

비즈니스 프로세스 모델링 연계 방법론: IDEF0, IDEF3, Petri Net

임동순*, 김철한**, 우훈식**, 김종인***

Integration of Business Process Modeling Methodologies: IDEF0, IDEF3, and Petri Net

Dong-Soon Yim, Cheol-han Kim, Hoon-Shik Woo, Joong-In Kim

Abstract

In order to achieve a successful introduction of CALS, an enterprise model which is a concise description of what an enterprise does is required. The enterprise model mainly consists of a process model and a data model. The process model describes processes that create, change, combine, or destroy the entities within the enterprise. Several process modeling methodologies have been proposed. Each modeling methodology requires its own modeling view point, elements, and syntax. In developing a process model for CALS, these models created at different view points are required to analyze and design a system in a broad view. This paper aims at proposing an integration methodology for a process model. Specifically, IDEF0, IDEF3, and Petri Net are considered to be integrated. An IDEF0 model describing static functions of enterprise is transformed to an IDEF3 model describing behaviour of a system with additional information. Also, the IDEF3 model is transformed to a Petri Net model. These transformations will be automatically accomplished once the additional information for the transformation is provided.

* 한남대학교 산업공학과

** 대전대학교 산업공학과

*** 홍익대학교 경영정보학과

1. 서론

CALS를 성공적으로 수행하기 위하여는 기본적으로 기업들이 무슨 일을 어떻게 하는지에 대해 함축적인 묘사를 한 기업모델(enterprise model)이 필요하게 된다. 기업모델은 주로 그래픽 도구나 구조적인 언어를 사용하여 표현되며, 다양한 종류의 모델 묘사 방법을 사용할 수 있다. 기업 통합을 목적으로 하는 경우, 기업모델은 주로 프로세스 모델과 데이터 모델로 구성된다[Eirich 1992].

프로세스는 공통의 목적을 수행하는 행위들의 집합으로 정적인 프로세스 모델과 동적인 프로세스 모델로 나눌 수 있다. 동적인 프로세스 모델은 다시 실행 불가능한 모델과 실행 가능한 모델로 나뉜다. 정적인 프로세스 모델은 시간의 개념이 없는 단지, 기능과 그 기능들의 활동에 필요한 조건을 묘사한 것이다. 반면에 동적인 프로세스 모델은 시간에 따른 프로세스의 활동과정을 묘사한 것으로 언제 어떤 조건하에서 각 프로세스가 활동하는지를 묘사한다. 실행 가능한 동적 모델은 모델링 작업에 많은 노력이 요구되지만, 시뮬레이션을 통하여 기업 활동을 예측, 분석할 수 있는 장점이 있다.

또한, 프로세스 모델은 대상으로 하는 프로세스의 성격에 따라 비즈니스 프로세스, 정보 프로세스, 물류 프로세스 모델로 나눌 수 있다[Medina-Mora et al. 1993]. 정보 프로세스 모델은 정보처리 기능과 정보의 흐름을 파악하는 것으로 주로 정보 시스템의 설계를 목적으로 한다. 물류 프로세스

모델은 물리적인 객체들에 대한 처리 순서와 이 객체들의 흐름을 파악하는 것을 목적으로 한다. 한편, 비즈니스 프로세스 모델은 정보와 물류 프로세스를 포함하는, 보다 넓은 의미의 프로세스 모델이다. 비즈니스 프로세스 모델의 이용은 그 함축성의 정도에 따라 다양하다. 가장 보편적으로는 비즈니스 프로세스의 재설계 및 분석에 효과적으로 이용된다. CALS, ERP, CIM등과 같은 정보 위주의 시스템을 구축하는데 있어 비즈니스 프로세스의 재설계는 필수적이다. 불필요한 프로세스를 제거하고, 비 효율적인 프로세스를 개선하여 효과적인 정보 위주의 시스템을 성공적으로 구축, 운영할 수 있기 때문이다. 뿐만 아니라 정보 위주 시스템의 개념설계 단계에서도 효과적으로 이용된다.

본 연구에서는 CALS를 도입하기 위하여 기업이나 조직에서 수행하는 다양한 종류의 프로세스 모델들을 통합할 수 있는 방법론을 개발하는 것을 목적으로 한다. 대부분의 프로세스 모델은 정적인 기능 모델링에서 출발한다. 그 이유는 기능 모델링이 모델링 작업의 초기에 쉽게 수행될 수 있기 때문이다. 그러나, 정적인 기능 모델과 동적인 프로세스 모델은 관점의 차이로 인하여 모델 자체가 상이하다. 동적인 프로세스 모델은 기능 모델에 부가적으로 각 프로세스의 수행순서 및 수행조건을 명세를 필요로 한다. 실행 가능한 프로세스 모델을 작성하기 위하여는 더욱 복잡한 절차가 요구된다. 본 연구에서는 이러한 다양한 관점에서의 프로세스 모델링 방법들을 통합할 수 있는 방법론을 개발한다. 모델의 통합은

각 모델들이 서로 독립적으로 작성되고, 분석되어지는 것이 아니라 서로 연관된 모델들로 구성되어 한 모델에서 다른 모델로 쉽게 변형 될 수 있는 모델의 일관성을 유지할 수 있음을 의미한다.

프로세스 모델의 통합을 위하여 세가지 중요한 프로세스 모델링 방법인 IDEF0, IDEF3, 그리고, Petri Net를 대상으로 한다. 이들 모델들의 점진적인 변환을 통하여 모델의 연계성을 이루도록 한다. 우선, 기능 모델링을 IDEF0로 수행하는 방법론을 채택하여, 기능 모델은 부가적인 자료를 제공받아 IDEF3로 변환된다. IDEF3는 변환에 필요한 부가적인 자료를 바탕으로 실행 가능한 동적인 프로세스 모델인 Petri Net로 변환된다. 즉, 기능 모델링에서 출발하여 동적 프로세스 모델로 변환되도록 한다. 이때 변환은 가능한 자동적으로 수행되도록 하고, 필요한 경우, 변환에 요구되는 부가적인 자료를 모델링 작업자로 하여금 제공받도록 한다.

2. 관련 연구

기능과 동적 프로세스 모델의 통합에 관한 연구들은 주로 기능모델과 동적모델의 연계, 기능모델에 동적인 요소의 추가, 그리고, 기능모델과 동적모델의 합성을 제안하는 방향으로 추진되어 왔다.

기능모델과 동적모델의 연계를 제안하는 연구에서는 대표적인 기능적 모델링 방법인 IDEF0 또는 SADT[Ross 1977], OFM(Object Flow Diagram)[Delcambre et al. 1990] 등으로 작성된 모델을 SLAMII 코드,

SIMAN 코드 등의 시뮬레이션 코드로 변환하는 노력을 시도 하였다[Gay et al. 1991, Young et al. 1991]. 그러나, IDEF0와 시뮬레이션 모델의 구조적인 차이와 IDEF0의 동적인 묘사의 부족으로 두 모델간의 정확한 변환은 많은 노력과 시간을 요구한다. 기능 모델을 시뮬레이션 코드 대신에 Petri Net 모델과 연계 시키는 사도들은 두 모델간의 변환을 용이하게 하는 목적으로 수행되었다. 즉, IDEF0 또는 SADT로 기능 모델링을 수행하고 이를 바탕으로 Petri Net 모델로의 변환을 제안하였다[Pinch et al. 1991, Boucher et al. 1992, Zaytoon et al. 1994]. 이상의 기능-Petri Net 모델의 연계에 관한 연구들은 대체적으로 기능 모델이 활동(activity) 중심으로 표현된다는 점에 착안해서, 활동을 Petri Net의 트랜지션과 플레이스의 적당한 조합으로 변환하고, 기타 기능 모델을 구성하는 요소들을 아크로 표현함으로써 변환의 메커니즘을 제공하고 있다. 이들 연구는 기능 모델이 가지는 한계 즉, 정량적 분석이 뒷받침되지 못한다는 점을 보완하고는 있지만, 요구 정의 단계에서의 정성적 특성을 반영하는 기능 모델을 정량적이고 수리적인 특성을 가지는 Petri Net에 직접 연계 시킴으로써 사실상 기능 모델의 사용에 많은 제약을 가하고 있다. 그 밖에 IDEF0모델을 PROWD(Process Flow Description) 모델로 변환하고 다시 이를 SIMAN 시뮬레이션 코드로 변환한 연구가 있다[박 1996].

기능 모델에 동적인 요소를 추가하는 연구로는 DFD(Data Flow Diagram)와 SSAD(Structured Systems Analysis and Design)방법론에 동적인 면을 부가한 DSADE(Dynamic

Systems Analysis, Design and Evaluation) [Eddins et al. 1991]를 제안하거나, IDEF0의 기능에 동적인 묘사를 추가[Malhotra et al. 1992]한 연구 등이 있다.

기능모델과 동적모델의 합성에 대한 연구로는 상위계층에 대해서는 IDEF0로 모델을 작성하고, 최하위 계층에 속하는 부분은 IDEF3로 모델링 함으로써 IDEF0/IDEF3의 복합적 모델링을 제안한 방법론들이 있다 [Rensburg et al. 1995, Colquhoun et al. 1996].

3. IDEF0, IDEF3, Petri Net의 모델링 관점 비교

IDEF0[Ross 1977, Gay 1991, FIPS-183 1993] 기능 모델링 방법론은 기업이나 조직의 시스템이 어떤 일을 하는가에 대한 기능적인 관점에서 그래픽 표현방법을 사용하여 모델을 작성토록 하는 도구이다. IDEF0는 기업이나 조직의 활동을 추상적인 단위의 기능으로 표현토록 하고, 각 기능간에 연관된 정보, 물류, 자원을 기능 모델 안에 표현 할 수 있도록 하는 절차와 언어를 제공한다. IDEF0 모델 다이어그램의 기본 요소는 박스 형태로 표시되는 기능(function)과 화살표로 표시되는 ICOM(Inputs, Outputs, Controls, Mechanisms)으로 구성되어 있다. 기능은 시스템의 자원에 의해 수행되는 활동이나 행위를 표현하며, 어떠한 상황에서 무엇이 수행되는지를 서술한다. ICOM은 기능이 수행되는데 필요하거나, 혹은 기능간에 관여된 정보, 물류, 자원 등의 객체들을 의미한다. IDEF0 도구에서는 시간적인 측면에서의 모델링을 포함하고

있지 않기 때문에, 동적으로 수행되는 행위를 묘사하는 기능들이 IDEF0로 묘사됐을 때 동적 모델에서 요구되는 특성을 표현하지 못한다[Department of Defense 1993].

IDEF3[Mayer et al. 1995, Gay 1991]는 활동이나 행위를 시간적인 측면에서 순차적인 형태로 표현하기 위한 모델링 도구이다. IDEF3에는 두 가지 모델화 방법인 프로세스 흐름 묘사(Process Flow Description)와 객체 상태 전이 묘사(Object State Transition Description)가 제공된다. 프로세스 흐름 묘사는 시스템 내의 행위들이 어떠한 조건과 순서에 의해 수행되는지를 표현하고, 객체 상태 전이 묘사는 객체가 특정 프로세스를 통과할 경우 발생 가능한 전환 상태를 표현한다. 본 연구에서는 IDEF3의 프로세스 흐름 묘사만을 대상으로 한다. 프로세스 흐름 다이어그램의 주된 구성성분은 행동단위(UOB: Units of Behaviour), 연결(link), 접속(junction)이다. UOB는 박스 형태로 표현되며, 시스템내의 기능, 활동, 행위, 운용, 의사결정 등을 의미한다. UOB간의 관계는 연결과 접속을 통하여 이루어진다. 연결은 두 UOB간을 잇는 화살표로 표현되며, UOB간의 선후 연관관계를 나타내는 단순선행, 제한된 선행, 관계연결로 구성되어 있다. 접속은 UOB 다이어그램에서 동기적(synchronous) 또는 비동기적(asynchronous) 진행관계를 포함하는 AND, OR, XOR의 논리적 의미를 지원한다.

Petri Net은 비동기성(asynchronous)과 동시성(concurrency)의 성격을 가진 시스템을 잘 표현하는 모델링 도구이다[Peterson 1981]. Petri Net는 함축적으로 표현될 수 있

을 뿐만 아니라 컴퓨터를 이용하여 실행될 수도 있다. 실행을 위해서 토큰(token)이라는 모델링 객체를 부가적으로 포함할 수 있다. 기본적인 Petri Net는 방향을 가진 아크로 연결되는 두 가지 형태의 노드인 플레이스(place)와 트랜지션(transition)으로 구성된 그래프이다. 시스템의 상태, 행위 등을 플레이스로 나타내고, 상태, 행위의 변화를 야기하는 사건을 트랜지션으로 나타낸다. 토큰들이 어느 노드들에 있는가에 따라 어느 시점에서의 시스템 상태를 나타낼 수 있다. 그래픽 표현 방법으로는 플레이스를 원으로, 트랜지션을 상자로, 아크(arc)를 화살표가 있는 선으로, 그리고, 토큰을 작은 원으로 표현한다. 노드들에 있는 토큰들은 트랜지션 활성화(enabling) 조건과 발생(firing) 조건에 의해 소멸되고, 새로운 토큰이 다른 노드에 생성되는 실행 규칙을 갖는다. 이러한 Petri Net의 실행규칙은 네트의 변화를 결정하여 시스템의 변화를 파악할 수 있는 장점을 제공한다.

지금까지 설명된 3가지 모델링 방법론은 모델링 작업자로 하여금 서로 다른 관점에서 모델을 생성하도록 한다. IDEF0 모델은 기업 프로세스의 기능이 무엇인지를 분해(decomposition)에 의해 파악하도록 한다. 기능은 정적인 의미를 가지고, 언제, 어떻게 수행이 이루어 지는지에 대해 설명하지 않는다. 단지, 기능의 수행에 필요한 조건이 무엇인지만을 설명한다. 때문에 모델링의 초기에 높은 추상화 수준에서 프로세스를 파악할 수 있는 장점을 제공한다. IDEF3는 정적인 기능보다는 동적인 행위들의 수행에 초점을 맞추어 동적인 프로세스를 파악

할 수 있도록 한다. 즉, 행위들이 언제, 어떻게 수행될 수 있는지를 설명한다. IDEF0와 IDEF3모델은 기능 또는 행위들의 수행 순서에 대한 정보를 일부 가지고 있다. IDEF0에서는 기능 A의 수행에 필요한 조건이 기능 B의 수행결과 발생할 수 있는 경우 이를 반영한다. 즉, 기능 B가 수행된 후에 기능 A가 수행될 수 있음을 의미한다. 그러나, IDEF0에서는 기능들의 수행에 동시성, 동기성, 자원의 공유 등이 요구되는 동적인 구조를 표현하지 못한다. IDEF3에서는 선행 연결이 행위들의 수행순서를 포함하는 의미를 갖는다. 선행연결이 두 UOB를 연결하고 있을 때 한 행위의 수행이 끝나야 다음 행위의 수행이 시작될 수 있음을 의미한다. 또한, 동기성, 동시성 등을 표현할 수 있는 접속을 제공한다. 그러나, IDEF0에서의 같이 제한된 자원을 공유하는 경우 이를 표현하지 못한다. 만약, 기능 또는 행위의 수행에 필요한 조건 중에 제한된 용량을 갖는 자원이 매우 중요한 역할을 하는 경우, 자원이 유효하고, 기능 또는 행위에 자원이 할당되어야만 수행이 가능하다. Petri Net는 IDEF0와 IDEF3에 비해 프로세스의 흐름을 매우 자세하게 묘사한다. 동기성, 동시성 뿐만 아니라 자원의 공유에서 비롯되는 기능 또는 행위들간의 충돌을 묘사토록 한다. 기본적인 Petri Net의 구성요소는 매우 단순하다. 단 세가지 구성성분으로 프로세스의 구조를 묘사한다. 때문에, 복잡한 시스템을 모델링 하기 위하여는 Petri Net에 대한 지식과 많은 모델링 경험이 요구될 뿐 만 아니라 복잡한 모델이 생성될 수 있어 IDEF0와 IDEF3가 가지는

모델의 단순함과 의사소통의 수단으로서의 장점을 가질 수 없다.

[표 1]은 IDEF0, IDEF3, Petri Net의 모델링 관점, 모델링 대상, 모델링 구성 성분들이 어떻게 다른지를 요약, 설명한다.

4. 프로세스 모델링 연계

이미 설명된 바와 같이 IDEF0, IDEF3, Petri Net는 기업이나 조직의 프로세스를 나타내는 표현 방법의 차이와 모델링 관점의 상이함으로 인하여 동일한 프로세스를 서로 다른 성격의 모델로 나타낸다. 즉, 실제 세계에서의 프로세스를 모델링 할 때 실재는 하나의 프로세스 이지만 관점의 차이에 따라 다양한 모델이 생성된다. 만약, 모든 관점을 포함하는 하나의 모델링 방법론이 존재 한다면 바람직하지만, 복잡한 실세계를 하나의 모델로 표현하는 것은 불가능하다. 현존하는 모델링 방법론들은 그들 나름의

독특한 모델링 관점을 요구한다. 기업통합과 같은 목적을 위해서는 다양한 모델링 방법이 필요케 되어 모델링 작업자에게는 많은 부담을 준다.

본 연구에서는 모델링 통합 방법론으로 한 모델에서 다른 모델로의 변환을 용이하게 할 수 있는 방법론을 채택한다. 새로운 관점에서 모델을 처음부터 작성하기 보다는 이미 작성된 기존의 모델을 바탕으로, 새로운 관점에 알맞게 모델이 변환될 수 있다면 이는 모델링 작업자에게 부담을 덜어준다. 이는 기능모델과 동적 모델의 연계를 제안한 기존의 연구들과 유사한 배경을 가지나, IDEF0로 부터 IDEF3, Petri Net으로의 점진적 변환과 한 모델로 부터의 직접적인 변환 보다는 필요한 부가자료를 제공받아 유효한 다른 모델로의 변환이라는 특징을 갖는다. 서로 다른 관점에서의 두 모델간에 변환 가능성은 한 모델이 다른 모

[표 1] IDEF0, IDEF3, Petri Net의 모델링 관점 비교

구분	IDEF0	IDEF3 (Process description)	Petri Net
관점	<ul style="list-style-type: none"> - 프로세스는 어떤 기능들로 구성되어 있는가? - 각 기능들의 활동에 필요한 엔터티와 기능의 수행결과 발생하는 엔터티는 무엇인가 	<ul style="list-style-type: none"> - 한 프로세스는 어떤 행위들로 구성되어 있는가? - 각 행위들간의 시간적인 관계는 무엇인가? - 각 행위들간의 객체 흐름은 어떤 순서로 수행되는가? 	<ul style="list-style-type: none"> - 기능이나 행위의 활동은 어떤 상태와 사건으로 구성되는가? - 각 사건들이 발생할 수 있는 조건은 시스템의 어떤 상태에 의존하는가?
모델링 대상	기능, 조건, 엔터티의 흐름	단위 행위, 단위 행위들간의 시간적 관계	활동의 상태 및 사건, 사건 발생 조건
모델링 성분	기능, ICOM	UOB, Link, Junction	Place, Transition, Arc, Token

델로 변환될 수 있는 정보를 모두 포함하고 있다는 가정하에서 가능하다. 그러나, 본 연구에서 다루는 IDEF0, IDEF3, Petri Net 모델들은 서로의 변환을 위한 필요한 정보를 모두 포함하고 있지 않다. 때문에, 한 모델에서 다른 모델로의 변환을 하기 위하여는 변환에 필요한 부가적인 자료가 제공되어야 한다.

4.1 IDEF0/IDEF3 변환 방법

본 연구에서는 IDEF0를 IDEF3의 프로세스 흐름 묘사 모델로 변환하는 것을 대상으로 한다. IDEF3의 객체 상태 전이 다이어그램 모델은 본 연구에서 제외한다.

4.1.1 추상화 수준(Abstraction level)

IDEF0 와 IDEF3의 근본적인 차이는 각 기능 또는 행위의 활동에 있어 시간적인 제한조건을 표현 여부이다. IDEF3는 시간적인 제한조건을 포함하기 때문에 보다 자세한 모델이 요구된다. 반면, IDEF0는 시간적인 제한이 포함되지 않고 기능의 활동에 필요한 엔터티(정보, 물류, 자원 등)가 무엇인지만을 표현하므로, 기능의 분할이라는 분석방법을 이용하여 다양한 추상화 수준을 추구할 수 있다. 행위들의 시간적인 제한조건을 도출하기 위하여는 보다 자세한 수준에서의 단위 행위들을 대상으로 하여 이 행위간의 상호 작용을 분석하여야만 한다. 본 연구에서는 IDEF0를 IDEF3로 변환하기 위하여 가능한 자세하게 분할된 기능을 표현하는 최하위 수준에서의 IDEF0 모델을 대상으로 한다.

4.1.2 모델링 성분의 대응

- ① IDEF0의 기능 박스는 시스템내의 자원에 의해 수행되는 어떠한 활동, 행위 또는 동작과 같은 일을 의미한다. 이들은 IDEF3에서 단위 행위를 나타내는 UOB에 대응될 수 있다.
- ② IDEF0의 입력은 기능에 의해 처리되는 엔터티를 의미하는 것으로, 타 기능의 출력으로 인해 생성되는 엔터티이거나, 시스템의 외부에서 들어오는 엔터티이다. 기능의 활성을 위하여 이러한 입력 엔터티들이 준비되어 있어야 한다. 즉, 준비 여부가 기능의 활성을 결정한다. 이는 어느 때 준비되어 있느냐 하는 시간적인 요소를 포함한다. IDEF3에서 이들 입력은 UOB의 활성화에 시간적인 제한조건으로 작용하여 연결로 대응될 수 있다.
- ③ IDEF0의 조정은 입력의 경우와 같이 기능의 활성화에 필요한 것이지만, 처리되지 않고, 다만 필요조건으로만 작용한다. 이러한 필요조건은 시간적인 요소를 포함하고 있을 수도 있고, 또한 포함하고 있지 않을 수도 있다. 주기적으로 발생하는 신호 등과 같이 시간적인 요소를 포함하는 제어의 경우에는 기능의 활성화에 시간적인 제한 조건으로 작용한다. IDEF3에서 이들은 연결로 대응된다.
- ④ IDEF0의 출력은 기능의 활성화 후 생성되는 엔터티이다. 즉, 기능의 활

성이 발생되어야 만 생성되는 시간적인 요소를 포함한다. IDEF3에서 이들을 연결로 대응한다.

- ⑤ IDEF0의 미케니즘은 기능의 활성을 담당하는 자원을 의미한다. 자원은 제한된 용량을 가질 수도 있고, 그렇지 않을 수도 있다. 예를 들어 사람과 같은 자원은 그 수가 제한되어 있다. 기능을 수행할 사람이 부족한 경우, 기능의 활성은 연기된다. 이 경우에는 자원이 기능의 활성화에 시간적인 요소로 작용하여 IDEF3의 연결로 대응된다. 그러나, 무한대의 용량을 가진 자원의 경우에는(물론, 무한대의 용량은 존재하지 않지만, 모델이 표현하는 추상화 수준에서 무한대로 가정할 수 있는 경우가 있다) IDEF3에서 제외한다.

4.1.3 IDEF3로의 변환을 위한 추가 정보

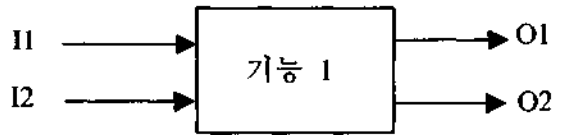
① 기능의 활성화에 필요한 ICOM 조건 IDEF0에서는 기능에 연결된 ICOM 엔터티들 간의 관계를 표현하지 않는다. 예를 들어 한 기능에 두개의 입력이 있다고 하자[그림 1]. 이들 두 입력간의 관계는 IDEF0에서 설명하지 않는다. 이 들간의 관계는 기능의 활성화에 필요한 조건에 따라 다음 경우 중의 하나일 수 있다.

- 두 입력이 모두 존재하여야 한다. (AND)
- 두 입력 중 하나만이 존재하여야 한다. (XOR)
- 두 입력 중 하나 이상의 입력이

있어야 한다. (OR)

각 경우는 IDEF3의 세가지 로직인 AND, XOR, OR의 개념으로 이해될 수 있다. 출력의 경우도 입력과 마찬가지로 3가지 경우가 발생한다. [그림 1]과 같이 한 기능에 두개의 출력이 있다고 하자. 기능의 활성화 후 발생하는 출력에는 다음 경우가 있다.

- 두 출력이 모두 생성된다. (AND)
- 두 출력 중 하나만이 생성된다. (XOR)
- 두 출력 중 하나 이상의 출력이 생성된다. (OR)



[그림 1] IDEF0에서의 기능과 ICOM

입, 출력 조건의 3가지 경우 중 OR조건은 다른 조건에 비해 다양한 경우를 포함한다. 즉, OR조건은 XOR, AND 조건을 포함한다. OR 조건을 포함하는 모델을 top down방식에 의해 자세히 분석하면, AND와 XOR 조건들을 가지는 하부 모델로 분해될 수 있을 것이다. 그러나, 현 모델이 표현하는 추상화 수준에서는 정확한 조건들을 도출할 수 없어 애매한 조건인 OR로 표현하는 경우가 발생한다.

또한, IDEF0에서는 기능의 활성화에 필요한 ICM 엔터티와 기능의 활성화 후 생성되

는 출력 엔터티 간의 관계를 표현하지 않는다. [그림 1]과 같이 한 기능에 두개의 입력(I1, I2)과 두개의 출력(O1, O2)이 있을 때 입력에 대한 전 조건(pre-condition) 과 출력에 대한 후 조건(post-condition)을 표현하는 관계식은 입력의 3 경우와 출력의 3경우를 각각 일대일로 대응시키는 총9가지 경우가 발생한다. 그 중의 한 경우는 다음과 같다.

$$AND(I1, I2) \rightarrow AND(O1, O2)$$

이는 두 입력이 모두 존재하여야 기능이 활성을 시작할 수 있고, 활성 후 두 출력을 모두 생성 시킨다는 의미이다. 이 표현 방법은 project의 위험평가를 위해 사용한 IDEF0모델의 확장에서 제안한 방법과 유사하다[Ang, et al. 1993]. 이 연구에서는 기능의 활성을 유발할 수 있는 모든 고유한 입력, 출력의 관계식으로 활성규칙을 표현하였다. 관계식에서의 연산자는 AND, OR, NOT을 사용하였다.

IDEF0를 IDEF3로 변환시키기 위하여는 위에서 설명된 ICOM간의 관계가 분석되어야 한다. 각각 두개 이상의 입, 출력이 한 기능에 존재하는 경우, 매우 복잡한 관계가 설정될 수 있다. 이는 기능의 활성화에 필요한 조건, 또는 활성 후 생성되는 출력들의 조건을 표현하기 위해 둘 이상의 AND, OR, XOR 조건을 사용하여야 함을 의미한다. 만약, 입, 출력의 관계를 명확히 설정할 수 있다면, 3 연산자 중 OR연산자는 제외시킬 수 있으므로, AND와 XOR연산자로만 관계식들을 표현한다고 하자. 여러 개의 연산자로 구성된 관계식은 입,출력에 각

각 AND 연산자를 하나씩 포함하는 단위 관계식들로 분해될 수 있다. 또한, 역으로, AND 연산자만을 포함하는 여러 개의 단위 관계식은 하나의 복합된 관계식으로 변환 가능하다. 예를 들어, 한 기능의 ICOM들간에 다음과 같은 복합 관계식이 성립한다고 하자.

$$XOR(AND(I1, I2, I3), AND(I1, I2)) \rightarrow AND(XOR(O1, O2), AND(O3, O4))$$

이 관계식으로부터 어떠한 입력이 있을 때 이에 대응되는 어떠한 출력이 발생하는가에 따라 배타적(exclusive)인 모든 경우를 도출하면 다음과 같다.

$$\begin{aligned} AND(I1, I2, I3) &\rightarrow AND(O1, O3, O4) \\ AND(I1, I2, I3) &\rightarrow AND(O2, O3, O4) \\ AND(I1, I2) &\rightarrow AND(O1, O3, O4) \\ AND(I1, I2) &\rightarrow AND(O2, O3, O4) \end{aligned}$$

즉, 하나의 복합 관계식은 입, 출력에 하나의 연산자만을 갖는 단위 관계식으로 분해된다. 역으로, 위와 같은 4개의 단위 관계식은 하나의 복합 관계식으로 변환된다. 4개의 단위 관계식 중 입력조건이 동일한 관계식들에 대해 출력들을 XOR조건으로 묶어 다음과 같은 두개의 식으로 표현될 수 있다(각 경우는 배타적이므로, XOR 연산자를 사용할 수 있다).

$$\begin{aligned} AND(I1, I2, I3) &\rightarrow XOR(AND(O1, O3, O4), \\ &\quad AND(O2, O3, O4)) \\ AND(I1, I2) &\rightarrow XOR(AND(O1, O3, O4), \\ &\quad AND(O2, O3, O4)) \end{aligned}$$

이 두식은 출력 조건이 동일하므로, 입력에 XOR연산자를 사용하여 하나의 식으로 표현하면 다음과 같다.

$$\text{XOR}(\text{AND}(I1, I2, I3), \text{AND}(I1, I2)) \rightarrow \\ \text{XOR}(\text{AND}(O1, O3, O4), \text{AND}(O2, O3, O4))$$

오른쪽 조건을 적절히 변환하면 다음과 같다.

$$\text{XOR}(\text{AND}(I1, I2, I3), \text{AND}(I1, I2)) \rightarrow \\ \text{AND}(\text{AND}(O3, O4), \text{XOR}(O1, O2))$$

그러므로, 복합 관계식과 배타적으로 분해된 여러 단위 관계식들은 동일한 의미를 갖는다. 기능에서의 ICOM간의 관계를 도출하기 위하여 모델링 작업자는 복합 관계식으로 표현할 수도 있고, 여러 개의 단위 관계식으로 분해하여 표현할 수도 있다. 그러나, 복잡한 하나의 복합 관계식을 도출하기 보다는 각 기능에서 ICOM간의 관계를 AND 조건의 ICM과 이에 대응되는 AND 조건의 출력으로만 구성된 단위 관계식들을 도출하는 것이 보다 용이하다. 모델링 작업자는 입력과 이에 대응되는 출력을 각 경우에 대하여 쉽게 도출할 수 있기 때문이다.

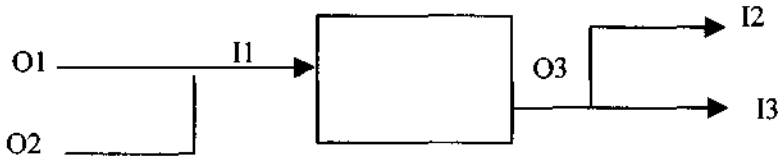
ICOM간의 관계는 IDEF3에서 접속의 의미와 같이 동기적 또는 비동기적으로 구분될 수 있다. 입력조건에서 AND 조건으로 설정된 두 입력 I1, I2가 동시에 들어와야 한다는 조건이 있으면 동기적이다. 두 입력이 어느 때 들어오든지 간에 모두 들어오는 경우엔 비동기적이며 이 기능이 활성화 된다면 이는

비동기적이다. 비동기적의 의미는 동기적의 의미를 포함하고 있다고 볼 수 있다. 본 연구에서는 ICOM의 관계들이 모두 비동기적인 시간적 의미를 가지고 있다고 가정한다.

IDEF3의 연결은 UOB간의 활성화 부과된 시간적 제약의 종류에 따라 단순 선행, 제한된 선행, 논리적 연결들로 구분되어 있다. IDEF0에서는 이러한 시간적인 제약사항을 포함하지 않는다. 본 연구에서는 IDEF0에서의 기능이 위에서 언급된 ICOM 단위 관계식 중 어느 한 식을 만족할 경우에 한하여, 반드시 활성을 시작하고, 활성화 후 반드시 관련된 출력을 생성한다고 가정한다. 즉, ICOM들은 이러한 성격의 IDEF3 연결로 변환된다.

② 결합(join)과 분기(split)의 의미

IDEF0에서 결합과 분기는 그 의미가 매우 애매모호하다. 기능들에서 생성된 두 출력(O1, O2)이 결합으로 연결되어 한 기능의 입력(I1)으로 표현되었다고 하자(그림 2). 이 때 두 출력과 입력의 세부적인 관계는 위에서 설명된 ICOM관계에서와 같이 다양한 경우가 있을 수 있다. 즉, 두 출력이 AND, OR, XOR 중 하나의 조건으로 합쳐져 하나의 입력을 형성한다고 생각할 수 있다. 분기의 경우에도 결합의 경우와 유사하다. 둘 이상의 다수 선들이 결합 또는 분기된 경우에는 복잡한 관계식이 성립한다. 이러한 경우, 기능에서의 ICOM관계식 표현에서와 같이 AND, XOR 중 하나의 연산자만으로 표현된 배타적인 단위 관계식들로 표현하도록 한다.



[그림 2] IDEF0에서의 분기와 결합

4.1.4 변환절차

IDEF0를 IDEF3로 변환하는 절차는 다음과 같다.

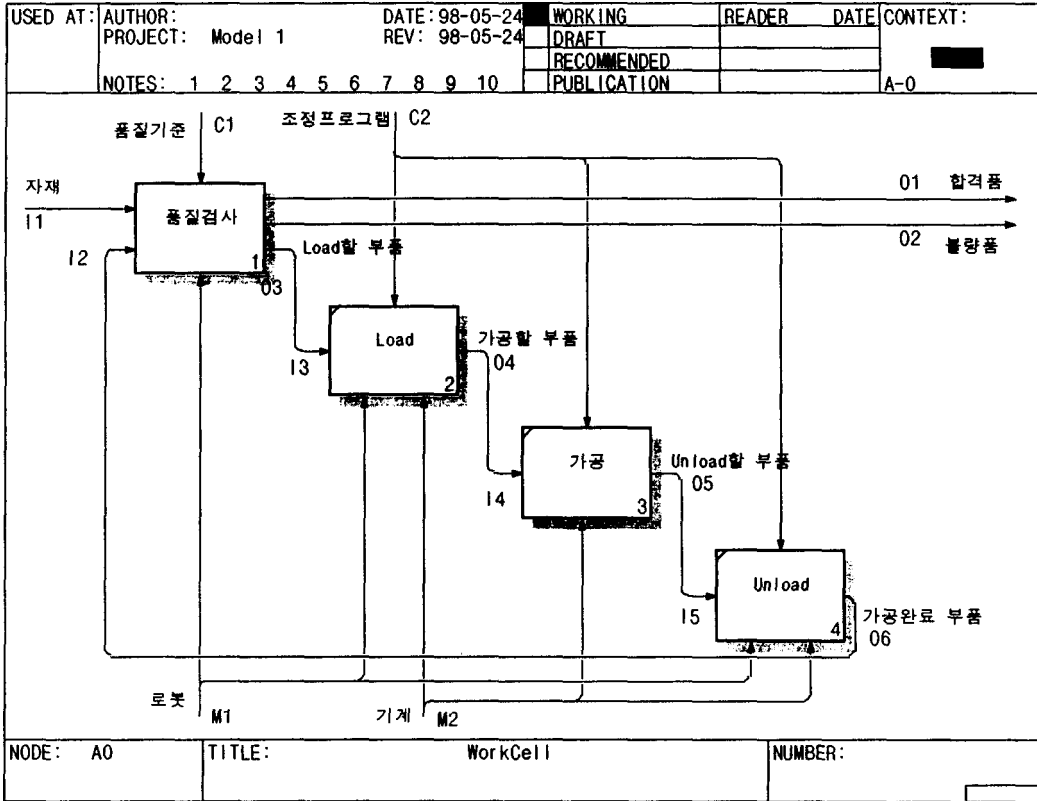
- ① IDEF0 다이어그램에서 시간적인 제약을 받지 않는 ICOM을 제거한다.
- ② 각 기능에 대하여 서로 배타적인 ICOM 단위 관계식들을 입력 받아 입력이 같은 관계식들을 출력에 XOR연산자로 합병한 후, 출력이 같은 관계식들을 입력에 XOR연산자로 합병한다.
- ③ 각 ICOM의 관계식을 UOB, UOB의 입력 연결, 입력 접속, 출력 연결, 출력 접속으로 변환한다.
- ④ 각 UOB의 입력과 출력의 관계로부터 입력과 출력이 동일한 경우 서로 연결한다.
- ⑤ 결합과 분기에 대해 배타적인 관계식들을 입력 받아 기능의 ICOM관계식에서와 같이 XOR연산자로 병합한다. 각 연산자를 접속으로 하여 해당 연결을 잇는다.

제안된 변환절차를 설명하기 위하여 workcell에서의 IDEF0모형을 예로 제시한다. [그림 3]은 하나의 로봇과 기계로 구성된 workcell에서의 작업을 IDEF0모형으로 표현한 것이다.

단계 1] IDEF0 모형에서 시간적인 제약을 주지 못하는 ICOM인 C1과 C2를 제거한다. C1은 품질 기준으로서 품질검사에 필수적인 조건이 되지만, 언제라도 필요할 경우에 즉시 제공되는 성질의 것으로 간주된다. C2는 기계와 로봇을 운영하기 위한 조정 프로그램으로 역시 언제라도 즉시 제공된다고 간주된다. 이들 조정들은 IDEF3변환에 필요한 자료에서 제외된다.

단계 2] 각 기능에 대한 ICOM 단위 관계식을 입력 받는다.

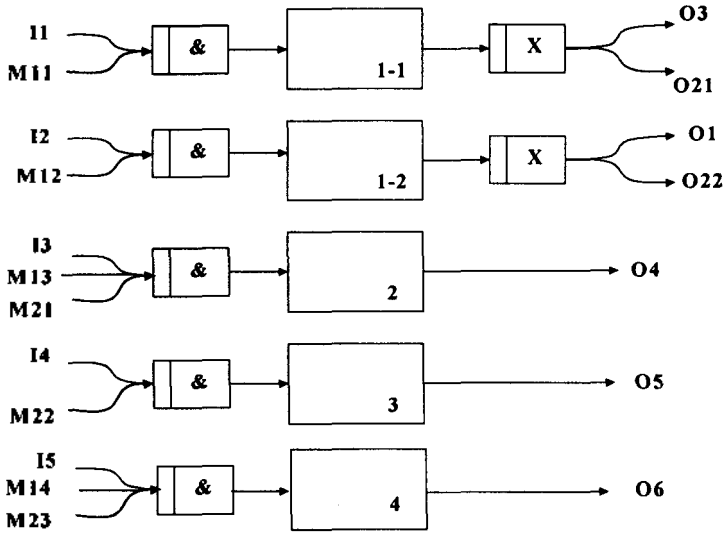
각 기능에 대한 ICOM 단위 관계식은 [표 2]와 같다. 기능 1에서 동일한 전 조건을 갖는 단위 관계식들이 존재하므로 이들을 후 조건에 XOR 연산자로 병합한다.



[그림 3] Workcell의 IDEF0 모형

[표 2] Work cell의 각 기능에 대한 관계식

기능	단위 관계식	병합된 관계식
1	$AND(I1, M1) \rightarrow O3$ $AND(I1, M1) \rightarrow O2$ $AND(I2, M1) \rightarrow O1$ $AND(I2, M1) \rightarrow O2$	$AND(I1, M11) \rightarrow XOR(O3, O21)$ $AND(I2, M12) \rightarrow XOR(O1, O22)$
2	$AND(I3, M1, M2) \rightarrow O4$	$AND(I3, M13, M21) \rightarrow O4$
3	$AND(I4, M2) \rightarrow O5$	$AND(I4, M22) \rightarrow O5$
4	$AND(I5, M1, M2) \rightarrow O6$	$AND(I5, M14, M23) \rightarrow O6$



[그림 4] 각 단위관계식의 IDEF3 성분 변환

단계 3] 각 ICOM 관계식을 IDEF3 구성요소로 변환한다.

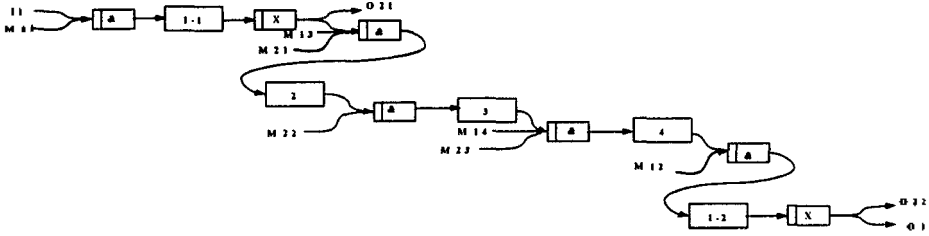
[표 2]에서 병합된 관계식은 모두 5개가 존재한다. 이들 각 관계식을 [그림 4]와 같이 IDEF3성분으로 변환한다. 이 때 각 단위 관계식의 ICOM들이 유일하게 인식되도록 한다. 즉, 기능 2, 3, 4의 단위 관계식에서 전 조건에 모두 M2가 존재한다. M2가 각 단위 관계식에서 유일하게 인식되도록 M21, M22, M23로 구분한다. 이들은 세 개로 구분됐지만, 모두 M2인 기계를 의미한다. M2와 M21, M22, M23의 관계는 단계 5

에서 설정된다.

단계 4] UOB의 동일한 입출력을 서로 연결한다.

[그림 3]의 IDEF0 모형에서 O3, O4, O5, O6들은 각각 I3, I4, I5, I2와 동일한 관계에 있다. 동일한 ICOM들을 서로 연결한다. [그림 5]는 이러한 결과를 보여준다.

단계 5] 결합과 분기에 대한 관계식을 입력받아 이들을 IDEF3 구성요소로 변환한다.



[그림 5] 동일한 입출력 연결

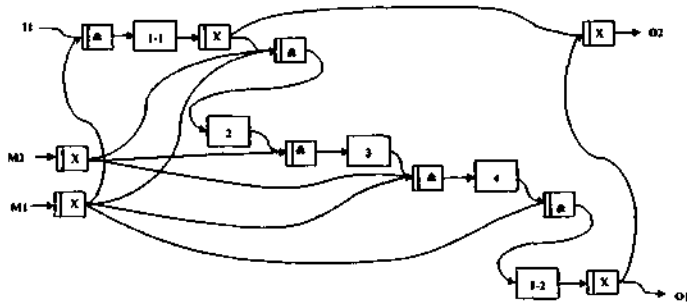
[그림 5]에서 M1은 M11, M12, M13, M14로 분기되었고, M2는 M21, M22, M23로 분기되었다. 또한, O21과 O22는 O2로 결합되었다. 이들의 관계는 다음과 같다.

$$M1 \rightarrow \text{XOR}(M11, M12, M13, M14)$$

$$M2 \rightarrow \text{XOR}(M21, M22, M23)$$

$$\text{XOR}(O21, O22) \rightarrow O2$$

이들 관계식을 이에 대응되는 IDEF3성분으로 변환한다. [그림 6]은 단계 5의 결과를 나타낸다.



[그림 6] 분기, 결합의 변환

4.2 IDEF3/Petri Net 변환 방법

Petri Net에서의 플레이스는 시스템의 활동 또는 상태를 표현하고, 트랜지션은 사건을 표현한다. IDEF3에서의 UOB는 활동의 의미가 있으므로, Petri Net의 플레이스로 변환될 수 있다. 접속 중 합쳐진(fan in) 또는 분기된(fan out) AND 접속은 두개 이상의 연결이 합쳐지거나 분기되는 사건으로 간주하여 트랜지션으로, XOR 접속은 의사결정을 의미하는 플레이스로, 그리고, 연결은 아크로 대체될 수 있다. 구체적인 변환절차는 다음과 같다.

- 1) IDEF3 성분을 Petri Net성분으로 대체한다[그림 7].
 - UOB를 플레이스로 대체한다.
 - AND 접속을 트랜지션으로 대체한다.
 - XOR 접속을 플레이스로 대체한다.
 - 연결을 아크로 대체한다.
- 2) 플레이스와 트랜지션을 추가하여 유효한 Petri Net 모델이 되도록 한다.

1)에서 IDEF3의 각 성분을 이에 대응되는 Petri Net 성분으로 대체하였을 경우, 플레이스에서 플레이스로 잇거나, 트랜지션에서 트랜지션으로 잇는 아크들이 존재할 수 있다. 유효한 Petri Net 모델이 되기 위하여 아크는 항상 플레이스와 트랜지션사이를 이어야 한다. 만약, 플레이스와 플레이스를 잇는 아크가 있을 경우, 중간에 트랜지션을 추가하고, 트랜지션과 트랜지션을 잇는 아크가 있으면 플레이스를 중간에 추가하여 유효한 모델

이 되도록 한다.

- 3) 자원 플레이스를 추가한다.

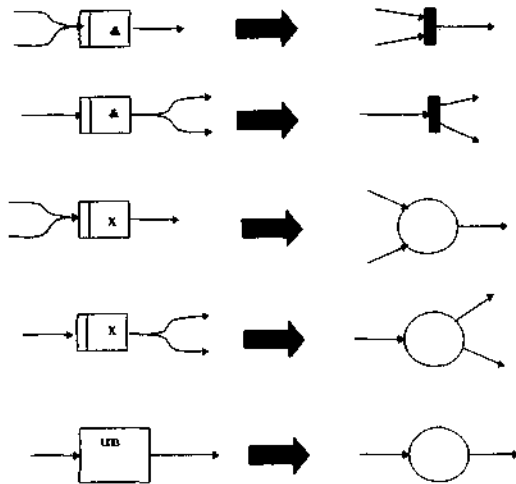
실행 가능한 동적인 프로세스 모델에서 기능 또는 활동을 수행하는 자원을 모델에 포함하여야 한다. 모든 활동들의 실행은 제한된 자원을 사용하여야만 가능하기 때문이다. 만약, IDEF0모델에서 활동에 필요한 자원을 미케나즘으로 모델링했다면, 이는 변환된 IDEF3 모델에도 포함되어 있을 것이다. 그러나, IDEF3에서는 연결로 자원의 흐름만을 모델링하기 때문에 Petri Net 모델에서는 자원의 유효한 상태를 나타내기 위한 플레이스가 추가되어야 한다.
- 4) 자원 플레이스로 복귀하는 아크를 추가한다.

한 활동에서 제한된 자원을 필요로 하여 사용하였다면, 사용 후 이를 되돌려주어 다른 활동에서도 공유하도록 하여야 한다. 이를 위하여 자원을 나타내는 아크의 입력이 있는 트랜지션에 대하여 출력 플레이스의 모든 출력 트랜지션들에서 자원 플레이스로의 아크를 추가한다. 만약, 어느 트랜지션에서 동일한 자원의 입, 출력이 동시에 존재한다면, 플레이스 아크, 트랜지션을 추가하여 자원의 출력이 새로이 추가된 트랜지션에서 발생되도록 조정한다. [그림 8]의 플레이스, p13,14,15와 트랜지션 t10,t11,t12가 이 경우에 해당한다.
- 5) 필요할 경우, Petri Net 모델을 축소한다.

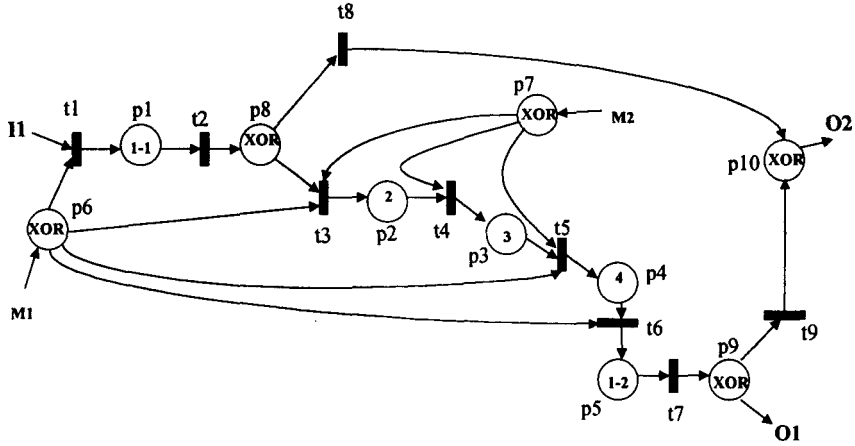
한 트랜지션이 하나의 입, 출력 플레이스만을 갖고, 입, 출력 플레이스들이 각각 하나의 출력, 입력 트랜지션을 갖는다면, 두 플레이스와 트랜지션을 합쳐 하나의 플레이스로 축소될 수 있다.

[그림 6]의 IDEF3모형을 제안된 절차에 의해 Petri Net모형으로 변환한다. 각 IDEF3성분을 Petri Net성분으로 변환하여 유효한 모델을 생성한 한 결과는 [그림 8]과 같다. [그림 8]에서 트랜지션 t2, t7, t8, t9는 유효한 Petri Net모형을 위하여 추가되었다. [그림 9]는 자원 플레이스들

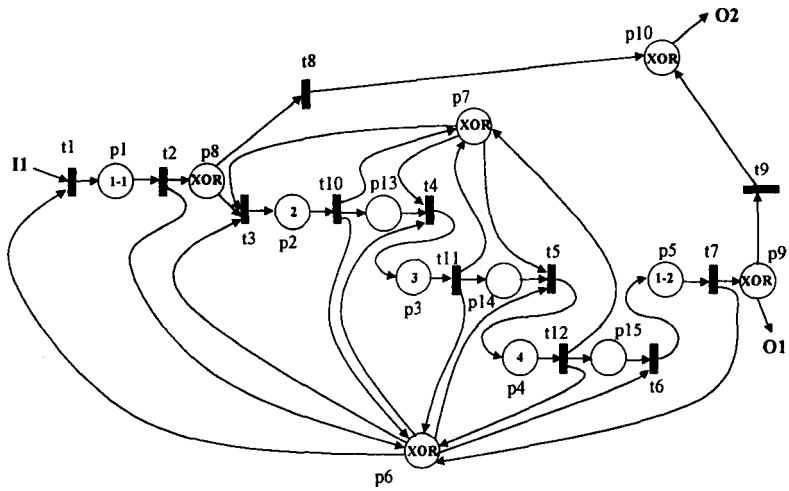
추가하고, 이 플레이스로 복귀하는 아크를 추가한 후 자원 플레이스와 자원의 XOR 관계를 나타내는 플레이스를 합쳐 하나의 플레이스로 축소한 결과를 보여준다. [그림 8]에서 자원 플레이스로 복귀하는 아크를 추가할 경우 트랜지션 t4, t5, t6에서 자원의 입출력이 동시에 발생한다. 출력 후 입력이 발생하도록 플레이스 p3, p4, p5와 트랜지션 t10, t11, t12이 추가되었다. 즉, 자원 플레이스의 출력이 새로운 트랜지션들에서 발생되도록 하였다.



[그림 7] IDEF3 구성요소의 Petri Net 변환



[그림 8] IDEF3모형으로부터 Petri Net 모형의 변환



[그림 9] 자원 플레이스로의 복귀를 나타내는 아크가 추가된 Petri Net 모형

5. 결론

본 연구에서는 기업 통합의 목적에서 이용될 수 있는 비즈니스 프로세스 모델링 통합 방법론을 제안하였다. 구체적으로는 IDEF0를 이용한 기능 모델링, IDEF3를 이용한 프로세스 흐름 묘사, Petri Net을 이용한 실행 가능한 동적 모델링 방법들을 대상으로 하였다. 각 모델링 방법은 묘사 방법의 차이와 모델링 관점의 상이함으로 인하여 서로 다른 모델을 생성한다. 모델링 통합 방법론은 이렇게 다른 관점에서의 여러 모델들이 독립적으로 생성되고, 분석되기 보다 서로 연관된 모델들로 구성되어 한 모델에서 다른 모델로 쉽게 변환될 수

있는 모델의 연관성을 유지토록 함을 의미한다.

IDEF0기능 모델링은 누구에게나 쉽게 이해될 수 있는 모델링 관점을 제공하여 모델링 초기에 우선적으로 작성될 수 있다. 이 IDEF0모형은 프로세스의 시간적인 절차에 관한 추가적인 자료를 입력으로 하여 IDEF3로 변환된다. IDEF3 모델은 또한, 자원에 대한 상세한 자료를 입력으로 하여 Petri Net 모형으로 변환된다. 본 연구에서 제안한 모델의 변환 방법은 자동적으로 수행이 될 수 있는 장점을 제공하여 모델링 작업자에게 많은 부담을 줄일 수 있으리라 사료된다.

참고 문헌

- [1] 박찬권, 컴퓨터 통합 생산을 위한 프로세스 설계 지원시스템에 관한 연구, 서울대학교 대학원 박사학위논문, 1996.
- [2] Ang, C.L. and Gay, R.K.L., "IDEF0 Modeling for Project Risk Assessment", *Computers in Industry*, Vol. 22, pp. 31-45, 1993.
- [3] Boucher, T.O., and Jafari, M.A., "Design of a Factory Floor Sequence Controller From a High Level System Specification," *Journal of Manufacturing Systems*, Vol. 11, No. 6, pp. 401-417, 1992
- [4] Colquhoun, G.J, Baines, R.W, and Crossley, R., "A Composite Behavioural Modelling Approach for Manufacturing Enterprise", *International Journal of Computer Integrated Manufacturing*, Vol. 9, No. 6, pp. 463-475, 1996.
- [5] Delcambre, L.M.L., Landry, S.P., Pollacia, L. and Waramahaputi, J., "Specifying Object Flow in an Object-Oriented Database for Simulation," *Proceedings of the SCS Multiconference on Object Oriented Simulation*, A. Guasch (Ed.), pp. 75-80, 1990.
- [6] Department of Defence, *Functional Process Simulation - A Guidebook*, Department of Defence, 1993.

-
- [7] Eddins, W.R., Sutherland II, D.E. and Crosslin, R.L., "Using Modeling and Simulation in the Analysis and Design of Information Systems," *Dynamic Modeling of Information Systems*, H.G. Sol and K.M. van Hee (Eds.) Elsevier Science Publishers B.V. (North-Holland), pp. 61-68, 1991.
- [8] Eirich, P., Ed., "Enterprise Modeling: Issues, Problems & Approaches", *Reports of the ICEIMT Special Interest Groups*, 1992.
- [9] FIPS 183, *Draft Federal Information processing Standards Publication 183*, 1993
- [10] Gay, R.K.L., Lim, R. and Shing, L.W., "Using IDEF Methodology for Functional, Information and Dynamic Modeling of a FMS," *Proceedings of the International Conference on Computer Integrated Manufacturing: ICCIM '91*, B.S. Lim (Ed.), pp. 97-100, 1991.
- [11] Malhotra, R. and Jayaraman, S., "An Integrated Framework for Enterprise Modeling," *Journal of Manufacturing Systems*, Vol. 11, No. 6, pp. 426-441, 1992.
- [12] Mayer, R.J., Menzel, C.P., Painter, M.K., deWitte, P.D., Blinn, T., Perakath, B., *Information Integration for Concurrent Engineering IDEF3 Process Description Capture Method Report*, Knowledge Based Systems, Inc, 1995.
- [13] Medina-Mora, R., Winograd, T., and Flores, R., "Action Workflow as the Enterprise Integration Technology", *Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society*, Vol. 16, No. 2, June, 1993.
- [14] Peterson, J.L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.
- [15] Pinci, V.O. and Shapiro, R.M., "An Integrated Software Development Methodology Based on Hierarchical Colored Petri Nets," *Advances in Petri Nets 1991, Lecture Notes in Computer Science 524*, G. Rozenberg (Ed.), Springer-Verlag, 1991.
- [16] Rensburg, A.V. and Zwemstra, N., "Implementing IDEF Techniques as Simulation Modelling Specification," *Computers and Industrial Engineering*, Vol. 29, No. 1-4, pp. 467-471, 1995.
- [17] Ross, D.T., "Structured Analysis(SA): A Language for Communicating Ideas," *IEEE Transactions on Software Engineering*, Vol. 3, No. 1, pp. 16-34, 1977.
- [18] Young, R.E. and Vesterager, J., "An Approach to CIM System Development Whereby Manufacturing People Can Design and Build Their Own CIM Systems," *International Journal of Computer Integrated Manufacturing*, Vol. 4, No. 5, pp. 288-299, 1991.
- [19] Zaytoon, J., Niel, E., and Mille, A., "A Functional-to-Behavioural Transformation Framework for Specification and Design of Manufacturing Systems," *Computer Integrated Manufacturing Systems*, Vol. 7, No. 4, pp. 247-258, 1994.

저자 소개

임동순

한양대학교 산업공학과 학사

한국과학기술원 산업공학과 석사

Iowa State University 산업공학과 박사

현재 한남대학교 산업공학과 부교수

김철한

한양대학교 정밀기계공학과 학사

한양대학교 정밀기계공학과 석사

포항공과대학교 산업공학과 박사

현재 대전대학교 산업공학과 조교수

우훈식

한양대학교 산업공학과 학사

Iowa State University 산업공학과 석사

Iowa State University 산업공학과 박사

현재 대전대학교 산업공학과 전임강사

김중인

한양대학교 산업공학과 학사

한양대학교 산업공학과 석사

Arizona State University 산업공학과 박사

현재 홍익대학교 경영정보학과 조교수