블랙보드 구조의 그레이팅 스케쥴링 시스템에의 적용

최 규 성† · 고 종 영† · 조 대 호††

요 약

본 논문은 제조 공정의 생산관리 시스템을 개발함에 있어서, 생산 공정상의 스케쥴링 문제를 해결할 경우 시스템내의 여러 하위 모듈간의 협동을 통해서 처리하게 되는데, 이런 경우 각 모듈간의 원활한 협동을 위한 연동 방법을 정의하여야 한다. 이러한 연동 문제를 해결하기 위해서 분산 인공지능의 분산문제 해결 방법 중의 하나인 블랙보드 구조(Blackboard Architecture)를 사용하였다. 시스템의 문제 해결 과정을 여러 단계로 나뉘어진 공유 작업 공간(Shared Work Space)인 블랙보드에 나타내었으며, 구성 모듈간의 통신은 블랙보드를 통해서 이루어진다. 문제 해결의 처리 절차는 블랙보드의 구성요소 중의 하나인 제어기에 지식의 형태로 정의되어 있고, 제어기는 이 지식을 바탕으로 모듈간의 실행 순서를 제어한다. 이와 같이 블랙보드 구조를 적용하여 하위 모듈간의 협동시에 발생하는 연동 문제를 해결하였으며 또한 시스템의 수정 및 확장에 대처 가능한 환경을 구성하였다.

An Application of Blackboard Architecture to Grating Scheduling System

Kyu-Sung Choi† · Jong-Young Koh† · Tae-Ho Cho††

ABSTRACT

In the development of a production process scheduling system a collaboration method must be defined for the cooperation among submodules within the system. The blackboard architecture is exploited for solving the collaboration problem, which is one of the problem solving architecture that belongs to the distributed artificial intelligence. The dynamic states of the problem solving processes are presented in the hierarchically constructed shared working memory called as a blackboard. The communication for the collaboration is done through the blackboard. The problem solving steps are contained in the global controller, one of a component that consists the blackboard architecture, as knowledge. The global controller activates proper submodules based on the knowledge. By applying the blackboard architecture the collaboration problem among submodules in the grating production process scheduling system (GPSS) has been solved as well as the system became adaptable to the future modifications and expansions.

1. 서 론

컴퓨터의 성능향상 및 사용자의 요구가 다양화됨에 따라 시스템 환경이 급격히 변화함으로 해서 시스템 개발자나 관리자의 입장에서는 개발, 운용, 유지, 보수 등의 부담이 커지고, 시스템 사용자의 입장에서는 사용자의 노력으로 시스템 지식을 획득해야 하므로 이에 대한 부담이 커지게 되었다[3].

이와 같은 상황에 대하여 인간사회의 조직구조와 유

† 준 회 원 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부
†† 정 회 원 : 성균관대학교 전기전자 및 컴퓨터공학부 교수
논문접수 : 1999년 10월 15일, 심사완료 : 1999년 12월 15일

사한 인터페이스를 구현함으로써 적절히 대처할 수 있는데, 분산 인공지능, 에이전트, 객체지향[17] 등의 개념을 바탕으로 한 블랙보드 구조를 적용하여, 서로 다른 환경에서 개발된 소프트웨어 시스템들을 연동시킨다. 그러므로 각각의 시스템을 직접 수정하기 보다는 에이전트 그룹의 확장으로 해결함으로 해서 시스템의 모듈별 독립성을 보장하여 추후의 시스템 확장 및 진화를 용이하게 한다[3].

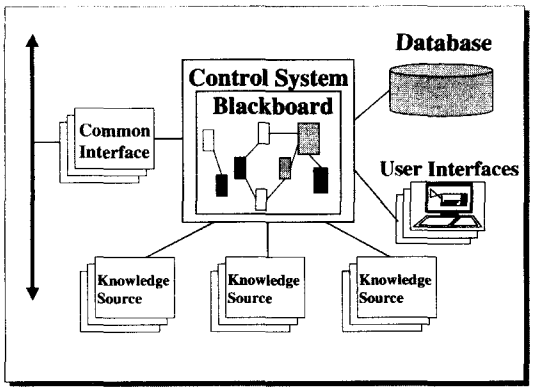
본 논문은 제조공정의 생산관리 시스템인 GPSS(Grating Process Scheduling System)를 개발함에 있어서 생산 공정상의 스케쥴링 문제를 해결할 경우 시스템내의 여러 하위 모듈간의 협동을 통해서 처리하게 되는데, 이런 경우 각 모듈간의 원활한 협동을 위한 연동 방법을 정의하여야 한다. 이를 위해 블랙보드 구조(Blackboard Architecture)를 적용하여 하위 모듈들간의 협동 시 발생하는 연동 문제를 해결하는 것이 목적이다. 본 논문의 구성은 다음과 같다. 2절에서는 연동 문제를 해결하기 위한 배경 이론에 대해 살펴보고, 3절에서는 대상 시스템의 도메인에 대해 살펴보고, 4절에서는 구현, 5절에서는 실행 예를 보이며, 6절에서 결론을 맺는다.

2. 배경이론

나날이 다양해지고 복잡해지는 사용자의 요구를 해결하기 위해서는 기존의 인공지능이 추구했던 단독 에이전트의 능력으로는 한계가 있었고, 그 해결방법으로 지역적으로 떨어진 다른 에이전트[7]의 도움을 받아 처리하는 분산협동 처리의 개념이 접목되기 시작하였다. 이와 관련된 분야가 분산 인공지능[11, 16]으로써 분산문제를 해결[14]하기 위해 소개된 것으로 블랙보드 구조(Blackboard Architecture)와 계약 망 프로토콜(Contract Net Protocol)이 있다

2.1 블랙보드 구조(Blackboard Architecture)

블랙보드 구조는 통신이 블랙보드(Blackboard)라고 불리는 공유된 지식구조(Shared Knowledge Structure)를 통해서 이루어지는 시스템으로서, 블랙보드 구조의 대표적인 예로는 특정 영역내를 이동하는 이동체(자동차등)의 궤적을 추적하는 시스템인 DVMT(Distributed Vehicle Monitoring Testbed)[15]와 Hearsay-II[12] 음성인식 시스템 및 센서 해석의 불확실성을 해결하기 위한 DRESUN[13]등이 있다.



(그림 1) 블랙보드 시스템

● 블랙보드 시스템 구성요소

(그림 1)은 일반적인 블랙보드 시스템의 구조를 표현한 것으로, 블랙보드 시스템의 구성 요소로는 블랙보드, 제어 시스템, 지식원으로 구성되며[5], 각각의 특성은 다음과 같다.

블랙보드(Blackboard) - 분산 인공지능의 공유 메모리 모델로서, 지식원들이 메시지를 쓰고, 부분적인 결과를 게재하며, 정보를 획득할 수 있는 저장소이다. 블랙보드는 지식원들간의 협동을 위한 메시지를 주고 받는 통신로이며, 문제를 해결하기 위한 추상화 된 레벨로 분할이 가능하다.

제어 시스템(Control System) - 문제 해결을 위해서 지식원들을 적용하는 순서를 결정하는 것으로, 항상 블랙보드의 내용을 주시하며 현재의 블랙보드 상태를 개선할 수 있는 지식원을 활성화 시킨다.

지식원(Knowledge Source) - 문제 해결에 필요한 시스템의 도메인 명세 지식을 갖고 있으며 에이전트(Agent)라고도 불린다[4, 9].

블랙보드 제어 메커니즘은 재사용을 위하여 높은 유연성과 성능을 갖는 환경에서 어플리케이션 시스템의 경험적 개발을 하는데 있어서 보다 높은 유연성과 모듈화 환경을 제공한다. 그러므로 블랙보드 구조는 독립적으로 행동하는 새로운 지식원을 첨가, 수정 및 제거하는 것이 가능하며, 이러한 특징으로 해서 시스템의 수정과 확장에 따른 개발 비용과 시간을 단축시킬 수 있다[9].

2.2 계약 망 프로토콜(Contract Net Protocol)

계약 망 프로토콜(Contract Net Protocol)은 Randall

Davis와 Reid Smith에 의해 설계된 (문제 해결을 위한) 분산 처리 노드들간의 협상 프로토콜로서, 한 에이전트가 조정에 대한 책임을 지고, 해결하고자 하는 문제를 작은 단위의 하위 문제들로 나누어 문제를 해결할 에이전트들과 협상하여 결정하는 시스템으로써, 관리자와 계약자의 두 가지 역할을 가질 수 있다[1, 2].

관리자(Manager) 문제를 재구성하며, 하위 작업을 책임질 계약자를 찾아서 할당을 하며, 작업의 실행을 모니터링 한다.

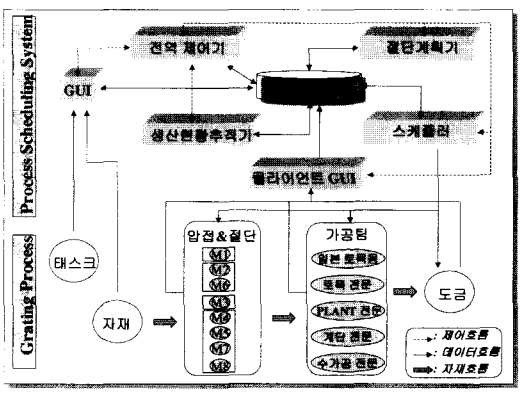
계약자(Contractor) 관리자에 의해 할당된 하위 작업을 수행하며, 작업을 수행하는 것도 가능하고, 자신이 관리자가 되어 다른 계약자들에게 자기 일을 재도급 할 수 있다.

계약 망 프로토콜은 완전한 분산 형태를 취하며, 관리자는 계약자가 작업을 처리하는 시간동안 다른 일을 처리할 수 있음으로 해서 성능이 높은 반면에 제어의 계층적 성질로 인해서 새로운 이벤트를 시스템에 재빨리 이식시키는데 많은 노력이 필요하며, 노드들 사이의 중간 결과에 대한 교환을 조정하는 문제가 발생한다[10].

3. GPSS의 설계

3.1 도메인 분석

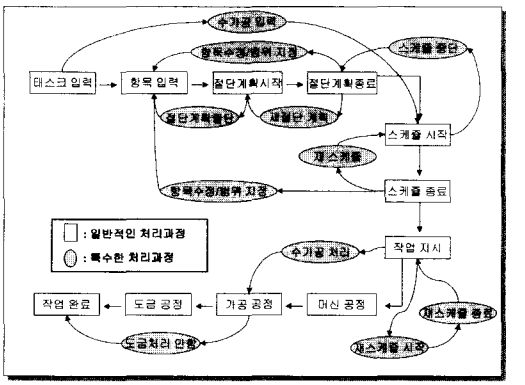
(그림 2)는 적용 대상 시스템의 도메인을 나타낸 것으로, 자재가 처리되는 과정을 표현한 GP(Grating Process)와 구성 모듈간의 상호작용을 표현한 PSS(Process Scheduling System)의 두 부분으로 구성된다.



(그림 2) GPSS의 도메인

GPSS를 구성하는 모듈로는 전역 제어기, GUI, 절단 계획기, 스케줄러, 생산현황 추적기, 클라이언트 GUI, 교차검사(Cross Checking)가 존재한다. (그림 2)는 GPSS에서 태스크 처리와 관련된 모듈들을 중심으로 표현한 것으로, 구성 모듈의 처리 결과에 대한 타당성을 검증하는 교차검사 모듈의 표현은 생략을 하였다.

(그림 3)은 GPSS의 구성 모듈들이 서로 협력하여 문제를 해결하는 태스크 처리 절차를 표현한 것으로써, 특정 상태에서 발생 가능한 모든 사건을 표현한 것이다.



(그림 3) GPSS의 태스크 처리 절차

일반적인 처리과정은 태스크를 처리하는데 반드시 거쳐야 되는 과정을 나타내며, 특수한 처리과정은 자재의 특성 혹은 사용자의 요구가 반영된 처리과정으로 이전 과정으로 되돌아가거나 특정 과정을 건너뛸 수 있다. 작업지시 이전 단계의 태스크는 항상 절단계획과 스케줄 재생성이 가능하지만, 작업지시 이후의 태스크는 절단계획에서 제외가 되며 머신 공정이 이루어지기 전의 태스크에 한해서 스케줄 변경이 가능하다.

3.2 블랙보드 구조와 GPSS

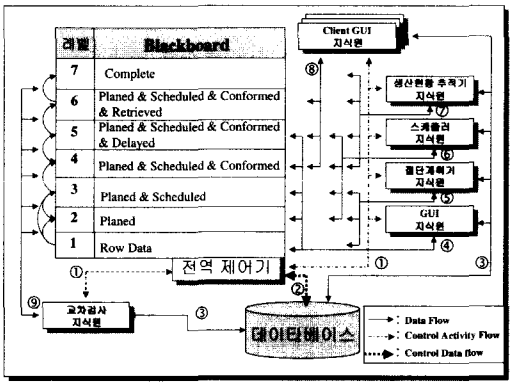
GPSS에서 현재는 고려의 대상이 아니지만, 특정 문제 해결 능력을 갖는 모듈(관리회계 등)들의 수정이나 확장을 고려해야 함으로 해서 수정 및 확장의 용이성을 보유해야 하며, 지식원들간의 연동 문제, 시스템에서 요구되는 문제 해결을 위해서 지식원에게 문제의 분배 및 할당, 지식원들의 행동이 모순이나 충돌없이 효율적으로 조정되어 전체 문제 해결 시스템이 목표를 달성하는 것을 보장하는 조정기법 등을 갖추어야 하는데, 블랙보드 구조가 적합한 구조중의 하나이다.

GPSS의 지식원은 사람, 즉 전문가들이 하던 일을 지식원이 문제 해결을 위해서 나름대로의 지식과 추론 기능을 바탕으로 정보의 교환과 통신을 통해서 문제를 해결하는 기능을 갖고 있는 모듈을 의미한다.

GPSS에서 지식원들간의 연동 문제를 해결하기 위해 제어기[5]는 제어지식, 제어기법, 통신기법이 정의되어야 한다. 제어지식은 제어(control)와 데이터(data)를 분리하였으며, 또한 제어는 제어행동(control Activity)과 제어 데이터(control data)로 분류하여 사용한다. 제어 행동은 지식원들간의 연동을 제어하기 위한 제어기의 문제 해결 절차이며, 제어 데이터는 대상 시스템의 진행 과정을 구분 가능한 몇 개의 단계로 나누어 표현한 것으로, 태스크의 진행 상태를 나타내는 블랙보드를 의미한다. 제어기법은 블랙보드를 기반으로 한 태스크의 상태기반 제어를 하며, 이를 달성하기 위해 제어기는 블랙보드 레벨마다 관련된 문제 해결 절차를 미리 정의하여 놓고, 제어 메시지가 도착하면 메시지와 태스크의 블랙보드를 분석하여 정의된 문제 해결 절차에 따라 실행을 한다. 통신기법은 블랙보드를 통한 메시지 패싱 기법을 사용하여 지식원들간 및 제어기와 지식원간에 통신을 한다.

3.3 세부설계

(그림 4)는 GPSS의 처리 절차를 구분하여 블랙보드에 표현하였으며, 문제 해결 능력을 보유한 지식원 및 지식원의 실행 순서를 제어하는 전역 제어기를 블랙보드 구조에 적합하게 표현한 것이다.



(그림 4) 블랙보드 구조를 적용한 GPSS

GPSS의 태스크 처리 과정은 아래와 같다. GUI를 통해서 특성 태스크의 처리를 위해 전역제어기가 호출이 되면, 전역제어기는 주어진 태스크의 처리를 위해서 태스크의 블랙보드와 수신된 제어 메시지(Control Message)를 참조하여 처리 가능한 지식원을 식별한다. 전역제어기는 식별된 지식원에게 태스크 처리를 일임하며, 지식원에서 처리된 결과의 타당성을 검사하기 위해 교차검사 지식원을 활성화시켜 검사를 한다. 교차검사 후, 태스크 처리를 의뢰한 GUI에게 결과를 통보한다.

전역 제어기에서는 지식원들에 대한 실행 순서를 제어하기 위해서, 각각의 지식원이 접근 가능한 블랙보드 레벨을 분류 및 정의해 놓아야 한다. 이를 위해 태스크 처리 절차를 구분하였으며, 각각의 단계마다 접근 가능한 지식원을 (그림 4)에 실선으로 나타내었다.

GPSS의 블랙보드 구조는 다음과 같다.

①은 전역 제어기가 지식원들의 실행 순서를 제어하기 위한 것으로써 제어 행동의 흐름을 표현한 것이다.

②는 전역 제어기가 태스크 처리의 진행 상태를 식별하기 위해 사용하는 제어 데이터(control data)를 얻기 위해 데이터베이스에 접근하는 것을 나타낸 것이며, 제어 데이터는 태스크의 블랙보드를 의미한다.

③은 전역 제어기에 의해 활성화된 지식원들이 데이터베이스를 통해서 태스크 처리를 위해 필요한 정보를 얻고, 처리되어 생성된 결과들을 저장하는 것을 나타낸 것이다.

④, ⑤, ⑥, ⑦, ⑧, ⑨는 전역 제어기에 의해 활성화된 각각의 지식원들이 접근 가능한 블랙보드 레벨을 나타낸 것이다.

● 지식원의 역할

GUI : 사용자로부터 문제 해결 요청을 받으며, 문제 해결을 전역 제어기에게 요청하고 결과를 수신하여 사용자에게 되돌려준다.

절단 계획기 : 머신에서 태스크를 처리하기 위한 압접 및 절단에 관한 정보를 생성한다.

스케줄러 : 작업 현장의 환경과 능력을 고려하여 머신, 가공, 도금 공정과 관련된 스케줄을 생성한다.

교차검사 : 지식원에서 처리한 결과에 대한 타당성을 검토하여 시스템에 적용유무를 판단할 수 있는 정보를 제시한다.

클라이언트 GUI : 머신, 가공, 도금 공정에서 처리해야 할 태스크에 관한 정보를 나타내며, 사용자로부

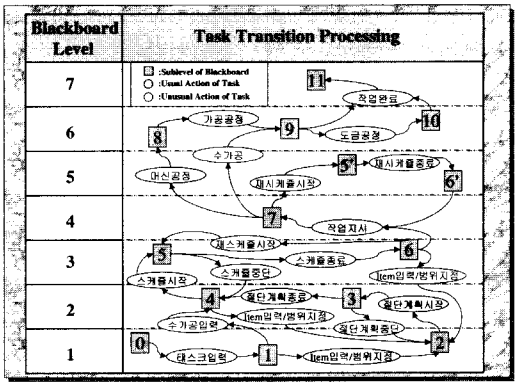
터 처리 결과를 입력 받아 기록한다.

생산현황 추적기 : 클라이언트 GUI와 통신하여 각각의 공정(머신, 가공, 도금)에 대한 작업 상태를 모니터링한다.

4. GPSS의 구현

4.1 블랙보드의 구성

(그림 5)는 지식원들이 협조하여 최종 목표에 도달하는 문제 해결 처리 과정을 체계적으로 분석하여 블랙보드에 적용시킨 것으로 태스크는 블랙보드가 진행되는 방향으로 처리가 된다.



(그림 5) GPSS의 블랙보드 구성

GPSS의 블랙보드는 태스크입력, 절단계획, 스케줄생성, 작업지시, 재스케쥴, 작업처리, 종료의 7 레벨로 구성이 되며, 보다 작은 단위인 14 레벨로 세분화되어 사용된다.

4.2 지식원의 문제 해결 행동 리스트

<표 1>은 각각의 지식원들이 태스크를 처리하는 과정에서, 각각의 지식원마다 태스크처리와 관련된 행동을 명확히 분석 및 정의함으로 해서 전역 제어기가 지식원들의 실행 순서를 제어하는데 사용한다.

이를 위해 각각의 지식원을 중심으로 태스크 처리와 관련된 행동을 정의하고, 정의된 행동에 의해 문제를 해결하기 위한 문제 해결 행동 (PSA : Problem Solving Action)을 정의한다. 전역제어기는 특정 지식원과 관련된 문제 해결 행동에 따라 실행을 함으로 해서 태스크의 블랙보드를 진전시킨다.

〈표 1〉 지식원의 문제 해결 행동 리스트

Knowledge Source	PSA	Problem Solving Action
GUI	RIA-1	설비표준 검신
	RIA-2	자재표준 검신
	RIA-3	작업시간 검신
	RIA-4	수주공정 계획조건 기본값 검신
	RIA-5	수주처 택택 검신
	TGA-1	0/1 ⇨ 1, 2 ⇨ 2, 4/6 ⇨ 4, 6' ⇨ 6' 태스크택택검신/태스크기본값 할당
	TGA-2	1/2/3/4/5/6 ⇨ 2 태스크 절단계획조건 검신
		1/4 ⇨ 2 태스크 절단계획조건 검신(수가공)
	TGA-3	2/3/4/5/6 ⇨ 2 태스크 분류 지정
	TRA-1	2 ⇨ 3, 4 ⇨ 4 공단계획 생성/삭제 명령
	TSA-1	2 ⇨ 2, 3 ⇨ 3, 4 ⇨ 4 도면택택별 검증사항 지정
	TSA-2	4/6 ⇨ 5, 6' ⇨ 5', 7 ⇨ 5' 스케줄 생성 명령
	TSA-4	6 ⇨ 6, 6' ⇨ 6' 작업 지시 명령
절단 계획기	TRA-2	3 ⇨ 4, 4 ⇨ 2 공단계획 생성/삭제
스케쥴러	TSA-31	5 ⇨ 4/6, 5' ⇨ 7/6' 스케줄 생성
	TSA-32	6 ⇨ 7, 6' ⇨ 7 작업 지시
교차검사	TCA-1	4 ⇨ 2/4, 6 ⇨ 4/6 자식원수행 결과에 대한 타당성 검사
머신 Client GUI	TPA-1	7 ⇨ 7, 8 ⇨ 8 머신 택택 명령
가공 Client GUI	TPA-2	9 ⇨ 9 가공 택택 명령
도금 Client GUI	TPA-3	10 ⇨ 10 도금 택택 명령
생산현황추적기	SCA-1	7/8 ⇨ 8, 9 ⇨ 9, 10 ⇨ 10 머신 공정 택택 정택도 파악
	SCA-2	7/8/9 ⇨ 9, 10 ⇨ 10 가공 공정 택택 정택도 파악
	SCA-3	8/9/10 ⇨ 10 도금 공정 택택 정택도 파악
	SCA-4	9/10 ⇨ 11 태스크 택택 정택도 파악

특정 블랙보드에 접근 가능한 지식원들이 다수 존재하는데 이를 구분하기 위해 전역 제어기와 지식원들간의 통신을 목적으로 사용하는 제어 메시지의 구성 요소중의 하나인 메시지 타입을 사용하여 구분한다.

4.3 전역 제어기 구성

<표 2>은 전역 제어기의 행동을 정의한 행동 제어 테이블(ACT : Action Control Table)로 태스크의 블랙보드와 메시지 타입을 기반으로 문제 해결을 위한 행동을 정의한 것이며, 전역 제어기가 문제 해결을 위해 참조하는 지식으로 활용한다. 전역 제어기는 특정 태스크 처리에 대한 요청을 받으면 해당 태스크의 블랙보드와 제어 메시지의 메시지 타입을 식별하여 행동 제어 테이블을 참조하여 문제 해결 가능한 행동에 따라 실행을 하며, 또한 <표 1>에서 정의한 지식원의 문제

〈표 2〉 전역제어기의 행동 명세

Blackboard Sublevel	Message	PSA	Blackboard Sublevel	Message	PSA
0	ACM-G0	TGA-1	6	ACM-G3	TGA-3
1	ACM-G1	TGA-1		ACM-S2	TSA-2
	ACM-G2	TGA-2		ACM-S32	TSA-32
	ACM-GP	TGA-2		ACM-SC1	TCA-1
2	ACM-G1	TGA-1	7	ACM-S4	TSA-4
	ACM-G2	TGA-2		ACM-S2	TSA-2
	ACM-G3	TGA-3		ACM-P1	TPA-1
	ACM-R1	TRA-1		ACM-P2	TPA-2
	ACM-S1	TSA-1	5'	ACM-S31	TSA-31
3	ACM-G2	TGA-2		AEM-S31	TSA-31
	ACM-G3	TGA-3	6'	ACM-G1	TGA-1
	ACM-R2	TSA-1		ACM-S2	TSA-2
	ACM-S1	TSA-1		ACM-S32	TSA-32
	AEM-R21	TRA-2	8	ACM-P1	TPA-1,SCA-1/6
4	ACM-G1	TGA-1		ACM-P2	TPA-2,SCA-2/6
	ACM-G2	TGA-2		ACM-P3	TPA-3,SCA-3/6
	ACM-GP	TGA-1	9	ACM-P1	TPA-1,SCA-1/6
	ACM-R11	TRA-1		ACM-P2	TPA-2,SCA-2/6
	ACM-S1	TSA-1		ACM-P3	TPA-3,SCA-3/6
	ACM-S2	TSA-2		ACM-C0	종 료
5	ACM-G2	TGA-2	10	ACM-P1	TPA-1,SCA-1/6
	ACM-G3	TGA-3		ACM-P2	TPA-2,SCA-2/6
	ACM-S31	TSA-31		ACM-P3	TPA-3,SCA-3/6
	AEM-S31	TSA-31		AEM-C0	종 료
6	ACM-G1	TGA-1	11		
	ACM-G2	TGA-2			

해결 행동을 참조하여 문제를 해결 가능한 지식원을 식별한다.

4.4 제어기 명세

전역제어기는 앞에서 정의한 블랙보드, 지식원의 문제 해결 행동 리스트 및 제어기의 행동 제어 테이블을 기초로 하여 제어기의 구문론적 형태를 (그림 6)과 같이 표현할 수 있다. 위의 명세에서 <HEADER> 부분은 태스크의 블랙보드, 제어기, 지식원들간 통신을 위해 사용하는 제어 메시지(CtrlMsg : Control Message) 및 문제 해결 절차를 보유하고 있는 행동 제어 테이블(ACT : Action Control Table)을 정의한다. 제어 메시지는 태스크 처리의 명령을 식별할 수 있는 메시지 타입, 처리할 태스크의 식별자, 문제 해결 능력을 보유한 지식원 식별자를 멤버로서 보유한다. <BODY> 부분은 전역 제어기가 문제를 해결 하기 위한 실행 과정을 표현한 부분으로 메시지 타입과 전역 제어기의 행동 제어 테이블을 기초로 하여 실행을 한다.

```
<HEADER>
BCL : Blackboard Current Level;
CtrlMsg : Control Message; <MID, TID, SID>
 Where
    MID : Message Type
    TID : Current Task ID
    SID : Knowledge Source

ACT : Action Control Table; <BCL, MID, PSA, SID>
 Where
    BCL : Blackboard Current Level
    MID : Message Type
    PSA : Problem Solving Action
    SID : Knowledge Source

<BODY>
Loop
 Wait until a CtrlMsg enters
 Interpret CtrlMsg
 Case MID is
 ACM_G0 : . . . .
    Read BCL
    Update BCL
    Search Next PSA on ACT
    Set CtrlMsg with MID
    Send New CtrlMsg
    Break
 ACM_G1 : . . . .
    :
   .....
 End case
End Loop
```

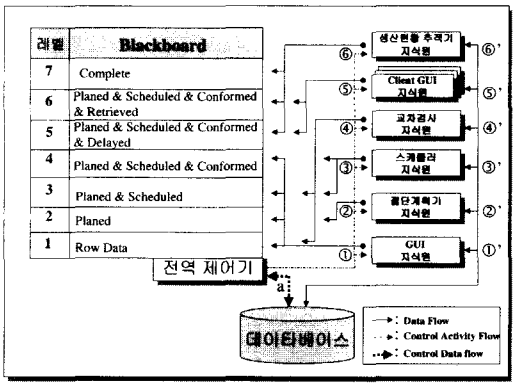
(그림 6) 제어기 명세

위와 같이 구현된 GPSS는 특정 문제 해결을 위한 모듈의 수정이나 확장을 고려할 때, 태스크의 처리 과정과 관련된 블랙보드와 모듈의 문제 해결 행동을 명확히 파악하여 지식원의 문제 해결 행동 리스트 및 전역 제어기의 행동 제어 테이블을 수정함으로 해서 전체 시스템을 대상으로 하지 않고도 수정이나 확장을 보다 쉽게 할 수가 있다.

5. 실 행 예

GPSS의 태스크 처리 절차를 앞에서 정의한 GPSS의 블랙보드 구조를 통해서 살펴보자. (그림 7)은 일반적인 태스크 입력을 받아서 처리할 때, 특정 태스크의 상태에서 전역제어기에 의해 실행이 되는 각각의 지식원을 블랙보드를 중심으로 표현한 것이다.

교차검사는 스케줄 정보에 대한 타당성을 검사하는 것으로, 전체 시스템에서 한번 실행되는 것으로 한정하여 표현하였다.



(그림 7) 블랙보드의 태스크 처리 과정

①의 GUI는 사용자의 요청을 받는 지식원이고, a는 전역제어기가 GUI로부터 태스크 처리의 요청을 받을 때 태스크의 진행상태를 나타내는 제어데이터를 액세스하는 것을 나타낸 것이다. 태스크의 처리가 진전됨에 따라 전역 제어기는 지식원의 실행 순서를 ②, ③, ④, ⑤, ⑥의 순으로 활성화시키며, 활성화된 지식원에 의해 생성된 데이터는 ②', ③', ④', ⑤', ⑥'의 과정을 통해서 데이터베이스에 저장이 된다. 교차검사는 블랙보드 1에서 입력된 납기일과 블랙보드 3에서 생성된 스케줄 결과 사이의 타당성 검사를 수행하는 것을 나타낸 것으로 접근 가능한 블랙보드를 1과 3으로 표현

하였다.

<표 3>은 태스크 처리와 관련되어 문제 해결 행동에 따른 지식원과 태스크의 블랙보드 변환 과정을 나타낸 것이다. 각각의 처리 절차는 일반적인 처리를 기준으로 하였고, 교차검사는 스케쥴 결과가 타당한 것으로 검사 되었다고 가정한 것이다.

〈표 3〉 지식원 처리 절차

Knowledge Source	PSA	Blackboard Transition
GUI	TGA-1	0 ⇨ 1
GUI	TGA-2	1 ⇨ 2
GUI	TRA-1	2 ⇨ 3
절단계획기	TRA-2	3 ⇨ 4
GUI	TSA-2	4 ⇨ 5
스케쥴러	TSA-31	5 ⇨ 6
교차검사	TCA-1	6 ⇨ 6
스케쥴러	TSA-32	6 ⇨ 7
머신 Client GUI	TPA-1	7 ⇨ 7
생상현황추적기	SCA-1	7 ⇨ 8
가공 Client GUI	TPA-2	8 ⇨ 8
생상현황추적기	SCA-2	8 ⇨ 9
도금 Client GUI	TPA-3	9 ⇨ 9
생상현황추적기	SCA-3	9 ⇨ 10
생상현황추적기	SCA-4	10 ⇨ 11

〈표 4〉 전역제어기의 실행 과정 표현

Blackboard Sublevel	Message	PSA
0	ACM-G1	TGA-1
1	ACM-G2	TGA-2
2	ACM-R1	TRA-1
3	ACM-R2	TRA-2
4	ACM-S2	TSA-2
5	ACM-S31	TSA-31
6	ACM-SC1	TCA-1
6	ACM-S32	TSA-32
7	ACM-P1	TPA-1, SCA-1
8	ACM-P2	TPA-2, SCA-2
9	ACM-P3	TPA-3, SCA-3
10	ACM-C0	SCA-4

<표 4>는 전역 제어기가 태스크의 블랙보드를 참조하여 태스크를 처리하는데 사용하는 문제 해결 행동 과정을 메시지를 중심으로 나타낸 것으로, 태스크는 블랙보드 레벨 0에서 태스크 입력을 받아 절단계획, 스케쥴, 교차검사, 작업지시, 태스크처리의 과정을 거쳐 레벨 12에서 작업이 종료가 된다.

6. 결 론

제조 공정의 생산관리시스템에서 사용하는 GPSS는 특정 문제를 해결하여 주는 지식원을 연계하여 사용하는 시스템으로써, 지식원들 사이에서 발생할 수 있는 연동 문제를 해결하여야 하며, 전체 시스템이 보다 나은 시스템으로의 수정 및 확장이 용이하도록 설계되어야 한다. 이러한 기능을 보유한 시스템을 구현하기 위해 보다 범용성 있고, 적용이 용이한 구조중의 하나인 블랙보드 구조를 사용하였다.

블랙보드 구조를 통해서, 시스템의 문제 해결 처리 과정을 여러 단계로 나뉘어진 공유 작업 공간(Shared Work Space)인 블랙보드에 나타내었으며, 구성 모듈간의 통신은 블랙보드를 통해서 이루어진다. 문제 해결의 처리 절차는 블랙보드의 구성요소 중의 하나인 제어기에 지식의 형태로 정의되어 있고, 제어기는 이 지식을 바탕으로 모듈간의 실행 순서를 제어한다. 이와 같이 블랙보드 구조를 적용하여 하위 모듈간의 협동시에 발생하는 연동 문제를 해결하였으며 또한 시스템의 수정 및 확장에 대처 가능한 환경을 구성하였다.

추후의 연구내용으로는 지식 관리 하위모듈을 적용해서 교차검사 지식원을 보다 많은 부분으로 확대 적용하는 것에 대한 연구와 한 호스트에서 사용하는 지식원들을 분산 환경에서의 지식원간의 협력 환경을 제공하는 것에 대한 구현이 필요하다.

참 고 문 헌

[1] E. Rich, K. Knight, 'Artificial Intelligence,' McGraw-Hill, Inc., pp.429-445, 1991.

[2] R. G. Smith, "The Contract Net Protocol : High Level Communication and Control in a Distributed Problem Solver," IEEE Transactions on Computers, Vol.29, No.12, pp.1104-1113, 1980.

[3] 조대호, 고종영, "이종의 소프트웨어 시스템들의 연동을 지원하기 위한 블랙보드 구조의 적용", 한국정보처리학회 논문지, 제5권, 제4호, pp.1234-1245, 1998.

[4] G. Van Zeir, J. P. Kruth, J. Detand, "A Conceptual Framework for Interactive and Blackboard Based CAPP," International Journal of Production Research, Vol.36(6), pp.1453-1473, 1998.

[5] 민병의 외, "분산 환경 기반 개방형 에이전트 구조", 한국정보과학회지, 제15권, 제3호, pp.39-46, 1997.

[6] T. Sandholm, V. R. Lesser, "Coalitions Among Computationally Bounded Agent," Artificial Intelligence 94(1), pp.99-137. 1997.

[7] K. Decker, A. Garvey, M. Humphrey, V. R. Lesser, "Control Heuristics for Scheduling in a Parallel Blackboard System," International Journal of Pattern Recognition and Artificial Intelligence, Vol.7, No.2, pp.243-264, 1993.

[8] F. Klassner, V. R. Lesser, S. H. Nawab, "The IPUS Blackboard Architecture as a Framework for Computational Auditory Scene Analysis," IJCAI-95 Workshop on Computational Auditory Scene Analysis, Montreal, Canada, Auguest 1995.

[9] D. D. Corkill, K. Q. Gallagher, P. M. Johnson, "Achieving Flexibility, Efficiency, and Generality in Blackboard Architectures," In Proceedings of the National Conference on Artificial Intelligence, pp. 18-23, Seattle, Washington, July 1987.

[10] V. R. Lesser, D. D. Corkill, "Functionally-Accurate Cooperative Distributed Systems," IEEE Transactions on Systems, Man, and Cybernetics-Special Issue on Distributed Problem-Solving, Vol. SMC-11, No.1, January, pp.81-96, 1981.

[11] 최중민, "에이전트의 개요와 연구방향", 한국정보과학회지, 제15권, 제3호, pp.7-16, 1997.

[12] L. D. Erman, et al., "The Hearsay-II Speech-Understanding System : Integrating Knowledge to Resolve Uncertainty," Computing Surveys, Vol.12, No.2, June 1980.

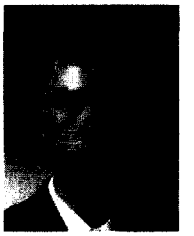
[13] N. Carver, et al., "Sophisticated cooperation in FA/C distributed problem solving systems," Proc. Nat. Conf. Artif. Intell., 9th Anaheim, CA, pp.191-198, 1991.

[14] 김은경, "분산 문제 해결을 위한 개념적 모델에 관한 연구", 한국정보처리학회 논문지 제3권 제1호, pp107-117, 1996.

[15] V. R. Lesser, D. D. Corkill, "The Distributed Vehicle Monitoring Testbed : A Tool for Investigating Distributed Problem Solving Networks," The AI Magazine, pp.15-33, Fall 1983.

[16] A. Poggi, "DAISY : An Object-Oriented System for Distributed Artificial Intelligence," Intelligent Agent, Wooldridge M.J.(Eds), Spring-Verlag, pp.341-354, 1995.

[17] B. P. Zeigler, 'Object and System,' Springer-Verlag New York, inc., 1997.



최 규 성

e-mail : kschoi@peter.skku.ac.kr

1998년 대전대학교 컴퓨터 공학과 졸업 (학사)

1998년~현재 성균관대학교 전기전자 및 컴퓨터공학부 석사과정

관심분야 : 컴퓨터시뮬레이션, 지능형시스템

고 종 영

e-mail : jykoh@peter.skku.ac.kr

1997년 성균관대학교 정보공학과 졸업 (학사)

1999년 성균관대학교 전기전자 및 컴퓨터공학부 졸업 (석사)

1999년~현재 성균관대학교 전기전자 및 컴퓨터공학부 박사과정

관심분야 : 컴퓨터시뮬레이션, 지능형시스템

조 대 호

e-mail : taecho@simsan.skku.ac.kr

1983년 성균관대학교 전자공학과 졸업 (공학사)

1987년 University of Alabama 전자공학과 졸업 (공학석사)

1993년 University of Arizona 전기 및 컴퓨터공학과 졸업 (공학박사)

1993년~1995년 경남대학교 전자계산학과 전임강사

1995년~현재 성균관대학교 전기전자 및 컴퓨터공학부 부교수

관심분야 : 컴퓨터시뮬레이션, 공장자동화, 지능형시스템