

CIM 공장의 단위 CELL 운용 방안 연구

김 성식 최 기범 김 진호
고려대학교 공과대학 산업공학과

A CELL operation scheme in CIM Factory

KIM S.S. CHOI K.B. KIM J.H.
Dept. of Industrial Engineering Korea University

Abstract

The objective of cell operation in CIM is to maximize system efficiency, not the cell itself.

We introduce a cell operation scheme that pursues the direction. For specific cases, work center and AS/RS are closely investigated. Structures of their operation mechanism and methods of job scheduling are introduced along with an expert system developed for the scheme.

The cell operation softwares developed are now under test at K.U.FMS, a model CIM plant.

본 연구에서는 전체 공장의 운영 최적화라는 전체하에서 단위 CELL(작업장, 자동창고)의 작업처리 방식과 단위기능의 모듈화 방안을 제시한다. 특히 작업순서를 결정하는 부분과 일부 모듈은 전문가 시스템화하여 Dynamic한 공장 상황의 변화에 대한 운영 시스템의 유연성(Flexibility)을 높이고 실시간 처리를 가능하게 한다.

2. 운영 시스템의 구조와 역할

공장 운영 시스템은 구성하는 사람의 시각에 따라 달라질 수 있으나 기본적으로 다음과 같이 3 단계의 수직적 계층구조를 갖는다. [1]

1. 연구의 배경 및 목적

자동화된 공장의 주요 구성요소로는 실제 물품을 가공하는 작업장(CELL), 공장내를 흘러다니는 물품을 보관하고 필요한 시기에 공장내로 공급 해 주는 자동창고(AS/RS), 이들의 흐름을 연결 해 주는 자재 운반 시스템, 가공품의 불량유무를 검사하는 검사 시스템 그리고 전체적으로 이들을 통합, 조정하는 중앙컴퓨터로 구분 해 볼 수 있다. 이중 중앙 컴퓨터를 제외한 나머지 구성요소들은 각자 자신의 영역을 통제하는데 이들이 수행하는 작업은 다르지만 각 운영 시스템의 구조와 기능은 동일하다고 볼 수 있다. 본 연구에서는 이들을 단위 CELL이라고 정의한다.

본 연구는 계층적 구조를 가진 분산 통제 시스템하에서 이들 단위 CELL중 작업장과 자동창고에 있어서의 운영방안을 제시하고자한다. 그간의 여러 연구 결과들을 보면 독립된 작업장이나 자동창고 자체 내의 운영 방안이나 스케줄링 기법들을 선보였지만 이들을 하나의 시스템으로 통합한 상황에서의 운영방안에 관해서는 그다지 활발한 연구가 되어있지 못했다. 이러한 결과들은 CIM공장의 운영에있어서 전체의 효율보다는 각 부분 단위 CELL의 최적화에 초점이 맞추어져 있기 때문에 이 결과들의 단순 결합이 곧바로 공장 전체의 운영 최적화에 연결된다고 볼수는 없다. 따라서 각 단위 CELL의 최적화가 아니더라도 공장 전체의 운용 효율을 높일 수 있는 방안이 연구되어야 한다.

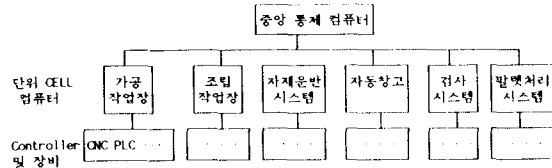


그림 1 : CIM 공장 통제 계층의 기본 구성

중앙 통제 컴퓨터는 공장 전체의 작업 흐름을 통제하는 상위 컴퓨터로서 작업주문(Order)에 대한 일정계획을 세우고 각 하부 통제 시스템으로부터 보고되는 정보를 바탕으로 공장이 계획된 일정에 따라 효율적으로 운영될 수 있도록 감독한다. 단위 CELL 컴퓨터들은 각각 자기가 담당한 구역을 통제하고 Controller 및 장비에 대하여 중앙 컴퓨터와 유사한 역할을 수행한다.

각 단위 CELL이 보고하는 여러 정보들은 중앙 컴퓨터에서 공장 내의 다른 단위 CELL의 상황과 비교한후에 전체 공장 상황을 나타내는 정보로 재처리하여 하부 단위 CELL에 전달한다. 그러면 각 단위 CELL에서는 전달받은 정보와 관련된 작업에 대한 적절한 대응을 취함으로써 전체 공장의 효율을 향상시키는 방향으로 작업을 진행시킨다. 본 연구에서는 이와같은 목적을 위해 사용하는 정보들 중에 하나로 다음과 같은 작업물의 등급(Class)을 고려한다.

① EE(Extra Expedition) : 납기를 맞추기위

해 매우 긴급히 처리해야할 작업

② E(Expedition) : 다음 작업장의 부하량이 적게 걸려

있어 빠르게 처리해야할 작업

③ N(Normal) : 정상적인 속도로 처리해도 무방한 작업

④ D(Delay) : 다음 작업장의 부하량이 과다하게 걸려

있어 지연시켜야할 작업

이 등급은 공장 전체의 상태를 반영하는 정보로 스케줄링 시 중요한 고려사항중 하나이다. 그러므로 단위 cell들은 자신의 운용을 최적화하는 과정에 이러한 전체 공장에 대한 정보를 반영함으로써 공장 전체 시스템에 대한 최적화도 병행하게 된다.

또한 여러 단위 CELL들간에 작업이 동시에 이루어질 때에는 중앙 컴퓨터를 매개로하여 단위 CELL들이 시간지연등의 제 문제를 최소화할 수 있도록 한다.

다음 3장과 4장은 각각 작업장과 자동창고 운용에 있어서 내부적으로 기능의 모듈화를 통한 시스템 유연성의 향상과 실시간 통제라는 목적과 외부적으로 전체 시스템의 최적화라는 목적을 동시에 달성하기위한 방안을 운영 시스템의 설명과 함께 제시한다.

3. 작업장 운영 시스템

3.1 제어 구조

작업장을 구성하는 여러가지 요소들 즉, 공작 기계나 자재 운반 시스템 과 같은 물리적인 요소들 뿐만아니라 작업 처리 방법의 기준이나 목표, 기계 고장과 같은 이상상황의 처리 방법 등등의 요소들의 상태는 공장 내외의 환경에 따라 계속적으로 변화한다. 따라서 작업장 운영 시스템이 여러 다양한 실제 작업환경에서 성공적으로 임무를 수행하기 위해서는 이러한 요소들의 변화에 대응한 시스템의 변경이 용이하게 이루어질수 있어야 한다. 일반적으로, 운영 시스템의 유연성(Flexibility) 이나 확장성(Extendability)으로 통칭되는 이러한 성질들은 그 운영 시스템의 각 기능들에 대한 모듈화의 정도에 깊은 영향을 받는다. 또한 운영 시스템의 제어 구조와 통제 방법 역시 매우 중요하게 고려해야할 사항중에 하나이다.

본 연구에서는 작업장 운영 시스템의 제어 구조로 Production System의 구조와 방법을 응용하였다. 이것은 Production System이 지식과 제어를 분리시켜 지식이나 제어의 변경을 독립적으로 수행할 수 있게 하고 규칙들 간의 직접호출을 방지하여 문법적인 상호연관을 제거함으로써 Production 규칙들의 모듈화를 용이하게 하기때문이다. 또한 제어방식에 많은 유연성을 제공하며 프로그래밍 언어에 영향을 받지않는다.[3]

작업장 운영 시스템의 제어 구조를 도시하면 다음과 같다.

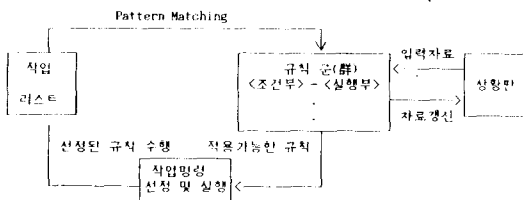


그림 2 : 작업장 운영 시스템의 제어 구조

다음 절에서는 작업장 운영 시스템을 구성하는 네 부분에 대하여 설명하고 각 부분에서 작업의 실 시간 처리와 시스템 모듈화 그리고 전체 공장의 효율적 운용을 위한 방안을 제시한다.

3.2 규칙들의 집합

조건부와 실행부로 이루어진 규칙에서 조건부에는 규칙이 문제 상황에 적용될지를 결정하는 Pattern들이 논리합, 논리곱으로 연결되어 있고 실행부는 본 시스템에서 '상황판' 으로 표시되는 Problem solving knowledge의 집합에서 필요한 정보를 입력하여 작업을 수행하는 프로시저어 (Procedure) 로서 각 모듈과 일부 함수들이 이에 해당한다.

규칙의 한 예를 보면 공장 가동시에 각 운영 시스템들간의 통신 라인을 개설하고 검사하는 작업을 수행하는 규칙은 조건부에 "COMMUNICATION_LINE_CONNEC T" 라는 pattern을 가지고 실행부에서는 외부통신 초기화 모듈인 'EPC_INITI ALIZATION()' 을 호출한다.

3.3 작업 리스트

실행할 작업들에 대한 정보들의 집합체로서 규칙의 조건부와 상응될 수 있는 pattern들의 집합과 이들 pattern 집합중에서 가장 먼저 사용할 pattern을 결정하기위해 필요한 정보들로 이루어져 있다. pattern을 결정하는데 사용되는 정보에는 내부의 작업흐름에 대한 정보뿐만 아니라 외부의 단위 CELL에 대한 정보도 포함된다. 즉 규칙의 결정에 공장 전체의 상태 정보를 반영함으로써 작업의 진행을 전체 시스템 최적화의 방향으로 유도한다. Pattern 집합은 상응하는 규칙의 특성에 따라 구분하여 그룹화하는데 상황에 따라 각 그룹에 우선순위를 주어 Pattern-Search에 효율을 높인다.

3.4 상황판

규칙의 실행부가 작업을 수행할때 입력자료로 요구되는 모든 정보들의 집합으로서 작업 리스트와 분리되어 있다. 상황판은 작업 리스트와 함께 규칙간의 정보를 유통시켜주는 부분으로 규칙 상호간의 직접호출을 제거하여 시스템의 모듈화를 용이하게할 수 있도록 해준다. 즉 사용자가 새로운 모듈을 추가시키거나 기존 모듈을 변경할때 시스템에 대하여 알아야할 정보로는 다만 상황판과 규칙의 Pattern들이 필요할 뿐이다.

3.5 작업명령 선정 및 실행부분

본 연구에서는 이 부분을 전문가 시스템화 하였는데 크게 두가지 과정으로 구분할 수 있다. 첫째는 작업 리스트 상에 표현된 복수의 pattern 들 중에서 하나의 pattern을 선택하는 과정이고 두번째 과정은 선택된 pattern 과 규칙의 조건부를 비교하여 적용가능한 규칙들을 모아 이들중에서 하나의 규칙을 선택하여 실행하는것이다. 두번째 과정은 메타 규칙(META RULE)들로 구성된 부분으로 Heuristic

을 포함한 여러가지 통제 방식을 상황에 따라 유연하게 적용할 수 있다. 이 부분은 작업장 운영 시스템의 작업진행을 결정하는 중요한 부분으로 작업명령 선정에는 작업장 내의 정보, 외부 단위 CELL의 정보 그리고 규칙들 간의 우선순위가 고려된다. 규칙들간의 우선순위는 전체 시스템의 효율과 관련되는 규칙이 작업장내의 일과 관련된 규칙보다 높다.

예를들어, 외부 단위 CELL과의 작업물 교환시 한 단위 CELL에서 해당 규칙의 처리를 지연시키면 관련된 모든 단위 CELL 또한 작업이 지연되므로 한 단위 CELL의 작업 효율은 상승할지라도 전체 시스템의 효율은 떨어지게 된다. 따라서 이런 상황에서는 해당 규칙의 우선순위를 높이는 것이 전체 시스템 최적화에 타당하다. 또한 외부 단위 CELL의 고장과 같은 정보는 전체 시스템에대한 해결책이 나올때까지 해당작업을 유보하여 문제가 더욱 복잡해지지 않도록 한다.

이상에서 운영 시스템을 구성하는 네 부분에 대하여 내용과 기능을 고찰하였다. 이밖에도 작업장 내외의 환경이 변함에 따라 운영 시스템의 유연한 대응이 요구되는데 다음절에서 운영 시스템의 수정에 대한 방법을 살펴본다.

3.6 작업장 운영 시스템의 수정

본 연구에서는 각 운영 시스템을 구성하는 네가지 부분(규칙들의 집합, 작업 리스트, 상황판, 작업명령 선정 및 실행부분)간의 상호 연관을 배제함으로써 운영 시스템의 수정이 네가지 부분 각각에 대해서 독립적으로 수행된다. 따라서 운영 시스템의 변경시 USER INTERFACE 를 통하여 사용자가 해당 부분만을 수정하면 변경이 완료되게된다. 즉, 작업장의 작업 통제 방식을 변경시키고자 할때는 오직 작업명령 선정 과정의 메타 규칙을 수정하는것 만으로 가능하다. 만일 운영 시스템의 기능이 바뀌었을 경우에는 각 규칙이 서로 독립적인 성격을 가지기때문에 해당 규칙만을 교체하거나 수정하면 된다. 작업 리스트나 상황판의 수정은 해당 데이터 구조에 따라 정보를 수정한다.

다음 절에서는 작업장 운영 시스템의 주요 기능모듈인 스케줄링 모듈과 異常 狀況 中에 내부와 외부 고장에 대한 처리 모듈에서 실시간 처리와 모듈화의 구현 그리고 전체 시스템의 최적화에 대한 보다 구체적인 방안을 살펴본다.

3.7 스케줄링 모듈

공장의 한 구성요소로서의 작업장에서 공장 전체의 효율을 우선하는 스케줄링에는 독립적인 작업장의 스케줄링에 비하여 여러가지 추가적인 제약조건이 고려되어야 한다.

본 연구에서는 외부의 단위 CELL에 매겨진 등급(Class)에 따라 작업장내의 각 작업물의 등급을 진행 경로(Route)에서 바로 다음 단위 CELL의 등급과 동일하게 두어 스케줄링의 제약조건에 포함시켰다. 단위 CELL의 등급은 그당시 단위 CELL내의 버퍼(Buffer)의 여유와 작업물의 작업완료

시점에 따라 등급 조정에 대한 정보가 중앙 컴퓨터에 보고 되고 중앙 컴퓨터는 이 정보를 공장 전체의 상황과 비교하여 단위 CELL의 등급을 판단한다. 이 제약 조건은 타 단위 CELL의 부하량에 대한 정보가 작업장의 작업순서 결정에 영향을주게 함으로써 전체 단위 CELL의 부하량에 시점상의 불균형을 평준화시켜주는 효과가 있다. 또한 다른 단위 CELL에 고장과 같은 이상 상황이 발생하였을 때에는 이와 관련된 작업물은 스케줄링에서 제외시키고 이에 대한 새로운 정보가 왔을때 해당 처리 모듈을 호출하여 관련 작업물을 처리한다.

본 연구에서는 전문가 시스템을 사용하여 이러한 스케줄링 문제를 해결하고 또한 사용자와의 진단 프로그램을 통하여 스케줄링 알고리즘을 새로운 목적에 맞게 변경시킬 수 있도록하였다.

3.7.1 스케줄링 알고리즘의 구현

사용자가 정의하는 규칙은 조건부, 실행부, 파라미터의 세부분으로 나뉜다. 조건부는 규칙이 선정될 상황을 나타내고 실행부는 작업을 수행할 함수들을 호출하는 부분이다. 이때 함수에는 작업 대상의 범위등과 같은 정보를 전달하여 수행시킬수 있는데 이러한 정보의 전달은 파라미터 부분을 통하여 이루어진다. 실행부는 각각의 모듈화된 단위기능을 처리하는 여러 함수들의 조합으로 이루어져 있는데 이들 단위기능은 SPT(Shortest Processing Time) 규칙이나 EDD(Early Due Date) 규칙등과 같이 잘알려져있는 규칙 뿐만아니라 사용자가 직접 구성하는 규칙들을 수행한다.

스케줄링 알고리즘을 구성하는 규칙들의 Search는 Pattern directed Search로 구현하였고 Search의 통제는 Goal-driven 방식을 사용하였다. 알고리즘의 정지(STOP) 조건은 더이상 적용할 규칙이 없거나 GOAL을 만족하는 것이다. GOAL은 스케줄링 알고리즘에 할당된 수행 가능 시간으로 주어지는데 이 시간은 스케줄링이 실시간으로 처리되기위해 필요한 최대시간격으로 상황에따라 변화한다. 스케줄링 알고리즘을 변경하고자 할때는 사용자가 진단 시스템을 통하여 해당 규칙을 수정할 수 있다.

3.8 고장 상황의 발견 및 해결

고장 처리 모듈은 문제의 성격상 전문가 시스템화하는것이 매우 적당한 부분으로 규칙 베이스 시스템을 사용하여 구축하였다. 앞서의 스케줄링 모듈과는 달리 내부 고장 처리 모듈은 복수의 규칙들로 구성되어있다. 먼저 고장원인의 발견에 있어서 문제의 원인들을 유사한 것끼리 묶어 그룹화하고 이들 각 그룹에 대하여 개별적인 규칙 베이스 시스템(Rule base system)을 구축하였다. 또한 고장원인 발견을 총괄하는 마스터(Master) 규칙 베이스 시스템을 두어 체계적인 원인분석을 행한다. 이와 같이 각 원인별로 그룹화하여 독립적인 규칙 베이스 시스템으로 구축하는것은 새로운 고장원인의 추가나 기존 규칙 베이스 시스템의 수정, 삭제를 용이하게 한다.

원인이 발견되면 이에 대한 대처를 위한 규칙을 호출하게 된다. 이 규칙이 수행하는 기능은 작업장 내부에 대한 복구와 중앙 컴퓨터에 예상 복구 시간등의 정보를 보내는 것이다. 중앙 컴퓨터는 이 정보에 따라 공장 상황을 고려하여 고장 처리 정보를 하부 단위 CELL에 전달하는데 이것은 전체 시스템의 효율화를 위해 중요하다. 외부 고장 처리 모듈은 바로 이러한 정보를 중앙 컴퓨터로 부터 전달받았을때 수행된다. 외부 고장 처리 모듈이 수행하는 일은 우선 중앙 컴퓨터로부터 전달된 정보에 따라 상황판과 작업 리스트를 변경한다. 그리고 유보되었던 해당 작업명령과 작업물에 대하여 새로이 전달받은 정보에 따라 취소하거나 재 작업한다. 재 작업시 작업명령과 작업물에 대한 정보는 초기 정보와는 달라지게되는데 이것은 고장 처리시의 공장 상태 정보가 새로이 반영되기 때문이다. 따라서 작업명령 선정이나 스케줄링시 작업장 내부의 최적화에 대한 제약조건 뿐만아니라 전체 시스템의 최적화에 대한 제약조건도 고려되게 되는 것이다.

4. AS/RS 운용 시스템

4.1. CIM공장에서 요구되는 자동창고의 기능

자동화된 공장에서는 입, 출고를 포함한 창고의 물리적인 기능 이외에 공장 전체의 효율을 높이기 위한 운용상의 기능이 필수적이다. 이 기능에는 작업 Scheduling, Rack들의 효율적 사용을 위한 정책, 재고처리 정책, 정보의 정리 유지 제시를 통한 User와의 Interface 기능 등이다. 특히 운용상의 기능을 효과적으로 수행하기 위해서는 단위 CELL들간의 원활한 정보교환이 필수적이다.

4.2. 운용 프로그램의 설계

본 운용 프로그램이 적용되는 AS/RS의 기본 운영정책은 다음과 같다.

공장에서 출입하는 물품의 저장정책은 Class based storage방법을 써서 물품의 종류에 따라 Rack들을 나누어 사용한다. 입, 출고의 실질적 역할을 담당하는 Stacker Crane은 Single command에 의해 운용되며, AS/RS내의 재고 처리에 관한 문제는 다음과 같은 (s, f)정책을 쓰기로 한다. 즉 각 물품에는 할당된 Rack들이 있으므로 만약 할당된 Rack들 중 물품이 들어가는 Rack의 수가 s이하로 떨어지면 shortage 방지를 위해 중앙 컴퓨터에게 Rack Empty 경고신호와 함께 해당 물품의 입하를 요청한다. 마찬가지로 이러한 Rack의 수가 f 이상으로 올라가면 할당된 Rack이 모두 차버리는 상황을 피하기위해 중앙 컴퓨터에게 Rack Fulling 경고신호와 함께 해당 물품의 출하를 요청한다.

이러한 기본적인 운영정책하에서 전체 공장의 효율을 반영하기 위해서는 내부적인 처리가 필요하다. 즉 실제 작업 처리를 위한 Scheduling 기능에 공장내 각 단위 CELL들의 상황을 반영시켜 주어야 한다. 이의 처리는 Host가 중재자의 역할을 하여 단위 CELL의 상황을 작업 Priority의 형태

로 AS/RS에 전송하게 된다. 또한 각 작업에 따르는 Due Time을 맞추는 것도 Scheduling의 고려사항이 된다.

4.2.1. 운용 프로그램의 구분

AS/RS의 운용 프로그램은 다음 5가지로 구분되어 있다. 즉 중앙 컴퓨터나 AGV등과 연락을 취하며 Device control program과 Data base control 프로그램을 통제하는 MAIN 프로그램, Stacker crane과 I/O conveyer를 직접 통제하는 Device control 프로그램, DB handler의 기능을 수행하는 DB control 프로그램, User와의 Interface를 담당하는 프로그램 그리고 Host나 외부의 다른 CELL들에서 들어오는 통신을 받는 Server 프로그램이다.

4.2.2. 필요한 모듈

효과적인 운용 프로그램을 구축하기 위해서는 프로그램의 각 부분들을 모듈화시키는 작업이 필요하다. 이러한 각각의 모듈에는 고유한 기능이 있고 구축 시스템이 변하면 해당 모듈만 변경, 추가시키면 된다. 우선 AS/RS 뿐만이 아니고 CIM공장내의 각 단위 CELL들이 공통적으로 사용하고있는 모듈에는 다음과 같은 것들이 있다.

①통신 기능 모듈: AS/RS 내부에서의 통신은 Message Queue를 사용하고, 외부(중앙 컴퓨터나 AGV 등의 MHS)와의 통신은 Socket을 이용한다.
②Code 초기화 모듈: 공장 운용시 사용하게 될 Code들을 Tree structure로 초기화 시켜놓고, 통신 Code가 들어올 때마다 해당 Code를 Tree에서 찾아내어 Code를 인식하게 해준다.

③작업처리 준비 모듈: 계속해서 들어오는 작업 Code들을 받아들이고 Code list에 쌓아놓는다.

AS/RS 운용 프로그램내에서 독자적으로 사용하고 있는 기능모듈은 다음과 같다. 특히 이에 해당하는 기능모듈들은 공장 전체의 효율 뿐만 아니라 자동창고 자체의 효율에도 상당한 영향을 미치므로 중요하다. 이 중 JOB SCHEDULING 모듈에 관해서는 다음 절에서 좀 더 설명을 추가한다.

①초기 데이터 인출 기능 모듈: 공장 가동 시작시 DB에서 공장내 자동창고의 초기상황을 나타내는 정보를 인출해와서 상황판에 올린다

②Rack policy 모듈: Class based storage 방법 사용. 각 부품별로 따로 할당된 Rack들 가운데서 해당 작업에 가장 합당한 부품과 수량을 소유하고 있거나(출고, 출하) 또는 소유할 수 있는(입고, 입하)Rack number를 관련 함수들을 통해서 얻는다.

③JOB SCHEDULING 모듈: 공장 전체의 Throughput 향상을 위하여 중앙 컴퓨터와 빈번한 연락을 취하며, 현재 작업 list내의 작업들 중 우선적으로 처리해야 할 작업을 선정한다.

④AS/RS 상황판 update 기능 모듈: 작업이 완료될 때마다 변동되는 사항을 자동창고가 가지고 있는 내부 상황

판에서 변동시킨다.

⑤MHS과의 물품교환 기능모듈: MHS과 물품을 주고받을 때 작업의 성공적 수행 및 상대방의 이상상황 발생을 check.

⑥Priority Change 모듈: 변화된 공장상황에 의해 이미 주어진 Job의 등급을 바꿀 필요가 있을 때 Host로부터 오는 Priority change 명령내용을 해당 Priority list에 적용

⑦User Interface 모듈: 필요시 AS/RS 상황판이나 통계량을 User에게 display 해주고, 가지고 있는 상황판 정보의 Manual 수정이 필요할 때 이를 처리할 수 있게 한다.

4.3. 전체 시스템 성능향상을 위한 스케줄링 설계

기본적으로 AS/RS 작업 Scheduling의 대상은 입고, 출고, 입하 그리고 출하작업이고 이 기능의 수행은 MAIN 프로그램에서 담당한다. 또한 이 Scheduling 기능모듈은 들어온 작업들을 등급에 따라 Group을 지워주는 Sequencing 기능과 실제 Device가 일을 시작하려고 할 때 작업 List에서 수행할 하나의 작업을 선정하는 Dispatching 기능으로 나누어진다.

● Sequencing 설정

Host로부터 오는 여러 통신 Code들 중 스케줄링의 대상이 되는 Job Code가 들어오면 해당 Priority list에 넣고 (Sequencing) 따로 관리.

● Sequencing 규칙

4개로 나누어 관리하는 Priority list내 Job들의 처리순서는 EE, E, N, D의 순서를 따른다. 납기를 맞추기 위해 우선적으로 처리해야 할 EE나 E 작업들이 있는 List를 먼저 처리하고 D등급의 작업들은 해당 Cell의 상황이 호전

될 때 까지 일단 보류해 두었다가 처리하게 된다. 즉 상위 priority list가 empty가 아닌 이상 하위 Priority list내의 작업들은 dispatching의 대상이 될 수 없다. 또 일단 작업이 해당 Priority list내에 들어가면 그 List내의 작업들은 가지고 들어온 Due time에 의해서 Sorting이 실시된다. 즉 동일 Priority 그룹에 있는 작업들은 항상 Due time의 오름차순으로 정렬이 된 후에 Dispatching 시점을 기다리게 된다. 이런식으로 AS/RS의 작업 처리과정에 공장상황을 반영하는 작업등급을 고려함으로써 전체 공장효율에 도움이 되도록 하였다.

● Dispatching 설정

Stacker crane이 하나의 작업을 완료하고 Idle 상태일 때

● Dispatching 규칙 (Case별로 규칙 Base화)

Priority list내에 어떤 작업들이 있는가에 따라 15가지의 가능한 상황이 생기는데 이 상황 각각에서 해당 Dispatching 규칙을 적용하여 바로 처리할 작업 하나를 골라낸다. 이때 적용되는 Dispatching 규칙의 기본은 다음과 같다. Priority list내에 존재가능한 작업의 형태는 입고, 출고, 입하, 출하의

4가지인데 납기를 맞추기 위한 Due Time을 넘기지 않는 범위내에서 해당 작업을 바로 처리하지 않음으로써 생기는 공장내의 손실도를 고려 해 볼 때, 입고와 출고작업이 입하나 출하작업보다는 우선 순위를 갖도록 하였다. 또 입고와 출고작업들간에는 입고를, 입하와 출하작업들간에는 입하를 우선적으로처리하게 하였다.

이는 입고를 제 때에 해주지 않았을 때는 AGV 등 Material Handling System이 자동창고가 물품을 받아줄 때 까지 다른 일을 할 수 없고 마냥 기다리는 상황이 되지만, 출고시엔 중앙 컴퓨터의 중재로 MHS을 마냥 묶어두는 상황은 방지할 수 있기 때문이다.[2] 또 원자재가 없어서 공장내의 어느 작업장의 작업이 중지되는 상황이 Rack이 꽉 차서 더이상 물품을 받을 수 없는 상황 (이 경우 프로그램 내에서는 일종의 Buffer 역할을 하는 공용 Rack을 미리 지정해서 이 Rack을 사용하고 있다.) 보다 더 치명적이라고 판단하여 입하를 출하보다 먼저 시행하도록 한 것이다.

입고, 출고, 입하, 출하작업들 중 어느 한 가지의 작업들만이 해당 Priority list에 남아있을 때는 List내에 있는 순서대로 (즉 Due time이 빠른 것부터) 작업을 처리한다.

● Deadlock의 처리

Host로부터 받은 작업지시들은 일단 작업 List에 쌓아두었다가 공장내 각 CELL단위마다의 고유한 스케줄링에 의해서 작업순서가 정해지게 된다. 이렇듯 중앙 컴퓨터에 의한 중앙통제식이 아닌 분산통제방식의 공장에서는 Deadlock이 일어날 가능성이 많다. 이를 위해 본 시스템에 구축하고자 하는 AS/RS에는 Buffer의 기능을 이용하고 있다. 즉 출고시에 가져간 물품을 바로 I/O point에 놓지않고 일단 Buffer에 내려놓고 작업 List에 남아있는 다른 작업을 계속 수행하게 된다. 그 뒤 중앙 컴퓨터로부터 해당 물품을 MHS으로 넘기라는 별도의 지시가 있으면 Buffer에서 그 물품을 찾아 MHS으로 넘긴다.

5. 결론 및 추후 연구과제

이상의 연구에서 공장내의 여러 단위 CELL들이 통합되어져 돌아갈때 발생하는 제 문제를 다루었다. 작업장 운영 시스템에서는 Production 시스템을 응용하여 모듈화가 용이하게하고 공장 환경을 반영하여 전체 시스템의 운영 최적화를 기했다. 또한 스케줄링이나 고장 처리부분과 같이 최적해를 찾는것이 불가능하거나 일반적인 방법으로 정형화하기 어려운 부분은 전문가 시스템을 사용하여 실 시간으로 문제를 처리할 수 있도록 하였다.

그러나 고장원인 발견에 대한 규칙 베이스 시스템의 기능은 센서와같은 물리적인 장비의 성능제한을 받는다. 그리고 고장에 대한 처리 방안은 아직도 많은 연구가 필요하다. 또한 USER INTERFACE를 통하여 사용자가 운영 시스템의 수정을 보다 용이하게할 수 있도록 규칙의 형식과 제어 방식에 대하여 보다 깊은 연구가 이루어져야 한다.

AS/RS 운영 시스템에서는 Dispatching 루틴을 규칙 베이스화해서 처리하였지만 좀 더 많은 시간을 두고 성능을 측정하여 해당 공장 상황에 더욱 잘 맞고 실시간 통제에 더욱 근접하는 규칙으

로 개발시켜나가야 할 것이다.

참고문헌

- [1] 김 성식, 배 경한, "컴퓨터를 이용한 실제에 준하는 FMS 구축", IE Interface 산업공학, 4권, 1호, 대한산업공학회, 1991.
- [2] MIN HONG HAN, LEON F. MCGINNIS, JIN SHEN SHIEH, JOHN A. WHITE, "On Sequencing Retrieval In An Automated Storage/Retrieval System", IIE, pp56-66, 1987.
- [3] George F. Luger, William A. Stubblefield, "ARTIFICIAL INTELLIGENCE and the Design of Expert System", The Benjamin/Cummings Publishing Company, 1989.