

# 시스템 형식론에 의한 사용자 인터페이스 시스템 설계 방법

김은하, 조대호

성균관대학교 전기전자 및 컴퓨터 공학부

## A Design Method of UI System based on Formal System Specification

Kim, Eun Ha Cho, Tae Ho

Dept. of Electrical and computer engr., Sung kyun kwan University

### 요 약

본 논문에서는 소프트웨어 시스템의 설계 및 구현 과정에서 있을 수 있는 설계 변경 및 이에 따른 다른 변경 요인들을 정확하게 파악하고, 구현상의 변경으로 인한 전체 시스템의 영향 등을 체계적으로 정립하는 소프트웨어 설계 방법론을 특정 공정의 사용자 인터페이스 시스템을 통해 제시하였다. 대상 시스템, 즉 인터페이스 시스템을 시스템 이론(System theory)에서 정의하는 구조적 입출력 시스템 레벨(Structural I/O System level)의 요소들로 표현하고 다시 구조적 입출력 시스템 레벨을 입출력 시스템 레벨(I/O System level)로 변환하였다. 이를 다시 DEVS 모델로 재구성하여 DEVS(Discrete Event system Specification) 시물레이션 환경에서 제공하는 시물레이터를 통하여 대상 시스템의 중요한 동적 특성을 소프트웨어 초기설계 시 또는 설계 변경 후 미리 파악할 수 있도록 하였다.

### 1. 서론

사용자 인터페이스는 사용자와 응용 시스템 간의 상호 대화를 위한 대화창구로서 사용자가 응용 시스템에 입력을 주고, 응용 프로그램이 사용자에게 결과를 보여준다. 인터페이스 소프트웨어는 매우 크고 복잡하여 작성하기 어려울 뿐만 아니라, 많은 장치들을 빈번히 제어해야 하고, 비동기적으로 사건의 입력 스트림들을 보내야 하며,

자주 프로토타입으로 만들어지고 반복적으로 수정되어야 한다[6]. 그러나 인터페이스 시스템은 수행절차가 단순하고 처리과정이 복잡하지 않으므로 사상이론을 적용하여 자동 시스템을 구현하는 것이 용이하다. DEVS 시물레이션 환경에서 제공하는 시물레이터를 통하여 인터페이스 시스템의 전체적인 구조와 상태의 변화를 정확하게 파악하고, 중요한 동적 특성을 초기 설계 시 또는 설계 변경 후 미리 파악할 수 있는 시스템을

제안하고자 한다.

## 2. 관련 기초 이론

시스템 설계시 적용되는 이론은 다음과 같다.

- 객체지향형 프로그래밍 : 프로그램을 객체로 분해하고, 그들 사이의 관계를 설정하는 과정으로 객체를 중심으로 모듈을 분해하는 기법이다 [8].
- 시스템 명세(System Specification) : 비형식적인 모델(informal model)을 컴포넌트(component)와 컴포넌트간의 상호작용(component interaction), 영향도(influence diagram) 등에 의해 형식적인 5계층으로 표현한다[1]. 5계층 중에서 2,3 계층을 적용하여 설계하였다.
- 시스템 구조관계(System Morphism) : 같은 레벨에서 한 시스템과 다른 시스템과의 관계를 정의한다[1].
- DEVS(Discrete Event system Specification) Formalism : 이산사건 시스템을 객체지향 관점에서 모듈화하여 계층적으로 잘 기술해 줄 수 있으며 모델들의 동적특성을 시물레이션 해주는 알고리즘이 제공되는 형식론이다. 모델링 대상인 객체들을 명료하고 일반적인 방법으로 기술함으로써 일관된 시스템의 표현을 가능케 한다. 형식론을 채택하여 시스템을 관찰하고자 하는 초점에서 추상화하여 모델링함으로써 모델과 실제 문제와의 관련성을 높일 수 있다. DEVS 모델은 다음과 같은 항들로 명세할 수 있다[7] :

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

각 항들의 의미는 다음과 같다.

X : 입력사건 집합(Input Events Set);

S : 상태변수 집합(State Set);

Y : 출력사건 집합(Output Events Set);

$\delta_{int} : S \rightarrow S$ , 내부전이 함수(Internal Transition Function);

$\delta_{ext} : Q \times X \rightarrow S$ , 외부전이 함수(External Transition Function);

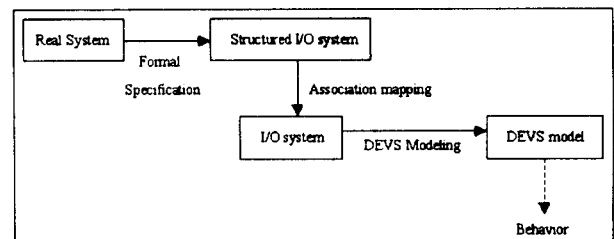
$Q = \{(s, e) \mid s \in S, 0 \leq e \leq ta(s)\}$  : total state of M;

$ta : S \rightarrow Real$ , 시간진진 함수(Time Advance Function);

$\lambda : S \rightarrow Y$ , 출력함수(Output Function);

## 3. 시스템 명세 및 구조관계

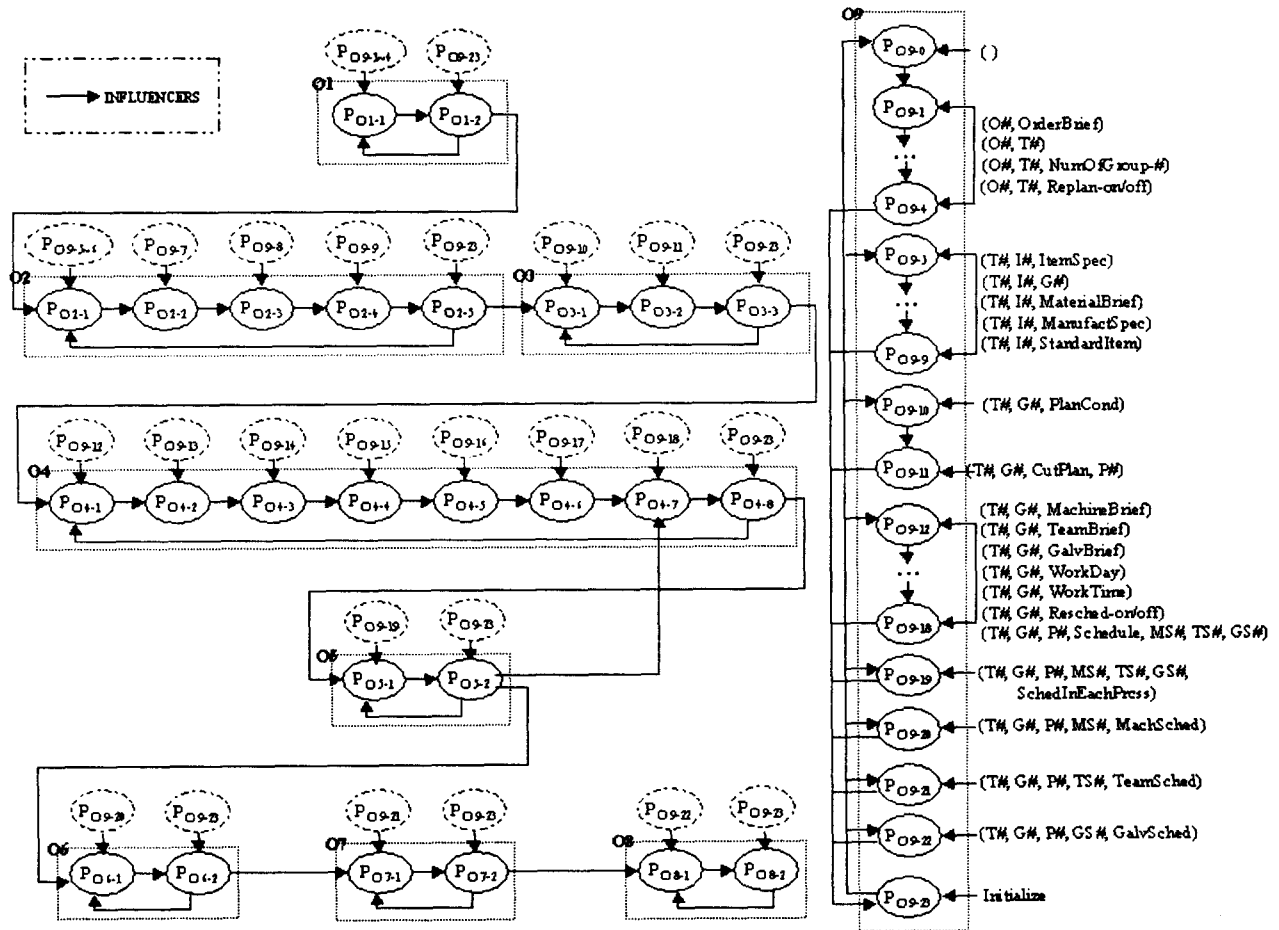
인터페이스 시스템을 시스템 이론(System theory)[2]에서 정의하는 구조적 입출력 시스템(Structured I/O System)의 요소들로 표현하고 다시 구조적 입출력 시스템을 입출력 시스템(I/O System)로 변환한다. 이를 다시 DEVS 모델로 재구성하여 DEVS 시물레이션 환경에서 제공하는 시물레이터를 통하여 대상 시스템의 중요한 동적 특성을 소프트웨어 초기 설계 시 또는 설계 변경 후 미리 파악할 수 있도록 설계한다. 각 시스템과 시스템간의 구조관계는 <그림 1>과 같다.



<그림 1> 시스템과 구조관계

### 3.1 시스템 명세

- 실시스템(Real System) : 인터페이스 시스템으로서 소프트웨어 공학적인 토대와 객체지향형 프로그래밍 사고들에 의해서 구성되어 있고, 모델에 적용된 예는 <표 1>와 같다.
- 구조적 입출력 시스템(Structured I/O System) :



<그림 2> 구조적 입출력 시스템의 구성요소간의 영향도

하위 레벨의 원시 집합과 함수를 추상화하여 표현한 구조적 모델[1]로서 정의 및 적용 예는 <표 2>과 같다. 각 구성요소(Coordinate)간에 미치는 영향은 <그림 2>와 같다.

- 입출력 시스템(I/O System) : 입력 세스먼트에 의한 상태전이와 특정 상태에서 발생하는 출력함수 등을 표현한 추상화된 집합과 함수로 구성되는 시스템[1]으로서 정의 및 적용 예는 <표 3>와 같다.

- DEVS 모델(DEVS Model) : 이산사건 시스템을 모듈화하고 계층적으로 잘 기술하여 동적특성을 DEVS 시물레이션 환경에서 시물레이션할 수 있는 모델로서 정의 및 적용 예는 <표 4>와 같다.

### 3.2 구조 관계

- 형식 명세(Formal Specification) : 실시스템, 즉 인터페이스 시스템을 구조적 입출력 시스템의 요소들로 추상화하여 표현한다. 정의 및 적용 예는 <표 5>과 같다.

- 관계 사상(Association Mapping) : 구조적 입출력 시스템에서 입출력 시스템으로 변환한다. 정의 및 적용 예는 <표 6>과 같다.

- DEVS 모델링(DEVS Modeling) : 입출력 시스템에서 DEVS 모델로 재구성한다. 정의 및 적용 예는 <표 7>과 같다.

<표 1> 인터페이스 시스템

| 인터페이스 시스템   |   |
|---|---|
| . 객체지향 프로그래밍 개념 적용<br>. 개발에 사용된 툴 : Visual C++, SpreadSheet, ChartFX  |   |
| <클래스><br>COrderBriefView → OnInitialUpdate(), OnEdit(), ..., OK(), OnGoTop()<br>CPlanDetailView → OnInitialUpdate(), OnDbISpread(), ..., OnOK() | <멤버함수><br>CUIApp → InitInstance(), OnInitialize() |

<표 2> 구조적 입출력 시스템

| 구조적 입출력 시스템 |  |
|-------------|--|
| 정의          | $M(\text{machine}) = \langle Q, \delta \rangle$<br>Where $Q$ : the cross product of the component states sets<br>$\delta$ : the cross product of component functions<br><br>structure of $M: (D, \{Q_\alpha\}, \{I_\alpha\}, \{\delta_\alpha\})$<br>where $D$ (a set of coordinates)<br>$\{Q_\alpha   \alpha \in D\}$ (a family of range sets)<br>$\{I_\alpha   \alpha \in D, I_\alpha \subseteq D\}$ (influencers)<br>$\{\delta_\alpha   \alpha \in D, \delta_\alpha: X_\alpha \rightarrow Q_\alpha\}$<br>(a family of local transition functions)<br>such that $Q \subseteq \times_{\alpha \in D} Q_\alpha$ and<br>$\delta = \times_{\alpha \in D} \delta_\alpha \circ \text{proj}_{I_\alpha}$ restricted to $Q$   |
| 적용          | $M = \langle Q, \delta \rangle$<br>$Q = (P_{01}, P_{02}, \dots, P_{09})$ , $\delta = (\delta_{01}, \delta_{02}, \dots, \delta_{09})$<br><br>$(D, \{Q_\alpha\}, \{I_\alpha\}, \{\delta_\alpha\})$<br>$D = \{00, 01, 02, \dots, 09\}$<br>$\{Q_\alpha\} = \{P_{01}, P_{02}, \dots, P_{09}\}$<br>for each $\alpha \in D$ , and $P_{01} = (P_{011}, P_{012})$ ,<br>$P_{02} = (P_{021}, P_{022}, P_{023}, P_{024}, P_{025})$ ,<br>$\dots$ , and $P_{09} = (P_{091}, P_{092}, \dots, P_{09n})$<br>$\{I_\alpha\} = \{I_{01}, I_{02}, \dots, I_{09}\}$<br>such that $I_{01} = \{01, 09\}$ , $I_{02} = \{01, 02, 09\}, \dots$ ,<br>and $I_{09} = \{09\}$<br>$\{\delta_\alpha\} = \{\delta_{01}, \delta_{02}, \dots, \delta_{09}\}$<br>such that $\delta_{01}: P_{01} \times P_{09} \rightarrow P_{01}$ ,<br>$\delta_{02}: P_{01} \times P_{02} \times P_{09} \rightarrow P_{02}, \dots$ , and $\delta_{09}: P_{09} \rightarrow P_{09}$ |

<표 3> 입출력 시스템

| 입출력 시스템 |   |
|---------|---|
| 정의      | $S(\text{system}) = \langle T, X, \Omega, Q, Y, \delta, \lambda \rangle$<br>Where $T$ : time base<br>$X$ : the input value set<br>$\Omega$ : the input segment set (a subset of $(X, T)$ )<br>$Q$ : the state set<br>$Y$ : the output value set<br>$\delta$ : the state transition function<br>$\lambda$ : the output function  |
| 적용      | $S = \langle T, X, \Omega, Q, Y, \delta, \lambda \rangle$<br>$T = \text{Integer}$<br>$X = \{O\#, (O\#, \text{OrderBrief}), (O\#, T\#), \dots, \text{Initialize}\}$<br>$\Omega = \{\omega   \omega: \langle 0, 1 \rangle \rightarrow X\}$<br>$Q = \{q_0, q_1, q_2, \dots, q_{31}\}$<br>such that $q_0 = (P_{011}, P_{021}, P_{031}, P_{041}, P_{051}, P_{061}, P_{071}, P_{081}, P_{091})$ ,<br>$q_1 = (P_{012}, P_{022}, P_{032}, P_{042}, P_{052}, P_{062}, P_{072}, P_{082}, P_{092})$ ,<br>$\dots$ , and $q_{31} = (\dots, \dots, P_{081}, P_{091})$<br>$Y = \{O\#, (O\#, \text{OrderBrief}), (O\#, T\#), \dots, \text{Initialize}\}$<br>$\delta(q) = (\delta_{01} \circ \text{proj}_{I_{01}}(q), \delta_{02} \circ \text{proj}_{I_{02}}(q), \dots, \delta_{09} \circ \text{proj}_{I_{09}}(q))$ ,<br>$q \in \{q_0 \times q_1 \times \dots \times q_{31}\}$<br>$\lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{30})$ |

<표 4> DEVS 모델

| DEVS 모델 |  |
|---------|--|
| 정의      | $M(\text{model}) = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$<br>Where $X$ : set of external (input) event types<br>$S$ : sequential state set<br>$Y$ : output set<br>$\delta_{int}: S \rightarrow S$ , the internal transition function<br>$\delta_{ext}: Q \times X \rightarrow S$ , the external transition function<br>where $Q$ is the total state set = $\{(s, e)   s \in S, 0 \leq e \leq ta(s)\}$<br>$ta: S \rightarrow R^+, 0, \text{inf}$ , the time advance function<br>$\lambda: S \rightarrow Y$ , the output function |
| 적용      | $M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$<br>$X = \{', (O\#, \text{OrderBrief}), (O\#, T\#), \dots, \text{Initialize}\}$ , $x \in X$<br>$S = \{ \text{PASSIVE}, \text{TGA-1}, \text{TGA-2}, \dots, \text{PROCESSED} \}$ , $s \in S$<br>$Y = \{ (O\#, \text{OrderBrief}, T\#, \text{NumOfGroup-}\#, \text{Replan-on/off}), \dots, (T\#, G\#, P\#, GS\#, \text{GalvSched}) \}$ , $y \in Y$<br>$\delta_{int}(s \in X) = s + x$<br>$\lambda(s) = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_p)$                             |

<표 5> 형식 명세

| Formal Specification |   |
|----------------------|---|
| 정의                   | Recall that a homomorphism from $M = \langle Q, \delta \rangle$ to $M' = \langle Q', \delta' \rangle$ is a map $h$ such that $h: Q \rightarrow Q'$ ,<br>$h(\delta(q)) = \delta'(h(q))$ for all $q \in Q$<br>Now let $M$ and $M'$ be structured by $(D, \{Q_\alpha\}, \{I_\alpha\}, \{\delta_\alpha\})$ and $(D', \{Q'_\alpha\}, \{I'_\alpha\}, \{\delta'_\alpha\})$ , respectively<br>$d: D \rightarrow D'$ (coordinate mapping), where $d^{-1}(\alpha') = \{\alpha   d(\alpha) = \alpha'\}$<br>$h: Q \rightarrow Q'$ , where $h = \times_{\alpha \in D} h_\alpha \circ \text{proj}_{I_\alpha}$ |
| 적용                   | Real System $\rightarrow$ Structured I/O System<br>$D: COrderBriefView \times CPlanDetailView \times \dots \times CUIApp$<br>$\rightarrow O1 \times O2 \times \dots \times O9$<br>$Q: COrderBriefView.P1 \times CPlanDetailView.P2 \times \dots \times CUIApp.P9$<br>$\rightarrow P_{01} \times P_{02} \times \dots \times P_{09}$<br>such that $P_{01} = (P_{011}, P_{012})$ ,<br>$P_{02} = (P_{021}, P_{022}, P_{023}, P_{024}, P_{025}), \dots$ ,<br>and $P_{09} = (P_{091}, P_{092}, \dots, P_{0923})$  |

<표 6> 관계 사상

| Association Mapping |   |
|---------------------|---|
| 정의                  | Structured machine $M = \langle Q, \delta \rangle$<br>Structure of $M: (D, \{Q_\alpha\}, \{I_\alpha\}, \{\delta_\alpha\})$<br>such that $Q \subseteq \times_{\alpha \in D} Q_\alpha$ and<br>$\delta = \times_{\alpha \in D} \delta_\alpha \circ \text{proj}_{I_\alpha}$ restricted to $Q$<br>Letting $\delta'$ denote the cross product, $\delta'(q) = \times_{\alpha \in D} \delta_\alpha \circ \text{proj}_{I_\alpha}(q)$   |
| 적용                  | Structured I/O System $\rightarrow$ I/O System<br>$Q: (P_{01}, P_{02}, \dots, P_{09}) \rightarrow (q_0, q_1, q_2, \dots, q_{31})$<br>such that $q_0 = (P_{011}, P_{021}, P_{031}, P_{041}, P_{051}, P_{061}, P_{071}, P_{081}, P_{091})$ ,<br>$q_1 = (P_{012}, P_{022}, P_{032}, P_{042}, P_{052}, P_{062}, P_{072}, P_{082}, P_{092})$ ,<br>$\dots$ , and $q_{31} = (\dots, \dots, P_{081}, P_{091})$<br>$\delta: \delta(q) = (\delta_{01} \circ \text{proj}_{I_{01}}(q), \delta_{02} \circ \text{proj}_{I_{02}}(q), \dots, \delta_{09} \circ \text{proj}_{I_{09}}(q))$ ,<br>$q \in \{q_0 \times q_1 \times \dots \times q_{31}\}$ |

<표 7> DEVS 모델링

| DEVS Modeling |  |
|---------------|--|
| 정의            | $M = \langle X_M, S_M, Y_M, \delta_{int}, \delta_{ext}, \lambda_M, ta \rangle$<br>Where $X_M = X'$<br>$S_M = Q'$<br>$Y_M = Y'$<br>$\delta_{int}: Q_M \times Q' \rightarrow Q_M, Q_M = \{(s,e)   s \in S_M, 0 \leq e \leq ta(s)\}$<br>$\delta_{ext}: Q_M \times \phi \rightarrow Q_M$<br>$\lambda_M = \lambda'$ |
| 적용            | I/O System $\rightarrow$ DEVS Model<br>$S: (q_0, q_1, q_2, \dots, q_{31})$<br>$\rightarrow (PASSIVE, TGA-1, \dots, PROCESSED)$<br>$\delta: \delta(q) \rightarrow \delta_{ext}$   |

#### 4. 결론

소프트웨어를 개발할 때는 반복적이고 지속적인 기능 추가 또는 변경에 의한 수정이 이루어지기 마련이다. 그러나 소프트웨어의 규모가 커지고 복잡해질수록 이와같은 작업은 어려워진다. 특히 많은 장치들을 빈번히 제어해야 하고, 사용자의 행동과 시스템 반응 사이에 지체감이 없도록 보장해야 한다는 강력한 요구를 만족해야 하는 인터페이스 시스템[6]의 경우는 더욱 그러하다. 그러나 반면 수행절차가 단순하고 처리과정이 복잡하지 않은 인터페이스 시스템의 특성을 이용하여 시스템 이론에서 정의하는 구조관계를 통해 구조적 입출력 시스템 레벨의 요소들로 표현하고, 입출력 시스템 레벨로 변환한 다음 이를 다시 DEVS 모델로 재구성하였다.

추후 연구되어야 할 과제는 DEVS 시뮬레이터를 통해 소프트웨어 시스템을 개발할 때마다 반복적으로 되풀이되는 수정에 의한 번거로움을 덜어줄 뿐만 아니라, 전체 시스템의 구조나 흐름을 파악하여 변경된 부분을 자동으로 추적하고, 그 역의 기능도 가능하도록 구현하는 것이다. 그리고 인터페이스 시스템에 한정된 모델링이 아니라 시간 의존적인 시스템 등과 같이 더 일반적인 시스템에의 적용도 가능할 수 있도록 연구가 이루어져야 할 것이다.

#### 참고문헌

- [1] Bernard P.Zeigler, "Theory of Modelling and Simulation", New York: Wiley, 1976.
- [2] L.Padulo and M.A.Arvid, "System Theory", Suders, Philadelphia, Pa., 1974.
- [3] Bernard P.Zeigler, "Multifaceted Modelling and Discrete Event Simulation", Orlando, FL, USA: Academic Press, 1984.
- [4] Bernard P.Zeigler, "Object-Oriented Simulation with Hierarchical, Modular Models", San Diego, CA,USA: Academic Press, 1990.
- [5] 조대호,이철기, "계층의 구조를 갖는 시물레이션 모델에 있어서 단계적 접근을 위한 모델 연결 방법론과 그 적용 예", 시물레이션학회 논문지, Vol 5, No 2, 1996.
- [6] 김상근,홍찬기,이경환, "소프트웨어 재사용 시스템을 지원하는 사용자 인터페이스 구축기의 설계 및 구현", Vol 2, No 3, pp.324-333, 1995.
- [7] 안명수,박성봉,김탁곤, "DEVSsim++: 의미론에 기반한 이산사건 시스템의 객체지향 모델링 및 시물레이션 환경", 정보과학회 논문지, Vol 21, No 9, pp.1652-1664, 1994.
- [8] 허계범, 최영근, "객체지향 설계를 위한 모듈 분해 방법", Vol 2, No 3, pp.299-313, 1995.
- [9] 김행곤,한은주,정연기, "객체지향을 기반으로 한 추상화 정보의 시각화 시스템에 대한 연구", Vol 4, No 10, pp.2434-2444, 1997.
- [10] 임근,이경환, "객체지향 프로그램에서 이벤트 추상화 표현", 정보처리 논문지, Vol 4, No 5, pp.1257-1266, 1997.
- [11] 김덕현,박성주, "확장된 객체지향 데이터 모형을 이용한 소프트웨어 변경관리 시스템", 정보과학회 논문지, Vol 22, No 2, pp.249-260, 1995.