

공장모니터링을 통한 수평구조 공장제어시스템의 개발

이 석 회*, 배 용 환**

Development of Heterarchical Control System through Automated Plant Monitoring

Seok-Hee Lee*, Yong-Hwan Bae**

ABSTRACT

The heterarchical structure provides a more attractive solution to the conventional hierarchical structure as the density and level of distribution of computing resources in manufacturing system expands. Tracing the evolution of control structures for automated manufacturing systems, this paper discusses the design principles for heterarchical system to reduce complexity, increase extendability, flexible configurability and suggests a good example of real-time adaptation using the concept of intelligent agent of manufacturing entities and fault diagnosis system.

Key Words : Heterarchical Control(수평적 제어), Distributed Control(분산제어), Manufacturing Entity(제조 엔티티), Intelligent Agent(지능형 에이전트), Fault Diagnosis Control(고장진단 제어)

1. 서론

소비자의 증가되는 품질에 대한 요구, 제품의 다양화, 납기단축, 짧아지는 제품수명 및 생산조직 개발시간, 새로운 기술과 방법, 정보 통신시스템의 발달과 격심한 경쟁력등 이러한 제반사항은 비선형적인 상호작용으로 기업의 주위환경을 더욱 혼잡스럽고 복잡하게 만든다. 이러한 복잡성은 기업이 변화하는 환경에 적응할 것을 요구하고, 지속적인 생산시스템의 개조를 요구한다. 생산구

조는 가치창조를 위한 전개과정에서 인적, 조직적, 기술적 및 정보적인 상호연관성을 기술한다. 생산시스템 설계 과정에서 생기는 문제는 초기 설계시 미래에 일어날 생산시스템의 변형을 생각해야 하며, 또한 앞에서 언급했듯이 제품의 다양화에 기인한 생산시스템의 유연성 있는 운용이다.

과거 컴퓨터 제어 제조시스템 설계자들은 중앙집중식 제어방법을 개선한 트리형식의 조직과 같은 계층적 구조의 개념에 기반을 두고 설계를 하였다.⁽¹⁾ 계층적 시스템

* 부산대학교 생산기계공학과, 기계기술연구소 연구원 (정회원)

** 부산대학교 기계기술연구소 지능생산연구실 전임연구원 (정회원)

의 특징들은 최하위레벨에서 기계들을 향한 하향방향의 명령데이터 흐름과 상위레벨을 향한 센싱데이터 흐름 사이에 존재하는 주종관계, 시스템 전체를 통한 분산적이거나 보다는 구조의 한층 높은 레벨에서 접근되는 공통 데이터베이스들로 구성된다. 계층적 제어구조로 인한 컴퓨터 통합 제조시스템의 복잡함은 점점 더 심화되어지고, 그 결과 개발, 설치, 운용, 유지보수, 변형시 많은 비용이 소요된다.^(2,3)

Hatvany⁽¹⁾는 "고도의 중앙집중과 계층화로 순서화된 시스템은 고정화되고, 미리 정해진 행동절차를 따르는 바로 그 형식주의에 의하여 제약받는다. 그러나 그들의 움직임이 아주 세심하게 최적화되어 있을지라도 조직변화 그 자체에 대한 고유의 저항감과 같은 특성이 파괴적 붕괴에 도달되게 하는 원인이 된다." 라고 언급하면서 계층구조의 견고성에 대하여 지적하고 이것을 개선하는 방법으로 각각의 멤버는 확실한 특권을 얻기 위하여 어떤 규칙을 확정해야 함에도 불구하고 시스템 속에는 상위레벨 컨트롤러가 없는 협동적인 수평구조를 제시하였다. Piche등은⁽³⁾ 시스템의 최상위 레벨 제어가 서로간의 통신할 수 있는 몇몇 조정기 속에 주권이 주어지는 수평구조와 수직구조의 중간체계를 제안하였다. Duffie와 Bollinger⁽⁴⁾등은 시스템 속의 엔티티들 사이가 주종관계가 아닌 그리고 "레벨"에 대한 고려없이 각각 협동할 수 있는 기계제어를 위한 수평제어구조를 제시하였다.

수평제어구조는 정보와 제어를 지역화시킴으로써 복잡도를 감소시키고, 감독레벨들을 제거함으로써 감소된 소프트웨어 개발비용, 높은 유지보수성, 개선된 모듈화와 자기조정, 고장허용 접근을 취함으로써 개선된 신뢰성으로 특징지워질 수 있다.

이 논문의 목적은 이제 까지 변천되어 온 생산시스템의 장단점에 대하여 살펴보고 기존의 계층구조방식의 고정성(rigidity)을 개선하기 위한 방법의 하나로, 고도의 자치권과 통신능력을 지닌 지능형 에이전트를 이용하여 가상의 제조시스템을 설정하여, 이들 간의 통신방법 및 시스템의 운용과 고장에 대한 진단모듈을 접목하고, 제조시스템의 수평구조 제어시스템을 제시코자 한다.

2. 이론적 배경

제조에서 자동화의 사용이 점차 증가함에 따라 증가된 정보의 양에 대한 효율적, 신뢰성 있는 처리가 첨단제조설비의 제어를 위하여 필수적이 되었다. 제어구조를 위하

여 이용가능한 방법론이 많이 생겨났고, 이러한 각각의 방법론들은 신뢰성, 고장허용, 변경가능, 확장가능, 그리고 제어시스템의 재구성가능과 적응가능성으로 요약된다.^(6,8) 여기서, 첨단제조시스템을 위한 제어구조의 변천을 조사하고, 그 구조들 사이의 가장 중요한 차이점을 규정한 다음, 각각 제어구조에 영향을 받는 상한설계 결정방법을 규정하고자 한다. 다음 절에서는 제어구조를 정의하고, 자동 제조시스템을 위한 주요한 설계 결정요소로서 그들의 중요성을 강조한다. 그 다음은 자동 제조시스템 구조를 구성하는데 필요한 요구 조건에 대하여 설명하고, 부수적으로 설계의 변천을 표시하는 기본적인 제어구조를 언급한다.

2.1 중앙집중식 제어구조 (Centralized Form)

제어구조의 발전 형태를 간단히 나타내면 Fig.1과 같다. 여기서 중앙집중식 제어구조는 모든 계획과 정보처리 기능과 전체 셀단위 시스템의 활동을 기록하기 위하여 전역데이터 베이스들의 관리를 수행하는 메인프레임 컴퓨터에 의하여 특징지워진다.⁽²⁾ 기계는 제조환경을 통하여 분산되는 반면에 샵레벨과 셀레벨의 역할은 하나의 특정한 위치에 집중된다. 머신 컨트롤러들은 중앙집중 제어장치들로부터 받은 명령을 수행한다. 명령들은 물리적 제조공정들을 조정하기 위하여 그와 같은 방식을 따른다. 중앙집중 제어컴퓨터들은 그들의 전역제어결정에 사용하기 위하여 shop-floor 센서들과 머신 컨트롤러들로부터 모니터링 정보를 받는다.

중앙집중식 제어구조는 몇가지 장점을 제공한다. 첫째, 완벽한 전역정보에 접근은 훨씬 빨리 이용가능한 최적상태(즉, 생산 스케줄링)를 제공한다. 둘째, 훨씬 적은 컴퓨터를 요구하므로 지원노력이 적다. 마지막으로 전체시

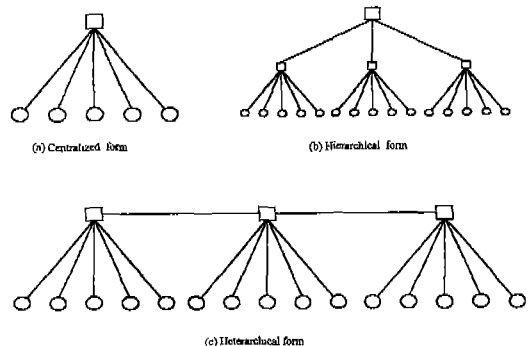


Fig.1 Control structure of automated manufacturing system

스텝 상태정보는 단일소스(중앙제어 컴퓨터)로 부터 회수된다. 중앙집중식 접근방법은 다음과 같은 단점을 가진다. 일반적으로 응답스피드가 시스템이 커짐에 따라 훨씬 많은 온라인, 리얼타임 사용자 즉 기계레벨 컨트롤러들을 관리해야 하므로 분산 제어방식에 비하여 느리다.⁽⁷⁾ 이것은 많은 개별적인 업무를 관리하는 부담에서 기인한다. 두번째 단점은 중앙제어 컴퓨터에 대한 의존도이다. 단일 중앙제어 컴퓨터가 고장이 일어난다면 전체 제조시스템은 더 이상 기능을 발휘하지 못한다. 세번째, 중앙집중 제어구조 소프트웨어에 수정이나 변형은 프로그램이나 전역데이터 구조에 숨겨져 있기 때문에 어렵다.⁽¹¹⁾ 마지막으로 수정성과 확장성은 중앙제어 컴퓨터의 처리능력에 의하여 심하게 제약 받는다.

2.2 계층제어구조 방식 (Hierarchical Form)

컴퓨터기술의 발전과 더불어 중앙집중 제어구조의 단점을 보완하기 위하여 계층구조 형태가 나타나게 되었다. 이러한 형태는 "제어레벨이 프라미드 형태로 정렬된 몇몇 제어 모듈들을 포함한다."는 것으로 특징지어진다.⁽⁶⁾ 이러한 독특한 레벨들은 그들 자신의 목적과 기능을 가진다.⁽⁸⁾ 하부(slave)레벨의 모든 활동들은 상부(master)레벨로부터 지시되고, 하부는 그것에 그냥 따르기만 할 뿐이다. 계층구조의 최상부는 하나이며 전역목표 세팅과 장기적 전략, 즉 성취되어지는 목표들과 선택된 목표에 따른 행동 경로를 조정하고 전체 계층구조를 통일성 있게 짜맞추는 것이다. 제어명령 전달은 top-down방식이고, 상태보고는 bottom-up으로 작동한다. 분류된 제어레벨의 구현은 컴퓨팅 기술의 다양성을 사용함으로써 최상위 레벨은 메인프레임, 중간레벨은 미니컴퓨터 최하위 레벨은 마이크로 컴퓨터로 변화될 수 있다. 중요한 총체적인 결정은 최상위 레벨에서 이루어진다. 이러한 결정들은 하위레벨로 내려갈수록 상세한 사항들이 첨가된다.⁽⁹⁾

이와같이 다른 제어활동들은 변화하는 요구에 부응할 수 있도록 더욱 더 다양화 된다. 계층제어구조를 위한 두가지 참조모델(reference model)들이 제안되었다. 첫째, AMRF(Automated Manufacturing Research Facility)에서 사용한 NBS(National Bureau of Standards)제어구조이고, 두번째 제어구조 모델은 CAM-I(Computer Aided Manufacturing - International)에 의하여 제안된 것으로 AFMCS(Advanced Factory Management and Control System)이다.⁽¹⁰⁾ 앞의 예에서 언급한 것처럼 자동제조시스템을 위한 계층적 제어구조의 선택과 연관된 중요

한 장점이 있을 수 있는데 이것은 점진적인 구현, 여유도, 모듈화로 감소된 소프트웨어 개발시간, 개개의 제어 모듈의 사이즈와 기능, 복잡도가 제한된다는 것, 견고한 구조와 계층적 제어구조의 긴밀한 주종관계의 결합은 통상적으로 빠른 응답시간을 나타낸다. 반면에 많은 단점들도 있다. Cassandras는 로컬 컨트롤러의 계산능력 제한과 상호레벨간 신뢰할 수 없는 통신링크들을 포함하여 계층구조의 몇몇 실제적인 설계제한들을 규정하였다.⁽¹¹⁾ 로컬컨트롤러는 전형적으로 탁월한 계산능력을 가지지 못하는, 반면 상위레벨 컨트롤러들은 상세한 센서정보에 접근하지 못한다. 이것은 실시간 사건발생에 대한 적절한 대응에 문제점이 있음을 나타낸다.

더구나 상호레벨간 통신링크에서 고장은 그들의 직속상부로 단절을 표시하고, 로컬 컨트롤러의 즉각적인 마비를 나타낸다. 링크고장이 훨씬 상부로 갈수록 마비되는 하부레벨 컨트롤러 수가 많아짐을 알 수 있다.⁽¹²⁾ 역시 수평적 접근을 지지하는 Duffie등도 임의의 계층적 제어구조에 몇몇 단점을 제시하고 있다.⁽⁶⁾ 첫째, 계층구조는 초기 설계단계에서 견고하게 고정되는 경향이 있다. 이것은 미래의 보이지 않는 변형을 매우 어렵게 만든다. 둘째, 비록 구조적인 수정이 가해질지라도 모듈 상, 하부의 부수적인 지식이 요구된다. 이와같은 상호의존성은 반대로 수정능력과 확장능력에 영향을 미친다. 모듈화의 정도가 높아짐에 따라서 그리고 모듈들간의 연결정도가 낮아짐에 따라서 시스템 구현은 용이해지고, 소프트웨어의 디버깅 횟수, 테스트 및 정비횟수와 복잡도는 감소한다.⁽¹³⁾

마지막으로 특히 고장과 관계된 수많은 복잡한 상호관계들은 시스템 개발에 상당한 비용이 소요되고, 시스템 복잡도가 증가할수록 고장허용 결과들을 성취하기 위하여 시스템 속으로 명백하게 프로그램 되어져야 한다.⁽¹⁶⁾ 증가된 소프트웨어 복잡도는 논리에러가 시스템 안으로 개입되어서, 시스템 작동에 치명적인 영향을 주고, 그 원인을 쉽게 찾을 수 없다는 것을 의미한다.⁽¹⁴⁾ 그 결과 시스템의 신뢰성을 저하시킨다.

2.3 수평제어구조 방식 (Heterarchical Form)

최근에 앞에서 언급한 두가지 제어구조 형태와 연관된 단점을 줄이기 위해서 몇몇 저자들에 의하여 수평구조(heterarchical)접근 방법이 제시되었다.^(1,3,5,6,15,16) 수평제어구조들은 계층구조에서 발견된 주종관계가 아니라 다른 엔티티들과 통신하는 지역적인 자치엔티티들로 분산되어져 있다. "분산 계산(distributed computing)영

역이 수평제어구조 원칙들을 위한 몇몇 정의들을 가능하게 하는 원천이다.⁽⁶⁾ 네트워크 통신에서의 진보가 수평제어구조 형태를 가능하게 만들었다. 수평제어구조의 핵심은 충분한 지역적 자치권과 전체 결정에 상호협동할 수 있는 시스템 구현이다. 상호협조 체계에서 외부적인 상위레벨은 존재하지 않는다.⁽⁵⁾ 상호협조 체계에서는 자연적으로 시스템속에서 제어 요소들 사이에 주종관계가 없다.⁽⁶⁾

완전한 지역적 자치는 전체적인 정보는 최소화하거나 제거하는 것을 요구한다. 이것은 어떤 지역적인 데이터베이스들이 시스템 내부에서 유지된다는 것을 의미한다. 그러나 이와같은 데이터베이스들은 상당한 양의 상세정보를 포함해야 한다. 셀레벨 수평구조에서 Rana와 Taneja는 각각의 셀콘트롤러는 기준생산에 관한 데이터베이스, 제조지식 데이터베이스, 개고관리 데이터베이스, 공정계획 데이터베이스, 작업장 상태 지식데이터베이스와 같은 몇몇 지역적인 데이터베이스들을 가져야 한다고 제안하였다.⁽⁵⁾

분산 네트워크를 통하여 생산요소들은 관리/영업 정보서브시스템에 접근가능하고, 각 요소에서 직접 필요한 자료들을 사용할 수 있게 한다.⁽¹⁷⁾ 이와같은 특징들은 충분한 지역적인 자치권이 상호 협동작용을 통하여 유지되는 것을 보장해야 하며 이러한 요구들을 만족하는 하나의 협상 규약이 계약 네트워크 프로토콜(Contract Network Protocol)이다.⁽¹⁸⁾

수평형 제어구조가 제공하는 장점은 주로 완전한 국부적 자치권의 성취로부터 얻어진다. 첫째, 감소된 소프트웨어 복잡성은 강화된 모듈성, 모듈들(제어요소)사이의 감소된 연관성, 절대적인 고장허용등으로 인하여 실현된다.⁽⁶⁾ Duffie와 Piper가 중앙집중, 계층, 수평제어구조 접근을 비교할 때, 계층구조에 비하여 약 1/10, 중앙집중식에 비하여 1/2정도의 소스코드가 줄어들었다. 이 결과는 운영체제의 복잡성이 셀레어 소프트웨어 개발자들에게 부담을 주었기 때문이다. 개발자들은 단지 간략하고, 지역적인 소프트웨어 모듈과 데이터베이스만을 만든다.⁽¹⁾ 소스코드의 라인수는 특히 소프트웨어 복잡도를 보여주는 좋은 척도이다.⁽¹⁹⁾ 앞에서 언급한 것처럼 모듈사이의 강화된 모듈성과 감소된 연관성으로 인하여 복잡도는 줄어들고, 개발과 보수(변형, 확장)가 단순하게 된다. 절대적인 고장허용은 완전한 지역적인 자치권의 긴밀한 구현에 기인하여 성취될 수 있다. 자치적인 기능을 수행하는 시스템은 하나 혹은 그 이상의 제어요소들이 이상기능을

수행 할 때에도 비교적 안전하다.⁽⁶⁾ 또한 협동적인 문제해결을 위한 각 스테이션간의 의사결정 프로토콜과 방식들은 하나 혹은 그 이상의 상호협동 엔티티들의 고장에도 불구하고 연속적으로 작동할 수 있게 한다.^(5,18) 협동적 의사결정 프로토콜과 방법들을 변경할 수 있다는 것은 재구성과 적응성을 의미하며 전체적인 최적화 기법이 항상 요구되기도 한다. 기계가 물리적으로 자동제조시스템에서 분리되거나 재연결될 때 그들의 제어실행을 책임지는 제어요소 엔티티는 비슷하게 제어시스템에서 분리되거나 재연결된다. 협동 프로토콜들은 이것에 응답하도록 설계되어져야한다.⁽⁶⁾ Table 1에 각 공장제어구조의 장단점을 나타내었다.

3. 수평구조 공장제어시스템 설계원칙

고도의 변형성과 고장허용을 보장하면서 상호협동하는 자치적인 엔티티들을 가진 시스템을 만들려는 쪽으로 수많은 설계 원칙들이 개발되고 적용되었다. 이 원칙들은 다음과 같다. 첫째, 엔티티들은 각자 고도의 지역적 자치권을 가지고 있어야 하고, 엔티티들 사이에 주종관계가 존재해서는 안되며, 엔티티들은 가능할 때마다 다른 엔티티와 상호 협동해야 한다. 둘째, 엔티티들은 다른 엔티티들이 그들과 상호작용하지 않는다고 가정하며, 엔티티들은 가능한 한 오랫동안 관계설정을 미루어야 하고, 가능한 한 빨리 관계를 끝내야 한다. 셋째, 엔티티들은 모든 자원들에게 접근의 동등한 권리를 가지면서, 작동에 있어서 독립성을 가진다. 마지막으로 엔티티들은 전체시스템의 프로토콜-규칙들에 엄격히 순응한다. 이와같은 규칙하에서 설계자는 규칙들을 설정하고 코드화한다. 시스템을 구성하는 엔티티들에 의한 정해진 규칙들은 통신을 위한 특권을 얻기 위해 반드시 준수되어야 한다. 각각의 엔티티는 두가지 목표를 갖는다. 첫째는 지역적인 자치성과 유연성, 예기치 않는 사건들에 대응된 고도의 지역적 지능을 강조하는 것이고, 두번째는 예기치 않는 사건들에 대하여 최적 전체시스템 운용을 강조하는 것이다.

분산 기계제어 시스템의 설계를 강조함에 있어서 Duffie등은 Fig.2에서 보인 것과 같은 설계접근 방법을 제시한다. 여기서 시스템기능 요구조건들이 분석되고 일련의 준-독립적인 통신 엔티티들로 분할하고 설계과정을 위해 비교적 약한 상호작용을 가진 준-독립적인 성격을 갖도록 한다. 엔티티들간의 모든 통신은 네트워크상에서

전달되는 메시지 형태를 취하고, 물리 시스템들의 구성은 시스템 내에서 엔티티들에 투명성을 보장해야 하고, 엔티티들은 다른 엔티티들의 상세 위치에 영향을 받지 않는 탄력과 활성이 구비되어야 한다.⁽²⁰⁾ 이는 나아가서 전역정보의 최소화라는 원칙을 뒷받침하고 있다. 전역정보는 하나의 유일 엔티티에 대하여 지역적이 아닌 그 어떤 정보를 나타낸다. 전역정보와 엔티티들 사이의 복잡한 관

Table 1 Advantages and disadvantages of control structures of automated manufacturing systems

Type	Advantages	Disadvantages
Centralized Form	<ul style="list-style-type: none"> - access to global information - global optimization possible - single source for system status information 	<ul style="list-style-type: none"> - slow and inconsistent speed of response - reliance on single control unit - difficult to modify control software
Hierarchical Form	<ul style="list-style-type: none"> - gradual implementation, redundancy, and reduced software development problems - incremental addition of control possible - possibility of adaptive behavior - allowance for differing time scales - fast response times 	<ul style="list-style-type: none"> - computational limitations of local controllers - increased number of inter-level communication links - difficulties with dealing with dynamic adaptive control - difficulty of making future unforeseen modification - connectivity problems - increased difficulty of control system
Heterarchical Form	<ul style="list-style-type: none"> - full local autonomy - reduced software complexity - implicit fault-tolerance - ease of reconfigurability and adaptability - faster diffusion of information 	<ul style="list-style-type: none"> - few standards for communication protocols or operating systems - only local optimization possible - requires a high network capacity - lack of availability of software

계들의 존재로 인하여 공장제어시스템 수정시 많은 비용을 요구하고, 논리적 오류로 인해 현장에서 시행착오 없이 이용이 가능하지 않다. 하나의 시스템내에서 전역정보의 제거는 엔티티들 내에 고장의 억제, 다른 엔티티들의 고장으로 부터 빠른 복구, 시스템 모듈화, 변화에 대한 유연성, 확장성, 복잡도 감소, 개발비용 저감등의 효과를 얻을 수 있다. 한 곳에 모아 두는 전역정보는 편리하지만 위험할 수 있기 때문이다.

또한 전체적인 긴밀도는 최소한의 전역적인 정보와 주종관계 부재에서 지역적 행동만에 의하여 성취되어야 한다. Cammarata⁽²¹⁾ 등은 분산시스템에서 고장허용을 성취하는 열쇠는 분산 에이전트들이 주어진 업무를 해결하기 위하여 잠재적으로 자치적인 어떤 해결능력을 가져야 한다고 주장했다. 예를 들면 개개의 엔티티는 하나의

고장을 탐지하기 위하여 시스템 각각의 응답에 대처하여야 하고, 또한 복잡한 문제해결을 위하여 다른 엔티티들로 하여금 그렇게 하도록 요구할 수 있다.

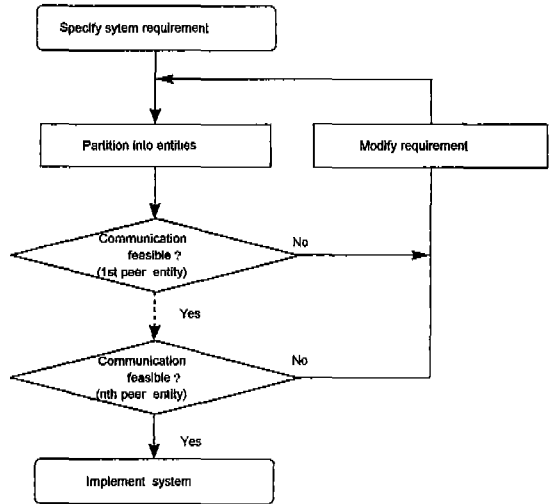


Fig. 2 Heterarchical control system design flow

4. 지능형 에이전트 기반 협동적 수평제어구조

몇몇 실험적인 제조시스템이 수평제어 시스템의 평가와 설계, 설치연구에 사용하기 위해서 대학교 실험실 기반에서 구축되었다. 기본적 장비로 구축된 시스템은 Fig.3에 보인 것과 같은 하드웨어 및 가상기계들로 구성되어 있으며, 앞 절에서 주어진 설계 원칙들이 시스템의 요구조건들을 일종의 이기종 기능들과 통신 엔티티들로 분할하고 분해하는데 적용되었다. 시스템을 구성하는 주요 엔티티들은 자원 엔티티들(빌링, 세척검사, EDM, CMM), 물류취급 엔티티들(로봇, 팔렛이동기), 부품엔티티와 고장진단 엔티티들을 포함하고 각각의 엔티티들은 분산된 지능형 에이전트에 의하여 운영된다. 이와같은 엔티티들은 LAN환경 하에서 여러대의 마이크로컴퓨터에 분산되어 있다. 시스템속에서 엔티티들이 제어하는 생산자원들은 이러한 마이크로컴퓨터에 연결된다.

시스템속에서 엔티티들의 수평적인 특성 때문에 엔티티들과 그들의 연결이 균일한 형태로 취급되도록 허락하는

통일된 엔티티 구조가 요구된다. 각각의 엔티티들은 3가지 요소로 이루어져 있다. 시스템내에서 다른 엔티티들의 통신을 제어하면서 문제를 해결해 나가는 지능형 에이전트

하고 어떤 정보가 사용가능한지를 결정하는 능력이 추론이며, 계획은 어떤 목표를 이루기 위하여 어느 에이전트의 기능들이 조합되어 이용되어야 하는가를 알아내는 능력이다. 에이전트는 블랙보드 구조를 기본으로 하고있다. (22) 개방형 에이전트 구조에서는 클라이언트 개념에 해당하는 지능형 에이전트들이 블랙보드에 해결할 작업을 기록하는 식으로 진행된다. 블랙보드는 부모에이전트 (parent agent)와 자식에이전트(child agent)에 의하여 제어된다. 부모에이전트의 역할은 일종의 글로벌 데이터를 저장관리하고, 어느 에이전트가 어떤 일을 할 수 있는지 알아내고, 분산적으로 블랙보드를 포함하며 이를 관리하고, 모든 교류는 블랙보드를 통하여 이루어진다. 에이전트들간의 교류를 위한 메세지(Inter Communication Language :ICL)는 UNIX에서 사용되는 IPC(Interprocess Communication)를 통해 이루어진다. 부모에이전트의 주된 임무는 ICL을 분석하여 이를 해결해 줄 수 있는 자식에이전트에게 전달해 주는 것으로 일종의 중계자 역할이며, 메세지에 따라서 임의의 방향으로 진행된다. 각각의 에이전트들은 사용자나 다른 에이전트의 요구를 만족시켜 주기 위하여 정보를 주거나 어떤 행위를 할 수 있다. 또한 조건이 만족되는지를 감시하기 위하여 트리거를 설치할 수 있다. 트리거란 조건부와 행위부로 나누어지는 데이터의 형태이다. 트리거의 조건부는 일정시간을 주기로 만족여부를 결정하기 위하여 조회된다. 트리거 조건부는 블랙보드메세지, 블랙보드데이터 또는 특정 에이전트의 조건에 관련된 것들이다. 이와같은 필요성을 만족시키기 위하여 표준형 엔티티구조에서 지능형에이전트와 문제해결을 위한 전문가시스템셀의 작동구조는 Fig.4와 같다.

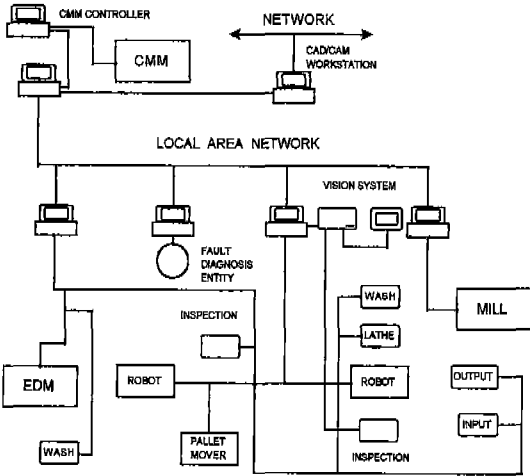


Fig. 3 Experimental system for heterarchical control and monitoring

트(Intelligent Agent : IA), 엔티티와 관련된 외부 하드웨어와 상호작용하는 통신기(communicator), 그리고 필요에 따라 경험적 추론을 담당하는 지역적 전문가시스템들로 구성되어 있고, 이러한 지역적 전문가시스템은 지능형에이전트와 필요시 다양한 시뮬레이션 모델에 의하여 추론을 행하면서 문제를 해결해 간다.

4.1 지능형 에이전트

엔티티 내부에서 문제해결의 핵심요소인 에이전트(agent)는 네트워크에 연결된 원격 컴퓨터에서도 수행될 수 있도록 짜여진 프로그램으로서 다른 에이전트들에게 자신의 기능을 제공할 수 있고, 사용자로부터 구체적이지 않은 작업지시를 받고도 해당되는 작업을 수행하며 또한 사용자가 항시 수행과정을 일일이 지시하지 않아도 되도록 한다. 또한 에이전트는 자료에 의한 추정, 통신, 추론, 계획 같은 4가지 기능을 가지고 있는데 이는 스스로 특정의 정보를 항상 감시하며, 자신이 어떤 일을 언제 수행하여야 할지를 결정하는 능력과, 사용자와의 대화뿐만 아니라 다른 에이전트들 간에도 정보를 교환하는 능력이다. 또한 자신이 활동을 개시할 조건이 만족되었는지를 파악

에이전트 시스템의 문제해결 과정은 다음과 같다. 해결되지 않는 문제는 다른 에이전트 시스템으로 전달된다. 이를 위하여 블랙보드들이 계층화를 이루며 연결되어 있다. (23) 하나의 에이전트 시스템이 네트워크상에 분산 배치되어 수행될 수 있듯이 여러개의 에이전트시스템들로 분산배치되어 서로 교류할 수 있다. Fig.5 (a)는 일반적인 지능형 에이전트의 구조이고, (b)는 분산된 각 시스템의 내부에 있는 계층형 문제해결을 보이는 것이다.

4.2 통신기 (Communicator)

엔티티의 네트워크 통신기능들의 모든 것은 통신기에 의하여 수행된다. 통신기는 엔티티로 하여금 통신네트워크를 통하여 다른 엔티티들과 비동기적으로 메세지를 교

환하도록 허락한다. 통신기는 다른 엔티티들 내부에 존재하는 에이전트의 수행, 고장발생 그리고 엔티티의 상태에 대한 정보요구와 연관된 메시지를 받는다. 통신기는 하드웨어적인 기계와 엔티티들 사이에 수행과 연관되어 발생된 메시지의 전달과 고장발생 여부에 관여하는 전문가시스템에 출력을 감시하면서 요구된 메시지를 전달한다. 통신기는 다른 엔티티로부터 메시지의 송수신이 즉각적으로 응답하고 다음에 바로 리셋 시키는 자발적으로 취급되는 사전지향적이다. 엔티티내부의 에이전트 상호간의 기능 수행에 있어서 가장 중요한 것은 에이전트들간의 통신언어(Interagent Communication Language: ICL)이다. 이것에 대한 설명은 형태와 내용이라는 관점으로 나누어야 한다. 형태적 측면에서 본다면 문제해결요구, 요구수행, 문제공지 등의 3가지 형태이다. 지능형 에이전트의 요구에 의해 다른 엔티티의 내부나 자기 엔티티에 있는 블랙보드에 수행요구를 기록한다. 그러면 블랙보드에 기록된 내용에 따라 다른 에이전트에게 정보를 전달해 준다. 이때 다른 컴퓨터상에 존재하는 정보전달은 정해진 TCP/IP프로토콜을 사용하는 소켓(socket)통신을 통하여 이루어지며 동일한 컴퓨터상에 존재하는 에이전트들 간에는 메시지큐나, 메시지 관련 통로와 같은 파이프를 통하여 메시지를 전달한다.

4.3 지역적 전문가시스템 (Local expert system)

지역적 전문가시스템은 분산된 문제해결과 다중 고장진단과 같은 경험적인 추론을 지원하기 위한 것이다. 이것은 통신기를 통하여 다른 엔티티들과 메시지를 교환할 수 있고, 적절한 고장복구 기능을 회수하여 기계에 조치사항을 송신할 수 있다. 또한 고장감지와 진단, 복구, 리포팅

기능을 수행할 수 있다.

본 연구에서 하나의 예로 제안된 고장진단 모듈속의 전문가 시스템셀은 진단과정의 순서를 결정하는 경험적(heuristic)지식으로 이루어져 있고, 이것은 고장종류와 고장의 속성으로 이루어져 있다. 본 고장진단시스템은 다수의 전문가시스템이 상호협조하는 방식으로 이루어져 있는데 메인모듈에서의 전문가시스템 셀의 역할은 전체 시스템의 증상과 이것에 관련된 부속모듈의 고장정보를 담고있다. 메인모듈의 진단추론은 가장 확률 높은 고장 부속모듈을 찾는 것이다. 이렇게 하여 후보(candidate)로 확정된 부속모듈의 전문가시스템셀은 부속모듈의 고장진단을 위하여 새로운 프로세스로 생성된다. 부속모듈 전문가시스템셀도 메인모듈과 마찬가지로 휴리스틱에 의한 방법으로 진단을 수행한다. 즉 그 모듈에 속해 있는 부품요소의 고장종류에 따른 고장속성의 매칭을 if-then의 추론규칙에 의하여 추론해 나가는데 여기서 고장속성을 알기 위하여 수식모델, 정성적 모델, 신경모델 등으로 현재 시스템 상태 데이터를 보내서 그 고장속성 진위여부를 조사한다. 이렇게 휴리스틱 기법과 모델진단의 통신을 통하여 고장원인에 해당되는 고장부품요소가 밝혀지면 시스템은 이것의 진위를 확인하기 위하여 운영자에게 메시지를 전달해야 하는데 우선 고장부속모듈과 고장부품

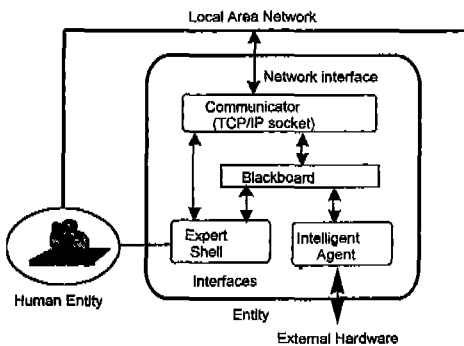
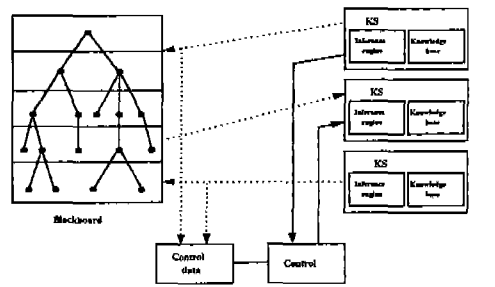
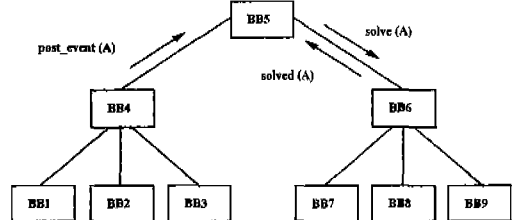


Fig.4 Generic entity structure for heterarchical control



(a) Agent structure



(b) Problem solving by blackboard

Fig.5 Agent structure and problem solving by blackboard

요소를 출력하고, 예상되는 고장원인을 출력한다. 또한 이것에 대한 처치를 컴퓨터화면을 통하여 출력하고, 이제까지의 추론과정을 표시한다. 또한 현재의 고장부품 요소의 도면과 정비이력을 화면을 통하여 출력함으로써, 수리를 지원할 수 있도록 한다.

4.4 시뮬레이션 로봇 엔티티

시뮬레이트된 로봇엔티티는 네트워크상에 연결된 지능형 에이전트 형태를 통하여 주로 외부와 통신을 하고 내부적으로 복잡한 제어상의 문제를 해결한다. 여기서 일반적으로 외부로부터 들어오는 파드엔티티들의 요청에 따라 특정 위치에 있는 로봇은 서비스 요청을 받게되고 현재 부품위치를 인식하여 좌표값을 계산하고, 셀콘트를 러로부터 전해진 이동좌표를 받아서 실제 각 기구부의 움직임을 계산한 후 제어부로 명령을 내보내는데 주로 로봇트의 제어를 담당하는 부분은 크게 세부분으로 나누어질 수 있다. 첫째는 외부 엔티티들과의 통신을 담당하는 로봇 통신 에이전트와 실제 로봇트의 좌표를 입력받아서 로봇트 각각의 실제적인 위치와 각속도등을 계산하는 경로계획부, 이와같은 위치와 각속도 정보를 실제 기구부에 전송하여 분산 피드백 제어를 수행하는 운동축 제어부로 구성된다. 이와같이 시뮬레이트된 로봇엔티티가 Fig.6에 나타나있다. 로봇 통신 에이전트는 초기에 가공셀에서 전해오는 스트링 형태의 메세지를 받아서 해독하여 현 위치와 이동할 로봇트체적 정보를 얻기위해 경로 계획부에 전송하면, 여기서 현재의 로봇트 위치를 고려하여 로봇트 각 기구부가 이동할 장소의 위치와 각속도를

계산하여 이 정보를 운동제어부에 전송해 주면 그 하부에 있는 각각의 해당 축제어부로 분산적으로 동시에 보내어 실제 로봇트를 제어하게 된다.

4.5 고장진단 엔티티

수평제어구조 제조시스템의 고장허용과 효율적인 진단을 위하여 에이전트 기반 고장진단 엔티티가 제안되었다. 일반적으로 기계고장트리(Fault Tree : FT)의 구성은 각 기계요소들 사이의 구조 및 기능, 동정에 관한 구성으로 이루어진다. 또한 가장 최상단에는 일반적으로 시스템 전체고장을 목표로 삼고, 그 하부에는 각 부속모듈(sub-module)로 연결되고, 최하부에는 각 기계요소의 종류와 그에 따른 고장요인으로 이루어진다. 그러나 컴퓨터화된 진단시스템을 구축하기 위해서 컴퓨터속에 시스템에 대한 구조적인 정보와 기능적정보, 시스템 동정에 관한 정보를 컴퓨터에 이식시키고, 시스템의 각 부분 센서로부터 들어오는 신호에 따른 증상과 예상되는 동정과 의 차이점을 바탕으로 고장요소와 그 부위를 판단한다. 현재의 연구는 이러한 증상과 이상형태를 지식베이스 형태로 컴퓨터에 저장해 두었다가 패턴매칭에 의하여 휴리스틱(heuristic : shallow knowledge)에 의하여 고장을 추론하는 방식을 주로 채용하고, 최근에는 지식베이스에 존재하지 않는 고장에 대한 진단을 수행하기 위하여 모델기반 고장진단기법이 소개되었고, 이 두가지 형태를 조합한 통합진단시스템(Integrated Diagnosis System)에 관한 연구가 진행되고 있다. 그러나 시스템이 구조적, 기능적 복잡해 짐에 따라 좀더 정확하게 표현할 수 있는 다중 고장진단을 위해 고장트리를 기초로 하여 신경망 모델을 내장한 지적 진단 시스템이 개발되었다.⁽²⁴⁾

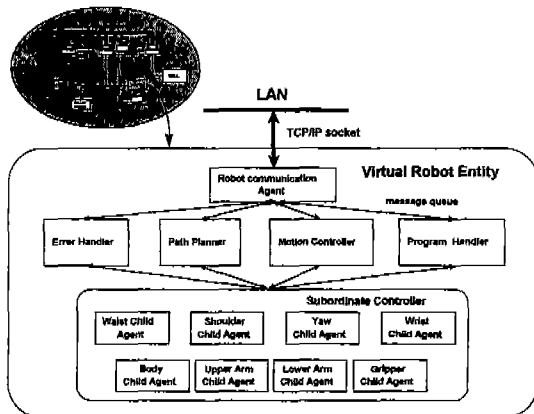


Fig. 6 Simulated robot entity for heterarchical control

4.5.1 고장진단 엔티티의 구성요소

고장진단 에이전트들의 내부구성을 주요한 부분으로 나타내면 Fig. 7과 같다. 고장진단 엔티티 내부는 다음과 같은 주요 모듈들로 구성되어 있는데, 전문가 시스템 셀과 통신기는 앞에서 언급한 것과 같은 특성을 가지고 있으며, 이외에 다음과 같은 특징적인 내부모듈을 가지고 있다.

1) 모델기반 고장진단 모듈

센서로부터 시스템에 관한 동정을 표시하는 정보가 들어오면, 전처리모듈과 신호처리모듈에서 처리된 데이터가 입력데이터가 된다. 그 다음 각 서브모듈과 각 부품요

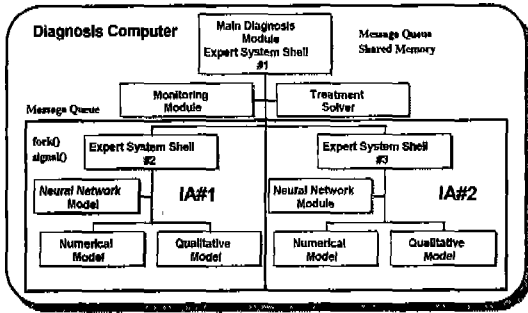


Fig. 7. Fault diagnosis entity for heterarchical control system

소의 동정과 상태를 표현하기 위한 각종 수식모델(예: 피로모델, 마모모델, 열전도모델 등...)과 같은 것들이 작동되고, 수식모델의 시뮬레이션 결과값에 따라 현재의 상태를 파악할 수 있다. 진단시스템이 현재의 시스템의 상태보다 더 정확한 데이터를 요구할 때 구동된다. 또한 정성적 모델은 일반적으로 수식표현으로 다루어지지 않는 물질계에 대해, 현재 상태의 정확한 수치값 보다는 현재의 물리적 상태표현을 중심으로 물리적 법칙을 지식베이스 형태로 저장하고 있다가 요구시 마다 활성화하여 각 입력조건에 따른 물리적인 상태를 상위로 전달한다. 신경모델에서는 부속시스템 또는 부품요소에서 고장이 정확하게 각 경우마다 센서에서 나타나는 응답과 일치하지 않는 경우에 고장사례를 기초로 하여 그때의 특정부위의 증상(센서신호)이 서로 대응이 되도록 한다. 베어링, 기어 등의 여러가지 요소고장 종류에 따른 신호패턴을 FFT(Fast Fourier Transformation)에 의하여 전처리한 입력데이터를 입력받거나, 그 고장의 감각형태에 따라 자기 다른 신경망 모델을 사용한다.^[35] 예를 들어 진동 센서로부터 들어오는 패턴은 백프로게이션을 이용한 다층퍼셉트론을 사용하고, 시각 센서정보로부터 들어오는 시각패턴은 홉필드 신경망을 이용한다. 신경망모델은 요구되는 순간에 특정의 사례고장패턴에 대한 정보를 데이터베이스에서 읽어들이어서 학습을 시작한 후, 입력데이터를 읽어들이어서 패턴매칭을 시도한다.

2) 에이전트 메시지 설비

UNIX시스템은 IPC설비((Inter Process Communication facility)라는 새롭고 다양한 프로세스간 통신방법을 제공하는데, 이러한 IPC설비를 통하여, 단위작업들이 서로 협력하는 시스템을 프로그래밍할 때 다양한 방법론

적 접근을 가능하게 한다. 이러한 설비는 메시지 전달을 위한 설비, 그리고 프로세스 생성과 소멸을 위한 설비와 프로세스의 동기화를 위한 세마포어 설비, 데이터의 공동사용을 위한 공유메모리 설비로 나눌수 있다.

본 연구에서 상하위 모듈과 동등레벨의 요소들 상호간의 협동과 추론을 위해서는 정보의 전달이 필요하다. 일반적으로 전달되는 정보의 형태는 센서로부터 입력값과 신호처리단계를 거친 패턴값, 그리고, 시각정보와 각각의 모델들로부터 상위레벨로 전달되는 판단결과등의 문자열에 해당된다. 본 연구에서 이용한 메세지 설비는 UNIX 시스템의 IPC 설비중 메시지큐이다. 이는 하나의 상위 모듈은 하나의 메세지큐를 이용하여 다수의 하위모듈과 통신 할 수 있다. 다수 프로세스간 메세지 전달은 메세지큐 생성, 메세지 수신, 메세지 송신, 3가지 과정을 거쳐 이루어진다. 또한 공유메모리 설비는 고장진단 시스템에서 메인 진단모듈로부터 각 서브모듈로 센서 입력값이 신호처리되어 메세지큐를 통하여 전달된다. 그러나 일반적으로 특정의 값이 각각의 다른 서브모듈이나 기타 다른 요소에 공통의 관심이 있거나 입력이 되어야 할 때 프로세스의 동기화와 효율을 위하여 공유메모리를 사용한다.

3) 에이전트 생성기법과 프로세스 제어설비

연구에 사용된 운영시스템인 UNIX시스템에서는 프로그램을 통하여 프로세스를 생성하여야 한다. 두개의 프로세스(부모프로세스, 자식프로세스)를 생성하여 자식프로세스는 이전의 부모데이터를 상속하여 공유하고, 그 이후에는 자신의 독자적인 프로세스를 유지한다. 그러나 고장진단시스템의 정밀진단을 위한 추론과정에서 수식모델이나 신경망모델등을 기동시켜야 하는데 이때 자식프로세스는 기존의 부모프로세스와는 다른 데이터와 알고리즘을 가져야 한다. 따라서 이때 자식프로세스는 자신의 프로세스를 다른 모델 프로세스로 대체시켜야 한다. 이와 같이 다수의 진단모듈을 연속적으로 만들기 위해서는 시스템 루틴을 사용하는데 이들이 fork()와 exec()이다. 또한 프로세스의 생성, 소멸, 동기화가 필요한데 이들의 조합을 번갈아 사용하면 가능하다.

5. 결 론

본 연구에서 제시된 지능형 엔티티 기반 수평제어구조는 다른 시스템 엔티티속에서 지역적 전문가를 가지고 분산된 문제해결에 협동할 수 있도록 허락함으로써 문제해결 능력과 유연성 있는 추론을 할 수 있다. 독립성, 통

신, 시스템내부에서 문제해결기능을 가지면서 높은 성능과 낮은 개발비용, 개선된 변경성, 높은 고장허용성이 성취될 수 있으므로 한층 크고 복잡한 제조시스템의 설계와 설치에 유효할 것으로 생각된다. 또한 시스템속으로 결합가능한 사용자 목적에 알맞은 전문가 시스템셀은 심각한 고장들이 일어났을 때, 시스템 불능이나 성능저하상태로 부터 구해내기 위하여 아주 효과적으로 사용될 수 있다. 시뮬레이션들은 실제 시스템 제어와 결합될 수 있고, 하나의 시스템에 시뮬레이트된 엔티티로서 설치될 수 있다.

향후 수평적 접근이 장래의 유연생산 시스템과 로봇셀의 제어를 위하여 매우 유용할 것으로 생각된다. 이와 같은 접근방법이 충분히 가능해지기 위하여 여러 관련분야에 많은 연구가 수행되어야 한다. 다중프로세스의 존재, 통신, 시스템속에서 독립적인 지능을 가진 에이전트는 시스템 제어와 조직에 새로운 제어기법을 제공할 수 있다. 제시된 수평제어원칙들은 낮은 단가, 낮은 복잡도, 높은 고장허용도가 얻어질 수 있는 수단들을 제공한다. 지능도와 복잡도가 높은 미래의 자동화 시스템들은 수평제어구조를 통하여 설계 및 변경이 효율적이고, 변경, 보수가 용이하게 만들어져야 한다.

본연구에서 제시된 지능형 에이전트를 이용한 진단은 통신모델을 통한 복합시스템에서의 한 부속시스템이나, 모듈, 혹은 복합요소의 동정을 기술하는데 각각 사용되어 시스템 운용시 국부적인 고장에 대한 시스템의 운용성을 높인다.

후 기

본 연구는 교수 국비해외파견 일반연구로 Wisconsin-Madison 대학에서 수행한 것이며, 실험장비와 연구에 협조해 준 N.A.Duffie 교수와 교육부 및 학술진흥재단 관계자 여러분께 감사드립니다.

참 고 문 헌

1. J.Hatvany, "Intelligence and cooperation in heterarchical manufacturing systems", Annals of CIRP, Vol.14, No.1, pp.5-10, 1985.
2. N.A. Duffie and R. Piper, "Non-hierarchical control of a flexible manufacturing cell",

- Robotics & Computer Integrated Manufacturing, Vol.3, No.2, pp.175-179, 1987.
3. P.Piche, M.Kreiger and M.Santoro, "Oligarchy, a control scheme for flexible manufacturing systems", in: Proc. 2nd IASTED Int. Sym. of Robotics and Automation, Lugano, Switzerland, pp.35-39, 1983.
4. N.A.Duffie and J.Bollinger, "Distributed computing systems for distributed industrial control", Annals of CIRP, Vol.29, No.1, pp.357-362, 1979.
5. S.P.Rana, S.K. Taneja, "A Distributed Architecture for Automated Manufacturing Systems", International Journal of Advanced Manufacturing Technology, Vol.3, No.5, pp.81-98, 1988.
6. N.A.Duffie, R.Chitturi, J.Mou, "Fault-Tolerant Heterarchical Control of Heterogeneous Manufacturing System Entities", Journal of Manufacturing Systems, Vol.7, No.4, pp. 315-327, 1988.
7. G.M. Booth, The Distributed System Environment, McGraw-Hill Book Company, NY 1981.
8. M.P.Groover, Automation, Production Systems and Computer Integrated Manufacturing, Prentice-Hall International Inc., Engle wood Cliffs, NJ, pp.175-179, 1987.
9. P.J.O'Grady, Controlling Automated Manufacturing Systems, Kogan Page Ltd., London, 1986.
10. P.G. Bunce, "Developments in Advanced Factory Control and Management Systems", Control and Programming in Advanced Manufacturing, IFS Publication Ltd., UK, Springer-Verlag, Berlin, 1988.
11. C.G. Cassandras, "Autonomous Material Handling in Computer Integrated Manufacturing," Modelling and Design of Flexible Manufacturing Systems Elsevier Science Publishers B.V., Amsterdam, 1986.
12. B.Henoch, "A Strategic Model for Reliability

- ty and Availability in Automatic Manufacturing", International Journal of Advanced Manufacturing Technology, Vol.3, No.1, pp.1-10, 1984.
13. R.E.Fairley, Software Engineering Concepts, MacGrw-Hill Company, NY, 1985.
 14. L.L.Gremillion, "Determinants of Program Repair Maintenance Requirements", Communications of the ACM, Vol.27, No.8 August, pp.826-32, 1984.
 15. M.J. Shaw, "Dynamic Scheduling in Cellular Manufacturing Systems : A Framework for Networked Decision Making," Journal of Manufacturing Systems, Vol.7, No.2, pp.35-46, 1984.
 16. R.Musterer, "A Case for Programmable Controllers in Process Control", Instruments and Control Systems, pp.65-72, June 1989.
 17. P.H.Enslow, Jr., "What is a 'Distributed' Data Processing System?", Computer, Vol.11, No.1, pp.13-21, January 1978.
 18. 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, December 1980.
 19. L.L.Gremillion, "Determinants of Program Repair Maintenance Requirements", Communications of the ACM, Vol.27, No.8, pp.826-832, August 1984.
 20. N.A. Duffie, "Synthesis of Heterarchical Manufacturing Systems", Computers in Industry, Vol.14, pp.167-174, 1990.
 21. S. Cammarata, D.McArthur and R.Steeb, "Strategies of cooperation in distributed problem solving", in: Proc. 8th Int. Joint Conf. of AI, Karlsruhe, FRG, Vol.2, pp.767-770, 1983.
 22. Palajiappan, M., "The Envoy Framework : An Open Architecture for Agents", ACM Transaction on Information Systems, Vol.10, No.3, pp.233-264, 1992.
 23. Robert, E. and Tony, M., Blackboard Systems, Addison-Wesley, pp.10, 1988
 24. 배용환, 배태용, 이석희, "실시간 다중고장진단 제어기법에 관한 연구", 한국정밀공학회 '95년도 춘계학술대회논문집 pp.457-462, 1995.
 25. 배용환, 이석희, 배태용, 이형국, "에이전트기반 실시간 고장진단 시뮬레이션기법", '95년도 추계한국정밀공학회 논문집, pp.790-795, 1995.