論文 96-21-8-8

시간 제약 분석이 가능한 객체 지향 실시간 시스템 모델링

正會員 김 영 란*, 권 영 희**, 홍 성 백***, 박 용 문***, 구 연 설****

Object-Oriented Real-Time System Modeling Considering Predictable Timing Constraints

Young-Ran Kim*, Young-Hee Kweon**, Sung-Back Hong***,
Yong-Mun Park***, Yeon-Seol Koo**** *Regular Members*

요 약

객체 지향 개발 방법을 이용하여 실시간 시스템을 개발할 경우에는 반드시 시간제약 문제를 고려해야 한다. 본 논문에서는 사건의 발생에 따른 시스템의 상태 변이를 상태도로 나타내는 동적 모델에 대한 실시간 규격화로서, OMT와 SDL을 이용한 객체 지향 실시간 시스템 모델링 방법을 제안하였다. 현재 가장 널리 활용되고 있는 OMT 방법론의 분석 과정에서 생성되는 동적 모델에 실시간 시스템의 시간제약을 반영함으로써 예상시간 도표를 작성하였고, 실시간 예측이 가능하도록 순차 수행, 반복 수행, 병렬 수행에 대한 최대 및 최소 실시간 예측 공식을 제시하였다. 또한, 제안된 방법론을 이용하여 ATM 시스템의 최대 수행 시간을 예측해 보았고 객체 상호 작용 그래프와 SDL의 구문적 표현을 이용하여 ATM 시스템의 사용자 인터페이스에 대한 기능 명세서를 기술하였다.

ABSTRACT

In the case of developing the real-time system using object-oriented method, the problem of the timing constraints is certainly considered. We propose the method of modeling the object-oriented real-time system using the OMT methodology and the SDL. And we also present the predictable time table that reflects the constraints of real-time system into dynamic model of OMTs and the predictable time formula of the sequence, repeat, and parallel routine.

The proposed method is applied to the estimate of the maximum process time of the ATMs(Automatic teller machines) and is used to specifying the functional specification for the user interface of the ATMs using the SDL syntax and the object interaction graph.

*충청전문대학 전임강사
**충북대학교 대학원 전자계산학과 박사과정
***한국전자통신연구소 교환기술연구단 선임연구원
****충북대학교 컴퓨터과학과 교수
論文番號:96119-0410
接受日字:1996年 4月 10日

Ⅰ. 서 론

실시간 시스템은 비결정적인 요소들을 포함하고 있을 뿐만아니라, 시스템의 상태나 행위에 관한 명세서(specification)에 시간 제약(timing constrints) 개념이 포함되어 있기 때문에 개발하기가 매우 어렵다[1][2][8]. 실시간 시스템에 대한 요구사항을 정형적으로 표현하기 위한 명세서 언어에 관한 기존의 연구는 매우 활발히 진행되었고, 이러한 언어로는 SDL[11], PAISLey[3], RTRL[4] 등이 있다. SDL과 RTRL은 시스템의 특징을 유한 상태 기계로 모델화하고 PAISLey는 프로세서의 상태가 값의 집합이 될 수 있는 상호 비동기 프로세서로 모델화함으로써 실시간 시스템에 대한 좀 더 일반적인 관점을 명세화한다.

오늘날 통신 수요의 증대와 통신 환경의 발전 추세에 따라 통신 소프트웨어의 개발 및 유지보수에 소요되는 비용중에서 소프트웨어가 차지하는 비용이 상대적으로 높아진다. 대규모적이면서 복잡한 실시간 소프트웨어 시스템을 개발하기 위해 객체 지향 방법론을 적용하고자 하는 시도가 진행중이다[5][6]. 왜냐하면, 실세계의 모든 개념적 개체(entity)를 단일 개념 개체로 모델링함으로써, 데이타 추상화 개념을 효과적으로 지원하기 때문에 효율적인 유지보수 작업이 가능하고, 객체와 클래스 개념에 의해서 이미 개발된 소프트웨어의 재사용성을 향상시킴으로써 소프트웨어의 개발기간을 단축시킬수 있는 장점을 갖고 있기 때문이다. 이러한 시도는 객체 지향 방법론이 실세계의 개념을 반영하는 측면에서 볼 때, 복잡한 소프트웨어의 분석 및 설계를 효율적으로 수행할 수 있는 반면에 기존의 객체 지향 모델링 방법으로 실시간 시스템의 중요한 특징인 시간 제약 문제, 병행 처리 프로세서의 식별, 프로세스간의 통신 및 동기화등을 반영하지 못하는 문제점을 갖고 있다[6][7][8][9].

본 논문에서는 가장 널리 활용되고 있는 객체 지향 방법론인 OMT방법론의 동적 모델에 시간 제약 개념을 반영한 정형화된 객체 지향 실시간 시스템 모델 방법을 제안하고, 설계 단계에서 병행 프로세서의 식별, 프로세스간의 통신 및 동기화를 반영할 수 있는 OMT/SDL 시스템 명세서의 구성 방안을 제시한다. 제안된 방법을 실시간 시스템 개발에 적용함으로써 기대되는 효과로는 첫째, 분석 단계에 시간 제약 개념을 명세화함으로써 실시간 시스템에 대한 명확한 분석이 가능하게 되고, 둘째, 설계단계에 SDL을 이용하여 정형적인 시스템의 기능을 명세화함으로써 검증성(verifiability)을 향상시키고, 세째, 정형화된 시스템 명세서를 기술함으로써 유지보수 작업이 용이해 진다. 본 논문의 구성은 2장에서 기존의 객체 지향 방법론과 실시간 소프트웨어 명세언어를 분석하여 기술하였고, 3장에서 실시간 시스템에서의 시간 제약 표현 방법과 시간 제약 개념이 반영된 객체 지향 동적 모델에 대한 정형화된 논리구조의 구성방안을 제안하였다. 4장에서는 제안된 OMT/SDL 방법론을 ATM 시스템에 적용한 시스템 기능 명세서를 작성하였고, 5장에서 결론 및 앞으로의 연구 방향을 제시하였다.

Ⅱ. 관련연구

1. 객체 지향 방법론 분석

기존의 객체 지향 소프트웨어 개발 방법론으로는 Coad & Yourdon(OOA) 방법론, Shlaser & Mellor 방법론, Booch 방법론, 그리고 OMT 방법론등이 있다. Coad & Yourdon 방법론[10]은 객체 모델의 주요소인 객체들의 분류화(classification)구조, 상속성을 표현하는 객체 구조 객체들 사이의 서비스 이용을 표현하는 객체통신으로 구성된다. 이 방법론은 구조적 방법론에 객체 지향 개념을 반영한 방법론으로서, 실세계의 복잡한 성질을 표현하는데 어려움이 있다. Shlaer & Meller 방법론[11]은 객체 및 속성 그리고 관계를 묘사하는 정보 모델로 이루어지며, 각 모델은 정보 구조 다이어그램, 상태 전이 다이어그램, 데이타 흐름 다이어그램으로 표현한다. 이 방법론은 관계 데이타베이스의 테이블과 키에 의존하기 때문에 객체 지향 상속의 개념이 없다. Booch 방법론[12]은 실제 시스템을 자료 흐름도로 모형화 한 후, 6개의 다이어그램을 이용하여 객체로 분해하고 객체들간의 인터페이스를 표현한다. 이 방법론은 클래스와 객체 식별, 클래스와 객체의 의미 식별, 클래스와 객체의 관계 식별, 그리고 구현 4단계로 구성된다.

OMT(object modeling technique) 방법론[13]은 소프트웨어 개발 전 단계에 객체지향 개념을 일관되게 지원하는 방법으로서, 객체들의 연관성을 강조한 조직적인 모델링 방법론을 이용하여 시스템의 기능을 상세

히 나타낸다. 이 방법론은 복잡한 시스템의 특성에 대한 객체 모델(object model), 동적 모델(dynamin model), 기능적 모델(functional model)의 3가지 모델 기법을 이용하여 소프트웨어 시스템에 대한 분석 모델을 형성한다. 객체 모델은 객체, 클래스, 링크, 결합, 상속성, 집합관계 등의 중요한 개념을 기반으로 하여 객체나 클래스들의 정적인 자료구조와 그들 사이의 관계를 객체 다이어그램을 이용하여 표현한다. 이러한 객체모델은 동적 모델과 기능적 모델을 구성하는데 기반이 된다. 동적모델은 상태 다이어그램을 이용해서 시간 경과에 따른 객체들의 변화와 객체들간의 관계를 표현하는 모델로서, 이 상태도의 중요한 개념은 사건(event)과 상태(state)이다. 기능적모델은 자료흐름도를 이용해서 시스템 내부에서의 자료값의 변형을 표현하는 것으로서, 연산이 어떻게 혹은 언제 계산되는지는 기술하기 않고 계산 결과만을 기술한다.

객체 지향 방법론중에서 현재 널리 활용되고 있는 방법론은 Booch 방법론과 OMT 방법론이다. Booch 방법론은 시스템의 동적 시맨틱스를 타이밍 다이어그램으로 표현하기 때문에 실시간 시스템 분석 및 설계에 활용될 수 있는 반면에, OMT 방법론은 시간 경과에 따른 객체들의 변화와 객체들간의 관계만을 동적 모델로 표현함으로써 실시간 시스템의 특징중에 하나인 시간 제약 개념을 반영하지 못하는 문제점이 있다. 따라서 본 논문에서는 ITU-T에서 고려하고 있는 OMT 방법론의 객체 지향 동적 모델에 시간 제약 개념을 반영한 정형화된 논리구조를 제안한다.

2. 객체 지향 SDL

원격통신 시스템은 매우 복잡하고 다양한 면을 내포하기 때문에, 어떠한 경우라도 대처할 수 있는 언어를 필요로 한다. ITU-T는 전기통신 관리 체제와 관리자에게 원격통신(telecommunication) 시스템에 대한 간단하고 명확한 의사교환 공용 언어를 제공하기 위해 SDL이란 명세기술언어를 권고하였다[14][15]. SDL (functional Specification and Description Language)은 원격 통신에 제한되지 않고 이산적인 행위의 자극/응답 형태를 가진 실시간 시스템에 일반적으로 유용하다.

객체 지향 SDL[16]은 객체 지향, 광역 파라메타, 라이브러리 패키지 개념을 도입하고, 표준 SDL 인스턴스 정의를 타입 정의로 확장함으로써 시스템의 명세서 기술만이 목적이 아니라 재사용 가능한 명세서를 목표로 한다. 이 명세서 언어는 객체 지향 개념의 특수화(specialization)와 관련하여 수퍼타입의 transition이나 local 타입(블럭, 프로세스, 프로시듀어)이 재정의 될 수 있도록 지원한다. 서브 타입에서 재정의되는 타입과 transition을 각기 virtual type, virtual transition이라 하고, 재정의된 타입은 virtual이면서 서브타입에서 재정의 할 수 있다. 객체 지향 SDL은 다양한 타입을 제공하며, 그중에 하나가 TIMER타입이다. TIMER는 변수와 유사하게 정의되고 시간 값(time value)을 갖는다. 타입 TIMER의 행위를 유한 상태 기계로 표현하면 그림 1과 같다. TIMER T를 정의하고 T변수에 시간 값을 초기화 하면, 현재 시간 q에 시간 값이 반영됨과 동시에 TIMER의 행위가 활성화(active)되고, 지정된 시간이 경과되면 변수 T는 reset 된다.

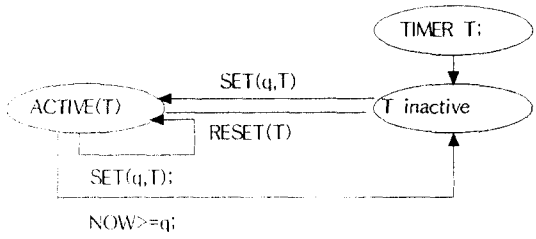


그림 1. TIMER 작동에 대한 유한 상태 기계
Fig. 1 Finite state machine of TIMER behavior

Ⅲ. 객체 지향 실시간 시스템 모델링

본 논문에서는 OMT 방법론을 이용하여 실시간 시스템을 분석하는 과정에, 시간제약 개념을 동적 모델에 적용하기 위해 시스템의 행위에 대한 시간 제약(timing constraints)을 분류하였고, 시스템의 정확한 동작을 예측할 수 있도록 예상 시간 도표를 작성하였다. 또한 작성된 도표로 부터 시스템의 행위에 대한 실시간 예측 공식을 유도하였다.

1. 시간제약 표현

동적모델은 사건의 발생에 따른 상태변이를 상태

도로 작성하여 시스템의 행위를 분석하는 과정이다. 시스템의 행위들은 주어진 시간제약 범위내에서 수행되어야 하고, 시간제약은 시간변이에 따른 일시적 제약을 주는 것으로서, 시스템의 응답시간에 한계를 부여하는 수행시의 제약과 시스템에 자극을 가했을 때 발생하는 행위의 제약으로 구분하였고, 이는 각기 최대/최소 시간제약으로 분류하였다. 왜냐하면, 동적 모델을 구성하는데 있어서의 행위는 상태내에서 이루어지는 활동(activity)과 상태간의 전이를 야기시키는 사건에 의한 행위(action)로 크게 구분된다. 상태내에서의 활동은 행위적 요구(behavioural requirement)와 수행적 요구(performance requirement)가 모두 필요하다. 따라서 활동에서의 수행시간을 나타내는 시간제약을 최대 시간제약과 최소 시간제약으로 분류하며, 각각은 MAX_t와 MIN_t로 표시한다. 반면에, 사건의 발생에 의한 행위는 하드웨어 자체에서 수행되는 시간으로 수행시간은 일정하나, 한 상태로 여러개의 비동기 사건(asynchronous event)이 입력된 경우에 각 행위는 먼저 들어온 행위가 수행을 끝마칠 때까지 기다려야 한다. 이러한 행위의 대기시간은 W_t로 표시한다. 표 1은 동적모델에 반영되는 시간제약을 분류한 것이고, 그림 2는 상태 다이어그램의 기본 구조로 부터 시간제한을 구분한 것이다.

표 1. 동적 모델에서의 시간 제한 분류
Table 1. Classifying of timing constraint in dynamic model

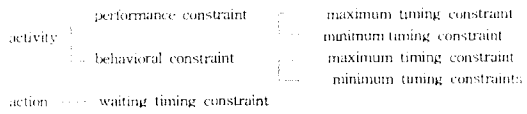


그림 2에서 a는 상태내에서의 활동이 수행되는 시간으로 경과된 시간에 따라 최대/최소 시간으로 구분되고, b는 사건의 발생에 따른 행위로서, 다음 상태의 활동이 수행되기 전까지의 대기시간을 나타낸 것이다.

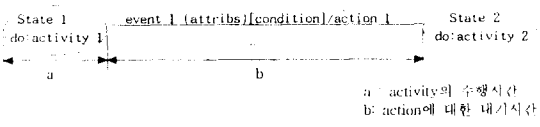


그림 2. 상태도의 시간제한
Fig. 2 Timing constraint of state diagram

2. 동적모델에 대한 정형화된 논리구조 구성

동적 모델링에서의 사건 추적 다이어그램으로 부터 구성된 상태도를 통해, 상태변화에 따른 수행시간을 계산하기 위한 예상 시간 도표를 작성한다. 예상 시간 도표와 정형화된 논리구조를 기반으로 하여 시스템내의 부분 상태 변화에 따른 시간의 경과나 전체 시스템의 상태 변화에 따른 시간의 경과를 예측하고, 이 때 발생하는 오류는 동적모델에 다시 반영하여 수정할 수 있다.

동적 모델에서는 activity를 기본활동(primitive activity)과 복합활동(composite activity)으로 구분한다. 기본활동은 제한된 양의 시스템 자원(CPU 혹은 communication bandwidth)을 소모하는 동작으로서 한 시점에서의 활동을 뜻하며, 복합활동은 다른 활동들의 부분적 순서로서 부분활동의 우선순위를 제한하는 동작으로 다음과 같이 나타낼 수 있다. 복합활동의 기본 유형을 정형화 하기 위한 상태도의 구조는 그림 3과 같다. 상태간의 순차적 수행은 S1;S2로 명세화하고, 한 상태에서의 연속적인 반복수행은 그 횟수를 기술하는 Sn로 명세화하며, 사건의 발생에 의한 제어분할로 두개의 상태가 동시에 수행되는 경우는 S1‖S2로 명세화한다.

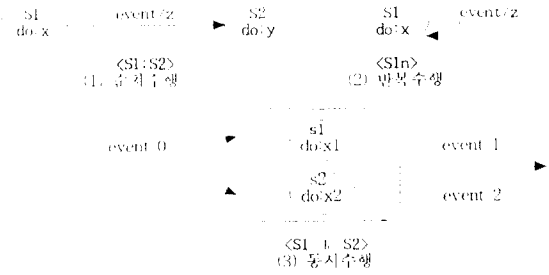


그림 3. 상태도의 기본 구조
Fig. 3 Framework of state diagram

상태도에 대한 논리구조를 정형화함으로써 실시간 예측을 위한 공식은 다음과 같이 유추된다. 순차수행은 그림 3의 (1)과 같이 연속된 수행에 대한 처리로서, 식 (1-1)과 (1-2)를 통해 수행 시간에 대한 실시간 예측이 가능하다. 즉, 상태 S1에서 S2로 변환되는 최대 시간은 상태 S1과 S2에 자극을 가했을때 발생하는 행

위의 최대 제약 시간인 MAX_t(X) 및 MAX_t(Y)과 선행하는 행수가 수행을 끝마칠 때까지 기다려야 하는 행위의 대기 시간 w_t(Z)의 합이다.

$$MAX\_t(S1;S2) = MAX\_t(X) + MAX\_t(Y) + W\_t(Z) \quad (1\text{-}1)$$

$$MIN\_t(S1;S2) = MIN\_t(X) + MIN\_t(Y) + W\_t(Z) \quad (1\text{-}2)$$

위의 식은 일정한 주기를 가지는 상태로서, 한 상태의 도착시간과 다음 상태의 도착시간과의 간격이 주기가 되고, 상태의 종료시한은 주가기 끝나는 시간과 일치하는 경우이다. 만약, 일정한 주기를 가지는 상태로서, 상태의 처리 완료시간이 종료시간을 넘기는 경우에는 상태 X와 상태 Y의 최대 및 최소 교차시간을 감산하거나, 다음 주기에 단순한 약식처리 등의 방법으로 융통성있게 대처하면 된다. 일정한 주기를 갖지 않고 도착시간이 불규칙적인 상태들의 순차수행은, 상태의 종료시한이 주기가 끝나는 시간보다 길어짐으로 인해 다음 상태의 주기와 일부 중첩(overlap)되거나, 상태의 종료시간이 주기가 끝나는 시간보다 짧음으로 인해 다음 상태의 대기시간이 발생하기도 한다. 비주기적인 상태 스케줄링인 경우의 실시간 예측은, 상태 0에서 상태 n 사이에 일어나는 중첩사건의 발생 횟수를 랜덤한 것으로 간주하여, 확률변수 N(o, n)으로 나타낼 수 있다. 따라서, 시간이 흐름에 따라 발생하는 사건의 횟수는 확률변수들의 모임으로서 $\{N_{(t)} : t \geq 0\}$이고 또한, 사건은 서로 독립적이므로 포아송 확률과정으로서, 단위 시간당 사건의 발생률은 다음과 같다.

$$\lim_{h \to \infty} \frac{1}{h} P(N_{(h)} = 1) = \lambda$$

따라서, 임의의 i번째 상태에서 발생되는 중첩시간을 $(t_i + \triangle_i + 1) - (t_i + \triangle_i) = \alpha_i$라 가정하면, 비주기적인 상태에 대한 실시간 예측공식은 상태 S_i와 $S_i + 1$행위의 최대 제약시간과 대기 상태 주기의 대기시간의 합에서, 랜덤하게 발생하는 포아송 분포의 중첩사건에 대한 중첩시간을 감산한 결과이다.

$$MAX\_t(S_1;S_2;\ldots;S_n) = \sum_{i=1}^{n} (Max(X_i) + Max(Y_i) + W\_t(Z_i)) - \sum_{t=\zeta_t} (P(N_{(t)} - N_{(t-1)}, \lambda) * \alpha_t) \quad (1\text{-}3)$$

$$MIN\_t(S_1;S_2;\ldots;S_n) = \sum_{i=1}^{n} (Min(X_i) + Min(Y_i) + W\_t(Z_i)) - \sum_{t=\zeta_t} (P(N_{(t)} - N_{(t-1)}, \lambda) * \alpha_t) \quad (1\text{-}4)$$

Where, For any $0 \leq t_0 < t_1 < \cdots < t_n$.
rnadom variable $\xi_{t_1} - \xi_{t_0}$, $\xi_{t_2} - \xi_{t_1}$, ...은 독립적이고 포아송 확률분포임 *

반복수행은 그림 3의 (2)와 같이 반복된 수행에 대한 처리이며 수행시간은 반복횟수 n의 크기와 상관관계가 있다. 수행시간에 대한 실시간 예측 공식은 식 (2-1)과 (2-2)와 같다.

$$MAX\_t(S1n) = \sum_{k=1}^{n} (MAX\_t(S1k) + W\_t(Z) \quad (2\text{-}1)$$

$$MIN\_t(S1n) = \sum_{k=1}^{n} (MIN\_t(S1k) + W\_t(Z) \quad (2\text{-}2)$$

마지막으로 그림 3의 (3)과 같은 동시수행은 사건에 의해 제어가 분할된 후, x1과 x2의 활동을 동시에 수행하고 다시 제어가 동기화됨을 나타내며 수행시간에 대한 시간예측 공식은 식 (3-1)과 (3-2)와 같다.

$$MAX\_t(S1 \| S2) = MAX\_t(\max(MAX\_t(S1), MAX\_t(S2))) \quad (3\text{-}1)$$

$$MIN\_t(S1 \| S2) = MIN\_t(\min(MIN\_t(S1), MIN\_t(S2))) \quad (3\text{-}2)$$

Ⅳ. ATM 시스템의 분석 및 설계

1. ATM 시스템의 구성

본 논문에서는 자동 출납기 시스템에 시간제한을

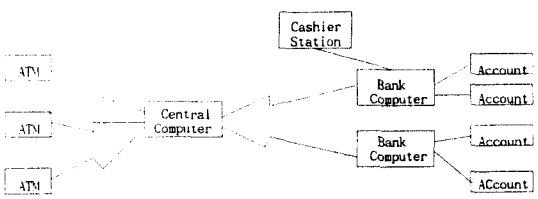


그림 4. ATM network 구성도
Fig. 4 Diagram of ATM network

추가한 실시간 자동 출납기 시스템에 대한 분석 및 설계를 수행하였다. 그림 4는 자동출납기의 네트워크 구성도이다.

2. 동적 모델에 대한 시스템 분석

동적 모델은 시스템과 객체들 간의 신호처리를 기술하기 위해 사건추적 도표(event trace diagram)를 이용한다. 이때 각 순서는 하나 혹은 몇개의 기능과 연관되어 있으며, 가능한 상호작용에 대한 시나리오를 기술한다. 시나리오를 기반으로 하여 동적 모델에 대한 상태도를 작성하고, 상태에 대한 시간제한을 표현하기 위해 예상시간 도표를 작성함으로써 시스템에 대한 정확한 처리를 파악 할 수 있다. 다음은 동적 모델링 과정을 통한 분석과정과 동적모델에 대한 요구명세서를 작성하는 과정이다.

2.1 시나리오 구성

시나리오란 시스템이 특정 기능을 수행하는 동안 일어날 수 있는 모든 경우의 사건들에 대한 흐름으로 정의할 수 있다. 다음은 정보모델중 고객과 자동출납기 사이에 발생할 수 있는 시나리오이다.

> 자동출납기가 고객에게 카드를 입력할 것을 요청하면 고객이 카드를 입력한다. 자동출납기가 카드를 받아들이고 카드 번호를 읽은 후, 고객에게 암호를 요청한다. 고객이 암호를 입력하면 자동출납기는 중앙컴퓨터에 카드번호와 암호를 보내어 사용 가능한 카드임을 검사한다. 중앙 컴퓨터는 자동출납기에게 맞는다는 신호를 보낸다. 자동출납기는 고객에게 거래의 종류를 선택할 것을 요청하고, 고객이 인출을 선택하면 자동출납기는 중앙컴퓨터에게 거래를 처리할 것을 요청한다. 중앙컴퓨터는 처리를 수행하여 계좌의 잔액을 갱신하고 자동출납기에 처리가 성공적으로 수행되었음을 알리고 고객에게 현금을 가져갈 것을 요청한다. 고객이 현금을 가져가면 자동출납기는 영수증을 출력하고 카드를 배출한 후, 고객에게 그것들을 가져갈 것을 요청한다. 고객이 영수증과 카드를 가져가면 자동출납기는 다음 고객에게 카드를 입력할 것을 요청한다.

2.2 사건 추출 및 상태도 작성

사건은 상태 변화를 일으키게 하는 원인이다. 그림 5는 앞에서 작성한 시나리오를 바탕으로 사건을 추적한 다이어그램이고, 사건 추적에 세 가지에 클래스(고객, 자동 출납기, 중앙 컴퓨터)만이 추출된 이유는, 이들 세 가지 클래스만이 서로 사건을 교환하며 상태 변화가 이루어지기 때문이다. 시스템의 동작은 사용자가 ATM으로 카드를 삽입함으로써 시작되며, 이때 발생하는 사건들은 그림 5와 같은 순서에 따라 사용자와 ATM, 그리고 중앙 컴퓨터와 통신하며, 최종적으로 거래가 종결되면 ATM은 사용자에게 현금과 영수증, 그리고 카드를 돌려준다.

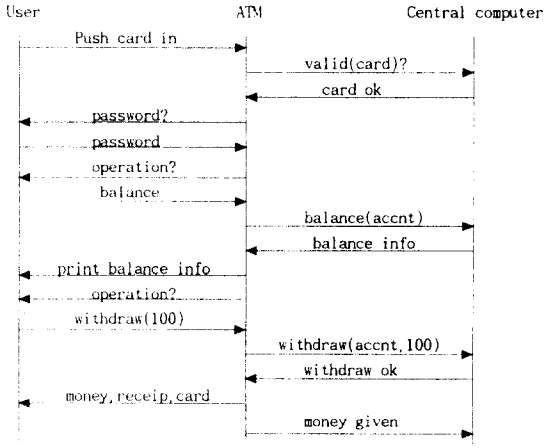


그림 5. ATM의 사건 추적
Fig. 5 Event trace of ATM

그림 6은 그림 5에서의 사건 추적을 바탕으로 작성한 자동 출납기의 상태도이다.

2.3 예상시간 도표 작성 및 실시간 예측

시간 제약은 일시적 제한을 주는 것으로 시스템의 응답시간에 한계를 부여하는 수행시의 제한과 시스템에 자극을 가했을 때 발생하는 행위제한으로 구분하였고 이는 다시 최대/최소 시간제한과 수행기간으로 분류하였다. 본 논문에서는 실시간 예측을 위해 동적 모델링 과정에서 작성된 상태도로 부터 유추한 activity와 action에 대한 시간제약의 예상시간 도표를 그림 7과 같이 구성한다.

실시간 시스템의 경우 최악의 조건(worst case)에 대해 행위가 성립하면 시스템은 안정적이라고 할 수 있

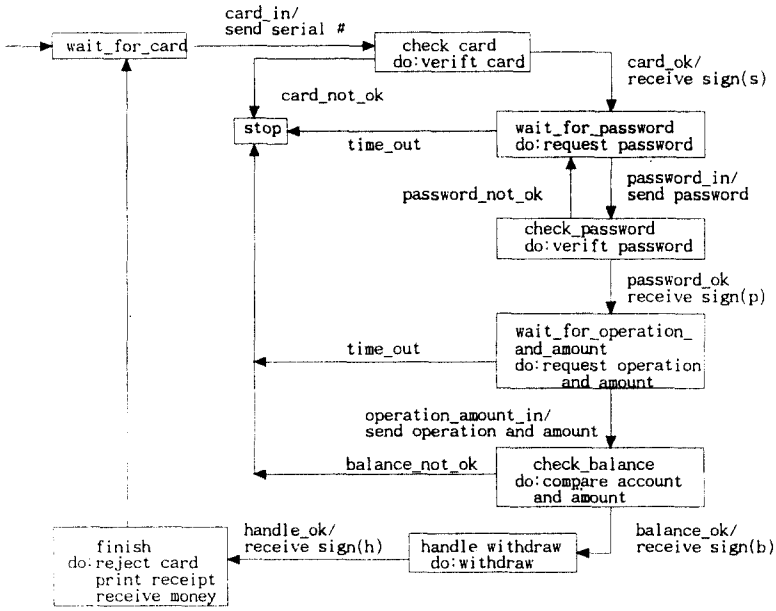


그림 6. ATM의 상태도
Fig. 6 State diagram of ATM

(단위 : 초)

activity	MAX_t	MIN_t	action	W_t
verify card(v_c)	1	0.5	senc serial #(s_s)	0.5
request password(r_p)	5	0.5	receive sign(s)	0.1
verity password(v_p)	1	0.5	send password(s_p)	0.5
request operation and amount(r_o_a)	5	0.5	receive sing(p)	0.1
compare account and amount(c_a_a)	1	0.5	send operation and amount(s_o_a)	0.5
withdraw(w)	1	0.5	receive sign(b)	0.1
eject card, print receipt and receive money(e_p_m)	3	2	receive sign(h)	0.1

그림 7. ATM의 예상 시간 도표
Fig. 7 Predictable time table of ATM

으므로, 여기에서는 최대시간에 대한 수행시간을 예측하였다. 시스템이 오류없이 안정적으로 동작할 경우 소요되는 최대시간을 예측하면 다음과 같다.

$$MAX\_t(w\_f\_c;e\_p\_m) = MAX\_t(v\_c) + MAX\_t(r\_p) + MAX\_t(r\_o\_a) + MAX\_t(c\_a\_a) + MAX\_t(w) + MAX\_t(e\_p\_m)$$

+W_t(s_s)+W_t(s)+W_t(s_p)+W_t(p)+W_t(s_o_a)
+W_t(b)+W_t(h)
=1+5+1+5+1+1+1+3
+0.5+0.1+0.5+0.1+0.5+0.1+0.1
=19.9(초)

3. ATM 시스템 설계 및 기능 명세서

소프트웨어의 설계 단계는 개발 과정의 분석단계와 구현단계 사이에 위치한다. 프로그램이 어떠한 일을 해야 하는지를 기술한 기능적인 명세서(functional specification)가 설계 단계의 입력자료가 된다. 설계단계의 목표는 시스템에 구현될 소프트웨어가 기능 명세서를 만족할 수 있도록 계획서를 작성하는 것이다. 본 논문에서는 시스템 설계를 위해 객체들간의 상호작용을 나타내는 객체 상호작용 그래프를 이용한다. 왜냐하면, 분석 단계의 객체모델을 그대로 설계단계에서 받아들일 수 없으므로 주요 기능을 모듈 단위로 나눔으로써 효율적인 해결방안을 제공하기 때문이다. 상세 설계를 위해서 SDL을 이용하여 ATM 시스템의 기능적 명세서를 작성하였다. 시스템 설계 단계에서는 객체들이 원하는 행위에 따라 통합하는가를 기술하기 위해 객체 상호작용 그래프를 작성한다. 먼저, 자동지급기의 객체들을 모듈화하기 위해 각각 사용자 인터페이스와 제어부분으로 나눌수 있다. 이때 이러한 객체들을 각각 ATM_ui와 ATM_control이라고 하자. 자동 지급기의 각 연산들은 객체 상호작용 그래프와 일치하도록 기술되어야 한다. 객체들과 그들간의 인터페이스를 나타내는 함수들로부터 균형질의(Balancy Query)를 위한 자동 지급기의 객체 상호작용 그래프를 기술할 수 있다. 그림 8은 자동지급기의 균형질의에 대한 객체 상호작용 그래프이다.

객체 상호작용 그래프는 SDL을 기반으로 하여 구현하기 위한 지침으로 사용된다. 객체들은 SDL 프로세스들로 구현되며 그외 다른 객체들은 그들의 기능적 인터페이스를 제공함으로써 구현된다. 종종 관련된 객체의 집합을 하나의 SDL 프로세서로 그룹화 하는 것이 효과적인 경우도 있다. 다음은 사용자 인터페이스 ATM_ui에 대한 기능 명세서이다. 여기서 하위 객체들(즉, keypad, printer, card_reader등)은 SDL 프로시져 호출로써 나타낸다.

```
PROCESS ATM_ui;
  FPARcontrol pid;/* the control process of the ATM*/
  CONSTANT card_poll=3*100;
    password_poll=30*100;
  TIMER card_timer, password_timer;
  DCL card_data          card_data_t,
      successful         boolean,
      password           password_t,
      transaction        transaction_t.
      amount             integer,
      account_info       account_info_t;
STATE;
  CALLask_for_card_form_user();
  SET(NOW +card_poll, card_timer);
  NEXTSTATE wait_for_card;
STATE wait_forcard;/* waiting for customer to insert card*/
INPUT card_timer;
```

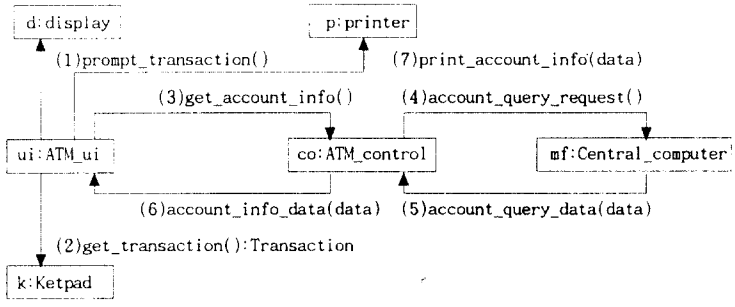


그림 8. 균형질의에 대한 객체 상호작용 그래프
Fig. 8 Object interaction grapgh for balancy query

```
    CALL slot_check(card_data, successful);
    DECISION successful;
      (true) :OUTPUT card_in(card_data) TO control;
              NEXTSTATE check_card;
      (false) :SET(NOW+card_poll, card_timer);
              NEXTSTATE-;
    ENDDECISION;
ENDSTATE wait_for_card;
```

V. 평 가

본 논문에서는 실시간 시스템 개발시 고려해야 할 여러 문제점들을 해결하기 위한 객체 지향 실시간 시스템 개발 방법으로 OMT/SDL 개발방법을 제안하였다. 제안된 방법을 통하여 실시간 시스템개발을 위한 자료흐름 중심의 개발방법들과 기존의 객체지향 개발방법들에서 고려하지 않았던 실시간 문제들을 해결할 수 있다. 표 2는 실시간 문제 해결을 위한 기존의 실시간 시스템 개발 방법들과 본 논문에서 제시한 OMT/SDL 방법간의 비교이다. 여기에서 알 수 있듯이 OMT/SDL 개발 방법은 기존의 실시간 시스템 개발 방법에서 고려하지 않았던 실시간 시스템 개발시의 문제점 즉, 시간 제약 및 시간예측과 연산의 우선순위등을 해결할 수 있음을 알 수 있다.

VI. 결 론

실시간 시스템을 개발하기 위해서는 시간제약 사항을 비롯한 여러가지 실시간 특성을 고려해야한다. 따라서 본 논문에서는 객체 지향 실시간 시스템 개발 방법으로 분석 단계에서 시간제약을 고려한 개선된 OMT방법을 제안하였고, 설계단계에서는 SDL을 이용해 기능적 명세서를 작성하는 OMT/SDL 방법론을 제안하였다. 이 방법은 분석단계에서 ITU-T에서 권고하는 OMT 방법의 3가지 모델 즉, 객체 모델, 동적 모델, 기능적 모델등을 이용하여 실시간 시스템을 분석하는 방법으로서 특히, 동적 모델링 과정에 실시간 시스템의 특성인 시간 제약 사항을 추가하기 위해서 동적모델을 표현한 상태도에서의 활동(activity)과 행위(action)에 각각 시간제약을 두어 이를 예상시간 도표(predictable timing table)에 표시하였다. 또한, 이 도표를 실시간 예측 공식에 이용함으로써 시스템의 수행에 따른 실시간을 예측할 수 있었다.

객체 설계과정은 시스템 설계와 세부적 설계로 구성되며, 첫번째 단계인 시스템 설계단계에서는 분석과정에 생성된 객체 모델을 재검토함으로써 시스템과 시스템 환경(enviornment)간의 객체에 대한 서비스(service/operation)와 객체간의 상호작용 순서(interaction sequence)를 조사하였고, 이때 요구된 서비스에 대한 시스템내에서의 모든 통신(communication)을 객체 상호작용 그래프(Object Interaction Graph)를 통하여 표현하였다. 두번째 단계인 세부적 설계단계에서는 SDL을 이용하여 실시간 시스템의 기능적 명세서(functional specification)를 작성하였다. 객체 지향 실시간 시스템 개발을 위해 제안된 방법을 적용함으로써 객체지향 분석단계에서의 문제점이었던 시간 제한 문제를 해결하였고, 설계단계에서는 SDL을 이용한 정형적인 시스템의 기능을 명세화함으로써 검증

표 2. 실시간 시스템 개발 방법론 간의 비교
Table 2. Comparison of real-time system development methodologies

	MASCOT	Ward & Mellor	Gomaa의 DARTS	Shlaser & Mellor	Rmbaugh의 OMT	Booch	OMT/SDL
병렬처리 프로세스	▲	●	●	●	●	●	●
병렬처리 프로세스간의 통신과 동기화	▲	●	●	●	●	●	●
프로세스 스케쥴링	▲	●	●	▲	▲	▲	▲
시간예측 및 시간제약	□	□	□	▲	□	▲	●
예외사항처리	□	□	□	□	□	□	□
소프트웨어 재사용	□	▲	▲	●	●	●	●
소프트웨어 유지보수	□	□	▲	●	●	●	●

□:고려하지 않음, ▲:완전하지 못함, ●:해결

성(verifiability)을 확대할 수 있다. 또한 정형화된 명세화를 통하여 유지보수작업이 용이해지는 효과가 기대된다.

앞으로, 객체지향 실시간 시스템을 개발하는데 있어서 고려되어야 할 시간제약 이외의 많은 문제들을 해결하기 위한 더 체계화된 방법을 연구해야 하며, 더 나아가서는 이 방법들을 CASE Tool화하여 실시간 시스템을 개발함에 있어, 보다 적은 비용으로 생산성 향상을 이루어야 할 것이다.

참 고 문 헌

1. Kwel Jay Lin, "Real-Time Realities," IEEE Software, vol.9, no.5, pp.13-15, september 1992.
2. Philip Laplante, "Real-Time System Design and Analysis," IEEE Press, 1992.
3. P. Zave, "An Operational Approach to Requirements Specifications for Embedded Systems." IEEE Trans. SoftwareEng., Vol. SE-8, pp.250-269, May 1982.
4. B. Taylor, "A Method for Expressing the Functional Requirements of ReaL-Time Systems," in Proc, 9th IFAC/IFIP conf. Real-Time Programming. New York : Pergamon, 1980. pp.111-120.
5. Thomas E. Bihari, "Object-Oriented Real-Time Systems : Concepts and Examples." IEEE Computer vol.25, no.12, pp.25-32, December 1992.
6. Paul T. Ward, "Transformation Schema : An Extension of the Data Flow Diagram to Represent Control and Timing." IEEE Trans. Software Eng., vol.SE-12, no.2, pp.198-210, Feb. 1986.
7. B. Dasarathy, "Timing Constraints of Real-Time Systems : Constructs for Expressing Them, Methods of validating them," IEEE Trans. Software Eng., vol.SE-11, pp.80-86, January 1985.
8. Farnam Jahaniam & Aloysius Ka-Lau Mok, "Safty Analysis of Timing Properties in Real-Time System." IEEE Trans. Software Eng., vol.SE-12, pp. 890-904, September 1986.
9. Gustay Pospischil, "Developing Real-Time Tasks with Predictable Timing." IEEE Software, vol.9, no.5, pp.35-44, September 1992.
10. P. Coad & E. Yourdon, Object-Oriented Analysis, Second Edition, Prentice-Hall International Inc., 1991.
11. S. Shlaser & S.J.Mellor, Object-Oriented System Analysis : Modeling the World in Data, Prebtice-Hall, 1989.
12. G.Booch, Object-Oriented Design with Applications, Benjamin/Cummings, 1991.
13. Rumbaugh, Js., et al, "Object-Oriented Modeling and Design," July, December 1992.
14. CCITT, "Functional Specification and Description Language(SDL)." Recommendation Z.100 and Annexes A, B, C, D, and E, Blue Book, Nov. 1988.
15. Roberto Saracco, J.R.W.Smith, and Rick Reed, "Telecommunications Systems Engineering using SDL," North-Holland, 1989.
16. Rolv Brðεk & ϕystein Haugen, Engineering Real Time Systems : An Object-Oriented Methodology using SDL, Oslo Trondheim, June, 1992.

김 영 란(Young-Ran Kim) 정회원

1988년 2월 : 충북대학교 전산통계학과 졸업(이학사)

1989년 3월~1991년 2월 : 충북대학교 대학원 전자계산학과 졸업(이학석사)

1994년 2월 : 충북대학교 대학원 전자계산학과 박사과정 수료

1994년 3월~현재 : 충청전문대학 전임강사

※주관심분야 : 객체 지향 실시간 소프트웨어 명세화, 소프트웨어 재사용

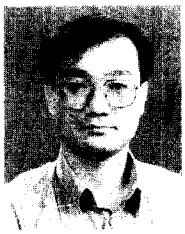
권 영 희(Young-Hee Kweon) 정회원

1987년 2월 : 충남대학교 계산통계학과 졸업(이학사)

1989년 2월 : 충남대학교 대학원 계산통계학과 졸업(이학석사)

1994년 3월~1996년 2월 : 충북대학교 대학원 전자계산학과 박사과정 수료

※주관심분야 : 소프트웨어 검색 기법, 소프트웨어 재사용



홍 성 백(Sung-Back Hong)정회원

1982년 2월 : 광운대학교 전자통신 졸업(공학사)

1990년 8월 : 연세대학교 대학원 전자계산학과 졸업(공학석사)

1982년 3월~현재 : 한국전자통신연구소 교환기술연구단 선임연구원

※주관심분야 : S/W형상관리, S/W 신뢰도

박 용 문(Yong-mun Park) 정회원

1983년 8월 : 숭전대학교 계산통계학과 졸업(이학사)

1985년 8월 : 중앙대학교 대학원 전자계산학과 졸업(이학석사)

1985년 12월~현재 : 한국전자통신연구소 교환기술연구단 선임연구원

※주관심분야 : S/W 개발환경, 패킷 망연동, 데이타 통신망

구 연 설(Yeon-Seol Koo)정회원

1964년 : 청주대학교 상학과 졸업(상학사)

1975년 : 성균관대학교 경영대학원 전자자료처리학과 졸업(경영학석사)

1981년 : 동국대학교 대학원 통계학과 졸업(이학석사)

1988년 : 동국대학교 대학원 전자계산학과 졸업(이학박사)

1979년~1996년 : 충북대학교 전자계산소장 역임
한국정보과학회 이사·전산교육연구회 위원장·충청지부장·부회장
충북대학교 자연과학 대학장 역임

현재 : 충북대학교 컴퓨터과학과 교수

※주관심분야 : 소프트웨어 공학, 정보통신, 알고리즘 등.