

규칙 기반의 지능적 사용자 인터페이스*

양영수*·이민석*·김재희*·장문익*·박충식**

An Intelligent User Interface using Rule-based Method

Young Soo Yang*·Min Suk Lee*·Jai Hie Kim*
Chang Moon Ik*·Choong Shik Park**

요 약

이 논문에서는 상황에 적합한 정보들을 선택하고 이를 해당되는 GUI(Graphic User Interface) 객체를 통해 사용자에게 제시하기 위한 규칙 기반의 지능적 사용자 인터페이스 구조를 제안하고, 이를 이용하여 실제로 군사 분야 정보 융합 시스템의 그래픽 사용자 인터페이스를 개발하였다. 군사 분야와 같이 복잡하고 급변하는 상황에서는 상황에 따라 필요한 정보가 다를 뿐만 아니라, 불필요한 정보의 제시로 인해 오히려 사용자의 판단에 혼란을 초래할 수 있다. 이 논문에서는 이러한 점을 고려하여, 제시할 정보와 GUI 객체의 적절한 대응뿐만 아니라, 상황에 따라 사용자에게 필요한 정보를 해당 분야의 전문가로부터 획득한 규칙에 의해 선택적으로 제시하도록 하기 위한 GUI 구조를 제안하였다. 개발한 GUI의 구조는 입력 해석, 제시할 정보의 선택, GUI 객체의 선택, GUI 객체의 생성 및 속성 지정, GUI 객체의 배치 및 출력 단계들로 구성되며, 각각의 단계는 규칙에 의해 처리된다. 제안한 구조에 의해 공중 작전 수립을 위한 정보 융합 시스템의 사용자 인터페이스를 개발한 결과, 다양한 매체를 통해 많은 데이터가 들어오는 상황에서 필요한 정보만을 선택하여 적절히 제시해 줌으로써 사용자의 신속한 상황 판단과 결정 수립을 지원할 수 있었다.

Key Words : 지능적 사용자 인터페이스(intelligent user interface), 규칙 기반 시스템(rule-based system), 객체 지향 기법(object-oriented method), 전문가 시스템(expert system)

*이 연구는 과학재단의 핵심기초연구(과제번호:961-0901~003~2)의 지원으로 이루어졌음
*연세대학교 기계·전자공학부
**영동공과대학 전자계산학과

1. 서 론

최근 컴퓨터 사용 인구가 급속히 증가하면서 컴퓨터와 사용자간의 효율적인 사용자 인터페이스에 대한 관심이 증가하였다. 사용자 인터페이스는 시스템과 대화하는 수단이다.

사용자는 시스템의 내부나 소프트웨어의 구현 방법에는 상관없이 제공되는 사용자 인터페이스에 의해 시스템을 사용하게 된다. 하드웨어적인 제약이 많았던 과거에는 복잡한 소프트웨어의 개발에 한계가 있었을 뿐만 아니라, 메모리 제한으로 인해 적은 양의 코드만이 사용자 인터페이스에 할당되었다. 그러나 하드웨어적인 제약이 적어진 지금은 소프트웨어의 사용 편리성이 중요한 문제로 부각되어 사용자 인터페이스의 중요성이 크게 대두되고 있다[Ebe94].

특히, 복잡한 문제 영역을 다루는 시스템의 경우 사용자가 최소한의 노력으로 원하는 작업을 수행할 수 있도록 하는 지능적인 사용자 인터페이스가 요구된다. 이러한 사용자 인터페이스는 상황에 따라 가장 적절한 방법으로 사용자와 시스템간의 상호 작용이 이루어지도록 지원해야 한다. 그러나 복잡하고 가변적인 모든 상황에 대해 사용자와 시스템간의 상호 작용을 일일이 정의하여 디자인하는 것은 매우 어려울 뿐만 아니라 많은 시간과 노력을 필요로 한다. 이러한 문제의 해결을 위해 지식 기반 기법을 이용하여 사용자 인터페이스를 개발할 수 있다. 즉, 전문가의 지식을 이용하여 어플리케이션 영역 내에서 추론을 수행함으로써 사용자와 시스템간의 복잡한 인터페이스를 쉽게 표현할 수 있을 뿐만 아니라 일반적인 경우로의 확장이 가능하다.

지금까지 지식 기반 기법을 이용한 사용자 인

터페이스 개발에 관한 연구는 주로 사용자의 불확실한 입력 해석[Nea91]이나 모순 해결[Chi91], 자연어 인식[Nea91] 등 사용자 입에 관한 지능적인 처리에 초점을 맞추었다. 그러나 군사 분야 시스템 등과 같이 신속하고 정확한 대응이 요구되는 시스템에서는 사용자의 적절한 상황 판단을 뒷받침하기 위한 정보의 지능적인 제시 측면이 중요하다. 그러므로 이 논문에서는 이러한 측면에 초점을 맞추어 사용자 인터페이스를 구현하였다.

기존의 지능적 정보 제시를 위한 사용자 인터페이스들은 상황에 따라 사용자가 요구한 정보를 적절한 방법으로 제시하거나[Ty191], 정해진 정보들에 대해 대응되는 GUI 객체의 속성을 적절히 변화시켜 보여주는 데[Are91] 그치고 있다. 그러나, 군사 분야와 같이 복잡하고 다양한 상황이 발생하는 영역에서는 상황에 따라 필요한 정보가 다를 뿐만 아니라, 불필요한 정보가 제시되는 경우 오히려 사용자의 판단에 혼란을 초래하게 된다.

이 논문에서는 이러한 점을 보완하기 위해, 규칙 기반 기법과 객체 지향 기법을 바탕으로 하여, 작업 영역 객체와 GUI 객체의 적절한 대응뿐만 아니라 상황에 따라 사용자에게 필요한 정보를 선택적으로 제시하기 위한 GUI 구조를 제안한다. 그리고, 제안한 구조에 의해 공군 작전 수립을 위한 정보 융합 시스템의 그래픽 사용자 인터페이스를 개발하였다. 인터페이스에 필요한 지식들은 규칙(rule)을 이용하여 표현하였고, 제안한 GUI 구조는 인터페이스 모듈로 들어오는 입력을 해석하는 부분과 제시할 정보를 선택하는 부분, 작업 영역 객체에 따라 적절한 GUI 객체를 선택하는 부분, 그리고 GUI 객체를 실제로 생성하고 속성을 지정하는 부분과 이를 적절히 배치하여 출력하는 부분으로 구성되어

있다. 특히 앞서 언급한 상황에 따른 선택적 정보 제시 기능을 위해, 전문가로부터 획득한 작업 영역 지식들에 의해 제시할 정보를 선택하는 부분을 포함하였다.

이처럼 규칙 기반 기법을 이용하여 사용자 인터페이스를 개발함으로써 일반적인 서술에 의한 사용자 인터페이스의 개발이 가능하고 개발 속도가 상대적으로 단축되며, 소프트웨어의 검증과 수정 작업을 훨씬 용이하게 수행할 수 있는 부수적인 효과도 거둘 수 있다.

제 2 절에서는 지능적 사용자 인터페이스의 특징과 필요성에 관한 개괄적인 내용을, 제 3 절에서는 본 논문에서 제안한 사용자 인터페이스의 구조에 관한 설명을, 제 4 절에서는 제안한 방법을 사용하여 실제로 정보 융합 시스템의 사용자 인터페이스를 제작한 결과를 설명한다. 끝으로 제 5 절에서는 결론과 향후 연구 과제에 대해 논의한다.

2. 지능적 사용자 인터페이스

2.1 지능적 사용자 인터페이스의 특징

소프트웨어 시스템의 기능이 복잡해지고 사용자의 계층이 다양해짐에 따라, 사용자 인터페이스의 중요성이 크게 강조되고 있다. 시스템이 제공하는 기능들이 제대로 사용되기 위해서는 사용자 인터페이스는 사용자가 원하는 작업을 중심으로 환경을 제공하여야 하고, 다양한 계층의 사용자들을 지원할 수 있어야 한다.

사용자 인터페이스를 향상시키고자 하는 노력은 다양한 분야에서 이루어져 왔다. 입력과 출력 도구에 관한 연구를 통해 사용자의 음성이나 필기체 등을 이용한 입력, 출력이 가능하게 되

었으며[Sch87], 컴퓨터와의 대화 방법에 관한 연구를 통해 명령 언어 방식을 비롯하여 메뉴 방식과 아이콘을 이용한 직접 제어 방식이 널리 이용되고 있다. 그리고 나아가서는 멀티미디어를 이용한 다양한 인터페이스가 개발되었다. 그러나, 이러한 기술적인 면에 대한 연구만으로는 사용자가 사람과 대화하듯이 편리하게 컴퓨터를 사용하려는 목표가 이루어지기 힘들기 때문에, 지능적 사용자 인터페이스의 연구가 필요하게 되었다.

사실, 사용자 인터페이스가 지능적인가 아닌가를 판별하는 기준은 매우 모호하고, 많은 사용자 인터페이스들이 나름대로의 지능성을 주장하기도 한다. 그러나, 지능적 사용자 인터페이스들은 대략 다음과 같은 특징을 갖는다고 할 수 있다.

첫째, 상황에 따라 사용자의 작업 목표와 의도를 파악하여 적절하고 효율적인 인터페이스를 제공한다. 지능적 사용자 인터페이스의 구현을 위해서는 지능적인 동작을 위해서 상황을 판단하고 적절한 동작을 취하기 위한 지식을 가지고 있어야 한다[Hol91]. 지능적 사용자 인터페이스는 이러한 형태의 지식을 체계적이고 충분한 양으로 보유하여, 사용자의 애매한 요구에도 적절한 동작을 할 수 있어야 한다.

둘째로는 사용자가 주어진 인터페이스에 자신을 맞추는 것이 아니라, 사용자에 따라서 인터페이스가 사용 방법을 변화시켜서 넓은 사용자층을 끌고루 만족시킬 수 있게 하는 것이다. 초보자에게는 사용의 편리성을 주고, 숙련자에게는 효율성을 동시에 제공할 수 있어야 한다. 즉 사용자의 특성과 숙련도에 따라서 인터페이스를 변화시키는 적응형 인터페이스[Tyl91]도 지능적 사용자 인터페이스의 큰 특징이다.

마지막으로 지능형 인터페이스는 작업에 대

한 사용자의 잘못된 인식을 바로 잡는다는 특징을 갖는다. 인터페이스의 심각한 문제점들은 대부분 사용자가 요구한 명령이 작업 영역에서 의미적으로 잘못되어 있을 때에 발생하므로, 지능적 인터페이스는 이러한 사용자의 명령에 대한 적당한 피드백을 제공해야 한다. 즉 사용자가 문제 영역에서 의미적으로 바른 작업을 요구할 수 있도록 돕는 것이다.

이러한 지능적 인터페이스는 사용자 인터페이스의 중요성이 가장 극대화된 것이며 사용자 인터페이스가 가장 발달된 형태라 할 수 있다.

2.2 군사 분야에서 지능적 사용자 인터페이스의 필요성

최근 급속한 컴퓨터 기술의 발달과 더불어 군사 분야 시스템의 복잡성이 나날이 증가함에 따라, 시스템과 사용자간의 효율적인 인터페이스의 필요성이 증대되고 있다. 즉, 군사 분야의 시스템 성능이 향상되고 센서가 다양해짐에 따라 얻어진 많은 양의 정보를 사용자로 하여금 보다 신속하고 용이하게 파악할 수 있도록 지원하는 효율적인 사용자 인터페이스의 중요성이 증대되고 있는 것이다.

이를 위해, 시스템으로부터 들어오는 많은 데이터를 최종적으로 사용자에게 제시하기에 앞서 중요도에 따라 선별하고, 이를 일정한 그래픽 이미지와 심벌들로 변환시켜 출력하는 과정을 거침으로써 사용자가 보다 신속하고 용이하게 정보를 받아들일 수 있도록 해야 한다. 또한 시스템이 사용자의 명령을 받아들이고 수행하며, 수행 결과를 보고하는 일련의 과정이 신속하고 원활하게 이루어져야 할 것이다. 다시 말해서, 어떤 긴급한 사건이 발생했을 경우 사용자는 신속하게 필요한 관련 자료를 제공받고, 이로부터

신속하게 판단을 내린 후 다음 동작을 시스템에 명령하고, 시스템이 다시 이를 수행하는 일련의 과정이 원활하게 이루어져야 한다. 이 때, 작업 영역에서의 상황에 따른 사용자의 다양한 정보 요구 내용과 사용자의 인지적 특성 등을 모델링하여, 이를 시스템 내에 지식 베이스로 구성하여 구동함으로써, 사용자가 직접 시스템에 정보를 요구하지 않고도 시스템 스스로 효율적으로 정보를 제시하도록 할 수 있다. 그에 따라 어떤 상황에서 사용자가 취해야 할 동작을 최소화 할 수 있고, 이것은 사용의 편리성을 제공할 뿐만 아니라 상황에 따른 신속한 대응을 가능하게 한다.

3. 제안한 사용자 인터페이스 개발 기법

지능적 사용자 인터페이스를 개발하기 위해서는 작업 영역과 사용자 영역에 관한 여러 가지 지식들을 필요로 한다[Mac91]. 크고 복잡한 시스템인 경우에는 사용자에게 영향을 미치는 작업 영역의 모든 상황을 고려하여 사용자 인터페이스를 디자인하는 것은 매우 복잡하고 어려운 일이다. 그러므로, 이러한 문제를 해결하기 위해 사용자 인터페이스의 개발에 지식 기반 기법 [Mac91]을 도입하고 사용되는 지식들을 규칙에 의해 표현하였다.

이 논문에서 규칙 기반 기법 및 객체 지향 기법을 이용하여 개발한 사용자 인터페이스는 작업 영역 객체와 인터페이스 객체 그리고 이 두 가지 요소를 연결하는 규칙들으로써 구성된다. 이것들은 각각 MVC(Model-View-Controller)[Ade90] 구조의 모델과 뷰 그리고 컨트롤러

에 해당한다. 즉, 일정한 형식에 의해 모델과 뷰에 해당하는 객체를 정의하고, 제안한 구조에 의해 규칙을 사용하여 이 두 가지 요소를 적절히 대응시키는 컨트롤러를 구현한 것이다.

먼저 제안한 시스템을 구현하기 위하여 사용된 객체, 규칙의 표현 및 처리 방법에 대하여 살펴보면 다음과 같다.

3.1 객체의 표현

3.1.1 객체의 구조

객체는 객체의 이름과 객체의 특성을 나타내는 속성값들, 그리고 일정한 메시지가 전달되었을 때 수행되는 메소드(method)로 구성되어 있다.

객체의 속성은 O-A-V(Object-Attribute-Value) 형식에 의해 나타낸다. 즉, 맨 앞에 객체의 이름을 나타내고 차례로 속성과 그에 따른 속성값을 지정하는 형식으로 나타낸다. 예를 들면, (rectangle width 100) (rectangle height 200) 등으로 나타내는데, 이것은 각각 객체 rectangle의 속성들 중 폭(width)이 100이고 높이(height)가 200임을 나타내는 것이다. 경우에 따라서는 이러한 속성값으로 다른 객체를 지정하기도 한다. 예를 들어, (text font-is font)에서는 객체 text의 폰트 속성값으로 지정된 font 역시 일정한 속성을 갖는 객체의 이름이다.

객체의 메소드는 (객체이름 method (:메소드 이름) (실제 수행할 코드들))로 표현하며 이는 특정 메시지가 이 객체에게 전달될 때 이 메시지를 처리하기 위한 방법을 표현한 것이다. 메소드 이름은 객체에게 전달될 메시지의 이름과

동일하게 사용된다.

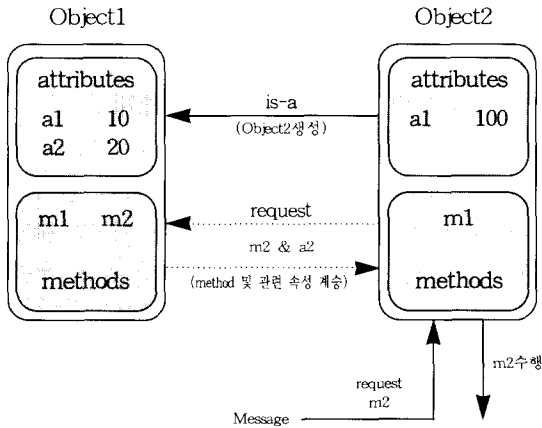
3.1.2 객체들간의 계층적 관계

객체들간의 계층적 관계는 "is-a"라는 키워드(keyword)에 의해 표현될 수 있다. 이 때 해당 객체는 지정된 객체의 하위 객체가 된다. 예를 들어, (w1 is-a window)라고 정의하면 w1은 객체 window의 하위 객체가 된다.

하위 객체는 상위 객체의 속성을 상속받는다. 위의 예에서, w1은 window객체의 위치 좌표, 넓이, 높이, 메뉴 등의 모든 속성을 상속받게 된다. 이 때, 상위 객체와 다른 값을 가지는 속성이 존재한다면 해당 속성값만을 따로 지정해 주면 된다. 예를 들어, w1의 넓이를 window객체와 다른 100으로 하고 싶다면, (w1 width 100)라고 정의해 줌으로써 넓이가 100이고 다른 모든 속성들은 window 객체와 동일한 객체 w1이 생성된다.

3.1.3 객체로의 메시지 전달

객체를 정의할 때에는 그 객체로 보내어지는 특정한 메시지에 대하여 수행되는 메소드를 정의할 수 있으며, 이것은 속성과 마찬가지로 상위 객체에서 하위 객체로 상속된다. [그림 1]은 이러한 객체들 간의 상속 관계를 나타낸다. 즉, "is-a"라는 키워드에 의해 object2가 object1의 하위 객체로 정의되면, object2는 object1의 속성과 메소드를 상속받게 된다. 그러므로, 메소드 m2가 정의되어 있지 않은 object2에 메소드 m2를 요구하는 메시지를 보내면, object1의 메소드를 상속받아 m2가 수행된다.



[그림 1] 객체들 간의 상속 관계

메소드를 지정할 때에는 리스트의 형태로 여러 개의 함수들을 지정해 줄 수 있으며, 이때 이러한 함수들은 리스트에 나타난 순서대로 차례로 수행된다. 또한 (send-message)라는 함수를 이용하여 상황에 따라 어떤 객체로 특정한 메시지를 보내도록 정의할 수 있다.

이러한 기능을 이용하면, 신속하게 정보를 확인할 필요가 있을 경우, 사용자의 입력을 기다리지 않고 시스템 스스로 관련 객체에 해당 메시지를 보내어 필요한 정보를 디스플레이 함으로써 사용자의 신속한 상황 파악을 지원할 수 있다.

3.2 규칙의 표현

이 논문에서 사용되는 규칙은 작업 영역 객체들과 인터페이스 객체들을 연결시켜 줌으로써 일정한 정보를 그에 상응하는 디스플레이의 형태로 변환하는 역할을 하는 것으로서, 인터페이스를 처리하는 각 단계 즉, 입력을 해석한 후 출력 객체를 선택하고 생성하며 화면상에 배치하여 출력하는 일련의 과정들이 모두 규칙에 의

해서 표현된다. 이러한 규칙들의 표현 형태는 다음과 같다.

```
(rule_name if (conditions))
(rule_name then (actions))
```

: (conditions)가 만족되면 (actions)를 수행한다. 여기서 조건부는 여러 개의 조건들로 이루어질 수 있다.

```
(conditions) => (condition1)...(conditionn)
```

결론부의 (actions)는 역시 여러 개의 동작들로 이루어질 수 있으며 동작 자체는 임의의 개수의 문자열들로 이루어진다.

```
(actions) => (action1)(action2)...(actionn)
```

이 동작의 결과는 새로운 사실을 도출할 수도 있고, 어떠한 특정 기능의 처리가 될 수도 있다. 결과로 새로운 사실이 추론되면 이를 단순히 작업 메모리에 추가한다. 만약 동작 부분에 나타나는 문자열이 미리 지정되어 있는 함수이면 그 함수를 수행하며, 또한 그함수의 수행 결과로서 어떠한 결과를 새로이 작업 메모리에 추가할 수도 있다.

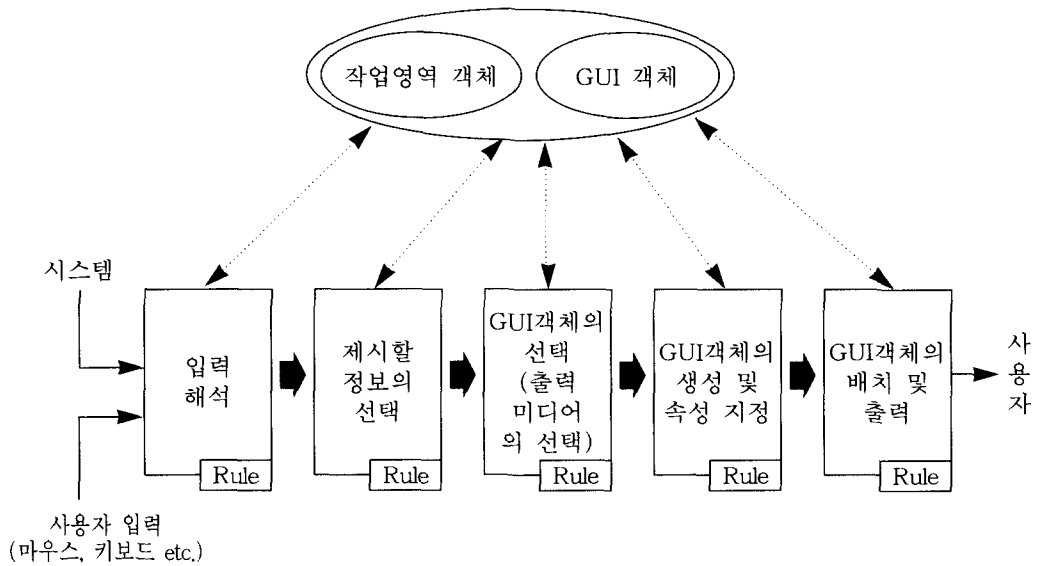
만약 인터페이스 시스템의 수정이 요구된다면, 단지 이 규칙들과 관련 객체들을 변화시킴으로써 그러한 과정이 신속하게 이루어질 수 있다.

3.3 제안한 구조

기존의 지능적 사용자 인터페이스 기법에 대한 분석을 통하여 이 논문에서 제안한 인터페이

스의 구조도는 [그림 2]와 같다. 이 구조는 크게 다섯 부분으로 나뉜다. 먼저 입력 해석 부분에서는 사용자로부터 넘어오는 데이터들을 일정한 심볼릭 데이터로 변환하고, 제시할 정보의 선택 부분에서는 전문가의 지식을 이용하여 디스플레이할 정보들을 선택한다. 그 후 선택된 정보에 가장 적절한 GUI 객체를 GUI 객체의 선택 부분에서 선택한다. 선택된

GUI 객체는 GUI 객체의 생성 부분에서 실제로 생성되어지며, 이는 최종적으로 GUI 객체의 배치 및 출력 부분에서 화면상에 표시된다. 이러한 각 부분에서 실제 사용되는 규칙의 수는 대략 10개 정도가 되며 “제시할 정보의 선택”에 해당하는 부분에서는 약 20개 정도의 지식이 구축되어 총 60개 정도의 규칙이 구축되어 있다.



[그림 2] 제안한 규칙 기반 사용자 인터페이스의 구조

3.3.1 입력 해석

입력 해석 부분에서는 사용자로부터 넘어오는 데이터를 일정한 심볼릭 데이터로 변환하고, 시스템으로부터 넘어오는 실세계 좌표를 스크린상의 좌표로 변환한다.

입력 메시지는 사용자로부터 받아들이는 메시지와 시스템 내부로부터 받아들이는 메시지로 구분할 수 있다. 사용자로부터 입력

메시지를 받아들이는 경우에는, 사용자가 마우스 입력을 가한 위치의 좌표 데이터를 심볼릭 데이터로 변환한 후, 이로부터 그 위치에서 일정한 범위안에 있는 객체를 찾아 제시할 정보를 선택한다. 따라서 사용자가 어느 객체에도 속하지 않는 불확실한 위치에 입력을 가했을 때에도 이에 적절히 대응할 수 있다.

3.3.2 제시할 정보의 선택

<의미>

규칙 S1 : 적기가 발견되고, 적기의 예상 공격 목표지가 결정되면, 예상 공격 목표지에 대한 관련 정보가 제시되어야 하고, 따라서 화면상에 표시되는 정보의 개수도 1만큼 증가시킨다.

규칙 S2 : 화면에 표시된 정보가 5개 보다 많으면, 좀 더 중요한 정보들만을 화면에 표시한다.

<표현>

```
(rule_S1 if (?x is detected) (?x allegiance enemy) (?x attack-target ?y))
```

```
(rule_S1 then (?y must-be displayed) (increase-information-number))
```

```
(rule_S2 if (?x must-be displayed) (?y must-be displayed) (!=?x ?y) (> (get-information-number) 5) (?x priority ?m) (?y priority ?n) (> ?m ?n) )
```

```
(rule_S2 then (?x is selected))
```

[그림 3] 제시할 정보의 선택 관련 규칙 예

군사 분야 등과 같이 복잡하고 다양한 상황이 발생하는 영역에서는 시간에 따라 매우 다양한 정보가 시스템으로부터 넘어오게 된다. 그러나, 사용자가 이들을 모두 파악하기는 매우 어려울 뿐만 아니라, 불필요한 정보가 제시되는 경우 오히려 사용자의 판단에 혼란을 초래하게 된다. 그러므로, 무수한 정보들 중에서 상황에 따라 중요한 정보만을 선택적으로 제시하고, 사용자가 필요할 때에는 추가적인 정보를 요구할 수 있도록 함으로써 보다 용이한 상황 판단과 결정 수립을 지원

할 수 있다. 또한, 사용자의 동일한 객체에 대한 정보 요구 메시지에 대해서도 상황에 따라 필요한 종류의 관련 정보를 선택적으로 제시하여 주는 것이 바람직하다. 이를 위하여, 제시할 정보를 선택하는 부분에서는 시스템으로부터 넘어온 데이터로부터 상황을 분석하여 일정한 사실(fact)을 생성하고, 이로부터 디스플레이할 정보들을 선택하게 된다. 이러한 과정은 작업 영역 전문가로부터 얻어진 지식에 의해 이루어지며 그러한 지식을 표현한 규칙을 예로 들면 다음의[그림 3]과 같다.

요약하면, 입력 해석과 제시할 정보의 선택 과정은 GUI로 넘어오는 데이터를 사용자가 확인할 수 있는 형태의 시각 객체로 변환하여 출력하는 다음 과정을 수행하기 위한 일종의 전처리 단계라고 할 수 있다.

3.3.3 GUI 객체의 선택

<의미>

규칙 O1 : 작업 영역 객체가 비행 기지일 경우에는 텍스트를 이용하여 관련 정보를 나타낸다.

규칙 O2 : 비행 편대일 경우에는 일반적으로 여러 대의 전투기로 구성되어 있다는 점을 고려하여 테이블을 이용하여 나타내도록 한다.

<표현>

```
(rule_O1 if (?x must-be displayed) (?x is-a air_base) )
```

```
(rule_O1 then (?x displayed-by text))
```

```
(rule_O2 if (?x must-be displayed) (?x is-a package))
```

```
(rule_O2 then (?x displayed-by table))
```

[그림 4] GUI 객체의 선택 관련 규칙 예

이 부분에서는 사용자의 정보 요청이 있을 때, 혹은 시스템으로부터 새로운 정보가 입력될 때, 상황에 따라 해당 정보를 표시하기 위해 가장 적절한 GUI 객체를 선택한다. 이 때 GUI 객체는 윈도우와 텍스트, 그래픽 또는 그것들의 조합 등이 선택된다. 이 부분을 구성하고 있는 규칙들의 예를 [그림 4]에 나타내었다.

3.3.4 GUI 객체의 생성 및 속성 지정

이전 단계에서 적절한 GUI 객체가 결정되면, 이 단계에서는 해당 GUI 객체들을 실제로 생성하고 속성값(attribute value)을 지정한다. 즉, 아직 정의되지 않은 객체를 새롭게 생성하거나, 이미 정의된 객체의 속성값을 갱신(update)시켜 준다. 이 때, 해당 객체는 상위 객체의 속성값을 상속받게 된다. 그러한 규칙을 예로 들면 다음의 [그림 5]와 같다.

〈의미〉

규칙 P1 : 제시되도록 선택된 정보가 비행 기지이고, 이를 텍스트 객체로 표시하도록 결정되면, 텍스트의 색깔을 빨간색으로 하고, 크기는 10으로 한다.

〈표현〉

```
(rule_P1 if (?x must-be displayed)
  (?x is-a air_base)
  (?x displayed-by text)
  (set 't1 (gensym)))
(rule_P1 then (t1 is-a text)
  (t1 content (get-information ?x))
  (t1 correspond-object ?x)
  (t1 color red) (t1 font-size 10))
```

[그림 5] GUI 객체의 생성 및 속성 지정 관련 규칙 예

3.3.5 GUI 객체의 배치 및 출력

GUI 객체의 배치 및 출력 단계에서는 이전 단계에서 생성된 GUI 객체들을 실제로 해당 윈도우로 출력하여 디스플레이 함과 동시에 화면상의 윈도우를 재배치한다. 예를 들면, 화면상에 새로운 윈도우를 생성시켜 출력할 때마다 화면상에 나타나 있는 윈도우의 개수가 일정한 기준을 초과하는지를 체크한다. 윈도우의 개수가 일정한 수 이상이 될 경우, 가장 하위에 위치한 윈도우가 제시하는 정보가 계속적으로 필요한 정보이면 이를 아이콘화하여 유지하고, 그렇지 않으면 소멸시킨다. 이렇게 함으로써 화면상에 지나치게 많은 윈도우가 복잡하게 디스플레이 되어 사용자가 정보를 신속하게 확인하기 어려워지는 것을 방지하고, 계속적으로 확인할 필요가 있는 정보인 경우에는 사용자가 아이콘화된 윈도우를 복구시킴으로써 언제든지 정보를 다시 확인할 수 있도록 한다. 만약, 윈도우가 아이콘화되어 있는 상태에서 디스플레이된 정보의 내용에 변화가 발생한 경우에는, 해당 윈도우를 스스로 복구시켜 사용자가 이를 확인할 수 있도록 한다. 이에 관한 규칙을 예로 들면 다음의 [그림 6]과 같다.

〈의미〉

규칙 L1 : 화면상의 윈도우의 개수가 5개보다 많고 가장 하위 윈도우의 정보가 필요한 것이면, 이를 아이콘화하여 보존하도록 한다.

〈표현〉

```
(rule_L1 if (> (get-windows-number) 5)
  ((get-bottom-window) is-necessary yes))
(rule_L1 then (minimize-window
  (get-bottom-window))
  (decrease-windows-number))
```

[그림 6] GUI 객체의 배치 및 출력 관련 규칙 예

3.4 제안한 구조에 의해 개발한 시스템의 특징

현재까지 이루어진 지능적 사용자 인터페이스에 관한 연구들은 주로 자연어의 인식[Nea91]이나 도움말 생성[Kas91] 혹은 사용자의 요구가 있을 때 그에 대해 상황에 따라 가장 적절한 방법으로 관련 정보를 제시[Tyl91][Are91][Ber96]하는 방법들에 대한 것들이 대부분이었다.

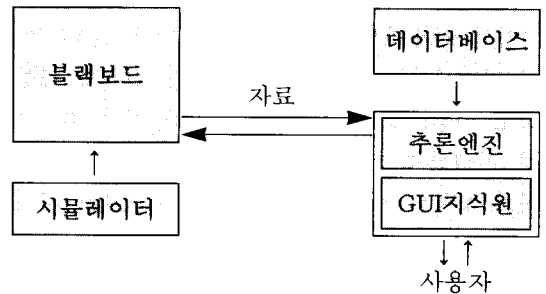
그러나, 이 논문에서 제안한 GUI 구조에서는 정보를 효율적으로 디스플레이하는 측면에 초점을 맞추었다. 군사 분야 시스템과 같이 사용자의 신속한 상황 파악과 결정 수립이 요구되는 시스템에 적합하도록, 상황에 따라 필요하다고 판단되는 정보를 적절히 선택하여 디스플레이하도록 하였다. 이렇게 함으로써 사용자의 추가적인 정보 요구 필요성이 최소화 될 수 있을 뿐만 아니라, 불필요한 정보까지 제공됨으로써 오히려 사용자의 판단에 혼란을 초래하는 일을 방지할 수 있다.

또한 이처럼 규칙 기반 기법에 의해 개발된 사용자 인터페이스가 지니는 일반적인 장점으로써 평가 후 수정 작업이 용이하게 이루어질 수 있다는 점을 들 수 있다. 시스템의 제작 후 테스트 과정을 거쳐 수정이 필요하다고 판단되면, 단지 문제점이 발견된 해당 부분의 규칙들과 그에 관련된 객체들만을 바꾸어 줌으로써 소프트웨어의 보수 작업이 신속하게 이루어질 수 있다.

4. 제안한 기법의 구현

제안한 구조에 의한 그래픽 사용자 인터페이스

를 구현하기 위해, 연세대학교 전자공학과 컴퓨터 응용 연구실에서 개발 중인 공중 작전 상황 판단 및 결정 지원을 위한 정보 융합 전문가 시스템인 IFS(Intelligent Fusion System)의 사용자 인터페이스 모듈을 개발하였다. IFS는 이러한 정보 융합에 필요한 기능인 다단계 추론 기능, 불확실성 처리 기능, 실시간 처리 기능등을 전반적으로 처리하기 쉬운 블록보드 구조를 기반으로 하여 개발된 시스템이다.



[그림 7] IFS의 사용자 인터페이스 구조

IFS의 사용자 인터페이스 구조를 [그림 7]에 나타내었다. 사용자 인터페이스 모듈에서는 시뮬레이터에서 발생된 가상의 데이터에 의해 단계적으로 융합이 수행된 결과를 블랙보드로부터 넘겨받고, 이것을 GUI에 관련된 추론을 통해 효과적인 방법으로 사용자에게 제시하게 된다. 여기에서 데이터베이스는 아군과 적군의 기지 규모, 보유 기종, 그리고 각 기종의 특성 등 변하지 않는 정보들을 저장하고 있는 부분이다.

4.1 적용 시나리오

시뮬레이터로부터 입력되는 데이터에 따라 정보 융합을 수행한 결과를 사용자에게 효율적으로 디스플레이하는 예를 보이기 위해 다음과 같은 내용을 갖는 시나리오를 설정하여 실험하였다.

시나리오 개요 : 적 전투기(MIG-25)가 의도적인 위장으로 고도를 낮추고 마치 전폭기인 것처럼 침투하다 고도를 높이며 전투기 본연의 임무를 위한 상태로 변하는 상황에서 빠른 시간 내에 상황을 판단하여 적절한 대응을 할 수 있도록 지원한다.

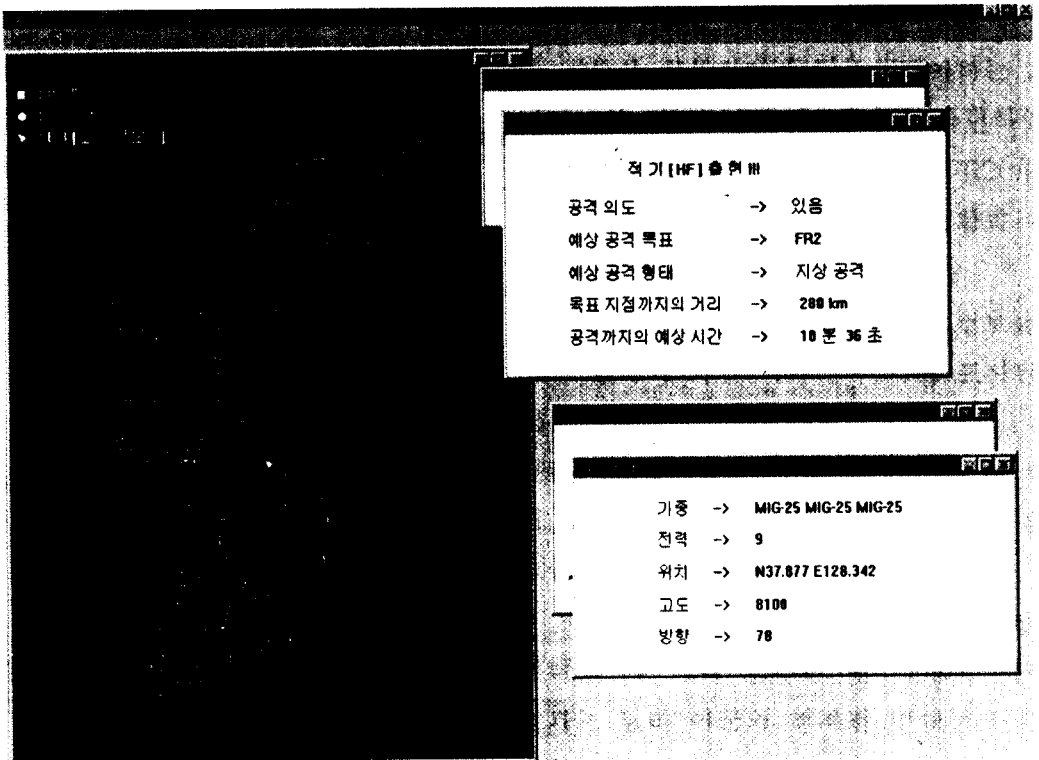
4.2 사용자 인터페이스의 동작

제안한 기법에 의한 실제적인 사용자 인터페이스의 개발에는 연세대학교 전자공학과 컴퓨터 응용 연구실에서 개발한 리스트 처리를 위한 인공 지능 언어인 NEO를 사용하였고, 이를 구현하여 실험한 결과, 시나리오의 각 단계별 GUI의

동작 중 주요 화면을 다음에 제시하였다.

[그림 8]은 적기가 출현한 상황에서 필요한 정보를 사용자에게 디스플레이하는 화면을 나타낸다. 화면 우측에 적기가 출현하였음과, 공격 의도가 있고 공격 목표는 FR2(아군 레이더 기지)로 예상된다는 정보 융합의 결과를 나타내고 있다.

이 때, 사용자가 적기에 관한 정보를 확인하기 위하여 화면 좌측의 적기로 표시된 이미지를 마우스로 클릭하면, 화면 우측 하단에 관련 정보를 텍스트 객체를 사용하여 나타내어 준다. 이 때 사용된 규칙의 예를 다음의 [그림 9]에 나타내었다.



[그림 8] 적기 출현 상황에서 적기에 관한 정보 제시

<의미>
 규칙S7 : 적기가 발견되고, 적기가 image로 표현된 경우에, 해당 image에 마우스의 왼쪽 버튼을 누르면, 일반적인 정보를 표시하게 한다.

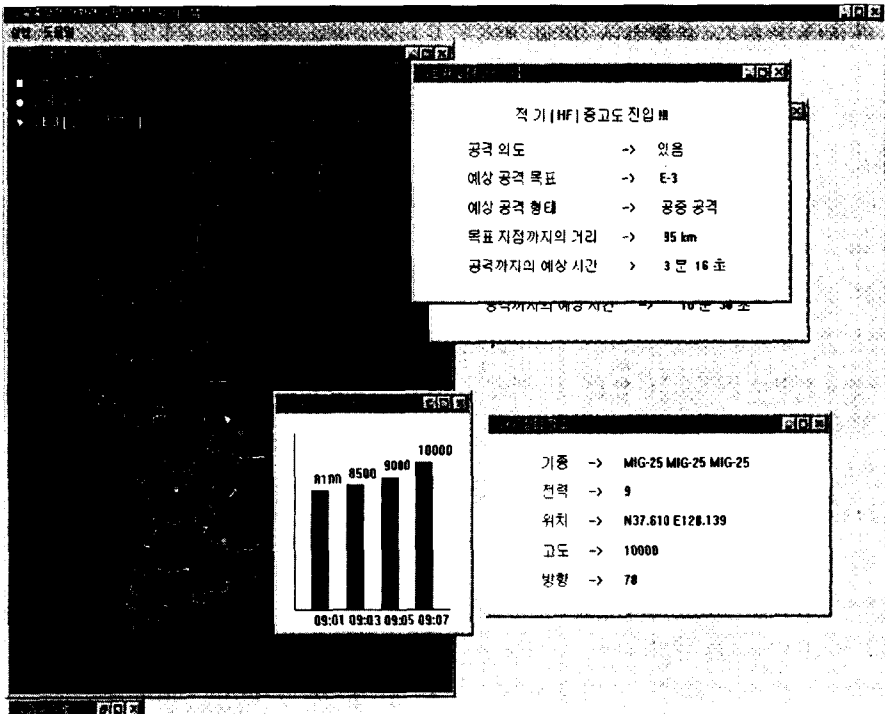
<표현>

```
(rule_S7 if (?x is detected)
  (?x allegiance enemy)
  (?y is-a image)
  (?y correspond-object ?x)
(rule1_S7 then
  (update-method ?y (:left-button-down)
    ((show-general-inform ?x))))
```

[그림 9] 적기 출현 상황에서 적기에 관한 정보 제시 관련 규칙 예

[그림 10]은 지금까지 적기가 지상의 레이더 기지 폭격 임무를 띤 것으로 예측되던 것과는 달리 아군의 전자전 편대군 E-3를 공격할 것으로 공격 목표가 변경됨에 따라, 우측 상단에 새로운 윈도우를 생성하여 적비행기 HF의 변경된 공격 목표와 관련된 정보들을 디스플레이하고 있다. 특히, 사용자가 HF 관련 정보를 다시 확인하기 위하여 좌측의 해당 이미지를 클릭하면, [그림 8]과는 다른 결과가 나타난다.

즉, 우측 하단에 새로운 윈도우를 생성하여 HF의 현재의 위치와 고도, 전력 등의 정보를 디스플레이함과 동시에, 적군의 공격 목표 변화 시에 특히 중요한 정보로 판단되는 시간에 따른 HF의 고도 변화 과정도 함께 보여주고 있다.



[그림 10] 공격 목표 변경 상황에서 적기에 관한 정보 제시

이 때, 시간에 따른 고도 변화는 사용자가 보다 쉽게 파악할 수 있도록 그래프 객체를 이용하여 나타내고 있다. 또한 화면상에 일정한 개수 이상의 윈도우가 나타나게 됨에 따라, 가장 하위 윈도우인 기상 정보 윈도우를 아이콘화하여 나타내었다. 이 때 사용된 규칙의 예를 [그림 11]에 나타내었다. 이것은 일반적으로 적기의 예상 공격 목표가 변경된 경우에는 해당 적기의 고도 변화 과정이 중요하게 작용한다는 전문가의 견해에 따른 것이다.

<의미>
 규칙 S9 : 적기의 예상 공격 목표가 변경되고, 적기가 image로 표현된 경우에, 해당 image에 마우스의 왼쪽 버튼을 누르면, 일반적인 정보와 더불어 고도 정보를 표시한다.

<표현>

```

(rule_S9 if (?x target changed)
    (?x allegiance enemy)
    (?y is-a image)
    (?y correspond-object ?x)
(rule_S9 then
    (update-method ?y (:left-button-down)
        ((show-general-inform ?x)
        (show-height ?x))))
    
```

[그림 11] 공격 목표 변경 상황에서 적기에 관한 정보 제시 관련 규칙 예

[그림 8]과 [그림 10]을 비교하면, 상황에 따라 주요 정보를 선택하여 제시해 줄뿐만 아니라, 동일한 사용자 입력에 대해서도 다른 출력 결과를 나타내는 지능적인 사용자 인터페이스를 보여준다. 즉, 객체에 대한 메소드를 실행 시간

중에 갱신할 수 있으므로, 상황에 따라 가장 적절한 형태의 정보 제시가 손쉽게 이루어질 수 있다. 이와 달리, 일반적인 형태의 사용자 인터페이스에서는 변화하는 상황에서도 필요하든 불필요하든 동일한 항목의 정보를 화면상에 제시함으로써 사용자의 불편을 초래할 수 있다.

5. 결 론

이 논문에서는 사용자의 신속한 판단을 지원하기 위하여, 상황에 따라 제시할 정보들을 선택하고 이를 적절한 GUI 객체를 통해 사용자에게 제시할 수 있도록 구성된 규칙 기반의 지능적 사용자 인터페이스 구조를 제안하였고, 제안한 구조에 의해 실제로 군사 분야 정보 융합 시스템의 그래픽 사용자 인터페이스를 개발하였다.

제안한 GUI 구조는 입력 해석, 제시할 정보의 선택, GUI 객체의 선택, GUI 객체의 생성 및 속성 지정, GUI 객체의 배치 및 출력 단계들로 구성되어 있고, 각각의 단계는 규칙에 의해 처리된다. 입력 해석 단계에서는 사용자로부터 들어온 입력을 일정한 심볼릭 데이터로 변환하거나 시스템으로부터 들어온 실세계 좌표 데이터를 스크린 상의 좌표 데이터로 변화하고, 제시할 정보의 선택 단계에서는 전문가로부터 획득한 지식에 의해 상황에 따라 제시할 정보들을 선택한다. GUI 객체의 선택 단계에서는 해당 정보를 사용자에게 효과적으로 제시하기 위한 GUI 객체를 선택한다. 객체의 생성 및 속성 지정 단계에서는 선택한 GUI 객체를 실제로 생성하고 적절한 속성을 지정하며, 객체의 배치 및 출력 단계에서는 이러한 객체들을 화면상의 적

절한 위치로 출력한다. 이 때 제시할 정보를 화면상에 시각화하여 나타내기 위한 GUI 객체들은 일정한 속성과 메소드로 구성되며, 계층 구조를 통하여 하위 객체는 상위 객체의 속성과 메소드를 상속받게 된다. 또한 객체간의 메시지 전달에 의해 메소드가 수행되므로 키보드나 마우스를 통한 사용자 입력뿐만 아니라 GUI 내에서 객체로 보내는 메시지를 통해서 인터페이스 기능에 영향을 미칠 수 있도록 하였다.

제안한 구조에 의해 공중 작전 수립을 위한 정보 융합 시스템의 사용자 인터페이스를 개발한 결과, 다양한 매체를 통해 많은 데이터가 입력되는 상황에서, 필요한 정보만을 선택한 후, 적절한 방법으로 제시해 줌으로써 사용자의 신속한 상황 판단과 결정 수립을 지원할 수 있었다.

그러나, 갑자기 발생한 긴급한 사건같은 것은 제안한 구조에 의하여 어떻게 표현할 것인지를 결정하도록 노력하는 것보다는 먼저 간단히 표현하는 것이 더 효과적일 수 있다. 따라서 이러한 경우에 해당되는 데이터들에 대한 처리를 위한 연구가 추가적으로 수행되어야 할 것이다. 또한 이 논문에서는 GUI 객체들을 이용하여 상황에 따라 화면을 구성함에 있어서 체계적이고 완벽한 사용자 모델에 바탕을 두었다기 보다는, 개발자 자신의 주관적 견해가 많이 포함되어 있다. 향후 체계적인 사용자 모델 획득과 다양한 매체를 통한 정보 제시 등에 관한 연구가 진행된다면 대공 작전 수립뿐만 아니라 다른 군 관련 분야 시스템의 사용자 인터페이스에도 적용될 수 있을 것이다.

참고문헌

- Design Using an Integrated Knowledge Base", Intelligent User Interfaces, ACM Press, 1991, pp.241-258.
- [Ber96] O. Bernsen, "A Reference Model for Output Information in Intelligent Multimedia Presentation Systems", ECAI'96, Budapest, August, 1996.
- [Chi91] D. N. Chin, "Intelligent Interfaces as Agents", Intelligent User Interfaces, ACM Press, 1991, pp.177-206.
- [Ebe94] Raye Eberts, User Interface Design, Prentice Hall, 1994, pp.6-17.
- [Hol91] J. Hollan, et al., "An Introduction to HITS: Human Interface Tool Suite", Intelligent User Interfaces, ACM Press, 1991, pp.293-337.
- [Kas91] R. Kass and T. Finin, "General User Modeling: a Facility to Support Intelligent Interaction", Intelligent User Interfaces, ACM Press, 1991, pp.111-128.
- [Mac91] Jock D. Mackinlay, "Search Architectures for the Automatic Design of Graphical Presentations", Intelligent User Interfaces, ACM Press, 1991, pp.281-291.
- [Nea91] J. G. Neal and S. C. Shapiro, "Intelligent Multi-Media Interface Technology", Intelligent User Interfaces, ACM Press, 1991, pp.11-41.
- [Ty191] Sherman W. Tyler, et al., "An Intelligent Interface Architecture for Adaptive Interaction", Intelligent User Interfaces, ACM Press, 1991, pp.85-109.
- [Are91] Yigal Arens, et al., "Presentation