The Journal of Korean
Institute of CALS/EC
Vol. 1, No. 2, Dec. 1996

기능 모형으로부터 프로세스 흐름 모형으로의 변환을 통한 시스템의 개념적 설계에 대한 연구

박찬권*, 신기태**, 박남규***, 박진우****

A Study on Conceptual Design through Function-to-Process Flow Model Transformation

Chan-kwon Park, Ki-Tae Shin, Nam Kyu Park, Jin-Woo Park

Abstract

An industrial process is a sequence of steps intended to perform a useful task in an industrial organization, and can be described using a model or a representation framework such as function model, information model, process flow model, etc. Many researchers, however, claim that, all those models from different perspectives are required to describe complex systems such as CIMS. Thus these perspectives should be put together to build an optimal system. And the need for interconnection or transformation among different view points has arisen for effective communications between system developers and easy propogation of local changes in a model. But it seems to be difficult to integrate such models into a unified framework as they describe different aspects of the system on different purposes. This study is an attempt to interconnect these different perspectives, and thereby to support conceptual design of systems in a more effective manner. A methodology to transform the IDEF0 functional model into a process flow model is proposed. The validity of the proposed approach is tested by developing a prototye of a computer-assisted design system on the operation processes of FMS installed at Seoul National University - Automation and Systems Research Institute.

* 영산국제산업대학교 경영정보학과 교수
** 대진대학교 산업공학과 교수
*** 생산기술연구원 생산시스템개발센터
**** 서울대학교 산업공학과 교수

1. 서론

모형은 특정 관점에서 복잡한 시스템을 관찰하기 위한 '개념적인 도구', 또는 현실 세계의 세부적인 사항을 생략하면서 실체를 추상화한 표현으로 정의할 수 있다. 이러한 모형의 주된 사용 목적은 대상 시스템을 연구 목적에 맞게 묘사하고 연구하기 위해서 이다.

한편 제조 시스템을 포함해서 모형화 대상으로서의 많은 실제 시스템은 범위, 수명 주기, 그리고 관점(perspective)과 같은 특성들을 가진다. 따라서 반영되는 연구 대상 시스템의 특성에 따라 다양한 형태의 모형이 존재한다. 예컨대, 전체 기업 수준, 공장 수준, 현장 수준과 같은 시스템의 범위는 모형의 규모를 규정하고, 설계, 분석, 구축 및 실행, 운영 등과 같은 시스템 수명 주기는 각 단계별로 모형의 주된 용도를 규정지으며, 관점(perspective)은 시스템의 기능적 측면, 행위적 측면, 조직적 측면, 정보 측면 등에 있어서 연구 대상 시스템에 대해서 누가, 언제, 어디서, 무엇을, 어떻게, 왜 해야 하는지를 묘사한다. 이 가운데 시스템을 보는 관점에 따른 기능 모형(function model), 정보 모형(information model), 행위 모형(behavioral model)은 특히 컴퓨터통합생산시스템을 비롯한 복잡한 시스템의 개발 및 구축에 있어 주축이 된다[Curtis, 1992]. 이들 세 가지 모형의 특성은 <표 1>과 같다.

이처럼 모형은 서로 다른 필요와 관점에 따라 시스템의 특정 측면을 묘사하기 때문에, 하나의 단일 모형을 통해서 전체 시스템에 대해 통합적으로 묘사하는 것은 불가능하다. 그러나 실제 시스템의 개발 과정에서는 여러 이유로 인하여 각 단계별, 용도별로 개발된 모형들간에 적절한 연계 또는 통합화의 요구가 제기된다. 이것은 서

<표 1> 프로세스 모형에 대한 세 가지 관점

	기능모형(Function Model)	정보모형(Information Model)	행위모형(Behavioral Model)
표현대상	시스템에서 수행되는 활동의 종류와 관련 정보 및 객체의 흐름을 표현	활동에 의해서 생성되고 다루어지는 정보나 데이터 실체, 그리고 이들 실체의 구조와 연관 관계를 묘사	활동간 수행 순서, 피이드백, 순환/반복, 시작/종료의 조건 등을 통해 시스템 구성요소의 동적변화를 표현
표현내용	· 활동/기능 · 입출력(input/output) · 제약 조건(constraint) · 자원 · 정보 및 객체 흐름	· 실체/객체(entity/object) · 관계(relationship) · 속성(attribute) · 추상화(abstraction) · cardinality	· 활동 · 시간 및 순차 관계 · 논리 및 조건 · 피이드 백/순환

로 다른 관점에서의 모형일지라도 시스템 설계 및 개발 과정에서 이들 모형들의 연계 관계가 규명되어야만 일관된 시스템 구축이 용이하기 때문이다. 또한 특정 관점에서 개발된 모형, 예컨대 기능 모형과는 별도로 전혀 다른 관점에서 새롭게 정보 모형이나 프로세스 흐름 모형 등을 개발하는 것보다는 기존의 기능 모형과의 연계 관계를 바탕으로 진화시키는 것이 바람직하다는 점에서도 모형 연계가 요구된다. 더욱이 전체 모형들이 개발된 후에는 특정 관점에서의 모형 변경 사항이 여타 모형들에 적절히 반영될 수 있기 위해서는 이러한 연계 관계가 필수적이다. 특히 설계 지원, 제조 통제, 검사, 자재 획득과 같이 다양한 구성 요소들을 바탕으로 기업의 제반 관련 기능들을 유기적으로 통합시키기 위해서는, 개발 과정에 존재하는 조직 관리 구조, 기능 구조, 정보 구조, 그리고 컴퓨터 시스템 구조와 같은 기능 단위들간의 기능적 연계 관계가 반드시 존재해야 한다.

본 연구는 CALS의 표준 방법론으로서 시스템 설계 단계에서 사용자 요구 정의(system requirement specification)에 보편적으로 사용되는 IDEF0 기능 모형(function model)을 행위 모형(behavioral model) 가운데 하나인 프로세스 흐름 모형(process flow model)으로 변환하기 위한 방법과, 이를 지원하는 시스템에 관한 것이다.

2. 관련 연구 현황

모형의 상호 연계 관계의 규명이나 모형의 통합을 위한 노력들은 크게 두 가지 관점에서 진행되어 왔는데, 프로세스 관점에서의 각 모형들을 통합(Multi-perspective Description)하려는 연구와 기업 통합 모형화(Enterprise Integration Modeling) 관점에서의 통합 노력이다.

먼저, 프로세스 관점에서의 통합에 대한 연구는 프로세스 모형에서의 여러 관점들을 연계해서 표현하려는 시도로서 기능-정보, 정보-행위, 그리고 기능-행위의 부분적 연계와 기능-정보-행위의 다중 관점 연계에 관한 연구가 이루어졌다.

기능-정보 모형의 연계에 관한 연구([신기태, 1995; Hsu, 1995; Kim, 1993])들은 정보 모형 관점 위주의 접근 방법이다. 즉, 궁극적으로 정보 모형을 구축하기 위한 요구 정의의 예비 단계 혹은 임시적인 단계로 기능 모형을 사용하고 있기 때문에 형식상 두 모형이 연계되고는 있지만 모형의 부분적 변경에 의한 연계 모형으로의 반영 등을 기대하기는 어렵다.

기능-행위 모형의 연계에 관한 연구로는 기능 모형에서 Petri Net으로의 변환을 다룬 연구([Pinci, 1993; Boucher, 1992; Vernadat, 1993; Liang, 1994; Zaytoon, 1994])와 시뮬레이션으로의 연계를 보인 연구([Mujtaba, 1994])가 있다. 이들 기능 행위 모형에 관한 연구는 대체적으로 기능 모형이 활동(activity) 중심으로 표현된다는 점에 착안해서, 활동을 Petri Net의 트랜지

션과 플레이스의 적당한 조합으로 변환하고, 기타 기능 모형을 구성하는 요소들을 화살표(arc)로 연결시킴으로써 변환의 메커니즘을 제공하고 있다. 이들 연구는 기능 모형이 가지는 한계 즉, 정량적인 분석이 뒷받침되지 못한다는 점을 보완하고는 있다. 그러나, 요구 정의 단계에서의 정성적인 특성을 반영하는 기능 모형을, 정량적이고 수리적인 특성을 가지는 Petri Net으로 직접 연계시킴으로써 사실상 기능 모형에 많은 제약을 가하고 있다. 예를 들면, 각 활동간의 객체의 흐름은 독립적으로 이루어질 수 있음에도 불구하고, 정보 흐름과 물류 흐름이 함께 발생하는 것으로 가정하든지 혹은 정보 흐름은 항상 물류 흐름에 앞서 다음 활동에 제공되어야 한다는 제약을 담고 있다.

정보-행위 모형의 연계에 관한 연구는 앞서의 연계 모형들에 비해서는 상대적으로 활발하지 못하다. Lewis et al.은 CIM 시스템의 제어를 위한 방법으로 데이터 흐름 접근 방법을 이용하고 있고, Warren et al.은 정보 시스템의 설계에 대한 확률적 시뮬레이션(stochastic simulation)에 DFD (data flow diagram)을 이용하고 있다 [Lewis, 1987; Warren, 1992]. 하지만 DFD는 정보 모형이라기 보다는 기능 모형의 일종으로 간주될 수 있다. Chaillet et al.은 Petri Net과 데이터베이스 모형을 병합함으로써 유연생산시스템의 통제 및 감시 요구를 모형화하고 있다[Chaillet, 1993]. 이처럼 정보-행위 모형의 연계는 대해서는 정형화된 연구 경향을 발견하기 어렵다.

한편 기능-정보-행위 모형의 연계에 관한 연구([Engelke, 1985; Banerjee, 1988; Young, 1991; Malhotra, 1992; Wang, 1993])들은 전반적으로 시스템 개발 방법론에 입각한 시스템 수명 주기에 따라 구조적인 방법의 적용을 제안하고 있다. 이러한 접근 방법에 있어 가장 큰 문제점은 제안된 방법론 또는 모형들 사이의 상호 연계성(cohesion)이 부족하다는 것이다. 즉, 하나의 모형에서 다른 모형으로 넘어가는 과정에 있어서는 전혀 다른 관점에서 다시 모형화를 시작해야 하거나, 한 모형에서 부분적 변경이 여타 모형에 주는 영향과 같은 요소들이 언급되지 않고 있다.

기업 통합 모형화의 관점은 특정 프로세스에 국한되지 않고 기업 모형 구축을 위한 참조 모형(Enterprise Reference Model)이나 참조 모형 구조(Enterprise Reference Architecture)의 관점에서 접근하는 것이다. 기업 참조 모형을 위한 접근 방법은 여러 단체나 연구자들에 의해 시도되고 있으나, 그 중 가장 두드러진 것들로는 CIM-OSA(Open System Architecture for Computer Integrated Manufacturing), PERA(Purdue Enterprise Reference Architecture), 그리고 GRAI-GIM(GRAI Integrated Methodology)이 있다[AMICE, 1988; Williams, 1994a; Williams, 1994b].

기업 통합 모형화의 관점의 큰 특징은 프로세스 통합이라는 측면과 함께 사용자 요구 분석에서 실제 시스템의 실행이라는 시스템의 전체 수명 주기를 표현하고자 한다는 점이다. 하지만 이러한 접근 방법들

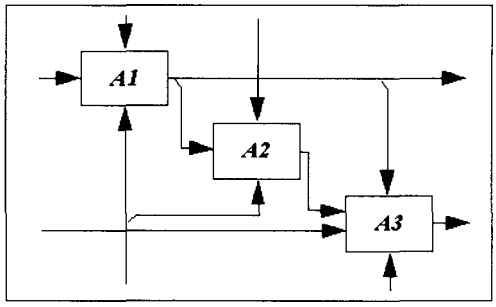
은 프로세스에 대한 개별 관점-기능, 정보, 행위 등 의 통합이나 연계에 대한 체계적인 접근보다는, 프로세스와 수명 주기의 통합을 바탕으로 프로젝트 전체의 구조적인 틀(framework)의 제공에 역점을 두고 있다고 보여진다.

3. IDEF0 기능 모형과 프로세스 흐름 모형

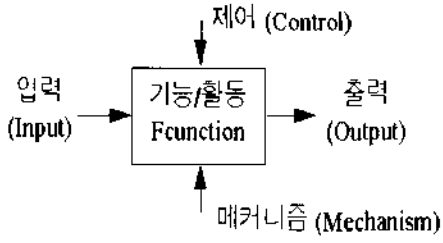
3.1 IDEF0 기능 모형

IDEF0는 시스템과 환경의 기능들 그리고 기능과 관련된 정보 또는 객체들을 그래픽 표현을 이용하여 구조적으로 표현하는 기능 모형이다. 기본적으로 SADT ([Ross, 1977])를 모태로 활동 모형(actigram) 부분을 확장시킨 IDEF0는 90년대에 들어와서 NIST(National Institute of Standards and Technology)에 의해 FIPS(Federal Information Processing Standard)의 표준 문서로 발표(FIPS PUBS 183)되었다[NIST, 1993; Mayer, 1994].

기능을 모형화 하는 도구로써 IDEF0는 제약 조건 아래서 입력을 출력으로 변환키는 활동을 묘사하는데 사용된다. IDEF0의 가장 중요한 요소는 그래픽 표현과 계층적 구조이다. IDEF 모형은 <그림 1>과 같이 정의되는 도표(diagram)의 연속된 계층으로 구성되는데, 박스는 기능을 나타내고 화살표는 기능간 관계를 정의한다.



<그림 1> IDEF0 모형의 도표(diagram)



<그림 2> IDEF0의 ICOM 구조

<그림 2>와 같이 정의되는 박스는 기본적으로 활동, 프로세스, 운용 또는 변환 작용을 나타내는 기능을 정의한다. 즉, 표현된 기능에서 무슨 일이 일어나는지에 관한 선명이다.

기능을 나타내는 박스와 관련되는 데이터 또는 객체를 표현하기 위해서 하나 또는 그 이상의 화살표를 사용하는데, ICOM이라 불리는 이 화살표는 박스에서의 상대적인 위치에 따라 역할이 주어진다. 즉, 기능을 나타내는 박스를 연결하는 화살표는 한 박스의 출력이 다른 박스의 기능 수행에 필요한 입력, 제어, 또는 메커니즘

으로 제공되는 것을 의미하게 된다. 화살표는 여러 박스로의 입력을 위해서 분기하기도 하고, 여러 박스에서의 입력을 위해 병합되기도 한다.

IDEF0와 여타 방법론들의 모형화 방법과 제공되는 모형화 도구들(modeling constructs)을 중심으로 한 비교 내용은 <표 2>와 같다.

3.2 프로세스 흐름 모형

IDEF0가 가지는 장점으로는 우선 이것이 광범위하게 적용 가능한 시스템 모형화 기법이라는 것이다. 또한 그래픽을 이용함으로써 단순하고 효율적인 의사 소통이 가능하며, 시스템을 일관성 있게 분석할 수

<표 2> 모형화 방법론의 비교

	IDEF0	CIM-OSA 기능 뷰	IDEF3
기능/활동에 대한 기본 모형화 도구	· 기능(Function)	· 영역 프로세스 (Domain Process) · 업무 프로세스 (Business Process) · 단위활동(Enterprise Activity)	· 행동단위(UOB)
기능/활동의 연결의 의미와 의미 표현 방법	· 입력/제어/메커니즘의 제공과 출력 산출 · 화살표(arrow) · 화살표의 위치	· 단위 활동의 시작(triggering) 화살표 · 절차 규칙 집합(PRS)	· 행동단위간의 순서, 관계, 객체 흐름 · 화살표 유형(procedenc/relational/object flow)
연결의 분기 및 결합	· 데이터 추상화 또는 연결 통로의 개념 · fan-in, fan-out	· 결합 : rendezvous, convergence · 분기 : spawning, conditional	· 접속(Junction) - 결합(fan-in) - 분기(fan-out)
논리의 표현	명시적인 표현 없음	· AND · (X)OR (OR/XOR의 명시적 구분 없음)	· AND · OR · XOR
시간적 관계의 표현	적용 안됨	· 활동들간의 직접적인 시간적 관계 표현은 없음	· 동시(synchronous) · 비동시(asynchronous)
상세 설명	· FEO	적용 안됨	· Referent
사건(event)	· 별도 설명 없음	· Enterprise Event	· 행동단위 자체로 표현
계층의 구성 방법	· 부모/자식 구조를 이용한 기능 분해	· 부모/자식 구조를 이용한 프로세스/활동의 분해와 규칙 집합	· 부모/자식 구조를 이용한 기능 분해

* 밑줄 친 부분은 모형화 도구(modeling construct)로 정의된 내용

있도록 해준다. 그리고 구조적이고 시스템적인 접근 방법이기 때문에 시스템 내에서의 연관 관계를 효과적으로 나타낼 수 있다. 다음으로는 계층적 모듈화된 모형 구조를 통하여 시스템에 대한 깊은 이해가 가능하다는 점이다. 이밖에도 컴퓨터를 이용한 그래픽 도구들이 다양하게 이 방법론을 지원하고 있다는 것도 하나의 장점이다. 이러한 장점으로 인하여 복잡한 구성 요소로 이루어져 있으며, 기능들이 계층적인 구조를 지니는 생산 시스템을 분석하는데 적합한 방법론이 될 수 있다.

반면에, IDEF0 방법론에 있어서는 몇 가지 단점들이 지적되기도 한다[Busby, 1993]. 대표적인 것으로는 먼저, 정형적이라기 보다는 구조적인 특성으로 인하여 모형의 의미(semantics)를 수리적인 방법에 의해 정의하기 힘들다는 것이다. 즉, 모형에 대한 정량적인 분석과 평가가 어렵다는 것이다. 다음으로는, 여타 IDEF 방법론간의 연관 관계가 취약하다는 것이다[신기태, 1995]. 즉, 기능 모형, 정보 모형, 그리고 동적 모형에 있어 구성 요소 상의 관계 확보가 어렵기 때문에 기능 모형에 제공하지 못하는 특성, 예컨대 시간의 추이에 따른 시스템의 동작 행태를 표현하거나, 기능 모형에 있어 활동의 시간적 순서 관계, 동시 수행 가능성 등을 파악하기가 어렵다는 것이다. 따라서 본 연구에서는 시간적 순서 관계와 동시 수행 가능성 등을 표현하고 IDEF0와의 연계가 가능하도록 제안된 PROWD(PRocess flOW Description) 프로세스 흐름 모형[박찬권, 1996b]을 사용한다.

프로세스 흐름 모형화 방법론 가운데 가장 많이 알려진 것 중 하나가 IDEF3 방법론인데, IDEF3는 IDEF0 기능 모형과의 연계성이 미흡해서 모형을 별도로 작성해야 하는데 반해, 실체 흐름을 기반으로 한 PROWD 모형은 IDEF0 기능 모형으로부터 변환이 용이하다는 장점이 있다. PROWD 모형은 IDEF0 모형의 계층적 분해에 의한 시스템 분석을 지원하는 한편, IDEF3 모형화 방법을 보완해서 기능/활동사이의 동기 관계(synchronization relationship)와 순차 관계를 보다 다양하게 표현할 수 있는 능력을 제공하고 있다. PROWD 모형에서 정의하고 있는 모형화 도구(modeling constructs)와 주요 모형화 도구에 대한 데이타 구조는 <표 3>, <표 4>와 같다.

4. IDEF0 모형으로부터 프로세스 흐름 모형으로의 변환

IDEF0 기능 모형은 시스템에서 수행하게 되는 기능/활동을 직접적으로 묘사하게 되는데, 여기에는 실제적으로 기능/활동의 대상이 되는 입력과 출력의 표현이 함께 이루어진다. 이러한 입력과 출력이 IDEF0 모형에서 ICOM 구조로 표현된다는 사실은 앞서 언급한 바 있다. 제조 시스템의 경우, 데이터, 물리적 객체 또는 자원과 같은 실체(entity)들이 바로 ICOM을 통해 표현된다.

한편, ICOM 구조에서 출력(output)으로 표현되는 실체로는 정보 실체와 물리적

<표 3> PROWD 모형의 모형화 도구(modeling constructs)

모형화 도구		의 미	그래픽 표현	속 성
기능(function)		기능/활동을 표현하기 위한 기본 모형화 도구로서, 기능(function), 활동(activity), 운용(operation), 행위(action) 등을 표현.	· box(단일실선) (입력/제어/ 메카니즘/출력)	[identifier, description, input, control, mechanism, output, layer, input/control logic, output logic]
단위 기능 (Unit Function)		별도의 변환 과정 없이 입력은 입력으로, 제어는 제어로 연결	· box(이중실선)	[identifier, input, control, output]
연결 (link)	입력	기능/활동의 수행 대상이 되는 입력의 제공을 의미	활동을 나타내는 박스의 왼쪽에 연결	[identifier, description, triggering type, start, end]
	제어	기능/활동의 수행에 대한 제약 또는 제어를 의미	활동을 나타내는 박스의 위쪽에 연결	
	메커니즘	기능/활동을 직접 수행하거나 수행을 지원하는 것을 의미	활동을 나타내는 박스의 아래쪽에 연결	
논리적 접속 (junction)	AND	분기 또는 결합되는 연결에 해당되는 입력 또는 제어가 모두 제공되어야 함을 의미	작은 사각형 안에 '&'	[identifier, type, description, rule]
	OR	분기 또는 결합되는 연결에 해당되는 입력 또는 제어 중 일부가 제공되어야 함을 의미	작은 사각형 안에 'O'	
	XOR	분기 또는 결합되는 연결에 해당되는 입력이나 제어 중 오직 하나만이 제공되어야 함을 의미	작은 사각형 안에 'X'	

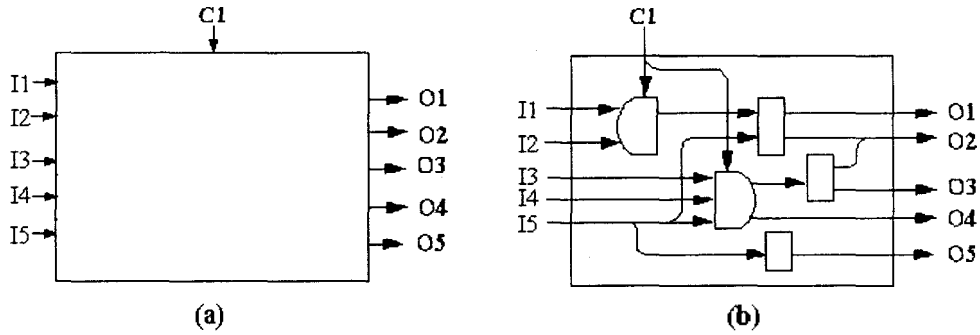
실체가 있는데, 이것은 또 다른 기능/활동에 대해서 각각 입력, 제어, 그리고 메커니즘으로 작용할 수 있다. 따라서 기능/활동들의 활성화에 관여하는 실체들은 그 역할과 유형에 따라 정보 흐름(information flow), 자재 흐름(material flow), 제어 흐름(control flow), 그리고 자원 흐름(resource flow) 등을 구성하게 된다. 이들 흐름은 기능/활동간의 연결을 통해 표현되고, 기능/활동간의 기본적인 순서 관계를 정의한다. 그럼에도 불구하고 IDEF0 기능 모형으로부터 일관되게 순서 관계를 정의하기는 어렵다. 그 이유는 하나의 기능 활동이 각기 다른 조건에 의해 각기 다른 입력 조합에

<표 4> PROWD 모형의 주요 모형화 도구에 대한 데이타 구조

FUNCTION Template			비 고
identifier		Axx	기능의 노드 identifier를 사용
description		textual description	
input		Ix	
control		Cx	
input/control logic		AND | OR | XOR	입력과 제어 사이의 논리
mechanism		Mx | Vx	
output		Ox	
output logic		AND | OR | XOR	여러 출력 사이의 논리
layer		n	위상 순서
LINK Template			**비 고**
identifier		Lx	link에 identifier를 부여
description		textual description	
triggering type		TE | NTE	격발 또는 비격발 여부
tail	function	SYS | Axx[/Jn]	연결의 시작(시스템 외부 또는 기능 Axx) 접속을 통할 경우는 [/Jn]옵션
	type	NULL | Output	시스템 외부일 경우는 NULL
	identifier	NULL | Ox	시스템 외부일 경우는 NULL
head	function	SYS | [Jn/]Axx	연결의 끝(시스템 외부 또는 기능 Axx) 접속을 통할 경우는 [Jn/]옵션
	type	NULL | Input | Control	시스템 외부일 경우는 NULL
	identifier	NULL | Ix | Cx	시스템 외부일 경우는 NULL
JUNCTION Template			**비 고**
identifier		Jn	접속마다 identifier를 부여
logic		textual description	AND | XOR | OR
type		fan-out | fan-in	분기 또는 결합 여부
rule		textual description	분기 또는 결합의 선택을 위한 규칙

대해 활성화되는 다중 활성 모드의 존재가 존재하거나, 피이드 백에 의한 사이클이 존재하기 때문이다. 그러므로 본 논문에서는 이들 다중 활성 모드와 사이클에 대한 처리를 통해, IDEF0 기능 모형을 프로세스 흐름 모형으로 변환하게 된다.

이러한 변환은 <표 1>에서 살펴본 것처럼 기능 모형이 기능(function) 또는 활동(activity) 위주로 표현되고, 프로세스 흐름 모형(process flow model) 또한 활동



<그림 3> 추상화된 기능에서의 입출력과 분해된 기능에서의 입출력 조합

(activity) 위주의 묘사라는 점에서 출발한다. 즉, IDEF0 모형의 중심이 되는 기능(function box)이 시스템에서 수행되는 활동(activity), 의사 결정(decision making), 행위(action), 또는 운영(operation) 등을 묘사하는 데 사용되고, 이들 기능이 프로세스 흐름 모형에서 그대로 단위 활동(unit of activity)으로 전환될 수 있기 때문이다.

4.1 활성 모드(activation mode)

IDEF0 모형이 하위 계층으로 분해되어 내려감에 따라, 화살표의 존재 즉, 실체의 흐름이 어느 정도 순차 관계로 직접 해석될 수 있는 가능성을 가지고 있음에도 불구하고 전체적으로는 순차 관계에 의한 프로세스 흐름을 표현하지 못하는 이유는 하나의 기능/활동이 활성화(activation)되는데 모든 입력과 출력이 동시에 관여하지 않기 때문이다.

예를 들어 <그림 3>의 (a)에서와 같이 기능/활동이 정의되어 있는 경우, 출력 O1, O2, O3, O4, O5를 내놓기 위해서는 모든 입력과 제어 I1, I2, I3, I4, I5 그리고 C1이 동시에 전부 필요한 것이 아니라, (b)와 같이 일부의 입력/제어/메커니즘만이 특정 출력의 제공에 관여할 수 있기 때문이다. 여기서 'D' 모양의 게이트와 작은 사각형 게이트는 각각 입출력의 논리적 조합과 변환 과정을 나타낸 것이다. IDEF0 모형에서는 이처럼 입력과 제어의 부분적인 조합들이 서로 다른 출력의 조합을 제공할 수 있기 때문에, 이들 조합의 존재는 기능이 하위 계층으로 분해되고 나서야 구체적인 관계를 확인할 수 있다.

한편 기능/활동이 매 번 수행(occurrence)되는 동안 모든 입력과 제어가 동시에 관여해서 모든 출력을 제공하는 경우, 즉 입출력의 조합이 유일하게 존재하는 경우에는 기능/활동 사이에서의 순차 관계가 정의될 수 있다. 하지만 경우에 따라서 기능/활동이 수행되는 동안 서로 다른 조합의 부분적인 출력만을 제공할 수도 있다. 예컨대 피이드 백이 존재하는 경우, 처음

수행에는 피이드 백 받는 입력이나 제어가 관여할 수 없다. 왜냐하면, 시간적으로 나중에 발생하는 피이드 백의 결과를 해당 기능/활동의 활성화 이전에 입력이나 제어로 요구하는 것은 논리적으로 어긋나기 때문이다. 이런 경우는 교착 상태(dead lock)가 발생한다. 그러므로 매번 기능/활동의 수행에 관여하게 되는 입력/제어/메커니즘과 출력의 조합을 하나의 활성 모드로 정의함으로써, 처음 활성화에는 피이드 백 결과가 미리 입력이나 제어로 사용되는 것을 배제해야 한다.

결국, IDEF0에는 이러한 다중 활성 모드가 존재하기 때문에 순차 관계에 의한 프로세스 흐름 분석이 불가능하게 되는 것이다. 따라서 다중 활성 모드가 존재하는 경우에는, 한 번의 수행을 통해 관여하게 되는 입력/제어/메커니즘과 출력의 적절한 조합을 찾아냄으로써 프로세스 흐름 모형으로 변환이 가능하다.

다중 활성 모드는 하나의 기능/활동에 관여하는 입력, 제어, 메커니즘, 그리고 출력의 수가 많지 않을 때는 직관적으로 쉽게 찾을 수 있지만, 상위 레벨에서와 같이 여러 개의 입력, 제어, 메커니즘, 그리고 출력이 있는 경우는 쉽지가 않다. 따라서 이와 같은 경우에 적용할 수 있는 체계적인 방법이 필요한데 다음과 같은 절차를 이용한다.

먼저, l개의 입력, m개의 제어, n개의 메커니즘, 그리고 o개의 출력이 정의되어 있는 기능/활동 A^i에 대해서 공헌행렬(contribution matrix) CM^i를 작성한다. 입력과 제어, 메커니즘은 행을, 그리고 출력은 열을 각각 구성한다. cm_{ij}는 입력 또는 제어 i가 출력 j를 제공하는 데 관여하는 회수를 나타낸다. 즉, 서로 다른 입력 조합에 의한 활성화를 통해 출력 j를 제공하는 회수를 의미한다. 이 때, 가능한 출력 조합의 수(NOC)는 다음과 같이 정의할 수 있다.

$$NOC = \text{Rank of } \langle CM^i \rangle \ (\ i\ 1,\ldots,\ l,\ l+1,\ldots,\ l+m,\ l+m+1,\ \ldots,\ l+m+n,\ j=1,\ \ldots,\ o\ )$$

그렇게 되면, 행렬 CM^i의 Rank는 서로 독립인 행 또는 열의 수를 나타내고, 이것은 출력의 조합을 제공하는 데 필요한 입력 및 제어 조합으로 해석될 수 있다. 이들 입출력 조합들 가운데, 기능/활동이 한 번 활성화할 때 관여하는 모든 입출력 조합이 하나의 활성 모드를 정의한다.

<그림 3>의 예제에 대해서, 기능/활동 별로 공헌 행렬을 작성해보면 다음과 같다.

$$CM = \begin{array}{c} \\ I1 \\ I2 \\ I3 \\ I4 \\ I5 \\ C1 \end{array} \begin{array}{c} \begin{array}{ccccc} O1 & O2 & O3 & O4 & O5 \end{array} \\ \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 0 \end{vmatrix} \end{array}$$

여기서, CM의 Rank는 3으로 세 개의 입력 및 제어와 출력의 조합이 존재한다. 즉, 입력 I1, I2, I5와 제어 C1이 출력 O1 및 O2를 생성하고, 입력 I3, I4, I5와 제어 C1이 출력 O2, O3, 그리고 O4의 생성에

관여한다. 마지막으로 입력 I5는 출력 O5를 내놓게 된다. 이 경우 각 입출력 조합을 나타내는 공헌 행렬은 다음과 같다.

$$CM_1 = \begin{array}{c} \\ I1 \\ I2 \\ I5 \\ C1 \end{array} \begin{array}{c} \begin{array}{cc} O1 & O2 \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

$$CM_2 = \begin{array}{c} \\ I3 \\ I4 \\ I5 \\ C1 \end{array} \begin{array}{c} \begin{array}{ccc} O2 & O3 & O4 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \quad CM_3 = \begin{array}{c} \\ I5 \end{array} \begin{array}{c} O5 \\ [\ 1\ ] \end{array}$$

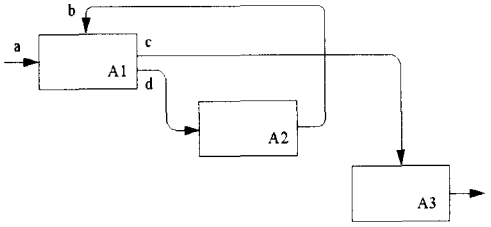
이 세 가지 조합 가운데 한 번의 기능/활동 활성화에 의해 적집 나올 수 있는 출력의 조합들이 하나의 활성 모드를 결정하게 된다. 예컨대, 처음 활성화 때, 출력 조합 O1/O2와 O5가 생성될 수 있다면, O1, O2, O5는 하나의 활성 모드를 구성하게 된다. 따라서 기능/활성이 한 번 수행될 때 이들 조합 가운데 하나씩만이 활성화된다면, 예제의 기능/활동에 대한 활성 모드는 3으로서 모두 세 개의 기능/활동으로 분해될 수 있다. 기능/활동이 분해될 때, 기존의 연결 관계는 모두 상속되기 때문에, 입출력 조합을 나타내는 각 공헌 행렬 CM_1, CM_2, CM_3으로부터 연결 관계를 쉽게 정의해낼 수 있다.

4.2 사이클

기능/활동이 여러 조건에서 다르게 활성화되는 다중 활성 모드의 존재는 상위 수준에서의 기능/활동이 여러 가지 하부 기능/활동을 추상화해서 표현하는 데에도 원인이 있지만, 근본적으로는 피이드 백과 같은 사이클이 존재하기 때문이다. 따라서 앞서 도출한 활성 모드를 바탕으로 기능/활동을 분해(split)해서, 이러한 사이클의 발생을 제거하도록 한다.

피이드 백은 피이드 반복 수행이 발생하는지 여부에 따라 정 피이드 백(positive feedback)과 교정 피이드 백(corrective feedback)으로 나눌 수 있다[Mandel, 1990]. 정 피이드 백은 피이드 백을 생성시키는 기능/활동의 성패 여부를 판단 할 수 있는 기준이 있지만, 반복해서 기능/활동을 수행 할 필요가 없는 경우로, 실패한 경우에 대해서도 교정 작업을 수행하지 않고 계속 진행하게 된다. 따라서 정 피이드 백은 제어보다는 입력으로 간주될 수 있다. 이 때 최초 입력의 기준에서 보면 피이드 백 입력은 기능/활동을 수행하는 데 선택적으로 필요한 사항으로서 IDEF0 기능 모형의 작성 시에는 이러한 사항을 피하도록 권하고 있다[Mayer, 1994]. 한편 교정 피이드 백은 일정한 기준에 따라 교정 작업을 위한 반복 수행을 수반하는데, 이 가운데 부 피이드 백은 성패의 여부를 가를 수 있는 기준을 가지는, 가장 일반적인 형태의 피이드 백이다.

교정 피이드 백의 경우는 최초 입력과 피이드 백 입력이 어떻게 출력에 대응되는가에 따라 다시 단순 순환 반복(iteration)과 복합 기능/활동으로 나눌 수 있다. <그림 4>와 같은 예에서, 순환 반복은 최초 입력 a와 피이드 백 입력 b가 피

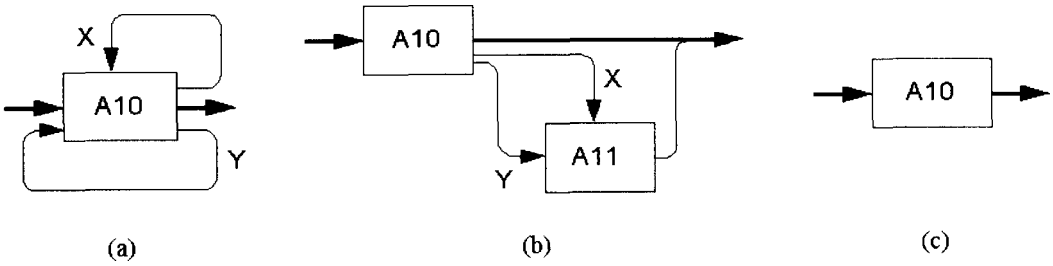


<그림 4> 사이클의 발생

있다.

순환 반복(iteration)의 경우는 동일한 기능/활동에서의 반복 순환과 여러 개의 기능/활동에 걸쳐 반복 순환이 발생하는 경우가 있다. <그림 5> (a)와 같이 동일한 기능/활동 내에서의 순환 반복은 활성 모드에 따라 기능/활동을 분해하면 (b)와 같이 표현될 수 있다.

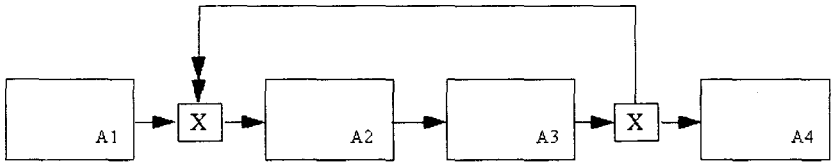
이러한 단순 반복은 굵은 실선으로 나타나는 주 흐름을 분석했을 때, 전체 프로세스의 흐름을 파악하는 데 도움이 되지 않음을 알 수 있다. 따라서, 동일한 기능/활동에서의 순환 반복(iteration)은 (c)와 같이 생략하도록 한다.



<그림 5> 단순 순환 반복(iteration)의 표현

이는 백 출력 c와 일반 출력 d를 구분하지 않고 동일한 출력 집합 { c, d }를 변역으로 가지는 경우이다. 즉, a 또는 b가 들어갈 때 출력은 c 또는 d가 나올 수 있는 경우이다. 이때는 기능 A1에서 일정한 조건을 만족시키기까지 기능/활동이 반복해서 수행되는 것을 나타내다. 반면에, 복합 기능/활동은 최초 입력 a와 피이드 백 입력 b가 입력의 종류에 따라 별도의 출력이 미리 지정되어 있는 경우로서 {a} → {c}, {b} → {d} 일 때를 나타낸다. 이때는 앞 절에서 설명한 활성 모드의 정의에 의해 기능/활동을 분해함으로써 사이클을 해제할 수

여러 개의 기능/활동에 걸쳐 순환 반복이 발생하는 경우도 마찬가지로 해석될 수 있고, 이 때도 순환 반복 자체는 프로세스 흐름을 분석하는 데에는 큰 영향을 미치지 않는다. IDEF0에서는 이와 같은 형태의 모형을 작성하지 않도록 권고하고 있기 때문에[Mayer, 1994], 주로 <그림 6>과 같은 유형이 순환 반복(iteration)의 주된 형태가 된다. 따라서, 여러 개의 기능/활동에



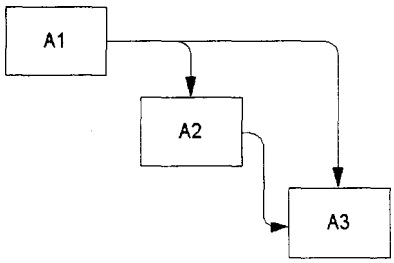
<그림 6> 여러 개의 기능/활동에 걸친 순환 반복(iteration)

서의 순환 반복(iteration)은 <그림 6>과 같이 두 개의 머리를 가진 화살표를 이용해서 표현하도록 한다.

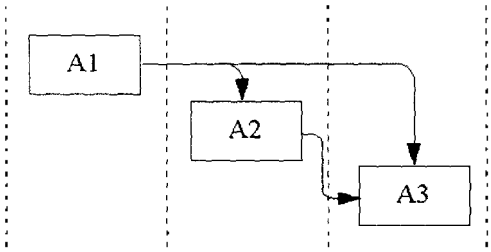
4.3 연결 의미의 해석과 위상 순서

IDEF0에서 화살표는 하나의 박스에서 또 다른 박스로의 입력이나 제어 또는 메커니즘의 제공을 나타냄으로써 두 개의 박스 즉, 기능/활동 사이에 존재하는 관계를 표현하는 데 이용된다. IDEF0에서 두 활동을 연결하는 화살표는 기본적으로 제약 조건(constraints)의 의미와 동시 실행(concurrency)의 의미로 해석될 수 있다 [NIST, 1993; Mayer, 1994]. 따라서 이들 연결 관계로부터 순차 관계와 동시 수행 관계를 정의하도록 한다.

먼저 <그림 7>은 한 기능에서의 출력이 또 다른 기능에 대해 제약 조건(constraints)으로 작용하는 경우이다. 즉, 기능 A1이 종료되고 A1에 의해 생성된 데이터나 객체가 A3에 제공되어야 기능 A3이 시작할 수 있음을 의미한다. 마찬가지로 A3의 수행에는 기능 A2로부터의 출력이 필요하다. 이 것은 IDEF0가 입력, 제어, 메커니즘의 역할이 제약 조건임을 명시하고 있기는 하지만, 의미상으로 기능/활동들 사이에 순차 관계(precedence)가 존재하고 있다는 것을 보여준다.



<그림 7> 제약 조건으로서의 입력 및 제어



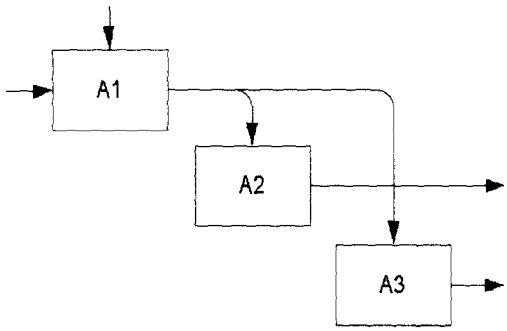
<그림 8> 순차 관계의 정의

따라서 이와 같이 선후 관계가 분명히 존재하는 경우는 <그림 8>과 같이 순차(precedence) 관계를 정의한다. 원래의 IDEF0 모형이 기능을 표현하는 박스를 대각선으로 배열하는 반면, 여기서 순차 관계로 정의되는 박스는 세로 축을 기준으로 하여 나중에 수행되는 기능이 오른쪽에 위치하도록 배열한다.

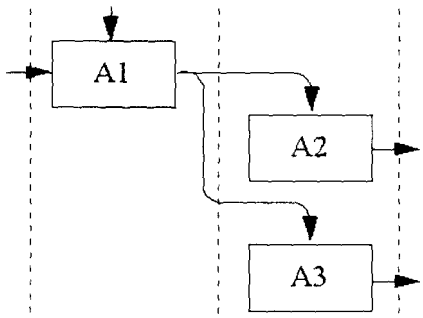
한편, <그림 9>와 같은 경우, IDEF0는 기능/활동의 동시 실행(concurrency)을 의미하게 되는데, 기능 A1이 종료한 후 일단 출력으로 데이터 또는 객체가 제공되기만 하면, 그 출력을 제약 조건으로 하는 활동 A2, A3은 동시에 실행이 가능하다는 것이다.

이 경우에 있어서도 A1은 A2와 A3보다는 먼저 실행 및 종료되어야 하는 순차 관계(precedence)를 내포하게 된다. 따라서 A1과 A2, A1과 A3 사이에는 앞서의 <그림 8>과 같은 순차 관계가 정의되고, A2와 A3 사이에는 <그림 10>과 같은 동시 수행 관계가 정의될 수 있다. 순차 관계와 마찬가지로 동시 수행 관계에 놓이는 기능들은 가로축을 기준으로 상하에 위치하도록 배열한다.

출력 제어를 의미하는 연결은 대부분 <그림 8>과 같은 순차 관계로 해석되지만, 선행하는 기능/활동이 시작되면서 연결된 기능/활동이 동시에 수행되어야 함을 나타내는 동기 제어는 순차 관계로 해석하는 대신 <그림 10>과 같은 동시 수행 관계로 해석해야 한다. 그러므로 이 때는 상하로 기능/활동을 배열하면서, 이중 실선을 이용해서 연결을 나타내도록 한다.



<그림 9> 동시 실행의 표현



<그림 10> 동시수행 관계의 정의

한편, 개별 기능/활동 사이에서의 연결 의미는 순차 관계로 해석될 수 있기 때문에 비교적 쉽게 해석이 가능하지만, 모형의 규모가 커서 기능/활동의 수가 많아지면 개별 기능/활동 사이의 순차 관계를 바탕으로 전체적인 동시 수행 관계의 파악이 쉽지 않다. 따라서 전체 모형에서의 기능/활동들에 대한 순차 관계 및 동시 수행성을 파악하기 위해서는 개별 기능/활동간의 순차 관계를 바탕으로 전체 기능/활동들의 순서를 정의해야 한다. 전체 기능/활동에 대한 순서의 부여는 무환 네트워크에서 각 노드의 위상 순서(topological order)를 찾는 문제로 볼 수 있다. 따라서, 동기 제어(synchronization control)를 고려한 각 기능/활동의 위상 순서(topological order)를 찾기 위해서는, 각 연결에 대해 0과 1 또는 ε $(0<\varepsilon<1)$의 연결 값(indegree)을 정의하고, 연결 값을 바탕으로 변형된 횡단 탐색(MBFS: Modified Breadth First Search) 알고리즘([박찬권, 1996a])을 적용한다.

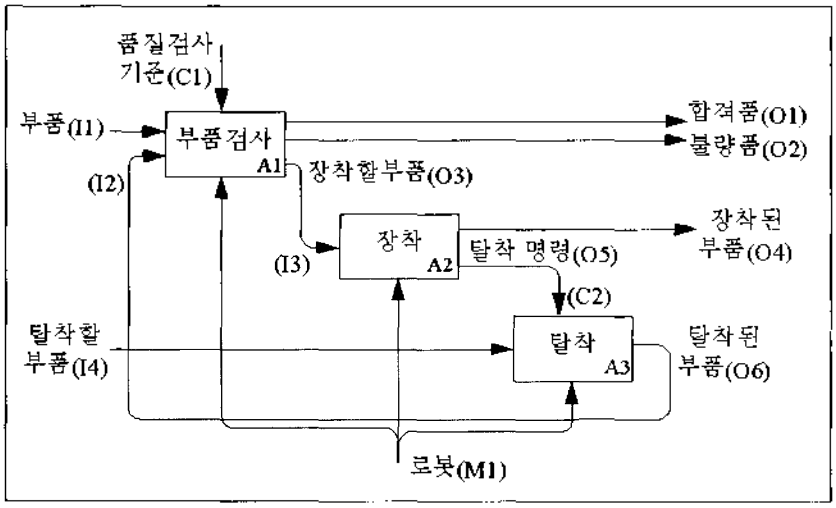
4.4 격발/비격발 순차 관계와 주 흐름

동시 수행 관계 및 순차 관계의 분류에 따라 전체 기능/활동에 대한 위상 순서가 결정되면, 각 순차 관계에 대해서는 다음 기능/활동이 바로 시작되도록 하는지 여부에 따라 격발 순차 관계(triggering precedence relationship)와 비격발 순차 관계(non-triggering precedence relationship)를 결정한다. 격발 순차 관계는 하나의 기능/활동으로부터 출력 실체가 다른 기능/활동에 입력 또는 제어로 제공될 때, 실체가 도착하면서 곧바로 다음 기능/활동이 시작되는 경우로 정의하고, 실선으로 표현하며, 비격발 순차 관계는 입력이나 제어를 제공받는 기능/활동이 시작되기 이전에 실체가 도착해야 하는 경우로 정의하고 점선을 이용해서 표현하도록 한다[박찬권, 1996a]. 따라서 위상 순서가 결정된 후에, 한 기능/활동으로부터의 출력이 위상 순서에 있어서 바로 다음 계층(layer)에 속하거나 같은 계층(layer)에 속하는 기능/활동에 제공되는 경우는 격발 순차 관계로 정의하고, 하나 이상의 계층(layer)을 건너서 제공되는 경우는 비격발 순차 관계로 정의할 수 있다.

시뮬레이션에 의한 프로세스 수행도 분석과 같은 정량적 평가 과정에 들어가기 전에, 이렇게 해서 정의된 격발/비격발 순차 흐름을 통해서 설계된 프로세스에 대한 일차적 분석을 수행할 수 있다. 즉, 격발 순차 관계에 의해 나타나는 실체 흐름은 전체 프로세스의 수행 시간에 영향을 미치는 주 흐름(critical process flow)을 결정하게 된다. 그러므로, 격발 순차 흐름을 포함한 기능/활동이 프로세스 흐름에 있어 병목(bottle neck) 프로세스가 될 수 있고, 관련된 정보 및 물리적 실체와 메커니즘을 관리할 수 있다.

4.5 모형 변환 예제

<그림 11>은 하나의 로봇이 workcenter에 투입되는 부품과 workcenter에서 작업을 끝낸 부품에 대한 검사를 수



<그림 11> PROWD 모형으로의 변환 과정을 위한 기능 모형 예제

행하고, 각각 부품을 workcenter에 장착시키고 탈착시키는 작업에 대한 기능 모형을 작성한 예다. 여기서 장착과 탈착 작업은 동시에 이루어진다. 이 절에서는 <그림 11>의 예제를 중심으로 프로세스 흐름 모형으로의 변환 절차를 살펴보도록 한다.

4.5.1 활성 모드의 결정

먼저 다음과 같은 과정을 통해 기능/활동별로 공헌 행렬을 작성해서 활성 모드를 결정한다.

(1) 기능/활동 A^z에 대하여, 공헌 행렬 CM^z을 작성한다. 여기서 l개의 입력과 m개의 제어는 모두 행을 구성하고, n개의 출력은 열을 구성한다. cm^z_{ij}는 입력 또는 제어 i가 출력 j를 생성하는 데 직접적으로 기여하는 회수를 나타낸다.

(2) CM^z의 Rank를 구하면, 이것은 곧 기능/활동 A^z에 있어 각각의 입력 또는 제어의 부분 집합이 특성 출력 집합을 만들어 내는 조합의 수에 해당된다. 그리고 4.2절에서 설명한 알고리즘을 적용해서 각 활성 모드에 관여하는 입력, 제어, 그리고 출력의 조합을 구한다.

(3) 기능/활동이 몇 번 활성화되어야 이들 전체 입출력의 조합을 만들어 낼 수 있는가가 활성 모드의 수가 된다.

따라서, <그림 11>의 예제에서, 각 기능에 대한 공헌 행렬과 활성 모드는 각각

$$CM^1 = \begin{array}{c} \\ I1 \\ I2 \\ C1 \\ M1 \end{array} \begin{array}{c} \begin{array}{ccc} O1 & O2 & O3 \end{array} \\ \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{vmatrix} \end{array} \quad \mathrm{Rank}\langle CM^1 \rangle = 2$$

$$CM^2 = \begin{array}{c} \\ I3 \\ M1 \end{array}\begin{array}{c} \begin{array}{cc} O4 & O5 \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array} \quad \mathrm{Rank}\langle CM^2 \rangle = 1$$

$$CM^3 = \begin{array}{c} I4 \\ C2 \\ M1 \end{array}\begin{array}{c} O6 \\ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \end{array} \quad \mathrm{Rank}\langle CM^3 \rangle = 1$$

이 된다.

4.5.2 기능/활동의 분해에 의한 사이클의 제거

CM^1의 Rank는 2로, 두 개의 출력 조합이 존재함을 알 수 있다. 즉, CM^{10}과 CM^{11}는 각각

$$CM^{10} = \begin{array}{c} I1 \\ C1 \\ M1 \end{array}\begin{array}{c} O2\ O3 \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array} \quad \mathrm{Rank}\langle CM^{10} \rangle = 1$$

$$CM^{11} = \begin{array}{c} I2 \\ C1 \\ M1 \end{array}\begin{array}{c} O1\ O2 \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array} \quad \mathrm{Rank}\langle CM^{11} \rangle = 1$$

이 된다. 그리고 이들 입출력 조합은 동시에 활성화되지 않는다. 그러므로 <그림 11> 예제에서 기능/활동 A1은 2개의 활성 모드를 가짐으로써 사이클이 발생하는 복합 기능/활동의 경우에 해당된다. 공현 행렬 CM^{10}과 CM^{11}에 나타난 것처럼 각각의 활성 모드에 관여하는 입력, 제어, 그리고 출력의 조합에 따라 기능/활동을 분해(split)해서 표현하면, 사이클을 제거할 수 있다. 즉, I1과 C1, 그리고 M1이 출력 O2 또는 O3를 생성하고, I2와 C1, 그리고 M1이 O1 또는 O2를 생성한다. 물론, 이 경우는 관여하는 입력, 제어, 그리고 출력 사이의 관계가 비교적 간단하기 때문에 애초에 기능 모형을 작성하는 데 관여한 전문가가 곧바로 활성 모드의 수를 발견할 수 있다. 하지만 많은 수의 ICOM이 복잡하게 존재하는 경우는 공현 행렬을 이용해서 활성 모드를 결정할 수 있다. 결과적으로 <그림 11>의 예제에서 기능/활동 A1은 <그림 12>에서 보는 것처럼 A10과 A11로 분해될 수 있고, 사이클이 제거되었음을 발견할 수 있다.

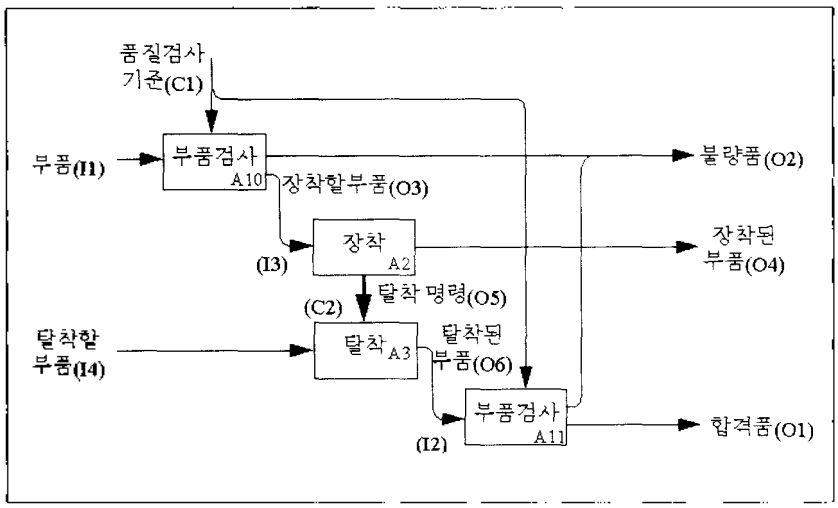
4.5.3 순환 반복 및 동시 수행 관계의 정의

기능/활동 A2에서 기능/활동 A3에 보내는 탈착 명령(C2)의 경우, 기능/활동 A3를 동시에 수행시키는 동기 제어에 해당되기 때문에 연결 값(indegree)을 ε ($0<\varepsilon<1$)으로 설정하고, 이중 실선으로 표현한다. 나머지 연결은 연결이 이루어진 두 기능/활동들 사이에서 순차 관계를 규정하는 것으로 해석되기 때문에 연결 값(indegree)을 1로 설정해준다.

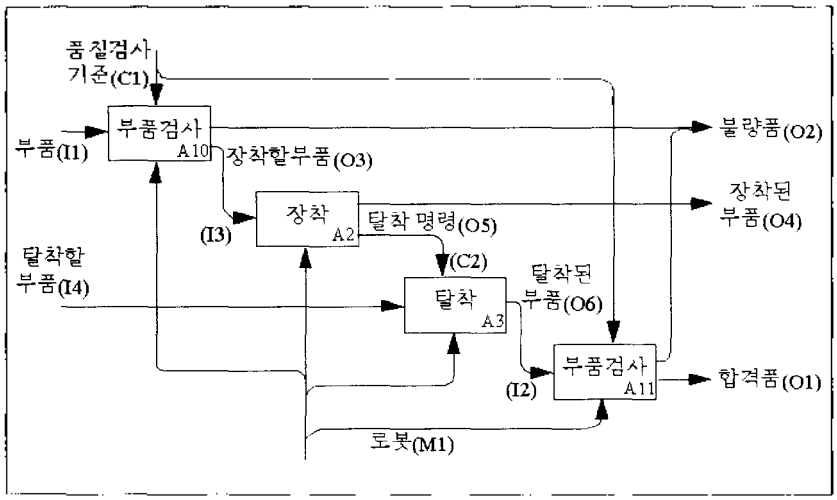
<그림 11>의 예제에는 순환 반복을 의미하는 연결이 포함되지 않았다. 하지만 순환 반복으로 분류되는 연결에 대해서는 연결 값(indegree)을 0으로 설정한다. 이들 연결 값(indegree)은 다음 단계에서 위상 순서를 결정하는 데 사용된다.

연결 의미에 대한 해석을 거쳐 <그림 12>의 모형은 <그림 13>과 같이 변환시킬 수 있다.

이후의 변환 과정에서 메커니즘은 영향을 미치지 않기 때문에, 본 예제에서는 편의를 위해 메커니즘에 대한 표현은 생략했다.



<그림 12> 동기 제어가 정의된 프로세스 흐름 모형



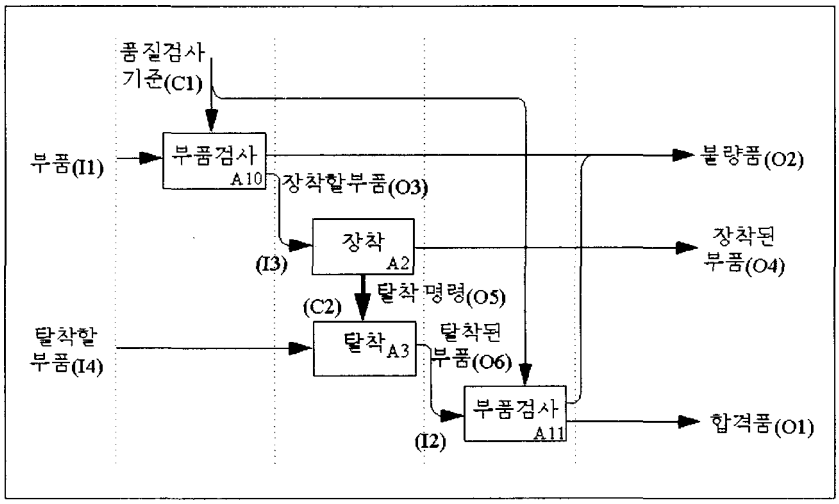
<그림 13> 기능/활동의 분해에 의한 사이클의 해체

4.5.4 위상 순서의 결정

활성 모드에 따른 기능/활동의 분해에 의해 사이클을 제거하고 화살표 유형 분류에 따라 연결 값(indegree)을 설정함으로써, 도출된 프로세스 흐름 모형에서는 기능/활동들 사이의 전체적인 순차 및 동시 수행 관계를 고려한 위상 순서를 결정할 수 있다. 즉, 기능/활동을 노드로 하고 변

환된 순차 관계를 호로 하는 네트워크를 구성해서 변형 횡단 탐색 (MBFS : Modified Breadth First Search) 알고리즘[박찬권, 1996]을 적용함으로써 위상 순서를 구한다. 이렇게 구성된 네트워크는 그 자체로 전체 프로세스의 흐름을 표현하기 때문에, 기능/활동의 수행 순서에 따른 객체의 흐름이나 제어의 흐름을 파악할 수 있다.

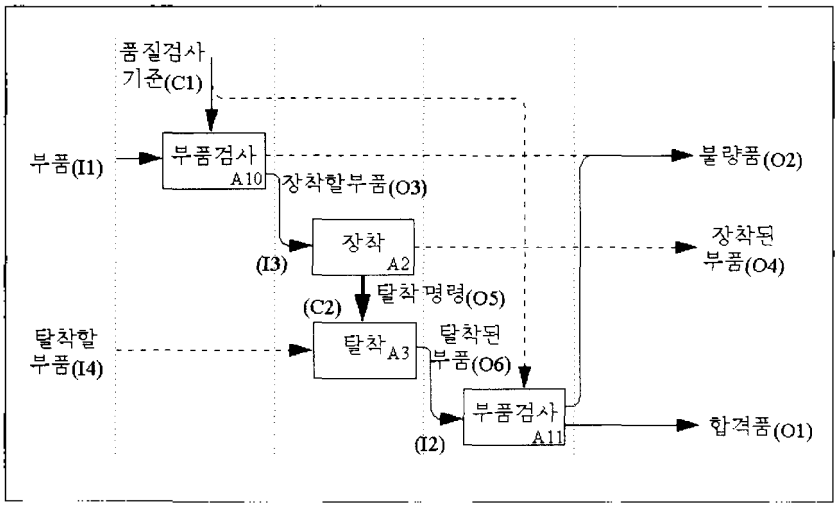
<그림 14>는 이와 같은 과정을 적용해서 구성된 프로세스 흐름 모형이다. 탈착 명령(C2)은 장착과 동시에 탈착의 수행을 알리는 동기 제어(synchronization control)이기 때문에 A2와 A3은 이중 실선에 의해 연결되고, 위상 순서에 있어 같은 계층(layer)에 속함으로써 동시 수행이 가능한 기능/활동임을 나타낸다.



<그림 14> 위상 순서가 결정된 프로세스 흐름 모형

4.5.5 격발/비격발 순차 관계의 결정

동시 수행 관계 및 순차 관계의 분류에 따라 전체 기능/활동에 대한 위상 순서가 결정되면, 각 순차 관계에 대해서는 다음 기능/활동을 바로 시작시키는지 여부에 따라 격발 순차 관계와 비격발 순차 관계를 정의하게 된다. 따라서 위상 순서가 결정된 <그림 14>와 같은 프로세스 흐름 모형은 바로 인접하는 계층(layer) 사이에 속한 기능/활동 사이의 실체 흐름에 대해서는 격발 순차 흐름으로 정의하고, 인접하지 않은 계층에 속하는 기능/활동 사이의 실체 흐름은 비격발 실체 흐름으로 정의한다. 모형의 외부, 즉 부모 계층(hierarchy)으로부터의 실체 흐름의 경우, 첫 번째 계층(layer)으로 입력되는 실체 흐름과 마지막 계층(layer)으로부터 제공되는 출력 실체 흐름은 격발 실체 흐름으로 정의하고, 그 이외의 실체 흐름은 비격발 실체 흐름으로 정의한다. 따라서 <그림 14>로부터 <그림 15>와 같은 결과를 얻게 된다.

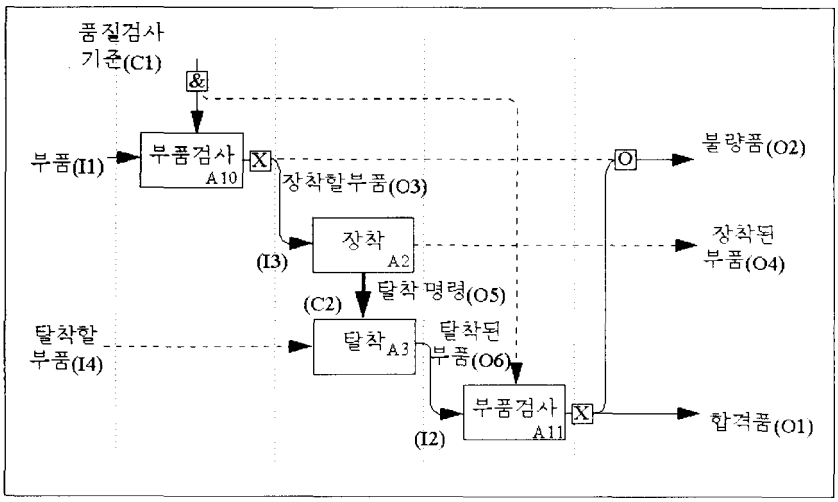


<그림 15> 격발/비격발 순차 흐름이 결정된 프로세스 흐름 모형

이들 격발 순차 관계는 위상 순서를 기준으로 결정되기 때문에 궁극적으로는 전체 기능/활동의 수행 순서에 있어 주 흐름(critical process flow)을 결정하게 된다.

4.5.6 접속의 정의와 자료 구조

IDEF0 기능 모형을 입력받아서, 사이클 해체, 순차 및 동시 수행 관계의 정의, 위상 순서의 결정, 격발/비격발 순차 관계



<그림 16> 변환된 PROWD 모형

<표 5> 기능/활동에 대한 템플리트

Function Template				
identifier	A10	A2	A3	A11
description	부품검사	장착	탈착	부품검사
input	I1	I3	I4	I2
control	C1	-	C2	C1
in./ctl. logic	AND	-	AND	AND
mechanism	M1	M1	M1	M1
output	O2, O3	O4, O5	O6	O1, O2
output logic	XOR	AND	-	XOR
layer	1	2	2	3

를 바탕으로 한 주 흐름(critical process flow)의 결정 과정을 거쳐서, 마지막으로 연결의 유형에 따라 접속의 논리적 관계를 분류해서 정의함으로써 <그림 16>과 같은 프로세스 흐름 모형을 도출할 수 있다. 접속에 대한 논리적 관계 또는 규칙은 접속의 속성(attribute)으로 표현한다.

최종적으로 변환된 <그림 16>의 프로세스 흐름 모형을 저장하기 위한 모형화 도구들의 자료 구조는 <표 5>, <표 6>, 그리고 <표 7>과 같이 정의된다.

<표 6> 접속에 대한 템플리트

Junction Template				
identifier	J1	J2	J3	J4
logic	XOR	XOR	OR	AND
type	fan-out	fan-out	fan-in	fan-out
rule	품질	품질	--	

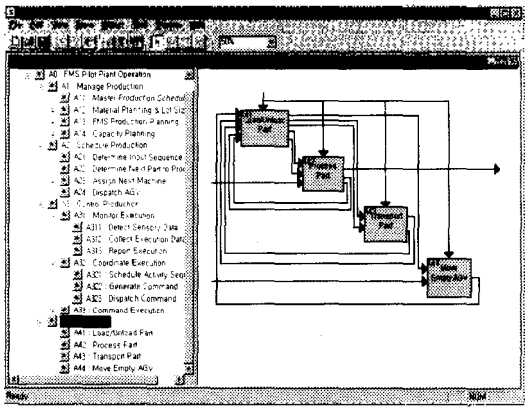
5. 시스템의 구현과 적용

4장에서 소개된 방법을 통해 IDEF0 기능 모형으로부터 PROWD 프로세스 흐름 모형변환을 지원하는 시스템을 구현하였다. 지원 시스템은 마이크로소프트 윈도우 환경에서 Visual C++을 이용하였다. 구현된 시스템을 적용하기 위한 예제([그림 17])는 서울대학교 자동화시스템공동연구소(Automation and Systems Research InstituteI)에 설치된 FMS를 운용하는 과정에 대한 IDEF0 기능 모형의 일부이다[박찬권,1996b].

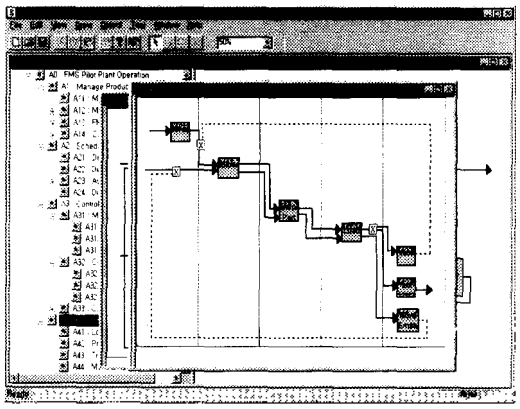
<표 7> 실체 흐름에 대한 템플리트

Link Template												
identifier		L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11
description		부품	탈착할 부품	불량품	장착할 부품	장착된 부품	탈착된 부품	불량품	합격품	품질검사기준	품질검사기준	탈착명령
격발 유형		TE	NTE	NTE	TE	NTE	TE	TE	TE	TE	NTE	TE
tail	기능	system	system	A10/J1	A10/J1	A2	A3	A11/J2	A11/J2	system	system	A2
	type	-	-	output	output	output	output	output	output			output
	id.	-		O2	O3	O4	O5	O2	O1	-		O5
head	기능	A10	A3	system	A2	system	A11	system	system	A10	A11	A3
	type	input	input	-	input	-	input	-	-	control	control	control
	id.	I1	I4	-	I3	-	I2	-		C1	C1	C2

[그림 17]과 같이 입력된 IDEF0 기능 모형은 앞에서 설명한 과정을 거쳐 [그림 18]과 같은 PROWD 모형으로 변환된다. 지원 시스템에서는, 이후에 추가적으로 사용자 정의 정보를 입력받아서 시뮬레이션 코드를 생성시키도록 하고 있다[박찬권, 1996b].



[그림 17] IDEF0 기능 모형 예제



[그림 18] 변환된 PROWD 모형

6. 결론 및 추후 연구 방향

본 연구에서는 시스템 개발 과정에서 요구되는 다양한 관점에서의 모형들을 통합하고 연계시키기 위한 틀(framework)로서 기능 모형으로부터 프로세스 흐름 모형을 작성하는 방안을 제시하였다. CALS의 표준 방법론으로서 컴퓨터통합생산시스템을 비롯한 시스템 설계에 포괄적으로 이용되고 있는 IDEF0 기능 모형을 입력으로하여 변환에 필요한 활성 모드, 사이클의 해체, 그리고 위상 순서의 개념 등을 정의하였다. 이로써 초기 설계 과정에서 산출된 기능 모형을 토대로 체계적인 프로세스 흐름 모형의 작성이 가능하고, 따라서 전체 시스템 설계에 요구되는 모형 개발 비용의 절감과 모형들의 일관성 확보가 용이해질 수 있을 것으로 기대된다.

변환된 프로세스 흐름 모형에는 객체의 흐름과 더불어 시스템에서 수행되는 기능의 순서 관계가 묘사되고 있어서, 동시 수행 가능한 프로세스의 규명과 프로세스에 대한 주 흐름(critical process)의 정의를 통해 전체 프로세스의 수행 시간이 단축되도록 프로세스의 개선을 기할 수 있다. 이와 더불어 보다 정량적인 시스템 성능 평가와 분석을 위해 변환된 프로세스 흐름 모형으로부터 시뮬레이션을 수행을 위한 사양을 도출할 수 있다.

한편, 프로세스 흐름 모형에 표현되는 객체의 흐름은 그 사제로 객체 상태

전이를 묘사하고 있기 때문에, 이를 바탕으로 시스템 설계 과정에 필수적인 정보 모형과의 통합 방안이나 소프트웨어의 사양 설계에 관한 연구도 필요하리라 생각된다.

참고문헌

[박찬권 등 96a] 박찬권, 박남규, 박진우, CIM 시스템 설계를 위한 모형변환에 관한 연구, 제어계측신기술연구센터 '96년도 참여교수 동계 워크샵, 서울대학교 한국과학재단 제어계측신기술연구센터, 1996.1

[박찬권, 96b] 컴퓨터 통합 생산을 위한 프로세스 설계 지원 시스템에 대한 연구, 서울대학교, 박사학위 논문, 1996. 8.

[신가태, 95] 제조 데이터베이스 구축을 위한 개념설계 지원 시스템에 관한 연구, 서울대학교, 박사학위논문, 1995. 2.

[AMICE, 88] Open System Architecture for CIM, *ESPRIT Research Reports*, Vol. 1, 1989

[Banerjee, 88] "A Structured Approach to FMS Modelling," *Int. J. of CIM*, Vol. 1, No. 2, pp77-88, 1988

[Boucher 등, 92] Boucher, T. O. and M. A. Jafari, "Design of a Factory Floor Sequence Controller from a High Level System Specification," *J. of Mfg. Sys.*, Vol. 11, No. 1, pp 401 417, 1992

[Busby 등, 93] Busby, J. S. and G. M. Williams, "The Value and Limitations of Using Process Models to Describe the Manufacturing Organization," *Int. J. of Prod. Res.*, Vol. 31, No. 9, pp2179-2194, 1993

[Chaillet 등, 93] Chaillet, A., M. Courvoisier, M. Combacau, and A. D. Bonneval, "Merging Petri Nets and Database Models for Control and Monitoring Requirements in FMS," *Proc. of IEEE Int. Conf. on SMC*, pp42-27, 1993

[Curtis 등, 92] Curtis, B., M. I. Kellner and J. Over, "Process Modeling," *Communications of the ACM*, Vol. 35, pp75-90, 1992

[NIST, 1993] Draft Federal Information Processing Standards Publication 183 : Integration Definition for Function Modeling(IDEF0), FIPS Waiver Decisions, Dec. 1993

[Engelke, 85] Engelke,H. et.al., "Integrated Manufacturing Modeling System," *IBM J. Res. Develop.*, Vol. 29, No. 4, pp343 355, 1985

[Hsu 등, 95] Hsu, C., J.Cho, L.Yee and L.Rattner, "Core Information Model: A Practical Solution to Costly Integration Problems," *Computers and IE*, Vol. 28, No. 3, pp523-544, 1995

[Kim 등, 93] Kim, C., K. Kim and I.Choi, "An Object -oriented Information Modeling Methodology for Manufacturing Information Systems," *Computers and IE*, Vol. 24, No. 3, pp337-353, 1993

[Lewis 등, 87] Lewis, W. M. M. Barash and J. J. Solberg, "Computer Integrated Manufacturing System Control: A Data Flow Approach," *J. of Manufacturing Systems*, Vol. 6, No. 3, pp177-191, 1987

[Liang 등, 94] Liang, G.R. and H.M.Hong, "Hierarchy Transformation Method for Repetitive Manufacturing System Specification, Design, Verification and Implementation," *CIMS*, Vol. 7, No. 3, pp191-205, 1994

[Malhotra 등, 92] Malhotra, R. and S. Jayaraman, "An Integrated Framework for Enterprise Modeling," *J. of Mfg. Sys.*, Vol. 11, No. 6, pp426-441, 1992

[Mandel, 90] Mandel, K., Graphical Process Description-Views and Diagrams," *Int. J. CIM*, Vol. 3, No. 5, pp314-327, 1990

[Mayer, 94] Mayer,R.J., *IDEF0 Function Modeling*, College Station, TX 77840-2335, KBSI, 1994

[Mujtaba, 94] M.S.Mujtaba, "Simulation Modelling of a Manufacturing Enterprise with Complex Material, Information and Control Flows," *Int. J. CIM*, Vol. 7, No. 1, pp29-46, 1994

[Pinci 등, 93] Pinci, V.O., R.M. Shapiro, "An Integrated Software Development Methodology Based on Hierarchical Colored Petri Nets," *Lecture Notes in Computer Science, Advances in Petri Nets*, Vol. 524, pp542-547, 1993

[Ross, 77] "Structures Analysis(SA) : A Language for Communicating Ideas," *IEEE Tr. on Software Engineering*, Vol. 3, No. 1, pp16-34, 1977

[Vernadat 등, 93] Vernadat, F. and M.Zelm, "Advanced Modeling Approach to CIM Systems," *Advances in Factories of the Future, CIM and Robotics*, M. Cotsaftis and F. Vernadat(Eds), pp77-89, Elsevier, 1993

[Wang 등, 93] Wang, W., K.Popplewell and R. Bell, "An Integrated Multi-view Systems Description Approach to Approximate Factory Modelling, *Int. J. CIM*, Vol. 6, No. 3, pp165-174, 1993

[Warren 등, 92] Warren, J. R., J.W.Stott and A. F. Norcio, "Stochastic Simulation of Information Systems Designs from Data Flow Diagram," *J. of Systems and Software*, Vol. 18, pp191-199, 1992

[Williams, 94a] Williams, T.J., " The Purdue Enterprise Reference Architecture," *Computers in Industry*, Vol. 24, No. 2-3, pp141-158, 1994

[Williams, 94b] Williams, T.J. et al., "Architectures for Integrating Manufacturing Activities and Enterprises," *Computers in Industry*, Vol. 24, No. 2-3, pp111-139, 1994

[Young 등, 91] Young, R. E. and J. Besterager, " An Approach to CIM System Development whereby Manufacturing People Can Design and Build Their Own CIM Systems," *Int. J. of CIM*, Vol. 4, No. 5, pp288-299, 1991

[Zaytoon 등, 94] Zaytoon, J., E. Niel and A.Mille, " A Functional-to-behavioural Transformation Framework for Specification and Design of Manufacturing Systems," *CIMS*, Vol. 7, No. 4, pp247 258, 1994

저자소개

박찬권

서울대학교 공과대학 산업공학과 졸업(학사, 석사, 공학박사)
금성정보통신 근무, 현재 영산국제산업대학 전임강사

신기태

서울대학교 공과대학 산업공학과 졸업(학사, 석사, 공학박사)
현재 대진대학교 전임강사

박남규

서울대학교 공과대학 산업공학과 졸업(학사, 석사, 공학박사)
Purdue University 연구원, 동신대학교 산업공학과 전임강사 역임
현재 생산기술연구원에서 CALS 및 IMS(지적 생산시스템) 분야 연구

박진우

서울대학교 공과대학 산업공학과 졸업(학사)
한국과학기술원(KIST) 졸업(석사)
University of California at Berkeley 졸업(공학박사)
현대양행 생산관리부 근무, 현재 서울대학교 교수