



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년08월13일
 (11) 등록번호 10-0852141
 (24) 등록일자 2008년08월07일

(51) Int. Cl.

G06F 17/00 (2006.01)

(21) 출원번호 10-2006-0105462
 (22) 출원일자 2006년10월30일
 심사청구일자 2006년10월30일
 (65) 공개번호 10-2008-0038505
 (43) 공개일자 2008년05월07일
 (56) 선행기술조사문헌

WO9963469 A2
 US5890133 B1

전체 청구항 수 : 총 5 항

(73) 특허권자

경기대학교 산학협력단

경기도 수원시 영통 이의동 산 94-6 경기대학교

(72) 발명자

김광훈

경기 수원시 영통구 이의동 산 94-6 경기대학교
 8강의동 2층8210호

(74) 대리인

정연용

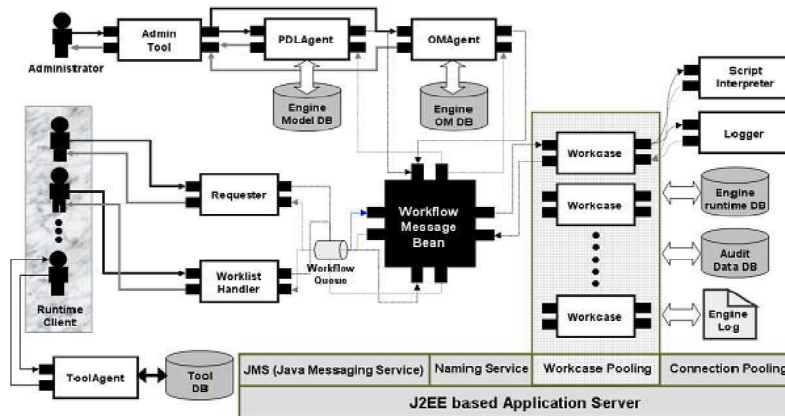
심사관 : 손영태

(54) 워크케이스 기반의 워크플로우 엔진

(57) 요약

오늘날 기업 내외에서 이루어지는 다수의 요청되는 비즈니스 업무들을 처리하기 위해서는 일반적인 워크플로우 엔진보다 처리량(throughput)이 높은 워크플로우 엔진이 요구된다. 하드웨어 및 인프라의 지속적인 발전에도 불구하고 비즈니스 관련 업무의 발생량을 감당하는데에는 분명한 한계점이 존재한다. 이러한 문제점에 대해서 기업이 보유하는 한정된 시스템 리소스들을 이용하여 초대형의 작업들을 처리해내기 위한 극복 방안이 마련되어야 한다. 따라서, 본 발명은 다수의 업무 요청에 의해 발생하는 대량의 워크플로우 인스턴스인 워크케이스들을 효율적으로 처리해낼 수 있는 메커니즘인 프로세스 인스턴스를 나타내는 '워크케이스 기반의 초대형 워크플로우 아키텍처'를 제시하고, 해당 아키텍처를 바탕으로 초대형 워크플로우 엔진을 제시하고자 한다.

대표도 - 도2



특허청구의 범위

청구항 1

OMG(Object Management Group)에서 표준 아키텍처로서 제시하고 있는 액티비티 기반 워크플로우 구조의 개선을 통하여 워크플로우 엔진의 능력을 높여 대량의 작업을 수행하는 워크플로우 엔진에 있어서,

클라이언트들과 상호 작용을 하는 객체로서 상기 클라이언트로부터 작업 요청을 받아들여 작업을 실행시키고 그에 대한 모니터링 정보를 제공하는 리퀘스터(Requester);

상기 리퀘스터(Requester)의 클라이언트들과 상호 작용을 하며 비즈니스 업무를 수행할 사용자에게 할당되는 단위 업무인 워크아이템과 관련된 서비스를 처리해 주는 워크리스트 핸들러(Worklist Handler);

상기 리퀘스터(Requester)와 워크리스트 핸들러(Worklist Handler)와 연결되어 상기 클라이언트들로부터 요청되는 워크플로우 서비스 오퍼레이션 또는 워크플로우 엔진 내부 컴포넌트들의 유기적 처리 중 발생하는 내부 서비스들에 대한 오퍼레이션들을 메시지로 구성화되어 적재하는 워크플로우 큐(Workflow Queue);

상기 워크플로우 큐(Workflow Queue)에서 메시지가 적재되면 서버에 의해 발생되어 감지된 이벤트에 의거하여 큐에 적재된 메시지를 획득하고, 상기 가져온 메시지의 내용들을 추출하여 그에 맵핑되는 워크플로우 관련 서비스와 연동하여 주는 워크플로우 메시지-드라이븐 빈(Workflow Message-Driven Bean);

상기 워크플로우 메시지-드라이븐 빈(Workflow Message-Driven Bean)에서 적재 이벤트를 감지하면 워크플로우 프로세스 정의 표준에 근거하여 프로세스 모델링 도구에 의해 작성된 XPD(XML Process Definition Language) 모델 데이터와 워크플로우 관련 데이터를 이용하여 작업 처리를 하는 워크케이스(Workcase);

의 J2EE(Java 2 Enterprise Edition)의 EJB(Enterprise Java Bean) 컴포넌트 프레임워크를 기반으로 구성되어 있는 것을 특징으로 하는 워크케이스 기반의 워크플로우 엔진.

청구항 2

제 1항에 있어서,

상기 워크리스트 핸들러는 작업을 이루는 액티비티 중에서 작업 수행자가 필요한 액티비티에서 발생하는 워크아이템 작업을 수행하는 주체인 클라이언트와 연결하며, 작업 수행자 또는 클라이언트를 관리하는 것을 특징으로 하는 워크케이스 기반의 워크플로우 엔진.

청구항 3

제 1항에 있어서,

상기 워크리스트 핸들러는 작업을 이루는 액티비티 중에서 작업 수행자가 필요한 액티비티에서 발생하는 워크아이템을 작업을 수행하는 주체인 클라이언트와 연결하는 것을 특징으로 하는 워크케이스 기반의 워크플로우 엔진.

청구항 4

제 1항에 있어서,

상기 워크케이스는 프로세스 인스턴스인 상기 워크케이스가 주체가 되어 모든 작업에 대해서 처리를 하게 되며, 요청한 작업의 수와 동일한 수의 워크케이스가 증가하는 것을 특징으로 하는 워크케이스 기반의 워크플로우 엔진.

청구항 5

제 4항에 있어서,

상기 워크케이스의 수가 요청한 작업의 수와 동일하게 존재하므로 처리 범위를 넘어서는 작업 요청에 대해서 상기 작업 요청에 대한 병목 현상이 일어나는 시간까지의 작업도 처리하는 것을 특징으로 하는 워크케이스 기반의 워크플로우 엔진.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 워크케이스 기반의 워크플로우 엔진에 관한 것으로서, 보다 상세하게는 초대형 워크플로우 엔진은 워크플로우를 사용하는 조직의 거대화로 인하여 나타난 워크플로우 동향인 초대형 워크플로우 환경을 처리하기 위한 워크플로우 엔진에 관한 것이다.

워크플로우(Workflow)란 한 조직체 내에서 발생하는 여러 단계의 다양한 비즈니스 업무 흐름을 정의하고, 이의 수행을 위한 효율적인 상호 작업 환경을 제공하는 자동화된 서비스를 의미한다고 할 수 있다. 따라서 워크플로우의 궁극적인 목적은 전체 비즈니스 업무의 목표를 효율적으로 달성하기 위함이라고 말할 수 있으며, 워크플로우는 비즈니스 프로세스를 나타내는 일련의 물리적 또는 논리적 단위인 액티비티(Activity)들과 이를 수행하기 위한 참여자(Participants), 그리고 액티비티들간에 전달되는 문서 또는 정보들로 구성된다.

상기 워크플로우 시스템을 설계하는 일은 데이터 구조와 알고리즘, 다양한 사용자에 대해서 고려해야 하며, 이를 위하여 크기와 유연성, 강건성, 속도, 사용성에 이르는 여러 가지 특성을 만족해야 한다.

도 1은 종래의 액티비티 기반 워크플로우 구조를 나타내는 구성도이다.

상기 도 1에서 보는 바와 같이 종래의 워크플로우 제품군의 워크플로우 엔진에서 사용되는 기반 아키텍처는 OMG(Object Management Group)에서 제안하는 조인트 플로우(Joint Flow) 아키텍처를 바탕으로 구현하고 있으며, 상기 조인트 플로우(Joint Flow)를 구성하는 컴포넌트인 wfRequester, wfProcessMgr, wfProcess, wfActivity는 각각 작업에 대한 요청, 워크플로우 프로시저, 워크케이스, 액티비티를 나타낸다.

그리고 작업 주체인 상기 액티비티에서 세부 작업이 이루어지며 실제적인 작업에 관련된 워크케이스나 액티비티는 감사 데이터를 남겨 마이닝(Mining)이나 시스템 복구 등의 여러 분야에 사용되어 진다.

상기와 같은 액티비티 기반의 워크플로우 시스템은 워크플로우 프로세스에 대한 인스턴스 뿐만 아니라 액티비티 인스턴스도 동적으로 생성되어 작업 증가에 따라 인스턴스들의 수가 급격히 증가하게 되어 워크플로우의 복잡도를 증가시키는 문제점이 있었다.

즉, 실제적인 인스턴스의 개수와 깊은 연관을 가지는 시스템의 복잡도가 급격히 증가하게 되며, 예를 들어, 만약 n개의 액티비티를 가진 워크플로우 프로시저 P1에 대하여 상기 P1의 작업 요청이 m번 들어왔다고 가정하면, 총 인스턴스의 개수는 $m+(m*n)$ 개가 되며, 예를 들어 6개의 액티비티를 가진 워크케이스를 1000개를 처리하기 위해서는 7000개의 인스턴스가 필요하게 되었다.

따라서 종래의 워크케이스 기반 구조에서는 워크플로우 프로세스들의 처리는 작업 요청 개수 m개 즉, 1000개의 인스턴스가 필요한 것에 비하여 너무 많은 인스턴스를 필요로 하게 되며, 상기의 이유로 인하여 요청 작업이 증가하게 됨에 따라 시스템에 과다한 처리 주체가 존재하여 처리를 위해 사용되는 리소스가 부족해져서 워크플로우 엔진의 병목 현상이 일어날 가능성이 높아지는 문제점이 있었다.

발명이 이루고자 하는 기술적 과제

상기와 같은 문제점을 해결하기 위하여 본 발명은 다수의 업무 요청에 의해 발생하는 대량의 워크플로우 인스턴스인 워크케이스를 효율적으로 처리해낼 수 있는 매커니즘인 프로세스 인스턴스를 나타내는 워크케이스 기반의 초대형 워크플로우 아키텍처를 제시하고 해당 아키텍처를 바탕으로 초대형 워크플로우 엔진을 제공하는데 그 목적이 있다.

발명의 구성 및 작용

상기와 같은 목적을 달성하기 위하여 본 발명은 OMG(Object Management Group)에서 표준 아키텍처로서 제시하고 있는 액티비티 기반 워크플로우 구조의 개선을 통하여 워크플로우 엔진의 능력을 높여 대량의 작업을 수행

하는 워크플로우 엔진에 있어서,

클라이언트들과 상호 작용을 하는 객체로서 상기 클라이언트로부터 작업 요청을 받아들여 작업을 실행시키고 그에 대한 모니터링 정보를 제공하는 리퀘스터(Requester)와;

상기 리퀘스터(Requester)의 클라이언트들과 상호 작용을 하며 비즈니스 업무를 수행할 사용자에게 할당되는 단위 업무인 워크아이템과 관련된 서비스를 처리해 주는 워크리스트 핸들러(Worklist Handle)와;

상기 리퀘스터(Requester)와 워크리스트 핸들러(Worklist Handler)와 연결되어 상기 클라이언트들로부터 요청되는 워크플로우 서비스 오퍼레이션 또는 워크플로우 엔진 내부 컴포넌트들의 유기적 처리 중 발생하는 내부 서비스들에 대한 오퍼레이션들을 메시지로 구성화되어 적재하는 워크플로우 큐(Workflow Queue)와;

상기 워크플로우 큐(Workflow Queue)에서 메시지가 적재되면 서버에 의해 발생되어 감지된 이벤트에 의거하여 큐에 적재된 메시지를 획득하고, 상기 가져온 메시지의 내용들을 추출하여 그에 맵핑되는 워크플로우 관련 서비스와 연동하여 주는 워크플로우 메시지-드라이븐 빈(Workflow Message-Driven Bean)과;

상기 워크플로우 메시지-드라이븐 빈(Workflow Message-Driven Bean)에서 적재 이벤트를 감지하면 워크플로우 프로세스 정의 표준에 근거하여 프로세스 모델링 도구에 의해 작성된 XPDL(XML Process Definition Language) 모델 데이터와 워크플로우 관련 데이터를 이용하여 작업 처리를 하는 워크케이스(Workcase);

의 J2EE(Java 2 Enterprise Edition)의 EJB(Enterprise Java Beans) 컴포넌트 프레임워크를 기반으로 구성되어 있는 것을 특징으로 하는 워크케이스 기반의 워크플로우 엔진이다.

상기 워크리스트 핸들러는 작업을 이루는 액티비티 중에서 작업 수행자가 필요한 액티비티에서 발생하는 워크아이템 작업을 수행하는 주체인 클라이언트와 연결하며, 작업 수행자 또는 클라이언트를 관리하는 것을 특징으로 한다.

상기 워크리스트 핸들러는 작업을 이루는 액티비티 중에서 작업 수행자가 필요한 액티비티에서 발생하는 워크아이템을 작업을 수행하는 주체인 클라이언트와 연결하는 것을 특징으로 한다.

상기 워크케이스는 프로세스 인스턴스인 상기 워크케이스가 주체가 되어 모든 작업에 대해서 처리를 하게 되며, 요청한 작업의 수와 동일한 수의 워크케이스가 증가하는 것을 특징으로 한다.

상기 워크케이스의 수가 요청한 작업의 수와 동일하게 존재하므로 처리 범위를 넘어서는 작업 요청에 대해서 상기 작업 요청에 대한 병목 현상이 일어나는 시간까지의 작업도 처리하는 것을 특징으로 한다.

이하, 본 발명을 구체적으로 설명하기 위하여 첨부된 도면을 참조하여 상세히 설명한다.

도 2는 본 발명에 따른 워크케이스 기반의 워크플로우 엔진의 구조도이며, 도 3은 본 발명에 따른 워크케이스 기반의 워크플로우의 구조도이고, 도 4는 본 발명과 종래의 워크플로우 구조 간 성능 비교 분석도이다.

초대형 워크플로우 관리 엔진은 워크플로우를 사용하는 조직의 거대화로 인하여 나타난 워크플로우의 동향인 초대형 워크플로우 환경을 처리하기 위한 워크플로우 엔진으로서, 초대형 워크플로우 관리 시스템의 핵심적인 목표는 백만 건 이상의 대량 작업의 처리가 가능한 대규모 처리 능력을 가지는 것이다.

이와 같은 초대형 워크플로우 관리 시스템의 엔진을 설계 및 개발하기 위하여 상기 도 2에서 보는 바와 같이 기존의 OMG(Object Management Group)에서 표준 아키텍처로서 제시하고 있는 액티비티 기반 워크플로우 구조의 개선을 통하여 워크플로우 엔진의 능력을 높여 대량의 작업을 사용하게 하는 것으로, 본 발명의 워크케이스 기반의 워크플로우 엔진은 J2EE(Java 2 Enterprise Edition)의 EJB(Enterprise Java Beans) 컴포넌트 프레임워크를 기반으로 설계된다.

상기 워크케이스 기반의 워크플로우 엔진에서의 주요 구성 요소는 시스템 안에 존재하는 리퀘스터(Requester), 워크리스트 핸들러(Worklist Handler), 워크케이스(Workcase)이며, 상기 세 개의 컴포넌트들 중 외부와 상호작용이 있는 리퀘스터와 워크리스트 핸들러는 클라이언트에게 인터페이스를 제공한다.

상기 리퀘스터(Requester)는 클라이언트들과 상호 작용을 하는 객체로서 상기 클라이언트로부터 작업 요청을 받아들여 워크케이스를 생성하거나 생성된 작업을 그에 대한 모니터링 정보를 제공하여 주며, 상기 워크리스트 핸들러(Worklist Handler)는 상기 리퀘스터(Requester)의 클라이언트들과 상호 작용을 하며 비즈니스 업무를 수행할 사용자에게 할당되는 단위 업무인 워크아이템과 관련된 서비스를 처리하여 준다.

그리고 워크플로우 큐(Workflow Queue)는 상기 리퀘스터(Requester)와 워크리스트 핸들러(Worklist Handler)와

연결되어 상기 클라이언트들로부터 요청되는 워크플로우 서비스 오퍼레이션 또는 워크플로우 엔진 내부 컴포넌트들의 유기적 처리 중 발생하는 내부 서비스들에 대한 오퍼레이션들을 메시지로 구성화되어 적재를 하는데, 상기 워크플로우 큐(Workflow Queue)에 적재된 메시지들은 워크플로우 관련 서비스들에 대한 오퍼레이션(=메소드) 및 오퍼레이션 처리를 위해 사용되는 관련 매개변수(=파라미터)들이 기록되어 있다.

또한 워크플로우 관련 서비스 오퍼레이션이 메시지화되어 상기 워크플로우 큐(Workflow Queue)에 적재되는 순간 큐를 관리하는 서버는 큐에 적재가 수행되었다는 이벤트를 발생하게 되는데 이러한 적재 이벤트를 감지하는 역할을 담당하는 컴포넌트가 워크플로우 메시지-드라이븐 빈(Workflow Message-Driven Bean)이며, 상기 워크플로우 메시지-드라이븐 빈(Workflow Message-Driven Bean)은 상기 워크플로우 큐(Workflow Queue)에서 메시지가 적재되면 서버에 의해 발생되어 감지된 이벤트에 의거하여 큐에 적재된 메시지를 획득하고, 상기 가져온 메시지의 내용들을 추출하여 그에 맵핑되는 워크플로우 관련 서비스와 연동하여 준다.

상기 워크케이스는(Workcase)는 상기 워크플로우 메시지-드라이븐 빈(Workflow Message-Driven Bean)에서 적재 이벤트를 감지하면 워크플로우 프로세스 정의 표준에 근거하여 프로세스 모델링 도구에 의해 작성된 XPDL(XML Process Definition Language) 모델 데이터와 워크플로우 관련 데이터를 이용하여 작업 처리를 하며 모델에서 정의된 워크플로우 프로시저에 따라 작업을 처리하며 작업의 상태 관리 등의 일을 한다.

여기서 액티비티 작업의 처리도 워크케이스 내부에서 일어나기 때문에 액티비티에서 수행자와 작업을 할 경우에 상기 워크케이스 핸들러(Worklist Handler)를 이용하여 작업을 처리하며, 작업을 이루는 액티비티 중에서 작업 수행자가 필요한 액티비티에서 발생하는 워크아이템을 작업을 수행하는 주체인 클라이언트와 연결하며, 작업 수행자 또는 클라이언트를 관리하는 역할도 상기 워크케이스 핸들러(Worklist Handler)에서 처리하게 되므로, 상기 워크케이스 기반의 워크플로우 엔진은 메시지 큐를 이용하여 비동기적으로 작업을 처리함으로써 주요 내부 컴포넌트들 간의 커플링을 최소화하고 엔진 시스템에 추가 컴포넌트가 생길 경우 메시지 핸들러에 연결하여 추가를 용이하게 한다.

그러나 상기 워크케이스 기반의 워크플로우 엔진은 핵심 구성 컴포넌트들의 내부적인 상호작용을 통해 외부로부터의 서비스를 처리하게 되지만, 워크플로우 파트너들 간에 단순 또는 복잡한 서비스들의 상호교환을 처리할 수 있기 위해서는 동기적인 서비스 뿐만 아니라, 비동기적인 서비스를 처리할 수 있는 환경이 되어야만 하므로, 상기 워크케이스 기반의 워크플로우 엔진의 외부 서비스 요청을 받아들이는 창구 역할을 하는 컴포넌트들인 리퀘스터(Requester)와 워크리스트 핸들러(Worklist Handler)는 클라이언트 측으로부터 전달된 서비스 요청을 수행하고 그에 대한 결과를 응답한 후 서버 측과 클라이언트 측의 연결된 세션을 종료하는 동기적인 형태의 메커니즘을 사용한다. 따라서 장시간 트랜잭션 처리가 빈번히 요구되는 워크플로우 상호운용 환경에서 대량의 서비스들에 대한 신뢰성이 있는 처리를 가능하게 하기 위해서는 비동기 메커니즘을 기반으로 서비스 창구를 구현하는 것이 바람직하다.

상기 도 3에서 보는 바와 같이 본 발명의 워크케이스가 작업의 처리 주체로서 존재하여 작업의 처리는 모두 워크케이스를 통하여 이루어지며, 워크케이스 기반의 워크플로우 시스템 구조의 특성은 요청한 작업의 수와 같은 수의 동적인 워크케이스 인스턴스가 증가하게 된다.

보다 상세하게는 모든 작업에 대해서 프로세스 인스턴스인 워크케이스가 주체로서 처리를 하게 되며, 요청한 작업의 수와 같은 수의 워크케이스가 증가하는 것이 특징이라 할 수 있으며, 상기 워크케이스는 액티비티를 워크케이스 내부적으로 포함하는 데이터로서 보고 있으며 작업을 처리하기 위한 인터페이스들을 외부로 제공한다.

그러므로 요청된 작업의 처리는 상기 워크케이스가 제공하는 인터페이스를 참조하여 프로세스 관련 데이터, 액티비티 데이터, 관련 데이터들을 가지고 이루지며, 본 발명의 워크케이스 기반 초대형 워크플로우 시스템에서는 작업 처리 주체가 되는 워크케이스들이 작업 요청 수와 같은 수로 시스템에 존재하게 된다.

따라서 상기 워크케이스 인스턴스는 워크플로우의 실행시간에 동적으로 증가되는 데 이런 방식은 워크플로우 시스템에서 작업을 처리하는 처리 주체의 부족으로 리소스의 여유에도 불구하고 병목 현상이 일어나는 것을 예방하며, 상기 워크케이스 기반의 초대형 워크플로우 시스템 구조에서 작업의 처리는 작업의 요청에 따라서 워크케이스가 생성되어 워크케이스가 워크케이스의 정보와 액티비티의 데이터를 통하여 작업을 처리하는 방식을 가진다. 이런 워크케이스 기반 방식은 작업의 증가에 따른 워크플로우 시스템의 구성 요소의 부하가 처리 주체의 동적인 증가로 정적인 구조에 비하여 상당량 감소하게 된다. 또한 상기 구조는 워크플로우의 복잡도가 실질적인 작업 객체의 증가로 나타나는 시스템의 복잡도와 비례하여 증가하게 된다.

그러므로 워크케이스 수에 따라서 시스템의 부하가 걸리는 것을 나타내는 상기 도 4에서 보는 바와 같이 일단 모든 워크플로우 아키텍처는 하드웨어 적으로나 소프트웨어 적으로 처리 범위를 넘어서 작업 요청을 받으면 병목 현상이 생기게 되므로, 초대형 워크플로우 아키텍처를 선택하는 것은 상기의 병목 현상이 늦게 발생하는 즉, 더 많은 수의 작업을 처리하는 구조이기 때문이다.

따라서 종래의 액티비티 기반의 워크플로우 구조는 처음에는 시스템 부하가 적지만 작업 요청이 많아지면서 작업에 따르는 액티비티 인스턴스를 생성시켜 많은 수의 작업을 처리하는데 적합하지 않은 반면에, 워크케이스 기반의 워크플로우 구조는 비록 상기 액티비티 기반의 워크플로우 구조보다는 부하가 더 걸리지만 병목 현상이 일어나는 시간까지 더 많은 작업을 처리할 수 있다.

즉, 본 발명에서 제시하는 워크케이스 기반의 워크플로우 엔진 구조가 종래의 액티비티 기반의 워크플로우 구조에 비해서 거대량의 작업을 처리하는 비즈니스 도메인에 더 적합한 아키텍처라 할 수 있다.

상술한 바와 같이 본 발명에 따른 바람직한 실시예를 설명하였지만, 본 발명은 상기한 실시예에 한정되지 않고, 이하의 특허청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 분야에서 통상의 지식을 가진 자라면 누구든 다양한 변경 실시가 가능한 범위까지 본 발명의 기술적 정신이 있다고 할 것이다.

발명의 효과

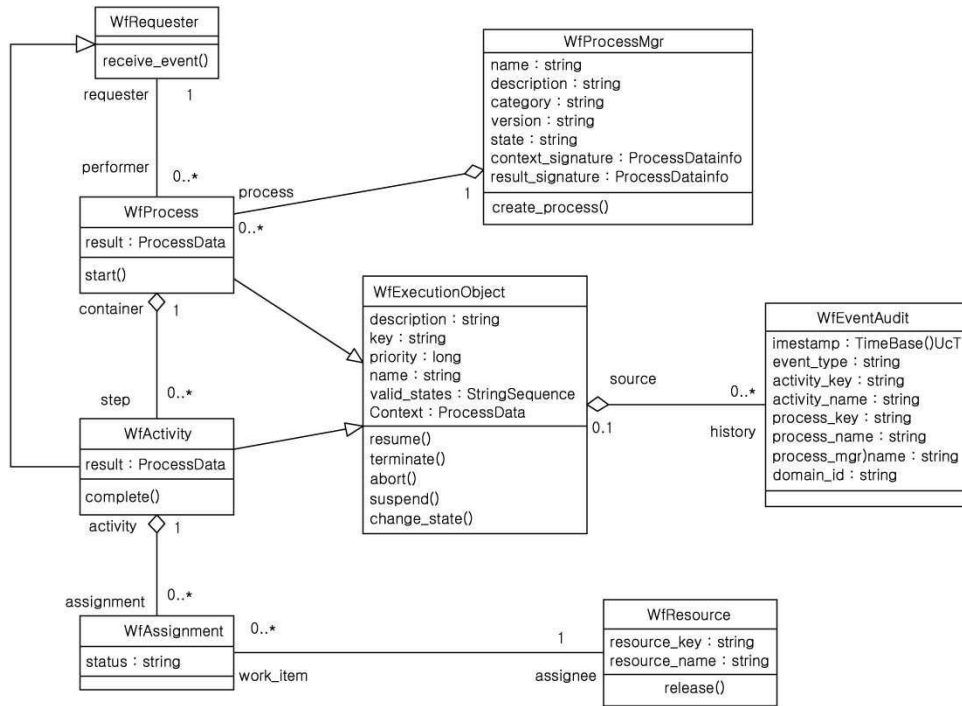
이상에서 상세히 살펴본 바와 같이 본 발명에 따른 워크케이스 기반의 워크플로우 엔진은 종래에 제시된 아키텍처 기반의 워크플로우 엔진에 비하여 광역화되고 거대화된 비즈니스 프로세스들의 대량의 업무들을 보다 효율적인 리소스 관리를 이용해 처리해낼 수 있는 매커니즘을 제공함으로써 복잡도가 높은 대규모의 기업 인프라 환경에서의 업무 처리에 적합한 시스템으로 활용 가능한 효과가 있다.

도면의 간단한 설명

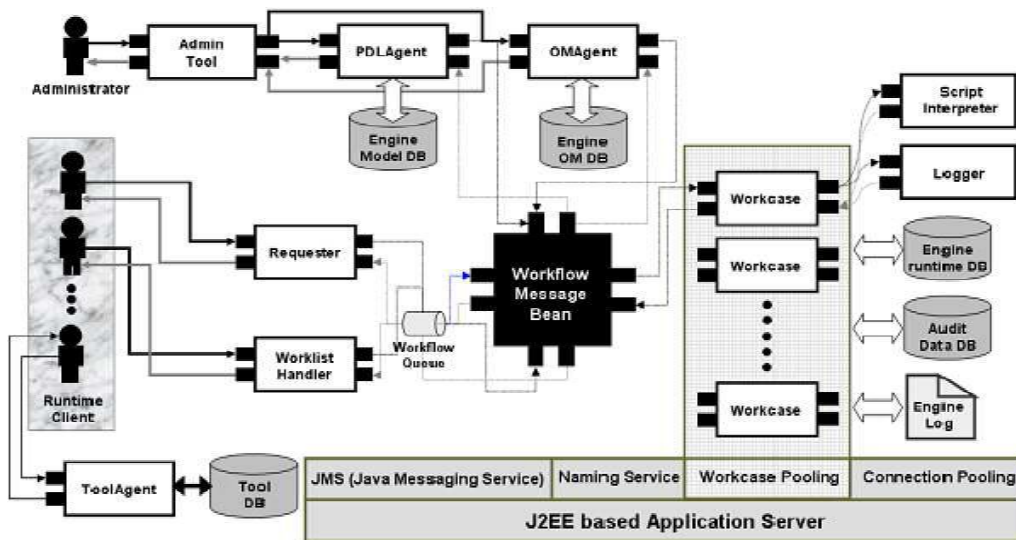
- 도 1은 종래의 액티비티 기반 워크플로우 구조를 나타내는 구성도.
- 도 2는 본 발명에 따른 워크케이스 기반의 워크플로우 엔진의 구조도.
- 도 3은 본 발명에 따른 워크케이스 기반의 워크플로우의 구조도.
- 도 4는 본 발명과 종래의 워크플로우 구조 간 성능 비교 분석도.

도면

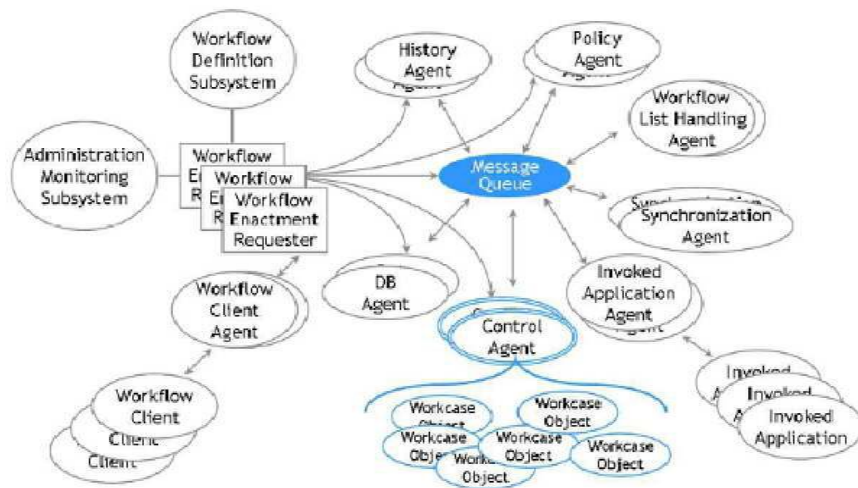
도면1



도면2



도면3



도면4

