

이메뉴팩처링을 위한 협업 프로세스 모델링

류광열^{1*} · 조용주² · 최헌종² · 이석우²

¹인시아드 / ²한국생산기술연구원

Collaborative Process Modeling for Embodying e-Manufacturing

Kwangyeol Ryu¹ · Yongju Cho² · Honzong Choi² · Seokwoo Lee²

¹Technology Management Area, INSEAD, Boulevard de Constance, Fontainebleau Cedex 77305, France

²Korea Institute of Industrial Technology, Incheon, 406-130

e-Manufacturing can be referred to as a system methodology enabling the integration of manufacturing operations and IT technologies to achieve objectives of an enterprise. It is recently regarded as a powerful solution to survive in a chaotic marketplace. While conducting an e-Manufacturing project, we first needed to capture collaborative processes conducted by multiple participants in order to define functions of a system supporting them. However, pervasive process modeling techniques including IDEF3, Petri nets, and UML are not sufficient for modeling collaborative processes. Therefore, we first briefly investigate several process modeling methods including aforementioned modeling methods and ARIS focusing on the collaboration point of view. Then, we propose a new modeling method referred to as Collaborative Process Modeling (CPM) to clearly describe collaborative processes. Also, we develop and illustrate a rule for transforming collaborative process models into Marked Graph models to use the analysis power of the Petri nets. Using CPM empowers us to develop collaborative process models with simple notations, understand and facilitate the realization of the collaboration, and verify models before rushing into the development.

Keywords: Collaborative Process Modelling (CPM), Marked Graph, e-Manufacturing

1. 서론

현재 제조 산업은 점차 예측하기 힘든 여러 상황 조건에 직면하고 있으며, 이는 주로 급변하는 소비자들의 요구와 사회적 요구조건의 변화 뿐 아니라, 지역적 또는 국가 정부차원의 환경적 규제 변화로 인한 것이다. 이러한 치열한 경쟁 상황과 수시로 변화하는 환경 조건에 대응하기 위해서 오늘날의 제조 산업은 기초 인프라 확충은 물론 이를 이용할 수 있는 고도의 IT 기술을 보유하고 있어야 한다. 특히 지난 수십 년 간의 인터넷의 급속한 보급 및 확산은 기업 간 전자상거래를 활성화시키

는데 결정적인 요인을 제공하였으며, 결국 e-Business는 기업의 비즈니스 프로세스에 대한 정의를 새로운 시각에서 해석하게 하는 기회를 제공하게 되었다. 각 기업은 e-Business를 기업 내에 적용하기 위해 여러 기법을 이용하게 되는데, 이는 주로 전사적 자원관리(Enterprise Resource Planning; ERP), 공급망 관리(Supply Chain Management; SCM), 제품 주기관리(Product Lifecycle Management; PLM) 등이 있다. 이러한 기법을 적용함으로써 생산 비용의 절감 및 제품의 고품질화는 물론, 생산시스템에서의 제품 생산 개발주기 및 납기의 단축 등으로 소비자의 다양한 요구 조건을 만족시키는 데 큰 도움이 되어 왔다(Lee, 2003).

*연락처 : 류광열, Technology Management Area, INSEAD, Boulevard de Constance, Fontainebleau Cedex 77305, France,

Fax : +33 (0)1 60 74 55 00/01, E-mail : ducksal@postech.ac.kr

2004년 11월 11일 접수, 2회 수정 후 2005년 7월 15일 게재 확정.

이러한 e-Business의 다양한 요소 기술과 인터넷 및 IT 기술을 제조 산업에 적용시켜 제조업의 업무 프로세스, 관리방식, 거래 및 사업 방식을 혁신하고 통합하는 일종의 제조 혁신 전략의 수단이 바로 e-Manufacturing이다(Ryu *et al.*, 2004).

e-Manufacturing은 기존의 생산시스템 요소를 단순히 인터넷에 연결하는 것뿐만이 아닌, 궁극적으로는 제조업의 전략적 가치를 재구성할 수 있는 새로운 제조 산업의 패러다임으로서 제조 프로세스의 통합은 물론, 제품 개발 과정에서의 협업, 구매 조달 및 생산 현장의 통합을 이루기 위해 모든 비즈니스 주체 즉, 공급자, 생산 설비요소, 소비자를 웹 기반 기술을 사용함으로써 효과적으로 연계시키는 방법론이라 할 수 있다. 특히 IT 기술을 이용한 e-Manufacturing의 보급 및 확산으로 생산 공정상 가져온 변화 중 대표적인 것이 바로 제품개발 과정에서의 협업이 가능하게 된 것이다.

이러한 협업의 의미는 기존의 오프라인 상에서의 협업이 아닌, 인터넷 및 IT 기술을 이용한 온라인 협업을 뜻하며 이는 기존의 동시 공학(Concurrent Engineering; CE)적 접근 방법의 연장선상에서 해석이 가능하다고 할 수 있다.

그러나 e-Manufacturing의 범주는 기존 제조 산업의 모든 범위를 포함하는 동시에 기업의 경영 측면에서의 최상위에 존재하는 비즈니스 프로세스를 포함하는 더 큰 범주로 해석이 가능하므로 동시 공학적 접근 방법은 e-Manufacturing 범주에 속하는 하나의 요소기술로 해석할 수 있다.

일반적으로 프로세스(Process)는 행위의 주체에 따라 크게 협업 프로세스(Collaborative Process)와 일반 프로세스(Normal Process)로 나눌 수 있다. 여기서 일반 프로세스란 단일 주체에 의해 행해지는 일련의 작업들을 말하며, 협업 프로세스는 다수의 행위자들의 협력에 의해 행해지는 일련의 작업들을 말한다. 협업 프로세스는 해당 프로세스를 분해하였을 경우 최종적으로 얻어지는 각 단위 작업의 범위를 명확히 구분하기 힘들 수 있다.

그러나 본 연구에서는 서로 다른 주체에 의해 수행되는 다수의 기본 작업의 집합을 갖는 프로세스 또한 협업 프로세스로 간주하며, 다수의 기본 작업의 집합이라 할지라도 단일 주체에 의해 행해지는 프로세스는 일반 프로세스로 간주한다.

또한, 본 논문에서 말하는 협업은 오프라인 협업이 아닌 온라인 협업을 말하며 이는 동기 협업(Synchronous Collaboration)과 비동기 협업(Asynchronous Collaboration)을 모두 포함한다. 이러한 협업 프로세스는 다시 내부 협업(Intra-collaboration)과 외부 협업(Inter-collaboration)의 두 가지로 분류될 수 있다. 이를 구분 짓는 경계는 보통 행위자가 속해 있는 기업을 지칭한다. 따라서 같은 기업 내의 다른 부서 사이의 업무 협조는 내부 협업으로, 다른 기업과의 업무 협조는 외부 협업으로 간주한다. 이를 구분 짓는 이유는 내부 협업과는 달리 외부 협업 시에는 관련 데이터의 공유, 유출 및 배포와 관련해 서로 상이한 보안 정책을 사용하기 때문이며, 협업을 지원하는 시스템 개발 시 이와 관련된 사항이 반드시 반영되어야 한다.

중소 금형업체를 대상으로 설계 부문의 온라인 협업을 지원하기 위한 시스템 개발을 위해 e-Manufacturing 시범 사업이 진행된 바 있다. 현재 금형업체 간의 협업은 시간적·지역적 한계로 인해 활발하게 이루어지지 못하고 있으며, 관련 업체들 간의 긴밀한 협조로 기술 교류 및 공유를 통해 경쟁력을 키워 나가야 함을 인정하면서도 보안상의 문제로 인해 협력사나 하청업체 등 과거 거래실적이 있는 신뢰할만한 업체들과의 업무 협조에 한계를 두고 있다. 경쟁력이 점차 낮아져 결국 존재위기에까지 몰린 국내 금형업체의 재도약은 결국 업체 간 협업을 통해 기업의 경쟁력을 갖추므로써 가능할 것으로 보인다. 가령, 온라인 협업을 통한 제품개발 과정에서의 시방변경 회수 및 납기 단축의 실현은 협업을 통해 얻을 수 있는 경쟁력의 전형적인 예라 할 수 있다.

기업 간 원활한 협업을 지원할 수 있는 시스템을 구축하기 위해서는 우선 각 금형업체들 간의 협업 프로세스를 정확히 파악하고 이를 구체화 시켜야만 한다. 이는 협업 프로세스를 지원하기 위해 협업 시스템이 가져야 할 기능을 설계하는데 가장 핵심적인 역할을 하기 때문이다. 협업 프로세스의 구체화는 모든 협업 프로세스를 모델링하고 이를 분석함으로써 가능하다. 일반적인 업무 프로세스에 비해 협업 프로세스가 갖는 가장 큰 특징은 다수의 주체에 의해 수행된다는 것이며 프로세스 모델링 시 이에 대한 정보가 명확히 표현되어야 한다. 내부 협업 프로세스는 어느 부서들이 담당하는지, 외부 협업 프로세스는 어떤 기업들의 어느 부서들에서 담당하는지 그 행위의 주체가 명시되어야 해당 프로세스에 대한 책임소계가 분명해지며 해당 프로세스를 수행하기 위해 누구와 협력해야 하는지 파악할 수 있기 때문이다.

따라서 협업 프로세스를 명확히 모델링하기 위해서는 각 프로세스가 어떠한 형태의 프로세스인지(가령, 내부 협업인지 외부 협업인지), 각 프로세스를 수행하는 주체가 누구인지, 그리고 각 프로세스에서 수행하기 위한 메시지의 흐름이 어떻게 되는지가 표현되어야 한다. 또한 협업 프로세스 모델링 방법은 모델의 검증을 지원할 수 있어야 한다. 모델 상에 정의된 각 협업 프로세스는 Formalism을 통한 수학적 모델 검증이나 시뮬레이션 등을 통해 수행 이전에 미리 유효성을 검증해 볼 수 있어야 한다. 이것은 일반적으로 협업 프로세스가 일반 프로세스에 비해 더 많은 자원(예: 각 수행 주체의 시간 및 비용 등)을 낭비하기 때문이다.

본 논문에서는 우선 기존의 프로세스 모델링 기법을 분석해 보고 이를 바탕으로 협업 프로세스를 쉽고 명확하게 도출할 수 있는 협업 프로세스 모델링(Collaborative Process Modeling; CPM) 방법을 제안하고자 한다. CPM은 간단한 표기법(Notation)을 이용하여 협업 프로세스를 명확히 도출하고 모델링할 수 있으며, 이를 협업 시스템의 상세 메커니즘을 설계·운영할 수 있게 하는데 도움을 줄 수 있다. e-Manufacturing 사업의 온라인 협업 지원 시스템 개발 과정 중 본 논문의 목표 영역은 <그림 1>과 같다.

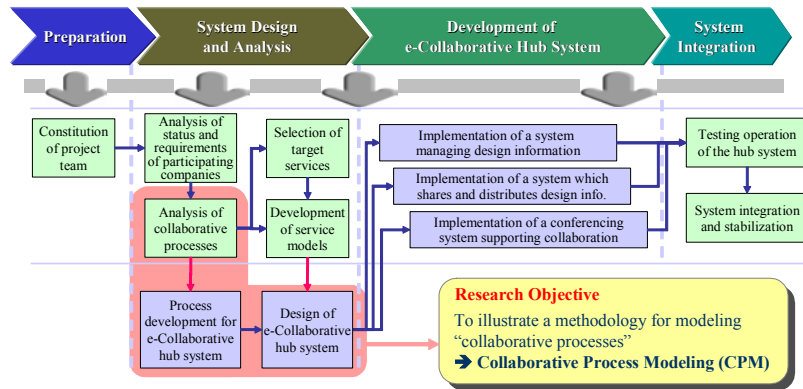


그림 1. 온라인 협업 지원 시스템 개발을 위한 연구 주제의 도출 영역.

2. 협업을 위한 프로세스 모델링

2.1 IDEF3

IDEF는 Integrated Computer-Aided Manufacturing(ICAM) 프로젝트의 일환으로 1981년에 처음 소개가 되었으며 IDEF0, IDEF1x, IDEF3, IDEF5 등 여러 모델링 방법을 정의하고 있다 (Mayer *et al.*, 1995). 이 중 시스템에 존재하는 프로세스를 정의하기 위해 개발된 것이 바로 IDEF3이며 IDEF0도 어느 정도 프로세스를 정의하는데 도움을 줄 수 있다. IDEF0가 시스템 내에서 행해지는 행위(Activity), 즉 기능(Function)이 무엇인가를 정의할 수 있는 반면(NIST, 1993), IDEF3는 시스템이 어떤 프로세스의 흐름에 의해 작동되는가에 대한 현장 전문가의 지식을 표현할 수 있는 구조적 방법을 제공하는 것이다.

이러한 IDEF 모델링 방법의 가장 큰 특징은 목적에 따른 도구의 정의가 명확히 구분되어 있다는 것과 간단하면서도 강한 표현력을 가지고 있다는 것이다. IDEF3를 이용한 프로세스 모

델은 시스템에서 진행되는 프로세스를 순차적으로 표현하며 프로세스 간 연관관계를 시간의 흐름에 따라 표현할 수 있다 (Mayer *et al.*, 1995).

금형업체에서의 금형설계를 위한 프로세스는 IDEF3의 Process Flow Network(PFN)을 이용하여 <그림 2>와 같이 모델링할 수 있다(색칠해진 프로세스는 협업 프로세스를 표현하기 위함이다). IDEF3를 이용한 모델링은 프로세스를 쉽게 표현하고 파악할 수 있다는 장점이 있으나 이는 정적(Static)인 모델링 방법이므로 프로세스의 흐름을 다이내믹하게 표현할 수 없으며 설계된 모델을 검증할 수 없다는 단점이 있다. 또한 IDEF3를 이용해 설계된 모델에서의 각 행위(Unit of Behavior; UOB)의 주체는 모두 동일하다고 간주해야만 하며, UOB 간의 협업을 요하는 프로세스인 경우 이와 관련된 다수의 행위 주체를 표현할 수가 없다는 한계가 존재한다. Referent를 사용하여 행위의 주체를 표기할 수 있기는 하나 이는 프로세스 모델에 직접적으로 반영된 것으로는 볼 수 없다. IDEF0의 ICOM(Input, Control, Output, Mechanism) 요소 중 메커니즘을 이용하여 정의된 기능

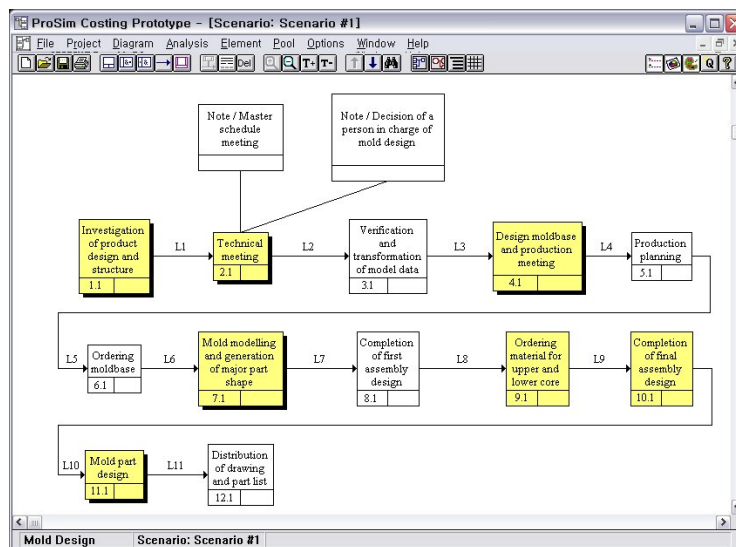


그림 2. IDEF3를 이용한 금형설계 프로세스 모델의 예.

과 관련된 리소스를 표현할 수는 있기 때문에 IDEF0 모델의 각 기능들이 IDEF3 모델의 각 프로세스와 매핑(mapping)된다면 메커니즘으로 표현된 리소스를 해당 프로세스 행위의 주체로 간주할 수 있다. 그러나 IDEF3는 원칙적으로 프로세스의 흐름을 표현할 뿐 IDEF0의 ICOM 요소가 직접적으로 적용된다고는 할 수 없다.

2.2 Petri Nets

Petri nets은 자원(기계나 장비, 작업자 등)을 공유하는 시스템의 공동 프로세스(Concurrent Process)에 대한 분석이 가능하다는 사실이 검증된 모델링 방법으로 여러 종류의 시스템을 설계하거나 분석하기 위해 널리 사용되어지고 있다(Santarek and Buseif, 1998). 일반적인 Petri nets은 플레이스, 트랜지션, 아크, 토큰의 네 가지 요소로 모델링된다. 특히 Marked Graph는 Petri nets에 존재하는 각 플레이스가 오직 한 개의 입력과 한 개의 출

력을 갖는 경우를 말한다(Thirusangu and Rangarajan, 1995). Petri nets을 이용하여 모델링된 시스템은 토큰(Token)의 흐름을 추적함으로써 시스템의 동적 상황을 미리 예측할 수 있다.

Marked Graph를 이용하여 금형 제작과 관련된 일련의 행위를 모델링한 예는 <그림 3>과 같으며 각 플레이스와 트랜지션은 <표 1>에 명시되어 있다. 앞서 IDEF3를 이용한 모델링의 예(<그림 2>)와 마찬가지로 Petri nets 모델 또한 협업에 대한 표현이 명확하게 나타나지 않는다. 가령 O5는 업체 외부와의 협업이, 그리고 O4, O7, O11은 실제로 한 업체 내에서도 여러 부서 간의 협업이 필요한 오퍼레이션들이지만 모두 같은 모양의 플레이스로 표현되므로 직관적으로 구분하기는 힘들다. 협업 프로세스를 나타내는 오퍼레이션을 정의하기 위해서 예를 들어 <표 1>의 Description 항목에 명시할 수는 있으나 Petri nets 모델의 그래픽 표현에 대한 직관적인 이해는 쉽지 않다. 즉, Petri nets 모델링 또한 IDEF3와 마찬가지로 각 프로세스를 수행하는 주체에 대한 모델링 관점을 제시하지는 않는다. 결국

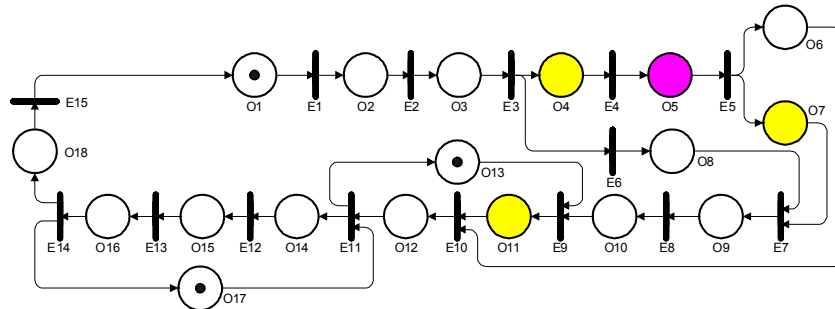


그림 3. Marked Graph를 이용한 금형제작 프로세스 모델의 예.

표 1. <그림 3>의 Operation과 Event

Operation	Description	Event	Description
O1	Waiting for drawing/part list/mold models	E1	Drawings including mold models and part list are arrived
O2	Generation of CAM data	E2	Generating CAM data is completed
O3	Transmission of CAM data and corresponding process info.	E3	CAM data and process info. are arrived
O4	Detailed production scheduling	E4	Detailed production planning is completed
O5	Verification of part drawings and production planning	E5	Decision of outside order or self-production
O6	Production of parts of a mold by a contractor	E6	e (always occurring event)
O7	Detailed production planning	E7	Material and part is deposited and verified
O8	Delivering process information	E8	Processing operations are distributed
O9	Distribution of processing operations	E9	Schedule is rearranged or additional works are completed
O10	Rescheduling	E10	Making parts of a mold is finished or parts produced by a contractor is delivered
O11	Making parts of a mold	E11	Inspection is done or additional finishing touches are completed
O12	Inspection of mold parts	E12	Post-processing is finished
O13	Perform additional works	E13	Parts are assembled into a mold
O14	Post-processing	E14	Final inspection is completed
O15	Assembling parts of a mold	E15	e
O16	Final inspection of a mold		
O17	Perform additional finishing touches		
O18	Finishing up mold production		

<그림 3>에 나타난 Marked Graph 모델의 예는 어느 한 곳의 금형 가공업체의 모델이며 업체 내 여러 부서 간, 또는 다른 업체와의 협업은 표현되지 않는다. Petri nets을 이용한 협업 모델링은 다른 업체의 프로세스 모델 전체를 하나의 Petri nets 모델에 함께 표현하여 연결시킴으로써 업체 간 협업을 표현할 수는 있으나 모델의 크기가 커져 이러한 경우 Petri nets 모델의 관독성은 더욱 떨어지게 된다. Petri nets을 이용한 여러 가지 분석 방법은 Petri nets을 이용하는 중요한 이유 중의 하나이지만 모델링 초기에는 모호하지 않은 다이어그램으로 행위(Behavior)를 명시할 필요가 있다(van der Aalst, 1999).

2.3 Unified Modeling Language(UML)

UML은 객체지향 시스템을 모델링하는 도구 중의 하나로 1997년 Object Management Group(OMG)이 다양한 시스템을 모델링 하는 데 사용하는 사실상의 표준으로 규정하고 있다(Booch *et al.*, 1999; OMG, 2003; Rumbaugh *et al.*, 1999; Pooley and Stevens, 1999). UML은 사용하기 쉽고, 표현력이 강하며, 확장성이 뛰어난 비주얼 모델링 언어이며(Quatrani, 1998), 시스템의 크기와는 관계없이 다양한 관점에서 모델링을 할 수 있다(Ryu *et al.*, 2003).

UML은 크게 구조 다이어그램(Structure Diagrams), 행동 다이어그램(Behavior Model), 그리고 모델 관리 다이어그램(Model Management Diagram)의 세 가지 카테고리를 갖고 있으며 각 카테고리 별로 여러 가지 다이어그램을 지원한다(OMG, 2003). 가령, 행동 다이어그램은 상태 다이어그램(Statechart Diagram), 활동 다이어그램(Activity Diagram), 시퀀스 다이어그램(Sequence

Diagram) 등을 지원한다. 또한 UML은 Object Constraint Language(OCL)을 이용할 수 있게 함으로써 부가적인 관계 조건을 표현할 수 있게 하여 모델링의 표현력을 한층 높일 수 있다(Siikarla *et al.*, 2004).

금형의 설계 및 제작에 앞서 수행되는 제품설계 과정을 UML 활동 다이어그램을 이용하여 모델링한 예는 <그림 4>와 같다. <그림 4>의 UML모델에서 “Design review”, “Final revision of product drawing”, 그리고 “Final evaluation of product design”은 협업 프로세스이다. 그러나 UML에서는 이러한 협업을 표현할 수 있는 별도의 방법을 제공하고 있지 않으므로 각 협업행위의 주체가 누구인지 명확히 표현할 수 없다. 구획선(Swimlane)을 이용하여 각 행위의 주체를 표현할 경우 <그림 4>에서와 같이 구획선의 경계부분에 협업 프로세스를 표현할 수는 있으나 3 곳 이상의 주체가 동시에 참여하는 협업일 경우 이에 대한 표현은 더욱 어려워지게 된다.

2.4 ARIS

ARIS(Architecture of Integrated Information Systems)는 비즈니스 프로세스 모델링을 위한 모델링 기법으로 Event-driven Process Chain(EPC) 방법을 제공한다(Scheer, 1994). EPC 모델링은 ARIS (IDS Prof. Scheer GmbH)를 비롯하여 SAP R/3(SAP AG), LiveModel/Analyst(Intellicorp Inc.), 그리고 Visio(Visio Corp., MicroSoft Corp.)에서 이용되고 있는 모델링 방법으로 기본적으로 논리 커넥터(예: AND, OR, XOR)를 이용한 이벤트(Event)와 기능(Function)의 연결로써 프로세스를 표현한다. eEPC(extended EPC)는 EPC 구성요소에 데이터 등의 비즈니스 오브젝트와 조직 유닛을 추가

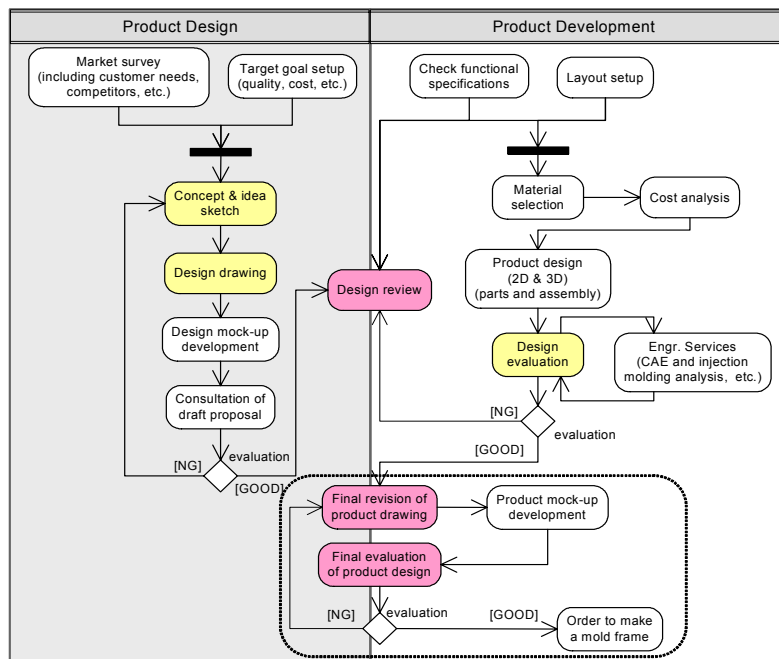


그림 4. UML 활동 다이어그램을 이용한 제품설계 프로세스 모델의 예.

시켜 모델링하는 방법이다. ARIS는 크게 기능(Function), 조직(Organization), 데이터(Data), 결과(Output) 관점과 각 관점에서 정의된 개체들과의 관계를 이용하여 프로세스를 표현하는 컨트롤(Control) 관점의 다섯 가지 뷰(View)를 갖고 있으며 이는 ARIS toolset을 통해 지원된다. 여기서 주목할 것은 바로 조직 관점이며 정의된 기능을 누가 수행하는지 그 수행 주체에 대한 모델링이 가능하다는 것이다. 따라서 다수의 수행 주체가 연관된 협업 프로세스는 ARIS 방법을 통해 모델링이 가능하게 된다. 예를 들어, <그림 4>의 점선으로 표시된 부분을 eEPC 모델로 간략히 표현하면 <그림 5>와 같다(관련 데이터 및 결과물에 대한 객체 표현은 생략됨).

van der Aalst(1999)는 EPC의 Event를 Petri nets의 플레이스로, Function을 트랜지션으로 매핑시킴으로써 EPC 모델을 Petri nets로 변환하는 방법에 대해 연구하였다. 이는 UML 및 IDEF3

과 마찬가지로 ARIS 방법 또한 consistency나 completeness 등 모델에 대한 검증 방법이 없기 때문이다(van der Aalst, 1999). 이와 같은 이유로 여러 모델링 방법에 대해 Petri nets로의 변환 및 보완 사용에 대한 많은 연구가 이미 수행된 바 있다(2.5절 참조). 그러나 그의 연구는 EPC 모델에 대한 변환에 관한 것으로 수행 조직은 고려되지 않았다. 협업 프로세스에 대한 Petri nets 변환 시 해당 프로세스와 관련된 선·후행 프로세스는 관련 조직에 따라 다르며 각 수행 조직의 정책 또한 다르므로 이에 대한 고려가 반드시 필요하다.

2.5 프로세스 모델링 방법 비교

시스템의 프로세스를 모델링하기 위해 사용되는 IDEF3, Petri nets, UML, 그리고 ARIS는 개발의도 및 스펙에 따라 각각

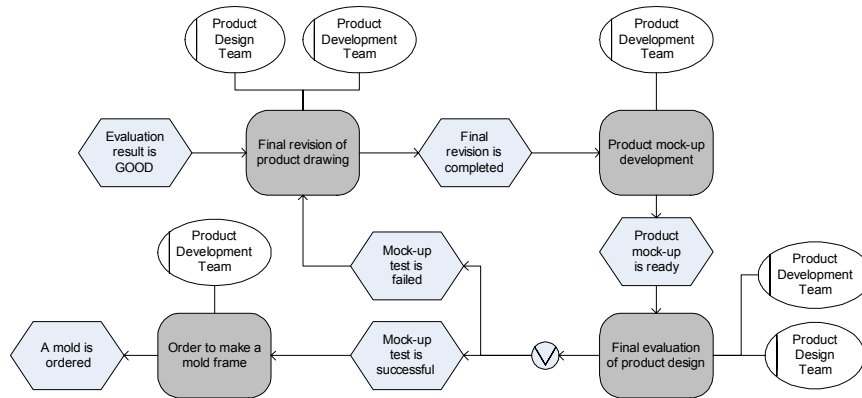


그림 5. eEPC 모델의 예.

표 2. 프로세스 모델링 방법 간의 특성 비교

Criteria	IDEF3	Petri Nets	UML	ARIS
Normal characteristics	Model complexity	Very simple	Fairly simple, but complex when a model size becomes large	Simple
	Understandability of models	Very easy	Mostly difficult	Easy
	Standardization	Strong	Very weak(many versions)	Fairly strong
	Formalism	Not existing or very small(elaboration)	Existing strong formalism	Partially existing(e.g., OCL)
Collaboration Process Modeling	Representability	Not very large	Very large	Fairly large(restricted to diagrams supported)
	- Dynamics	- Limited(temporal relation links and junctions)	- Powerful	- Limited(temporal relations between entities or objects)
	- Collaboration	- Not possible	- Possible (but models normally become very complex)	- Possible but limited(with collaboration diagram)
	- Multiple actors of a behavior	- Not possible	- Not possible	- Limited(with swimlane)
	Model verification	Not possible	Powerful(with formalism)	Not possible
	Model simulation	Not possible	Possible	Not possible

의 장단점이 분명히 존재한다. 일반적인 특징 및 협업과 관련된 각 모델링 방법의 비교 결과는 <표 2>와 같다.

협업 프로세스에 대한 표현력은 ARIS > Petri nets > UML > IDEF3의 순서로 볼 수 있다. ARIS는 기능을 수행하는 다수의 주체를 표현하는 조직 심벌(Organizational Symbol)을 제공함으로써 이를 이용하여 협업 프로세스를 명확하게 표현할 수 있다. 또한 프로세스 시뮬레이션을 통해 프로세스의 수행 시간도 출력하거나 자원 병목 지점 및 취약 지점을 도출하는 등 모델의 유효성을 검사할 수 있으나 가령 Petri nets의 Formalism을 이용한 시스템의 Liveness, boundness, conservativeness, consistence 등과 같이 모델 자체에 대한 검증은 할 수 없다.

한편 Petri nets을 이용한 협업 모델은 이를 관독하고 직관적으로 이해하는 데 있어 다른 모델링 방법에 비해 어렵다는 단점이 있다. 이러한 Petri nets 모델의 단점을 보완하면서 동시에 Petri nets의 모델 분석 방법을 이용하기 위해 UML을 통한 모델링 후 UML 모델을 Petri nets 모델로 변환시키는 연구(Han, 2004; López-Grao *et al.*, 2004)나 ARIS 모델을 Petri nets 모델로 변환시키는 연구(van der Aalst, 1999), 또는 Petri nets을 UML과 함께 사용함으로써 UML의 기능을 보완하는 방법에 대한 연구(Baresi, 2001)가 진행된 바가 있다.

또한 UML만을 이용한 시스템 모델링의 한계를 극복하기 위해 IDEF를 함께 사용하는 방안에 대한 연구(Kim *et al.*, 2003), 혹은 IDEF와 Petri nets을 보완적으로 사용하여 시스템을 모델링하는 연구(Santarek and Buseif, 1998) 등 이들 프로세스 모델링 기법 사이의 보완적 사용에 대해 연구가 수행되어 왔다. 이러한 연구들은 결국 어느 하나의 모델링 방법으로 시스템 전체를 명확히 표현한다는 것은 한계가 존재하므로 이들의 보완적인 사용이 가장 바람직하다는 것을 말한다.

본 논문에서는 협업 프로세스를 명확히 표현하기 위해 UML을 기반으로 한 협업 모델링 방법을 제안하도록 한다. 또한 작성된 협업 모델을 Marked Graph 모델로 변환하기 위한 규칙을 제시함으로써 협업 모델에 대한 Petri nets의 분석기법을 적용할 수 있도록 하고자 한다.

3. 협업 프로세스 모델링(Collaborative Process Modeling; CPM)

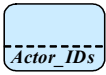
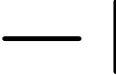

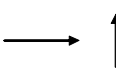
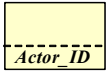
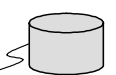
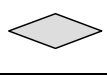
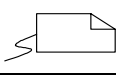
3.1 CPM의 정의, 특성 및 구성요소

협업 프로세스 모델링(Collaborative Process Modeling; CPM)에서 표현되는 프로세스의 종류는 크게 세 가지로 일반 프로세스(Normal Process), 내부 협업 프로세스(Intra-collaboration), 그리고 외부 협업 프로세스(Inter-collaboration Process)이다. 내부 협업은 같은 조직 내 다른 작업 그룹 사이의 협업을 지칭하며(예: 같은 회사의 서로 다른 부서 간의 협업), 외부 협업은 다른 조직과의 협업을 말한다(예: 서로 다른 회사 간의 협업). CPM은 시스템 내에 존재하는 협업 프로세스를 명확히 정의하고 도출할 수 있는 비주얼 모델링 방법이며, 다음과 같은 특성을 갖는다.

- 1) CPM은 프로세스 중심(Process-oriented) 모델링 방법이다(협업 프로세스에 대한 객체 중심의 메시지 모델링 지원).
- 2) CPM의 구성요소는 기본적으로 UML의 구성요소를 바탕으로 하고 있으며 8가지의 구성요소를 갖는다(<표 3>).
- 3) CPM은 일반/내부 협업/외부 협업 프로세스를 별도의 기호를 이용하여 표현하므로 이를 직관적으로 파악하기 쉽다.
- 4) CPM 모델의 각 프로세스는 이를 수행하는 주체가 명시되어 있으며 필요에 따라 하나의 모델 내에 여러 주체의 프로세스를 함께 표현할 수 있다.
- 5) CPM을 통해 완성된 모델은 Marked Graph로의 변환이 가능하며 이를 통한 모델 분석 및 검증이 가능하다.

CPM에서는 <표 3>에서 볼 수 있는 바와 같이 둥근 사각형(Rounded Rectangle)으로 내부 협업 프로세스를, 각진 사각형(Edged Rectangle)으로 외부 협업 프로세스를, 그리고 보통의 사각형(Rectangle)으로 일반 프로세스를 나타낸다. 판단(Decision)을 위해서는 다이아몬드 형태의 표식을, 프로세스 간 동기화(Synchronization)를 표현하기 위해서는 굵은 선을 이용하며, 프

표 3. CPM 구성요소

Symbol	Description	Symbol	Description
	Intra-collaboration process		Horizontal and vertical synchronization
	Inter-collaboration process		Process transition
	Normal process		Resource
	Decision		Reference note

로세스의 전이는 화살표로, 리소스의 표현은 원통으로 나타내며 참고사항은 Reference Note를 이용하여 표기한다. 상태(State)에 대한 정보는 각 프로세스의 흐름으로 파악할 수 있으므로 별도의 표시는 하지 않는다.

CPM을 이용한 협업 모델링은 각 프로세스에 대한 수행 주체를 모두 명시하도록 하여 해당 프로세스 진행에 따른 보고 및 행위의 책임이 누구에게 있는지를 명확히 할 수 있다. 행위의 주체에 대한 표현은 크게 Company, Department, 그리고 Individual의 세 단계로 나누며 각 단계별 정의 및 각 프로세스별 행위자 표현(<표 3>의 Actor_IDs)은 다음과 같다.

- Company : $C = \{c_i \mid i = 1, \dots, n\}$, $D^i \subset C$,
 - Department : $D^i = \{d_j^i \mid d_j^i \in c_i, j = 1, \dots, m\}$, $P^{ij} \subset D^i$,
 - Individual : $P^{ij} = \{p_k^{ij} \mid p_k^{ij} \in d_j^i, k = 1, \dots, l\}$.
- 1) 일반 프로세스 : $c_i(d_j^i(p_k^{ij}))$, 표현 예: $\langle c_1(d^1_3(p^{13}_2)) \rangle$,
- 2) 내부 협업 프로세스 : $c_i(D^i)$, 또는 $c_i(d_j^i(p^{ij}))$, 표현 예: $\langle c_1(d^1_2(p^{12}_3), d^1_3(p^{13}_1)) \rangle$, $\langle c_1(d^1_2(p^{12}_3), p^{12}_4) \rangle$,
- 3) 외부 협업 프로세스 : C , 표현 예: $\langle c_1(d^1_2(p^{12}_3)) \rangle$, $\langle c_2(d^2_1(p^{21}_3), p^{21}_4) \rangle$, $\langle c_1(d^1_1(p^{11}_3), d^1_2(p^{12}_1), p^{12}_3) \rangle$, $\langle c_2(d^2_1(p^{21}_3), p^{21}_4) \rangle$.

3.2 CPM을 이용한 협업 프로세스 모델링의 예

<표 3>에 설명된 CPM의 요소를 이용한 협업 프로세스 모델링의 예는 <그림 6>과 같으며 이는 <그림 2>~<그림 4>에 나타난 “제품설계→금형설계→금형제작”의 전 과정을 나타낸다. 표현의 복잡성을 줄이기 위해 각 프로세스는 약간씩 재구성되었으며, 각 프로세스를 수행하는 행위자는 Department 단계까지만 표현하였다. 모델에 표현된 관련 업체 및 부서는 <표 4>와 같다. 모델에 표현된 프로세스 중 그림자 표시가 있는 프로세스는 IDEF3에서와 마찬가지로 내부에 서브 프

로세스 모델이 존재함을 나타낸다. <그림 6>의 ①~⑥의 표기는 3.3절에서의 모델 변환을 위한 프로세스들을 나타낸다.

표 4. <그림 6>의 금형업체 및 관련 부서

Company/ Department	Description
c_1	Part production company
d^1_1	strategic planning team
d^1_2	part design team
d^1_3	part production team
d^1_4	marketing team
c_2	Mold design company
d^2_1	marketing/technology team
d^2_2	mold design team
d^2_3	purchasing/procurement team
c_3	Mold production company
d^3_1	marketing/strategy team
d^3_2	mold production & management team
d^3_3	quality management team
c_4	Engineering Service Company
d^4_1	engineering service team1

3.3 협업 모델의 Petri nets 모델로의 변환

CPM을 이용하여 작성된 모델은 2.2절에서 설명한 Marked Graph로의 변환이 가능하다. 그러나 총 8개의 구성요소를 갖는 CPM의 구성요소(<표 3>) 각각을 구성요소가 4개뿐인 Marked Graph, 즉 Petri nets의 구성요소에 1:1로 매핑시킬 수는 없다. 따라서 본 논문에서는 Petri nets의 구성요소인 트랜지션, 플레이스, 아크, 토큰의 조합으로 CPM의 구성요소를 표현하는 빌딩 블록(Building Block)을 정의하고 이를 조합하여 사용함으로써

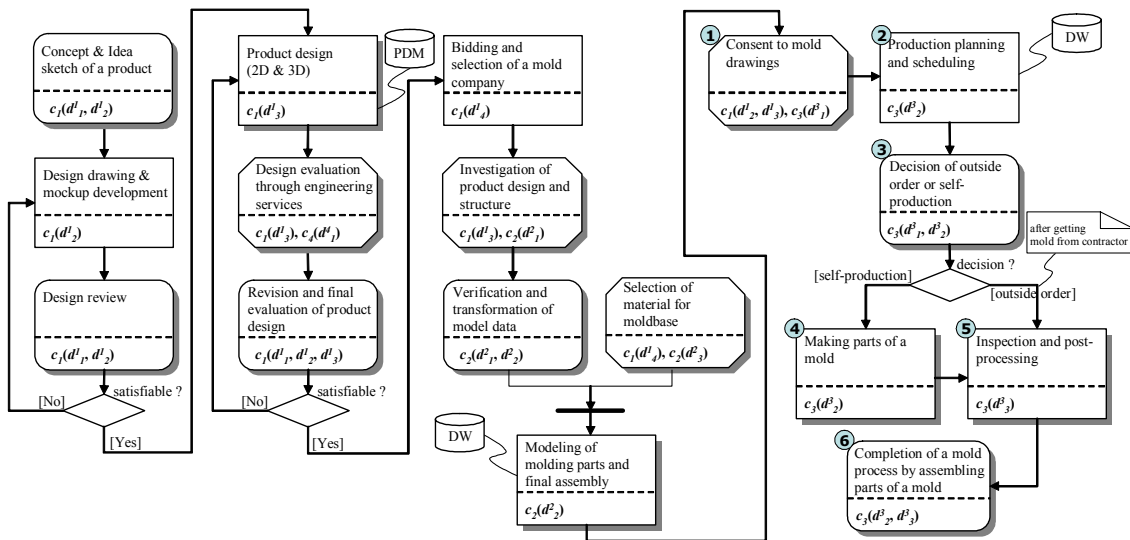


그림 6. CPM을 이용한 금형업체 간 협업 프로세스 모델의 예.

CPM모형을 Marked Graph로 변환시키도록 한다. Marked Graph Building Block(MGBB)은 <표 6>에 정의되어 있으며 CPM모형을 Marked Graph로 변환하는 방법은 다음과 같다.

Rule 1 CPM모형 구성요소 중 MGBB로 정의되지 않는 Reference Note를 제거한다.

Rule 2 변환될 Marked Graph의 수준(Level of detail)을 결정한다. 수준은 Company/Department/Individual의 세 가지 중에 택일한다.

Rule 3 정의된 MGBB를 이용하여 각 요소를 Petri nets 모델로 변환시킨다. 선택된 Marked Graph의 수준이 Company 수준이 아닌 경우의 모델변환 시 현재 행위자의 소속이 다르더라도 상위 수준의 소속이 같은 경우 프로세스 간 연결을 할 수 있으며, 이때 연결되는 행위 주체 사이에 “split-type” 또는 “join-type”을 추가하여 소속 그룹 별도의 프로세스 모델 추출을 가능하게 한다 (decision 또는 ending mark와의 연결 시는 제외).

각 프로세스의 Marked Graph 변환 시 사용되는 MGBB는 <표 5>와 같다. 여기서, P0 ~ P3은 <표 6>에 정의되어 있다. P1은 P0를 한 단계 확장시킨 MGBB로서 입력 부분에 확장 입력요소(Input for extension)를, 최종 플레이스에 확장 출력요소(Output for extension)를 추가하는 것으로 하위 수준에 소속되어 있는 모든 행위 주체들을 표현할 수 있게끔 한다. 이와 마찬가지로 방법을 통해 P2는 P1으로부터, P3는 P2로부터 정의된다.

표 5. 프로세스의 변환 수준에 따른 MGBB 사용

Level \ Process	Company	Department	Individual
Normal Process	{ P0 }		
Intra-collaboration Process	{ P0 }	{ P1 }	{ P2 }
Inter-collaboration Process	{ P1 }	{ P2 }	{ P3 }

Rule 4 변환된 Petri nets의 마지막 프로세스를 나타내는 빌딩 블록의 플레이스와 첫 프로세스를 나타내는 빌딩 블록의 트랜지션 사이에 엔딩마크를 추가하여 Marked Graph를 완성시킨다. 엔딩마크 추가 시 행위 주체의 소속이 다르더라도 상위 수준의 소속이 같은 경우 프로세스 간 연결이 가능하다.

CPM 모델의 변환 룰과 정의된 MGBB를 이용하여 <그림 6>의 ①~⑥으로 표기된 프로세스를 Company 수준의 Marked Graph로 변환시킨 결과는 <그림 7>과 같다. 이해를 돕기 위해 최종 결과에 각각의 프로세스에 대한 구분과 Company의 구분을 표시하였다. 변환된 Marked Graph의 결과는 원칙적으로 변환 수준에 맞는 Marked Graph의 “그룹”으로 표현된다. 그러나 해당 수준에 맞는 프로세스의 정보가 부족할 경우 그 상위 수

준의 그룹으로 표현될 수 있다. 즉 <그림 7>에서 보이는 결과는 c_3 가 중심이 되어 형성된 하나의 Marked Graph 모델이며 c_3 를 중심으로 단힌 경로를 갖고 있음을 알 수 있다. <그림 6>의 모든 프로세스를 변환시켰을 경우 각각의 Company의 프로세스 정보가 충분하다면 Marked Graph의 변환 결과는 각 Company 중심의 단힌 경로를 갖는 4개(c_1, c_2, c_3, c_4 중심)의 Marked Graph를 얻게 된다. 같은 프로세스를 Department 수준의 Marked Graph로 변환시킨 최종 결과는 <그림 8>과 같다. Individual 수준의 Marked Graph 모델 또한 마찬가지로 방법으로 변환하여 얻을 수 있다.

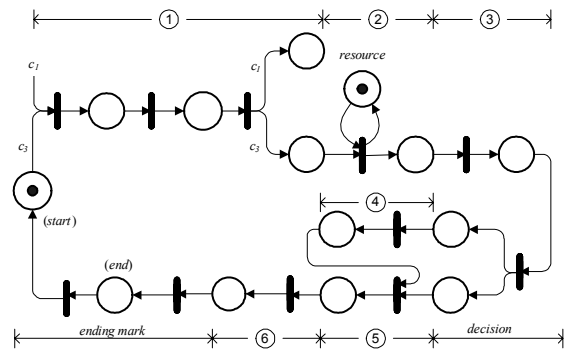


그림 7. Company 수준에서의 변환 결과.

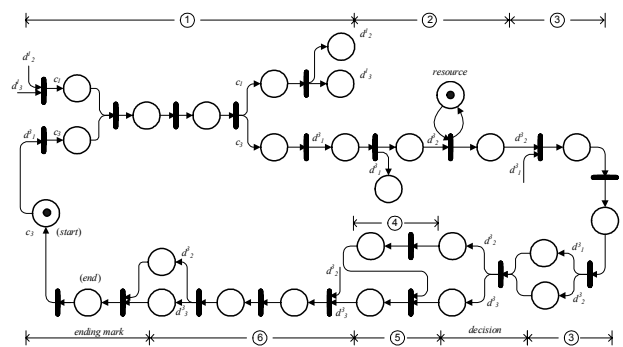


그림 8. Department 수준에서의 변환 결과.

3.4 CPM에서의 제어 메시지

CPM에서는 프로세스 수행을 위한 제어 메시지는 프로세스 트랜지션의 파라미터로 입력된다. 각 프로세스의 입력 메시지는 주로 선행 프로세스가 끝났음을 알리는 메시지 또는 현 프로세스의 응답을 요청하는 메시지이며, 출력 메시지는 현재 프로세스가 끝났음을 후행 프로세스에 알리는 메시지 또는 다른 프로세스의 응답을 요구하는 메시지이다. 예를 들어 <그림 6>의 CPM 모델에서 프로세스 ①에서 ②로의 입력 메시지는 금형 디자인에 대한 협의가 끝났음을 알리는 메시지가 되며 이는 프로세스 ②를 시작하게 한다. 프로세스 ②에서 ③으로 는 생산 계획과 스케줄링이 끝났으므로 c_3 의 생산 계획에 맞추어 금형의 파트를 자체 생산할 것인지 아니면 외주 생산을 할

것인지를 판단하여 그에 맞는 다음 프로세스를 진행하게 된다. 변환된 Marked Graph 모델에서는 각 프로세스의 최종 플레이스들이 해당 프로세스가 끝났음을 알리는 상태를 표현한다. 이와 같이 프로세스 중심의 모델에서는 프로세스 사이의 제어 메시지 흐름이 쉽게 파악되는 장점이 있으나 협업을 위해 수행 주체 사이에 주고받는 메시지는 파악하기 힘들다.


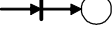
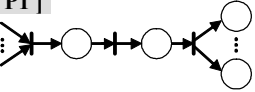
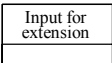
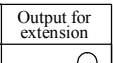

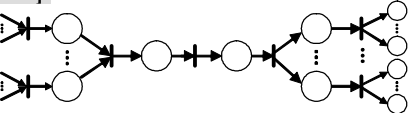

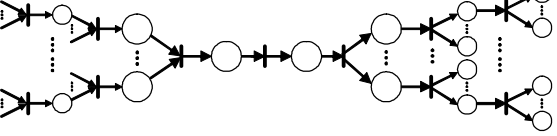

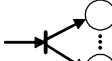
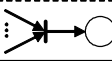

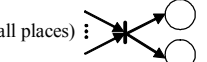

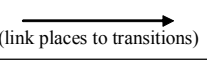

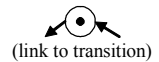
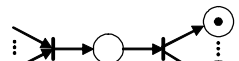
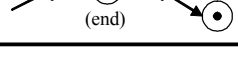
협업 프로세스 수행을 위한 제어 메시지는 프로세스 중심이 아닌 오브젝트, 즉 행위의 주체를 중심으로 이들 사이의 제어 메시지의 흐름을 표현하여 파악할 수 있다. 예를 들어 <그림 9>는 <그림 6>의 프로세스 ①의 수행을 위해 필요한 각 수행 주체들 간의 제어 메시지 흐름을 표현한다. <그림 9(a)>는 각 협업 프로세스 내에 존재하는 행위의 주체 사이의 메시지를 모델링하기 위한 다이어그램이며 <그림 9(b)>는 이에 대한 의미를 설명한다. <그림 9(b)>에서와 같이 각 수행 주체는 입력 메시지에 따라 처리해야 할 세부 프로세스에 대한 모델이 있으며 협업을 위해 중간 중간 필요한 메시지를 다른 수행 주체에 요

청하거나 이를 받아들임으로써 협업 프로세스를 진행하게 된다. 프로세스 중심의 모델링에서는 이와 같은 메시지의 흐름이 잘 보이지 않으므로 CPM에서는 <그림 9(a)>와 같이 각 협업 프로세스에 대해 수행 주체 사이의 메시지 흐름을 모델링할 수 있도록 객체 중심 메시지 모델링을 지원한다. 각각의 협업 프로세스에 대한 객체 중심의 메시지 모델을 통합함으로써 모델에 존재하는 모든 프로세스에 대해 객체 중심의 메시지 모델을 얻을 수 있을 뿐만 아니라, 이를 이용한 프로세스 제어가 가능하다.

3.5 협업 지원 시스템 구현을 위한 CPM의 적용

e-Manufacturing 시범사업에서는 <그림 6>과 같은 CPM을 이용한 협업 모델링을 통해 각 참여업체들 간의 협업 프로세스를 파악하고 이를 액티비티(Activity) 단위로 세분화하여 업무 템플릿을 도출하였다. 업무 템플릿이란 해당 액티비티를 수행함

표 6. Marked Graph Building Block(MGBB)

Symbol	Marked Graph Building Block
 Normal process	[P0]  [P1]   
 Intra-collaboration process	[P2] 
 Inter-collaboration process	[P3] 
 Synchronization	 (Split-type)  (Join-type)
 Decision	 (all places)
 Process transition	 (link places to transitions)
 Resource	 (link to transition)
Ending mark (additional)	 (start)  (end) - link to the first transitions

에 있어 필요한 표준 문서들을 모아놓은 양식들의 집합을 말한다. 모델링된 프로세스들은 폴더 형태로 시스템에 반영되었으며 각각의 프로세스 수행을 위해 필요한 표준 양식들은 개별 파일 형태로 각 폴더 내에 저장되어 이를 이용하게 된다. 내부 협업을 위한 업무 템플릿의 구현 예는 <그림 10>과 같으며 이러한 업체들 간의 협업은 협업 Hub 시스템을 통해 온라인상으로 이루어지게 된다.

4. 결론

현존하는 모든 시스템에서는 상호 협력적인 관계를 요하는 프로세스가 반드시 존재하며 이를 명확히 표현하고 분석할 수 있는 도구가 필요하다. e-Manufacturing뿐 아니라 현존하는 대부분의 프로젝트 진행은 곧 이러한 협업 프로세스를 수행해 나가는 것으로 해석할 수 있을 정도로 대부분의 프로세스가 협업을 필요로 한다. 따라서 본 논문에서는 우선 기존의 프로세스 모델링 기법들(IDEF3, Petri nets, UML, ARIS)을 비교·분석해 봄으로써 협업 프로세스 모델링이 가능한지의 여부를 알아보았다.

Petri nets과 UML은 어느 정도 협업을 표현할 수 있기는 하나 모델링 자체가 어렵거나 협업을 표현하기 위한 능력이 충분하지 않다. 이것은 일반적인 프로세스 모델이 보통 하나의 행위의 주체가 수행하는 프로세스를 묘사하고 있으며 이러한 관점은 협업 프로세스를 나타내는 데 한계가 있기 때문이다. 이에 반해 ARIS 모델링 방법은 각 프로세스의 주체를 표현할 수 있어 해당 프로세스가 협업을 요하는지 아닌지에 대한 여부를 명확히 표현할 수 있다.

그러나 ARIS 방법은 모델에 대한 검증 방안을 제공하지 않으며 협업을 표현할 수는 있으나 협업의 구분이나 협업을 위한 별도의 해석 방안이 존재하지 않는다. 따라서, 본 논문에서는 업체 간 또는 부서 간에 발생하는 협업을 명확히 파악하고 이를 모델링하기 위한 방법으로 협업 프로세스 모델링(Collaborative Process Modeling; CPM) 방법을 제안하였다. 협업의 형태에 따라 프로세스를 일반/내부협업/외부협업 프로세스로 구분하여 형태에 따른 MGBB를 사용하여 이를 Marked Graph로 변환할 수 있게끔 하였다.

CPM은 기본적으로 UML의 구성요소를 응용하여 정의되었으며 IDEF3와 같이 단순하지만 강력한 프로세스 표현력이 있

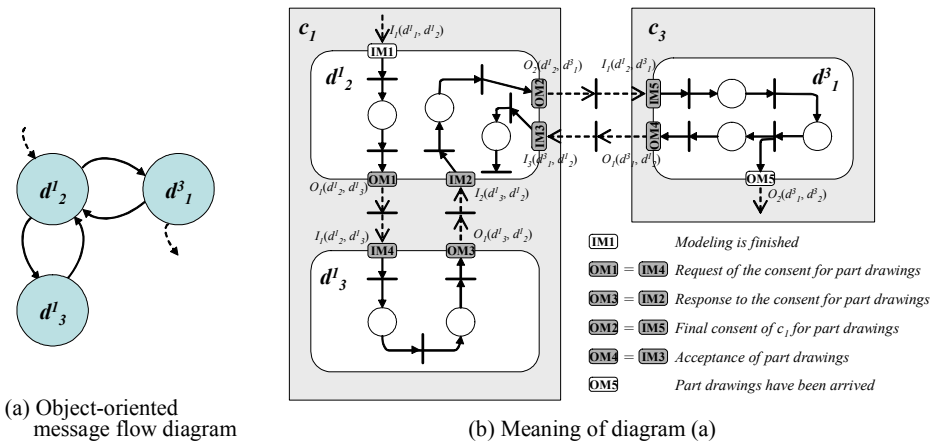


그림 9. 협업 프로세스에 대한 수행 주체 중심의 메시지 모델링의 예.

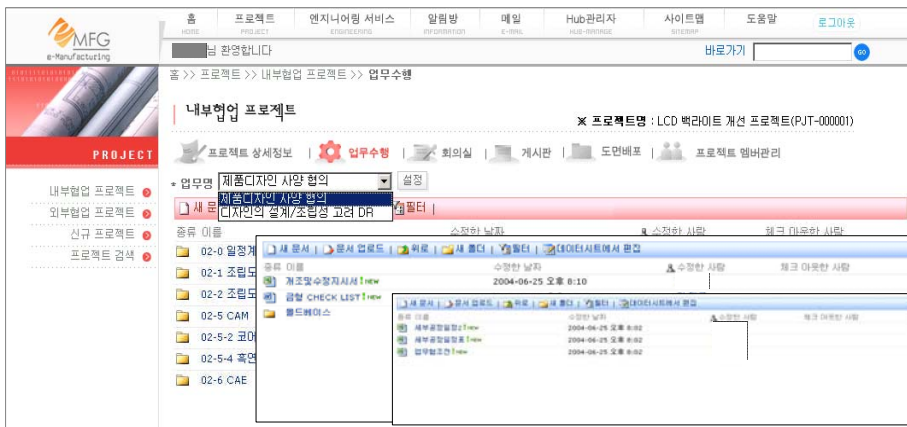


그림 10. 내부협업을 위한 업무 템플릿의 구현 예.

고 또한 Marked Graph로의 변환이 가능하여 Petri nets의 여러 수학적 분석 또한 가능하게 한다. 그러나 본 논문에서는 직접적인 모델 분석 내용은 일반적인 Petri nets 모델 분석 과정과 같으므로 생략하였다. 이러한 CPM을 이용한 협업 모델링은 모델링 자체가 쉽고 모델을 이해하기 쉬우며 시스템 내에 존재하는 업무 프로세스는 CPM의 8가지의 구성요소를 이용하여 모델링함으로써 각 프로세스를 담당하는 행위의 주체가 누구인지를 명확히 파악할 수 있을 뿐 아니라 모델을 통해 프로세스의 흐름에 따라 누구와의 업무 협조가 필요한지 미리 파악할 수 있으므로 생산 스케줄 및 생산 공정의 흐름을 더욱 원활하게 진행할 수 있다. CPM의 기본원칙은 바로 프로세스 중심의 모델링(Process-oriented Modeling)이라는 것이다. CPM은 프로세스 중심의 모델을 세우고 해당 프로세스와 관련된 행위의 주체를 포함함으로써 여러 주체 간에 발생하는 모든 행위를 쉽게 표현할 수 있으며 주체 간 독립적인 프로세스 모델이 존재할 경우 손쉽게 CPM 모델로의 통합이 가능하다. 반대로 하나의 CPM 모델로부터 각 주체별 모델의 추출 또한 가능한데, 이는 CPM 모델에서 해당 프로세스를 직접 추출하거나 또는 Marked Graph로 변환시켜 얻어지는 Marked Graph의 그룹 중 해당하는 부분을 취함으로써 가능하다. 또한 CPM에서는 협업 프로세스를 수행하는 주체 사이에 존재하는 제어 메시지를 모델링할 수 있게 함으로써 협업 프로세스 진행에 따른 메시지의 흐름을 쉽게 파악하고 이를 이용할 수 있게 한다.

본 논문에서 제안한 CPM은 협업 프로세스를 모델링하는데 사용하기 쉽고, 프로세스 표현이 명확하며, 모델에 대한 이해가 쉽다. 그러나 아직 개념적으로만 설계된 모델링 기법이므로 앞으로 CPM의 Formalism에 대한 연구 및 CPM 전용 소프트웨어의 개발, 그리고 이와 더불어 Marked Graph로 자동 변환하는 모듈 및 Petri nets을 이용하여 할 수 있는 모델의 분석 지원에 대한 연구가 계속되어야 할 것이다.

참고 문헌

Baresi, L. (2001), Improving UML with Petri nets, Electronic Notes in Theoretical

Computer Science, 44(4), 1-13.

Booch, G., Rumbaugh, J., Jacobson, I. (1999), The Unified Modeling Language User Guide, Addison-Wesley.

Han, K. (2004), A Workflow Analysis Using the Transformation of an UML Activity Diagram into a Petri Net, IE Interfaces, 17(2), 200-207.

Kim, C., Weston, R.H., Hodgson, A., Lee, K. (2003), The complementary use of IDEF and UML modelling approaches, Computers in Industry, 50, 35-56.

Lee, J. (2003), E-manufacturing fundamental, tools, and transformation. Robotics and Computer Integrated Manufacturing, 19, 501-507.

López-Grao, J.P., Merseguer, J., Campos, J. (2004), From UML Activity Diagrams to Stochastic Petri Nets: Application to Software Performance Engineering, Proceedings of the 4th International Workshop on Software and Performance (WOSP '04), CA, USA, 25-36.

Mayer, R.J., Menzel, C.P., Painter, M.K., deWitte, P.S., Blinn, T., and Perakath, B. (1995), Information Integration For Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report, Knowledge Based Systems Inc.

NIST (1993), Draft Federal Information Processing Standards Publication 183, Integration Definition for Function Modeling (IDEF0), <http://www.edef.com>.

OMG (2003), Unified Modeling language (UML) Version 1.5, <http://www.uml.org/>.

Pooley, R., Stevens, P. (1999), Using UML Software Engineering with Objects and Components, Addison-Wesley.

Quatrani, T. (1998), Visual Modeling with Rational Rose and UML, Addison-Wesley.

Rumbaugh, J., Jacobson, I., Booch, G. (1999), The Unified Modeling Language Reference Manual, Addison-Wesley.

Ryu, K., Choi, H., Lee, S. (2004), Framework of e-Collaborative Engineering Services for Mold Companies in Korea, Proceedings of IMS International Forum 2004, Italy, 1128-1137(Part 2).

Ryu, K., Son, Y., Jung, M. (2003), Modeling and Specifications of Dynamic Agents in Fractal Manufacturing Systems, Computers in Industry, 52(2), 161-182.

Santarek, K., Buseif, I.M. (1998), Modelling and design of flexible manufacturing systems using SADT and Petri nets tools, Journal of Materials Processing Technology, 76, 212-218.

Scheer, A.W. (1994), Business Process Engineering, ARIS-Navigator for Reference Models for Industrial Enterprises, Springer, Berlin.

Siikarla, M., Peltonen, J., Selonen, P. (2004), Combining OCL and Programming Languages for UML Model Processing, Electronic Notes in Theoretical Computer Science, 102, 175-194.

Thirusangu, K., Rangarajan, K. (1995), A note on the construction of marked graphs, Information Processing Letters, 55, 211-215.

van der Aalst, W.M.P. (1999), Formalization and verification of event-driven process chains, Information and Software Technology, 41, 639-650.



류 광 열

포항공과대학교 산업공학 학사
포항공과대학교 산업공학 석사
포항공과대학교 기계산업공학부 박사
현재: 인시아드(INSEAD) Post-doc 과정
관심분야: e-Manufacturing, Fractal Manufacturing System



조 용 주

아주대학교 컴퓨터공학 학사
아주대학교 컴퓨터공학 석사
연세대학교 정보산업공학 박사과정
현재: 한국생산기술연구원 선임연구원
관심분야: 협업정보시스템, 정보시스템 수준 평가



최 현 중

한양대학교 기계공학 학사
한양대학교 기계공학 석사
하노버공대 생산공학 박사
현재: 한국생산기술연구원 수석연구원
관심분야: e-Manufacturing, 나노부품가공



이 석 우

부산대학교 기계공학 학사
부산대학교 기계공학 석사
부산대학교 기계공학 박사
현재: 한국생산기술연구원 수석연구원
관심분야: e-Manufacturing, 정밀부품가공,
고정밀 가공시스템 운용기술 개발