워크플로우의 EMFG 모델링과 분석

허 후 숙† · 여 정 모††

요 약

IT 산업의 발달로 인해 조직의 자동화된 업무 흐름인 워크플로우에 대한 관심과 연구가 활발해지고 있다. 본 논문에서는 워크플로우 설계시 자주 사용되는 기본 구조와 다양한 병행 처리 구조를 개념적인 설계가 가능한 EMFG(Extended Mark Flow Graph)로 먼저 표현한 후 워크플로우를 모델링한다. 모델링된 EMFG를 시뮬레이션하여 워크플로우의 완료 가능 여부를 알 수 있는 수행완료 가능성 검증 방법과 수행완료 후에 나타날 수 있는 논리적 오류 검증 방법을 제안한다. 제안한 워크플로우의 EMFG 모델링 및 분석 방법을 사용하면 시각적이고 정형적인 설계가 가능하고 직관적인 수학적 해석이 가능하다.

The EMFG Modeling and Analysis for a Workflow

Hu Sook Heo† · Jeong Mo Yeo††

ABSTRACT

Owing to the developed IT industry, many people are more and more interested in Workflow which is an automated business processing system. In this paper, we present basic structures and various parallel processing structures of a workflow using EMFG (Extended Mark Flow Graph) that makes conceptional design available. And then, We design a workflow with EMFG. We propose the commitability verification method to check whether a workflow can complete or not and the logical bug verification method to find whether invalid components exist or not by using simulation result of it. The proposed EMFG modeling and analysis method for a workflow make it possible to become a visual and formal design. Also, it makes intuitional and mathematical analysis possible.

키워드 : 확장된 마크흐름선도(EMFG : Extended Mark Flow Graph), 워크플로우(Workflow), 패트리넷(Petri Net), 모델링(Modeling)

1. 서 론

IT 산업의 급격한 발전으로 기업의 시스템은 자주 변화하고 조직 규모의 방대화로 수행해야 할 업무의 형태도 복잡해졌다. 이러한 환경에 대응하기 위해서 시스템의 유연성이 요구되어지고 있지만 기존의 업무 시스템은 환경이 바뀌어 업무 절차가 바뀌게 되면 그 시스템은 더 이상 의미가 없어지게 된다. 따라서 이러한 변화에 좀더 유연하게 대처할 수 있는 워크플로우(Workflow)[1]에 대한 관심이 높아지고 있다. 워크플로우는 조직의 자동화된 업무 흐름을 말하며, 목적을 달성하기 위한 여러 개의 태스크(task)들로 구성되어 있다. 이러한 워크플로우의 수행을 컴퓨터와 통신망을 이용하여 체계적으로 정의하고, 생성 관리하는 시스템을 WFMS(Workflow Management System : 워크플로우 관리시스템)이라 한다. WFMS는 기업 내외의 업무들과 관련된 사람들 그리고 정보 및 기타 자원들의 흐름을 통합적으로 관리, 지원하는 자동화 도구들의 집합이라고 할 수 있다[2-3]. WFMS를 구축함으로써 이에 따른 비용과 시간을 절감할 수 있으며 외국에서는 제조, 금융, 의료 등의 분야에서 폭 넓게 도입되어 기업의 생산성 향상 도구로서 자리를 잡았고, 국내에서도 제조, 금융 등의 분야에서 적극적으로 도입 되고 있다[1-4, 9-11].

워크플로우의 모델링에는 자료흐름도, 상태 전이도 등 비정형적이고 도형적인 명세모형들이 사용되어 왔다. 그러나 이러한 도구들은 비형식적이고, 정적인 업무의 흐름만 표현할 뿐 워크플로우의 수행가능성이나 데드록(deadlock) 등의 동적 특성들을 분석할 수 없었다. 따라서 정형적이고 도형적이며 수행가능적인 모델링 툴이 요구[15]된다. 이러한 점을 보완한 것으로 페트리넷(Petri Net)을 이용한 워크플로우 모델링 연구가 있다[5-9]. 그러나 페트리넷을 이용한 모델링에서는 워크플로우 차트나 워크플로우 그래프를 먼저 명세한 후에 다시 페트리넷으로 모델링[12-13, 15] 함으로써 많은 시간과 비용이 요구된다. 또한 병행적 수행을 모델링함에 있어 하나의 태스크를 세 개의 플레이스(place)와 임시 플레이스로 표현[15]함으로써 구조를 복잡하게 하고 수행시간을 길게 하며 직관적 이해를 어렵게 한다.

본 논문에서는 개념적 설계와 시각적 표현이 가능하고 정형적이며 동적 특성을 분석할 수 있는 EMFG(Extended Mark Flow Graph)를 이용하여 워크플로우를 모델링 하고

† 준 회 원 : 부경대학교 교육대학원 전산교육전공
†† 정 회 원 : 부경대학교 전자컴퓨터정보통신공학부 교수
논문접수 : 2003년 4월 14일, 심사완료 : 2003년 8월 20일

자 한다. 모델링 된 EMFG를 이용하여 워크플로우의 수행 완료 가능성과 수행 완료하더라도 나타날 수 있는 논리적 오류에 대하여 검증한다.

본 논문의 2장에서는 워크플로우와 EMFG에 대한 관련 연구를 알아보고 3장에서는 워크플로우에서 일어나는 기본적인 구조와 태스크간의 다양한 병행처리 구조를 EMFG로 표현한다. 4장에서는 워크플로우를 모델링 한 EMFG에 대해 시뮬레이션을 수행하여 수행 동작을 알아보고, 해당 워크플로우의 수행 완료 가능성 검증과 논리적 오류의 존재 여부를 검증한다. 마지막으로 5장에서는 본 논문에 대한 결론을 기술한다.

2. 관련 연구

2.1. 워크플로우

2.1.1 워크플로우의 정의

워크플로우는 사전에 정의된 절차와 규칙에 따라 한 담당자에서 다른 담당자로 전달되는 문서, 정보, 과제의 흐름을 자동화하는 것으로 기업 내의 비즈니스 프로세스의 성능을 극적으로 향상시킬 수 있다[3]. 초기의 워크플로우는 전자문서교환을 통한 종이없는 사무실을 구현하는 것이 목적이었으나 최근의 워크플로우는 기업, 공장, 정부 단체 등의 거의 모든 업무를 지원하며 조직내 또는 조직 간의 핵심 프로세스 상에서 발생되는 대규모 트랜잭션을 처리하는 단계로 발전하며 변화하고 있다[16]. 워크플로우의 도입은 첫째, 업무 프로세스 구축 및 유지보수 비용을 대폭 절감시켜 주고, 둘째, 업무 흐름을 가시화하고 그에 따라 모니터할 수 있기 때문에 업무의 생산성을 향상시켜 준다. 셋째, 신규 직원의 경우에도 자신의 업무를 쉽게 파악할 수 있어 업무 투입에 필요한 교육비용을 줄일 수 있으므로 전체적인 생산성을 향상시킬 수 있다. WfMC(Workflow Management Coalition)는 워크플로우와 관련된 표준화를 담당하는 기관으로써 서로 다른 제품들 간의 상호교류를 위한 인터페이스를 개발한다. 여기서 규정하고 있는 인터페이스 규약만을 준수하면 다른 응용프로그램들을 손쉽게 결합할 수 있으므로 통합성[3]을 높일 수 있다.

2.1.2 워크플로우 시스템

WFMS는 워크플로우의 생성, 실행, 관리를 지원하는 여러 가지 도구로 이루어진 시스템으로 정보의 흐름과 관련된 기업 또는 부서 및 관련자들간의 업무 연계를 돕고, 다양한 작업 활동의 순서와 관련된 사람간의 정보기술 자원을 공유함으로써 업무 프로세스의 자동화를 제공한다. WFMS의 구조는 프로세스를 정의하는 정의 도구, 워크플로우를 처리하는 워크플로우 엔진, 데이터 저장 시스템 및 사용자 인터페이스와 사용자 응용프로그램 등으로 구성되어 있으며 프로세스 모델링과 워크플로우 명세화 단계, 프로세스 리엔지니어링 단계, 워크플로우 구현과 자동화 단계로 구분된다[17]. WFMS의 구성요소로는 작업들의 집합인 워크플로우, 작업의 처리를 나타내는 태스크, 작업의 대상, 결과물을 나타내는 처리 객체(Manipulated Object), 정보시스템의 서비스를 나타내는 역할(Role), 워크플로우를 실행하는 담당자인 작업자가 있다[16].

2.2 EMFG

2.2.1 EMFG의 정의

동기나 비동기의 이산제어 시스템을 설계하여 구현하거나 분석하는데 적합한 마크흐름선도를 개선한 EMFG는 박스(box), 트랜지션(transition), 아크(arc)들로 구성되는 마크를 갖는 방향성 선도로 정의된다. 박스는 상태(개념적인 상태, 제어 상태 등)를 나타내며, 상태의 만족 여부는 박스 내에 마크를 두어 표시한다. 즉, 박스의 상태가 만족되면 마크를 두고, 만족되지 않으면 마크를 두지 않는다. 트랜지션은 박스(들)의 상태가 조합되어 박스 자신의 상태가 변화하거나 다른 박스(들)의 상태를 변화시키는 곳, 즉 전이가 일어나는 곳으로 일반 트랜지션과 시간 트랜지션이 있다. 여기서 전이가 일어나는 과정을 트랜지션이 점화(fire)한다고 한다. 아크는 박스와 트랜지션 사이에서 트랜지션의 점화 조건을 결정하고, 트랜지션이 점화될 때 박스의 마크 상태를 결정한다. 아크의 종류에는 일반아크와 역아크, 조건아크 등이 있다.

2.2.2 EMFG의 동작

트랜지션의 점화는 트랜지션의 모든 입력 박스(들)의 마크 상태가 모두 조건에 부합되어야 한다. 이 점화조건들은 연결된 아크의 종류에 따라 달라진다. 즉 일반/역 아크로 연결된 입력 박스에는 마크가 있/없어야 점화 조건에 부합된다.

트랜지션이 점화하는 경우, 트랜지션에 연결된 박스의 마크 상태는 연결된 아크의 종류에 따라 다르게 변화한다. 즉, 일반아크로 연결된 입력 박스의 마크는 소멸되고, 트랜지션에서 일반/역 아크로 연결된 출력 박스에는 마크가 없/있으면 생성/소멸되고, 마크가 있/없으면 그대로 유지[20-23] 된다.

3. 워크플로우의 EMFG 표현

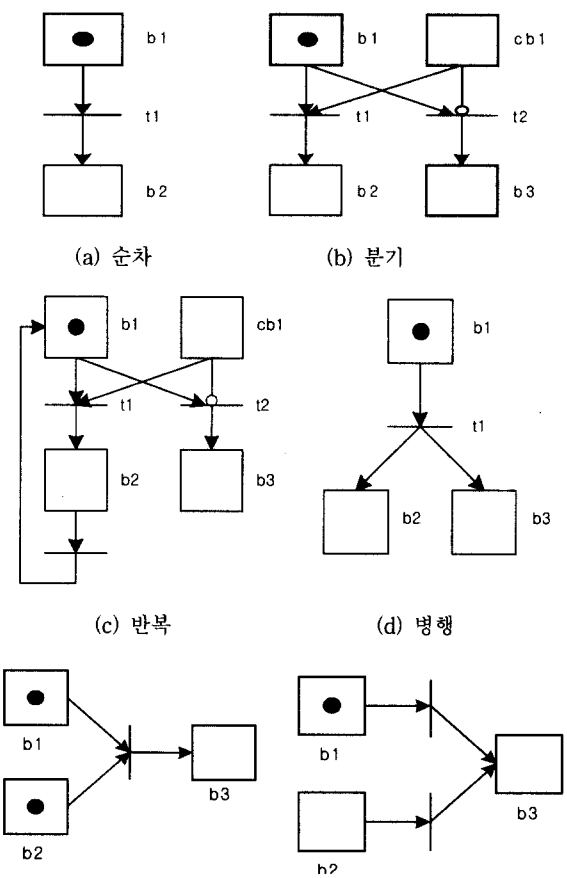
3.1 기본 구조에 대한 EMFG 표현

개념적 설계와 직관적 해석이 용이한 EMFG를 이용하여 워크플로우를 모델링하고자 한다. 워크플로우에서 자주 사용되는 기본적인 구조를 EMFG로 먼저 표현해 두고 설계를 한다면 더욱 빠르고 쉽게 설계할 수 있으며, 표현된 EMFG도 해석과 분석이 용이할 것이다.

워크플로우를 EMFG로 표현하기 위해서는 워크플로우의 각 구성 요소들을 EMFG의 구성 요소로 변환하여야 한다. 하나의 태스크는 EMFG의 박스로 표현하고 다음 태스크로 진행되기 위한 조건은 아크로 나타내며 이전 태스크가 완료하고 다음 태스크로 진행하는 것은 트랜지션이 점화하여

마크가 이동하는 것으로 나타낸다.

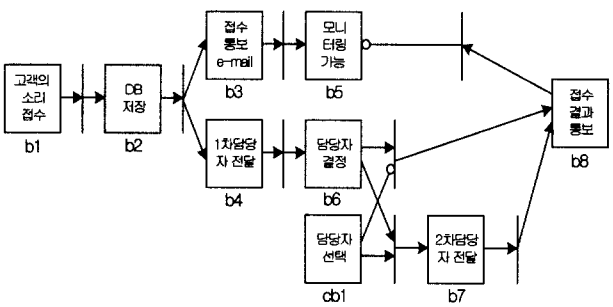
워크플로우를 구성하는 태스크들의 수행은 순차, 분기, 반복, 병행 구조를 가진다. 순차 수행은 하나의 태스크가 완료된 후에 다음 태스크가 진행되는·것을 의미하며, 분기 수행은 주어진 조건에 따라서 서로 다른 태스크가 수행됨을 의미한다. 반복 수행은 태스크들의 주어진 횟수만큼 반복적으로 수행하는 것을 의미하며, 병행 수행은 두 개 이상의 태스크를 동시에 수행함을 의미한다. 태스크간의 연결 형태에는 모든 조건이 만족하면 다음 태스크로 진행하는 AND 연결 형태와 여러 조건들 중에서 하나의 조건만 만족하면 다음 태스크로 진행하는 OR 연결 형태가 있다.



(그림 1) 워크플로우 기본 구조의 EMFG 표현

(그림 1)은 워크플로우 기본 구조를 EMFG로 표현한 것으로 (그림 1)(a)는 순차수행을 나타낸다. (그림 1)(b)에서 cb1은 조건이 만족되면 마크가 생성되고 조건이 만족되지 않으면 마크가 생성되지 않도록 하는 조건 박스(condition box)를 나타낸다. 따라서 조건이 만족되면 t1이 점화하고, 조건이 만족되지 않으면 t2가 점화하게 된다. (그림 1)(c)는 cb1의 조건이 만족되면 b1과 b2가 반복 수행되며 조건이 만족되지 않으면 b3을 수행한다. (그림 1)(d)는 t1이 점화하면 b2와 b3이 동시에 수행하는 것을 나타내며 (그림 1)(e)와 (그림 1)(f)는 워크플로우의 AND와 OR 연결형태를 EMFG로 표현한 것이다.

(그림 2)는 고객으로부터 의견을 받아 담당자에게 연결하여 접수 결과를 고객에게 통보해주는 고객지원 절차 워크플로우를 (그림 1)에서 표현한 워크플로우의 기본 구조와 연결형태를 가지고 모델링 한 것이다. b1, b2는 순차 구조이며, b2, b3, b4는 병행 구조를 나타낸다. b6, cb1, b7, b8은 분기 구조와 AND 연결 형태를 가지고 있으며, b6, b7, b8은 OR 연결 형태를 나타내고 있다.



(그림 2) 기본 구조의 모델링 예

3.2 병행 처리 구조에 대한 EMFG 표현

워크플로우를 구성하는 태스크들은 태스크들의 특성에 따라 다양한 병행처리 형태를 가지고 있다. 하나의 태스크 t1이 시작된 후에 또 다른 하나의 태스크 t2를 시작하여 병행 수행해야 하거나, t1이 시작한 후에 t2가 수행시작해서 완료하고 나면 t1이 수행완료 되어야 하는 경우가 있다. 이러한 다양한 병행처리에 대하여 EMFG로 표현하고자 한다.

이전 연구에서는 이러한 병행 처리에 대하여 특정 레이블이 표기된 워크플로우 그래프로 표현[12]하거나 또는 워크플로우의 모든 각각의 태스크에 대해 태스크의 시작, 수행, 완료를 나타내는 세 개의 플레이스와 그것들을 연결하는 임시 플레이스로 표현한 페트리넷 모델링[15] 방법이 있었다. 전자의 것은 그래프 자체만으로 이해하기가 쉽지 않을 뿐만 아니라 동적인 해석이 불가능하였다. 후자의 것은 과다한 플레이스로 모델링이 복잡해지고, 또한 동적 해석을 함에 있어 플레이스의 수가 많아 시뮬레이션 수행에서도 많은 시간이 요구되었다.

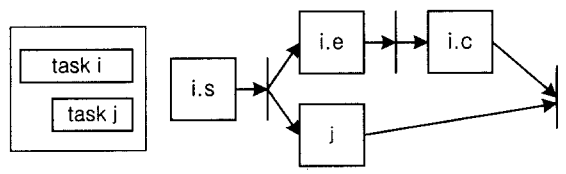
본 연구에서는 이러한 점들을 해결하기 위하여 태스크들 사이에 병행 처리가 존재할 때 특정 태스크 만을 세부적으로 나누어 표현하고자 한다.

[정의 1] 워크플로우를 EMFG로 표현할 때 두 태스크가 병행적으로 수행하는 경우, 태스크들의 정확한 흐름을 표현하기 위해 특정 태스크에 대해 태스크의 시작, 수행, 완료를 나타내는 시작 박스(start box), 수행 박스(execute box), 완료 박스(commit box)로 나누어 표현한다. □

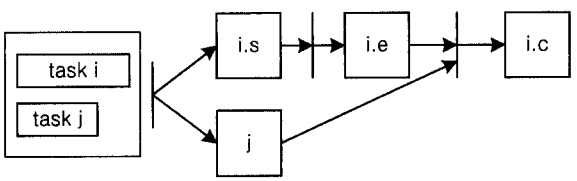
병행 처리되는 두 태스크에서 상대적 수행시간이 긴 하

나의 태스크와 상대적 수행시간이 비교되지 않는 경우에는 두 개의 태스크 모두에 대해서 위에서 정의한 세 개의 박스로 나누어 표현한다. 하나의 태스크에 대해 세 개의 상태를 나타내는 박스로 나누어 표현함으로써 이전에는 복잡하게 표현되어졌던 다양한 병행처리 표현이 간략화 되었고 이해하기 쉬워졌다. 또한 EMFG를 이용하여 표현함으로써 동적 해석이 불가능하였던 것을 해석 가능하게 하였다. 그리고 다양한 병행처리에 대한 구조를 미리 표현해 두어 모델링할 때 사용함으로써 편리성을 가져다 준다.

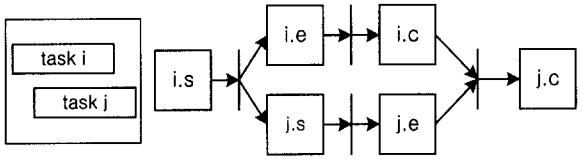
기업 내의 업무에서 일어나는 다양한 병행처리 구조로는 여러 가지 구조가 있을 수 있으나 대표적인 것으로는 순차시작 동시종료(SSPC ; Sequential Starts Parallel Commits), 동시시작 순차종료(PSSC ; Parallel Starts Sequential Commits), 순차시작 순차종료(SSSC ; Sequential Starts Sequential Commits), 중첩된 병행구조(NE ; Nested Execution) 등이 있다. 이들을 EMFG로 표현하면 (그림 3)과 같다.



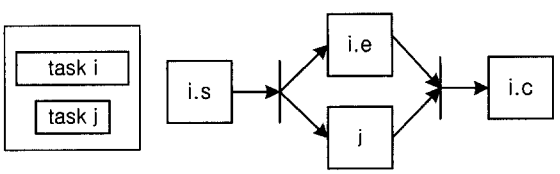
(a) SSPC 구조



(b) PSSC 구조



(c) SSSC 구조



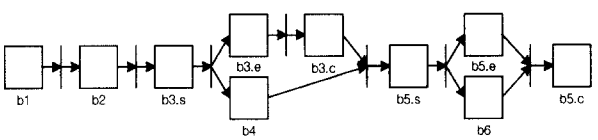
(d) NE 구조

(그림 3) 병행처리 구조의 EMFG 표현

(그림 3)(a)를 보면 task i와 task j가 순차적으로 병행 수행되어 동시에 종료하고 있음을 나타낸다. 여기서는 task i가 task j보다 상대적 수행시간이 길다. 그러므로 task i에 대해서 태스크의 시작을 나타내는 i.s, 수행을 나타내는 i.e, 완료를 나타내는 i.c의 세 개의 박스로 나누어 표현하였다. (그림 3)(b)의 구조는 두 태스크가 동시에 시작하고 task j가 종료하면 task i 종료할 수 있는 구조이다. 따라서 task i가 상대적 수행시간이 더 길기 때문에 task i에 대해 박스를 나누어 표현하였다. (그림 3)(c)의 구조는 task i와 task j의 상대적 수행시간의 비교가 되지 않음으로 두 개의 태스크를 세 개의 박스로 나누어 표현하였다. (그림 3)(d)의 구조는 task i가 시작하면 task j가 시작하고 task j가 종료하면 task i가 종료하는 구조이다. task i의 상대적 수행시간이 더 길기 때문에 task i에 대해 세 개의 박스로 나누어 표현하였다.

다음의 예는 병원에서 환자에 대한 의료서비스 워크플로우를 고려한 것이다. 워크플로우는 환자의 접수에 의해 시작되며 이를 바탕으로 진료 태스크가 수행되고 진료에 따른 검사 태스크로서 임상병리 및 CT(computed tomogram) 검사를 가정할 때, 환자에 대한 임상병리 검사를 시작한 후 CT 검사를 시작한다. 또한 수술 범위를 결정하는 동결조직 검사는 수술 시작 후 시작되고 종료 전에 완료되어야 한다[12].

위의 예를 가지고 (그림 3)에서 표현한 다양한 병행처리 구조를 이용하여 모델링한 것은 (그림 4)와 같으며 각 태스크에 대한 박스 명칭은 <표 1>과 같다.



(그림 4) 다양한 병행처리의 예

<표 1> (그림 4)에 대한 박스명과 태스크

박 스 명	태 스 크	박 스 명	태 스 크
b1	환자 접수	b4	CT 검사
b2	의사 진료	b5	수술
b3	임상병리 검사	b6	동결조직 검사

(그림 4)에서 b3과 b4는 SSPC 구조를 가지고 있으며 b3의 태스크가 긴 수행시간을 가지므로 b3을 세 개의 박스를 나누어 표현하였다. b5와 b6은 NE 구조를 가지고 있으며 b5의 태스크가 b6의 태스크보다 긴 수행시간을 가지므로 b5의 박스를 세 개의 박스로 나누어 표현하였다.

4. 워크플로우를 모델링한 EMFG의 분석

하나의 워크플로우를 EMFG로 모델링한 다음, 이것를 분석하고자 한다.

하나의 워크플로우는 하나 이상의 태스크 집합으로 이루어져 있고, 각각의 태스크들은 지정된 경로로 수행된다. 따라서 마지막 태스크가 완료해야 하나의 워크플로우가 완료되었다고 할 수 있다. 그러므로 하나의 워크플로우에 대해서 시작과 완료를 나타내는 박스를 추가하여 워크플로우 상태를 표현하고자 한다.

[정의 2] 워크플로우를 EMFG로 표현하고자 할 때에는 첫 번째 태스크의 앞에 워크플로우 시작을 나타내는 박스를 추가하여 워크플로우 시작 박스(wfsb : workflow start box)라 하고, 마지막 태스크의 뒤에 워크플로우 완료를 나타내는 박스를 추가하여 워크플로우 완료 박스(wfcb : workflow commit box)라 한다. □

4.1 워크플로우 모델링의 예

전자어음 결제 워크플로우에 대해 워크플로우 모델링을 수행하고자 한다. 현재의 어음 결제는 전자어음보다는 실물 어음을 휴대하여 직접 주고 받는 형태로 대부분 오프라인 상에서 이루어지고 있다. 그러나 전자상거래의 발달로 온라인 상에서 결제가 이루어지는 전자어음의 거래가 확산되고 있다. 전자어음 결제 워크플로우의 업무 흐름은 아래와 같으며 (그림 5)는 이 워크플로우를 EMFG를 이용하여 모델링 한 것이다. 각 태스크에 대한 박스 명칭은 <표 2>에 나타나 있다.

[전자어음 결제 처리 단계]

(업무 당사자)

A : 전자어음 발행회사

B : 전자어음 수취인

C : 전자어음 보관과 결제를 담당하는 금융기관

(업무 흐름 절차)

① A가 전자어음 발행하면 B에게는 발행 사실을, C에게는 보관어음 발생을 알린다.

② B는 발행 사실을 확인하고 보관할 것인지 할인할 것인지를 결정한다.

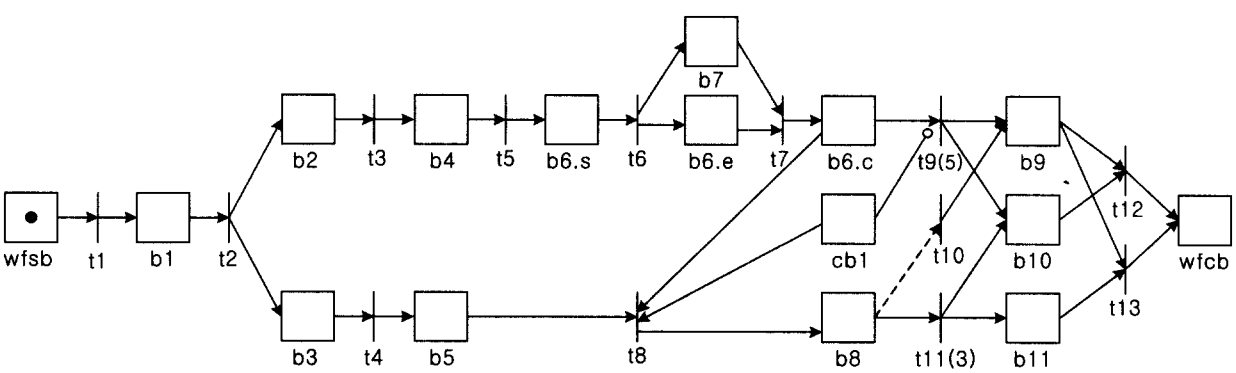
③ 할인 신청 가능 상태가 시작되면 B의 신용조사를 완료한 후에 할인 신청 가능 상태가 완료되도록 한다.

④ C는 보관어음 등록 후 B가 계속 보관한다면 전자어음 만기일에 결제하여 B의 계좌에 입금하고, 할인 신청을 한다면 할인 신청일에 계좌입금하고 만기일에 할인 회수함으로 워크플로우는 종료된다.

(그림 5)에서 wfsb와 wfcb는 워크플로우 시작과 완료 상태를 나타낸다. cb1은 조건 박스로 조건이 만족하면 마크를 생성시키고 그렇지 않으면 마크를 생성시키지 않는다. 따라서 할인 신청 가능 상태이고 B가 할인을 신청한다면 cb1에 마크가 생성된다. 그러므로 b8에 마크가 생성되어 할인 신청일에 B의 계좌로 입금되고 일정 시간 경과 후 만기일에 C는 할인을 회수하여 워크플로우가 종료하게 된다. 만약 할인을 신청하지 않으면 cb1에 마크가 생성되지 않으므로 일정 시간 경과 후 만기일에 B의 계좌로 입금됨으로써 워크플로우는 종료된다. t9(5)과 t11(3)는 시간 트랜지션으로 t9(5)은 만기일을 나타내며 t11(3)은(만기일~신청일)로써 신청일로부터 만기일까지의 기간을 나타낸다. 여기서는 시간을 월 단위로 가정하였으며 만기일은 5개월 후이고, 할인을 신청한다고 가정하였을 경우 신청은 발행일에서 2개월이 되는 날에 신청한 것으로 가정하였다.

4.2 EMFG의 수행완료 가능성 검증과 논리적 오류 검증

워크플로우의 목적은 워크플로우 내의 모든 태스크들이 지정된 경로로 마지막 태스크까지 수행되어 일련의 업무를 완료하는 것이다. 따라서 수행완료를 검증하기 위해서 워크



(그림 5) 전자어음 결제 워크플로우의 EMFG 모델링

〈표 2〉 (그림 5)에 대한 박스명과 태스크

박 스 명	태 스 크	박 스 명	태 스 크	박 스 명	태 스 크
wfsb	워크플로우 시작	b5	보관어음 등록	b9	계좌 입금
b1	전자어음 발행	b6	할인 신청 가능	b10	만기일
b2	수취인 통보 알림	b7	신용조사	b11	할인 회수(대출 감소)
b3	금융기관 전자어음 송부	cb1	할인 신청 여부	wfcb	워크플로우 완료
b4	수취인 통보신호 확인	b8	할인 실행(대출 증가)		

플로우의 수행완료 가능성을 정의한다.

[정의 3] 워크플로우가 지정된 경로상에 있는 태스크들의 수행을 통해 마지막 태스크까지 완료될 수 있다면, 이를 '수행완료 가능성'이 있다고 한다. □

모델링된 EMFG에서 수행완료 가능성을 검증할 수 있는 방안을 제안한다.

[정리 1] 워크플로우를 표현한 EMFG를 가지고 도달 가능한 상태를 표현한 마크벡터에서, wfsb에만 마크가 존재하는 초기 마킹 M_0에서 대해, 마지막 단계에서 wfcb에 마크가 존재하는 마크벡터 M_c가 존재하면 이 워크플로우는 수행완료 가능하다.

증명 도달가능 상태를 나타내는 EMFG의 동작해석 알고리즘의 마크벡터 특성과, 박스에 마크가 존재하면 해당 박스의 상태가 만족함을 의미하는 EMFG 특성에 의해 자명하다. □

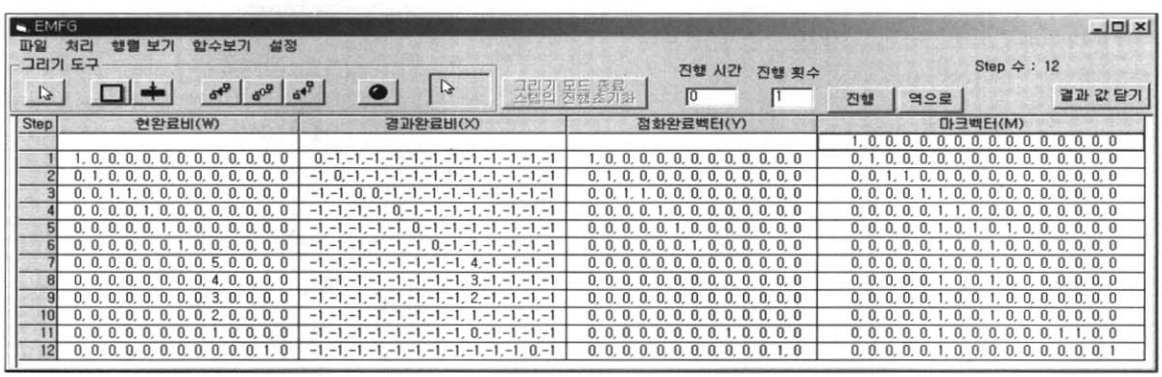
워크플로우 수행완료 가능성을 알기 위해서는 워크플로우 종료 박스인 wfcb에 마크가 존재하는지를 검사한다. 수행완료 가능성 여부를 검증하기 위해서 EMFG의 점화 순서를 나타내는 마크벡터를 이용한다. 모델링된 EMFG에서 워크플로우의 태스크 동작 중에서 모든 동작을 수행하고 난 후에 마크가 존재하게 될 wfcb에 마크가 존재한다면 이 워크플로우는 수행완료 가능함을 나타내며, wfcb를 제외한 나머지 박스에 마크가 존재한다면 이 워크플로우가 수행완료가 되지 않는 경우로 완료될 수 없음을 의미한다.

(그림 5)의 EMFG를 시뮬레이션하면 결과는 (그림 7)과 같다. 먼저 EMFG의 박스 집합 B = {wfsb, b1, b2, b3, b4, b5, b6.s, b6.e, b6.c, b7, cb1, b8, b9, b10, b11, wfcb}에서 wfsb에 마크를 주어 초기 마크 벡터 M_0 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]이 된다. (그림 6)의 (a)와 (그림 6)(b)의 마크벡터의 마지막 단계를 보면 wfcb에 마크가 존재하므로 이 워크플로우는 [정리 1]에 의해 수행완료 가능하다는 것을 알 수 있다.

그러나 (그림 6)(a)에서 마지막 단계의 마크벡터에서 wfcb에 마크가 존재하여 워크플로우의 수행완료 가능함은 알 수 있지만 b5에도 마크가 존재하는 것을 볼 수 있다.

EMFG의 워크플로우 모델링에 있어 특정 박스에 마크가 존재하는 것은 그 박스에 어떠한 상태가 수행되고 있음을 의미한다.

[정의 4] 워크플로우가 지정된 경로상의 태스크들의 수행을 통해 수행완료는 되었으나 해당 워크플로우 내의 특정



(a) cb1 = 0 인 경우

(b) cb1 = 1 인 경우

(그림 6) (그림 5)의 시뮬레이션 결과

태스크 수행이 남아 있다면, 이것을 워크플로우의 '논리적 오류'라고 한다. □

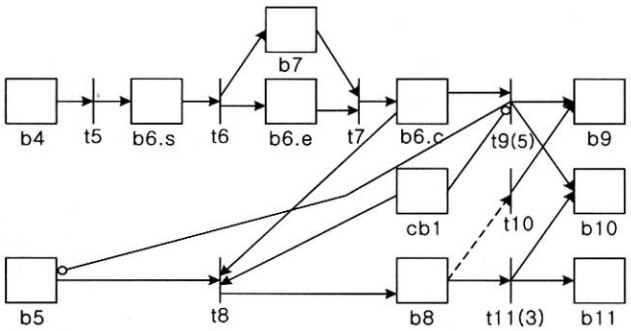
모델링된 EMFG에서 워크플로우의 논리적 오류을 검증할 수 있는 방안을 제안한다.

[정리 2] 워크플로우를 표현한 EMFG를 가지고 도달 가능한 상태를 나타내는 마크벡터에서 wfsb에만 마크가 존재하는 초기 마킹 M_0에서 대해, 마지막 단계에서 wfcb에 마크가 존재하고 그 이외의 박스에도 마크가 존재하는 마크벡터 M_c가 존재하면 이 워크플로우는 논리적 오류가 존재한다고 한다.

증명 도달가능 상태를 나타내는 EMFG의 동작해석 알고리즘의 마크벡터 특성과, 워크플로우가 완료하면 그 안에 속해있는 모든 태스크는 수행 완료되었다는 워크플로우의 특성에 의해 자명하다. □

따라서 (그림 6)(a)에서는 b5에 워크플로우의 논리적 오류가 존재한다고 볼 수 있다. 전자어음 결제 워크플로우는 수행완료 가능성 검증으로 인해 종료되었다. (그림 5)에서 할인을 신청하지 않는 경우 만기일에 계좌에 입금되어 워크플로우는 종료하였다. 그러나 계좌 입금일에 보관어음 등록은 삭제되어야 함에도 불구하고 계속 기록이 남아 있음을 발견할 수 있다. 따라서 이러한 논리적 오류를 제거하기 위해서는 할인 신청을 하지 않으면 만기일에 계좌 입금과 동시에 보관어음 등록을 해제해야 하므로 t9(5)에서 b5로 향하는 역아크를 추가하면 된다. 즉, 논리적 오류가 존재하는 박스와 그와 관련된 트랜지션을 찾아 역아크를 추가하여 마크를 소멸시킴으로써 오류가 제거된다.

(그림 5)의 오류을 제거한 EMFG는 (그림 7)와 같다. (그림 8)은 (그림 6)(a)의 경우에 대해 오류 제거 후의 시뮬레이션 상태를 나타낸 것으로 마지막 단계의 마크벡터를 보면 wfcb에 마크가 존재하고, wfcb를 제외한 어떠한 박스에도 마크가 존재하지 않으므로 이 워크플로우는 수행 완료가능하며 또한 어떠한 논리적 오류도 존재하지 않음을 알 수 있다.



(그림 7) (그림 5)의 오류가 제거된 EMFG 모델링

5. 결 론

본 논문에서는 업무의 자동화를 의미하는 워크플로우에 대해 개념적 모델링 도구인 EMFG를 이용하여 모델링한 후 수행완료 가능성과 논리적 오류를 검증하는 방법을 제안하였다. 이로써 이전의 모델링 작업에서 동적분석이 불가능하거나 많은 수의 플레이스 생성으로 인한 문제가 해결된다.

EMFG를 이용한 워크플로우 모델링에서는 기본적인 구조와 다양한 병행처리 구조를 미리 정의해 놓음으로써 설계를 더욱 쉽고 신속하게 하도록 하였으며 직관적이고 개념적인 설계를 가능하게 하였다. 또한 모델링된 EMFG의 시뮬레이션 동작은 마크의 이동을 단계별로 수행함으로써 설계의 잘못된 부분를 직접 찾아낼 수 있다. 그리고 워크플로우의 수행완료 가능성을 검증함으로써 수행완료 도달 여부를 바로 알 수 있고, 논리적 오류를 검증함으로써 수행완료 후에 나타날 수 있는 문제를 미리 발견함으로써 해당 워크플로우가 올바르게 설계되었는지를 알 수 있다.

따라서 워크플로우를 EMFG로 설계하였을 때 시간과 비용을 절약할 수 있으며, 기존의 워크플로우의 수정하거나 보완할 경우 이미 모델링되어 있는 것을 부분 수정하여 사용함으로써 재사용성을 높일 수 있다.

Step	경과완료비(X)	점화완료벡터(Y)	마크벡터(M)
			1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
1	0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1	1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
2	-1, 0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1	0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
3	-1,-1, 0, 0,-1,-1,-1,-1,-1,-1,-1,-1,-1	0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
4	-1,-1,-1,-1, 0,-1,-1,-1,-1,-1,-1,-1,-1	0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0
5	-1,-1,-1,-1,-1, 0,-1,-1,-1,-1,-1,-1,-1	0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0
6	-1,-1,-1,-1,-1,-1, 0,-1,-1,-1,-1,-1,-1	0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0
7	-1,-1,-1,-1,-1,-1,-1,-1, 4,-1,-1,-1,-1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0
8	-1,-1,-1,-1,-1,-1,-1,-1,-1, 3,-1,-1,-1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0
9	-1,-1,-1,-1,-1,-1,-1,-1,-1, 2,-1,-1,-1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0
10	-1,-1,-1,-1,-1,-1,-1,-1,-1, 1,-1,-1,-1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0
11	-1,-1,-1,-1,-1,-1,-1,-1,-1, 0,-1,-1,-1	0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0
12	-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1, 0,-1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1

(그림 8) 오류 제거 후의 시뮬레이션

참 고 문 헌

[1] A. Elmagarmid and W. Du, "Workflow Management : State of the Art vs. State of the Market," Advances in Workflow Managment System and Interoperability, A. Dogac (Ed), NATO, pp.1-17, 1997.

[2] D. Hollingsworth, "Workflow Management Coalition - The Workflow Reference Model," TC00-1003 issue 1.1, 1994.

[3] Work Group1, "Interface 1 : Process Definition Interchange Process Model," Workflow Management Coalition Specification, TC-1016, 1998.

[4] P. C. Attie, M. P. Singh, A. Sheth and M. Rusinkiewicz, "Specifying and Enforcing Intertask Dependency," VLDB conference, pp.134-145, 1993.

[5] James L. Peterson, "Petri net theory and the modeling of system," Prentice-Hall, 1981.

[6] Rene David and Hassane Alla, "Petri Nets for Modeling of Dynamic Systems a Survey," Automatica, Vol.30, No.2, pp. 175-202, 1994.

[7] C. S. Hwang and J. M. Lee, "Analysis of Matrix Equation Based on Petri Net for Discrete System Control," Proceedings of the 29th SICE Annual Conference International Session, pp.639-696, July, 1990.

[8] Nasreddine Aoumeur, Gunter Saake, "A component-based Petri net model for specifying and validating cooperative information systems," Data & Knowledge Engineering 42, pp.143-187, 2002.

[9] 김소연, 이강수, "워크플로우 모형화 및 관리시스템", 정보처리논문지, 제3권 제5호, pp.18-30, 1996.

[10] 한동일, 박세권, "워크플로 분석/설계를 위한 모델링 기법 개발", 한국경영정보학회 96 추계학술대회논문집, pp.73-84, 1996.

[11] 권종구, "에이전트 기반의 워크플로우에 관한 연구", 서울대학교 대학원 석사학위논문, 1998.

[12] 정희택, 이도헌, 김문자, 류영철, "시간 제약을 포함한 워크플로우 모델링 및 검증", 정보처리논문지, 제6권 제2호, pp.261-273, 1999.

[13] 이성호, "페트리넷을 이용한 워크플로우 모델링 도구 개발", 전남대학교 대학원 석사학위논문, 1999.

[14] 신동일, 신동규, "워크플로우 관리 시스템의 설계 및 구현", 정보처리논문지, 제7권 제5호, pp.1609-1619, 2000.

[15] 정희택, 이도헌, "페트리넷을 이용한 워크플로우 명세의 완료 가능성 및 무결성 검증", 정보처리논문지, 제7권 제7호, pp. 2028-2040, 2000.

[16] 홍현기, "워크플로우 시스템 구축을 위한 프로세스 지향적인 방법론에 관한 연구", 한독경상학회, 경상논총, Vol.21, pp. 221-242, 2000.

[17] 이 헌, 박규석, "웹기반 분산워크플로우 관리시스템의 프로세스 엔진 설계", 멀티미디어학회논문지, 제4권 제5호, pp.377-389, 2001.

[18] 여정모, "마크흐름선도의 확장", 부산대학교 대학원 석사학위논문, 1982.

[19] 여정모, 황창선, "확장된 마크흐름선도와 시퀀셜제어시스템에의 응용", 부산대학교 공과대학 연구보고, Vol.25, pp.209-219, 1983.

[20] 여정모, "이산제어시스템 설계를 위한 확장된 마크흐름선도의 동작해석", 정보처리논문지, Vol.5, No.7, pp.1896-1907, 1998.

[21] 김희정, 허후숙, 정안나, 여정모, "접속 행렬을 이용한 EMFG의 수학적 해석", 한국멀티미디어학회 추계학술발표논문집, 제4권 제2호, pp.815-820, 2001.

[22] 백형구, 허후숙, 정명희, 여정모, "조건아크를 이용한 릴레이 회로의 EMFG 변환", 한국멀티미디어학회 추계학술발표논문집, 제4권 제2호, pp.821-826, 2001.

[23] 김희정, 여정모, 서경룡, "EMFG의 개선된 동작해석 알고리즘", 정보처리학회논문지A, 제9-A권 제3호, pp.371-378, 2002.

[24] 허후숙, 김남혁, 박희광, 여정모, "워크플로우의 EMFG 모델링", 한국정보처리학회 추계학술발표논문집, 제9권 제2호, pp. 2301-2304, 2002.

허 후 숙

e-mail : husuke@naver.com

2001년 부경대학교 전자계산학(이학사)

2003년 부경대학교 교육대학원 전산교육전공(교육학석사)

관심분야 : 전자상거래, 데이터베이스

여 정 모

e-mail : yeo@pknu.ac.kr

1980년 동아대학교 전자공학과(공학사)

1982년 부산대학교 대학원 전자공학과(공학석사)

1993년 울산대학교 대학원 전자및전산기공학과(공학박사)

1986년~현재 부경대학교 전자컴퓨터정보통신공학부 교수

관심분야 : 전자상거래, 데이터베이스