

제약식 기반의 능동적 워크플로우 컴포넌트 아키텍처 사례 연구

A Case Study of Active Workflow Component Architecture on Constraints Based

서 장 훈* 심 상 용* 이 건 혁* 박 명 규**

Seo Jang Hoon* Shim Sang Yong* Lee Kun Hyuk* Park Myeong Kyu**

Abstract

Many technical and nontechnical issues hinder enterprise wide workflow management. The most significant technical issue is the inability to deal with the heterogeneity among users, workflow types, and WFMSs. Not all users demand the same workflow functionality, so user interfaces of different levels of sophistication are required. Because workflow types cannot always be fully predefined, they often need to be adjusted or extended during execution. Unlike relational database management systems, however, each WFMS often has differing workflow metamodels. This leads to incompatibility between WFMSs, making integration into an environment comprising many heterogeneous WFMSs a troublesome and sometimes impossible task. Current Workflow system consists mainly of Database system. It contains some problems like that the integration relationship among system processes cant be expressed properly.

This research has been focused on two phases that should be considered in the Workflow system. First of all, the first phase is the analysis phase; one of its role is to figure out independent execution task unit (Workflow component). The second phase is design phase that provides with the framework to execute these task units actively. The Workflow component extraction method in the analysis phase uses a analysis method called C-C Net and, in the design phase, the architecture that makes the these Workflow component executed actively is provided.

Through this research, each process is divided into a task unit and more effective

* 썬더 부설 기술연구소

** 명지대학교 산업시스템공학부 교수

Workflow system could be formed by executing these units actively. Current system layer calls task units, on the other hand, the Workflow system this research implemented provides with the architecture that places a layer between them that controls task units actively.

Key Words : Workflow Management system, Workflow component, C-C Net

1. 서론

최근 워크플로우 시스템의 연구 동향은 시스템의 규모가 커짐에 따라 분산 워크플로우 시스템을 지양하고 있다. 워크플로우 시스템들은 공통 구성 요소는 물론 유사한 구조와 운용 형태를 가지고 있으나 모든 이용자에 대해서 똑같은 상황으로 전개되지 않으며, 상호 바람직한 방향으로 조정된 상황에서 이질적인 시스템을 유지하고 있는 실정이다. 이에 대하여, 워크플로우의 적용 분야의 확장에 따른 워크플로우의 표준화가 대두되게 되었는데, C-C Net이라는 분석법을 이용한 워크플로우 컴포넌트 확장 방법을 연구하고자 한다. 결과적으로, 본 논문에서는 워크플로우의 구조적인 관점에서 사례를 구현하고 그 효율성과 실용성을 제시하고자 한다.

2.1 문제 제기 및 연구목적

현재의 워크플로우 시스템은 대부분이 DBMS를 중심으로 이루어져 있다. 이것은 정보시스템 프로세스들의 연동 관계를 제대로 표현하지 못하는 등의 몇 가지 문제점을 포함하고 있다. 그러한 문제점을 지적하면 다음과 같이 세 가지로 요약할 수 있다. 첫째, DBMS 중심의 관리 즉, 데이터중심의 접근법은 여러 어플리케이션에 적용하여 사용하기에는 너무 고정적(strict)이기 때문에 유연성이 부족하다. 따라서 구현한 어플리케이션은 DBMS에 종속이 됨으로써, 기업의 상황에 따라 변화하는 요구사항을 어플리케이션에 반영하기 위해서는 DBMS에서 제공되는 환경 하에서만 가능하게 되었다. 둘째, DBMS를 중심으로 관리함으로써, 어플리케이션과 데이터베이스 등의 여러 자원들 사이에 발생 할 수 있는 상호작용(Interaction)과 협동(Collaboration)작업을 제대로 반영 할 수 없다. 셋째, DB 스키마 생성이 주된 관심사항이 됨으로써, 이는 실제시스템을 정보시스템으로 구현하고자 할 때, 가장 중요한 비즈니스 프로세스를 적절히 표현하지 못하고 있다. 또, DB 스키마의 변경은 곧 전체 정보시스템의 변경을 초래하게 되어 시스템 전반에 걸친 데이터 일관성에도 문제점이 발생할 우려가 있다.

오늘날의 비즈니스 환경은 역동성과 불확실성 그리고 오류가 자주 일어나는 환경으로 특징지을 수 있다. 요즘 기업에서 많이 행해지고 있는 비즈니스 프로세스 리엔지니어링(Business Process Reengineering : BPR)도 바로 프로세스를 중심으로 기업의 정보시스템을 구축하려는 움직임의 일환이다. 이러한 환경에 대응하기 위해 조직들은 비

즈니스 프로세스 재설계, 조직 재구조화, 권한부여(empowerment)와 같은 기법들을 통하여 끊임없이 변화하려는 자세를 취하게 되고 이의 결과로 조직의 비즈니스 프로세스는 계속적으로 변화하게 된다. 따라서, 변화하는 비즈니스 프로세스를 자동화하기 위한 워크플로우 시스템은 보다 유연한 모형이 필수적이다.

위에서 언급하는 바와 같이, 급변하는 정보시스템의 변화에 대한 인식과 DBMS의 문제점에 대한 문제 해결을 위해 최근 몇 년 전부터 기업의 정보시스템을 DBMS 중심의 관리에서 워크플로우관리시스템(Workflow Management System : WFMS)으로 변화하고 있다. 이는 프로세스 중심의 접근법으로서 기업의 정보시스템을 구현한 어플리케이션이 비즈니스 프로세스중심으로 모델링되고, 실행(Executing)되고, 모니터링(Monitoring)되는 기법이다.

기존의 DB 스키마를 중심으로 정보시스템을 구축하는 DBMS보다 프로세스 중심의 WFMS를 통해 얻을 수 있는 이점은 기업이 요구하는 비즈니스 프로세스의 변화와 서로 다른 환경에서 오는 정보 시스템의 변화에 대하여 효율적으로 대처를 할수 있게 되었다. 하지만, 기업의 요구사항이 복잡해지고, 개발 환경도 다변화가 되면서 WFMS는 그 실제 시스템의 업무를 다루는데 있어 한계성을 드러내고[4], 기업의 요구사항을 모두 수렴하지는 못하게 되었다. 특히 서로 다른 회사들에 의해 개발된 상용화된 WFMS는 수행도(performance), 신뢰도(reliability), 재사용성(reusability)등에서 문제점이 발생하였다.

특히, 현대 기업의 특성상 기업 내부의 요구사항은 수시로 변화를 요구하고, 이에 따라 정보시스템도 적절히 대처를 해야 한다는 측면에서 워크플로우의 재사용성 문제가 가장 큰 이슈로 떠오르게 되었다. 워크플로우의 재사용은 워크플로우 전체 시스템을 재사용의 단위로 설정 하는 것이 아니라, 워크플로우에서 가장 중요한 개념인 프로세스를 이루고 있는 각각의 테스크들에 대하여 재사용을 하는 것이다. 워크플로우의 재사용을 통해 기업은 급변하는 환경에 능동적으로 대처를 할수 있으며, 경제적/시간적으로도 많은 효과를 볼 수 가 있게 되는 것이다.³⁾

본 연구 관점에서 이를 실현하기 위해 우선적으로 해결하여야 하는 워크플로우 문제점을 정리를 하면 다음과 같다.

첫째, 전체 정보시스템에서 적절한 재사용의 단위를 분할하기 위해 필요한 프로세스를 어떻게 기술 하는나 하는 문제와 프로세스를 이루고 있는 테스크 단위의 설정을 어떻게 할 것인가에 대한 문제이다. 이는 워크플로우시스템에서 재사용 가능한 워크플로우를 도출해 내기 위한 프로세스 분석 단계의 문제이다.

둘째, 기업의 급변하는 요구사항에 대해 워크플로우도 적절한 조치가 취해져야 한다. 즉, 워크플로우의 실행시에 발생할 수 있는 데이터의 일관성(Consistency)문제를 어떻게 생산적이고 효율적으로 해결하는지에 대한 문제이다. 이것은 워크플로우의 설계 단계에서의 문제이다. 전자의 문제를 해결하기 위해 요즘 많이 사용하고 있는 시스템 모델링 방법론으로서 UML 기반의 객체지향 방법론(Object - Oriented

Methodology)을 통해 프로세스를 분석하는 방법을 사용했다.

UML(Unified Modeling Language)이란 소프트웨어 시스템에 대한 모델을 작성하기 위한 표준어로서, 모든 시스템에 대하여 가시화하고 명세하고 구축하고 문서화하는데 필요한 모든 관점과 과정에 대한 노테이션을 제공한다. UML에서 기업의 프로세스를 분석하기 위한 방법으로서 유스케이스(Usecase)를 사용한다. 하지만 유스케이스와 같은 프로시저(Procedure)한 방식으로 프로세스를 분석하는 것은 일관성 있는 산출물 도출이 힘들다. 또한 프로세스를 재사용 가능한 단위로서의 분할하는데 있어서도 적절한 방법을 제공하지 못했다. 이와 반해 재사용 가능하도록 기업의 프로세스를 좀더 정형화된 형식으로 분석하고, 도출하기 위한 방법을 제공하는 패러다임이 최근에 많은 연구가 이루어지고 있는 컴포넌트기반 방법론(Component-based Development : CBD)이다.

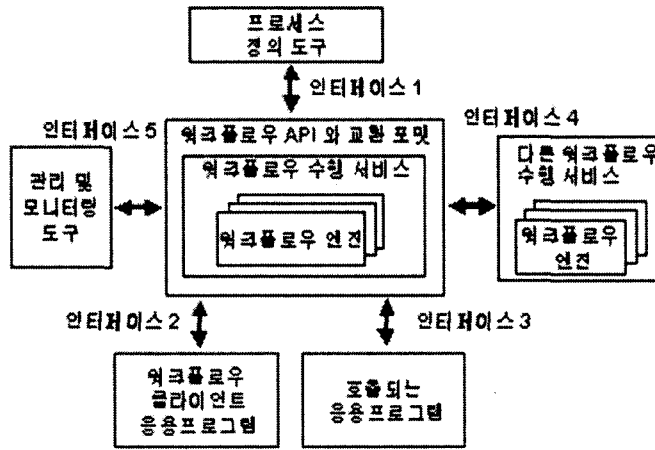
본 연구에서는 이러한 논의 과정에서 워크플로우 시스템에서 고려되어야 할 두 가지 단계-분석단계로서 독립실행 가능한 태스크 단위(워크플로우 컴포넌트)를 추출과 설계단계로서 이들 태스크 단위를 능동적으로 실행시키기 위한 프레임워크 제공-에 초점을 두고 있다. 분석단계에서 워크플로우 컴포넌트 추출방법은 C-C Net이라는 분석 방법을 사용하였고, 설계단계에는 이들 워크플로우 컴포넌트의 능동적인 실행을 위한 아키텍처를 제시하게 될 것이다.

2. 이론적 고찰

2.1 워크플로우 시스템 개념과 구성

워크플로우는 1980년대 초반 전산업계에 큰 영향력을 미치면서 파급된 비즈니스프로세스 리엔지니어링(Business Process Reengineering)에 힘입어 발전된 개념이다. 기업 구성의 핵심 3요소, 사람, 조직, 프로세스와 변화를 관리하고 지속적인 프로세스 개선을 실현하는 것이 확장된 개념의 워크플로우라고 할 수 있다.

특히, 워크플로우는 사무 자동화, DBMS, 전자우편, 문서 관리, 소프트웨어 프로세스 관리, 비즈니스 프로세스 모델 (BPM: Business ProcessModeling), 전사적 모델링과 구조(Enterprise Modeling and Architecture)와 같은 여러 가지 기술군들의 영향을 많이 받았다.



<그림 2.1> 워크플로우 구성요소

<그림 2.1>은 WfMC에서 제시한 참조 모델로서 워크플로우 관리 시스템의 일반적 구성요소는 크게 여섯 개로 나뉘어 있다. 가장 핵심적인 역할은 중앙의 워크플로우 수행 서비스로서 구성요소 전부를 하나의 시스템으로 묶어 주는 역할을 한다. 나머지 다섯 개의 구성요소들은 인터페이스를 통해 워크플로우 수행 서비스와 교류하게 되는데, WfMC에서는 각 인터페이스 명세를 표준 문서로 제공한다. 이러한 인터페이스를 표준으로 정한 이유는 워크플로우 관리 시스템의 여섯 개 구성 요소가 상이한 업체에 의해 개발될 수 있다는 것을 가정하였기 때문이다. 따라서, 각 구성요소들이 해당 인터페이스를 정확히 구현하였다면, 상이한 업체가 개발한 구성요소들과 함께 전체 시스템을 구성할 수 있다.⁴⁾

2.2 C-C Net의 특징과 구성요소

본 연구에서 워크플로우 컴포넌트 추출방법은 C-C Net이라는 분석 방법을 사용하였기 때문에 여기에 대한 특징과 구성요소에 대한 내용을 정리하면 아래와 같다.

(1) C-C Net의 특징

우선 기업의 비즈니스 프로세스를 분석하기 위한 방법으로 본 논문에서 참조를 하게 될 C-C Net에 대하여 우선 알아보면 다음과 같다.

C-C Net의 연구 배경은 비즈니스 프로세스 분석 모형으로서 기존의 절차적 프로세스 분석 방법의 한계인 객체 표현의 한계, 프로세스 단위의 불명확성, 사건의 연관성 문제, 및 협동 프로세스의 표현의 한계를 극복하고자 제시되었다. C-C Net은 다음과

같은 특징을 갖는다.

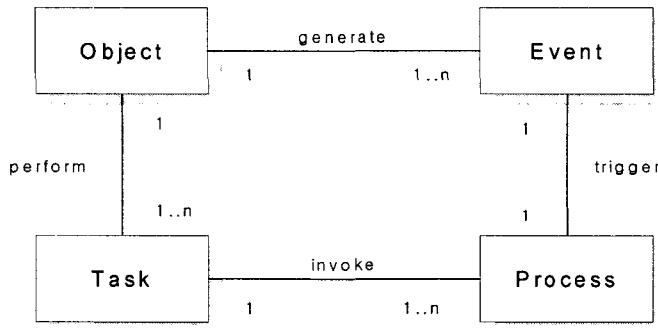
- 프로세스 수행 주체인 프로세스 분석 단계부터 명확히 표시하는 모형이다.
- 객체가 협동해야할 대상(객체)을 명확히 표현함으로써 프로세스의 기본 단위를 정의 할 수 있다.
- 제약식을 이용하여 사건에 의해서 서로 관련된 프로세스들을 하나의 선언적 메타 프로세스를 표현하는 모형이다. 선언적 방법에서는 객체의 상태 변수들 간에 유지되어야 할 일관성(Consistency)조건을 제약식으로 표현하고, 제약식이 파괴되는 사건이 발생 할 때 다시 제약식을 만족하는 상태로 복귀하기 위한 과정을 비즈니스 프로세스로 정의한다.
- 메타(mata) 프로세스로부터 사건-조건-활동 규칙(event - condition - action rule)을 이용하여 프로세스들을 추출할 수 있다.

(2) C-C Net 시스템 구성요소

정보시스템의 모델링의 핵심은 현실 비즈니스 시스템의 구성요소를 얼마만큼 자연스럽게 정보 모델에 완벽하고 추상적으로 반영하는냐에 달려있다. 비즈니스 프로세스를 분석하기 위하여 우선 기업의 비즈니스 시스템을 구성하고 요소들에 대하여 C-C Net에서는 다음과 같이 객체지향 관점에서 정의를 했다.

- 객체(Object) 및 객체형의 집합(Object type) - 객체는 시스템의 핵심 구성요소로서 업무 집행 역할 및 업무에 필요한 문서등 자료를 의미한다. 객체 = 상태변수 집합 + 메소드 집합. 객체는 객체형의 인스턴스이다.
- 사건(Event) - 사건은 객체에 의해서 발생한다. 시스템에서 발생하는 사건을 일반화(Generalization)하면 객체 추가, 삭제, 및 상태변수의 갱신으로 구분되어 질수 있다. 또한 객체들에 의해 발생하는 사건간에는 선행관계를 가지고 있다. 프로세스 진행 중에 수행 객체에 의해 발생하는 사건을 종속 사건이라고 한다. 이와 반대는 독립사건이라 한다.
- 업무(Task) - 객체들의 메소드에 해당한다. 활동(Activity)과 같은 의미이고, 프로세스를 이루고 있는 단위이다.
- 프로세스 - 기업의 입출력 데이터 관계에 의해서 미리 정해진 업무들의 실행순서로서 사건과 일대일 대응 관계에 있다. 각 프로세스는 수행 객체들의 메소드 호출순서에 의해서 결정된다.

이 같은 요소들의 상관관계는 다음의 <그림 2.2>와 같다.



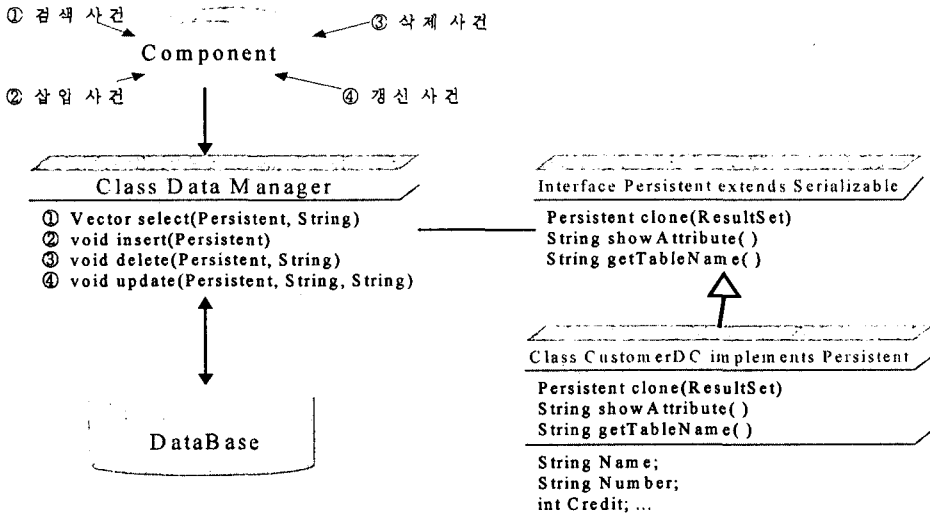
< 그림 2.2 > C-C Net에 의한 시스템 구성도

- ① 객체는 사건을 발생시킨다.
- ② 이 사건에 의해서 비즈니스 프로세스가 시작된다.
- ③ 수행 객체들의 메소드 호출 순서(즉 외부 엔티티에 의해 수행되는 업무들의 실행 순서)에 의해서 비즈니스 프로세스는 진행된다.
- ④ 만약 수행 객체들에서 종속 사건이 발생하면 새로운 비즈니스 프로세스가 생성된다.

C-C Net에서는 비즈니스 시스템에서 우선적으로 객체들을 분석하여 찾아 내야 한다. 찾아낸 객체들간에 계약을 하고 계약 사항을 완수하기 위해서 협동을 하는 과정을 비즈니스 프로세스로 정의한다.

C-C Net에서는 객체형의 종류는 도메인 객체형과 참조 객체형과 계약 객체형으로 나눌수 있다.

- 도메인 객체형(domain object type) - 스스로 사건을 유발시키는 능동 객체형으로서 자기 스스로 삽입, 삭제, 상태변수 갱신 사건을 생성하는 객체 종류를 도메인 객체라고 정의 한다.
- 참조 객체형(reference object type) - 스스로 사건을 생성해 낼 수 없는 수동 객체의 일종이다.
- 계약 객체형(contract object type) - 계약 객체형도 수동 객체의 일종이며, 도메인 객체간의 계약 함수로서 생성되는 객체이다.



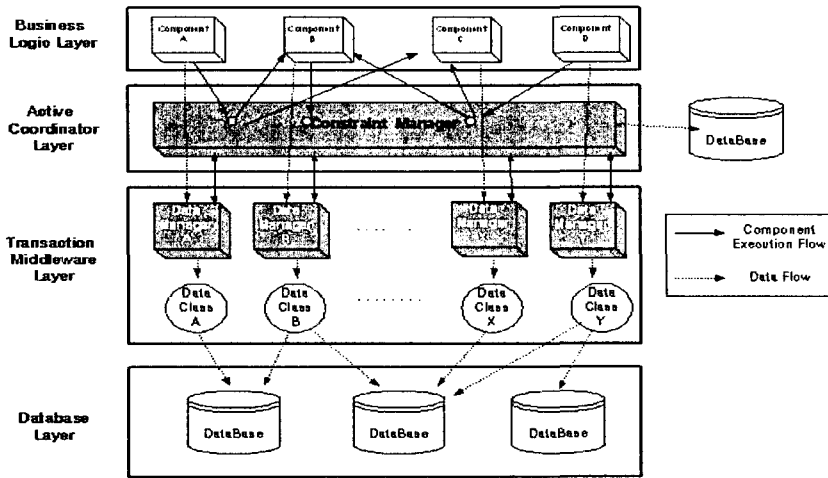
< 그림 2.3 > 기존의 워크플로우 구현 아키텍처

위에서 제시하는 < 그림 2.3 >는 세 개의 레이어에 대한 구현 기존의 워크플로우 구현 아키텍처이다. 워크플로우컴포넌트는 검색, 삽입, 삭제, 갱신 사건이 발생할 수 있다. 네 개의 사건은 각각 어플리케이션으로 구현될 수 있다. 이들 컴포넌트들이 다루는 데이터들의 데이터베이스에 영구히 저장을 해야만 한다. 저장을 하는데 있어 중요한 사항은 각각의 컴포넌트들에게 제공되어야 할 투명성(Transparency)문제이다. 투명성은 위치 투명성, 데이터 타입 투명성을 제공하여야 한다. 기존에는 데이터 관리자(Data Manager)와 같은 일반화되고 추상화된 형태의 미들웨어를 사용하므로써 컴포넌트들이 데이터베이스와 효율적인 접근이 가능했다.

단, 이 구조의 문제점은 데이터베이스와의 투명성 문제는 해결하였지만 컴포넌트들 간의 능동적인 실행순서는 해결하지 못하였다. 이와 같이 정적으로 실행순서가 정해진 아키텍처는 하나의 컴포넌트가 변경되는 사건에 대해 변경되는 컴포넌트와 계약을 맺고 있는 다른 컴포넌트들의 신속한 반응을 기대할 수 없다. 이는 곧 데이터 일관성 문제를 발생하게 된다. 워크플로우컴포넌트들의 효율적인 데이터 일관성 문제를 해결하기 위해서는 능동적인 아키텍처를 제공할 필요가 있다.

2.4 본 연구에서 제시하는 아키텍처

본 논문에서 제시하는 아키텍처는 워크플로우컴포넌트들의 실행순서가 정적으로 정해진 것이 아닌 능동적으로 호출을 하도록 하기 위한 아키텍처를 제시한다.



< 그림 2.4 > 제약식 기반 능동적 워크플로우 아키텍처

위 <그림 2.4>에서 제시하는 이 아키텍처는 워크플로우컴포넌트들의 실행순서를 능동적으로 제어하기 위하여 워크플로우컴포넌트들의 실행 순서를 총 관리하는 제약식 관리자(Constraint Manager)를 두었다. 제약식 관리자가 있으므로 해서 워크플로우컴포넌트들간의 정적으로 묶여진 레퍼런스를 더 이상 가지고 있지 않아도 되는 장점이 있다. 그리고, 기존의 아키텍처의 구성요소였다 워크플로우컴포넌트의 능동적인 실행을 위한 Active Coordinatot Layer가 추가되었다.

3. 기존연구 고찰

현재 워크플로우시스템에 대하여 제공되어지고, 연구되어지는 부문은 다음과 같은 두 가지로 그 특징을 찾을 수 있다.

첫째, 워크플로우시스템의 모델링(Modeling)부문이다. 이것은 기업의 활동을 이루고 있는 여러 요소들을 도메인에서 찾아내고, 또한 이들간의 관계 등을 분석하여 기업에서 요구하는 사항을 정확히 파악하여 워크플로우시스템에 적용시키기 위함이다. 비즈니스 프로세스의 모델링을 통해 찾아낼 수 있는 요소로는 우선 기업의 업무에 해당하는 프로세스(Process), 프로세스를 이루고 있는 단위인 활동(Activity - 또는 태스크(Task)), 활동의 실행을 시작하고 종료시키는 책임을 가지는 엔티티(Entity), 그리고 여러 활동들 사이에 주고 받는 데이터(Data)와 이들 데이터를 컨트롤하는 데이터 흐름(Data flow)등이 있다. 이 모든 요소는 비즈니스 프로세스를 이루고 있는 요소들로서

워크플로우시스템 모델링시 분석해야할 중요한 요소들이다.

둘째, 워크플로우시스템의 실행(Execution)부문이다. 이는 모델링 부문에서 정의를 내린 활동들을 실행시킬 때, 이들의 활동들에 의해 서로서로 발생하는 문제들을 해결하기 위한 스케줄링 문제이다.

워크플로우시스템에서의 모델링 하는데 있어서는 IDEF0과 같은 구조적 방법론은 비즈니스 시스템의 내부 사건을 반영할 수 없고, 1세대 객체지향 방법론은 현실 시스템에는 존재하지 않는 비즈니스 프로세스를 포함할 수 없다.[12] 또한 UML를 기반으로 하는 객체지향 방법론은 유스케이스를 통해 프로세스를 찾으려 고는 했지만, 각기 다른 도메인 지식을 가진 개발자마다 서로 다른 결과가 나타날 수 있는 문제점이 있다. 이것은 아직까지 모든 개발자들에게 있어 정형화된 방법을 제시 못하는데 있다.

이처럼 객체지향 방법론의 문제점을 지적하면서 컴포넌트 기반방법론에 대한 연구가 현재 활발히 연구되어지고 있다. 이에 한 예가 공리적 방법기반으로 한 정보시스템 모델링 방법[12]이 있다. 또, 워크플로우시스템에서의 문제점을 제시하고 이를 위한 해결 방법으로서 컴포넌트기반의 방법론을 사용한 연구도 있다. 선마이크로시스템즈에서 제공하는 엔터프라이즈 자바 빈즈(Enterprise Java Beans)도 바로 컴포넌트기반 방법론 안에 일어나는 모든 워크플로우 문제를 해결하기 위한 기본 스펙을 제공하고 있다.⁶⁾

현재 제공되는 동적인 워크플로우 시스템을 제공하기 위한 연구의 일환으로 요즘에는 ECA(Event - Constraint - Action)을 기반으로 하는 연구가 활발히 진행되고 있다.

ECA에 대한 연구는 WFMS 뿐만 아니라 DBMS에서도 ECA를 기반으로 하여 보다 동적인 데이터베이스를 위한 연구를 하고 있다. DBMS에서의 ECA를 적용한 능동적 데이터베이스(Active DataBase)는 테이블의 특정 필드가 수정될 경우 이에 대응되는 트리거가 발생하여 미리 저장된 프로시저 함수를 호출하게 된다. 이 함수는 특정 필드의 수정에 따른 데이터 일관성 문제를 트리거를 통하여 해결을 한다. 하지만 능동적 데이터베이스는 다음과 같은 문제점을 야기시킬 수 있다. 기업의 비즈니스 프로세스에 있어서 여러 태스크들을 능동적 데이터베이스를 통하여 데이터 일관성을 동적으로 해결함으로써 한 DBMS에 종속적인 환경이 된다. 요즘과 같은 이 기종의 환경에서는 어떤 한 플랫폼이나 개발 환경에 종속적 이어서는 안 된다. 이는 기업의 급변하는 환경에 보다 효율적으로 대처를 하는데 있어 많은 시간적/경제적으로 문제를 발생시킬 염려가 있다.⁸⁾

따라서, 본 논문에서는 아래 <표 3.1>에서 제시하는 바와 같이 능동적 데이터베이스를 통한 데이터일관성을 유지하는 것이 아니라, 활동들간의 발생하는 할 수 있는 모든 조건을 제약식으로 표현한 CC-Net[9]를 이용하고, 이 제약식을 파괴하는 조건을 체크하여 이와 관련된 조치를 취해주는 업무를 담당하는 제약식 관리자를 통하여 이 부문의 문제점을 해결하고자 한다.

< 표 3.1 > 워크플로우 연구의 분류

기존연구	본 논문
<p>UML(객체지향방법론)</p> <ul style="list-style-type: none"> • 프로세스 추출시(유스케이스 이용) 개발자의 도메인 지식에 따라다른 결과가 나타난다. 즉 보다 정형화된 방법을 제공하지 못함. • 워크플로우의 재사용성을 고려시 효율적인 방법제공하지 못함. 	<p>C-C Net(계약-협동 네트 모델)</p> <ul style="list-style-type: none"> • 기업의 요구사항에 적합한 프로세스 추출시의 정형화된 방법 제시 • 워크플로우 컴포넌트 모델을 이용하여 재사용을 향상시킬수 있음. • 컴포넌트 단위 추출
<p>능동 데이터베이스</p> <ul style="list-style-type: none"> • DBMS에 종속이 되어 이기종 환경에서의 기업의 요구사항의 변화를 효율적으로 대처하기 힘들. 	<p>본 논문 제시 아키텍처</p> <ul style="list-style-type: none"> • 한 시스템의 환경에 종속이지 않고, 컴포넌트들간의 제약식을 이용하여 데이터 일관성 문제를 해결함.

4. 결론 및 제언

워크플로우는 부분적 업무의 자동화와 전체 프로세스의 자동화라는 시각에서 달리 바라볼 수 있다. 즉, 단일 업무나 문서와 같이 단편화된 업무 프로세스의 자동화와 이를 위한 일종의 애플리케이션이라는 관점에서 볼 수도 있고 일상적으로 말하는 기업의 핵심 업무 프로세스 자동화 및 기업 전체의 프로세스 통합을 위한 도구라는 관점에서 볼 수 있다. 물론 모든 워크플로우가 후자의 기업 프로세스 전체의 통합 및 자동화라는 개념을 실현하는 것은 아니다. 극소수 공급사가 이러한 개념을 가지고 접근할 뿐 대다수의 워크플로우 제품은 부분자동화의 관점에서 개발돼 있는 실정이다.

본 논문에서 좀더 연구되어야 할 부분은 현재의 워크플로우 시스템은 문제해결을 통한 워크플로우 시스템의 시장규모 확대는 물론, 앞으로 도래할 자율분산 시스템 구현을 위한 중요한 대안인 이동에이전트 기술의 습득으로 자율분산 시스템 및 인터넷상의 응용시스템 발전방향 제시가 가능케 되었다는 점에서 현재의 제약식의 형태가 전체 시스템을 표현할 수 있는 평가체제 확립과 특정 도메인에 국한된 것이 아닌 모든 도메인에 적용시킬수 있는 제약식 형태를 유동적으로 만들어져야 할 것이다.

참 고 문 헌

- [1] 한승완, 임형석, “이동 에이전트 기반 전자상거래 시스템에서의 안전한 이동 에이전트 전송 프로토콜,” 한국정보처리학회 논문지, 제7 권 5S호, 5월, 2000.
- [2] 한승완, 임형석, “그룹 지정 암호화를 활용한 이동 에이전트 보호 메커니즘,” 한국정보과학회 봄 학술 발표논문집, 제28권 제1호, 4월, 2001.
- [3] 김창욱, 전진, 김성식 “객체지향 비즈니스프로세스 모델링 : 계약-협동 네트워크 모형”, 2000.
- [4] Jin, J. “공리적 방법 기반 재사용성 있는 소프트웨어를 위한 정보시스템 모델링 이론”, 고려대학교 대학원 박사학위 논문, 1999.
- [5] Alonso, G. Gunthor, R. Agrawal, D. El Abbadi, A., and Mohan, C. “Advanced Transaction Models in Models in Workflow Contexts,”. *International Conference on Data Engineering*, p125-128. 1996.
- [6] Amit P, Sheth., Wil van der Aalst., and Ismailcem B. Arpinar. “Processes Driving the Networked Economy” , *IEEE Concurrency* , Vol 3 , No 2 , pp. 143-159, 1999.
- [6] Georgakopoulos, D., Hornick, M., and Sheth, A. “An Overview of Workflow Management : From Process Modeling to Workflow Automation Infrastructure,” *Distributed and Parallel Databases* Vol 3, No 2 , pp. 119-152 , 1995.
- [7] Alonso, G. Agrawal, D, El Abbadi, A. and Mohan, C “Functionality and Limitations of Current Workflow Management Systems” , pp123-125, 1998.
- [8] M. Kamath and K. Ramamritham, “Failure Handling and Coordinated Execution of Concurrent Workflows,” Proc. Int’l Conf. on Data Engineering, pp. 334-341, 1998.
- [9] Work Group 1, “Workflow Management Coaliton - Interfacel:Process Definition Interchange Process Model,” WfMC TC-1016-P issue Nov. 12., 1998.