

스크립트 매칭을 통한 상황평가

(Situation Assessment through Script Matching)

김 석 원* 김 진 형**

(Seok Won Kim) (Jin Hyung Kim)

요 약

상황평가란 임의의 상황에 대하여 계속적으로 발생하는 사건을 수집하여, 그 상황에 대한 해석을 하고, 다가올 사건을 예측하는 추론행위를 말한다. 상황은 스크립트 - 일련의 전형적인 사건들 - 를 사용하여 모형화 할 수 있으며, 그에 따라 상황평가 작업은 어떤 상황이 진행중인지를 검사하기 위해 보고된 사건을 모형스크립트에 매칭하는 스크립트 매칭문제로 볼 수 있다. 여기에서 사건은 가설을 구성하고, 구성된 가설을 확인하는 근거로 사용된다.

본 논문에서는 스크립트의 구조가 새로이 정의되고 적용영역에 크게 의존하지 않는 일반적인 스크립트 매칭방식이 제안되었다. 그리고 이 방식을 이용하여 스크립트 기반의 상황평가 도구 SMAT(Script Matcher)가 개발되었다. SMAT는 보고된 사건으로 가설을 구성하고, 가설에 근거하여 발생이 기대되는 사건을 예측하며, 계속하여 발견되는 사건을 기대사건과 매칭함으로써 그 가설을 확인한다.

스크립트 구조는 그 안에 지역변수를 정의할 수 있으며 하위의 스크립트를 다른 스크립트와 공유할 수 있도록 구성되어 있다. 한 스크립트의 가능성은 스크립트 구성요소와 그곳에 매칭된 사건이 일치할 가능성에 가중치를 곱한 값을 모든 구성요소에 대하여 평균한 값으로 정의되었다. 매칭과정은 계층적인 구조에 대하여 재귀적(recursive)으로 적용하므로, 여러 단계를 갖는 복잡한 가설을 관리하기가 용이하다. 결론적으로 SMAT는 스크립트 기반의 상황평가 전문가시스템을 구성하는데 있어서 편리한 도구의 기능을 제공해 준다.

ABSTRACT

Situation assessment is the name given to the activity of evaluating a given sequence of events to make a coherent and abstracted interpretation and predicting coming events based on the interpretation. A situation can be modelled by a script - a sequence of stereotypical events. In that case, situation assessment task can be formulated as script matching in that reported events are matched to model scripts to check whether the situation of the model script has occurred. Events are treated as a clue for invoking hypotheses as well as an evidence for confirming hypotheses. A hypothesis may be construct-

* 종신회원 . 한국과학기술원 전산학과

** 종신회원 . 한국과학기술원 전산학과 교수

접수일자 . 1989년 10월 13일

ed with lower level hypotheses.

In this thesis, a script structure is defined and a generic script matching methodology is proposed. Based on the methodology, a script-based situation assessment tool called SMAT (Script Matcher) is developed. SMAT creates hypotheses with reported events, generates expectations of the invoked hypotheses and confirms the hypotheses by matching coming events with the expectations

The script structure is capable to represent various situations by providing the facilities of defining local variables in a script and sharing the lower level scripts with other scripts. The likelihood of a script is defined formally as a weighted sum of matching probability of script components and reported events. The matching process, which is applied recursively to hierarchically structured hypotheses, has been shown suitable to manage complex hypotheses of multiple levels. In conclusion, SMAT provides a useful tool for construction of script-based situation assessment expert systems.

1. 서 론

전문가시스템이란 인간의 전문성이 중요한 역할을 하기 때문에 풀 수 있는 문제를 전문적 지식과 추론기능을 사용하여 해결할 수 있도록 구성된 지능적인 컴퓨터 프로그램을 말한다. 전문가시스템 기법을 이용하여 효과적으로 해결할 수 있는 문제의 한 형태로서 상황평가를 들 수 있다. 상황평가란 임의의 상황에 대하여, 관측된 자료를 토대로 해석을 하고 앞으로 일어날 일을 예측하는 행위를 말한다. 생물이나 화학 등과 같은 자연과학 분야에서 실험을 하게 될 때 관찰되는 현상을 하나의 사건으로 보면 결과를 확인하고 예측하는 과정은 상황평가의 문제이며, 군사분야에서 상대방의 동태를 감시하여 적군의 작전, 의도 등을 파악하는 것도 같은 문제로 볼 수 있다. 또한 지진과 같은 자연현상을 관찰하여 실체를 파악하고 다가올 현상을 예측하는 것도 이러한 문제유형에 포함된다. [HAR85]

상황평가에서 입력자료로 사용되는 사건은 성격이 불연속적이고, 불완전하며, 상황에 따라 상호 모순적일 수 있다. 또한 서로 다른 원인에 의해 발생한 사건이 복잡하게 섞여서 관찰되므로 더욱 어려운 문제가 된다. 사용될 수 있는 지식은 전문가가 예측할 수 있는 모든 발생가능한 상황에 대한 묘사가 주를 이루게 되며, 이것은 매우 복잡하고, 상황묘사에 사용되는 각 요소간에서 서로 의존적인 성격이 있으므로 명확하게 정의하기가 어렵다.

그러나, 일련의 발생가능한 상황을 보다 작은 규모의 사건을 사용하여 순차적으로 표현함으로써 전형적인 상황을 정의할 수 있으며, 이런 목적으로 개발된 것이 스

크립트(script)라는 지식표현방식이다.

스크립트는 전형적인 사건의 순차를 표현하는 지식구조이다. 사건의 순차라 함은 사건이나 행동 등이 서로 시간적인 관계를 가지고 있어서 임의의 순서로 발생할 수가 없다는 의미이다. 이것은 프레임기반 지식표현방식(frame-based knowledge representation scheme)의 일종으로 프레임의 기본적인 표현방법인 슬롯(slot)과 슬롯값을 이용하여 사건의 전형적인 발생순서를 기록한다. 이러한 지식표현은 최초로 자연언어처리 분야에서 문장내용의 이해를 위해 문맥을 잘 처리할 수 있는 지식구조로서 고안되었으며, 그 후 자연언어처리 분야와 상황평가를 목적으로 하는 전문가시스템 분야에서 지식표현방식으로 사용되고 있다. 이와 같이 스크립트가 미지의 대상에 대한 해석 혹은 예측의 문제에서 지식표현방식으로 적합한 이유는, 사람이 미지의 대상에 대한 해석을 할 때 규칙이나 논리에 의하기 보다는 미리 지식으로 기억되어 있는 전형적인 상황과의 비교에 의하여 수행하기 때문이다. 일반적으로 사람이 자신에게 닥친 어떤 상황을 분석할 때에도 관찰된 사건을 연역적이나 귀납적으로 추론해 가는 것이 아니라 이미 지식으로 구성되어 있는 전형적인 상황중에서 어떤 것과 가장 흡사한가를 계속하여 그 상황에 대한 해석을 하고, 이에 따라 가설을 설정한 후, 뒤이어 발생한 사건을 사용하여 설정된 가설을 확인하는 과정을 취한다. 이런 특징 때문에 스크립트는 발생하는 사건에 대해 문맥을 알아 내기가 용이하고 알아낸 문맥을 추론에 사용하는 것이 자연스럽게 된다. 스크립트라는 지식표현방식이 제안된 것도 자연언어처리 분야에서 문장내용의 이해를 위해 문맥을 잘 처리할 수 있는 구조를 필요로 했기 때문

이다. [BAR81, MIN75, SCH77]

상황평가에 관한 문제를 스크립트를 이용하여 해결한다면, 지식으로 구성된 전형적인 상황은 스크립트로 구성된 모형으로 표현되고, 발견된 사건과 가장 유사한 모형을 선택하여 가설을 세우고 확인하는 작업 등은 스크립트 매칭(matching) 문제가 되며, 이 매칭과정이 상황평가의 추론과정이 된다. 기존의 모든 시스템에서는 사용된 스크립트를 매칭하는 방법론이 미리 정해진 적용영역에 의존적인 방식이므로 일반성이 부족하다. 따라서 이미 개발된 것과 다른 영역의 문제에 대해 관련 지식을 스크립트로 표현하고, 스크립트 매칭에 의해 문제를 해결하려면, 적용영역에 의존적인 기존의 방식으로는 지식을 스크립트로 표현하기가 어렵고, 매칭방식도 적당치 않으며, 적용영역이 바뀔 때마다 매번 새로운 스크립트와 새로운 매칭방법을 개발해야 한다. 이것은 스크립트를 사용함에 있어 큰 제약점이 되며, 개발된 시스템의 능력을 크게 감소시키는 원인이 된다. [DEJ82, DYE83, LAS87, CHE85]

스크립트 매칭의 전략은 크게 나누어서 모형유도방식(model driven approach)과 사건유도방식(event driven approach)으로 구분할 수 있다. 모형 유도방식은 모형으로 부터, 즉 조건문의 만족이나 특정사건의 발생에 의해서 스크립트를 가정에 놓고 이어서 입력되는 사건이 가정된 스크립트로 설명될 수 있는가를 확인해가는 하향식(top-down) 전략이다. 사건 유도방식은 사건으로부터, 즉 입력되는 사건을 모아서 그 사건들을 설명할 수 있는 스크립트를 찾고, 다시 이것을 이용하여 상위의 스크립트를 찾아나가는 상향식(bottom-up) 전략이다.

모형유도방식은 전체적인 지식이나 정보 등을 많이 이용할 수 있고, 문맥과 사건에 의해 모형을 쉽게 결정할 수 있는 문제영역에서는 효율적으로 적용되어 상황을 적절하게 평가할 수 있다. 반면에 선택할 수 있는 모형의 후보자가 여럿 있게 되면 선택이 어려워지고, 잘못 이해할 가능성이 많으며, 처리속도가 발견되는 사건의 갯수보다 저장된 모형의 갯수에 더욱 민감하게 영향을 받는 단점이 있다. [SCH77, DEJ82, LAS85, LAS87]

사건유도방식은 가설구성과정 중 구성된 가설만으로 부분적 상황에 대한 설명을 할 수 있고, 수행속도는 사건의 갯수에 의존하는 특징을 갖는다. 반면에 개개의

사건을 중요시하게 되므로 전반적인 상황에 대한 설명이 잘 안되는 수가 있고, 사건에서부터 여러 가능성을 고려하면서 가설로 구성해 가므로 처리시간이 길어지는 단점이 있다[SCH82, DYE83].

2. 상/하향식 전략을 결합한 스크립트 매칭방식

2.1 문제영역

발생되는 일련의 사건을 이해하고, 이를 바탕으로 앞으로 일어날 사건을 이해하기 위해서는 사건에 대한 해석 및 예측이 필요하다. 해석(interpretation)은 수집된 자료를 이용하여 상황에 대한 적절한 설명을 구해내는 것을 말하고, 예측(prediction)은 주어진 상황에 대하여 자료를 관측, 수집, 분석하여 그 상황에서 앞으로 일어나게 될 현상에 대하여 추론하는 행위를 말한다. 상황평가(Situation Assessment)란 임의의 상황에 대하여 계속적으로 발생하는 사건을 수집하여, 그 상황에 대한 해석을 하고, 다가올 사건을 예측하는 추론행위를 말한다[WAT86].

상황평가 작업에서 입력자료로 사용되는 사건은 불연속적이고 불완전하다. 불연속적이란 것은 실제로 발생한 사건이 발견되지 않을 수도 있음을 의미하고, 불완전하다는 것은 하나의 사건으로부터 얻을 수 있는 정보에 오차가 있을 수 있음을 의미한다. 더구나 여러 상황이 동시에, 나란히 전개되어 이로 인해 발생한 사건이 서로 섞여서 발견되는 경우도 있을 수 있다. 그밖에도 발생한 사건이 서로 모순을 일으키는 경우도 있을 수 있으나, 이런 경우에는 발견된 사건에 대한 진위를 고려할 수 있는 비단조적인(non-monotonic) 추론을 행해야 하는데, 이것은 사실상 매우 어려운 문제로 알려져 있으므로 본 논문에서는 이런 경우는 제외하기로 한다.

일련의 발생가능한 상황을 제충적인 스크립트 구조로 정의하게 되면 상황평가 작업은 입력되는 사건을 가지고 정의된 모형과 가장 일치하는 구조를 찾는 스크립트 매칭(script matching)의 문제가 된다. 그러나 모든 지식이 스크립트 형태로 표현될 수는 없으므로 스크립트만을 지식표현방식으로 사용하는 전문가시스템을 구성하여 지식을 사건의 순차만으로 표현하게 되면 성격상 적당하지 않은 경우도 있다. 이런 취약점을 보완하기

위하여 본 논문에서 설계하고 구현한 시스템에서는 규칙기반 시스템이 전처리용으로 사용되었다.

2.2 스크립트의 정의

스크립트 지식베이스는 파일에 리스트 형태로 저장되며 시스템에서는 동작하기 전에 이것을 읽어들여 시스템의 내부적인 구조로 바꾸어 모형베이스(model base)를 구축한다. 모형베이스에는 각 스크립트의 정의가 있게 되며 스크립트 구성에 사용된 다른 스크립트를 종스크립트(subscript)라는 형태로 스크립트 정의에 포함시킨다. 스크립트를 정의하는 형식은 아래와 같으며, 설명을 돕기 위하여 <그림 1>에서 programming이란 가상적 스크립트를 정의하였다.

```
(defscript <스크립트 이름>
  <지역변수 리스트>
  <종스크립트 리스트>
  <선결조건 리스트>)
```

<스크립트 이름>은 하나의 심볼(symbol)이며, 정의될 스크립트의 종류를 나타낸다. <그림 1>에서 정의된 모형스크립트의 이름은 programming이다.

<지역변수 리스트>는 스크립트에서 사용되는 지역변수를 정의한 것이다. 스크립트의 지역변수는 스크립트의 주제, 즉 주인공과 그 외 스크립트가 전개되면서 참조하거나 사용하게 되는 여러 물체(object)를 나타내며, 각 지역변수가 갖는 값이 스크립트의 상태를 결정한다.

<종스크립트 리스트>는 스크립트를 구성하는 다른 모형스크립트의 이름과 두 스크립트간의 연관관계정의를 원소로 하는 리스트이다. 이 리스트의 원소는, 같은 이름의 스크립트가 여러번 사용될 수 있으므로 이름뿐 아니라 리스트내의 인덱스로 지칭할 수 있는데, 각 종스크립트는 정의된 순서에 따라 0에서 부터 차례로 정수 값을 인덱스로 갖게 된다. 하나의 원소를 정의하는 형식은 아래와 같다.

```
(<종스크립트 이름> <지역변수의 대응관계> <중요도>
<시간관계>)
```

<종스크립트 이름>은 다른 모형스크립트나 사건의 이름이 되며, 이 슬롯에 사용될 수 있는 스크립트의 종류

를 규정하게 된다.

<지역변수의 대응관계>는 스크립트가 모자관계(parent-child relationship)를 갖게 되는 경우에 스크립트 지역변수의 대응관계를 표현한다. <지역변수의 대응관계>를 표현하는 형식은 아래와 같다.

```
(<종스크립트 변수이름> <스크립트 변수이름> {키워드 값*})
```

여기서 키워드로는 :error가 있는데 이것은 두 변수의 값이 반드시 일치하지 않아도 된다는 것을 의미한다. 이 값은 가능성(likelihood)을 계산할 때 사용한다.

<중요도>는 스크립트를 구성하는데 있어서 종스크립트가 얼마나 큰 비중을 차지하는 가를 나타내는 값이다. 중요도가 :necessary인 경우에는 반드시 일어나야 되는 종스크립트를 나타내고 :optional인 경우에는 일어나지 않을 수도 있는 종스크립트를 나타내게 된다. 이 값은 0과 1사이의 실수값으로 대치되어 나중에 스크립트의 가능성을 계산할 때에 종스크립트의 가능성에 곱해지며, 모든 종스크립트에 대하여 이 값을 합한 것이 스크립트의 가능성이 된다. 가능성에 대해서는 3.4절에서 자세히 기술한다. 가능성의 계산방식에 따라 중요도가 가질 수 있는 값은 더욱 다양해 질 수 있다. 이 밖에 중요도가 :must이면 부모스크립트가 발생하기 위하여 반드시 발생하고 관측되는 자식스크립트를 나타내며, :must-not은 절대로 일어나서는 안되는 종스크립트를 나타낸다.

종스크립트간의 <시간적인 관계>에는 종스크립트의 발생순서가 정의된다. 종스크립트가 정의된 순서대로 발생하는 경우에는 특별히 명시할 필요가 없으나, 이 순서를 따르지 않을 경우에는 아래의 형식을 써서 명시한다.

```
(<시간관계 키워드> <종스크립트 지시어> {<시간 간격>})
```

<시간관계 키워드>는 after, before, equal 등이며 이것은 각각 종스크립트가 <종스크립트 지시어>가 나타내는 종스크립트보다 후에 발생하는가, 전에 발생하는가를 표현한다. <종스크립트 지시어>는 스크립트내의 다른 종스크립트를 나타내는데, 종스크립트의 이름을 쓰

거나, 같은 이름의 종스크립트가 2개 이상 있을 때에는 이름과 인덱스 쌍의 리스트로 표현할 수도 있다. <시간 간격>은 선택적으로 명시되는데 두 종스크립트간 발생 시간의 간격을 나타낸다. 예를 들어 한 종스크립트의 시간적인 관계가 (after (come-to 0) 100)이라고 정의되었으면 종스크립트의 제일 첫번째 원소 - 0번째 index-인 come-to 스크립트가 발생한 뒤 100 단위의 시간이 지난 후에 이 스크립트가 시작한다는 것을 의미한다.

<그림 1>의 예문에서 보면 지역변수의 정의뒤에 나오는 리스트가 종스크립트로서 come-to, use-computer, wait-until-empty, leave 등이다. 이들 각각은 다른 곳에서 스크립트로 정의되거나 개별적 사건으로 사용된다. come-to 스크립트에 있는 지역변수 person, place 등은 각각 programming 스크립트의 programmer, site에 해당한다. 또 come-to, use-computer, wait-until-empty 등은 programming이 일어나는 때 꼭 필요하지만(necessary), left는 선택적이다(optional), use-computer의 지역변수 duration을 보면 error 2라는 것이 있는데 이것은 앞서 설명했듯이 programming의 지역변수 duration과 use-computer의 지역변수 duration의 값이 반드시 일치하지 않아도 된다는 뜻이다.

<선택조건 리스트>는 모형스크립트가 가설로 선택될 때 만족해야 하는 조건이다. 이것은 하나의 술어(predicate)로서, and, or, not 등의 연산자를 사용해서 표현된다. 모형선택의 제한을 사용자에게 제공하는 이유는 전체적인 상태 및 문맥 등과 무관한 가설 설정을 제한하여 시스템의 성능을 향상시키기 위한 것이다. <그림 1>의 (가)에서 programming이란 스크립트는 프로그래머가 프로그래밍 숙제가 있거나(have-programming-homework), 프로그래밍을 하고 싶을 때만(want-to-program) 선택된다. 시스템에서는 모형선택시에 조건에 나타난 지역변수들 실제 지정된 값으로 대치하여 규칙기반시스템에서 계산하도록 한다.

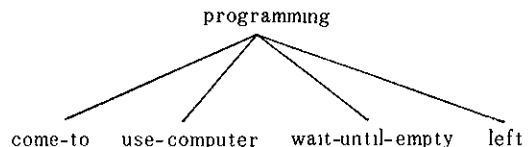
이러한 형식으로 정의된 모형스크립트가 모여서 모형베이스(model base)를 이루게 된다. 각 모형스크립트 사이의 관계는 <그림 1>의 (나)와 같이 도식적으로 간단히 표현할 수 있으며, 자세한 표현은 (다)에 나타나 있다. (나)에서는 programming이란 스크립트가 4개의 스크립트로만 구성된다는 것을 표현하고, (다)에서는 스크립트 정의가 구체적으로 어떻게 구성되어 모형으로

저장되는가를 보여준다.

(다)에서 볼 수 있듯이 각종 스크립트는 독립된 구조로 구성되어 programming 스크립트의 subscript 항목에 있는 해당 인덱스에 속하게 되는데, 종스크립트의 선행자(precedences)와 비선행자(non-precedences)는 이 구조를 구축하는 과정에서, 즉 화일에 정의된 모형을 읽어들이며 모형베이스를 생성할 때에 제산된다. 이것은 모

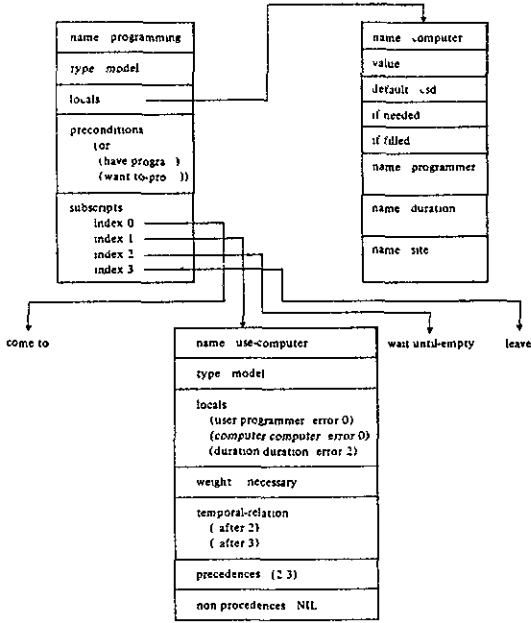
```
(defscript programming
  (programmer duration
   (computer :default csd)
   (site default cskast))
  ((come-to
    ((person programmer)
     (place site))
    (necessary)
    ()))
  (use-computer
   ((user programmer)
    (computer computer)
    (duration duration :error 2))
   ('necessary)
   (('after wait-until-empty)
    (:after come-to)))
  (wait-until-empty
   ((computer computer))
   (:necessary)
   ()))
  (leave
   ((place site)
    (person programmer))
   (:optional)
   (( after use-computer 100))))
  ((or
   (have-programming-homework programmer)
   (want-to-program programmer))))
```

(가) 예제 스크립트 정의



(나) (가)의 도식적 표현

(그림 1) 스크립트 정의의 실례



(다) (가)의 구체적 구조

(그림 1) 스크립트 정의의 실례(계속)

형에 정의된 종스크립트간의 시간적 관계에서 예외적 경우를 모아놓아 나중에 쉽게 사용할 수 있도록 한 것이다. 선행자는 이 종스크립트보다 큰 인덱스를 갖는 종스크립트중에 이것보다 먼저 발생해야 하는 모든 종스크립트의 인덱스로, 발생시간 관계에 의해 스크립트의 가능성을 따질 때 사용한다. 비선행자는 선행자와는 반대로 이 종스크립트의 인덱스보다 작으면서 나중에 발생해야 하는 모든 종스크립트의 인덱스이다. 따라서 선행자와 비선행자는 종스크립트 사이에서 서로 상대적 관계에 있게 되는데, <그림 1>의 (다)에서는 use-computer의 선행자에 2와 3이 있으므로, 인덱스가 2인 wait-until-empty 종스크립트와 인덱스 3인 left 종스크립트의 비선행자에는 use-computer의 인덱스 1이 속해 있어야 한다.

이런 식으로 모형이 정의되고 나면 모형베이스는 격자구조를 이루게 된다. 이 격자의 최하단 노드는 사건이 되고 나머지 노드는 모형스크립트이며 종스크립트는 두 모형을 연결해 주는 링크로 표현된다. 사건을 수집하여 상황을 평가한다는 것은 수집된 사건을 이 격자구조의 모형에 적절히 맞추어서 가장 그럴듯한 트리인스턴스를 구성하는 것이 된다. 매칭의 결과가 격자가 아

닌 트리구조란 말은 하나의 가정을 내포하고 있다. 즉 하나의 사건 혹은 스크립트는 오직 하나의 부모스크립트에만 속한다는 것이다.

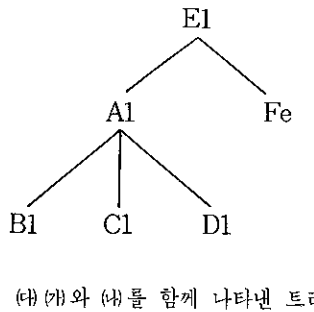
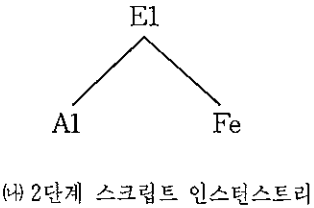
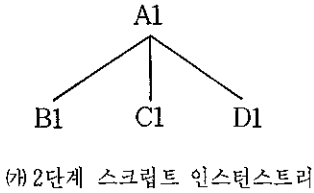
2.3 복합적 매칭방식

스크립트 매칭은 주어진 사건으로 스크립트인스턴스 트리를 구성하는 것이다. 하나의 스크립트인스턴스 트리는 그것에 속한 모든 사건들에 대한 해석이며, 트리의 루트가 가설을 대표한다. 모형유도방식에서는 전체적인 문맥과 발견된 사건에 가장 잘 매칭하는 모형을 선정하여 트리인스턴스를 구성하고, 이 트리인스턴스에 의해 앞으로 발생할 사건을 기대하고, 계속해서 발견되는 사건을 기대와 비교하여 가설을 확인함으로써 동작하며, 사건 유도방식에서는 입력사건을 사용하여 말단에서부터 점차적으로 복잡한 구조의 트리를 구성해서 전체구조를 완성한다.

본 논문에서 사용하는 매칭방식은 모형유도방식과 사건유도방식을 결합한 복합적 매칭방식이다. 여기서 복합적이란 말은 모형유도방식에서와 같이 입력된 사건을 사용하여 모형을 가정하고 사건을 기대하면서, 한편으로는 사건유도방식처럼 가설의 구성에 있어서는 상황식 전략을 사용한다는 뜻이다. 즉 모형이 한 단계 위까지만 가정되어 가설로 설정되며, 가설은 계속하여 발생하는 사건으로 확인하면서, 설정된 가설 자체가 다른 가설을 설정하는 자료로 사용된다. 이 방식에서 트리 인스턴스는 <그림 2>의 (가)와 같은 2단계 트리가 여럿 모여서 이루어진다. (가)에서 A1은 B1, C1, D1을 이용해서 구성된 스크립트인스턴스이고, B1, C1, D1은 사건이거나 다른 스크립트인스턴스이다. 마찬가지로 A1을 자식으로 취하는 (나)와 같은 스크립트인스턴스가 구성되었다고 하면 결국 E1은 (다)와 같은 구조를 갖게 된다. 이때 E1을 구성하는 과정에서 A1은 하나의 사건으로 취급되는 것이고, E1이 가설로 구성이 된 것은 A1 스크립트가 활성화되어서 E1의 가능성이 한계치를 넘어섰기 때문이며, A1에 속해있는 B1, C1, D1을 고려한 것은 아니다. 즉 한 스크립트인스턴스는 그것이 속한 부모에게는 사건으로 취급되면서, 자식에게는 가설스크립트로 취급받는다. 이 방식을 계속 적용하게 되면 결국 스크립트인스턴스 트리의 구성은 2단계 스크립트인스턴스 트리를 반복적으로 구성하는 것이 된다. 이

러한 스크립트인스턴스 트리는 가능성이 한계치를 넘는 모든 가설에 대해서 구성된다.

복합적 매칭방식에는 크게 두가지의 제어흐름이 있는데, 첫째는 사건에서부터 루트쪽으로 가설을 구성해가는 가설구성과정이고, 둘째는 설정된 가설이 틀렸을 때 그것에 속한 자식을 기각(reject)시키는 기각과정이다.



(그림 2) 스크립트 인스턴스 트리의 반복적 관계

가설구성과정에서는 몇개의 가설 스크립트인스턴스를 모아서 이것을 자식으로 삼는 새로운 가설을 설정하게 된다. 이때 가설을 설정하는 기준은 어느 것의 가능성이 한계치보다 높은가, 설정조건은 만족되는가 등이 사용된다. 설정된 가설은 이 가설스크립트를 자식으로 사용할 수 있으면서 위의 가설설정 기준을 만족하는 다른 가설이 있거나, 모형으로부터 또 다른 스크립트인스턴스를 구성할 수 있는 매칭이 발견되면 그 모형에 따라 가설구성을 계속하고, 그 밖의 경우에는 가설구성을 끝낸다.

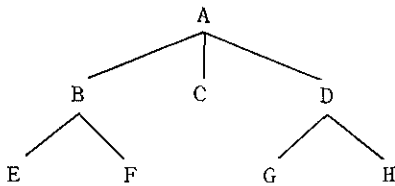
기각과정에서는 설정된 가설을 제거하며, 기각된 가

설스크립트의 자식스크립트도 가능성을 계산하여 한계치 이하이면 계속 기각하여야 한다. 가설이 기각되는 경우는 루트스크립트인스턴스가 예상한 사건이 발생하지 않았기 때문에 한계치 이하의 가능성을 갖게 되어 더 이상 가설로서 존재할 수 없는 상태가 된 경우, 반드시 일어나야 할 사건이 발생하지 않은 경우, 일어나서는 안되는 사건이 발생한 경우 등이 있다. 이 때 루트는 제거되고, 여기에 속했던 자식은 자식 자체가 가설이 될 수 있는가를 가능성에 의해 다시 확인한다. 만약 가설로 존재할 수 있으면 자식 스크립트인스턴스를 루트로 하는 스크립트인스턴스 트리가 가설로 등록되며, 가능성이 낮아서 가설이 될 수 없으면 기각시켜야 하므로 자식 스크립트인스턴스에 대하여 위의 과정을 반복한다.

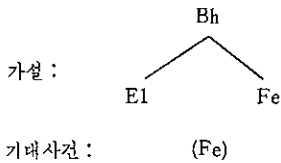
〈그림 3〉은 사건의 발생에 따른 시스템의 상태변화를 보여준다. 영어 대문자만 사용된 것은 모형을 나타내고, 대문자 다음에 소문자나 숫자가 사용된 것은 h가 가설인스턴스, e가 기대사건인스턴스를 나타내며, 숫자는 사건인스턴스를 나타낸다. 예를 들어 〈그림 3〉의 (가)와 같은 모형이 있다고 하자. E1이라는 사건의 발생으로 인해 B1 스크립트는 가설 Bh로 구성되고 빈 슬롯인 Fe는 기대사건으로 등록된다. 이 시점에서 시스템의 상태는 〈그림 3〉의 (나)와 같다. 이때 C1이라는 사건이 발견되면 기대사건에 C라는 종류가 없으므로 다른 가설과의 결합을 고려하게 된다. 현재 가설로 설정된 Bh와 방금 들어온 C1을 모두 설명할 수 있는 스크립트가 A이므로 Ah라는 가설 인스턴스가 구성되며 시스템의 상태는 〈그림 3〉의 (다)와 같이 된다. Ah의 남은 슬롯인 Dh는 Fe와는 달리 기대사건에 등록되지 않는데, 이는 D가 스크립트이기 때문이다. 즉, 자식스크립트가 사건이 아니라 스크립트이면 직접 기대사건으로 처리하는 것이 아니라 일단 비워두었다가 계속해서 발견되는 사건에 의해 가설로 설정되면 그때 비로소 상위가설의 일부로 포함된다. 모형에 의하면 Dh는 사건이 아니라 스크립트이므로 종류가 G나 H인 사건이 발견되어 Dh라는 가설이 구성될 때까지 비어있게 된다.

복합적 매칭방식의 가장 큰 특징이자 장점은 적용영역에 관계없이 사용될 수 있는 일반적인 매칭방식이란 점이다. 앞서 설명한 모형유도나 사건유도 등의 스크립트 매칭방식은 미리 적용영역이 정해진 상태에서 알고리즘이 개발되었기 때문에 일반성이 부족했었다. 모형

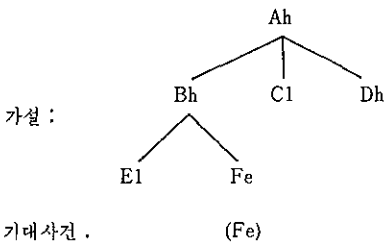
유도방식이나 사전유도방식과 같은 극단적 방식만을 사용했던 것도 시스템설계에 적용영역이 모형선택을 용이하게 하거나, 사건 하나하나가 중요한 의미를 갖는 경우였기 때문에 그에 알맞는 매칭방식이 개발된 것이다. 그러나 스크립트를 사용한 추론의 적용범위를 넓히기 위해서는 적용영역에 의존하는 가정을 완화한, 보다 일반적인 매칭방식이 필요하다. 복합적 매칭방식은 모형유도방식에서의와 같이 전체적인 가능성이 큰 가설을 발견할 수 있으면서, 사전유도방식에서처럼 사건의 발생빈도에 의해 성능이 결정되고, 부분적인 가설이 독립적으로 유지될 수 있는 장점을 갖게 된다. 이 복합적 매칭방식은 두 방식의 장점을 잃지 않는 가운데서 일반화를 추가했다는 점에서 의의를 갖는다고 볼 수 있다.



(가) 모델 스크립트



(나) E1이 발생한 후의 시스템 상태



(다) C1이 발생한 후의 시스템 상태

(그림 3) 사건 발생에 따른 시스템의 상태 변화

복합적 매칭방식을 취할 때 가장 문제가 되는 것은 가설이 잘못 구성되었을 경우이다 이런 경우에는 가설

에서 예측한 사건이 발생되지 않기 때문에 시간이 지남에 따라 점차적으로 가설의 가능성이 떨어지게 된다. 이러한 부식기법(decay mechanism)에 의하여 가능성이 한계치 이하로 감소한 가설이 생기면 가설을 기각하게 된다 가설을 기각하는 것은 구성했던 순서의 역으로 가장 상위의 가설부터 기각해야 하고, 이에 속해 있던 자식스크립트는 앞서 설명한 바와 같이 계속 기각되거나 독자적인 가설로 남아있게 된다

3. SMAT(Script Matcher) 시스템 구현

3.1 스크립트의 내부적 표현

시스템에서 모형으로 정의된 스크립트를 읽어 들이면서 시스템의 내부적인 모형스크립트 구조로 변환한다. 사건이 입력됨에 따라 모형스크립트를 참조하여 인스턴스를 만들게 되며, 인스턴스간에 트리구조가 관계지어지고, 이렇게 구성된 스크립트인스턴스 트리꼴써 상황을 나타내는 가설을 구성하게 된다. 시스템에서 사용하는 인스턴스는 크게 나누어서 스크립트, 종스크립트, 사건 등 3가지 종류로 나뉘게 된다. 서로 다른 스크립트인스턴스간의 관계란 지역변수 대응관계와 시간관계가 포함된다. 지역변수 대응관계는 스크립트간의 지역변수 이름의 일치름 나타내고, 시간관계란 모형에서 명시된 시간관계와 실제 인스턴스 발생시간의 일치름 나타낸다. 두 스크립트인스턴스가 관계를 갖게 되는 경우는 모자관계로 설정된 경우와 동일 스크립트인스턴스의 자식으로 속하게 되어 서로 시간적인 제한을 가하는 경우가 있다. 두 스크립트인스턴스가 모자관계를 맺게 될 때 그 사이에 종스크립트를 생성하여 관계를 표현하게 되는데, 그 이유는 스크립트의 인스턴스로서 가지고 있는 정보와 다른 스크립트인스턴스의 자식으로 속해 있을 때 가지고 있는 정보를 구분하여 매칭과정을 단순화하기 위한 것이다.

사건은 가장 기본적인 구조로서 관측된 각각의 사건에 해당한다. 구성속성(attribute)으로는 이름(name), 지역변수(locals), 발생시간(time-of-event), 부모스크립트(parent), 종스크립트로서의 인덱스(index-as-subscript) 등이 있는데, 이름은 사건의 종류를 의미하고, 지역변수는 사건을 구성하는 변수로서 각 지역변수 값이 실제

로 그 사건의 문맥을 결정한다. 발생시간은 사건이 발생한 시간을 의미한다. 부모스크립트, 종스크립트로서의 인덱스 등의 항목은 다른 스크립트의 자식으로 사용될 때 부모스크립트에 대한 포인터와 부모의 어느 종스크립트로 사용되었는가를 나타내는 인덱스를 저장한다. 사건은 가능성이 1인 스크립트라고 볼 수 있으며, 모형베이스에서 격자구조의 가장 바닥에 정의된 스크립트에 해당한다. 사건을 스크립트와 독립된 구조로 정의한 것은, 발생한 사건을 사용하여 시스템에서 사용할 수 있는 구조로 변환함에 있어서 스크립트인스턴스가 가지고 있는 종스크립트나 선결조건 등의 정보가 필요하지 않고, 또 사건이 스크립트인스턴스 트리의 말단에 속하게 되면 상위 스크립트인스턴스와의 관계를 나타내는 정보는 그 사이에 있는 종스크립트가 가지고 있기 때문이다. <그림 4>의 (가)는 사건의 실례이고, (나)는 (가)의 사건이 시스템 내부에서 어떤 구조를 갖는가를 보여준다.

스크립트인스턴스는 이름, 유형, 지역변수, 종스크립트, 시작시간, 선결조건, 가능성, 가능성계산 시간, 부모스크립트, 종스크립트로서의 인덱스 등의 많은 정보를 가지고 있다. 이름은 스크립트의 종류를 나타내고, 유형은 모형과 구분하여 인스턴스임을 나타낸다. 지역변수는 스크립트의 문맥을 결정하는데 종스크립트가 채워짐에 따라 값을 지정받는다. 스크립트도 모형스크립트와 마찬가지로 여러개의 다른 스크립트나 사건이 모여서 구성되므로 이 사이에 인터페이스 역할을 하는 종스크립트가 사용된다. 스크립트를 관계지어 주는 종스크립트는 한 스크립트 내의 다른 종스크립트와 모형에서 정의된 발생순서의 시간관계를 만족해야 한다. 시작시간은 사건 또는 종스크립트에 의해 추정된 스크립트 시작시간이다. 선결조건은 스크립트가 발생하기 위한 필요조건이다. 이것은 전처리용으로 사용되는 규칙 기반시스템의 작업기억장소(working memory)에 의해서 만족여부가 결정되며, 시스템이 동작하기 전에 전체적인 상황을 규칙기반시스템에 명언(assert)함으로써 조사항 모형의 갯수를 줄일 수 있다. 부모스크립트와 종스크립트로서의 인덱스는 사건의 경우와 의미가 같다.

예를 들어 <그림 1>과 같은 모형이 정의되어 있고, <그림 4>의 (가)와 같이 종류가 come-to인 사건이 발생하여 (나)와 같이 CT1이라는 인스턴스로 구성되었다고 하자. 만약 매칭의 결과로 programming이 가설로

설정되면, 시스템 내부적인 구조는 <그림 4>의 (나)와 같게 된다. 여기에서 P1은 설정된 가설스크립트이고, SUB1은 종스크립트이며, CT1은 사건이다. P1을 보면 지역변수는 자식스크립트인 CT1에 의해 결정되어 있고, 선결조건은 모형스크립트와 같다. P1은 스크립트인스턴스 트리의 최상위의 스크립트이므로 부모스크립트와 종스크립트로서의 인덱스가 NIL이며, CT1의 경우에는 부모스크립트가 P1이 되고, 종스크립트로서의 인덱스는 CT1이 P1의 0번째 종스크립트에 해당하므로 0이 된다. P1의 시작시간이 (before 100)인 것은 현재 종스크립트중에 가장 먼저 발견된 것이 CT1이고 CT1의 시작시간이 100이므로 P1은 적어도 100이란 단위시간 이전에 시작했음을 의미한다. 또한 P1의 가능성은 CT1이 입력되었을 때 계산되었으므로 가능성계산 시간이 100이 된다.

종 스크립트의 구성 속성으로는 이름, 유형, 중요도 (weight), 부모스크립트와 자식스크립트 사이의 지역변수 대응관계(mapping), 선행자(precedences), 비선행자(non-precedences), 부모스크립트, 인덱스, 가능성, 가능성계산 시간, 시작시간, 자체스크립트 등이 있다. 이

(EVENT

```
name come-to
locals ((person swkim) (place 2nd-floor))
time 100)
```

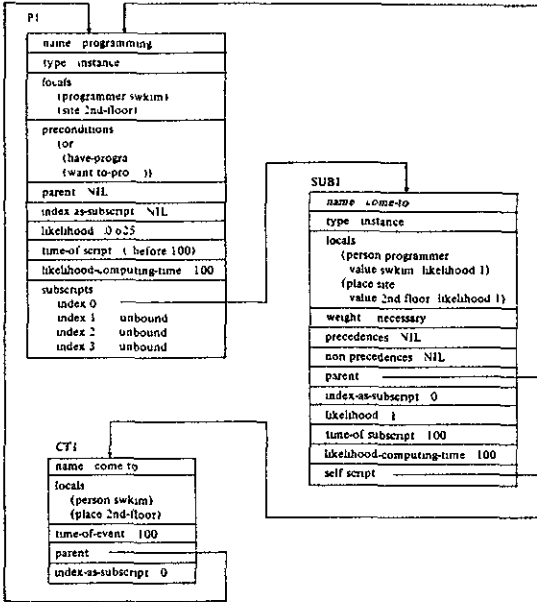
(가) 입력되는 사건의 예

CT1

name	come-to
locals	(person swkim) (place 2nd-floor)
time-of-event	100
parent	NIL
index-as-subscript	NIL

(나) (가)의 시스템 내부적 구조

(그림 4) 스크립트 인스턴스의 예



(다) 인스턴스간의 부모 자식관계의 예

(그림 4) 스크립트 인스턴스의 예 (계속)

중 이름, 유형, 부모스크립트, 인덱스, 가능성, 가능성 계산 시간, 시작시간 등은 스크립트의 경우와 같다. 지역변수의 대응관계는 모형에서 정의된 지역변수 대응관계를 통해 알 수 있으며, 여기에 value 키워드로 해당 자식스크립트의 변수값을 표현하고, likelihood라는 키워드를 써서 실제 인스턴스 지역변수사이의 가능성이 표현된다. 자식스크립트가 부모스크립트에 얼마나 중요한가는 중요도를 통해서 표현되며, 종스크립트에 해당하는 자식스크립트는 자책스크립트가 나타난다. 선행자, 비선행자는 모형에서 얻어진 것으로 다른 종스크립트와의 시간적 관계를 나타낸다.

〈그림 4〉에서 보는 바와 같이 CT1과 P1 사이의 지역변수의 대응관계나 가능성 등의 모든 대응정보는 SUB1이라는 종스크립트인스턴스가 포함하고 있게 된다. 따라서 P1이 가설로 존재할 때에는 P1은 SUB1을 참조하거나, SUB1의 자체스크립트 항목을 사용하여 CT1을 참조함으로써 자식에 대한 접근을 할 수 있다. 반면에 CT1에서 부모를 찾으려 할 경우에는 부모스크립트를 사용하여 P1을 찾을 수 있고, 종스크립트, 즉 SUB1을 참조할 때에는 CT1의 종스크립트로서의 인덱스 항목이 0이므로 P1의 종스크립트 중에서 0번째 것

을 찾으면 참조할 수 있다.

3.2 시스템의 구조

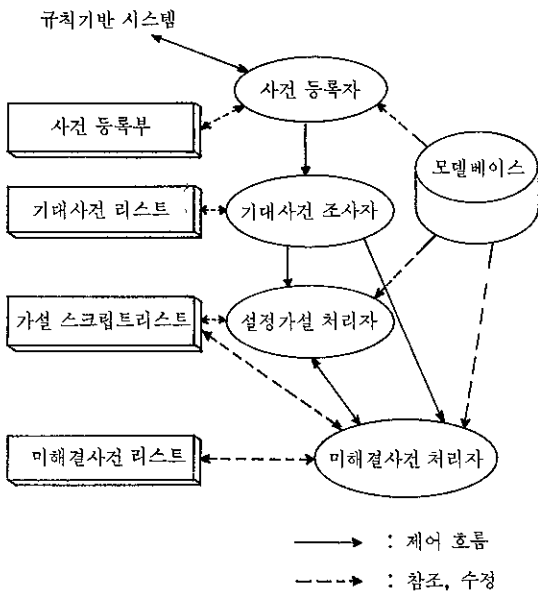
SMAT 시스템은 4개의 처리부분과 5개의 자료리스트, 모형베이스로 구성된다. 〈그림 5〉의 (가)는 SMAT 시스템의 구조를 나타내고, (나)는 사용자 인터페이스와의 관계를 나타낸다. 여기서 자료리스트는 (가)에서의 기대사건 리스트, 가설스크립트 리스트, 미해결사건 리스트 등이다.

사건등록부는 시스템에 입력된 모든 사건을 발견시간과 함께 기록해 놓은 리스트이다. 기대사건(expectation)은 가설스크립트에 의해 앞으로 일어나리라고 기대되는 사건이다. 가설스크립트(hypothetical script)는 수집된 사건을 사용해서 구성되어 현재 시점에서 조사 중인 스크립트이다. 미해결사건(unresolved event)은 수집된 사건중에서 가설스크립트의 구성에 아직 사용되지 않은 사건을 의미한다. 그리고 기대사건, 가설스크립트, 미해결사건 등을 모아놓은 것이 각각 기대사건 리스트, 가설스크립트 리스트, 미해결사건 리스트이다.

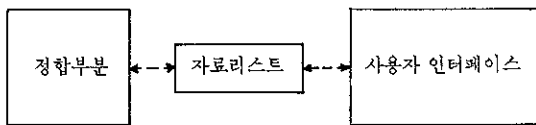
사건등록자는 입력되는 사건을 시스템에서 사용하는 자료구조로 재구성하며 사건등록부에 기록한다. 시스템 구성도에 나타나 있는 규칙기반시스템은 SMAT 시스템의 일부는 아니지만 선택적(optional)으로 연결시킬 수 있는데, 규칙기반시스템이 사용되는 경우에는 스크립트로 처리되기 어려운 추론부분을 규칙에 의해 먼저 처리하도록 한다. SMAT와 규칙기반시스템과의 인터페이스는 사건등록자가 맡게 되는데 입력된 사건을 규칙기반시스템에 넘겨 주어 처리한 후에 기대사건 조사자로 사건을 넘겨준다.

기대사건 조사자는 수집된 사건을 기대사건 리스트와 비교한다. 여기에서 기대하던 사건과 입력된 사건이 일치하면 기대사건은 기대사건 리스트에서 삭제되고 사건은 해당하는 가설스크립트에 속하게 된다. 새로운 사건이 가설스크립트에 속하게 됨에 따라 가설스크립트가 속해 있는 인스턴스트리는 아직 미지정이던 지역변수가 값을 가지게 되거나 혹은 스크립트 가능성의 변동 등이 일어나게 된다. 이러한 변동은 인스턴스트리의 말단에서부터 위로 파급되게 되는데 이것을 상향전파(upward propagation)라고 부른다. 기대사건 리스트에 적당한

기대사건이 없는 경우에는 입력된 사건을 미해결사건 리스트에 첨가하게 된다.



(가) SMAT 시스템의 구성도



(나) 사용자 인터페이스와의 관계

(그림 5) SMAT 시스템의 구성도

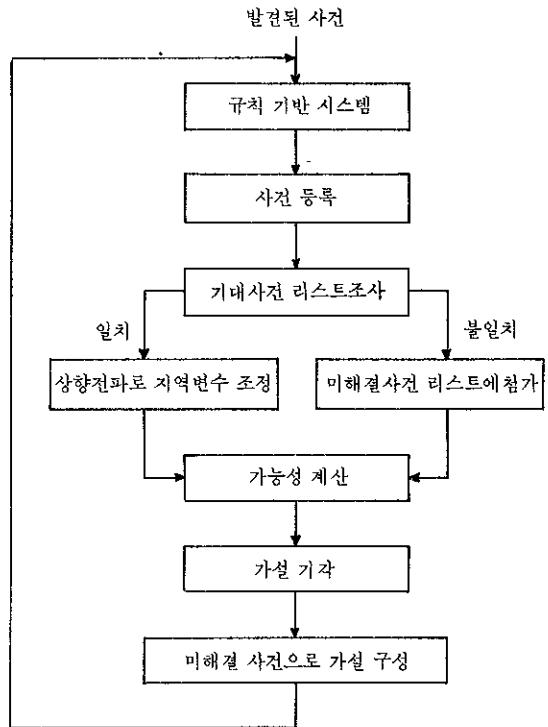
설정가설 처리자는 기대사건 조사자의 처리가 완료된 후에 시작된다. 가설스크립트의 가능성은 시간에 따라 변화한다. 일단 가설스크립트로 설정되고 기대사건이 구성된 후에 기대하던 사건에 발생하지 않으면 가능성은 점점 작아지고, 기대하던 사건이 발생하면 가능성은 증가한다. 따라서 가설스크립트 중에서 가능성이 낮은 것은 기각(reject)시킨다.

미해결사건 처리자는 가설에 속해 있지 않은 사건을 관리한다. 사건은 일단 기대사건과 비교하여 매칭되는 기대사건이 없는 경우에 미해결사건이 된다. 미해결사건은 계속 모아지다가 가설을 구성할 수 있으면 미해결사건 처리자에 의해서 가설로 구성되어 가설스크립트 리스트에 첨가된다.

3.3 제어흐름

SMAT 시스템의 제어흐름은 <그림 6>과 같다. 입력된 사건은 사건등록자에 의해 규척기반시스템과 사건등록을 거쳐서 기대사건 조사자로 보내진다. 기대사건 조사자는 기대사건 리스트를 조사하여 사건과 일치하는 기대사건을 찾는다. 일치하는 기대사건이 있으면 상황전파를 수행하고, 없으면 사건을 미해결사건 리스트에 첨가한다.

기대사건 조사자의 수행이 끝나면 설정가설 처리자의 수행이 시작된다. 설정가설 처리자가 제일 먼저 하는 일은 가설스크립트의 가능성을 현재 시간을 기준으로 다시 계산하는 것이다. 그 결과로 가설스크립트의 가능성이 최저한계(lower bound)이하로 떨어지면 그 가설은 기각당하게 된다. 기각은 기각당한 스크립트가 다른 가설에 소속해 있는가를 조사하여 만약 다른 가설에 속해 있으면 수행을 끝마치게 되고, 다른 가설에 속하지 않으면 계속적으로 하위의 스크립트로 파급한다. 기각은 스크립트, 종스크립트, 사건에 대해서 각각 다른 작업을 한다.



(그림 6) SMAT의 제어흐름

스크립트가 기각당하면 가능성을 찾아본다. 가능성이 최저한계 이하이면 그것에 속한 모든 종스크립트를 기각시킨다. 종스크립트가 기각당하게 되면 자체스크립트(self script)에 기각명령을 내린다.

사건이 기각당하면 이것은 더 이상 가설스크립트의 자식으로 남아 있을 수 없다는 것이므로 미해결사건 리스트에 포함시킨다.

설정가설 처리자의 처리가 끝나고 나면 미해결사건을 조사하여 가설스크립트를 구성해야 한다. 미해결사건을 스크립트로 구성하는 단계가 되면 각각의 리스트에서 개별적으로 가설스크립트를 구성할 수 있는 후보자(candidates)에 대한 처리가 이루어지고, 가능성이 상승한계(upper bound)를 넘는 후보자를 고른다. 선택의 기준은 가능성이고 후보자가 복수이면 각각을 가설스크립트로 구성한다.

미해결사건 리스트에서 후보자를 선정하는 방식은 휴리스틱적인 방법을 취한다. 사건을 시간순으로 배열한 뒤 모형의 종스크립트로 하나씩 차례로 적용을 시킨다. 즉 가장 먼저 발생한 사건을 모형의 자식으로 일단 적용하면 나머지 사건은 종스크립트간의 시간적인 관계때문에 적용가능한 범위가 적어진다. 이것을 계속하여 결국은 적용가능한 모든 후보자를 구해 낸다 이것은 저수함수 이상의 복잡도를 갖지만 하나의 리스트에 속한 사건의 갯수가 대개 열개 미만이라고 예상하므로 실제 시스템의 성능에는 큰 영향을 미치지 않는다.

SMAT 시스템은 지금까지 설명한 알고리즘이 계속 반복하면서 상황을 평가하게 된다. 어느 한 시점에서 가설은 가설스크립트 리스트에 속한 스크립트인스턴스 트리이며, 그때의 미해결사건 리스트는 아직까지 설명되지 못한 사건을 나타내게 된다.

3.4 가능성의 계산

정의되어 있는 모형을 참조하여 트리구조의 가설 스크립트인스턴스를 구성하려면 부모스크립트인스턴스와 자식 스크립트인스턴스와의 관계가 모형과 얼마나 매칭하는가를 측정할 수 있는 척도가 필요한데, 이것을 가능성(likelihood)이라고 정의한다. 현실적으로 가능성이란 것은 매우 다양하고 복잡한 요인에 의해 결정된다. 스크립트를 이용한 많은 시스템이 가능성에 대해서는 강조를 안한다거나 특별한 방식으로 처리하는 이유도

이렇게 상황이나 여건에 민감하고 복잡한 계산을 요구하면서도 특별한 기준을 찾기가 어렵기 때문이다.

일반적으로 스크립트의 내부상태, 발생시간, 중요도 등이 가능성계산의 중요한 인자에 해당한다. SMAT에서는 가능성계산을 이 세가지 인자를 사용하여 단순화하여 사용하였다. 가능성은 0 부터 1 사이의 값을 가지게 되는데 가능성이 0이면 두 객체가 절대로 매칭할 수 없다는 뜻이고, 가능성이 1이면 두 객체가 완전히 매칭한다는 뜻이다. 가능성을 계산하는 식은 <그림 7>에 나타나 있다.

스크립트인스턴스의 가능성은 각 종스크립트의 가능성, 지역변수간의 매칭정도(matching degree), 종스크립트의 중요도를 곱하여, 이 값을 모든 종스크립트에 대하여 평균한 값이다. 지역변수간의 매칭정도란 종스크립트에서 사용된 지역변수와 스크립트에서 사용된 지역변수가 얼마나 일치하는가를 나타낸다. 지역변수는 취하는 값에 따라 두종류로 나눌 수 있다. 단순히 값을 취하는 단순변수와 객체지향 환경이나 프레임기반 환경에서 사용하는 객체인스턴스를 값으로 갖는 객체변수(object variable)이다. 두 객체 변수를 비교할 때에는 값이 정확히 일치하지 않으면 같다고 할 수 없다. 물론 서로 다른 객체인스턴스의 경우라도 두 인스턴스의 내부변수(instance variable)를 비교해 가면서 동일한 물체에 대한 중복된 인스턴스라는 것을 찾아내는 것도 생각할 수 있겠으나 여기에서는 단순화하기 위하여 그런 경우를 제외하였다. 단순변수의 경우에도 취하는 값이 수치적인 값이 아닌 경우에는 정확한 일치를 요구하였다. 수치적인 값을 취하는 경우에는 편차를 허용하여 사용자가 스크립트 정의시에 error 키워드를 사용하여 명시할 수 있도록 하였다.

앞서 보았던 <그림 1>에서 use-computer 종스크립트의 duration 변수의 편차가 error를 사용하여 2로 정의되었다. 따라서 만약 programming 인스턴스의 duration 값이 20이고, use-computer 종스크립트의 duration 값이 22라면 두 변수사이의 매칭정도는 $P(|Z| < (22-20) / 2) = 0.683$ 이 된다.

이상과 같이 가능성의 계산에 대해 정의하였으나 한 가지 더 고려할 점은 종스크립트나 지역변수가 지정되지 않은 경우이다. 두 변수사이의 가능성을 계산하는데 있어서 둘 다 미지정인 경우, 둘 중 하나만 미지정인 경우가 발생하게 되고 이런 경우에 가능성을 어떤 식으로

정의하느냐가 문제가 된다. 또한 스크립트의 가능성계산에 있어서 종스크립트가 미지정인 경우도 있을 수 있다. 어떤 값이 미지정이란 의미는 앞으로 발생하는 사건에 의하여 종스크립트가 지정될 수 있고 이에 따라 가능성의 변화가 있을 수 있다는 것이다. 본 시스템에서는 값이 지정되지 않은 경우에는 가능성이 가질수 있는 값의 중간 값인 0.5를 취하여 중립으로 처리하도록 하였다.

$$\lambda(x) = \sum [w(\text{sub}) \{ \lambda(\text{sub}) * \sum \text{md}(v,u) \}]$$

$\text{sub} \in \text{all-subscripts}(x), (v,u) \in \text{LM}(\text{sub})$

$\lambda(x)$ x 스크립트의 가능성

$\text{all-subscripts}(x)$ x의 모든 종스크립트의 집합

$w(\text{sub})$ sub 종스크립트의 중요도(weight)

$\text{LM}(\text{sub})$ sub 종스크립트의 지역변수와 스크립트의 지역변수와의 대응관계

$\text{md}(v,u)$ u와 v의 매칭정도(matching degree)

(가) 가능성 계산식

$$\text{md}(x, y) =$$

1) 객체변수이거나 $\sigma = 0$ 일 경우

$$\text{if } x = y \text{ 1}$$

$$\text{else } 0$$

2) 단순변수이고 $\sigma > 0$ 일 경우

$$P(|Z| \leq (y-x) / \sigma)$$

σ 편차

Z 표준정규분포의 확률변수

(나) 지역변수의 매칭정도

(그림 7) 스크립트의 가능성 계산

4. 비교 및 분석

4.1 SCAN(Scripted Analyst) 시스템과의 비교

SMAT 시스템의 성능을 평가하기 위하여 SCAN 시스템과 비교한 결과가 <표 1>에 나타나 있다. SCAN은 상황평가 문제에 스크립트를 적용한 시스템으로서 개발된 ANALYST 전문가시스템의 후위(backend) 시스템으로 개발되었다 [LAS87] 표에서 알수 있듯이 SCAN과 SMAT의 차이는, 첫째 스크립트인스턴스의 문맥을 어떻게 정의하느냐, 둘째 가능성계산을 어떤 방

식으로 하는가 등이 된다. 여기에 한가지 덧붙인다면 가설을 구성하는 방식이 세번체의 차이점이 된다.

SCAN에서는 스크립트의 시간, 위치, 경과 등을 상태를 나타내는 정보로서 미리 결정하여 이것의 값으로 스크립트의 문맥을 나타내었다. 그렇지만 스크립트가 풍부한 표현력을 갖기 위해서는 Schank의 정의와 같이 스크립트에서의 행위의 주체와 그와 관련되는 여러 객체를 스크립트 정의자체에 포함하여 정의할 수 있어야 하며, 매칭과정에서 이것을 관리할 수 있어야 한다 [SCH77]. SMAT에서는 스크립트에 지역변수를 정의할 수 있도록 하여 이런 문제를 해결하였으며, 매칭과정에서 구성된 가설에서 지역변수간의 매칭정도가 스크립트의 가능성계산에 고려되었다.

<표 1> SCAN과 SMAT의 비교

	SCAN	SMAT
동작	ANALYST와 연결	독립적/연결가능
가설	복수	복수
문맥 정보	미리 정의되어 있음	사용자가 지역변수를 사용하여 정의
가능성 계산	스크립트마다 예외적으로 처리	공통적으로 정의
타 시스템과의 관계	강하게 연결	약하게 연결

SCAN에서의 가능성계산은 각 스크립트 정의마다 가능성계산식을 따로 정의할 수 있도록 하여 모든 스크립트마다 예외적인 방식으로 처리하였다. [LAS87] SMAT에서는 가능성계산식을 구체적으로 수식화하여 한 스크립트의 가능성은 각 종스크립트의 가능성과 종스크립트와의 지역변수의 매칭정도를 고려하도록 정의하였다. 구체적인 수식에 관해서는 3.4절에서 이미 자세히 기술한 바 있다.

SMAT에서 사용한 매칭방식은 모형유도방식과 시건 유도방식을 결합한 복합적 매칭방식을 사용하였다. 이 방식은 가설을 사건과 동일하게 취급하므로 가설스크립트인스턴스 트리의 단계가 많아지는 경우에도 단계가

작은 경우와 완전히 같은 방식으로 처리할 수 있어 가설구성에서 참조되는 모형스크립트의 단계를 갖게 할 수 있다. 따라서 지식표현에 대한 제한이 완화되게 된다.

이상과 같은 차이의 근거는 SMAT 시스템이 SCAN의 생각을 많이 이용하면서 약점을 보완하려는 시도였고, 본래 ANALYST와 연결되어 사용되는 것을 목표로 했던 SCAN과는 달리 독립적으로 동작할 수 있으면서 보다 일반적인 성격을 갖는 상황평가 전문가시스템을 개발하려는 의도가 있었기 때문이다.

4.2 SMAT의 제약점

SMAT 시스템에서는 3.3절에서 기술한 바와 같이 지역변수간의 정합을 단순화하여 지역변수가 프레임이나 심볼 등의 비수치적 값을 갖는 경우는 상위 스크립트인스턴스와 하위 스크립트인스턴스 사이에 완전히 매칭해야 한다고 규정하였고, 수치적 값을 취하는 경우에만 오차를 허용하였다. 실제로 발견되는 자료들은 같은 물체를 다른 인스턴스로 실제생성(instantiation)할 수 있으므로, 프레임과의 매칭정도를 정의하며 스크립트의 가능성제산에 고려하여야 한다.

또한 시스템에서 구성한 여러가지 가설에서 기대했던 기대사건 중에 비슷한 것이 많은 경우에 한 사건의 이해 여러 가설의 가능성이 동시에 증가하게 되므로, 가설의 갯수가 폭증해 버리는 경우가 발생할 수 있다. 이와 같이 비슷한 기대사건이 많은 경우에는 사람도 마찬가지로 갈등을 겪게 되므로 이런 경우를 대비할 필요가 있다. 해결방법으로는 기대사건에도 중요도를 두어 기대사건간의 유사성이 많이 있을 경우에 중요도를 낮추는 방법이 있을 수 있다. 즉, 가설의 선택에 영향이 큰 기대사건을 선호하는 매칭을 행하는 것이다.

SMAT는 독립적으로 운용될 수 있을 뿐 아니라 타 추론시스템과 연결되어 가설구성에 필요한 정보를 얻을 수 있도록 구성되었다. 본 논문에서는 모형을 선택할 때 규칙기반시스템에서 전체적 상황을 나타내는 정보를 얻어서 후보자의 갯수를 줄일수 있도록 하였다. 이와 같은 인터페이스는 논리를 기반으로 하는 시스템이나 그 밖의 다른 시스템과도 연결될 수 있도록 SMAT 시스템과는 독립적으로 정의되고 개발될 필요가 있다.

4.3 응용시스템 개발

SMAT의 상황평가 기능을 이용하여 군의 작전지휘 및 통제를 담당하는 C³I센터 운영요원을 지원하는 작전지휘 보조시스템을 시제품으로 개발하였다. 시스템의 추론부분은 SMAT와 규칙기반시스템으로 구성되며, 발생하는 사건을 근거로 상황을 해석 및 예측하여 운영요원을 지원하는 기능을 한다. 시스템 개발과정에서 SMAT의 스크립트를 이용한 표현방식이 매우 적합함을 알 수 있었으며, 현재 이 시스템은 시제품으로서 실용적으로 사용되고 있다.

5. 결론

상황에 대한 해석과 예측을 목적으로 하는 상황평가 문제에 있어서는 스크립트를 지식표현방식으로 사용하는 것이 문제의 성격과 부합하고 지식표현도 용이하다. 본 논문에서는 적용영역에 의존하지 않는 일반적인 스크립트 표현방식을 정의하고 그에 따른 스크립트 매칭방식을 개발하여, 다중상황하에서 관측되는 불연속적이고, 불완전한 사건을 사용하여 전개되는 상황을 파악할 수 있는 상황평가시스템 도구 SMAT(Scrppt Matcher)를 구현하였다.

모형스크립트는 계층적 관계를 갖게 되어 여러 모형스크립트간에 지식의 공유가 가능하도록 정의하였다. 또한 스크립트를 구성하는 하위의 스크립트는 차례로 발생하는 경우뿐 아니라 병렬적으로 발생하는 경우도 정의할 수 있도록 하여 전체적인 모형베이스는 모형스크립트간에 격자구조의 관계를 갖도록 구성된다. 정의된 모형과 발생사건을 정합하는 방식은 모형유도방식의 하향식 매칭전략과 사건유도방식의 상향식 매칭전략을 결합한 복합적 매칭방식을 제안하여 기존의 방식이 안고 있는 영역의존성을 제거하였다. 제안된 매칭방식에서는 트리구조의 가설구성을 단계가 2인 트리구성과정의 반복으로 처리하여 단계별 가설을 독립시킴으로써 매칭과정을 일반화하였다. SMAT는 스크립트로써 지식을 표현하고 복합적 매칭방식을 써서 문제를 해결할 수 있도록 구성된 시스템으로, 입력되는 일련의 사건을 모형과 가장 유사한 구조의 가설로 구성한다.

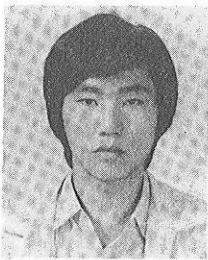
본 논문에서 정의된 스크립트는 기존의 정의에 비해 표현영역이 확장되었다. 매칭방식은 전체문맥에 대한 정보를 사용할 수 있으면서, 발생사건의 갯수에 의해 수행속도가 결정되며, 부분적인 가설을 독립시킴으로써

써 일반적으로 적용될 수 있는 장점을 갖는다. 반면 개선되어야 할 점으로는 종스크립트간의 시간적 관계의 종류를 세분화하여 상황을 더욱 자세히 표현할 수 있어야 하며, 지역변수가 복잡한 구조의 프레임인 경우에 대한 처리가 보완되어야 한다. 또 규칙기반시스템과의 인터페이스도 더욱 개선하여 매칭과정에서 전체적인 상황을 활용할 수 있도록 하는 것이 바람직하며, 가능성의 계산도 수학적인 형식화를 보완할 필요가 있다.

참 고 문 헌

[BAR81] A.I. Barr and E.A. Feigenbaum, The Handbook of Artificial Intelligence, William Kaufmann, Los Altos, 1981.
 [CHE85] D.C. Chen, "Progress in Knowledge-Based Flight Monitoring," Proc. the 2nd Conference on Artificial Intelligence Applications, Miami Beach, Florida, pp. 441-446, Dec. 1985.
 [DEJ82] G.F. DeJong II, "An Overview of the FRUMP System," In W.G. Lehnert and M.H. Ringle (Eds.), Strategies for Natural Language Processing, Lawrence Erlbaum, Hillsdale, pp. 149-176, 1982.
 [DYE83] M.G. Dyer, In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension, MIT Press,

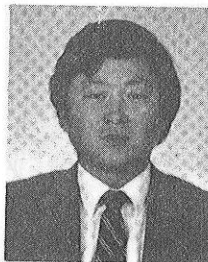
Cambridge, 1983.
 [HAR85] P. Harmon and D. King, Expert Systems: Artificial Intelligence in Business, Wiley Press, New York, 1985.
 [LAS85] S.J. Laskowski, H.J. Antonis'e and R.P. Bonasso, "ANALYST II: A Knowledge-Based Intelligence Support System," Proc. the 2nd Conference on Artificial Intelligence Applications, Miami Beach, Florida, pp. 558-563, Dec. 1985.
 [LAS87] S.J. Laskowski and E.J. Hofmann, "Script-Based Reasoning for Situation Monitoring," Proc. the 6th National Conference on Artificial Intelligence, Seattle, Washington, pp. 819-823, July 1987.
 [MIN75] M. Minsky, "A Framework for Representing Knowledge," the Psychology of Computer Vision, McGraw-Hill, New York, pp. 211-277, 1975.
 [SCH77] R.C. Schank and R.P. Abelson, Scripts, Plans, Goals and Understanding, Lawrence Erlbaum, Hillsdale, 1977.
 [SCH82] R.C. Schank, "Reminding and Memory Organization: An Introduction to MOPs," In W. Lehnert and M. Ringle (Eds.), Strategies for Natural Language Processing, Lawrence Erlbaum, Hillsdale, pp. 455-493, 1982.
 [WAT86] D.A. Waterman, A Guide to Expert Systems, Addison-Wesley, Menlo Park, 1986.



김 석 원

1987년 서울대 전자계산기 공학과 졸업.
 1989년 한국과학기술원 전산학과 석사학위 취득.
 1989년~현재 한국과학기술원 전산학과 박사과정.
 관심분야 : 인공지능, 전문

가시시스템, 지능형교육시스템 (Intelligent Computer Assisted Instruction)



김 진 형

1971년 서울대학교 공과대학 졸업.
 1979년 UCLA 전산학 석사학위 취득.
 1983년 UCLA 전산학 박사학위 취득.
 1973년~1976년 KIST 전산실

연구원.

1976년~1977년 미국 California 도로국 연구원.
 1981년~1985년 미국 Hughes 인공지능센터 선임연구원.
 1985년~현재 한국과학기술원 전산학과 부교수.
 1986년~1988년 본 학회 산하 인공지능연구회 위원장.
 주 관심분야는 패턴인식, 전문가시스템, 지능형 Man-Machine Interface 등임.