

# An Intelligent Planner of Processing Equipment for CSCW-based Shop Floor Control in Agile Manufacturing

Hwajin Kim, Hyunbo Cho, Mooyoung Jung

Dept. of Industrial Eng., Pohang University of Science and Technology,  
Pohang, 790-784, Republic of Korea

## ABSTRACT

A common control model used to implement computer integrated manufacturing(CIM) is based on the hierarchical decomposition of the shop floor activities, in which supervisory controllers are responsible for all the interactions among subordinates. Although the hierarchical control philosophy provides for easy understanding of complex systems, an emerging manufacturing paradigm, agile manufacturing, requires a new control structure necessary to accommodate the rapid development of a shop floor controller. This is what is called CSCW(computer supported cooperative work)-based control or component-based heterarchical control. As computing resources and communication network on the shop floor become increasingly intelligent and powerful, the new control architecture is about to come true in a modern CIM system. In this paper, CSCW-based control is adopted and investigated, in which a controller for a unit of device performs 3 main functions - planning, scheduling and execution. In this paper, attention is paid to a planning function and all the detailed planning activities for CSCW-based shop floor control are identified. Interactions with other functions are also addressed. Generally speaking, planning determines tasks to be scheduled in the future. In other words, planning analyzes process plans and transforms process plans into detailed plans adequate for shop floor control. Planning is also responsible for updating the process plan and identifying/resolving replanning activities whether they come from scheduling or execution.

Key words : CSCW-based shop floor control, component-based heterarchical control, planning, process plan

## 1. 서론

지금까지의 많은 생산 시스템에서는 주로 hierarchical control 모델, 즉 top-down decomposition에 기반을 둔 시스템을 사용하고 있다. 이러한 모델에

서는 생산 시스템의 구성이 상하관계(master-slave relationship)로 이루어지며(그림 1 의 (b) 참조), 생산을 위해 필요한 모든 활동들이 supervisor(상위 단계)를 통해 계획, 관리, 그리고 통제되게 된다. 이와 같은 control 모델이 널리 사용되어지고 있는 이유는 먼

저 전통적인 생산 관리 체계와의 유사성에서 찾아볼 수 있고, 따라서 복잡한 시스템을 이해하기 쉽도록 해주며 기존의 생산 현장에 도입하는 것이 용이하기 때문이다. 또한 시스템 내의 top-down 의사 결정 방식은 관리자로 하여금 적용하기 쉽도록 해 준다. 그러나 최근에는 고객들의 다양한 요구로 인해 제품 수명 주기가 짧아지고 고도의 제품 품질이 요구되는 등 시장 환경의 끊임없는 변화로 인해 생산 시스템에 민첩성(agility)이 요구되는 민첩 생산(agile manufacturing) 시스템의 도입이 절실해지고 있다. 또한 controller 및 통신망(communication network) 기술의 급속한 발달로 인하여 기존의 hierarchical control 모델로써는 이러한 기술의 발전을 효율적으로 활용하기 힘들며 변화하는 시장 환경에 효과적으로 대처하기 힘든 실정이다. 그러므로 민첩 생산 시

스템에 적합한 새로운 control 모델의 연구 및 도입이 필요한 시점이다.

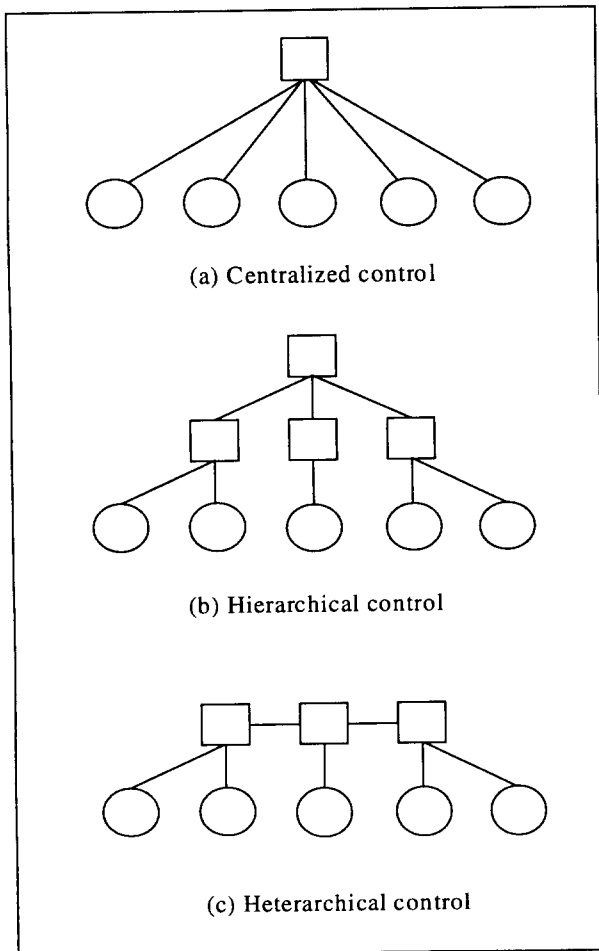
본 연구에서는 이를 위해 컴퓨터 지원 협력 생산(Computer Supported Cooperative Work)에 기반을 둔 control 모델을 채택하였으며, 이는 hierarchical control 모델에 대해 상대적인 모델로서, heterarchical control 모델이라고도 불리운다. Heterarchical control 모델은 완전한 local autonomy를 추구하고, global 의사 결정을 하는데 있어서 cooperative approach를 사용한다는 것이 기본 개념이다. 그러므로, [그림 1]에서 보는 바와 같이 heterarchical control 모델은 hierarchical control 모델과는 달리, 계층적인 상하 관계(master-slave relationship)가 존재하지 않으며, 개개의 component들이 동등한 권한을 가지고 서로간의 협력을 통해 생산 활동을 수행하게 되며, 따라서 supervisor가 존재하지 않고 그 역할을 component들이 분담해서 수행하게 된다.

Heterarchical control 모델을 생산 시스템에 적용시키기 위하여, 본 연구에서는 일반적인 SFCS(Shop floor control system)의 고찰을 2장에서 다루고 3장에서 CSCW-based SFCS의 구성 및 그 시스템을 구성하는 각 component들에 대해서 언급한다. 그리고 본 연구에서 다루고 있는 processing equipment 내의 세 기능들(planning, scheduling, execution)이 어떻게 작업을 수행해 나가는지에 대해서, 그리고 그 중 planning 모듈(Planner)이 수행해야 할 일들을 4장에서 규정한 후, 마지막 장에서 결론을 내린다.

## 2. 일반적인 SFCS의 고찰

SFCS 모델은 크게, [그림 1]에서와 같이 centralized control, hierarchical control, 그리고 본 연구에서 다루게 될 heterarchical control의 세가지로 분류된다.

centralized control 모델은, mainframe computer 한대가 shop에서 이루어지는 모든 생산 활



[그림 1] Control Architecture

동을 계획하고 또한 관련된 정보들의 처리 및 유지를 담당하는 시스템을 말한다(그림 1 의 (a) 참조). Hierarchical control 모델로 발전하기 전단계라고 할 수 있는 이 모델은 다음과 같은 몇 가지 장점을 가지고 있다.

- 시스템에 필요한 모든 관련 정보들을 하나의 controller에서 처리하기 때문에 global optimization 을 가능하게 한다.
- 다른 모델에 비해 controller 비용을 줄일 수 있고 관리가 용이하다.
- 시스템의 필요한 정보 및 상태를 쉽게 얻을 수 있다.

그러나 이 모델은 시스템의 규모가 커지는 경우, 다른 모델(distributed control system)에 비해 상황에 대처하는 능력이 떨어지게 된다. 즉 실시간(real time) control이 어려워지는 단점이 있고, 또한 중앙의 mainframe computer 한대로써 모든 정보를 처리하기 힘들어진다. 두 번째로는 mainframe computer에 이상이 생겼을 경우, 전체 시스템에 문제가 발생한다. 그리고 시스템 내의 모든 관련 data를 다루기 때문에 control software의 유지, 보수 및 확장이 힘들다는 단점이 있다.

Hierarchical control 모델은 서론에서 언급한 바와 같이, 시스템을 상부 구조, 하부 구조라는 계층별로 구성하는 모델을 말한다(그림 1 의 (b) 참조). 시스템을 구성하는 계층은 일반적으로, shop, workstation, equipment의 세단계로 나뉘며, 상부 구조에서 의사결정이 이루어진 후 하부 단계로 내려오게 된다. 이 모델이 갖는 장점은,

- control 구조가 계층별로 구성되어 있기 때문에 software를 개발하는데 있어 모듈화가 가능하다. 따라서 관리나 확장이 용이한 편이다.
  - 각각의 계층별로 역할이 분담되어 있기 때문에 상황에 대처하는 반응시간이 빠르다.
- 는 점이 있고, 단점은 다음과 같다.
- 하위 단계는 상위 단계에 의존하게 되므로 어떤 부분에 이상이 생기면 그 하위 단계의 controller들

도 기능이 마비된다..

- hierarchical structure는 초기 디자인 단계에서 확정, 설치되므로, 미래의 예측하지 못한 변화에 대처해서 수정, 확장하기 어렵다.

Heterarchical control 모델은 hierarchical control 모델의 단점을 극복하기 위하여 제안되었는데, 앞에서 언급한 바와 같이, 통신망(communication network), 분산 컴퓨팅 시스템(distributed computing systems) 분야와 controller 기술 발전이 그 배경이 되고 있다. 이 모델은 시스템 내에 계층적 구조가 존재하지 않는 대신, 시스템을 구성하는 구성원들 사이의 협력을 통해 원하는 목적을 달성한다. 즉, global 정보를 최소화시키고 대신에 local database들을 활용하며, controller간의 communication을 통한 협의(negotiation)하에 의사 결정을 한다.

Local controller들이 독립적으로 작업을 수행하기 때문에 plug and play 개념을 도입할 수 있어서 시스템의 확장이 용이하며, 한 controller의 이상 발생이 전체 시스템에 미치는 영향을 최소화시킬 수 있다. 또한 현재의 shop 상태를 기반으로 의사 결정이 이루어지므로, 실시간 control이 가능하며, component별로 모듈화 됨으로써 controller software complexity가 줄어들며, 유지, 관리가 용이한 장점을 지닌다. 단점으로는 local optimization이 발생할 수 있으며, controller 자체의 기술적인 제약이 있을 수 있다.

### 3. CSCW-based SFCS

Heterarchical control을 구현하기 위해 고려되어야 할 문제들은 다음과 같은 것들이 있다.

- Supervisor를 어떻게 제거할 수 있고 그 역할은 어떤 식으로 수행되는가?
- Supervisor 없는 상태에서 어떻게 최적화(optimization)을 이룰 것인가?
- Part data는 어디에 저장되는가, 또한 어떻게 관리 되는가?

- Part 가공에 대한 정보는 어떻게 관리되는가?
- 생산 자원(machine, tool 등)은 어떻게 관리되는가?
- 시스템의 deadlock은 어떻게 피할 것이며, 어떻게 해결할 것인가?
- Component는 어떻게 정의할 것인가?
- Component 들간의 communication protocol 은 어떻게 정의되는가

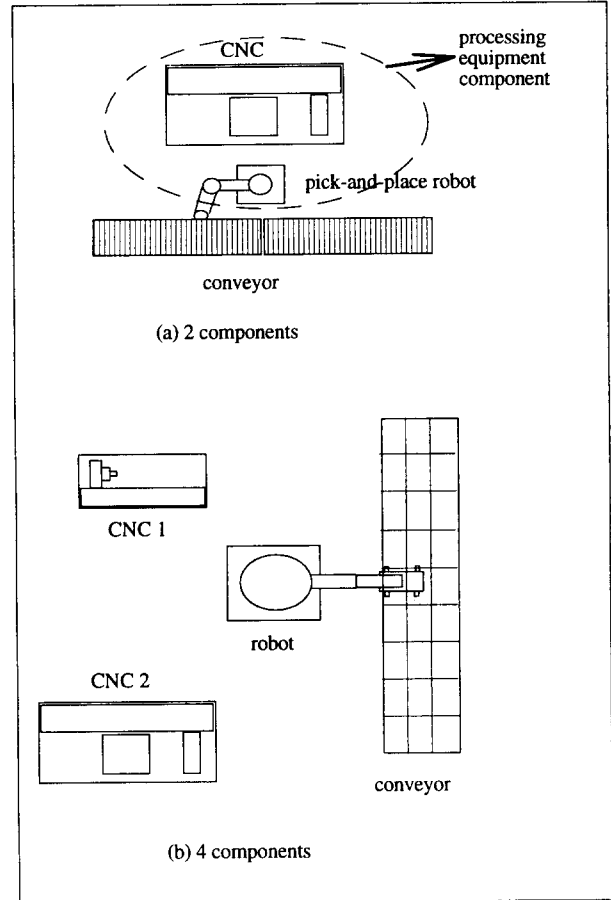
### 3.1 Component 의 정의 및 분류

Heterarchical control 모델을 구축하기 위해서는 생산 시스템을 구성하는 component들을 정의할 필요가 있다. 각 component들은 의사결정을 내릴 수 있는 적절한 단위로서, 이들은 서로 독립적으로 작동하며 통신망을 통해 서로간의 협력으로 생산 활동을 수행하는 생산 자원의 집합체이다. 하나의 component는 하나의 controller에 의해 제어 및 통제된다. 또한 생산 자원에 대한 동등한 이용 및 접근할 수 있는 권한을 가진다. Component들은 다음과 같이 분류될 수 있다.

- processing equipment : part 가공을 담당한다.
- AGV : 물류 흐름을 담당한다.
- conveyor : 물류 흐름을 담당한다.
- robot : 물류 흐름을 담당한다.
- ASRS : 자동 창고
- 공용 버퍼 : part의 임시 저장을 담당한다.
- assembly equipment : part 조립을 담당한다.
- inspection equipment : 제품 검사를 담당한다.
- InitMon component(initiating and monitoring component) : SFCS의 초기화 및 작업 진도 상황에 대한 모니터링을 담당한다. 즉, part 가공 정보, 조립 정보 등 다른 component들이 필요로 하는 정보를 생산 시작 단계에서 분배하는 역할을 하며, 생산 활동 중 component들로부터 이상 발생이나 작업 진도 상황에 대한 message를 전달 받아 관리자에게 display해주는 기능을 담당한다.

다.

- resource manager : shop 내의 생산 자원 및 생산 자원들 사이의 관계 등에 대한 정보의 유지,관리를 담당한다.

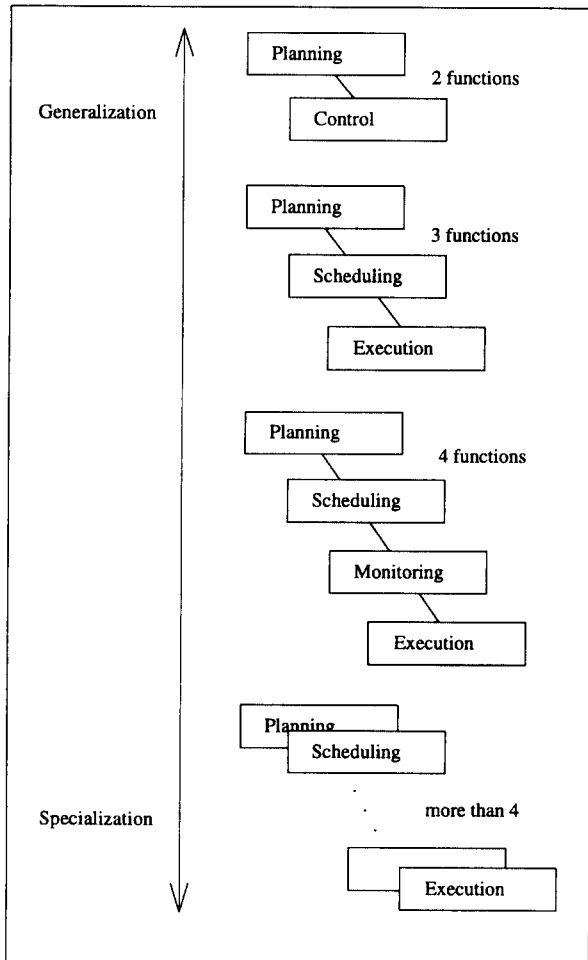


[그림 2] component의 예

이상에서 언급한 component들이 의미하는 바는 고정적인 것이 아니라 상황에 따라 달라진다. 즉, [그림 2]의 (a)에서와 같이 단순히 part를 집어서 놓는 기능만 가진 robot의 경우는 processing equipment component내에 포함되어지고, (b)의 경우 처럼 NC machine들(CNC 1, CNC 2)과 communication을 통해 의사 결정이 이루어져야 하는 경우는 robot component로 분리된다.

### 3.2 Component의 기능

위에서 언급한 component 각각의 경우에 대해서 [그림 3]와 같이 내부적으로 수행하는 기능을 분류할 수 있다. 본 연구에서는 component의 기능을 planning, scheduling, execution의 세가지로 분류하는 방법을 택한다.



[그림 3] 기능 분류 방법

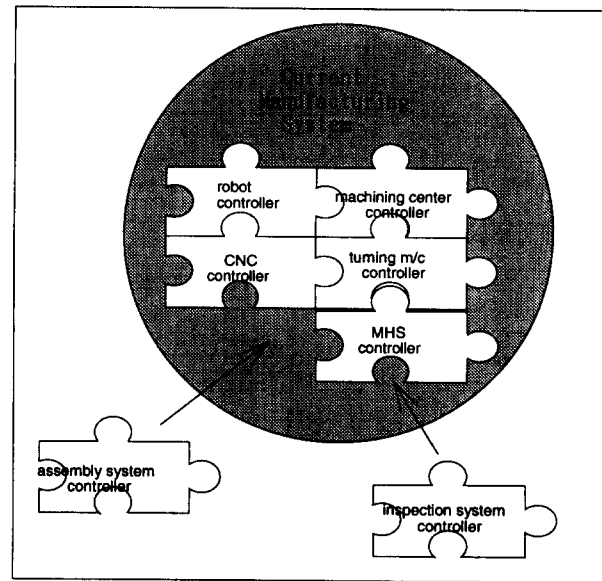
Planning function은, 앞으로 scheduling function이 수행하게 될 일련의 작업들을 결정, 생성하는 역할을 한다. 이는 scheduling function이 해결해야 할 문제들보다 발생 빈도수가 작은 문제들로서, 결과적으로 문제의 범위를 좁힘으로써 scheduling function의 부담을 덜어 주고 따라서 생산 활동을 실시간으로 제어할 수 있게 한다.

Scheduling function은 planning function에서 보내진 문제들을 해결하며, execution function에서 수

행하게 될 작업들을 생성해 낸다.

Execution function은 scheduling function에서 생성된 작업들을 실질적으로 수행하는 역할을 하며, 다른 component들과의 통신 및 현재 작업 상태에 대한 message 전달 등의 기능을 맡는다.

위의 세 기능들은 필요한 일을 수행하기 위해 function들 서로간의 communication뿐만 아니라 다른 component들과의 communication을 통해 협력 작업을 수행한다.



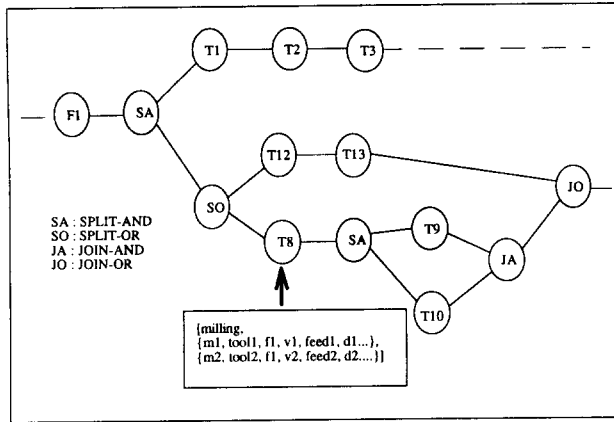
[그림 4] Rapid development

이상과 같은 component에 기반을 둔 heterarchical control 모델(component-based heterarchical control)을 구현함으로써 시스템의 확장, 수정이 용이해지고 따라서 민첩성과 유연성이 요구되는 현대의 생산 시스템에 쉽게 대처할 수 있게 된다(그림 4 참조). [그림 4]에서 보는 바와 같이 현재의 생산 시스템을 확장하고자 할 때, 필요한 component들(assembly, inspection)을 기존의 시스템에 단순히 끼워 넣기만 함으로써 신속하게 새로운 시스템을 구성할 수 있다.

#### 4. Planning function of Processing Equipment

#### 4.1 Process plan representation

process plan은, processing equipment의 planning이나 scheduling function이 알아야 할 part 가공 순서나 생산 자원(machine, tool 등)등에 대한 정보를 담고 있어야 하고, 또한 그에 대한 여러 가지 대안을 표현할 수 있어야 한다. 본 연구에서는 process plan을 표현하기 위한 방법으로 AND-OR graph 개념을 이용한다. 이 graph에는 서로 다른 5가지 node로 구성된다(그림 5 참조).



[그림 5] Process plan representation 의 예

- SPLIT-AND, JOIN-AND : 이 두 node 들 사이의 작업들은 임의의 순서로 모두 수행되어야 함을 의미한다(예 : T9, T10 혹은 T10, T9 순으로 작업).
- SPLIT-OR, JOIN-OR : 이 두 node 들 사이의 작업들은 여러 대안들 중 한가지 작업 순서만 선택되어야 한다(예 : {T12, T13} 혹은 {T8, (T9, T10)} 중 한 경로를 선택).
- FORM-FEATURE : 이 node는 machine, tool, 절삭 조건 등의 part 가공 정보를 표현한다(예 : T8).

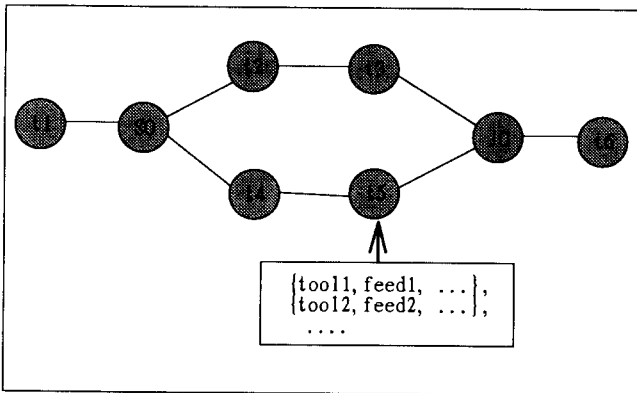
#### 4.2 Planner

Processing equipment의 planner 가 수행해야 할 일은 위 3장에서 언급한 일반적인 planning function 역할에 기반을 둔다. Planner 는 InitMon component(생산 시작 초기), 혹은 다른 processing equipment로 부터 받은 process plan 정보를 이용하여 scheduling 문제들을 생성해 냄으로써 scheduling function이 내려야 할 의사 결정의 범위를 좁히는 역할을 한다. 또한 process plan 정보를 관리, 갱신하는 역할도 맡는다.

Process plan 정보를 이용하여 planner가 수행하는 일은 크게 두 가지로 분류할 수 있다. 첫째는 part 가공을 위한 자원(machine, tool)을 선택하는 일이다(manufacturing resource selection problem). Process plan의 FORM-FEATURE node에는 작업을 수행할 수 있는 자원들에 대한 몇 가지 대안이 표현되어 있는데(그림 5 참조), 이 중 적합한 것을 선택하는 일이 planner가 해야 할 일이다. 두 번째는 동일한 결과를 얻을 수 있는 여러 가지 작업 경로들 중 최적의 것을 선택하는 일이다(process path selection problem). [그림 5]에서 예를 들면, SPLIT-OR 와 JOIN-OR 사이의 경로들 중 하나, 즉 {T12, T13}와 {T8, (T9, T10)} 중 최적의 것을 선택하는 문제이다. 또한 이들 문제들이 동시에 발생하는 경우(concurrent problem)도 고려하여야 한다.

기존의 hierarchical control 모델에서는 문제 해결을 위해 필요한 현장의 정보들을 추정, 예측에 의해 구한 후 off-line 상태에서 문제를 해결하였으나, 본 연구에서 구현하고자 하는 모델은 모든 것이 실시간 적으로 처리되므로, component 내부의 다른 function들(scheduling, execution) 뿐만 아니라 다른 component들과의 협력을 필요로 한다. 만약 한 processing equipment(MC1)가 [그림 5]의 F1 node를 수행해야 하고, F1 node의 내용이 아래 [그림 6]과 같다면, 그 MC1 의 planner는 적합한 tool및 절삭 조건들을 결정하고 최적의 가공 경로(즉, {t2, t3}나 {t4, t5}중의 하나)를 선택한 후, scheduling function에 가공 정보를 전달한다. 이때, scheduling function으로

부터 다시 replanning 요청이 올 수도 있다. MC1에서 이루어져야 할 작업이 완료되면, planner는 그 다음 작업을 누가 수행할 것인가를 정하기 위하여 다른 processing equipment들의 planner와 협의를 하게 된다. 예를 들어, [그림 5]에서 현재 T8 node의 작업을 수행하였다면, T9, T10 node 들과 관련된 processing equipment들과 협의를 거쳐 그 중 하나가 선택된다. 위에서 언급한 것처럼 이상과 같은 문제들이 동시에 발생하는 경우도 고려해야 한다. [그림 5]에서 현재 F1 node 의 작업을 수행하였고, 다음 작업을 수행할 processing equipment를 고를 경우에는, resource selection problem 과 process path selection problem이 동시에 발생한다.



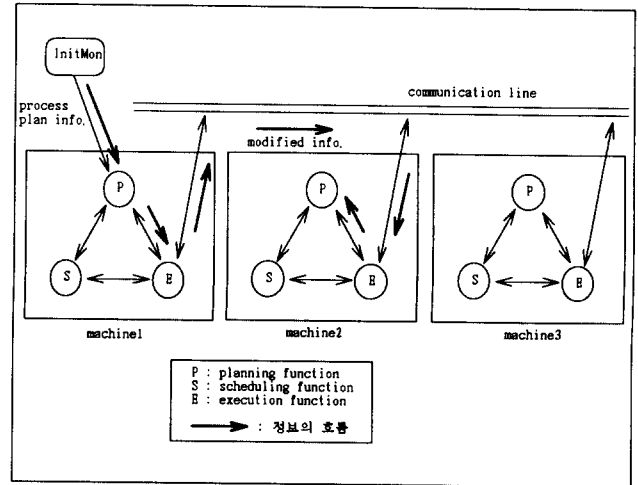
[그림 6] MC1에서 수행할 작업(F1 node)

#### 4.3 Heterarchical control 모델을 이용한 시나리오

[그림 7]은 InitMon component로부터 가공 정보가 processing equipment로 전달되었을 때, component 내부의, 그리고 component 간의 상호 작용 및 정보의 흐름을 보여 준다.

생산 시작 초기에, 각 processing equipment 와 InitMon component사이의 협의를 통해 part 가공을 최초로 시작하게 될 machine1 이 선택된 후 process plan 정보가 전달된다. machine1의 planner는 이 정보를 이용하여 앞에서 언급한 기능들을 수행한 후, scheduling function에 필요한 가공 정보를 넘겨 준다. 작업이 완료되면, planner는 다음 작업을 수행할

processing equipment를 선정하기 위하여 다른 component들과의 협상을 시도한다. Component들 사이의 협의를 통해 다음 작업을 수행하기 위하여 가장 적절하다고 판단된 machine2 로 process plan 정보가 전달된다. 이때, 그림에서도 알 수 있듯이 component 들간의 통신은 execution function을 통해 수행된다. Part 가공이 끝날 때까지 이러한 과정이 반복된다.



[그림 7] Process plan 정보의 흐름

## 5. 결론

3장에서, heterarchical control 모델을 구현하기 위하여 여러 가지 해결해야 할 문제들을 언급했다. 먼저 언급된 것은 hierarchical control 모델에서의 supervisor 역할을 heterarchical control 모델에서 어떻게 구현할 것인가에 관한 문제이다. 앞에서 설명한 것처럼 생산 활동을 위한 의사 결정 역할은 시스템을 구성하는 각 component들에게 분배되며, 이들 component 들이 협의를 통하여 결정을 내리게 된다. 그러나 최적 해를 구하는 과정에서 local optimization 경우를 방지할 수 있는 알고리즘이 개발되어야 한다. 그 외 현재 시스템의 상태에 대한 정보 및 이상 발생에 관한 정보를 관리자에게 전달하는, 즉 시스템과 관리자의 교량 역할은 InitMon component 가 담당한다. Part 가공에 대한 정보는 processing equipment 가 유지, 관리하며, 다음 작업을 수행할 processing

equipment로 전달될 때 갱신된다. part data 및 machine, tool 등에 대한 정보는 resource manager가 관리한다.

4장에서 규정한 planner의 역할을 효율적으로 수행할 수 있는 알고리즘 개발과, 시스템의 deadlock 방지 및 component들간의 communication protocol 을 정의하는 문제는 지속적인 연구를 필요로 한다.

## 참고 문헌

- [1] Neil A. Duffie, Res S. Piper, "Nonhierarchical Control of Manufacturing Systems," *Journal of Manufacturing Systems* Volume 5 No. 2, 137 - 139 (1986).
- [2] Cho, H., *An Intelligent Workstation Controller For Computer Integrated Manufacturing*, Ph.D. Dissertation, Texas A&M University (1993).
- [3] Richard A. Wysk, Richard J. Mayer and Hyuenbo Cho, "Shop-Floor Scheduling and Control: A System Approach," *Proceeding of DGOF/OFSA conference on recent developments and new perspectives OR in manufacturing*, Hagen, Germany (June 25-26 1992).
- [4] D. M. Dilts, N. P. Boyd and H. H. Whorms, "The Evolution of Control Architectures for Automated Manufacturing Systems," *Journal of Manufacturing Systems* Volume 10 No. 1.
- [5] Lee, K., *Formal Integration of Process Planning and Shop Floor Control for CIM*, Ph.D. Dissertation, Pohang University of Science and Technology (1994).
- [6] Lin, G., Y-J., and Solberg, J. J., "Integrated shop floor control using autonomous agents," *IIE Transactions* Volume 24, No. 3, 57-71 (1992).