

조립시스템 Configurator 에 관한 연구

김동주*(성균관대 대학원), 강무진(성균관대 기계공학부),
장인성, 김상명(현대자동차), 김기태(화천기공)

Design of a configurator for assembly system

D. J. Kim* (Graduate School, SKKU), M. Kang (Mechanical Eng. Dept. SKKU),
I. S. Jang, S. M. Kim(HMC), and K. T. Kim(Hwacheon Machinery Co.)

ABSTRACT

To cope with the challenge of competitive market, manufacturing system needs to be agile in terms of its reconfigurability and scalability. For a system to be adapted to changed requirements, decision support tools such as configurator have to be provided. This paper introduces the basic framework of a configurator for assembly system. Based on the factory data model(FDM) depicting the overall structure of a manufacturing system, functions of the configurator are described, i.e., requirements analysis, module selection and configuration optimization.

Key Words : Agile manufacturing system(민첩대응 생산시스템), Configurator(컨피규레이터), Factory Data Model(공장 데이터 모델), Reconfiguration(재구성), Axiomatic design(공리적 설계).

1. 서론

기업 환경이 생산자에서 고객으로 그 중심이 이동되고 있다. 이를 반영하듯이 변화하고 있는 시장은 동적이고 예측이 어려우며 고객 지향의 방향으로 나아가고 있다. 이런 시장의 변화에 대해서 기존의 생산시스템은 긴 설계 및 ramp-up 기간, 낮은 적응성 등을 가지는 한계를 드러내고 있다. 이로 인해 고객이 요구하는 고품질의 제품을 제때에 생산하지 못하여 결국 고객의 불만을 야기시킨다.

이러한 시장 변화에 대처하기 위해서 생산시스템은 mass production 에서 lean production 으로 변화해 왔으나, 현재 이후의 고객 요구와 시장 환경은 더욱 변화가 일정하지 않으며 예측 또한 어려워지고 있다. 이런 상황에 대응하기 위해서 생산시스템 자체를 reconfiguration 하는 민첩대응 생산시스템(agile manufacturing system)이 대두되었고 핵심 사항으로서 긴 전주기 지향성(long-term life-cycle orientation)과 민첩성(agility)을 내포하고 있다.[1]

민첩대응 생산시스템을 구현하기 위해서 시장 분석, 생산 시나리오 계획, 생산시스템 구성 및 재구

성, 생산시스템의 성능 및 전주기 평가의 절차를 수행한다. 특히 생산시스템의 구성 및 재구성 단계를 수행하는 툴을 Configurator 라고 명명한다. Configurator 는 공장내의 하부영역을 다루는 In-plant 영역과 공장간 및 공급망을 다루는 Cross-plant 영역으로 나눈다.

본고에서는 In-plant 영역에서 Configurator 설계를 위한 접근방법을 소개하고 Configurator 의 데이터 모델과 framework 을 제시하고자 한다.

2. Configurator 설계를 위한 접근방법

2.1 Building blocks & Modular concept

제품 설계자는 그 요구를 만족시키는 primitive function 들의 조합을 구상하고 모듈을 선택하고 CAD 시스템을 사용하여 모듈을 연결 및 결합하여 제품을 설계한다. 또한 제품을 모듈로서 분해함으로써 조립 순서를 만들어 낼 수 있다. 이와 같이 제품 설계시 모듈 개념을 사용한 것과 같이 생산시스템의 요소 자체를 모듈로 표현하고 이를 조합 및 결합함으로써 전체 생산시스템을 설계하고자 한다.

이 모듈 내지 building block 개념은 민첩 대응 생산시스템의 특징인 reconfigurability, reusability, scalability 을 충족시킬 수 있다.[2]

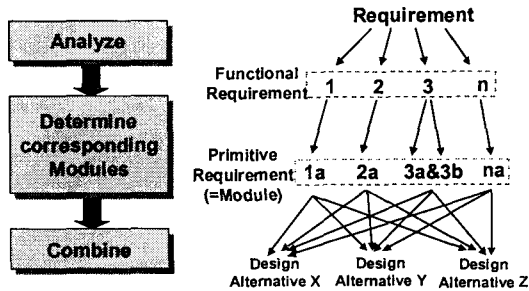


Fig. 1 Agile manufacturing system design process based on modular concept

Fig. 1 에서 보듯이 설계의 첫 단계로 추상화된 요구 사항은 설계자에 의해 여러 개의 기능들로 분석된다. 이 분해된 기능들은 대응하는 모듈내지 모듈의 조합으로 구현된다. 생성된 모듈들을 선택하고 결합함으로써 생산시스템 여러 설계안을 만들어 낼 수 있다.

2.2 Object-oriented approach

객체 지향 모델링 기술은 실세계와 그것을 표현하는 모델에서의 객체사이의 일대일 대응을 제공한다. 객체 지향의 기본 개념은 추상화(abstraction), 캡슐화(encapsulation), 상속(inheritance), 상태(state)가 있고 이러한 객체 지향 기술은 높은 모듈성(modularity), 유지보수성(maintainability), 재사용성(reusability)을 포함하고 있다.

이런 장점은 많은 연구자들로 하여금 생산시스템 모델링에 객체 지향 모델링을 적용하도록 이끌고 있다. 그러나 실세계를 표현하는데 이 모델링 방법은 아직 추상화 및 정확도가 부족하다. 따라서 객체 지향 모델링 방법과 추상화 및 분해 개념을 포함한 방법론을 통합하여 사용하는 hybrid method 를 사용하고자 한다.

2.3 Axiomatic design

설계에서의 공리적 접근방법은 좋은 설계를 판단하는데 절대적인 기준을 제시하여 주는 것을 의미한다. 공리적 설계는 설계에 대한 지식을 체계화, 논리화하여 정리할 수 있으므로 전임자의 지식을 후임자에게 용이하게 전달할 수 있다.

공리적 설계는 4 개의 영역, 즉, 고객의 속성(customer Attributes : CAs)을 나타내는 고객 영역, 고객의 요구를 반영할 수 있는 요구기능(Functional Requirements : FRs)으로 구성되는 기능 영역, 이

FRs 를 구현하기 위한 설계 변수(Design Parameters : DPs)로 구성되는 물리 영역, DPs 를 구현하기 위한 가공 변수(Process Variables : PVs)로 나타내는 가공 영역으로 구성된다. 분해과정은 기능 영역과 물리 영역 사이에 발생하는 연속적인 매핑으로 진행된다.[3]

좋은 설계를 얻기 위하여 기본적으로 항상 유지되어야 하는 것으로 독립 공리(independence axiom)와 정보 공리(information axiom)가 있다. 독립 공리는 설계의 요구 기능간에 독립성을 유지하라는 것이다. 정보 공리는 독립 공리를 만족하는 설계들 중에서 가장 최소한의 정보량(information contents)을 갖는 설계가 상대적으로 가장 좋은 설계라는 것이다.

3. Configurator 의 Factory Data Model(FDM)

Fig. 2 에 본 Configurator 의 Factory Data Model 이 제시되어 있다. 이 FDM 은 J. Zhao et al 가 제시한 FDM 을 확장하여 UML(Unified Modeling Language)의 class diagram 으로 표현하고 있다.[4] 이 class diagram 은 FDM 의 주요 클래스와 그들의 관계로 구성되어 있고 개념적인 생산시스템 모델의 구현을 가능하게 한다.

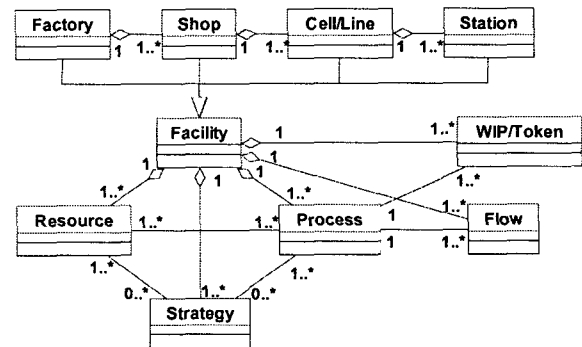


Fig. 2 Factory Data Model of In-plant Configurator using UML class diagram

Fig. 2 에 나타난 것과 같이 Facility 는 가장 낮은 레벨에서는 개개의 기계(Station), 더 높은 레벨에서는 Cell/Line 이나 Shop, 최상위 레벨에서는 Factory 로 간주되어진다. 즉, 이 Facility 클래스는 Factory, Shop, Cell/Line, Station 의 부모클래스이다. Factory, Shop, Cell/Line, Station 사이의 집합관계는 다음과 같다. 하나의 Factory 객체는 하나 혹은 여러 개의 Shop 객체로 구성되고 마찬가지로 하나의 Shop 객체는 하나 혹은 여러 개의 Cell/Line 객체로 구성된다. Cell/Line 객체도 동일하게 여러 개의 Station 으

로 구성되며 전체적으로 하나의 Factory 객체로부터 객체의 트리구조를 이루게 된다.

이 구조의 어느 레벨에서도 부모클래스인 Facility 객체는 하나 혹은 여러 개의 Resource 객체, Process 객체, Strategy 객체, Flow 객체, WIP/Token 객체로 구성되어 있다. 즉 Resource, Process, Strategy, Flow, WIP/Token 클래스는 Facility 클래스의 컴포넌트이다. 이 컴포넌트 클래스중 Resource, Process, Strategy는 다대다 관계, Process와 Flow, WIP/Token은 일대다 관계가 유지되고 있다. 이것은 특정 Resource 객체가 특정 Process 객체를 수행하고 특정 Strategy 객체가 위의 특정 Resource와 Process를 구속한다. 또한 이 특정 Process는 Flow와 WIP/Token 객체를 결정하게 된다.

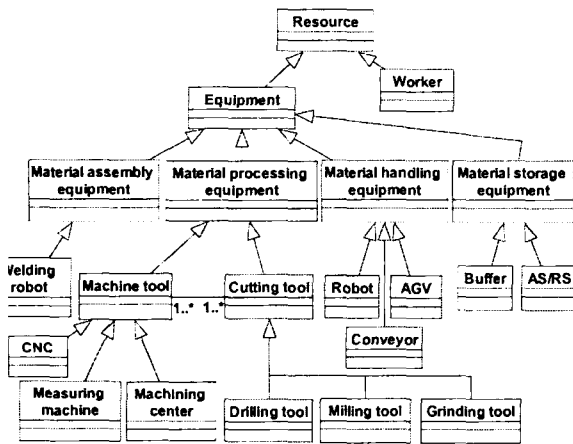


Fig. 3 Several sub-classes of Resource class

Resource 클래스는 원재료를 제품으로 변환시키는 Facility 클래스내의 모든 물리적인 요소를 나타낸다. Fig. 3에 Resource 클래스로부터 상속을 받아 만들어지는 다양한 하위클래스들을 보여주고 있다. Machine tool이 Cutting tool을 사용하는 것과 같이 Resource의 하위클래스들사이의 관계도 모델링될 수 있다.

Process 클래스는 어떤 원재료가 원하는 부품내지 제품으로 변환되도록 하는 활동내지 일련의 활동들로 정의된다. Fig. 4에 Process 클래스의 개략적인 하위클래스들을 나타내고 있다. Resource 클래스의 경우와 마찬가지로 이 하위클래스들은 Process 클래스의 상속을 받아 생성되고 Metal cutting process와 Machining feature와 같이 하위클래스들간의 관계를 모델링할 수 있다.

Strategy 클래스는 Resource와 Process를 사용할 때 적용되는 구속조건 내지 제한조건을 의미한다.

이 Strategy 클래스는 설비, 생산능력, 공정기술, 생산계획, 재료관리 등의 strategy로 분류할 수 있고 이 strategy의 데이터 구조는 전략적 의사결정, 작업규칙, 성능측정기준 등을 포함하고 있다.

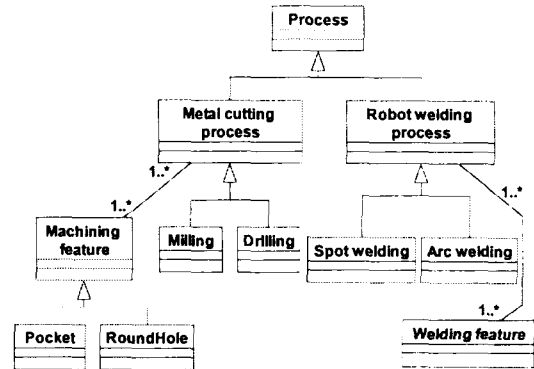


Fig. 4 Several sub-classes of Process class

Flow 클래스는 각각의 수행하고 있는 Process간의 선행관계를 표현한다. 즉 현재의 Process의 전후 Process를 내포하고 있다. 따라서 이 Flow 객체를 상호연결하면 전체적인 공정흐름을 알 수 있다.

WIP/Token 클래스는 수행하고 있는 Process의 전후 재공품 내지 부품을 나타낸다. 이 클래스를 이용하면 생산시스템의 전체적인 material flow를 표현할 수 있다.

4. Configurator의 framework

위에서 제시한 FDM을 근간으로 하여 Configurator의 전체적인 framework를 Fig. 5에 제시하고 있다. 크게 3부분으로 나누는데 Requirement Analysis, Module Selection, Configuration Optimization의 순서로 진행된다.

Requirement Analysis 단계는 top-down 방법을 사용하여 최상위 레벨의 요구조건을 분해하여 최하위 레벨에서 기본적인 기능이 되는 primitive function으로 나누어진다. 이 분해과정을 FDM의 4개의 층 즉 Factory layer, Shop layer, Cell/Line layer, Station layer으로 구분하여 이에 대응하는 Requirement layer, Process layer, Sub-process layer, Function layer에서 이루어진다. 분해과정은 제품을 생산하기 위한 공정들간의 전후관계도 유지하고 있으며 생산할 제품의 목표 산출량 및 cycle time도 각 층에서 설정가능하도록 기능분해가 이루어진다. 이러한 요구조건 분석은 Fig. 6에 나타난 것과 같이 공리적 설계의 기능 요구와 설계 변수사이의 지그재깅을 통한 분해과정으로 이루어진다. 그림에서 보듯이 기능 영역

의 4 개의 층을 function model, 원래 물리 영역이었던 영역을 조직 영역으로 재정의하여 class model로 정의하여 조직 구조에 맞게 기능 분해가 이루어지도록 한다.

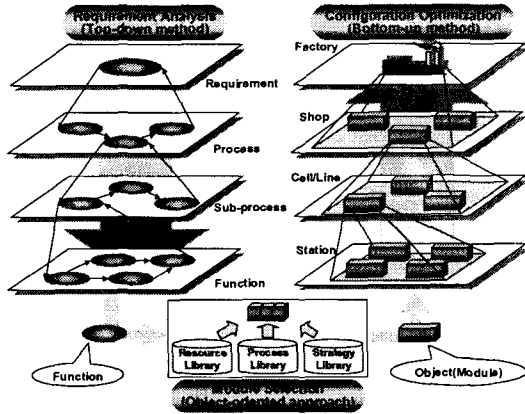


Fig. 5 The framework of Configurator

Module Selection 단계는 객체 지향 방법을 사용하여 각각의 primitive function을 수행할 수 있는 모듈 즉 객체를 생성하는데 이 객체는 station 객체를 의미한다. 객체를 생성하기 위해서 FDM의 Facility 클래스의 컴포넌트 클래스인 Resource, Process, Strategy 클래스가 필요하다. 앞장의 FDM에서 설명하였듯이 이 3개의 클래스는 상속을 통해서 좀 더 구체적인 하위클래스들로 분해되고 이 클래스들이 각각 Resource Library, Process Library, Strategy Library의 구성요소가 된다. 따라서 primitive function에 대응하는 Process 객체, 그리고 이 객체와 상호 관계를 가지는 Resource 객체, Strategy 객체가 각각의 Library에서 생성되어 Facility 클래스의 상속을 받은 Station 클래스의 객체로 통합된다.

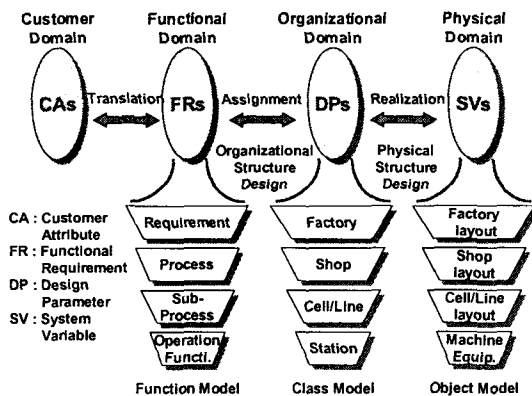


Fig. 6 The design domain for a systematic decomposition of Configurator

Configuration Optimization 단계는 Module Selection 단계에서 생성된 Station 객체들을 Station layer에서 배치하는데 각 Station 객체내의 Flow 컴포넌트 객체가 가지고 있는 process 전후관계 정보를 사용한다. 또한 WIP/Token 컴포넌트 객체를 사용하여 원활한 material flow를 위한 배치도 적용할 수 있다. 마찬가지로 몇개의 Station으로 구성된 Cell/Line 객체를 Cell/Line layer에 배치하는 것도 같은 방식이다. 또한 더 상위에 있는 Shop도 같은 방식으로 배치되어 최종적으로 Factory를 구성한다. Fig. 6에서 보듯이 원래 가공 영역이었던 영역을 물리 영역으로 재정의하여 조직영역의 class model을 구현하여 object model을 만든다. 이 object model이 공장의 생산시스템을 FDM로 모델링된 생산시스템을 의미한다.

5. 결론

동적이고 예측불가능한 시장 상황에 대처하기 위해서 민첩 대응 생산시스템이 대두되고 있다. 본 연구에서는 민첩 대응 생산시스템을 구현하기 위해서 생산시스템을 구성 및 재구성하는 과정을 Configurator라고 명명하고 In-plant 영역에서 Configurator를 구현하기 위한 접근방법을 고찰하였다. 또한 Configurator의 중심을 이루는 Factory Data Model(FDM)을 제시하였고 이 FDM을 근간으로 하여 Configurator를 구현하기 위한 framework을 제시하였다.

후 기

본 연구는 산업자원부에서 시행한 국제 IMS 프로그램 연구개발사업의 지원에 의한 것입니다.

참고문헌

1. Kenneth Preiss, "A Systems Perspective of Lean and Agile Manufacturing", *Agility & Global Competition*, Vol. 1, No. 1, pp. 59-76, 1997
2. Y. Naoda et al, "Total Support from Preliminary Design to Assembly Planning for Mechanical Products based on Functional Requirements and Modular Concepts", *Proceedings of The 35th CIRP International Seminar on Manufacturing Systems*, pp.512-517, 2002
3. Nam. P. Suh, "Axiomatic Design : Advances and Applications", Oxford University Press, 2001
4. J. Zhao et al, "A consistent manufacturing data model to support virtual enterprises", *International Journal of Agile Management Systems*, Vol. 1, No. 3, pp. 150-