

CALS/EC 를 지원하는 차세대 워크플로우 시스템

최인준, 현석진, 이창우
포항공대 산업공학과

요약문

인터넷의 발달로 인해 조직 안의 마케팅, 광고, 조달, 판매, 서비스 제품 기획 등의 각 부분들간의 전자상거래와 조직 간의 전자상거래, 그리고 빠르게 변화하는 기회를 잡기 위해 모인 기업의 컨소시엄의 형태인 가상기업에 대한 관심이 고조되고 있다. 이러한 조직내/조직간의 전자상거래와 가상기업에서 수행되어야 하는 비즈니스 프로세스를 통합, 향상시키기 위한 활동이 워크플로우 시스템이다. 워크플로우 관리 시스템은 기업의 비즈니스 프로세스를 정의하고 수행하며, 이를 추적하고 리엔지니어링한다. 하지만, 비즈니스 프로세스는 오랜 수행 기간과 복잡한 비즈니스 규칙 등으로 인해 정확하고 신뢰적으로 수행하는 데에 많은 어려움을 가지며, 따라서 워크플로우 관리 시스템은 비즈니스 프로세스의 자동화 및 개선을 위해서 작업 수행의 정확성과 신뢰성을 제공할 수 있어야 한다. 본 논문은 Event-Condition-Action(ECA) 규칙을 사용하여 비즈니스 프로세스를 올바르게 신뢰도 있게 정의·수행·분석·개선할 수 있는 트랜잭션의 지원을 위한 통합적이고 체계적인 프레임워크를 제시하고자 한다. 또한 워크플로우 모델과 그에 수반되는 오류 모델을 함께 제시하고 Petri Net 으로 비즈니스 프로세스를 분석하는 방법론을 제시하고자 한다.

I. 소개

현대의 기업들은 급변하는 외부환경에 적절하게 대응하고, 그 외부환경에 대해 영향력을 행사하여 그것을 유리하게 변화 시켜야만 생존할 수 있다. 특히, 기업은 급변하는 외부환경에 맞춰 역동적으로 변화할 수 있는 효과적인 비즈니스 프로세스들을 가져야만 한다.

한편, 정보기술은 기업의 경쟁우위를 만드는 수단으로써 이용되어지고 있다. 이러한 정보기술은 기업의 전략을 보조하는 역할에서 출발하여 이제는 기업의 핵심 전략이 되고 있다. 하지만, 기업들은 최근까지 정보기술을 단위 수준에서 부분적으로 사용하여 왔으며, 이러한 부분적인 개

받은 급변하는 외부환경에 적합한 역동적이고 효과적인 비즈니스 프로세스의 관리를 저해하고 있다. 결국, 정보기술을 이용하여 비즈니스 프로세스를 개선시키고 자동화하려는 활동인 워크플로우가 필요하게 된 것이다.

워크플로우 관리 시스템은 기업의 비즈니스 프로세스를 정의하고 수행시킨다. 그리고, 워크플로우 관리 시스템은 비즈니스 프로세스를 추적하고 리엔지니어링할 수 있다[GHS95]. 이와 같은 워크플로우 시스템은 비즈니스 프로세스, 프로세스를 구성하는 업무, 업무간의 종속성, 업무를 수행하는 비즈니스 객체 등으로 구성되며, 특히 비즈니스 객체는 사람이나 애플리케이션 등이 될 수 있다.

비즈니스 프로세스에는 기업의 비전, 목표 등이 녹아있으며, 기업 전체의 자원들이 이용될 수 있다. 하지만, 비즈니스 프로세스는 오랜 수행 기간과 복잡한 비즈니스 규칙 등으로 인해 올바르게 신뢰적으로 수행하는 데에 많은 어려움을 가진다. 따라서, 워크플로우 관리 시스템은 이러한 비즈니스 프로세스의 자동화 및 개선을 위해서 작업 수행의 정확성과 신뢰성을 제공할 수 있어야 한다. 아울러, 워크플로우 관리 시스템은 비즈니스 프로세스를 분석하여 리엔지니어링에 사용될 수 있어야 한다.

본 논문은 Event-Condition-Action(ECA) 규칙을 사용하여 비즈니스 프로세스를 올바르게 신뢰도 있게 정의·수행·분석·개선할 수 있는 트랜잭션 기능을 지원하기 위한 통합적이고 체계적인 프레임워크를 제시하고자 한다. 세부 사항으로 워크플로우 모델, 오류 모델을 제시하고 Petri Net 으로 워크플로우를 분석한다.

II. 관련 연구

현재까지 워크플로우에 대한 연구는 데이터 중심의 향상된 트랜잭션 모델(Advanced Transaction Model)과 프로세스 중심의 트랜잭션 워크플로우(Transactional Workflow)가 있어왔다. 그리고, 이러한 두 가지 방향의 연구들이 서로의 장점을 수용한 통합된 접근이 연구되어지고 있다.

워크플로우 관리 시스템은 기존의 업무를 프로세스 관점에서 통합하려는 활동이다. 이러한 워크플로우를 전통적인 트랜잭션으로 해석해서 처리하려는 활동들이 있어 왔다. 하지만, 워크플로우는 다음과 같이 전통적인 트랜잭션과 차이가 있다[KR96].

- 워크플로우는 일반적으로 오랜 기간동안 수행된다.
- 자체적으로 개발된 독립적인 로컬 애플리케이션에 대해 완전한 제어를 가지지 못한다. 따라서, 로컬 애플리케이션의 몇 가지의 상태들만을 알 수 있는 블랙박스 접근법이 사용되어야 한다.

이외에도, 트랜잭션을 지원하지 않는 업무를 접근할 수도 있기 때문에 전통적인 트랜잭션을 적용하기에는 불가능하다.

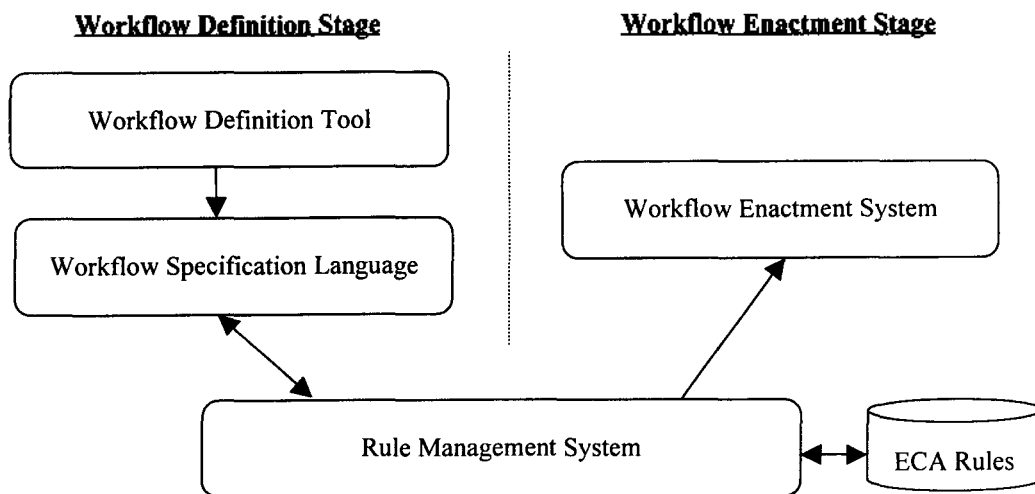
전통적인 트랜잭션 모델 이외에도 트랜잭션 요구와 애플리케이션 요구를 만족시키기 위해서 워크플로우 명세 자체에 오류를 명시하고 그것에 대한 처리 방법을 표현하는 트랜잭션 워크플로우를 동시에 적용함으로써 워크플로우 시스템이 가지는 요구 조건을 충족시킬 수 있다[BDS93]. 데이터 일관성을 위해서 멀티데이터베이스 트랜잭션 접근법을 이용하고 업무 일관성을 위해서는 트랜잭션 워크플로우를 이용하자는 것이다.

워크플로우에서는 오류가 발생했을 때, 전통적인 트랜잭션 모델에서 나타나는 Abort 를 하는 역방향 오류복구(Backward Recovery)를 할 수 없는 경우가 대부분이다. 따라서, 수행된 작업에 대해 의미적인 Abort 인 복구업무(Compensation)를 수행한다. 이것은 순방향 오류복구(Forward Recovery)에 해당한다. 이러한 복구업무는 수행된 작업을 무효화하는 효과를 낳는다. 하지만, 복구업무를 오류가 발생한 모든 워크플로우를 구성하는 업무에 대해 완전히 할 필요는 없다. 어떤 경우에는 복구업무를 할 필요가 없는 경우도 있고, 예전에 작업한 것을 처음부터 다시 하지 않고 수정하는 복구업무 즉 부분적 복구(Incremental Compensation)가 필요한 경우도 있다[KR98]. 또한, 성과를 고려하여 복구업무를 나중에 수행할 수도 있을 것이다. 복구 범위를 시스템이 자동적으로 찾아준다면 더욱 좋을 것이다[DDS97].

업무의 신뢰성과 Modularity 를 향상시키기 위해서, Task Execution Modes 와 Task Integration 이라는 개념을 적용하려는 시도가 있었다[PTV97]. 이외에도 Exotica Project[AKAA94, AKAA96], Meteor Project[KS95, MSKW96, WS97] 등이 있다.

III. 연구 프레임워크

워크플로우 관리 시스템은 초기에 워크플로우를 정의하는 부분과 정의된 워크플로우 명세를 실행하는 부분으로 나뉘어진다. 워크플로우 정의 부분과 실행 부분은 [그림 1]과 같이 ECA 규칙으로 연결될 수 있다.



[그림 1] 워크플로우 관리 프레임워크

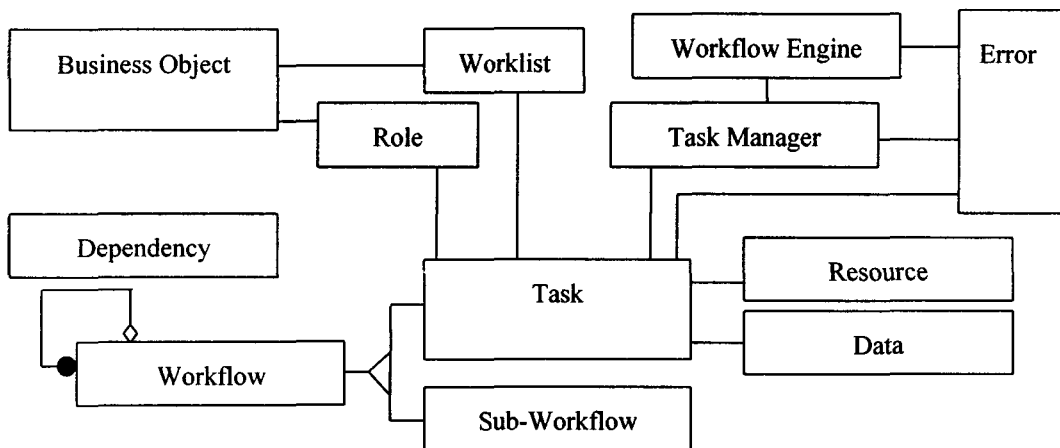
워크플로우 정의 부분에서는 그래픽 유저 인터페이스로 되어있는 워크플로우 정의 도구를 이용하여 비즈니스 프로세스의 정보들을 입력한다. 이렇게 정의된 워크플로우 프로세스 정보들은 워크플로우 관리 시스템이 이해할 수 있는 워크플로우 명세 언어로 바뀌게 된다. 워크플로우 명세 언어는 다시 규칙관리시스템에 의해 기존의 규칙과의 모순이나 루프가 없는 규칙으로 변환된다. 한편, 워크플로우 실행 부분에서는 규칙관리시스템의 도움을 받아 규칙을 실행한다.

워크플로우 관리 시스템에 ECA 규칙을 사용함으로써 역동적으로 변화하는 비즈니스 프로세스에 대해 적절한 조치가 용이하다. 기존의 워크플로우의 정의로부터 만들어진 워크플로우 인스턴스가 수행되고 있다면, 워크플로우의 명세의 변화는 수행되고 있는 워크플로우 인스턴스에 대해 영향을 미치기 어렵다. 하지만, 워크플로우 인스턴스의 수행이 ECA 규칙에 의한다면, 수행되고 있는 워크플로우 인스턴스는 변화된 ECA 규칙을 수행하기만 된다.

IV. 워크플로우 모델

4.1 워크플로우 시스템 모델링

워크플로우 시스템 모델은 워크플로우를 모델링하고 분석하는 데에 사용된다. [그림 2]는 워크플로우 시스템의 개괄적인 그림이다.



[표 2] 워크플로우 시스템 모델링

4.2 워크플로우 프로세스

워크플로우 프로세스는 다음과 같이 페트리 넷(Colored Petri net)로 표현될 수 있다.

- $WF = (\Sigma, P, T, A, N, C, E, I)$
 - ◇ Σ 는 비어있지 않는 타입들의 유한집합이다.
 - ◇ P는 Place의 유한집합이다. 워크플로우 프로세스에서의 상태에 해당한다.

- ◇ T는 Transition이다. 워크플로우 프로세스의 업무(Task)에 해당한다.
- ◇ A는 아크이다.
- ◇ N은 노드 함수($A \rightarrow P \times T \cup T \times P$)이다.
- ◇ C는 칼라 함수($P \rightarrow \Sigma$)이다.
- ◇ E는 아크 표현 함수($A \rightarrow \text{Expression}$)이다. 아크 전후의 토큰 변환의 역할을 한다. 또한, 워크플로우에서 Task의 선행조건으로 사용될 수 있다.
- ◇ I는 초기 상태이다. 초기에 각 Place에 존재하는 토큰의 개수를 나타낸다.

업무에는 정의된 일을 수행하는 실제업무와 업무를 수행하지 못하였을 경우 수행된 업무를 복구하는 복구업무(Compensation)로 나뉘어질 수 있다. 복구업무는 암시적으로 표현되어 있고 범위, 시기, 정도에 대한 정보가 포함되어 있으며, 이러한 업무는 비즈니스 객체의 오퍼레이션에 해당되게 된다.

업무는 업무를 수행하는 비즈니스 객체의 형태에 따라 분류된다. 비즈니스 객체의 상태는 워크플로우 관리 시스템의 구성 요소인 Task Manager를 통해서 알 수 있다. 비즈니스 객체의 분류에 의해 오류에 대한 복구 처리가 달라진다. 업무는 다음과 같이 분류될 수 있다.

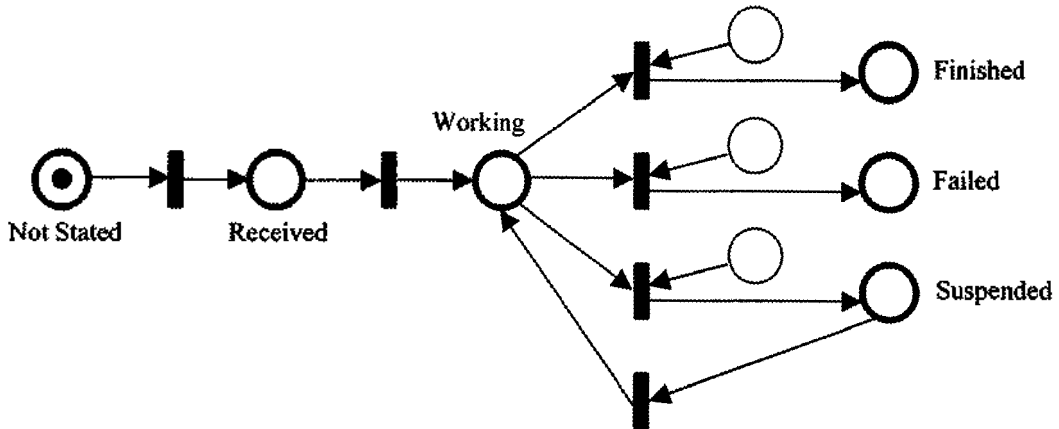
- 트랜잭션을 지원하는 업무(Transaction Task : TT) : Commit/Abort를 지원하는 업무
- 트랜잭션을 지원하지 않는 업무(Non-Transactional Task : NTT)
- 사람이 작업을 하는 업무(User Task : UT)

따라서, 업무 $T = TT \cup NTT \cup UT$ 이다.

각 업무는 블랙박스로 인식되고, 업무에 대해 완전한 컨트롤이 불가능하다고 가정한다. 하지만, 앞에서 밝힌 세 가지 형태의 업무에 대해서 서로 다른 몇 가지의 상태는 알 수 있다고 가정하는데, 다음은 세 가지 형태의 업무가 나타내는 상태이다.

- TT : Not Started, Received, Working, Committed, Aborted
- NTT : Not Started, Received, Working, Finished, Failed
- UT : Not Started, Received, Working, Suspended, Rejected, Finished, Failed

업무에 대한 상태들에 대해서 페트리 넷으로 표현이 가능하다. [그림 3]은 사람이 작업을 하는 업무의 경우이다. Failed의 경우에 복구 업무가 포함되면 Received의 상태로 변환된다. 그리고, 상태 정보는 워크플로우의 모니터링을 가능하게 한다. 결국, 워크플로우의 결함 포용 수준은 워크플로우를 구성하는 업무들의 형태 조합으로 알 수 있다.



[그림 3] 사람이 작업을 하는 업무에 대한 페트리 넷

4.3 워크플로우 시스템에서의 종속성

워크플로우 시스템 내에는 여러 가지의 종속성(Dependency)이 존재한다. 워크플로우의 종속성은 비즈니스 프로세스 자체의 요구와 워크플로우의 트랜잭션에 관련된 요구에 의해서 발생하며, 특히 워크플로우의 트랜잭션 요구는 워크플로우의 복잡한 비즈니스 규칙들을 표현하기에 알맞다. 다음은 애플리케이션 요구와 트랜잭션 요구를 세분화한 것이다.

■ 애플리케이션 요구(Application Requirements) [김성주 97]

- ◇ 무결성 제약 규칙(Integrity Rule) : 한 업무의 선행조건/후행조건간의 무결성 유지
- ◇ 행위 규칙(Behavior Rule) : 업무간의 선/후행 관계, 비즈니스 객체의 선택, 업무 선택 (Worklist Handling), 권한 부여(Authorization)
- ◇ 유도 규칙(Derivation Rule) : 업무 지연 보고 및 평가 등

■ 트랜잭션 요구(Transaction Requirements) [GHKM94]

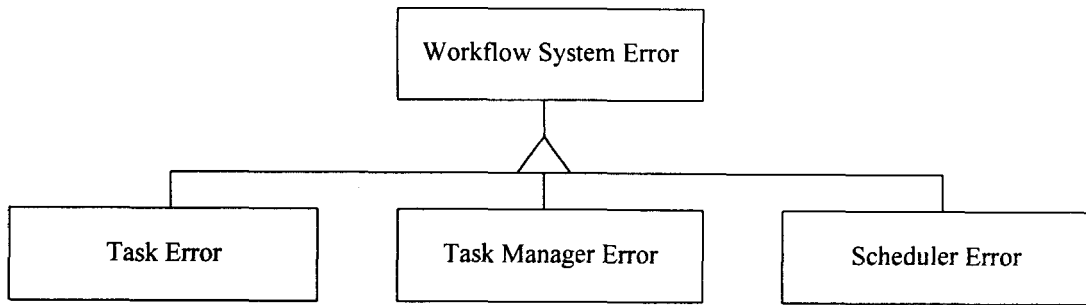
- ◇ 트랜잭션 상태 종속성(Transaction State Dependencies) : backward, forward, strong
- ◇ 옳음 종속성(Correctness Dependencies) : serialization, visibility, cooperation, temporal

종속성은 비즈니스 프로세스에 따라 요구하는 수준이 다를 수 있으며, 따라서, 종속성은 워크플로우를 정의할 때에 사전에 충분히 표현되어야만 한다. 이와 같은 종속성을 포함한 워크플로우 시스템을 페트리 넷으로 표현해서 6장에서 밝힌 분석을 한다.

V. 오류 모델

5.1 오류 클래스 구조

시스템에서 정의하는 오류 클래스 구조가 존재한다면, 오류를 정의하는 작업이 매우 편리할 것이다. [WSKM97]에서 위에서 아래로 오류 클래스를 정의한 것을 [그림 4]에 나타내었다.



[그림 4] 오류 클래스 구조

5.2 복구업무(Compensation)

실제업무의 오류에 대해서 복구를 하려고 할 때 다음의 문제들이 발생한다.

- 복구업무의 정의

복구업무 역시 실제업무에 해당한다. 따라서, 실제업무와 같은 방식으로 해야만 한다.

- 복구 범위 결정

복구의 전략에 따라 시스템이 자동적으로 처리할 수 있는 범위여야 한다.

- 복구 시기 결정

복구가 즉각적으로 되어야 하는 경우도 있지만, 시스템의 속도를 위해서 중요한 업무를 수행한 후에 처리할 수도 있다. 기업의 자원은 한정되어있기 때문에 복구 시기는 업무수행 속도를 고려하여 결정하여야만 한다.

- 복구 정도 결정

복구를 수행함에 있어, 항상 수행된 일을 완전히 무효화할 필요가 없는 경우가 많다. 따라서, 부분적인 복구 또한 가능해야 한다.

5.3 오류 복구 메커니즘

오류를 감지하는 데에는 규칙 관리 시스템의 도움이 절대적으로 필요하다. 업무의 오류 상태에 대해 오류 복구 구조에서 정의된 클래스의 인스턴스를 파라미터로 하는 이벤트를 발생하게 한다. 오류 이벤트를 규칙 관리 시스템이 워크플로우 관리 시스템에게 알리도록 하며, 트랜잭션을 지원하는 업무의 경우에는 Rollback을 수행하고, 트랜잭션을 지원하지 않는 업무인 경우에는 복구업무를 수행하게 한다. 특히 트랜잭션 사이에 종속성이 존재하는 경우에는 그 종속성이 반영되어야 할 것이다.

워크플로우 관리 시스템이 완전히 오류를 복구할 수 없는 경우에는 관리자에게 이를 알려서 오류

를 복구할 수 있는 알림(Notification) 기능 또한 제공되어야 한다.

VI. 워크플로우 분석

워크플로우 분석은 크게 타당성(Validation), 증명(Verification), 성능분석(Performance Analysis)로 구분될 수 있다[Aals98, Jens96]. 첫째, 타당성(Validation)은 워크플로우가 정의된 대로 작동하는가를 확인하는 것이다. 이것은 상호 시뮬레이션으로 알 수 있다.

둘째, 증명(Verification)은 워크플로우의 정확성 기준을 세우는 것이다. 기존의 데이터베이스 트랜잭션의 경우에는 Serializability 라는 정확성을 판단할 수 있는 기준이 있으나, 워크플로우의 경우에는 오랜 기간과 복잡한 비즈니스 규칙의 존재 등의 이유로 동일한 기준을 적용할 수가 없으며, 데이터베이스 트랜잭션과 같이 단일한 기준을 적용하는 것도 거의 불가능하다. 따라서, 정확성을 검증하기 위해서는 워크플로우를 정의한 다음에 정의된 워크플로우의 수행 가능한 모든 상태와 기존의 다른 워크플로우들과의 상호작용 등을 고려하여 시뮬레이션의 과정을 거쳐야 한다. 시뮬레이션과 병행해서 워크플로우 시스템에서 가장 기본적으로 만족해야 하는 워크플로우 프로세스, 업무, 종속성에 관련된 원칙(Axiom)을 깨뜨리는 경우에도 컴파일 에러를 나타낼 수 있어야 하며, 데드 락(Deadlock)도 찾아낼 수 있어야 한다.

셋째, 성능분석(Performance Analysis)는 정의된 워크플로우의 서비스 수준, 처리 시간, 자원 활용도 등을 분석하는 것이다.

VII. 결론 및 추후 연구 과제

CALS/EC 를 지원하는 워크플로우 시스템은 비즈니스 분야뿐만 아니라 제품 개발, 생산, 판매, 사후관리 등에 이르는 제품 생명주기(Life Cycle)에 적용 가능해야 한다. 또한, 복잡한 비즈니스 규칙 표현, 변화에 대한 유연성, 확장가능하고 안정된 시스템, 비즈니스 프로세서의 분석, 수행 객체들을 효율적인 지원 등의 능력을 가져야만 한다.

본 연구는 ECA 규칙을 바탕으로 트랜잭션 워크플로우 지원을 위한 통합적이고 체계적인 프레임워크를 제시하였다. 워크플로우 시스템에서 워크플로우 프로세스를 페트리 넷으로 모델링하고, 워크플로우 시스템에 존재하는 종속성을 정리하였다. 모든 소프트웨어는 오류를 포함한다. 워크플로우 시스템에서도 오류 복구 기능을 위해 오류 모델, 오류 복구 방법론을 제시하였다. 마지막으로, 정의된 워크플로우의 타당성, 정확성 증명, 성능분석 등의 워크플로우 분석에 대해 살펴보았다.

앞으로 본 연구의 워크플로우 모델, 워크플로우 분석 능력을 향상시켜야 할 것이다. 또한, 비즈니스 규칙을 정의하고 관리할 수 있는 분산된 워크플로우 관리 시스템 아키텍처에 대한 연구도 이루어져야 할 것이다.

참고 문헌

- [김성주 97] 김성주, “웍플로우 상에서의 규칙 관리”, 포항공대 산업공학과, 석사 학위 논문, 1996
- [AKAA94] G. Alonso, M. Kamath, D. Agrawal, A. El Abbadi, R. Gunthor and C. Mohan, “Failure Handling in Large Scale Workflow Management Systems”, Technical Report RJ9913, IBM Almaden Research Center, 1994
- [Aals98] W.M.P. van der Aalst, “The Application of Petri Nets to Workflow Management”, The Journal of Circuits, Systems and Computers, 1998
- [BDS93] Y. Breitbart, A. Deacon, H. J. Schek, A. Sheth, and G. Weikum, “Merging Application-centric and Data-centric Approaches to Support Transaction-oriented Multi-system Workflows”, ACM SIGMOD Record, 1993
- [DDS97] Weimin Du, Jim Davis and Ming-Chien Shan, “Flexible Specification of Workflow Compensation Scopes”, ACM SIGGROUP, 1997
- [GHKM94] D. Georgakopoulos, M. Hornick, P. Krychniak, and F. Manola, “Specification and Management of Extended Transactions in a Programmable Transaction Environment”, Data Engineering, 1994
- [GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth, “An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure”, Distributed and Parallel Databases, Vol. 3, No.2, April 1995
- [Jens96] Kurt Jensen, “Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use” Volume 1, Springer-Verlag, 1996
- [KR96] Mohan Kamath and Krithi Ramamritham, “Correctness Issues in Workflow Management”, Distributed System Engineering, Vol. 3, 1996
- [KR98] Mohan Kamath and Krithi Ramamritham, “Failure Handling and Coordinated Execution of Concurrent Workflows”, Data Engineering, 1998
- [KS95] N. Krishnakumar and A. Sheth, “Managing Heterogeneous Multi-system Tasks to Support Enterprise-wide Operations”, Distributed and Parallel Databases, Vol. 3, No. 2, April 1995
- [MSKW96] J.A. Miller, A.P. Sheth, K.J. Kochut and X. Wang, “CORBA-Based Run-Time Architectures for Workflow Management Systems”, Journal of Database Management, Special Issues on Multidatabases, Vol. 7, No. 1, 1996.
- [PTV97] J. Puustjarvi, H. Tirri, and J. Veijalainen, “Reusability and Modularity in Transactional Workflows”, Information Systems, Vol.22, No. 2/3, 1997
- [WS97] D. Worah and A. Sheth, “Transactions in Transactional Workflows”, In Advanced Transaction Models and Architectures. S. Jajodia and L. Kerschberg, Eds, Kluwer Academic Publishers, 1997
- [WSKM97] D. Worah, A. Sheth, K. Kochut, J. Miller, “An Error Handling Framework for ORBWork Workflow Enactment Service of METEOR”, Technical Report, LSDIS Lab. Dept. of Computer Science, Univ. of Georgia, July 1997