

# 프랙탈 생산시스템의 동적 재구성 프로세스 알고리즘

문정태<sup>1\*</sup> · 차영필<sup>1</sup> · 신문수<sup>1</sup> · 정무영<sup>2</sup>

<sup>1</sup>포항공과대학교 산업공학과/<sup>2</sup>생산기술연구소

## Algorithm for the Dynamic Restructuring Process in the Fractal Manufacturing System

Jungtae Mun<sup>1</sup> · Youngpil Cha<sup>1</sup> · Moonsoo Shin<sup>1</sup> · Mooyoung Jung<sup>2</sup>

<sup>1</sup>Department of Industrial Engineering, Pohang University of Science and Technology, Pohang, 790-784

<sup>2</sup>Advanced Product and Production Technology Center, Pohang, 790-784

In order to quickly respond to the rapidly changing manufacturing environment, it is imperative for the system to have such capabilities as flexibility, adaptability, reusability, etc. One of the promising approaches for new manufacturing paradigm satisfying those capabilities is the fractal manufacturing system (FrMS). The FrMS can be optimized by reorganizing the structure of fractals through the dynamic restructuring process (DRP) to deal with the changes of environment. Through the DRP, the logical connections between fractals and roles of fractals are autonomously changed. To determine an appropriate fractal structure, certain rules are needed to build and evaluate alternative structures for fractals involved in the DRP. In this paper, an algorithm for the DRP-, which is invoked when some equipment are unavailable, is introduced and explained with examples.

**Keywords:** fractal manufacturing system, agent negotiation, distributed manufacturing systems, dynamic restructuring process

### 1. 서론

오늘날의 생산시스템은 빠르게 변화하는 환경과 소비자의 요구, 예측할 수 없는 상황, 소프트웨어와 하드웨어 기술의 급격한 발전, 인프라의 발전 등에 유연하게 대응할 수 있어야 한다. 이러한 변화에 빠르게 대응하기 위해서 생산시스템은 유연성(Flexibility), 적응성(Adaptability), 재사용성(Resuability) 등을 갖추는 동시에 고객의 요구에 부합하는 고부가/고품질 제품을 빠른 시간 안에 생산하여야 한다. 이를 위해서 생산시스템은 능동적이고 지능적이어야 하며, 시스템 자체의 구조가 내적으로 주어지거나 고정된 것이 아닌, 외적 변화와 주어진 생산시스템의 목적에 적합하게 스스로 변화할 수 있어야 한다(Kidd

1994). 하지만 기존의 계층적인 생산시스템 구조로는 이에 대한 대응이 불가능하기 때문에 생산시스템 자체가 유연한 자가 재구성 능력을 가질 필요성이 제기되었다(Davis *et al.*, 1990). 위의 요구사항에 부합하는 차세대 생산시스템에는 바이오닉 생산시스템(Bionic Manufacturing System; BMS) (Okino, 1993), 프랙탈 생산시스템(Fractal Manufacturing System; FrMS) (Warnecke, 1993; Ryu and Jung, 2003a; Ryu and Jung, 2003b; Shin *et al.*, 2003; Ryu and Jung 2004; Mun *et al.*, 2004), 홀로닉 생산시스템(Holonc Manufacturing System; HMS) (Brussel *et al.*, 1998; Valckenaers *et al.*, 1994)이 있으며, BMS, FrMS, HMS는 모두 자율적(Autonomous)이고, 상호 협력적(Cooperative)이며, 지능적(Intelligent)인 특성을 가진 생산시스템이다.

이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2003-041-D00622).

\*연락처 : 문정태, 790-784 경북 포항시 남구 효자동 산 31 포항공과대학교 산업공학과, Fax : 054-279-5998,

E-mail : softmun@postech.ac.kr

FrMS는 프랙탈(Fractal)이라 부르며 자율적으로 협동하고 자가재구성 능력(Self-reconfigurability)을 가진 에이전트들로 구성된다. FrMS는 새로운 기기의 추가나 제거, 기기의 고장, 생산 제품의 변화 등에 동적 재구성 프로세스(Dynamic Restructuring Process; DRP)를 통해 작업자의 개입 없이 빠르게 대처할 수 있다.

본 논문은 기기의 고장 등으로 인해 DRP가 실행되었을 때 프랙탈 구조를 변경하는 알고리즘을 다룬다. 본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 연구 대상인 FrMS를 소개한다. 3장에서는 DRP에 대하여 간략히 설명한다. 4장에서는 프랙탈들의 협상을 통한 구조 변경 알고리즘과 알고리즘을 실행하는 예를 설명한다. 마지막으로 5장에서는 결론을 맺고 추후 연구 과제를 제시한다.

## 2. 프랙탈 생산시스템

FrMS는 계층적 제어구조(Hierarchical Control Architecture)와 분산형 제어구조(Distributed Control Architecture)의 복합 형태인 혼합형 제어구조(Hybrid Control Architecture)를 따르고 있다. <그림 1>은 FrMS의 구조를 보여주는 것으로 FrMS는 프랙탈들의 집합으로 생각할 수 있다. 각각의 프랙탈은 자가유사성(Self-similarity)과 자가재구성 능력을 갖춘 에이전트들의 집합이며, 이 에이전트들은 주어진 골(Goal)을 달성하기 위해 작업을 수행한다. <그림 2>에서와 같이 프랙탈은 기본적으로 다섯 개의 기능 모듈(Observer, Analyzer, Resolver, Organizer, Reporter)로 구성되어 있다.

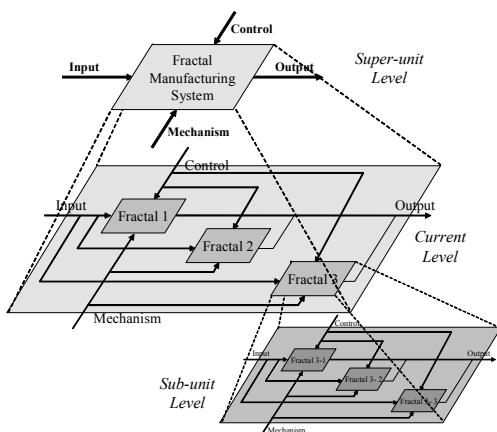


그림 1. FrMS의 구조.

<표 1>은 프랙탈에 존재하는 에이전트들과 그것들이 속한 기능 모듈을 정리한 것이다. Observer와 Reporter는 프랙탈의 입력과 출력을 담당하는 모듈이다. 프랙탈이 받아들이는 정보는 크게 네트워크를 통한 정보와 기기의 센서 정보로 분류할 수 있다.

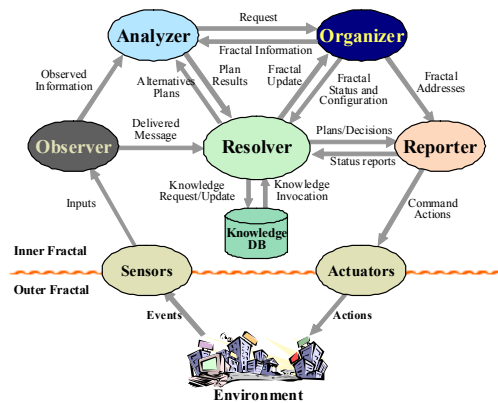


그림 2. FrMS의 기능 모듈 간의 관계.

Observer는 프랙탈이 필요한 외부 정보를 받아들이는 에이전트(Equipment Monitoring Agent, Network Monitoring Agent)를 가지고 있으며, 이것들은 다른 프랙탈이나 여러 센서들로부터 정보를 받아서 프랙탈 내의 다른 모듈에게 전달하는 역할을 수행한다. Reporter는 Equipment Control Agent와 Network Control Agent를 가지고 있는데, 이것은 다른 모듈로부터 여러 메시지를 받아 다른 프랙탈이나 센서들에게 전달하는 역할을 수행한다. Analyzer는 스케줄링과 시뮬레이션 등 여러 분석에 관련된 에이전트를 가지고 있다. Resolver는 골 생성이나 다른 프랙탈과의 협상, 의사 결정 등 프랙탈이 당면한 문제들을 해결하기 위한 에이전트들을 가지고 있다. Organizer는 프랙탈의 상태 체크 및 구조 변경에 관련된 에이전트들을 가지고 있다. <표 1>에서 에이전트들의 이름 뒤에 S가 붙은 것은 그 에이전트가 정적 에이전트(Stationary Agent)라는 것을 나타내고 M은 동적 에이전트(Mobile Agent)라는 것을 나타낸다.

표 1. 프랙탈의 기능 모듈과 에이전트

기능 모듈	소속 에이전트
Observer	- Network Monitoring Agent(S) - Equipment Monitoring Agent(S)
Reporter	- Network Command Agent(S) - Equipment Command Agent(S)
Analyzer	- Schedule Evaluation Agent(S) - Dispatching-rule Rating Agent(S) - Real-time Simulation Agent(S)
Resolver	- Schedule Generation Agent(M) - Goal-Formation Agent(S) - Task Governing Agent(S) - Negotiation Agent(M) - Knowledge Database Agent(M) - Decision-Making Agent(S)
Organizer	- Fractal Status Manager(S) - Fractal Address Manager(S) - Restructuring Agent(M)
Miscellaneous	- System Agent(S) - Network Agent(S)

프랙탈은 FrMS에서의 레벨에 따라 주어진 역할이 다르며, 역할에 따라 수행하는 일도 다르다. <표 2>는 프랙탈의 레벨에 따른 세 가지 역할을 정리한 것이다. 매니저부터 컨트롤러에 이르기까지 각 프랙탈은 주어진 역할을 수행하기 위한 내부 모듈의 차이는 있어도 프랙탈의 자가유사성은 보존된다.

표 2. 프랙탈의 세 가지 역할

역 할	설 명
매니저	최상위 프랙탈. 사람이나 다른 시스템으로부터 생산시스템을 운영하기 위한 프로세스 정보, 골 등을 받는다.
중계자	매니저와 컨트롤러 사이에 존재하는 프랙탈. 상위 레벨의 프랙탈로부터 수행해야 할 작업과 골을 받아 하위 프랙탈들에게 분산시킨다. 하위 프랙탈들의 협상을 도와준다.
컨트롤러	최하위 프랙탈. 기기에 연결되어 기기를 제어하는 역할을 수행한다.

### 3. 동적 재구성 프로세스

DRP는 FrMS의 내/외적 환경 변화에 빠르게 대응하여 프랙탈들의 구조를 재구성하는 프로세스이다. <그림 3>은 DRP의 예를 보여주는 것으로 왼쪽과 같은 프랙탈 구조에서 장비들이 추가되었을 경우 FrMS는 DRP를 통해 사람의 개입 없이 최적의 구조를 재구성한다. DRP에는 다음과 같이  $DRP^-$ ,  $DRP^+$ ,  $DRP^*$ 의 세 가지 종류가 있다(Ryu and Jung, 2003b).

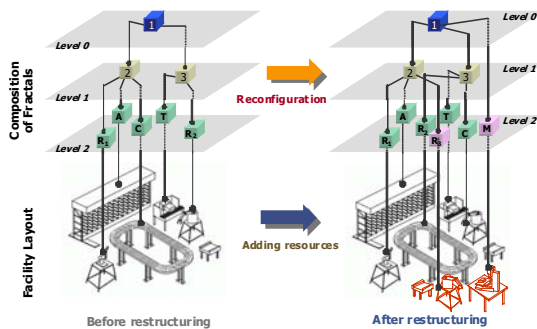


그림 3. DRP의 예.

- $DRP^-$ 는 기존 장비의 고장이나 주기적인 점검/수리 등으로 시스템에서 장비가 제거되는 경우에 실행된다. 기존 장비에서 수행되던 일을 시스템에 분산시키기 위해 프랙탈들의 구조와 역할의 변경 및 생산 계획의 재수립 등이 DRP-를 통해 이루어지게 된다.
- $DRP^+$ 는 새로운 장비나 기존에 제거되었던 장비가 다시 시스템에 추가되는 경우에 실행된다. 장비가 추가되면 해당 장비를 제어하기 위한 프랙탈이 생성되고 필요한 모듈의 설

치가 수행된다.

- $DRP^*$ 는 장비의 추가나 제거 없이 프랙탈들의 논리적인 연결을 변경하는 것으로 어느 한 프랙탈에 부하가 집중되거나 FrMS의 골에 변화가 있는 경우에 수행된다. 프랙탈은 지속적으로 시스템의 상태를 감시하여 어느 한쪽에 부하가 많이 집중되는 것을 방지하고 전체 시스템의 최적화를 위해  $DRP^*$ 를 실행한다.

DRP는 관련 프랙탈들의 Restructuring Agent와 Negotiation Agent의 도움을 받아 프랙탈들의 논리적 구조를 변경한 뒤 Knowledge Database Agent와 System Agent에 의해 각 프랙탈의 내부 구성을 변경한다(Ryu and Jung, 2003b; Mun et al., 2004). 프랙탈들의 논리적 구조가 변경되면, 프랙탈들은 자신에게 주어진 역할을 수행할 수 있도록 내부 모듈을 설치/수정하는 과정을 진행한다.

### 4. DRP 알고리즘

프랙탈 생산시스템은 계층적 제어구조를 가짐과 동시에 분산형 제어구조를 따르고 있기 때문에 최적의 프랙탈 구조를 찾는 것이 힘들다. 따라서 각 프랙탈들이 DRP에서 사용할 알고리즘과 협상에 이용할 기준이 필요하다.

#### 4.1 시나리오

본 논문에서 제시하는 시나리오는  $DRP^+$ 에 대한 것이다. 시스템 내의 기기 고장 시 FrMS는  $DRP^+$ 를 통해 새로운 프랙탈 구조를 구성하여 주어진 목표를 달성하고자 하는데, 이것은 프랙탈들 간의 협상을 통해 이루어진다. <그림 4>는 기기 고장 시  $DRP^+$ 에서 프랙탈 구조를 변경하는 세 가지 경우에 대한 예이다. FrMS가 <그림 4>의 (a)와 같은 프랙탈 구조를 가지고 있고, 중계자 프랙탈  $fractal_2$ 는 제품 A를, 중계자 프랙탈  $fractal_3$ 은 제품 B의 생산을 담당하고 있을 때 컨트롤러 프랙탈  $fractal_5$ 에 연결된 기기가 고장이 난 경우 진행되는 과정은 다음과 같다. 우선 프랙탈들은 세 가지 대안을 검토하여 새로운 구조를 결정하게 된다. 첫 번째는 프랙탈 구조의 변화가 일어나지 않는 경우이다. <그림 5>의 (b)에 있는 것처럼  $fractal_2$ 와  $fractal_3$ 의 하위 구조는 변경되지 않는다. 이 때,  $fractal_2$ 는 관리자에게 기기의 고장을 알리고 제품 A를 계속 생산할 수 있는 경우에는 스케줄링을 다시 하여 생산을 진행한다.  $fractal_5$ 에서 수행하던 작업을 수행할 수 있는 다른 기기가  $fractal_2$ 의 하위에 존재하지 않는 경우 제품 A를 생산할 수 없으므로  $fractal_2$ 의 하위 프랙탈들은 자신에게 주어진 일을 수행하며 기기가 수리되기를 기다린다. 두 번째는 기능을 수행할 수 없는  $fractal_3$ 를 대신하여 그와 같은 기능을 수행할 수 있는 다른 컨트롤러 프랙탈을 다른 중계자 프랙탈로부터 받아오는 경우이다. <그림 4>의 (c)에 있는 것처

럼  $fractal_2$ 는  $fractal_3$ 의 하위 프랙탈 중에서  $fractal_5$ 의 기능을 수행할 수 있는  $fractal_6$ 을 받아 생산을 계속 한다. 세 번째 경우는 서로 프랙탈들을 교환하는 경우이다. <그림 4>의 (d)는 복수 개 프랙탈들의 교환이 일어난 경우를 보여준다. 이것은  $fractal_5$ 의 부재로 인해 야기되는 생산 차질을 전체 시스템에 분산시켜 최적의 생산을 하고자 하는 경우이다. <그림 4>의 (d)에서처럼  $fractal_2$ 는 자신의  $fractal_4$ 를  $fractal_3$ 에게 주고  $fractal_3$ 으로부터  $fractal_6$ 과  $fractal_7$ 을 받아 자신에게 주어진 일을 수행한다. 이 과정은 DRP\*이며 본 논문에서는 다루지 않는다.

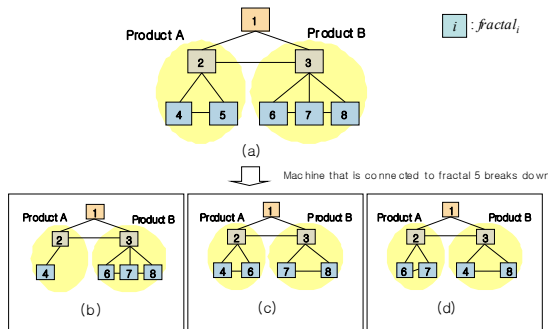


그림 4. 새로운 프랙탈 구조를 위한 세 가지 대안.

4.2 알고리즘

새로운 프랙탈 구조를 만들기 위한 알고리즘은 다음의 세 프랙탈에 의해 수행된다.

- $fractal_R$ : 제거된 기기와 연결된 컨트롤러 프랙탈의 중계자 프랙탈. 생산 계획의 차질을 최소화하기 위해 다른 중계자 프랙탈에게 기기 지원을 요청하는 프랙탈
- $fractal_P$ :  $fractal_R$ 로부터 기기 지원 요청을 받아 지원 가능 여부를 판단한 후 기기를 지원하는 중계자 프랙탈
- $fractal_M$ :  $fractal_P$ 와  $fractal_R$ 의 상위 프랙탈로 두 프랙탈의 협상 결과에 따른 프랙탈 구조 정보를 갱신하는 프랙탈

알고리즘은 4 단계로 진행이 된다. 단계 1은  $fractal_R$ 에 의해 자신이 수행할 수 없는 작업을 찾는 과정이다. 단계 2는  $fractal_R$ 의 요청을 받은  $fractal_P$ 에 의해 수행되며  $fractal_P$ 가 지원할 수 있는 기기를 찾는 과정이다. 단계 3은  $fractal_P$ 와  $fractal_R$ 에서 수행되며 서로에게 최선의 대안을 선택하는 과정이다. 단계 4는  $fractal_M$ 에서 수행되며 새로운 프랙탈 구조가 정해지게 된다. 알고리즘에 쓰이는 기호들은 <표 3>과 같다.

표 3. 알고리즘에 사용되는 기호

기호	설 명
$n$	number of machines involved in a fractal
$m$	number of operations executed by a fractal
$i$	index of a machine

기호	설 명
$j$	index of an operation
$k$	index of break-down machine
$A$	machine-operation matrix, where $a_{ij} = 1$ , if operation $j$ is executed by machine $i$ , otherwise $a_{ij} = 0$
$O$	operation set of break-down machine
$T$	impracticable operation set
$F$	feasible machine set

제1단계: 수행 불가능한 작업 찾기

$fractal_R$ 에서 실행되는 것으로, 고장난 기기에서 수행하던 작업 중에서 자신의 하위 프랙탈들이 수행할 수 없는 작업을 찾는다.

Find-Impractical-Operation-Set( $n, m, k, A, T$ )

1.  $O \leftarrow \text{null}$
2.  $T \leftarrow \text{null}$
3. for  $i \leftarrow 1$  to  $m$
4. if  $a_{ki} = 1$  then  $O \leftarrow O + \{i\}$
5. for  $i \leftarrow 1$  to length( $O$ )
6. if Find-Machine( $n, O, k$ )= $false$
7. then  $T \leftarrow T + O$

함수 Find-Machine( $n, O, k$ )은  $O$ 를 수행하는 다른 기기가 있는 경우에는  $true$ 를, 그 밖의 경우에는  $false$ 를 리턴한다. 최종적으로  $fractal_R$ 에서 수행할 수 없는 작업 집합  $T$ 가 얻어지며  $fractal_R$ 은 구해진  $T$ 를  $fractal_P$ 에게 전달한다.

제2단계: 지원 가능 기기 찾기

$fractal_P$ 에서 실행하는 것으로,  $fractal_R$ 의 요청에 의해 자신이 가지고 있는 기기들 중  $fractal_R$ 에게 줄 수 있는 기기를 찾는다. 우선  $fractal_R$ 에게 받은 수행 불가능한 작업 집합  $T$ 에 있는 작업들을 수행할 수 있는 자신의 기기들을 모두 찾는다. 찾은 기기들 중 자신에게서 제외해도 생산이 중단되지 않는 기기들의 집합이 지원 가능 기기 집합  $F$ 이다.

Find-Feasible-Machine-Set( $T, n, m, A, F$ )

1.  $F \leftarrow \text{null}$
2. for  $i \leftarrow 1$  to  $n$
3. for  $j \leftarrow 1$  to length( $T$ )
4. if  $a[T_j, i] = 1$
5. then  $V \leftarrow \text{null}$
6. Find-Impractical-Operation-Set( $n, m, i, A, V$ )
7. if length( $V$ ) = 0
8. then  $F \leftarrow F + \{i\}$

$fractal_P$ 는 위에서 구해진 집합  $F$ 와  $F$ 에 속한 기기 정보를  $fractal_R$ 에게 전달한다.

**제3단계: 대안 생성**

이 과정은  $fractal_P$ 와  $fractal_R$  모두에서 수행된다. 단계 2에서 구해진  $F$ 와  $F$ 에 속한 기기 정보를 바탕으로  $fractal_P$ 에서는  $F$ 에 속한 기기들 하나하나를 자신의  $A$ 에서 제외하였을 때의 생산성을 예측하고,  $fractal_R$ 에서는  $F$ 에 속한 기기들 하나하나를 자신의  $A$ 에 추가하였을 경우의 생산성을 예측한다. Koren et al. (1998)의 연구 결과를 이용하여 생산성 예측치  $E[P]$ 는 다음의 식에 의해 구해진다.

$$E[P] = \sum_{\lambda \in \Lambda} Pr(\lambda) V(\lambda)$$

여기서  $\Lambda$ 는 발생 가능한 기기 고장인 경우의 집합이다.  $n$ 개의 기기가 있을 경우, 경우의 수는  $2^n$ 이 된다.  $Pr(\lambda)$ 은 해당 경우가 발생할 확률이며,  $V(\lambda)$ 는 해당 경우의 생산성이다.  $Pr(\lambda)$ 은 기기들이 특정 기간 동안 얼마나 안정적으로 가동될 수 있는지를 나타내는 신뢰도를 바탕으로 계산될 수 있다.  $\lambda$ 의 고장난 기기의 집합을  $F_\lambda$ 라 하고, 이 때  $\lambda$ 에 속한 기기  $i$ 의 신뢰도를  $R_i$ 라 하면,  $Pr(\lambda)$ 은 다음과 같이 구해질 수 있다.

$$Pr(\lambda) = \left[ \prod_{i \in F_i} R_i \right] \cdot \left[ \prod_{i \in F_i} (1 - R_i) \right]$$

위에서의 신뢰도  $R_i$ 는 기기의 고장 확률  $FR_i$ 를 이용하여 구할 수 있고, 이 때 위의  $Pr(\lambda)$ 는 다음의 식과 같이 된다.

$$Pr(\lambda) = \left[ \prod_{i \in F_i} (1 - FR_i) \right] \cdot \left[ \prod_{i \in F_i} FR_i \right]$$

위의 식을 바탕으로  $fractal_P$ 와  $fractal_R$ 은 지원 가능 기기 집합  $F$ 에 있는 기기들을 자신에게 추가 혹은 제거하였을 경우의 생산성 예측치를 구할 수 있다.  $fractal_P$ 와  $fractal_R$ 은 각각 자신의 생산성 감소와 증가 정도를 바탕으로 협상을 하여  $\text{Max}[\text{증가치} - \text{감소치}]$ 의 값이 가장 큰 대안을 선택한다.  $fractal_P$ 는 이 결과를  $fractal_M$ 에게 전달한다.

**제4단계: 새로운 구조 결정**

$fractal_P$ 와  $fractal_R$ 의 상위 프랙탈인  $fractal_M$ 은  $fractal_P$ 와  $fractal_R$ 이 보낸 결과를 바탕으로 프랙탈 구조 정보를 갱신하며  $fractal_P$ 와  $fractal_R$ 에게 갱신된 정보를 전달한다.

**4.3 예 제**

다음은 위 알고리즘의 적용 예이다. <그림 4>의 프랙탈 구조를 확장시켜서  $fractal_2$ 에는 5개의 컨트롤러 프랙탈이 존재하고,  $fractal_3$ 에는 6개의 컨트롤러 프랙탈이 존재하는 경우를 가

정한다.  $fractal_2$ 에 있는  $M_3$ 이 고장난 경우  $fractal_2$ 가  $fractal_R$ 이 되며,  $fractal_3$ 은  $fractal_P, fractal_1$ 이  $fractal_M$ 이 된다.

- **제1단계:** <그림 5>는 Machine-Operation Matrix (MOM)를 나타낸 것이다.  $fractal_2$ 의 MOM이 <그림 5>의 (a)와 같을 때  $M_3$ 가 수행하는 작업은  $\{O_3, O_4\}$ 이다. 이 중에서 프랙탈2의 하위 프랙탈들이 수행할 수 없는 작업을 찾으면 이 집합은  $\{O_4\}$ 이다.  $fractal_2$ 는 이 수행 불가능 작업 집합을  $fractal_3$ 에게 전달한다.
- **제2단계:**  $fractal_2$ 에게 수행 불가능 작업 집합을 받은  $fractal_3$ 은 해당 작업을 수행할 수 있는 기기들을 찾는다.  $fractal_3$ 의 MOM이 <그림 5>의 (b)와 같다면  $O_4$ 를 수행할 수 있는 기기들은  $M_1, M_3, M_4$ 이다.  $M_1$ 이 수행하는 작업들을 MOM에 있는 다른 기기들이 수행할 수 있는지 검토해 보면  $O_4$ 를 수행하는 기기는 존재하지만  $O_5$ 를 수행하는 기기는 존재하지 않는다. 따라서  $M_1$ 은 불가능하다고 판단되어 지원 가능 기기에서 제외된다.  $M_3$ 와  $M_4$ 가 수행하는 작업들은 다른 기기들이 수행할 수 있으므로 지원 가능 기기 집합은  $\{M_3, M_4\}$ 로 결정된다.  $fractal_3$ 은 지원 가능 기기 집합과 이 집합에 포함된 기기 정보를  $fractal_2$ 에 전달한다.
- **제3단계:**  $fractal_3$ 은 지원 가능 기기 집합에 있는  $M_3$ 을 제외한 다른 기기들로 생산을 진행할 경우의 생산성과  $M_4$ 를 제외한 다른 기기들로 생산을 진행할 경우의 생산성을 예측한다. 각 경우에 대한 생산성 감소치를 바탕으로  $fractal_3$ 은 비드(Bid)를 생성한다.  $fractal_2$ 는 지원 가능 기기 집합에 있는 각 기기를 자신이 받았을 때 생산성 증가치를 예측하고, 이를 바탕으로 비드를 생성한다. 그 후 총 생산성 예측치의 감소 폭이 가장 작은 대안을 선택한다.
- **제4단계:**  $fractal_2$ 는 이 결과를  $fractal_1$ 에게 전달한다.  $fractal_1$ 은 선택된 대안을 바탕으로 프랙탈 구조 정보를 갱신한다.

	M1	M2	M3	M4	M5
O1	1	1			1
O2		1		1	
O3			1	1	1
O4				1	
O5	1				1

(a) Fractal 2

	M1	M2	M3	M4	M5	M6
O1		1			1	1
O2		1		1		
O3			1		1	1
O4	1		1	1		
O5	1					

(b) Fractal 3

그림 5. Machine-Operation Matrix(MOM).

**4.4 소프트웨어 아키텍처**

본 논문에서 제안한 알고리즘을 위해 구성된 소프트웨어 아키텍처는 <그림 6>과 같다. FrMS를 구성하고 있는 에이전트들은 Java™로 구현되며 모바일 에이전트의 구현에 Java Agent Development Environment (JADE) Platform을 사용한다. 각 기기들의 컨트롤 모듈은 Visual C++를 사용한 동적 연결 라이브러

리(Dynamic Linking Library; DLL)로 추상화되고 모듈화된다. Java™에서의 C++ 기반 라이브러리들의 사용을 위해 Java Native Interface(JNI)를 이용한다.

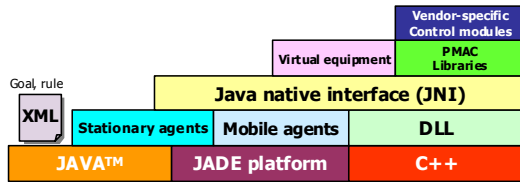


그림 6. FrMS의 소프트웨어 아키텍처.

### 5. 결론 및 추후 연구 과제

FrMS는 자율적이고 상호 협력적이며 지능적인 생산시스템이다. FrMS는 DRP를 통해 다양한 환경 변화에 대처할 수 있다. DRP는 계층적 제어구조를 변경하는 것으로 프랙탈들의 논리적 연결을 변경함으로써 이루어진다. DRP는 프랙탈 구조의 확장 유무에 따라 세 가지 종류(DRP\*, DRP+, DRP-)로 분류된다. 본 논문에서는 그 중 DRP\*가 수행될 경우 프랙탈들의 구조를 재구성하는 알고리즘에 대해 기술하였다. 프랙탈의 구조를 재구성하는 방식은 구조의 변화에 따라 세 가지가 있으며, 본 논문에서 제안한 알고리즘은 생산성 예측치를 바탕으로 프랙탈들 간의 협상을 통해 대안을 선택한다.

추후 연구 과제로는 DRP\*와 DRP+에 대한 알고리즘 연구가 있으며, 프랙탈의 구조를 평가할 때 사용할 생산성 예측치 이외의 다양한 평가 방법을 개발하여야 할 것이다.

### 참고문헌

Brussel, H.V., Wyns, J., Valckenaers, P., Bongaerts, L., and Peeters, P. (1998), Reference Architecture for Holonic Manufacturing Systems: PROSA. Computers in Industry, 37, 255-274.

Davis, W. J., Thompson, S. D. and White, L. R. (1990), The importance of decompositions in CIM control architectures, Proc. CIMCON90 Conference, Washington, DC.

Kidd, P. T. (1994), Agile Manufacturing: Forging new frontiers, Addison-Wesley.

Koren, Y., Hu, S., and Weber, T. (1998), Impact of manufacturing system configuration on performance, Annual CIRP, 47(1), 369-372.

Mun, J., Ryu, K., and Jung, M. (2004), Self-reconfigurable Software Architecture: Design and Implementation, Proceedings of The 33rd International Conference on Computers and Industrial Engineering, Jeju, Korea.

Okino, N. (1993), Bionic manufacturing systems, In J. Peklenik (ed.) CIRP, Flexible Manufacturing Systems Past-Present-Future, 73-95.

Ryu, K., and Jung, M. (2003a), Agent-based fractal architecture and modeling for developing distributed manufacturing systems. International Journal of Production Research, 41(17), 4233-4255.

Ryu, K., and Jung, M. (2003b), Dynamic Restructuring Process in the Fractal Manufacturing System, Proceedings of the 2003 KIIE Fall Conference, Pyoung Chang, Korea, 54-58.

Ryu, K., and Jung, M. (2004), Goal-orientation mechanism in the fractal manufacturing system, International Journal of Production Research, 42(11), 2207-2225.

Shin, M., Cha, Y., Ryu, K., and Jung, M. (2003), In: Proceedings of the 8th Annual International Conference on Industrial Engineering, Las Vegas, 1034-1039.

Valckenaers, P., Bonneville, F., Brussel, H. V., Bongaerts, L., and Wyns, J. (1994), Results of the Holonic Control System Benchmark at the K.U. Leuven. Proceedings of the Computer Integrated Manufacturing and Automation Conference, Rensselaer Polytechnic Institute, 128-133.

Warnecke, H. J. (1993), The Fractal Company: A Revolution in Corporate Culture, Springer-Verlag.



**문정태**

포항공과대학교 산업공학 학사  
 현재: 포항공과대학교 산업공학과 석박사  
 통합과정  
 관심분야: 지능형 생산시스템, 생산계획



**신문수**

포항공과대학교 산업공학 학사  
 포항공과대학교 산업공학 석사  
 현재: 포항공과대학교 산업공학과 박사과정  
 관심분야: 지능형 생산시스템, 시뮬레이션



**차영필**

포항공과대학교 산업공학 학사  
 포항공과대학교 산업공학 석사  
 현재: 포항공과대학교 산업공학과 박사과정  
 관심분야: 멀티 에이전트 시스템, 분산 제어,  
 지능형 생산시스템



**정무영**

서울대학교 항공공학 학사  
 캔사스 주립대 산업공학 석사  
 캔사스 주립대 산업공학 박사  
 현재: 포항공과대학교 산업공학과 교수  
 관심분야: 에이전트 기술, 웹서비스, 지능형  
 생산시스템, 바이오 인포메틱스 등