

활성 지식 원천들의 신속한 탐지를 위한 메커니즘

장혜진^{1*}

¹상명대학교 컴퓨터소프트웨어공학과

A Mechanism for Fast Detection of Knowledge Source Activations

Hai Jin Chang^{1*}

¹Dept. of Computer Software Engineering, Sangmyung University

요 약 블랙보드 구조는 복잡하고 풀이 방법이 잘 규정되어 있지 않은 문제들을 다루기 위한 구조이다. 블랙보드 구조를 사용하는 시스템의 효율을 높이기 위하여, 본 논문은 블랙보드의 상태가 변화할 때마다 문제 풀이에 잠재적으로 기여할 수 있는 활성 지식 원천들(knowledge source activations)의 집합을 신속하게 탐지하기 위한 메커니즘을 제안한다. 제안된 메커니즘은 블랙보드 데이터와 지식 원천들의 활성화 조건들(activation conditions) 간의 패턴 매칭을 신속하게 수행하기 위하여 지식 원천들이 등록하는 활성화 조건들로부터 Rete 네트워크를 생성하여 사용한다.

Abstract Blackboard architecture was designed as a means for dealing with ill-defined and complex problems. In order to improve the efficiency of the systems using blackboard architecture, this paper proposes a mechanism for fast detection of knowledge source activations which can contribute potentially to the problem solving of blackboard systems, whenever the state of blackboard is changed. The proposed mechanism uses a Rete network generated from the activation conditions subscribed by all knowledge sources to process the pattern matching between blackboard data and the activation conditions efficiently.

Key Words : Blackboard architecture, Detection of knowledge source activation, Rete network, Activation condition, Pattern matching.

1. 서론

블랙보드 시스템이란 블랙보드 구조를 사용하는 응용 시스템을 의미한다. 블랙보드 구조는 복잡하고 명확하게 정의하기 어려운 문제들을 지식 원천들의 협동에 의해 다루기 위하여 설계되었으며 데이터 융합, 지식 기반 시뮬레이션, 프로세스 제어, 컴퓨터 비전 등의 다양한 분야에 응용될 수 있다[1]. 블랙보드 구조는 시스템 구성의 유연성, 확장성, 소프트웨어 재사용성 등의 장점을 갖는다 [2,3].

블랙보드 시스템의 수행 방식은 혼자서 풀기 어려운 복잡한 문제를 풀기 위하여 다양한 분야의 전문가들이

토의하는 방식에 비유될 수 있다. 블랙보드 시스템의 기본 구성 요소들은 전문가들에 해당되는 지식 원천들(knowledge source), 공유 칠판에 해당되는 블랙보드, 그리고 전체 문제 풀이 과정을 제어하는 제어 모듈이다[4].

블랙보드에는 문제 풀이 과정에 필요한 입력 데이터, 부분 해답들, 대안, 최종 해답들, 그리고 제어 정보와 같은 공유 데이터들이 저장될 수 있다. 블랙보드의 상태는 문제 풀이 과정에 참여한 지식 원천들의 데이터 삽입, 삭제, 수정 연산에 의하여 변화된다.

지식 원천들은 블랙보드 시스템에 주어지는 문제들을 푸는 데 필요한 전문 지식을 가지고 있다. 블랙보드 시스템에 참여한 지식 원천들의 전문 지식의 품질은 블랙보

본 논문은 상명대학교 교내선발과제로 연구되었음.

*교신저자 : 장혜진(hjchang@smu.ac.kr)

접수일 10년 11월 29일

수정일 (1차 11년 01월 03일, 2차 11년 02월 08일)

계재확정일 11년 02월 10일

드 시스템의 성능에 큰 영향을 미칠 수 있다. 각 지식 원천은 블랙보드의 어떤 상태에서 자신이 활성화되어 문제 풀이 과정에 기여할 수 있는가를 판정하기 위한 기준 조건을 갖는다. 본 논문에서는 이후 그 기준 조건을 표현하기 위한 패턴을 지식 원천의 *활성화 조건 패턴*(activation condition pattern)이라고 기술한다. 블랙보드의 현재 상태가 어떤 지식 원천의 활성화 조건 패턴을 만족시킨다면, 그 지식 원천에 대한 활성화 지식 원천들(knowledge source activations)을 생성할 수 있다. 활성화 조건 패턴이 만족시키는 활성화 지식 원천들을 탐지하는 제안 메커니즘에 대한 보다 자세한 논의는 제 3장에서 기술된다.

제어 모듈은 다음 블랙보드 수행 사이클에 수행할 활성화 지식 원천(또는 활성화 지식 원천들)을 선정하고 수행시킨다. 제어 모듈은 다음에 수행할 활성화 지식 원천을 합리적으로 결정하기 위하여 활성화 지식 원천들을 평가하는 수단을 가질 수 있다.

많은 생성 규칙 시스템들이 충돌 집합(conflict set) 즉 작업 메모리의 현재 상태에서 잠재적으로 수행될 수 있는 활성화 규칙들의 집합을 신속하게 탐지하기 위하여 Rete 네트워크[5] 기술을 사용하고 있다. 생성 규칙 시스템에서 충돌 집합을 탐지하는 메커니즘의 성능은 전체 시스템의 성능 향상에 매우 중요하다. 블랙보드 시스템에서도 활성화 지식 원천들 즉 블랙보드의 현재 상태에서 문제 풀이에 잠재적으로 기여할 수 있는 활성화 지식 원천들을 효율적으로 탐지하기 위한 메커니즘이 블랙보드 시스템의 전체적인 성능 향상에 중요하다. 본 논문은 지식 원천들로부터 활성화 조건 패턴들을 등록받아 Rete 네트워크를 생성하고 그 Rete 네트워크를 이용하여 블랙보드의 상태가 변화할 때마다 문제 풀이에 잠재적으로 기여할 수 있는 활성화 지식 원천들의 집합을 신속하게 계산하기 위한 메커니즘을 제안한다. 제안된 메커니즘은 기존의 블랙보드 시스템들의 활성화 지식 원천 탐지 메커니즘들과 달리 Rete 네트워크에 기반을 두고 있다. 기존의 블랙보드 시스템들은 활성화 지식 원천들을 탐지하기 위하여 트리거, 데몬(demon), 또는 식별 네트워크(discrimination network) 등의 기술들을 단독으로 또는 혼합하여 사용하고 있다.

본 논문의 제 2장은 관련 연구들에 대한 것이다. 제 3장에서는 제안하는 메커니즘의 구조, 특성, 그리고 평가에 대하여 기술한다. 제 4장은 결론이다.

2. 관련 연구

블랙보드 구조를 최초로 사용한 시스템은 음성 인식을 위한 시스템인 Hearsay-II [6]이다. Hearsay-II는 음성 인

식에 관련된 다양한 수준의 전문 지식을 가진 지식 원천들과 공유 블랙보드를 사용하였다. HEARSAY-II의 이후로 AGE[7], BB1[8], ERASMUS[9], 그리고 GBB[10] 등의 블랙보드 시스템들이 개발되었다.

Rete 네트워크는 다량의 패턴들과 다량의 데이터간의 패턴 매칭을 신속하게 수행하기 위한 기술이다. OPS5[11], CLIPS[12], JESS[13], 그리고 Drools[14] 등의 대부분의 생성 규칙 시스템들이 Rete 네트워크를 사용하고 있다. Rete 네트워크는 다양한 데이터 및 패턴의 표현을 지원할 수 있다. 예를 들어, Drools 시스템에서 구현된 Rete 네트워크는 Java 객체들을 데이터로 사용하는 패턴 매칭을 수행할 수 있다. Rete 네트워크는 생성 규칙 시스템에서 충돌 집합을 신속하게 탐지하기 위하여 고안되었지만 생성 규칙 시스템만을 위한 기술은 아니다. 예를 들어, Rete 네트워크는 통신 보안 시스템에서 보안 공격들을 탐지하기 위한 보안 패턴을 표현하는 데 사용되거나 [15], 계획기반 에이전트 시스템의 성능을 향상시키는데 이용될 수도 있다[16].

블랙보드 시스템과 생성 규칙 시스템은 개념적으로 비교될 수 있다. 블랙보드 시스템에서는 지식 원천들에 의해 공유되는 데이터를 저장하기 위하여 블랙보드가 사용되며, 생성 규칙 시스템에서는 생성 규칙들에 의해 공유되는 데이터를 저장하기 위하여 작업 메모리가 사용된다. 또한 블랙보드 시스템의 지식 원천은 블랙보드의 상태가 그 지식 원천의 활성화 조건을 만족시킬 때 활성화될 수 있으며, 생성 규칙 시스템의 생성 규칙은 작업 메모리의 상태가 그 규칙의 조건부를 만족시킬 때 활성화될 수 있다.

하지만, 생성 규칙 시스템에서 생성 규칙들은 서로 동일한 문법 구조와 의미 체계를 갖는 반면에 블랙보드 시스템의 지식 원천들은 필요한 지식의 종류, 수준, 그리고 특성에 따라서 다른 지식 원천들에 독립적인 다양한 종류의 지식 표현 방식을 사용할 수 있다. 예를 들어, 신경망 기술을 사용하는 지식 원천과 퍼지 로직 기술을 사용하는 지식 원천이 하나의 블랙보드 시스템에서 협동할 수 있다. Rete 네트워크를 사용하는 생성 규칙 시스템을 블랙보드 시스템의 지식 원천으로 사용하는 시스템도 존재할 수 있다.

블랙보드 시스템을 구성하는 여러 가지 요소들의 성능이 시스템 전체의 성능에 영향을 미칠 수 있다. 블랙보드 자체의 데이터 저장 효율과 지식 원천들에 의한 데이터의 검색, 삽입, 삭제, 수정 연산자들의 성능 개선을 통하여 블랙보드 시스템의 성능을 향상시킬 수 있다[10]. 분산 환경에서 지식 원천들의 문제 풀이 과정을 병렬적으로 처리할 수 있도록 하기 위한 블랙보드 구조에 대한 연

구들[17,18]도 존재한다.

HEARSAY-II는 활성 지식 원천들의 효율적 탐지를 위하여 블랙보드 데이터의 특정한 변화에 의해서 자동으로 활성화되는 데몬 기반의 지식 원천 활성화 메커니즘을 사용한다[6,19]. HEARSAY-II의 데몬 기반의 지식 원천 활성화 메커니즘은 이후에 개발된 많은 블랙보드 시스템들에 영향을 미쳤다. HEARSAY-II의 지식 원천은 조건 컴포넌트(condition component)와 동작 컴포넌트(action component)로 구성된다. 조건 컴포넌트와 동작 컴포넌트는 임의의 프로그램들로 구현될 수 있다. HEARSAY-II는 블랙보드의 상태가 변화할 때마다 조건 컴포넌트들을 반복적으로 재평가 하는 방법 대신, 조건 컴포넌트마다 그것이 관심을 가진 기초적 상황들을 선언하게 하고, 블랙보드 상에 선언된 기초적 상황들을 생성하는 변화가 발생하였을 때에 데몬들을 이용하여 자동으로 해당 조건 컴포넌트들을 평가하기 위한 인터럽트가 통보되는 메커니즘을 사용한다.

Poligon[20] 시스템도 블랙보드의 상태 변화에 의하여 활성화되는 데몬들을 이용하여 지식 원천들을 수행시킨다. Poligon 시스템은 HEARSAY-II 시스템과 달리 활성 지식 원천들의 아젠다(agenda)를 사용하지 않으며, 잠재적으로 적용 가능한 지식 원천들 중에서 가장 유망한 지식 원천을 선택하기 위한 중앙 집중적 제어 기능을 갖지 않는다.

데몬 기반의 메커니즘들의 공통적인 문제점은 데몬들의 수가 많아지거나 데몬들이 감시하는 조건 패턴들이 복잡해지면 많은 자원을 소모하게 될 수 있다는 것이다. 본 논문이 제안하는 메커니즘은 데몬 기반의 지식 원천 활성화 메커니즘이 아니다. 본 논문이 제안하는 메커니즘은 다량의 패턴들과 다량의 데이터간의 패턴 매칭을 신속하게 수행하기 위한 기술인 Rete 네트워크에 기반을 두고 있다. 또한 본 논문의 메커니즘은 제 3.3절에서 기술된 것과 같이 중앙 집중적으로 제어되는 수행 사이클을 가진 블랙보드 구조를 지원할 수 있다.

BB1[8] 시스템의 각 지식 원천은 트리거 조건(trigger condition)이라 불리는 초기 활성화(initial activation) 조건과 선조건(precondition)이라고 불리는 완전 활성화(full activation) 조건, 그리고 제거 조건(obviation condition)을 갖는다. BB1의 활성 지식 원천 선발 절차는 트리거링 단계(triggering stage), 상태 기반 활성화 단계(state-based activation stage), 그리고 제거 단계(obviation stage)로 구성된다. 트리거링 단계에서는 트리거 조건이 만족되는 지식 원천들이 일차적으로 선발된다. 상태 기반 활성화 단계에서는 트리거링 단계에서 선발된 지식 원천들 중에서 선조건이 만족되는 지식 원천들만을 선발한다. 제거 단계

에서는 상태 기반 활성화 단계를 통과한 활성 지식 원천들 중에서 제거 조건을 만족하는 것들을 제거한다. 트리거링 단계와 상태 기반 활성화 단계를 통과한 활성 지식 원천들은 실행 아젠다(executable agenda)에 기록되며 트리거링 단계만을 통과한 지식 원천들은 트리거 아젠다(triggerd agenda)에 기록된다. 제거 단계까지 통과한 실행 아젠다 상의 활성 지식 원천들만이 BB1의 제어 모듈에 의해 선발되어 수행될 자격을 갖는다.

BB1 시스템은 지식 원천들의 트리거링 조건들을 식별 네트워크로 컴파일 하는 기법을 사용하여 트리거링 단계를 효율적으로 수행한다. 하지만, BB1 시스템의 초기 버전은 범용의 상태 기반 활성화 메커니즘을 지원하지 않으며, 수행 가능한 지식 원천들의 수가 많아지면 상태 기반 활성화 단계의 효율이 저하되는 문제점을 갖고 있었다. 그 문제를 개선하기 위하여 데몬 기반의 상태 기반 활성화 메커니즘[21]이 제안되었다. 그 메커니즘은 블랙보드 상의 객체에 변화가 생기면 그 객체에 결합된 데몬이 자동으로 수행되어 그 객체를 참조하고 있는 선조건들만을 재평가하도록 하여 블랙보드의 상태가 변화하였을 때 트리거링 단계를 통과한 모든 지식 원천들의 선조건들을 재평가해야 하는 부담을 감소시킨다. 하지만 제안된 메커니즘에 의해 블랙보드상의 데이터에 데몬들을 붙이고 떼어야 하는 새로운 부담이 생길 수 있다.

본 논문이 제안하는 메커니즘을 사용하면 여러 단계들로 구성된 BB1 시스템의 활성 지식 원천 선정 절차보다 단순화된 활성 지식 원천 탐지 절차를 가진 블랙보드 수행 사이클을 지원할 수 있다. 본 논문의 3.3절에 본 논문이 제안하는 메커니즘이 지원하는 블랙보드 수행 사이클에 대한 기술이 나온다.

3. 제안 메커니즘

본 논문이 제안하는 메커니즘은 Rete 네트워크가 지원할 수 있는 모든 종류의 블랙보드 데이터 및 패턴 표현 방식을 지원할 수 있으며, 특정한 지식 표현 방식들을 가정하지 않는다. 하지만 본 논문은 제안 메커니즘의 구현을 위하여 중첩 리스트 형태의 문법을 만족하는 데이터와 패턴을 사용하였다. 모든 리스트의 첫 번째 원소로 심벌만이 올 수 있다. 변수를 나타내는 모든 심벌은 \$로 시작된다.

3.1 지식 원천 활성화 조건과 활성 지식 원천의 생성

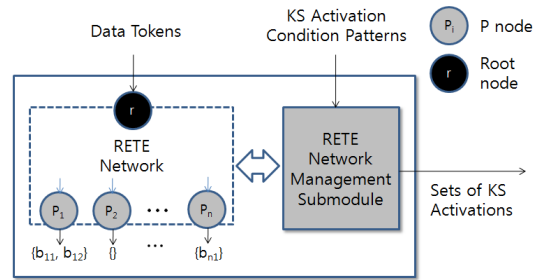
본 논문의 제안 메커니즘은 지식 원천 활성화 패턴들과 블랙보드 데이터간의 패턴 매칭의 결과로 생성되는 바인딩들(bindings)을 이용하여 그에 대응하는 활성화된 지식 원천들을 생성한다.

어떤 지식 원천의 활성화와 조건 패턴과 블랙보드 데이터 간의 패턴 매칭의 결과로 널(null)이 아닌 바인딩이 생성된다면, 그 바인딩을 이용하여 그 지식 원천을 활성화시킬 수 있다. 예를 들어, 어떤 지식 원천의 활성화 조건 패턴이 (point \$x \$y 10) 이라면, 그 지식 원천은 (point 33 44 10) 또는 (point 56 78 10)과 같은 블랙보드 데이터가 입력되었을 때 바인딩 ((\$x 33)(\$y 44)) 또는 바인딩 ((\$x 56)(\$y 78))을 이용하여 활성화될 수 있을 것이다.

블랙보드의 현재 상태에 대하여 지식 원천 활성화 조건 패턴들이 만족되는 지식 원천들이 여러 개 존재할 수 있다. 또한 지식 원천 활성화 패턴이 변수들을 포함하는 경우에는 하나의 지식 원천으로부터 여러 개의 활성화 지식 원천들이 생성될 수 있다. 예를 들어, 어떤 지식 원천의 활성화 조건 패턴이 (p (shape \$x) (size \$y) (color "red"))이고, 블랙보드에 그 활성화 조건 패턴을 만족하는 데이터 (p (shape "rectangle") (size 34) (color "red"))와 데이터 (p (shape "circle") (size 45) (color "red"))가 존재한다고 가정하면, 그 지식 원천을 활성화시킬 수 있는 두 개의 바인딩들 ((\$x "rectangle")(\$y 34))와 ((\$x "circle")(\$y 45)) 이 존재하므로 그 바인딩들을 이용하여 하나의 지식 원천으로부터 두개의 활성화 지식 원천들이 생성될 수 있다.

3.2 활성화 지식 원천 탐지 모듈

블랙보드 구조에서 문제 풀이 과정에 잠재적으로 기여할 수 있는 지식 원천들을 탐지하기 위하여, 블랙보드의 상태가 변화할 때마다 지식 원천들이 지속적으로 블랙보드의 상태를 폴링(polling)하는 방식은 블랙보드와 지식 원천들 간에 많은 통신 오버헤드를 발생시키므로 비효율적이다. 제안된 메커니즘은 지식 원천들이 반복적으로 블랙보드의 상태 변화를 탐지하기 위한 폴링을 수행하는 방법 대신, 각각의 지식 원천으로부터 그것의 활성화 조건 패턴을 등록받고, 등록받은 모든 활성화 조건 패턴들로부터 Rete 네트워크를 생성하고, 그 Rete 네트워크를 이용하여 블랙보드의 상태에 변화가 발생할 때마다 활성화 조건 패턴이 만족되는 지식 원천들을 계산한다. 다음 그림 1은 제안 메커니즘을 구현하는 모듈인 활성화 지식 원천 탐지 모듈(knowledge source activation detection module)의 구조이다.



[그림 1] 활성화 지식 원천 탐지 모듈

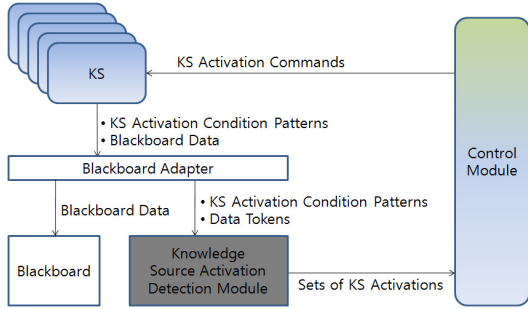
활성 지식 원천 탐지 모듈은 블랙보드 데이터 토큰들과 지식 원천 활성화 조건 패턴을 입력받아 활성화 지식 원천들의 집합을 출력한다. 그림 1에서의 Rete 네트워크 관리 서브모듈(Rete network management submodule)은 지식 원천들로부터 입력된 지식 원천 활성화 조건 패턴들로부터 Rete 네트워크를 생성한다. 일반적으로 Rete 네트워크는 하나의 루트 노드, p 노드들, 타입 노드들, 그리고 결합 노드(join node)들 등의 여러 가지 종류의 노드들로 구성된 사이클을 갖지 않는 방향성 그래프이다[5]. 루트 노드는 블랙보드에 입력되는 모든 데이터에 대응하는 데이터 토큰이 입력되는 노드이다. p 노드들은 Rete 네트워크의 말단에 존재하며 지식 원천 활성화 패턴들과 블랙보드 데이터간의 패턴 매칭의 결과로 생성되는 바인딩들의 집합을 출력한다. 활성화 지식 원천 탐지 모듈의 Rete 네트워크는 지식 원천들이 등록하는 지식 원천 활성화 조건 패턴들의 수만큼의 p 노드들을 갖는다.

Rete 네트워크에서 사용되는 데이터 토큰은 블랙보드 상에 존재하는 데이터들에 대한 하나의 참조 또는 복수개의 참조들을 담은 구조체이다. 지식 원천이 블랙보드에 데이터를 삽입하면 그 데이터에 대한 참조를 담은 데이터 토큰이 그림 1의 루트 노드를 통하여 Rete 네트워크에 입력되어 흐르게 된다. 복수개의 참조들을 포함하는 데이터 토큰들은 Rete 네트워크 내부의 결합 노드들에 의해 생성된다. 토큰들의 흐름이 끝나면 각 p 노드는 대응되는 지식 원천 활성화 조건 패턴과 블랙보드상의 데이터 간의 매칭의 결과로 생성되는 바인딩들의 집합을 출력하게 된다.

그림 1의 Rete 네트워크 관리 서브모듈은 블랙보드의 상태가 변화할 때마다 Rete 네트워크의 p 노드들이 출력하는 널(null)이 아닌 모든 바인딩들을 이용하여 활성화 지식 원천들의 집합을 생성한다. 그 집합의 원소의 수는 그림 1의 Rete 네트워크가 출력하는 널이 아닌 바인딩들의 수와 같다.

3.3 활성 지식 원천 탐지 모듈을 사용하는 블랙보드 구조의 구현

다음 그림 2는 3.2절에서 기술된 활성 지식 원천 탐지 모듈(knowledge source activation detection module)을 사용하는 블랙보드 시스템의 구조이다.



[그림 2] 활성 지식 원천 탐지 모듈을 사용하는 블랙보드 시스템의 구조

그림 2의 블랙보드 구조는 블랙보드 구조의 일반적 구성 요소들인 지식 원천들, 블랙보드, 그리고 제어 모듈을 갖고 있다. 블랙보드는 모든 지식 원천들에 의해 공유되는 자료 구조이므로 블랙보드에 대한 삽입, 삭제, 수정 연산은 동기화된 연산들(synchronized operations)로 처리된다. 그림 2에서 블랙보드 어댑터(blackboard adapter)는 지식 원천들에게 블랙보드와 활성 지식 원천 탐지 모듈에 대한 통합 인터페이스 역할을 제공한다. 블랙보드 어댑터가 제공하는 기능들은 다음과 같다.

- **지식 원천 활성화 조건 패턴의 전달** : 지식 원천들이 등록하는 지식 원천 식별자와 활성화 조건 패턴의 쌍들을 활성 지식 원천 탐지 모듈에 전달하여 그에 대응되는 Rete 네트워크가 생성되도록 한다. 모든 지식 원천은 고유한 식별자를 갖는다.
- **블랙보드 데이터의 삽입 명령의 전달과 처리** : 블랙보드 데이터의 삽입 명령을 블랙보드에게 전달하여 블랙보드에 새로운 데이터를 삽입되도록 하고, 그 데이터에 대한 참조를 담은 데이터 토큰을 생성하여 활성 지식 원천 탐지 모듈 내의 Rete 네트워크의 루트 노드로 입력되도록 한다.
- **블랙보드 데이터의 삭제 명령과 수정 명령의 전달과 처리** : 기존 블랙보드 데이터의 삭제 명령 또는 수정 명령을 블랙보드에 전달하여 처리되도록 하고, 활성 지식 원천 탐지 모듈에게 기존 블랙보드 데이터의 수정 또는 삭제 이벤트가 발생하였음을 통보한다.

다.

RETE 네트워크는 새로운 데이터 토큰의 입력 뿐 아니라 기존 데이터 토큰들에 대한 동적인 삭제나 수정을 지원할 수 있다. 활성 지식 원천 탐지 모듈은 블랙보드 어댑터로부터 블랙보드 데이터의 삭제 이벤트를 통보받으면 Rete 네트워크의 노드들에 저장되어 있는 해당 블랙보드 데이터에 대한 참조를 담고 있는 모든 데이터 토큰들을 무효화시킨다. 블랙보드 데이터의 수정 명령은 논리적으로 블랙보드 기존 데이터의 삭제와 새로운 데이터의 삽입 명령의 결합으로 처리될 수 있다. 활성 지식 원천 탐지 모듈은 블랙보드 어댑터로부터 블랙보드 데이터의 수정 이벤트를 통보받으면 Rete 네트워크의 노드들에 저장되어 있는 해당 데이터에 대한 참조를 담고 있는 데이터 토큰들을 모두 무효화시키고 수정된 블랙보드 데이터에 해당하는 데이터 토큰을 생성하여 Rete 네트워크의 루트 노드에 입력시킨다.

그림 2의 블랙보드 시스템의 구조는 다음 그림 3과 같은 블랙보드 수행 사이클을 지원한다.

Step 1. [활성 지식 원천 탐지]

활성 지식 원천 탐지 모듈을 이용하여 활성 지식 원천들의 집합을 탐지한다.

Step 2. [스케줄링]

활성 지식 원천 집합 내의 활성 지식 원천들을 평가하여 가장 유망한 활성 지식 원천(들)을 선정한다.

Step 3. [수행]

선정된 활성 지식 원천(들)을 수행시킨다. 활성 지식 원천(들)의 수행에 의해 블랙보드의 상태가 변화될 수 있다.

Step 4. [반복]

단계 1로 간다.

[그림 3] 블랙보드 수행 사이클

그림 3의 블랙보드 수행 사이클에서 본 논문이 제안하는 활성 지식 원천 탐지 메커니즘이 직접 담당하는 부분은 단계 1에 해당한다. 그림 3의 단계 2, 3, 4는 중앙 집중적 제어 기능을 갖는 블랙보드 시스템들에 공통적인 수행 단계들이다.

3.4 제안 메커니즘의 평가와 구현

제 2장에서 기술한 바와 같이 기존의 블랙보드 시스템들은 활성 지식 원천의 탐지를 위하여 블랙보드 데이터에 부착되는 트리거, 데몬, 또는 식별 네트워크와 같은 기

술들을 사용하고 있다. 제안된 메커니즘은 그런 기술들과 비교하여 다음과 같은 장점들을 갖는다.

첫째, 트리거나 데몬의 성능은 운영 체제의 특성의 영향을 받으며, 일반적으로 트리거나 데몬들은 수가 많아지거나 그것들이 감시해야 하는 패턴이 복잡해지면 많은 자원을 소모하게 될 수 있다. 하지만 본 논문이 제안하는 Rete 네트워크 기반의 방식은 패턴들과 데이터의 양이 커지는 경우에도 성능이 크게 저하되지 않는다. 왜냐하면 Rete 네트워크는 패턴들로부터 형성된 네트워크와 데이터에 대한 토큰들을 이용하여 데이터가 부분적으로 변화할 때마다 변화된 데이터에 대한 부분적 패턴 매칭만을 수행하기 때문이다[5]. 본 논문이 제안하는 메커니즘의 이론적 성능은 Rete 네트워크의 이론적 성능과 같다. 왜냐하면 본 논문이 제안하는 메커니즘의 활성 지식 원천 탐지 모듈은 Rete 네트워크 기술에 근거하여 지식 원천의 활성화 조건 패턴들과 블랙보드 상의 데이터들에 대한 패턴 매칭을 수행하기 때문이다.

둘째, 식별 네트워크 기술은 네트워크 구조를 이용하여 매칭을 수행한다는 면에서 Rete 네트워크와 유사하다. 하지만 식별 네트워크 기술만으로는 결합(join)을 포함하는 패턴들에 대한 매칭을 지원하지 못하므로[21], 결합을 포함하는 패턴에 대한 매칭을 지원하려면 BB1[8] 시스템의 경우처럼 추가적인 메커니즘을 사용해야 한다. 본 논문이 제안하는 메커니즘은 식별 네트워크를 사용하는 기존의 메커니즘들과 달리 결합을 위한 추가적인 메커니즘을 필요로 하지 않으며, 결과적으로 단순화된 활성 지식 원천 탐지 단계를 제공할 수 있다. 왜냐하면 식별 네트워크 기술과 달리 Rete 네트워크는 데이터의 결합을 포함하는 지식 원천 활성화 조건 패턴들에 대한 매칭을 지원하기 때문이다.

제안 메커니즘은 지능형 로봇의 분산 제어를 위한 블랙보드 구조[22,23]를 지원하기 위해 설계되고 구현되었다. 제안 메커니즘의 설계 및 구현의 세부적 개선과 수행 성능의 향상을 위해 제 3장의 그림 2에서 제시된 구조를 갖는 테스트용 블랙보드 시스템을 개발하였다. 테스트용 블랙보드 시스템은 Drools[14] 버전 4.0의 Rete 네트워크 관련 기능들을 라이브러리로 사용한다. 테스트에 사용된 운영 체제는 JDK 1.6이 설치된 MS Windows Vista이며, 테스트용 시스템은 Eclipse[24] 버전 3.5 Galileo 패키지 상에서 수행하였다. 테스트에 사용된 컴퓨터는 Intel Core2Duo 2.4GHz CPU와 2GB 용량의 주기억장치를 갖고 있다.

테스트용 시스템의 활성 지식 원천 탐지 성능의 검사를 다음과 같은 순서로 진행하였다. 첫 번째로는 100개의 지식 원천 활성화 패턴을 연속적으로 활성 지식 원천 탐

지 모듈에 등록하여 그 패턴들로부터 Rete 네트워크의 생성이 완료되는 시간을 측정하였다. 두 번째로는 25000개의 블랙보드 데이터를 블랙보드 어댑터로 입력하여 블랙보드 데이터의 입력 속도와 활성 지식 원천들의 생성 속도를 측정하였다. 성능 검사를 위하여 입력에 사용된 모든 블랙보드 데이터는 입력된 지식 원천 활성화 조건 패턴들 중의 하나 이상의 패턴에 매칭이 일어나도록 작성되었다. 테스트용 블랙보드 시스템의 성능 측정의 결과는 다음 표 1과 같다.

[표 1] 성능 측정 결과

측정 내용	성능
지식 원천 활성화 조건 패턴의 등록 성능	초당 평균 698개의 활성화 조건 패턴들을 등록함.
1개 이상의 지식 원천을 활성화시키는 블랙보드 데이터의 삽입 성능	초당 평균 4502개의 블랙보드 데이터를 삽입함.
활성 지식 원천 탐지 성능	초당 평균 10445번의 활성 지식 원천들을 탐지함.

표 1의 활성 지식 원천 탐지 성능 항목의 측정결과와는 그림 3의 블랙보드 수행 사이클의 단계 2 즉 활성 지식 원천들의 평가 및 선정 단계의 부담을 제외한 결과이다.

4. 결론

본 논문은 블랙보드의 상태가 변화할 때마다 문제 풀이에 기여할 수 있는 활성 지식 원천들의 집합을 신속하게 탐지하기 위한 메커니즘을 제안하였다. 제안된 메커니즘은 활성 지식 원천들의 탐지를 위해, 트리거, 식별 네트워크, 데몬 등의 기법들을 사용하는 기존의 블랙보드 시스템들과 달리, 각 지식 원천에게 그것의 활성화 조건 패턴을 등록하게 하고, 그 활성화 조건 패턴들로부터 생성된 Rete 네트워크를 이용하여 활성 지식 원천들의 집합을 신속하게 탐지한다. 제안된 메커니즘은 특정한 지식 표현 방식을 가정하지 않으며, Rete 네트워크가 지원할 수 있는 다양한 종류의 블랙보드 데이터 및 패턴 표현 방식을 지원할 수 있다.

참고 문헌

[1] Daniel D. Corkill, "Blackboard Systems", AI Expert,

- 6(9):40-47, September, 1991.
- [2] Hunt J., "Blackboard Architectures", JayDee Technology Ltd 27, 2002.
- [3] D. Rudenko and A. Borisov, "An Overview of Blackboard Architecture Application for Real Tasks", Scientific Proceedings of Riga Technical University, 2007.
- [4] Daniel D. Corkill, "Collaborating Software Blackboard and Multi-Agent Systems & the Future", In Proceedings of the International Lisp Conference, New York, New York, October 2003.
- [5] Charles Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", Artificial Intelligence, 19, pp 17 - 37, 1982.
- [6] Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy, "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty", Computing Surveys, 12(2), pp. 213-253, June 1980.
- [7] Nii, H.P. and Aiello, N., "AGE: A Knowledge -Based Program for Building Knowledge- Based Programs", In Proceedings of the Sixth International Joint Conference on Artificial Intelligence (IJCAI-79), pp. 645-655, 1979.
- [8] Hayes-Roth, B. and Hewett, M., "BB1: An Implementation of the Blackboard Control Architecture", Blackboard Systems, Englemore, R., and T. Morgan, editors, pp. 297-313, Addison-Wesley, 1988.
- [9] Baum, L.S., Dodhiawala, R.T. and Jagannathan, V. "The Erasmus System", Blackboard Architectures and Applications, Jagannathan, V., R. Dodhiawala and L. S. Baum, editors, pp. 347-370, Academic Press, 1989.
- [10] D. Corkill, K. Gallagher and K. Murray, "GGB: A Generic Blackboard Development System", Blackboard Systems, R. Englemore and T. Morgan, editors, Addison-Wesley, 1988.
- [11] Lee Brownston, Robert Farrell, Elaine Kant, Nancy Martin, *Programming Expert Systems in OPS5*, Addison-Wesley, ISBN 0-201-10647-7, 1985.
- [12] Joseph C. Giarratano and Gary D. Riley, *Expert Systems: Principles and Programming*, 3rd edition, Course Technology, ISBN 0-534-95053-1, 1998.
- [13] Ernest Friedman-Hill, *Jess in Action: Rule Java Rule-Based Systems*, Manning Publications, ISBN 1-930110-89-8, 2003.
- [14] <http://www.jboss.org/drools/>, the official website of Drools.
- [15] Kyong-jin Suh, Tae-ho Cho, "Application of Contract Net Protocol to the Design and Simulation of Network Security Model", 한국산학기술학회 학술대회 논문집, pp. 197-206, 2003.
- [16] 이재호, 김남형, 송남훈, "RETE 알고리즘을 이용한 계획기반 에이전트 시스템의 실행 성능 향상", 한국정보과학회 학술발표논문집 v.31 No.2 (1), pp.184-186, 2004년 10월.
- [17] Corkill, D.D., "Design Alternatives for Parallel and Distributed Blackboard Systems", *Blackboard Architectures and Applications*, Jagannathan, V., R. Dodhiawala and L. S. Baum, editors, pp. 99-136, Academic Press, 1989.
- [18] Lesser, V.R. and Corkill, D.D., "The Distributed Vehicle Monitoring Testbed", *Blackboard Systems*, Englemore, R., and T. Morgan, editors, pp. 353-386, Addison-Wesley, 1988.
- [19] V. Lesser and L. Erman, "A Retrospective view of the Hearsay-II Architectures", *Blackboard Systems*, R. Englemore and T. Morgan, editors, Addison-Wesley, 1988.
- [20] J. Rice, N. Aiello and H.P. Nii, "See How They Run... The Architecture and Performance of Two Concurrent Blackboard Systems", *Blackboard Architectures and Applications*, V. Jagannathan, R. Dodhiawala and L. S. Baum, editors, Academic Press, pp. 153-178, 1989.
- [21] Micheal Hewett, Rattikorn Hewett, "Mechanisms for Efficiency in Blackboard Systems", the Proceedings of AAAI-94, pp. 465-470, 1994.
- [22] 광별샘, 이재호, "지능형 로봇 제어를 위한 시스템 통합 방법론", 한국정보과학회지 v.24, No.3, pp.24-34, 2006년 3월.
- [23] 김현식, 신형철, 김만수, 김인철, "분산 제어 구조내의 로봇 작업 계획", 한국정보과학회 2006년도 가을 학술발표논문집 v.33 No.2 (B), pp.319-323, 2006년.
- [24] <http://www.eclipse.org/eclipse>, the official website of eclipse

장혜진(Hai Jin Chang)

[정회원]



- 1994년 2월 : 서울대학교 대학원 계산통계학과 박사 졸업(전산학 전공)
- 1994년 3월 ~ 현재 : 상명대학교 공과대학 컴퓨터소프트웨어 공학과

<관심분야>

통신 보안, 로봇, 분산 제어 시스템, DRM.