

◆ 첨단 제조 시스템 기술특집 ◆

Shop Floor 제어 시스템의 이론과 응용

장병열, 신준호, 조현보

포항공과대학교 산업공학과

1. 서 론

하나의 제품을 생산하기 위해서는 많은 종류의 작업들이 필요하다. 가령, 도면을 만든다든지, 공장에서 어떤 과정을 거쳐 만들어질지에 대한 계획을 세운다든지, 생산현장의 공작기계를 효율적으로 제어하여 제품을 생산하는 것들이다. 기존에는 이러한 작업들이 한대의 공작기계를 중심으로 그 분야의 전문가에 의해 수행되었다. 즉, 생산 기술자는 디자인 시스템을 이용하여 원하는 제품의 도면을 그린 후 NC 프로그램을 만들어서 자기의 Shop에 있는 공작기계에 Download하여 원하는 작업을 완성시켰다. 이와 같은 반복된 작업은 단품종 소량 생산을 주로 하는 최근의 산업 동향에는 이미 맞지 않은 지 오래되었다. 특히, 점점 짚아지는 제품의 수명 주기, 낮아지는 생산 비용, 높은 품질을 요구하는 현대 시장의 흐름은 제품 생산 방식을 변화시키고 있다. 이는 단위 공작기계 보다 '시스템'을 구성하여 원하는 제품을 신속하게 생산하려는 시도를 낳았으며 이로 인하여 FMS(Flexible Manufacturing System), CIM(Computer Integrated Manufacturing)과 같은 새로운 생산 철학이 등장하였다 (Chang et al. 1991). 전형적인 FMS는 「그림 1」에서 보는 바와 같이 NC 공작기계, 로보트, 컨베이어 벨트, AGV 등으로 구성되어 있다. 생산 지시가 내려지면 원자재가 자동 창고에서 출고되어 GE-A4 로보트에 의해 컨베이어에 전달되며, 컨베이어는 명령에 따라 해당 공작기계 앞의 로보트로 배달된다. 여러 가지 공작기계를 거치면서 예정된 제품이 완성되면 다시 자동 창고에 저장된다.

이들은 기존의 생산 방식이 가지고 있던 모순, 즉 "Islands of automation"을 극복하기 위한 수단으로써 각 생산 기능들의 통합이라는 데 역점을 두게 되었다. 여기서 통합이란 각 부분들을 하나의 전체로 만든다는 뜻이다. 실제로

통합을 위한 기술을 이용하여 통합된 생산 시스템을 구축할 수 있게 된다. 그럼 2는 효율적인 생산 시스템의 운영을 위

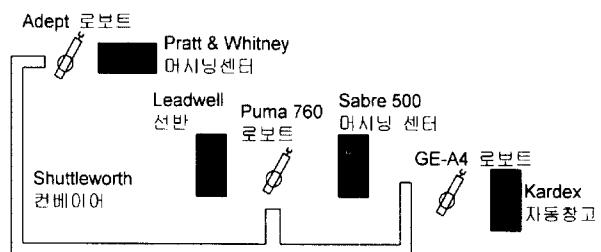


그림 1. 생산 제어 시스템이 관리하는 전형적인 FMS.

한 통합화된 기능들을 표현하는 수레바퀴를 나타낸다 (Wysk 1992). 그것은 설계, 공정계획, 그리고 Shop Floor 제어의 세가지 기능을 가지고 있다. 설계는 성능 및 기능을 만족 시키는 도면을 만들고, 공정계획은 원자재로부터 도면에 정의된 제품을 만드는데 필요한 정보를 만들어 내며, Shop Floor 제어는 공정계획 정보를 이용하여 제품 또는 반제품의 흐름을 효율적으로 스케줄링하고 수행시키는 역할을 한다. 공정계획에서 처리할 수 없는 도면은 설계 단계로 넘겨지며 Shop Floor 제어 단계에서 생산할 수 없는 제품은 공정계획 단계로 넘겨진다. 이와 같이 수레 바퀴 내부에 정의된 세가지 기능은 유기적으로 통합되어 굴러가야 한다.

본 논문에서는 Shop Floor 제어에 관한 이론 및 응용 사례를 논하고자 한다. Shop Floor 제어 시스템 (SFCS : Shop Floor Control System)은 주어진 생산 현장의 기계류를 효율적으로 계획, 스케줄링 및 운영하는 역할을 수행하는 소프트웨어를 일컫는다 (Cho and Wysk 1995). SFCS는 그림 3에서 볼 수 있는 바와 같이 주문 시스템에서 제품명, 주문량, 납기일 등의 주문 정보를, 공정계획 시스템으로부터

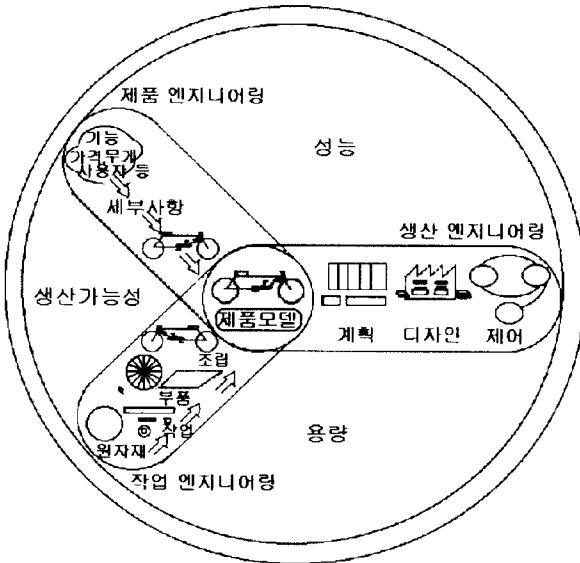


그림 2. 통합된 생산 시스템을 표현하는 수레바퀴.

공정계획을, 설계 시스템으로부터 도면 정보를, 그리고 하위 기계로부터 다양한 상태 신호를 입력받는다. 이와 같은 정보를 토대로 SFCS는 제품을 생산자원에 할당, 태스크 생성, 태스크들의 진행 과정을 모니터링, 오류 발견 및 수정, 정보 관리, 통신 등을 수행한다.

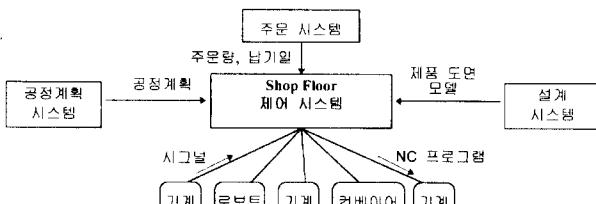


그림 3. SFCS의 주위 환경.

복잡한 기능을 수행하는 SFCS의 성공적인 개발을 위해서는 체계적인 접근이 필요하다. 기준에는 구체적인 계획이나 방법에 의존하지 않고 빠른 시일 내에 전시용 프로토타입을 만드는데 시간과 비용을 투자했기 때문에 올바른 SFCS를 만들 수 없었다. 특히, 기존의 SFCS는 정해진 하나의 FMS에만 적용되고 여러 종류의 FMS에 이식될 수 없었다. 이것은 SFCS를 단순한 소프트웨어로 잘못 이해한 결과이다. 실제로 SFCS는 제어, 통신, 데이터베이스, 모니터링, 외부 환경 등의 복합적인 문제들이 통합된 거대한 소프트웨어이다. 더 중요한 것은 SFCS가 적용될 Shop Floor는 대부분이 서로 상이하다는 것이다. 이와 같은 요소들을 반영하기 위해서 SFCS를 여러 가지 관점에서 바라볼 필요가 있다. 즉, 제어 흐름 방식을 다루는 제어구조 (Control Architecture), 기능의 정의와 역할을 다루는 기능구조 (Function Architecture), 정보의 종류와 흐름을 다루는 정보구조 (Information

Architecture), 통신 방식이나 흐름을 다루는 통신구조 (Communication Architecture) 등의 관점을 고려해야 한다 (Jung et al. 1996). 본 논문에서는 SFCS를 이와 같은 관점에서 서술할 것이다. 또한 그 응용 사례들을 포함하고자 한다.

2. 제어구조

SFCS의 제어구조 (Control Architecture)는 여러 가지 제어해야 할 대상을 정의하고 그들 사이의 상호 작용에 관한 법칙들을 제시한다. 구체적으로는 서로 다른 제어 시스템들 사이의 인터페이스 방법, 내부 구조, 사용자와의 인터페이스 방법 등을 포함하고 있다. 제어구조를 정의하기 위해서는 다음과 같은 과정을 거쳐야 한다. 첫째, 제어 되어야 할 기계류의 단위 (UCC : Unit of Control Component)를 결정해야 한다. UCC는 하나의 제어 컴퓨터에 의해 제어될 수 있는 기계 또는 기계 그룹을 나타낸다. 예를 들면, 그림 1에 나타난 공작기계, 로보트, 컨베이어 등이 UCC이다. 둘째, UCC를 제어해야 할 소프트웨어인 Equipment 제어 시스템 (ECS : Equipment Control System)을 정의해야 한다. ECS는 연결된 UCC의 운영을 책임지는데, 구체적으로는 태스크 계획, 태스크 스케줄링, 태스크 실행, 기계 언어의 Download, 상태 감시, 오류 발견 및 수정, 사용자와의 인터페이스 등을 수행한다. 세째, ECS들 사이의 상호 작용을 처리하는 제어 시스템이 필요하게 된다. 그 방식에 따라서 그림 4와 같이 크게 네 가지로 나눌 수 있다.

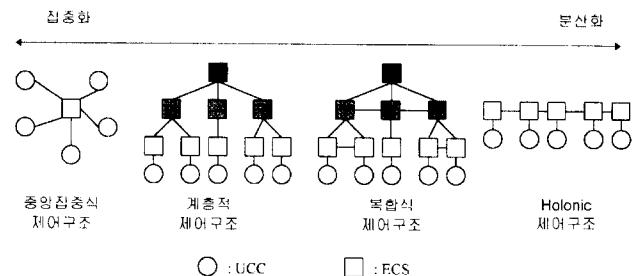


그림 4. ECS 사이의 인터페이스 방식에 따른 제어구조의 분류.

2.1. 중앙집중식 제어구조

중앙집중식 제어구조는 하나의 제어 시스템이 모든 UCC들을 관리한다. 이 구조는 초기의 SFCS 개발에 사용되었으나 미래의 시스템에서는 부적합한 것으로 판단된다. 이 구조의 장점은 전체의 정보를 쉽게 얻을 수 있으며, 따라서 전체의 최적화가 가능하고, 적은 수의 컴퓨터가 필요하며, 구현이 용이하다는 것이다. 그러나 시스템이 커짐에 따라서 전체적인 반응속도가 느려지고, 제어 시스템의 작동이 불가능하면 전체 시스템이 멈추어야 하며, 유지 보수가 힘들다

는 단점들을 안고 있다.

2.2. 계층적 제어구조

계층적 제어구조의 가장 큰 특징은 여러 ECS들 간의 인터페이스가 계층적으로 주종관계를 가지고 이루어진다는 것이다 (Jones and McLean 1990, Joshi et al. 1990, Simpson et al. 1982, Cho and Wysk 1995). 계층적 제어구조를 정의하기 위해서는 먼저 ECS를 어떻게 루프을 것인가를 결정해야 한다. 몇 개의 ECS 그룹이 정의 되면 그 그룹을 제어할 수 있는 워크스테이션 제어 시스템 (WCS : Workstation Control System)을 정의 한다 (Cho 1993). WCS는 그 특성과 역할에 따라서 제조 전용 WCS, 자동 창고 WCS, 물류 WCS 등으로 나누어질 수 있다. 예를 들면, 그림 1에서 제조 WCS는 Puma 760 로보트, Leadwell 선반, Sabre 500 머시닝 센터의 세 개의 ECS로 구성 되며, 자동 창고 WCS는 GE-A4 로보트와 Kardex 저장창고의 ECS로 구성 된다. 마지막으로, 이를 WCS들 간의 인터페이스를 책임지는 Shop 제어 시스템 (SCS : Shop Control System)을 정의하면 된다. 상위 제어 시스템은 명령 메시지를 아래로 보내고, 하위 제어 시스템은 Shop의 상태를 나타내는 메시지를 상위로 보낸다. 상위 제어 시스템일수록 긴 주기를 가지고 의사 결정을 수행한다.

계층적 제어 시스템의 ECS, WCS, SCS는 상위에서 하달되는 명령 메시지와 하부에서 보고 되는 상태 메시지에 기초하여 무엇을 할 것인가를 결정한다. 명령 메시지란 제품을 기계에서 가공을 시작하라, 제품을 기계로부터 Unloading 하라, 제품을 버퍼에 두어라, 로보트를 어디로 움직여라 등과 같은 지시 사항을 포함한다. 상태 메시지란 명령 메시지에 대한 성공 여부, 제조 중에 있는 제품의 남은 공정 시간, 공구의 상태, 로보트의 현재 위치 등과 같은 동적인 정보를 포함한다. 명령 메시지는 반드시 상부에서 하부로 전달되어야 하지만, 상태 메시지는 하부에서 상부로, 상부에서 하부로, 또는 옆으로 어느 방향으로든지 흐를 수 있다. 이와 같은 명령 메시지, 상태 메시지를 그림 5에서 제품의 흐름과 비교하여 나타내었다 (Cho and Wysk 1995). 계층적 제어구조는 기존의 어려운 문제를 분해하여 해결하는 방식과 같기 때문에 이해하기가 쉬우며 전체 최적화를 구할 수 있다. 그

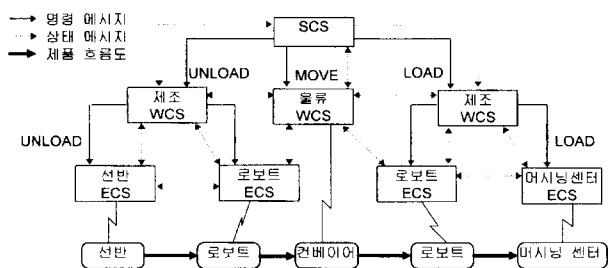


그림 5. 명령 메시지와 상태 메시지의 흐름도.

러나 하나의 제어 시스템 고장은 그 아래의 모든 제어 시스템을 정지시킨다는 것과 너무 많은 제어 시스템을 요구한다는 것이 단점이다.

2.3. Holonic 제어구조

Holonic 제어구조는 기존의 계층적 구조의 단점을 보완한 새로운 개념으로써 최근에 많은 연구가 진행 되고 있는 분야이다. 즉, ECS 사이의 인터페이스를 다루기 위하여 계층적으로 제어 시스템을 구축하는 대신에 ECS로 하여금 연결된 UCC의 운영 뿐만 아니라 서로 다른 ECS들과도 통신하여 스스로 인터페이스를 해결하게 하는 것이다 (Upton et al. 1991). 그럼 6에 보이는 바와 같이 Shop Floor 제어 시스템은 각 UCC에 일렬로 연결된 ECS의 집합으로 이루어 진다. Holonic 제어구조에서는 ECS를 Holon이라고도 일컫는다. 그 배경으로는 점차적으로 값이 싸지고 지능화된 컴퓨터 환경과 강력한 네트워크 통신의 대두에 기인한다. Holonic 제어구조는 다른 용어로도 사용되는데 비계층적 제어구조 (Hierarchical control), 태협을 이용한 제어구조 (Negotiation-based control), 분산 제어구조 (Distributed control), 협동 제어구조 (Collaborative control) 등으로 불리운다.

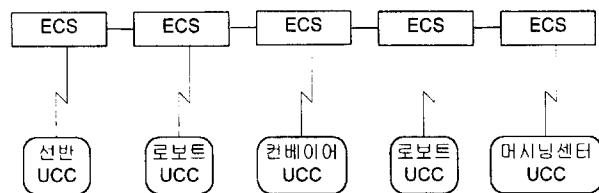


그림 6. Holonic 제어구조의 모습.

Holonic 제어 시스템은 ECS들로 구성 되어 있기 때문에 어느 일을 수행함에 있어서 서로 협상하여 일을 이끌어 나가게 된다. ECS 사이에 협상을 하는 방법들에 대해서는 많은 연구가 진행되고 있는데 가장 많이 사용되는 방법은 경매 시스템이다. 경매 시스템이란 말 그대로 어떠한 일을 어떤 ECS가 맡아서 할 것인가를 결정함에 있어서 가장 높은 경매값 (예를 들면, 가장 먼저 할당될 태스크를 마칠 수 있는 ECS)을 제시한 ECS가 선택되어 그 일을 수행하게 된다. 즉 어떤 하나의 ECS가 경매를 제기하면 그 외의 ECS들로부터 수집된 경매값을 기초로 하여 높은 값을 제시한 ECS에게 일을 수행하도록 하는 것이다. 상부에서 지시할 수 있는 매개체가 없기 때문에, 해당 제품을 생산하기 위한 최초의 경매를 시작하여 결정을 내리는 것이 가장 어려운 문제이다. 이와 같은 문제를 해결하기 위하여 ECS 외에 Coordinator 시스템을 만들어 공정계획 시스템 및 설계 시스템과 연결하게 하고 경매를 시작하게 할 수 있다. 또한 전체 시스템의 상태를 보여줄 수 있는 수단을 제공하게 할 수 있다.

Holonic 제어구조의 장점은 계층적 제어구조의 단점과 동

일하고, 단점은 계층적 제어구조의 장점들이라고 할 수 있다. 어느 제어구조가 더 우수한지는 관점에 따라 다를 수 있다. 가장 중요한 차이점은 계층적 제어구조의 경우 하나의 SCS가 모든 시스템을 장악하고 있기 때문에 전체의 최적화가 가능하지만, Holonic 제어구조에서는 어떠한 대표자가 존재하지 않으므로 부분적인 최적화만 가능하게 된다는 것이다. 그러나, 공장 전체의 Shop Floor 제어 시스템을 계층적으로 개발하면 많은 시간이 소모된다. 왜냐하면, 하나의 UCC가 시스템에 추가되거나 제거되면 그 UCC의 상부에 위치한 모든 제어 시스템에게 영향을 주므로 각 제어 시스템을 수정해야 한다. 반대로 Holonic 제어구조에서는 각 ECS들이 서로 독립적이므로 각 ECS를 개발하여 공장에서 서로 결합하는 작업만 하면 될 것이다. 따라서, 신속한 제어 시스템의 개발면에서 볼 때, Holonic 제어구조가 계층적 제어구조 보다 우위를 차지하고 있다.

2.4. 복합식 제어구조

복합식 제어구조는 계층적 제어구조와 Holonic 제어구조를 병합한 것이다(Jones and Saleh 1990). 이 방법은 아직까지 많은 연구가 진행되지 않았으며 서로 상이한 두 구조의 장점을 이용한다면 앞으로 계속 연구해볼 만한 제어구조이다.

3. 기능구조

기능구조 (Function Architecture)는 각 제어 시스템이 수행하는 기능들을 정의하고 이를 간의 상호작용에 대한 입력 출력 관계를 포함한다. 이것은 다음과 같은 방법에 의해 얻을 수 있다. 첫째, 제어 시스템 내부에서 처리해야 할 모든 태스크를 수집한다. 둘째, 각 태스크에 대해서 발생 빈도수를 조사한다. 예를 들면, 통신 태스크는 매우 종종 발생하는 방면에 계획을 만드는 태스크는 드물게 발생할 것이다. 세째, 각 태스크를 발생 빈도수에 따라 정렬한다. 넷째, 비슷한 발생 빈도수를 가지는 태스크들을 그룹으로 만들어 하나의 기능에 할당한다. 제어 시스템은 여기서 정의된 기능의 갯수 만큼의 모듈로 구성된다. 일반적으로, 제어 시스템의 기능을 의사 결정 (Decision making)과 실행 (Execution) 기능으로 양분할 수 있다. 그러면 제어 시스템 내부에서는 두개의 소프트웨어 모듈이 각각의 기능을 수행하게 된다. 대부분의 제어 시스템은 의사 결정 기능을 계획 (Planning)과 스케줄링 (Scheduling) 기능으로 나누어 실행 기능과 함께 세 가지의 모듈로 구성된다. 각 기능의 역할은 제어구조의 종류에 따라서 약간의 변동이 있을 수 있다. 본 논문에서는 계층적 제어 구조에서의 기능구조를 다룬다.

계층적 제어구조를 가지는 제어 시스템은 그림 7과 같이 계획, 스케줄링, 실행 기능으로 분해될 수 있다 (Joshi et al. 1990, Cho 1993). SCS, WCS, ECS 각각이 계획, 스케줄링,

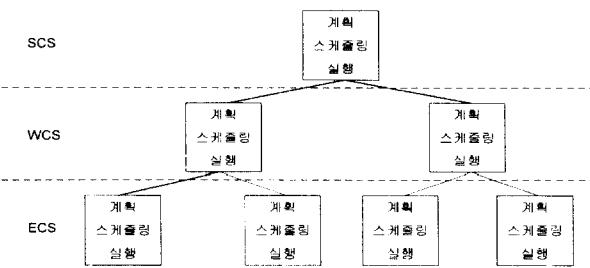


그림 7. 계층적 제어구조에서의 기능구조.

실행 기능을 가지고 있지만 어느 정도 그 내용은 다를 수 있다. 그러나 대체적으로 계획 기능은 제품의 흐름의 방법을 정하고 전체 완성 시간을 결정하는 역할을 한다. 스케줄링은 각 제품들의 가공에 필요한 작업의 시작 시간과 마치는 시간을 결정한다. 실행 기능은 계획과 스케줄링에서 만들어진 결과를 하위 제어 시스템에 전달하고 상위 제어 시스템에게 상태 정보들을 전달하는 역할을 한다.

각 계층별로 수행하는 세부적인 계획, 스케줄링, 실행의 역할은 표 1과 같다. 각 계층의 기능들은 비슷하지만 SCS는 Batch를 주로 다루며 웍스테이션 사이의 물류를 원활하게 운영하는 책임을 가진다. WCS는 SCS에서 할당된 태스크들을 주어진 시간 안에 마치기 위해서 책임지고 있는 공작기계, 로보트 등의 상태 감시, 오류 발견 및 수정, 통신, 제품의 스케줄링, Deadlock 해결 등을 수행한다 (Cho and Wysk 1993, Cho and Wysk 1995, Cho et al. 1996). ECS는 WCS의 명령에 따라 제품을 나르거나 (예; 로보트 ECS), 정해진 시간 안에 제품을 완성하기 위해서 공구경로 및 머시닝 인수 (Parameter)를 수정하고 수행해야 할 오퍼레이션의 순서를 결정한다. 또한 UCC와 직접 연결되어 있기 때문에 UCC에 기계 코드를 Download해야 하고 UCC의 상태를 감시해야 한다. 무엇보다도 중요한 역할은 동기화 작업인데 로보트가 제품을 공작기계에 Loading하거나 Unloading 할 때 서로 자기의 상태를 상대방에게 알려 주면서 제품을 안전하게 전달하는 일련의 동작이다 (Cho 1996).

SCS와 WCS의 스케줄링 기능이 반드시 해결해야 하는 재미있는 문제 중의 하나는 Deadlock이다. Deadlock이란 기다리면 언젠가는 길이 열리는 Blocking과 다르게 제품이나 자원들이 영원히 교착상태에 빠져서 특별한 구제 작업이 없이는 시스템을 정지시켜 버린다. 보통 FMS는 약간의 버퍼를 가지고 있고 생산 자원의 용량은 제한되어 있기 때문에 Deadlock은 항상 발생할 수 있다 (Wysk et al. 1991, Cho et al. 1995). 표 2에서 보는 바와 같이 Deadlock은 크게 네가지로 나누어지는데 가장 흔히 발생하는 것은 제품의 흐름에 의한 Deadlock이다. 이것은 제품을 어떤 기계에 전달하기 이전에 Deadlock을 일으키는 상태인지를 미리 판단하여 그 제품의 전달 여부를 결정하여 미리 방지할 수 있다. 만약 Deadlock이 일어난 경우에도 최소한 하나의 버퍼가

표 1. 계층적 제어구조에서 각 계층별 기능의 역할.

계층	계획(Planning)	스케줄링(Scheduling)	실행(Execution)
SCS	• 주문 접수 • 주문 완성 시간 결정 • Batch의 라우팅 결정	• Batch 스케줄링 • Deadlock 해결	• WCS 상태 감시 • TCP/IP 통신 • 오류 발견 및 수정
WCS	• Batch를 제품으로 분리 • 제품별 공작기계 할당	• 제품 스케줄링 • 로보트 위치 결정 • Deadlock 해결	• ECS 상태 감시 • TCP/IP 통신 • 오류 발견 및 수정
ECS	• 공구 할당 • 공구경로 수정 • 머시닝 인수 수정	• 오퍼레이션 순서 결정	• 기계 코드 Download • UCC 상태/오류 감시 • RS-232C 통신 • 동기화 작업

있으면 로보트는 임의로 하나의 제품을 버퍼에 두고 다른 제품을 옮긴 다음 버퍼로부터 제품을 원하는 공작기계에 옮기면 해결될 수 있다. 중요한 것은 제어 시스템은 Deadlock을 미리 방지할 것인지 또는 Deadlock을 발생시키고 버퍼 하나를 예약해 두어 Deadlock을 해결할 것인지를 결정해야 한다.

4. 정보구조

정보구조 (Information Architecture)는 Shop Floor에서 어떠한 정보들이 생성되며 이러한 정보들이 어떻게 변하여 가는가를 다룬다. 일반적으로 정보란 다양한 의미를 가지고

있다. 구체적으로 Shop Floor에서의 정보란 설계, 공정계획, 주문 시스템들이 만들어 내는 정보들과, 이러한 정보 사이의 관계, Shop Floor에서 정보들이 변화하는 과정을 말한다. 그 중에서 가장 중요한 정보는 가공을 위해 반드시 필요한 공정계획과 그것이 Shop Floor에서의 어떻게 변화하느냐는 것이다. 우선 공정계획을 표현하는 방법을 알아본 뒤 계층적 제어구조에서의 변천 과정을 서술하고자 한다.

공정계획을 나타내기 위한 기본적인 방법은 AND/OR 그래프를 이용하는 것이다. 이 그래프를 Feature 그래프 $G = (V, E)$ 로 정의하는데 노드는 Feature뿐만 아니라 이들의 선후 관계를 나타내기 위한 SPLIT-AND, SPLIT-OR, JOIN-AND, JOIN-OR 등의 형식을 포함한다. Feature를 나타내는 노드는 이 Feature를 제거하는데 사용될 공작기계, 고정구, 머시닝 인수, NC 프로그램 등을 포함한다. SPLIT-AND 노드와 JOIN-AND 노드 사이의 일련의 Feature들은 모두가 임의의 순서로 실행되어야 한다. SPLIT-OR 노드와 JOIN-OR 노드 사이의 일련의 Feature들은 그 중 하나만 선택되어서 실행되어야 한다. 그림 8은 위의 설명을 샘플 제품과 함께 그림으로 표시한 것인데, Feature 2에 대한 상세한 정보가 포함되어 있다. 참고로 Feature 5'은 Feature 5와 6을 합한 것이다 (Cho et al. 1994).

표 2. Deadlock의 종류와 정의.

종류	정의	예
제품흐름에 의한 Deadlock	공작기계에 위치한 각 제품들이 순환을 그리면서 다른 공작기계를 요구하는 상태(제품1은 기계3을, 제품3은 기계2를, 또는 기계1을 요구할 때)	
제품흐름에 의한 절박한 Deadlock	공작기계를 요구하는 상황이 당장은 Deadlock을 일으키지 않으나 언젠가는 Deadlock 상태으로 결론나는 상태(제품1은 기계3과 기계2를, 제품2는 기계3과 기계1을 차례로 요구할 때)	
생산자원에 의한 Deadlock	각 제품이 하나의 자원을 가지고 있는 상황에서 서로 상대방의 자원을 요구하는 상태(공구1이 각 기계에 하나씩 할당된 후, 각 제품이 공구1을 한 개씩 더 요구할 때)	
물류시스템에 의한 Deadlock	물류 시스템이 서로 꼬리를 물고 늘어서 움직이지 못하는 상태	

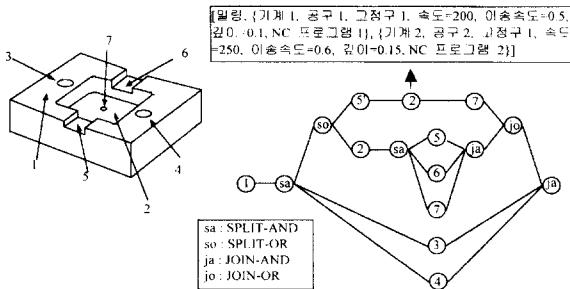


그림 8. 샘플 제품과 제작에 필요한 공정계획.

SCS로 주문과 함께 입력된 공정계획은 여러 형태로 변화되어 간다. 이러한 변화 모습을 표 3에서 나타내었는데 왼쪽 상단의 그래프가 공정계획 시스템에서 만들어진 공정계획이다. SCS는 입력된 공정계획을 적절하게 분리하여 할당될 제품 배치를 나타내는 Plan 그래프와 Sequence 그래프를 생성한 후, 각 WCS로 배분한다. WCS는 다시 이를 분리하여 제품과 함께 ECS로 배분한다. 이때 공정계획을 분리하는 기준은 각 공작기계의 부하량을 평준화 시키거나 Dead-lock을 줄이는 등의 시스템 전체의 최적화이다. 마지막으로 ECS는 할당된 Feature를 신속하게 제조하기 위하여 공구교환 회수를 줄이거나 공구의 이동 거리를 줄이는 기준을 이용하여 Feature의 순서를 정한다. 공작기계 제어기로 전송될 NC 프로그램은 각 Feature별로 저장되므로 여러 Feature를 동시에 제거하려면 이들을 하나의 프로그램으로 단축하는 것이 효율적이다.

5. 통신 구조

통신 구조 (Communication Architecture)는 제어 시스템 사이의 통신 방법을 제시한다. 각 ECS는 연결된 기계뿐만 아니라 다른 ECS, 키보드, 모니터 등과도 통신을 할 수 있어야 한다 (Brill and Gramm 1991). 연결된 기계와의 통신은 보통 RS232-C 시리얼 통신 방법으로 이루어진다. 제어 시스템 사이의 통신은 그림 9와 같이 주로 TCP/IP 통신 프로토콜이 이용된다. 통신은 성공적인 생산 시스템 자동화를 위해서는 필수적인 전제 조건이다. 이러한 필요성에 의해 많은 연구가 진행되어 왔는데 통신구조를 제어구조, 기능구조, 정보구조 등과 서로 혼합하여 개발해 왔다. 이와 같이 구조들을 혼합하여 개발하면 Shop Floor 제어 시스템이 도중에 중단되었을 때 오류의 발생 원인 및 장소를 규명하기가 어렵게 된다.

통신 소프트웨어가 가져야 할 조건으로는 여러 상황에 모두 적용될 수 있는 일반성이다. 이것은 플랫폼 (IBM-PC, Mac, Workstation 등)이나 프로토콜 (TCP/IP, RS232-C 등), 운영 시스템 (DOS, Windows 95, Windows NT, Unix 등)에 의한 영향을 받지 않아야 된다는 의미다. 또한 보내는 메시지가 원하는 시간 안에 확실히, 올바르게 전해졌는지의 보장 여부도 중요한 필수 조건이다. 이러한 조건들을 만족시키는 통신을 지원하면서 개발할 제어 시스템과의 독립성을 유지하기 위해서는 통신 전용 소프트웨어가 만들어져야 한다. 통신 전용 소프트웨어는 어디에서 오는 메시지를 읽은 다음 컴퓨터 내부의 메시지 버퍼에 저장할 수 있

표 3. Shop Floor상에서의 공정계획의 변화.

계층	Feature 그래프	Plan 그래프	Sequence 그래프
SCS			 워크스테이션 2 w1 -> w2 워크스테이션 1
WCS			 기계 1 o1 -> o2 -> o3 기계 2 o2 -> o3 기계 3
ECS			 공구 2 t1 -> t2 -> t3 -> t4 공구 1 t1 공구 4 t4

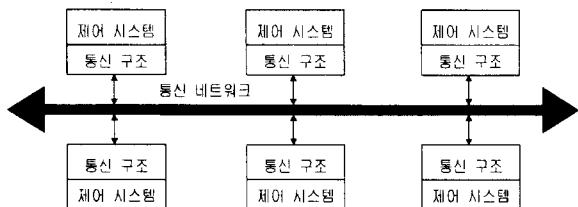


그림 9. 통신 구조의 개념도.

어야 한다. 이것은 통신 소프트웨어가 제어 컴퓨터 포트들을 연속적으로 계속 탐색함으로써 수행될 수 있다. 그러면 제어 소프트웨어는 단순히 메시지 버퍼에서 메시지를 읽어 해석하면 어디서 온 것인지를 파악할 수 있고 대응 메시지를 준비할 것이다. 또한 통신 소프트웨어는 메시지를 어디든지 보낼 수 있어야 한다. 그래서 제어 소프트웨어는 단순히 통신 소프트웨어에게 어떤 메시지를 원하는 곳으로 보내라고 지시만 하면 된다. 다시 말해서, 통신 전용 소프트웨어는 다음의 두 명령어만 사용하여 제어 시스템을 지원한다 : GetMessage (제어 소프트웨어가 하나의 메시지를 버퍼로부터 일을 때)와 SendMessage (제어 소프트웨어가 하나의 메시지를 원하는 곳으로 보낼 때).

6. Shop Floor 제어 시스템 응용 사례

6.1. 구축 환경

이 장에서는 실제 Shop Floor 제어 시스템이 구현되어지는 몇 가지 응용 사례를 열거한다. 먼저 Shop Floor 제어 시스템이 구축되어지는 환경을 포항공대 산업공학과에 위치한 PosTROL 시스템이라고 가정하자 그림 10. PosTROL 시스템은 Orac 자동선반, Triac 자동밀링, Jupiter 스카라 로보트, Centari 4축 다관절 로보트, FARA A1-U 6축 다관절 로보트, DAV-100 AGV, 컨베이어, 자동 창고, 두개의 Loading/Unloading 스테이션으로 구성된다.

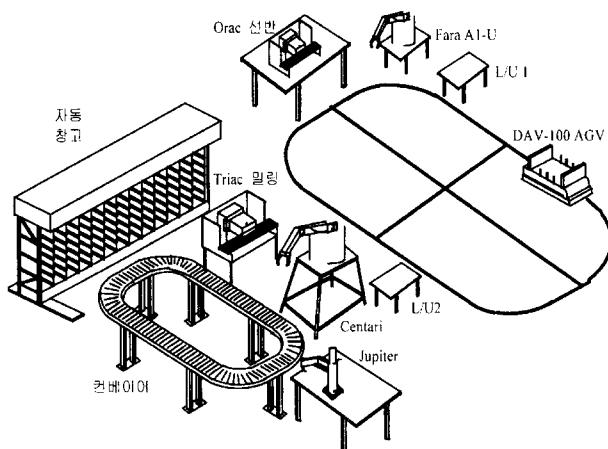


그림 10. 포항공대 산업공학과 Pos TROL시스템.

6.2. 응용 사례 : 계층적 제어, 3 기능 구조, Windows NT 환경

3단계 계층적 제어 구조는 각각의 UCC에 제어 시스템이 할당되어 UCC를 제어하며 몇 개의 ECS를 묶어 이를 WCS가 제어한다. 또한 이 WCS를 모두 제어하는 SCS가 존재한다. 위의 아키텍처를 사용해 구현한 PosTROL 시스템은 [그림 11]과 같은 구조를 갖는다.

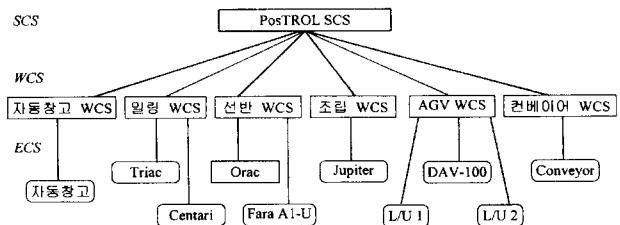


그림 11. PosTROL 시스템의 계층적 제어 구조.

위의 구조는 각 UCC들의 기능을 토대로 했으며 한 WCS이 제품의 특정한 작업을 수행하도록 되어 있다. 가령, 자동 창고 WCS는 완성된 제품을 저장하는 역할, 밀링 WCS는 로보트로 제품을 옮긴 후 밀링 작업을 수행하는 역할을 한다. 생산 명령이 내려지면 PosTROL SCS는 밀링 WCS나 선반 WCS에 할당된 작업을 지시한다. 만약 밀링 WCS에게 작업이 할당되었다면 밀링 WCS는 자신에게 할당된 명령을 수행하기 위해 다시 자신의 밑에 존재하는 ECS인 Triac, Centari ECS에게 여러 가지 세부 작업을 지시하게 된다. ECS들은 자신에게 주어진 작업들을 수행하게 되며 작업이 끝나거나 오류가 발생할 경우, 상위 제어 시스템인 Milling WCS나 다음 작업을 담당하는 ECS에게 이를 보고한다.

계층적 구조에서는 하위 단계 제어 시스템들이 다른 상위 단계 하에 있는 제어 시스템에게 직접 명령을 내릴 수 없는 제약을 가진다. 그래서 이러한 경우가 발생하면 자신의 상위 단계 제어 시스템에게 요청을 하게 된다. 가령 한 제품이 밀링 작업 후 선반 작업도 필요로 할 경우 밀링 WCS는 PosTROL SCS에게 이를 요청하게 된다. 그러면 PosTROL SCS는 AGV WCS에게 제품 운반을 요청하고 이 작업의 완료 메시지를 받으면 선반 WCS에게 선반 작업을 수행하도록 명령하게 된다.

이 구조에서 각각의 제어 시스템들은 계획, 스케줄링, 실행의 세 가지 기능을 수행하는 모듈을 가지고 있으며 다른 제어 시스템과 통신하기 위한 메시지 프로토콜을 사용한다. 그리고 Windows NT 환경에서 구현되어 여러 가지 기능이 동시에 수행되어진다. 따라서 제어 시스템은 자신에게 발생하는 여러 가지 사건 (Event)들에 대해 즉각적인 반응을 할 수 있다. 예를 들어, 컨베이어 제어 시스템은 자신에게 동시에 주어진 두 가지 작업 요청 자동 창고로의 제품 Unloading, Centari로부터의 제품 Loading 을 한꺼번에 처리

할 수 있다.

6.3. 응용 사례 : Holonic 제어구조, 3 기능 구조, Windows NT 환경

Holonic 제어 구조는 계층적 제어 구조에 존재하는 상하 관계를 없애고, ECS 사이의 통신을 통하여 시스템의 목적을 달성한다. 이 구조하에서는 모든 ECS가 제품에 대한 동등한 접근권을 가지며 제어 시스템 사이에 상호 접근이 가능하다. 또한 작업이 서로 독립적이며, 전체 시스템의 프로토콜이 엄격하게 지켜진다.

이러한 Holonic 제어 구조를 구축하기 위해서는 생산 시스템을 구성하는 독립적 ECS들을 정의하여야 한다. 여기서 ECS는 의사 결정을 내릴 수 있는 적절한 단위로써, 한 UCC에 대해 하나의 ECS가 존재한다. 또한 ECS는 각 UCC 와 연결되는 것 이외에 자원 관리자 (Resource Manager), 초기화 및 모니터 ECS (InitMon : Initiating and Monitoring ECS)와 같이 추상적인 것도 있다. 그림 12는 PosTROL 내에 존재하는 ECS들을 표현한 것이다. InitMon은 초기에 경매를 시작하여 처음으로 작업을 수행할 ECS를 결정한다. 그 ECS는 할당된 작업을 마치고 다음 작업을 수행할 ECS 를 경매를 통해 결정하게 된다.

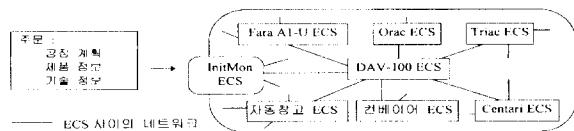


그림 12. PosTROL을 위한 Holonic 제어 구조의 ECS.

위의 ECS들은 InitMon을 제외하고는 그림 13와 같이 계획 모듈, 스케줄링 모듈, 실행 모듈로 구성되어져 각각의 역할을 수행하게 된다. 그리고 각각의 ECS들은 내부에 다른 ECS와, 혹은 내부의 각 기능 모듈간에 통신을 위한 모듈을 가지고 있어서 항상 서로 태협 (Negotiation)하는 특징을 갖는다. 독립적인 ECS들은 이런 태협을 통해서 Shop 전반의 상황을 고려한 최적의 결정을 내린다.

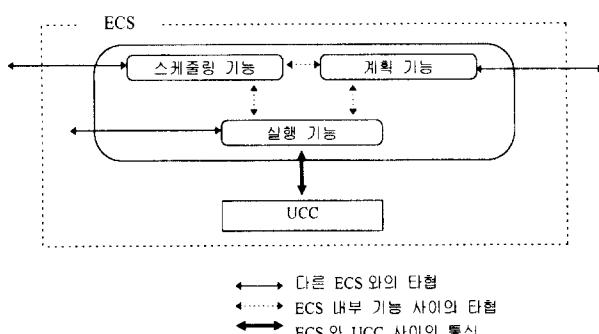


그림 13. ECS의 내부 기능.

7. 결 론

지금까지 Shop Floor 제어 시스템을 여러 가지 아키텍쳐 관점에서 살펴 보았다. 이것은 생산 시스템의 자동화를 위해 SFCS가 가져야 할 하드웨어와 소프트웨어를 설계 및 운영할 때 필요한 범위 및 목적, 자동화 대상의 범주 분석, 자동화에 필요한 부분들과 이들의 통합 방법 등을 제시하였다. 제어 구조는 여러 가지 Shop 개체, 즉 UCC들을 정의하고 그들 사이의 상호 작용에 관해 보여주는 방법론으로써, 서로 다른 제어 시스템과의 상호 작용 방법, 작동 상태, 내부 구조, 사용자와의 교류 방법 등을 포함하고 있다. 기능 구조는 각각의 제어 시스템들이 수행하는 기능들을 정의하고 이들 사이의 상호작용에 대해서 나타내고 있다. 또한 정보 구조는 Shop에서 어떠한 정보들이 생성되며 이러한 정보들이 어떻게 변하여 가는가를 다루고 있다. 이와 더불어 제어 시스템 사이의 통신 방법론에 관해 다루는 통신 구조를 서술하였다. SFCS를 개발하기 전에 반드시 여러 가지 구조를 미리 설정하여 독립적으로 연구한 후 이를 통합하는 작업을 거쳐야 한다.

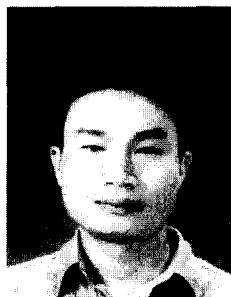
SFCS를 개발하는데 필요한 구조로써 본 논문에서 언급한 것 외에도 개발구조 (Development Architecture), 실행구조 (Implementation Architecture) 등이 있다. 이것은 계속 연구되어야 할 과제이며 위의 언급한 구조들에 대해서도 미흡한 부분은 앞으로 연구가 활발히 진행되어야 할 것이다.

참 고 문 헌

- [1] M. Brill and U. Gramm., "MMS : MAP applications services for the manufacturing industry," *Computer Networks and ISDN Systems* vol. 21, pp. 357-380, 1991.
- [2] T. C. Chang., R. A. Wysk. and H. P. Wang., *CAM*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [3] H. Cho., Intelligent Workstation Control for CIM, Ph.D. Dissertation, Texas A & M University, 1993.
- [4] H. Cho., A. Derebail., T. Hale., and R.A. Wysk., "A formal approach to integrating computer aided process planning and shop floor control," *ASME Transactions : Journal of Engineering for Industry* vol. 116, no. 1, pp. 106-116, 1994.
- [5] H. Cho., T. K. Kumaran. and R. A. Wysk., "Graph-theoretic deadlock detection and resolution for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation* vol. 11, no. 3, pp. 413-421, 1995.
- [6] H. Cho. and R. A. Wysk., "A robust adaptive schedu-

- ler for an intelligent workstation controller," *International Journal of Production Research* vol. 31, no. 4, pp. 771-790, 1993.
- [7] H. Cho. and R. A. Wysk., "An intelligent workstation controller for computer integrated manufacturing : Problems and models," *Journal of Manufacturing Systems* vol. 14, no. 4, pp. 252-263, 1995.
- [8] M. Jung., M. Chung. and H. Cho., "Architectural requirements for rapid development of agile manufacturing systems," *Computers and Industrial Engineering* vol. 31, no. 3/4, pp. 551-554, 1996.
- [9] H. Cho., J. S. Smith. and R. A. Wysk., "An intelligent workstation controller for integrated planning and scheduling of an FMS cell," (Accepted) *Production Planning and Control*, 1996.
- [10] H. Cho., "Petri net-based execution models for control of a flexible manufacturing system," (Accepted) *International Journal of Production Research*, 1996.
- [11] A. T. Jones. and C. R. McLean., "A proposed hierarchical control model for automated manufacturing systems," *Journal of Manufacturing Systems* vol. 5, pp. 15-25, 1986.
- [12] A. T. Jones. and A. Saleh., "A multi-level/multi-layer architecture for intelligent shop floor control," *International Journal of Computer Integrated Manufacturing* vol. 3, pp. 60-70, 1990.
- [13] S. Joshi., R. A. Wysk. and A. T. Jones., "A scalable architecture for CIM shop floor control," *Proceedings of CIMCON '90*, National Institute of Standards and Technology, Gaithersburg, Maryland, pp. 21-33, May, 1990.
- [14] J. A. Simpson., R. J. Hocken. and J. S. Albus., "The automated manufacturing research facility of the National Bureau of Standards," *Journal of Manufacturing Systems* vol. 1, pp. 17-32, 1982.
- [15] D. M. Upton., M. M. Barash. and A. M. Matheson., "Architectures and auctions in manufacturing," *International Journal of Computer Integrated Manufacturing* vol. 4, no. 1, pp. 23-33, 1991.
- [16] R. A. Wysk., "Integration requirements for the engineering of a product," *Proceedings of AUTOFACT'92*, Detroit, Michigan, 1992.
- [17] R. A. Wysk., B. N. Yang., and S. Joshi., "A detection of deadlocks in flexible manufacturing cells," *IEEE Transactions on Robotics and Automation* vol. pp. 7, pp. 853-859, 1991.

저자 소개



장 병 열

경북 포항시 남구 효자동 산 31
포항공과대학교 산업공학과
전화) 0562-279-8244
Fax) 0562-279-5998
Email) pjang · msi2.postech.ac.kr
학사, 포항공과대학교, 산업공학과, 1996.



신 준 호

경북 포항시 남구 효자동 산 31
포항공과대학교 산업공학과
전화) 0562-279-8243
Fax) 0562-279-5998
Email) jshin · msi2.postech.ac.kr
학사, 포항공과대학교, 산업공학과, 1996.



조 현 보

경북 포항시 남구 효자동 산 31
포항공과대학교 산업공학과
전화) 0562-279-2204
Fax) 0562-279-5998
Email) hcho · postech.ac.kr
포항공과대학교 산업공학과, 조교수,
1994.9.16~Knowledge Based Systems, Inc., 수석연구원
1994.3~1994.8 박사, Texas A & M University, 산업공학과, 생산공학 전공
1993 석사, 서울대학교, 산업공학과 1988 학사, 서울대학교, 산업공학과, 1986.