

실시간 워크플로우 모니터링 프레임워크 설계

정우진[○] 박민재 안형진 김광훈 백수기
경기대학교 전자계산학과 워크플로우 기술 연구실
{woojin[○], mipark, ctrl_workflow, kwang, skpaik}@kyonggi.ac.kr

Design of A Real Time Workflow Monitoring Framework

Woojin Jung[○] Meanjae Park Hyungjin Ahn Kwanghoon Kim Suki Paik
Dept. Computer Science, Kyonggi University

요 약

현재 기업 환경은 점차로 거대화되고, 기업들간의 상호운용을 통해 비즈니스 프로세스가 더욱 복잡해지고 있다. 또한 다양한 기업 환경 속에서 비즈니스 프로세스가 정상적으로 수행된다고 보장할 수 없다. 비효율적인 비즈니스 프로세스의 운용과 잘못 수행되는 프로세스는 조직 또는 기업의 관점에서 치명적인 문제가 될 수 있기 때문에 이것을 줄이기 위하여 비즈니스 관리자는 업무의 흐름과 상태를 모니터링할 수 있어야 한다. 워크플로우 시스템에서의 모니터링은 다양하게 변화하는 업무 상태를 추적하고 분석하여 효율적인 워크플로우 프로세스의 수행을 유도하는 것이다. 이에 가장 중요한 사항은 즉시로 업무의 상태가 분석되어서 관리자에게 가장 최신의 데이터를 전달하는 것이다. 본 논문에서는 워크플로우 관리 시스템에서 운용되는 비즈니스 프로세스를 실시간적으로 추적할 수 있는 모니터링 도구의 메커니즘을 제안한다.

1. 서 론

정보기술의 급속한 발전과 기업의 세계화는 기업환경에서의 비즈니스 프로세스에 대한 신속하고 정확한 처리를 요구하게 되었다. 프로세스란 개인, 부서, 관련 조직들 사이에 이루어지는 의사결정에 필요한 일련의 과정들을 의미하는 것으로 최종적인 의사결정에 이르기까지의 과정을 의미한다. 기업의 업무 활동은 수많은 프로세스로 이루어져 있고 기업들이 업무를 처리하는 모든 활동속에서 수많은 비즈니스 프로세스가 동시다발적으로 생성되고 처리된다. 전자업무 환경은 업무의 처리를 간편하게 만든 동시에 더욱 복합적인 업무 형태를 만들어내었고 이에 따라 비즈니스 프로세스의 효율적 관리가 더욱 절실하게 되었다. 워크플로우는 조직 내 또는 조직간의 비즈니스 프로세스와 사용자 그룹의 작업 및 활동이 어떻게 이루어지는가를 분석하고 업무의 신속하고 정확한 처리 및 효율적인 관리에 대한 요구를 충족시킬 수 있는 개념으로 떠오르고 있다.[4]

효과적인 워크플로우 시스템의 운용을 위해서는 전체 비즈니스 프로세스를 관리 감독하는 기능이 중요하다. 최근 워크플로우 관리 시스템이 처리해야 할 비즈니스 프로세스는 매우 복잡하고 긴 업무 처리 시간과 대규모화의 형태를 가지며, 관련 데이터의 증가와 고객의 요구 증가로 인해 단 순히 정의된 프로세스를 수행시키기가 쉽지 않게 되었다. 언제든지 수많은 예외상황이 발생할 수 있기 때문에 이것

을 처리할 수 있는 유연성을 증가시키고 대처방안을 수립하는것이 곧 기업과 조직의 경쟁력이 될 수 있다.

올바른 업무 상태의 파악과 예외상황에 대처할 수 있게 하기 위한 방안으로 비즈니스 프로세스에 대한 모니터링 기능이 필요하게 되었다. 본 논문에서는 워크플로우 관리 시스템에서의 모니터링의 기능을 살펴보고, 좀 더 효율적인 모니터링 기능을 수행하기 위한 실시간 모니터링 도구의 프레임워크를 기술한다.

2. 관련연구

2.1 일반 모니터링 서비스의 수행과 한계

일반적으로 워크플로우 시스템에서의 모니터링의 구조를 보면 어떠한 객체의 상태를 수집해 서비스를 하는 형태로 되어 있다. 모니터링 도구를 통해 기업이나 조직의 비즈니스 프로세스 관리자는 어떠한 정보가 필요할 때 다 혹은 주기적으로 프로세스나 액티비티 인스턴스의 상태 정보를 요청한다. 또는 워크플로우 관리 시스템이 통보해야 하는 이벤트를 만났다고 판단되어 질 때 호출되는 Call back 메커니즘을 통하여 특정 객체의 상태 정보를 돌려주기도 한다. 응답된 정보는 요청된 형태에 맞게 가공되어져 모니터링 도구를 통해 비즈니스 관리자 혹은 권한을 가진 사용자에게 보여지게 된다. 이러한 모니터링의 기능들을 통해 관리자는 업무 흐름의 상태와 업무

구조를 파악하여 비즈니스 처리의 편리성을 가질 수 있고 업무 프로세스의 재구성을 용이하게 할 수 있다.

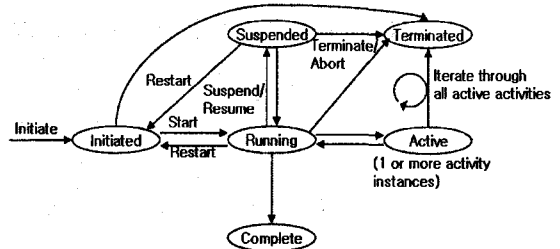
프로세스나 액티비티 인스턴스에 대한 모니터링을 요청하게 되면 엔진은 요청을 분석하여 그에 해당하는 데이터를 처리하여 응답을 보내게 된다. 하지만 모니터링 도구는 한 두가지의 프로세스 또는 액티비티 인스턴스에 대한 모니터링만 수행하는것이 아니다. 그물망같은 기업 환경에서는 수십 또는 수백가지의 프로세스가 존재할 수 있고 각 프로세스와 액티비티에 대한 인스턴스들의 개수는 수백만개가 될 수 있다. 이러한 수많은 인스턴스들에 대한 모니터링을 할 때 각 인스턴스에 대한 정보요청을 모두 엔진이 받는다면 워크플로우 엔진에는 엄청난 부하가 걸리게 되어서, 신속한 업무처리를 요하는 최근의 기업환경에서는 반드시 필요한 모니터링 서비스가 큰 부담으로 작용할 수 있다. 또한 기간, 수행자 또는 이벤트에 대한 여러 복잡한 조건들로 인스턴스 모니터링을 수행하고자 할 때는 여러 가지 작업을 처리해야 하는 워크플로우 엔진에 더 큰 무리를 줄 수 있다. 모니터링에 대한 요청이 들어오는 대로 엔진에서 각각의 요청에 대한 알맞은 데이터를 찾아 넘겨주는것도 엔진에 대한 부하를 증가시킬 수 있다. 엔진에 대한 부하는 요청을 처리하는 엔진의 작업 처리에 대한 지연 혹은 응답을 받게 되는 모니터링 도구들의 전달에 대한 지연을 발생시킬 수 있다. 이것은 비즈니스 흐름에 따른 인스턴스 변화에 대한 즉시적인 대처를 방해하며 발생하는 이벤트에 대한 적절한 처리를 어렵게 한다.

2.2 실시간 모니터링의 필요성

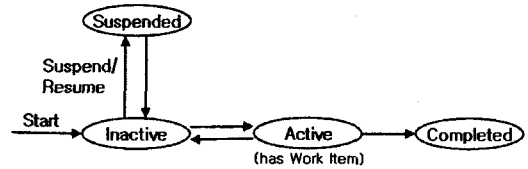
워크플로우 관리 시스템에서 처리되는 비즈니스 프로세스는 시스템 내,외부의 환경에 대해 많은 영향을 받을 수 있다. 비즈니스 프로세스 내의 단위 업무를 처리하는 동안의 수행 오류로 인한 문제가 발생할 수 있고 하드웨어적인 문제로 인한 업무 차질이 발생할 수도 있다. 또한 기업 간 협업 시 어느 한쪽의 급작스런 프로세스 변경이나 오류로 인한 문제도 야기될 수 있다. 비즈니스 조직이 점차로 거대화 되고 수많은 업무 부서들로 나뉘어짐에 따라 이러한 문제들이 발생할 가능성은 더욱 커지고 있다. 또한 전사적인 업무처리의 발달로 비즈니스 경계를 넘나드는 오늘날에는 더욱 복잡한 비즈니스 네트워크가 형성되어 비즈니스 프로세스 모니터링의 중요성은 더욱 커지고 있다.

앞서 언급한 대로 비즈니스 프로세스 모니터링은 관리자가 업무 상태를 파악할 수 있게 해준다. 하지만 만약 이러한 상태보고가 늦어지거나 과거 데이터에 기반한 상

대라면 심각한 문제가 발생할 수 있다. 워크플로우 시스템에서 비즈니스 프로세스는 특정 이벤트에 의해 정지(suspend 혹은 terminate)되지 않는다면 활성화되어 진행되는 구조이다. [3]



[그림 1] 프로세스 인스턴스의 트랜지션



[그림 2] 액티비티 인스턴스의 트랜지션

이것은 곧 관리자가 어떤 순간에 프로세스 인스턴스나 액티비티 인스턴스의 상태에 대한 요청을 했을때 그 순간이 지나면 프로세스가 진행되어 관리자에게 돌려진 데이터는 이미 예전의 데이터가 되어버릴 수도 있다는 것을 의미한다. 과거 데이터에 기반한 프로세스 분석은 복잡한 기업환경과 업무를 생각할 때 업무 진행에 오히려 역효과를 일으키며 기업 및 조직에 심각한 손실을 안겨 줄 수 있다. 따라서 모니터링 도구는 언제나 최신의 상태 데이터를 돌려줄 수 있어야 비즈니스 관리자가 올바른 판단과 분석으로 최적의 업무 운영을 이끌어낼 수 있다. 이런 이유로 현재 상태의 실시간 갱신 및 반영, 통계처리에 도움을 줄 수 있는 실시간 모니터링이 필요하다.

3. 실시간 모니터링 프레임워크

실시간 모니터링 프레임워크는 워크플로우 표준 기구인 WfMC에서 정의한 모니터링 서비스를 기본기능으로 가진다. WfMC에서 정의한 워크플로우 모니터링은 워크플로우 실행중에 발생한 이벤트를 추적하고 보고하는 능력을 말한다. 워크플로우 모니터링은 예를 들어 프로세스의 관리자가 실행중인 프로세스 인스턴스의 수행을 모니

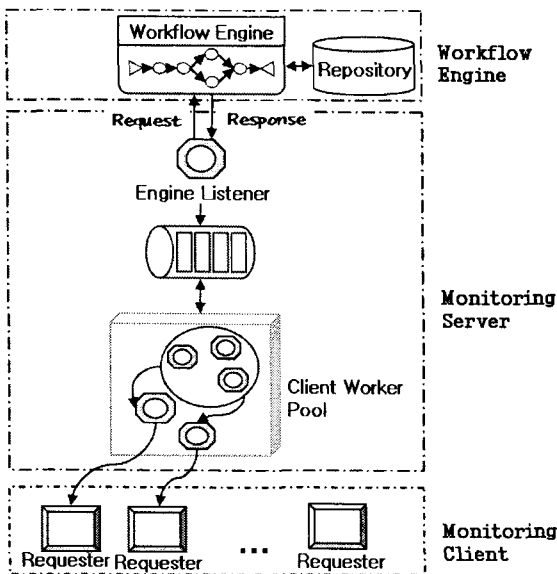
터하는데 사용된다.[2] 프로세스 수행 모니터링의 목적은 프로세스나 단위업무 인스턴스의 세부사항을 가져오고 예외 상황이 발생했을 때 관리자에게 적절한 행동을 수행할 수 있는 기능을 제공한다. 또한 특정 프로세스나 단위업무 인스턴스의 상태에 관한 정보를 기록하고 보여주며 워크플로우 관리 시스템의 결과에 대한 변화를 추적하기 위해, 개인이나 그룹 사용자에게 의해 처리된 일을 탐지하기 위해 분석된 상태 데이터를 축적한다. 모니터링의 결과로 축적된 로깅 데이터들은 워크플로우 관리 시스템이 제공하는 API를 이용하여 더 나은 비즈니스 프로세스 구성을 위한 기반이 된다.

3.1 실시간 모니터링 프레임워크 구조

실시간 모니터링 프레임워크는 그림 3과 같이 3-Tier의 서비스 처리로 구성된다.

워크플로우 엔진은 모델링된 비즈니스 프로세스를 수행한다. 이때 비즈니스 프로세스와 프로세스 내의 액티비티 인스턴스들은 수행되는 동안 각각의 전달되는 데이터와 이벤트 코드등 그 내용이 계속 갱신된다.

모니터링 도구는 서버와 클라이언트로 구성될 수 있다. 모니터링 서버측에서는 워크플로우 시스템이 진행중일 때의 정보를 지속적으로 수집하고 요청된 형식에 따라 가공하는 기능을 가진다. 모니터링 클라이언트 측에서는 모니터링 서버로부터 결과 데이터를 전달받아 원하는 형식으로 표현하는 기능을 담당한다.



[그림 3] 실시간 워크플로우 모니터링 프레임워크

3.1.1 실시간 모니터링 서버의 구조와 기능

모니터링 서버는 엔진 리스너, 순차적 데이터 저장소, 클라이언트 워커 풀로 구성된다. 모니터링 서버는 워크플로우 엔진에 등록되어 같이 구동될 수 있고 또는 워크플로우 엔진과 다른곳에 위치되어 독립적인 서비스를 제공할 수도 있다. 모니터링 서버는 클라이언트로부터 요청을 받아 조건에 맞는 인스턴스 데이터를 클라이언트에게 전달하는 기능을 한다.

▶ 엔진 리스너 : 모니터링 서버에서는 프로세스나 액티비티 인스턴스의 진행에 따른 변화를 감지하는 엔진 리스너를 둔다. 엔진 리스너는 지속적으로 워크플로우 엔진과 통신하며 엔진 내부에서 발생하는 모든 이벤트와 데이터 변경을 감지하며 진행중인 프로세스 또는 액티비티 인스턴스의 데이터를 모니터링 서버내의 데이터 저장소로 넘긴다.

▶ 순차적 데이터 저장소 : 엔진 리스너에서 받아들인 데이터는 모니터링 서버 내의 순차적 자료구조에 저장시킨다. 저장소에는 데이터가 들어오는 순서대로 저장되며 데이터를 처리할 때에는 먼저 들어와있던 데이터부터 처리되게 된다. 저장소의 크기는 워크플로우 서비스의 성능평가를 통해 적절한 크기로 지정한다.

▶ 클라이언트 워커 풀 : 서버 내에는 모니터링 클라이언트로부터 오는 요청을 받을 수 있는 클라이언트 워커를 포함시킨다. 클라이언트 워커는 여러 개를 생성시켜서 필요할 때 사용할 수 있는 풀(pool)의 형태로 구성한다. 클라이언트로부터 요청이 있을 때마다 워커를 생성시키는 방법은 대부분의 작업을 처리하기에 충분히 효율적이고 빠르지만, 클라이언트로부터의 연결과 종료가 매우 바쁘게 일어날 경우에는 계속해서 워커를 생성하고 종료시키는 비용을 무시할 수 없게 된다. 이런 이유로 워커 풀을 만들어두면 만약의 경우에 많은 수의 모니터링 요청이 들어오더라도 통신에 대한 딜레이를 줄이며 서비스가 가능하게 된다.

3.2 실시간 모니터링의 운용

워크플로우 시스템을 통해 비즈니스 프로세스를 수행하게 되면 모니터링 서버내의 엔진 리스너는 워크플로우 엔진과의 지속적인 통신을 통해 비즈니스 프로세스와 액티비티의 인스턴스 데이터를 서버 저장소에 저장시킨다. 엔진 리스너는 계속해서 데이터를 갱신해 받아오기 때

문에 저장소에는 항상 최신의 데이터가 쌓이게 된다. 클라이언트가 서버측으로 프로세스나 액티비티 인스턴스에 대한 모니터링 요청을 하게 되면 해당 요청에 대해 클라이언트 워커 풀에서 클라이언트에게 워커(리스너)를 할당한다. 모니터링 요청 하나에 워커 하나가 할당되며 워커는 저장소에 쌓인 데이터를 계속 소비하면서 요청에 맞는 데이터를 받아온다. 저장소에는 지속적으로 최신의 데이터가 채워지기 때문에 그것들을 사용하게 되는 클라이언트 워커 입장에서는 요청에 대한 트랜잭션이 유효한 동안 가장 최근의 데이터를 클라이언트에게 계속 돌려줄 수 있게 된다. 엔진 리스너는 항상 엔진과 통신을 하고 클라이언트 워커 또한 서버내의 저장소에 지속적인 접근을 하기 때문에, 워크플로우 엔진에서 처리된 데이터와 이벤트의 변화는 거의 동시에 클라이언트에게 적용될 수 있다.

4. 결론 및 향후 연구 과제

복잡하고 다양한 기업 환경에서 업무의 효율적 관리 감독은 곧 그 기업의 경쟁력으로 연결될 수 있다. 기업이나 조직 내에서 수행되는 업무 즉 비즈니스 프로세스가 얼마만큼의 안정성을 갖고 여러 가지 상황에 효율적으로 대처할 수 있는가 하는 점은 시장에서 그 기업의 신뢰도와 수익에 직접적인 영향을 끼친다. 비즈니스 프로세스가 수행되는 즉시로 관리자가 그 상태를 파악할 수 있다는 것은 기업이나 조직의 입장에서 볼 때 여러 상황에 대처할 수 있고 자원을 절약할 수 있는 큰 장점으로 작용한다. 이러한 개념을 바탕으로 본 논문에서는 워크플로우 관리 시스템에서 수행되는 비즈니스 프로세스를 실시간으로 관리할 수 있는 모니터링 툴의 메커니즘을 제시하였다. 실시간 모니터링 도구는 워크플로우 엔진에서 처리되는 비즈니스 프로세스의 데이터 흐름과 다양한 이벤트의 변화를 곧바로 감지하여 반영될 수 있도록 설계되었다. 실시간 모니터링 도구는 관리자가 프로세스 그룹을 실시간으로 파악하는 것뿐만 아니라 일반 업무 수행자에게도 적용되어 현재 자신이 속한 그룹의 업무현황과 자신의 업무 파악에도 도움을 줄 수 있다. 실시간 모니터링 도구와 함께, 진행중인 프로세스에 대한 여러 가지 제어가 가능한 관리자도구가 제공된다면 비즈니스 환경에서의 기업 운영은 더욱 유연하고 효율적이 될 것이다.

참고문헌

- [1] Frank Leymann, Dieter Roller "Production Workflow concepts and techniques", Prentice Hall
- [2] Workflow Management Coalition Specification Document, Workflow Management Coalition Terminology & Glossary, Document Number WFMC-TC-1011, Feb-1999, 3.0
- [3] Workflow Management Coalition Specification Document, Workflow Management Coalition The Workflow Reference Model, Document Number TC00-1003, Document Status - Issue 1.1, 19-Jan-95
- [4] 김광훈, "워크플로우 기술 I", TTA 저널 제 85호, 표준기술 동향, p107