티태스킹 기법, 병렬 알고리즘, 동기화 기법, 그리고 프로세스간의 통신기법들이 대표적인 예이며, 많은 기술들은 이미 실용화되었다. 이제

실시간 시스템 구축을 위하여는 4차원 프로그래밍 개념의 도입과 개발이 이를 모형화할 수 있는 방법의 연구가 필요하다. 특히 객체지향의 분석 및 설계의 개념에 근거한 모형화 방법의 개발이 필요하다. 실시간 객체 모형은 객체지향 방법의 여러 가지 장점들을 살리며 시스템의 실시간 요구사항을 명확히 표현할 수 있다. 앞으로 실시간 객체 모형을 이용한 검증방법과 도구의 개발, 그리고 모형으로부터의 자동(또는 반자동)구현에 관한 연구가 필요하다.

참고문헌

- [1] B. Berthomieu and M. Diaz, "Modeling and Verification of Time Dependent Systems Using Time Petri Nets", IEEE Trans. on SE, vol 17, no 3, pp. 259-273, Mar. 1991.
- [2] J. Coolahan and N. Roussopoulos, "A Timed Petri Net Methodology for Specifying Real-time System Timing Requirements", Workshop on Timed Petri Nets, Torino, Italy, July 1985.
- [3] H. Gomma, "Software Design Method for Real-Time Systems", Communications of the ACM, vol. 27, pp.938-949, Sept. 1984.
- [4] Y. Ishikawa and H. Tokuda, "Object-Oriented Real-Time Language Design : Constructs for Timing Constraints," OOPSLA '90.
- [5] F. Johanian and A. Mok, "A Graph-Theoretic Approach for Timing Analysis in Real-Time Logic", Real-Time Systems Symposium, Dec. 1986.
- [6] K. Kim and H. Kopetz, "A Real-Time Object Model RTO.k and an Experimental Investigation of Its Potentials," Proc. COMPSAC '94, Taipei, Nov. 1994.
- [7] K. Kim, "Realization of Autonomous Decentralized Computing with RTO.k Object Structuring Scheme and HU-DF Inter-Process-Group Communication Scheme," Proc. IEEE Computer Society's 2nd Int'l Symp. on Autonomous Decentralized

Systems, Apr. 1995, Phoenix, AZ, pp. 305-312.

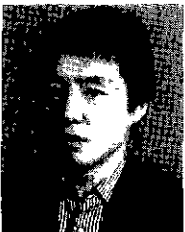
- [8] K. Kim, L. Bacellar, Y. Kim, C. Subbaraman, H. Yoon, Jungguk Kim and K. W. Rim, "A Timeliness-Guaranteed Kernel Model-DREAM Kernel and Implementation Techniques," RTCSA '95, Tokyo, Oct. '95.
- [9] J. Ostroff and W. Wonham, "Modelling, Specifying, and Verifying Real-Time Computer Embedded Systems", Real-Time Systems Symoisum, Dec. 1987.
- [10] J. Ostroff, "Verification fo Safety Critical Systems Using TTM/RTTL", REX Workshop on Real-Time, Theory and Practice, 1991.

김 문 회



1979.2 서울대학교 전기공학(학사)
 1981.2 서울대학교 전기공학(석사)
 1985.5 (미)Univ. of South Florida 전산학(석사)
 1991.5 (미)Univ. of California, Berkeley 전산학(박사)
 1991~현재 건국대학교 컴퓨터 공학과 부교수
 관심분야: 소프트웨어공학, 실시간분산처리시스템

김 정 국



1977.2 서울대학교 계산통계학(학사)
 1979.2 한국과학기술원 전산학(석사)
 1986.2 한국과학기술원 전산학(박사)
 1983~현재 한국외국어대학교 컴퓨터공학과 교수
 1994~95 (미)Univ. of California, Irvine, 교환교수
 관심분야: 운영체제, 소프트웨어공학

민 병 준



1983.2 연세대학교 전자공학(학사)
 1985.2 연세대학교 전자공학(석사)
 1991.12 (미)Univ. of California, Irvine 전기및컴퓨터공학(박사)
 1984~86 삼성전자 종합연구소
 1992~94 한국통신 연구개발원
 1994 감사원 전산담당관실
 1995~현재 시립인천대학교 전자계산학과 제직중
 관심분야: 실시간분산처리시스템, 결합허용

양 승 민



1978.2 서울대학교 전자공학(학사)
 1983.4 (미)Univ. of South Florida 전산학(석사)
 1986.12 (미)Univ. of South Florida 전산학(박사)
 1988~93 (미)Univ. of Texas at Arlington 조교수
 1992~현재 숭실대학교 컴퓨터 학부 부교수
 관심분야: 소프트웨어공학, 실시간분산처리시스템