

에이전트를 적용한 프로세스 관리 모델 설계 및 구현[☆]

Design and Implementation of Process Management Model applying Agent Technology

김정아^{*}
JeongAh Kim

최승용^{**}
SeungYong Choi

배제민^{***}
JeMin Bae

요 약

지식 기반 사회가 구축되면서 처리해야 할 업무 프로세스의 규모가 커지고 분석할 정보의 양이 증가함으로 인해 프로세스 관리와 개선에 대한 필요성이 높아지고 있다. 본 논문에서는 비즈니스 프로세스 관리를 위해, 프로세스 수행에 필요한 명확한 규칙과 정의를 관리하는 룰 에이전트를 적용하여 프로세스의 정확한 수행을 가능하게 하고 일정관리 에이전트를 적용하여 프로세스 수행에 투입된 개개의 일정을 정확하게 측정하고 관리할 수 있도록 했다. 에이전트를 적용한 프로세스 관리 모델은 조직 차원에서 비즈니스 프로세스의 자동화와 프로세스 중복 제거를 통한 프로세스의 통제성, 유연성, 민첩성, 신뢰성, 재사용성을 높이고 개인 차원에서는 개인의 프로세스 역량을 측정하고 프로세스 개선점을 식별하여 개선할 수 있도록 설계 및 구현하였다.

Abstract

As the knowledge-based society has been constructed, the size of work process that has to be done grows big and the amount of the information that has to be analyzed increases. In this paper business process can be accurately executed by the rule agent to manage precise rules and definitions to be needed by execution of process for management of business process. And individuals can accurately measure and manage personal schedules to execute process through supporting the agent of schedule management. The model is designed and implemented for organization to improve process of control, flexibility, agility, reliability, and reuse through business process automation and removing process overlapping and for personal to improve measuring personal process capacity and distinguishing process weak points.

☞ Keyword : 에이전트(Agent), 프로세스(Process), 프로젝트 관리(Project Management), PSP, 6시그마(Six Sigma)

1. 서 론

1980년대 관계형 데이터베이스 관리 시스템의 기술 발전으로 인해, 기업에 구축된 대부분

의 비즈니스 관리 시스템은 데이터 접근 관점에서 설계되었다. 그리고 해당 도메인에 대한 데이터의 수집, 저장, 조회와 관련된 데이터 관리에 높은 생산성을 보이며 기능 중심의 업무 구조에 중요한 역할을 담당했다. 그러나 데이터 접근 관점에서 설계된 비즈니스 관리 시스템은 기능 중심의 독립된 데이터 관리로 인해 데이터 구조가 정적이며 외부 데이터와의 연계성 및 통합성이 부족했다. 따라서 비즈니스가 갖고 있는 역동성에 대한 변화 관리에 적절히 대처할 수 없는 한계를 보였다.

1990년 초반 M. Hammer는 비용, 속도, 품질, 고객만족 등 핵심적 비즈니스 성과의 향상

* 정회원 : 관동대학교 컴퓨터교육과 교수
clara@kd.ac.kr

** 정회원 : 성균관대학교 정보통신공학부 컴퓨터공학과 박사과정
boromi@gmail.com

*** 정회원 : 관동대학교 컴퓨터교육과 교수,
교육과학연구소 소장
gemini@kd.ac.kr

[2006/03/20 투고 - 2006/03/21 심사 - 2006/09/06 심사완료]

☆ 본 논문은 관동대학교 2006년 학술연구지원사업의 결과입니다.

본 논문은 정보통신연구진흥원의 정보통신선도기반기술 개발사업의 연구결과로 수행되었습니다.

을 위해 기업의 업무 프로세스를 근간으로 비즈니스 시스템을 근본적으로 재설계할 것을 주장하며 BPR(Business Process Reengineering)[1]을 통해 프로세스 혁신의 필요성을 강조했다. 이로 인해 프로세스 관리에 대한 정보 기술 연구가 활발히 진행 되면서 IBM의 FlowMark[2], AT&T의 ProcessIT, Xerox의 InConcert 등 대표적 워크플로우 관리 시스템(WfMS: Workflow Management System)이 등장하게 된다. 워크플로우 관리 시스템의 표준화를 진행하고 있는 WfMC(Workflow Management Coalition)에서는 워크플로우를 문서, 정보, 태스크가 한 사용자(애플리케이션)에서 다른 사용자로 일련의 업무절차 규칙에 의한 처리를 위해 전달되는 비즈니스 프로세스의 자동화로 정의하며 워크플로우 관리 시스템을 하나 또는 그 이상의 워크플로우 엔진을 실행하는 소프트웨어의 이용을 통하여 워크플로우의 수행을 정의, 생성, 관리하는 시스템으로 정의하고 있다[3].

최근, 워크플로우 기술은 웹서비스, BPM(Business Process Management)[4], EAI(Enterprise Application Integration) 등 여러 정보 기술 분야와 접목되고 있으며 분산 환경을 지원하기 위해 WfMC와 OMG(Object Management Group)가 협력하고 있다.

2000년대에 들어서면서 BPM 기술이 워크플로우 기술을 계승하면서 정보기술의 핵심으로 부상하고 있다. M. Hammer(2003년)는 BPM을 프로세스 관점에서 기업을 경영하는 것이라 정의했으며, 가트너는 BPM을 민첩성과 운용 효율을 증대하기 위해, 비즈니스 프로세스 환경을 통제하는 일상적인 경영 활동으로 방법론, 정책, 측정 지표, 일상적인 경영 활동과 소프트웨어 도구를 활용해 조직의 업무와 프로세스를 끊임 없이 최적화하는 구조적인 접근 방법이라 정의하고 있다. BPM은 BPR, 워크플로우 등과 유사하지만 가치 사슬 상에 있는 모든 이해 관계자들(공급자에서부터 고객까지)의 프로세스를 통

합·운영하는 것을 목적으로 한다는 점에서 차이가 있다. BPM은 BPR과 달리 기존의 프로세스를 중심으로 점진적으로 개선하기 때문에 조직 구성원들이 프로세스 중심 업무 환경에 쉽게 적응할 수 있으며 주기적으로 프로세스를 개선하고 실시간으로 이를 반영하므로 인해 프로세스 개선에 대한 신뢰도가 높고 변화에 따른 위험을 최소화시킬 수 있는 장점을 갖고 있다. M. Hammer(2003년)는 BPM의 도입 효과로 주문 처리 프로세스의 리드 타임을 60~90% 정도, 구매를 위한 리드 타임을 90% 이상 단축시킬 수 있으며 신제품 개발 리드타임을 30~50% 정도 향상시킬 수 있다고 밝히고 있다[5].

최근, 시장 환경은 비즈니스 경계가 모호해지면서 기업의 생존 전략으로 작용했던 자동인자(enabler)가 반대로 소유하고 있던 시장 점유율을 잠식하고 기업의 생존을 위협하기도 한다. 이와 같은 시장 환경은 경영진의 더 정확하고 빠른 의사 결정을 요구한다. 이에, 기업들은 프로세스 관리의 핵심 정보 기술로 부상하고 있는 BPM을 도입하여 BPMS를 구축·운영하는 사례가 증가하고 있다. 그러나 기업들이 적극적으로 도입하고 있는 BPM의 문제점 중 첫 번째는 프로세스의 동적 변화와 복잡성으로 인해 프로세스 가시화 및 관리·통제가 쉽지 않아 BPMS를 해당 업무에 대한 자동화 시스템이나 기록 보관 시스템으로 단순 활용되는 경향이 있다는 점과 두 번째는 조직 차원에서 프로세스 관리를 운영함으로써 개인 차원의 프로세스 관리가 상대적으로 소홀하다는 점이다.

본 논문에서는 위와 같은 BPM의 문제점을 해결하고자 프로세스 관리 및 개선의 방법으로 에이전트 기술과 프로세스 개선 방법론인 6 시그마 프로세스를 적용하여 프로세스의 가시성을 확보하고 프로세스 관리를 용이하게 하고자 했다. 또한, 프로세스 수행 활동에 대한 측정 도구로 PSP의 일정 관리 기법을 도입하고 이를 에이전트가 관리하도록 하여 조직 및 개인 차원에

서의 정확한 프로세스 수행을 지원하고자 했다.

본 논문에서 제안하는 모델은 전체 종단 간(end-to-end) 비즈니스 프로세스를 공유하는 플랫폼을 지원함으로써 가치사슬 관점에서 조직에 최적화된 프로세스를 전개할 수 있게 한다. 또한, 프로세스 수행의 주체인 개개인의 역량에 대한 정확한 데이터의 수집과 반영이 가능하여 프로세스 수행 결과의 결함을 미리 예방할 수 있다. 이와 같은 프로세스 관리 환경을 기반으로 프로세스 개선점을 식별하고 프로세스 중심의 업무 문화(혹은 시스템)를 조직에 정착시키는데 일조할 것으로 기대된다.

본 논문에서 제안하는 모델의 목표는 다음과 같다. 조직 차원에서는 비즈니스 프로세스의 가시성 확보 및 자동화와 프로세스 중복 제거를 통해 프로세스의 통제성, 유연성, 민첩성, 신뢰성, 재사용성을 높이는 것이다. 개인 차원에서는 개인 프로세스 역량을 측정하고 개선점을 식별하여 개인의 생산성이 향상되도록 지원하는 것이다.

본 논문 중 2장에서는 본 논문에서 제안하는 모델에 적용된 에이전트, 6시그마, PSP의 장점 및 필요성을 고찰한다. 3장에서는 배경 지식을 바탕으로 에이전트 개념을 적용한 프로세스 관리 모델을 제안한다. 4장에서는 구현의 예와 기대 효과를 살핀다. 5장에서는 결론 및 향후 연구 과제를 언급한다.

2. 관련 연구

2.1 Agent

에이전트(Agent)[6]는 원하는 목적을 위해 사용자를 대신하여 작업을 하는 자율적 프로세스(autonomous process)로 독자적으로 존재하지 않으며 어떤 환경의 일부 또는 안에서 동작하는 시스템을 말한다. 즉, 에이전트는 사용자의 요구를 달성할 책임(goal-oriented)을 갖고

스스로 활동(self-starting)하는 능력을 갖고 있으며 일반적으로 사용자를 대신해서 원하는 작업을 자동적으로 해결해 주는 소프트웨어로 해석될 수 있다.

에이전트는 지식을 기반으로 한 내부 처리를 위해 지식베이스를 가지며 추론 및 학습 기능을 가진다. 문제 해결을 위해서 다른 에이전트나 자원과 정보를 교환하기도 하며, 자신이 속해 있는 환경의 변화가 있을 때 이를 감지하여 변화된 환경에 적응하는 행동을 취하기도 한다. 또한, 경험을 바탕으로 학습을 취하기도 하며, 주어진 작업을 수동적으로 처리하는 것이 아니라 에이전트 자신에게 주어진 목적을 달성하기 위해 능동적으로 행동을 한다. 이런 에이전트의 행동 결과로 자기 자신은 물론 환경의 변화도 가져올 수 있으며, 에이전트의 행동은 한 번에 끝나는 것이 아니라 일정기간 지속적으로 이루어진다.

따라서 지능(intelligence)을 바탕으로 에이전트는 같은 작업이라도 계획과 경험을 통해 더 나은 효과를 기대할 수 있다. 또한, 사회성(social ability)을 바탕으로 자신의 목표를 이루기 위해 다른 에이전트와 여러 가지 방법으로 상호작용하여 환경의 변화를 지각하고 내부의 지식과 외부와의 교환된 정보를 바탕으로 추론하고 그 결과에 따라 행동할 수 있는 능력을 지닌다.

위와 같은 에이전트의 특징은 비즈니스 프로세스 관리에 룰(Rule) 에이전트와 일정관리 에이전트 적용을 가능하게 한다. 따라서 프로세스 실행 시 프로세스 수행에 필요한 다양한 전제 조건 및 제약 사항에 대한 룰을 관리하는 룰 에이전트가 업무 담당자의 업무 활동에 대한 정확성을 지원하고 일정 관리 에이전트가 일정 관리를 지원하도록 적용할 수 있다.

2.2 PSP

PSP(Personal Software Process)[7]는 개발자

가 수행하는 작업에 공학적 접근 방법을 제시한다. 개인의 업무 계획과 업무 수행 능력을 측정·비교해서 추적할 수 있도록 계량화된 기준과 방법을 제공하고 있다. PSP는 ‘어떻게’에 해당하는 일하는 방법에 대한 직접적인 지침을 제공하기 때문에 개인 작업에 대한 계획 능력을 향상시켜 개인의 업무 생산성을 높일 수 있게 해준다. PSP에서 제공하는 자료 수집 방법, 측정 기법, 품질 향상 기법은 소프트웨어 분야 뿐만 아니라 시간 및 결함을 관리해야 하는 비즈니스 전 부문에 적용할 수 있다. 이에 본 논문에서는 PSP의 일정 관리 기법을 제안하는 모델에 적용하여 일정 관리를 담당하는 PSP 에이전트에 의해 프로세스 통제가 가능한 비즈니스 프로세스 관리 환경을 제공하고자 한다.

2.3 6시그마

6시그마[8]는 과학적 통계 기법을 적용하여 모든 업무에서 결함의 원인을 찾아내 분석하고 개선하고자 하는 전사적 활동으로 과학적 통계 기법을 기반으로 업무의 효율성과 고객 만족도의 향상을 통해 경영 성과를 높이고자 하는 경영 혁신 기법이다. 6시그마는 프로세스 개선 방법론을 전개하는데 필요한 다양한 통계적 도구를 지원하고 있으며 프로세스를 이해, 관리, 개선하는데 도움이 되는 것이면 어떤 기법이든 6시그마의 도구로 활용할 수 있다.

이에, 본 논문에서는 DMAIC(Define, Measure, Analyze, Improve, Control) 프로세스[9]의 장점을 제안하는 시스템에 적용하여 6시그마의 프로세스가 갖는 장점을 비즈니스 프로세스에 수월하게 응용할 수 있도록 자동화된 환경을 제공하고자 한다.

3. 프로세스 관리 시스템: 머큐리

비즈니스 프로세스 관리에서는 조직의 하부

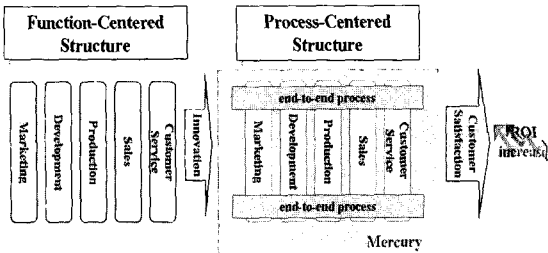
구조에서 실행되고 있는 기능 단위 업무의 자동화를 실현하고 운영되고 있는 독립 부서간의 워크플로우의 중복을 제거해야 한다. 그리고 이를 기반으로 조직 전반의 프로세스에 대한 가시성을 높여 조직의 내·외부의 프로세스간의 프로세스 상호작용이 원활하도록 관리할 필요가 있다. 또한, 비즈니스 프로세스 개선을 위해서는 정확한 프로세스 수행 데이터의 수집이 필요하다.

본 논문에서 비즈니스 프로세스의 수립, 운영, 개선, 평가 등 프로세스와 관련된 전 과정을 지원하게 되는 에이전트를 적용한 프로세스 관리 모델을 머큐리(Mercury)로 명한다.

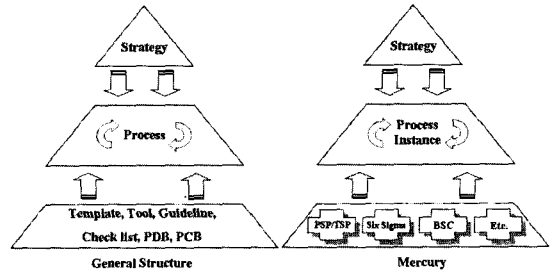
3.1 에이전트를 적용한 프로세스 중심 환경

머큐리는 기업에서 창조되는 비즈니스 프로세스의 전 영역을 범위로 한다. 머큐리를 구축하여 운영을 하게 되면 <그림 1>에서 보여주는 프로세스 중심 구조의 환경을 조직에 정착시킬 수 있다.

기능 중심의 업무 구조에서는 분업화된 업무에 기반을 두고 프로세스를 독립적으로 운영하면서 다른 독립 부서 또는 사업 단위와의 정보 소통이 원활이 이루어지지 않는 경직된 상태로 운영하는 경향이 있다. 그리고 해당 업무에 대한 데이터의 수집, 저장, 조회와 관련된 데이터 관리에 집중하며 부분 최적화를 추구함으로써 end-to-end optimization)를 손상시키는 결과를 초래한다. 기능 중심 업무 구조를 가진 기업에서 구축한 비즈니스 관리 시스템은 데이터 접근 관점에서 설계된 경우가 일반적이기 때문에 데이터 구조가 정적이며 외부 데이터와의 연계성 및 통합성이 부족하다. 따라서 비즈니스가 갖고 있는 역동성에 대한 변화 관리에 적절히 대처할 수 없게 된다.



<그림 1> 머큐리 기반의 프로세스 중심 구조



<그림 2> 머큐리의 프로세스 접근 구조

프로세스 중심의 업무 구조에서는 조직 전체의 가치사슬을 전사에 걸쳐 횡적으로 연결한 종단 간(end-to-end) 프로세스에 의해 운영된다. 따라서 변동이 심한 고객 니즈에 대응하기 위한 동적 업무 환경이 가능해진다. 동적 업무 환경에서는 내·외부 프로세스의 통합 및 상호 운영이 원활해야 한다. 그리고 프로세스 데이터의 공유가 가능하고 데이터의 중복이 제거되어야 한다. 또한, 프로세스 중심의 동적 데이터 관리가 가능하여 프로세스의 신뢰성을 증진시켜야 한다. 이를 지원하기 위해서는 BPA(Business Process Automation) 환경 조성이 필수적이다. 머큐리는 BPA뿐만 아니라 BPM(Business Process Management)도 가능한 환경을 지원하여 <그림 1>의 프로세스 중심 구조가 정착되도록 돕는다.

머큐리는 프로세스 관리 및 개선을 지원하기 위해 하향식(top-down) 프로세스 접근 방법을 지원한다. 하향식 접근 방법은 수행 일정이 구체적으로 나오도록 프로세스를 큰 단위에서 작은 단위로 계속 분할하여 전체 프로세스 구조가 체계적인 흐름을 가지도록 구조화한다. 따라서 머큐리 사용자는 자연스럽게 단계(stage or phase), 활동(activity), 항목(task) 순으로 프로세스를 구성하게 되며 단계와 활동에 중점을 두면서 필요한 속성을 찾아 설계하게 된다. 머큐리는 보다 쉽게 프로세스의 하향식 접근 방법을 활용할 수 있도록 <그림 2>와 같은 기본 구조를 갖고 있다.

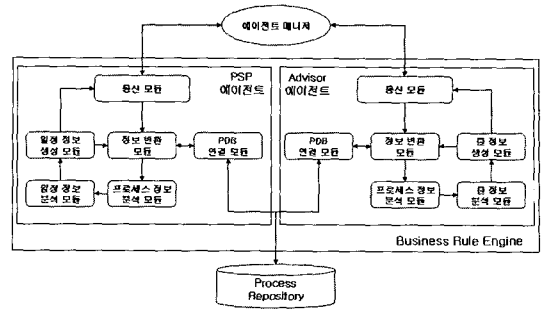
머큐리가 갖고 있는 프로세스 접근 구조는 추상적 수준(상부 구조)에서는 일반적인 프로세스 접근 구조와 같다. 즉, 조직의 전략이 프로세스의 생성과 소멸에 대한 직접적 영향을 미치며, 전략에서 나온 정책은 프로세스의 방향성을 제시하고 정책이 추구하는 목표는 프로세스의 성격을 결정한다. 그러나 프로세스의 가시성을 높이고 프로세스 통제의 유연성을 지원하는 구체적 수준(하부 구조)에서는 6시그마, PSP와 같은 공인된 프로세스 개선 기법과 도구를 이용하여 프로세스 수행을 지원한다.

3.2 머큐리의 기본 구조

변화는 비즈니스가 갖는 기본 요소이다. 이로 인해 비즈니스 프로세스는 역동성, 복잡성을 기본 성격으로 가지고 있다. 이 때문에 동적 성질을 가진 프로세스를 관리, 통제 및 개선하는 것은 사실 상 어렵다. 그럼에도 기업은 비즈니스 프로세스의 운영 상태 파악과 통제할 수 있는 방법을 원한다.

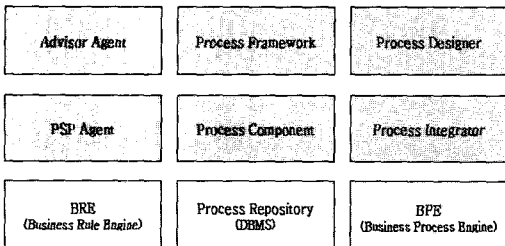
본 논문에서 제안하는 머큐리는 비즈니스를 기반으로 프로세스 수행을 지원하며 수행 데이터를 바탕으로 개선의 필요성과 대상을 식별한다. 비즈니스 프로세스의 정확한 실행을 위해서는 프로세스 수행에 필요한 제약 사항들 및 체크리스트가 명확해야 하고 가시성 확보가 보장되어야 한다. 제안하는 모델에서는 틀 에이전트의 지원으로 프로세스 수행에 필요한 지침과

정확성에 대한 검증을 받을 수 있다. 또한, 본 모델에 수집하고자 하는 평가 항목을 입력하고 실행하면 평가 항목의 수행 결과 데이터를 제공 받을 수 있다. 이렇게 수집된 데이터는 6시그마 등 프로세스 평가 및 개선 방법론의 분석단계를 수행하는 자료로 활용할 수 있고 프로세스 개선 영역을 식별하는 기초 자료가 된다. 식별된 개선 영역에 대한 개선안을 마련한 후, 다시 본 모델을 통해 개선된 비즈니스를 반영하여 프로세스를 실행할 수 있고 이러한 프로세스 개선 사이클은 반복적, 지속적으로 실행이 가능하다. 그러므로 정확하고 민첩하게 프로세스를 실행할 수 있도록 지원하는 머큐리를 통해 RTE(Real-time Enterprise)를 실현할 수 있다. 따라서 머큐리는 <그림 3>과 같은 구조를 가지고 <표 1>을 위한 환경을 지원하는 것을 목표로 한다.



<그림 4> 머큐리에서 지원하는 에이전트 구조

머큐리는 <그림 4>의 구조로 구성된 PSP 에이전트와 Advisor 에이전트를 지원한다. 조직 및 개인의 프로세스 일정 관리를 지원하는 PSP 에이전트는 통신 모듈을 통해 에이전트 매니저의 호출과 메시지 전송을 주고받는다. 에이전트 매니저로부터 전송된 메시지는 정보 변환 모듈을 통해 분석된 뒤 질의로 변환된다. 질의는 PDB 연결 모듈을 통해 프로세스 리퍼지토리에 전달되고 질의에 대한 모든 일정 관련 프로세스 정보는 다시 PDB 연결 모듈을 통해 정보 변환 모듈에 전달된다. 정보 변환 모듈은 전달된 질의에 대한 결과를 프로세스 정보 분석 모듈이 처리할 수 있는 메시지로 변환하여 전달한다. 프로세스 정보 분석 모듈은 전달된 메시지를 분석하여 프로세스 정보로 가공



<그림 3> 머큐리의 기본 구조

<표 1> 머큐리의 지원 목표

목 표	상세 설명
프로세스 정의, 실행, 통제, 개선이 가능한 통합 기반 환경	BPE(Business Process Engine)를 기반으로 프로세스를 정의하고 통합하여 실행할 수 있는 환경이다. BRE(Business Rule Engine) 기반으로 프로세스 실행에 필요한 제약 사항 및 규칙을 제공하는 Advisor 에이전트와 프로세스 수행 시간 및 수행 결과 크기를 측정하는 PSP 에이전트에 의해 실행되고, 그 결과를 바탕으로 개선할 수 있는 환경이다.
이미 검증된 프로세스를 사용자에게 제공하는 지식 기반의 프로세스 중심 업무 환경	조직에서 여러 번의 실행과 개선 과정을 거쳐서 품질을 보장할 수 있는 프로세스를 조직의 프로세스 자산으로 저장소에 등록할 수 있는 환경이다. 단일 프로세스 수준의 프로세스 컴포넌트와 프로세스간의 연결 관계가 이미 정의된 대규모의 프로세스 프레임워크 모듈을 재사용 가능한 지식으로 전제한다.
프로세스 운영을 통해 개인과 조직의 성숙도가 높아지는 개선 환경	머큐리는 궁극적으로 프로세스 수행의 정확성과 신속성을 보장하고 개선된 프로세스의 수행으로 통해 자연스럽게 개인과 조직의 성숙도 개선을 가능하게 하는 환경을 목표로 한다.

하고 일정 정보 분석 모듈은 분석된 프로세스 정보를 기반으로 일정 정보를 분석하게 된다. 분석된 일정 정보는 일정 정보 생성 모듈에 의해 취합되어 통신 모듈을 통해 에이전트 매니저에게 전달되거나 프로세스 리파지토리에 저장된다. PSP 에이전트는 PSP의 일정 관리 기법을 활용하여 정의한 각 업무에 대한 소요 시간을 확인할 수 있게 지원한다. 그러므로 조직 및 개인은 실제 업무 수행 방식에 대한 시간 사용 방식을 관리하는데 효과적이다.

를 에이전트인 Advisor 에이전트는 PSP 에이전트와 같은 방식으로 에이전트 매니저 또는 프로세스 리파지토리와 통신을 한다. Advisor 에이전트의 프로세스 정보 분석 모듈은 수집된 프로세스 정보를 비즈니스 워크플로우와 비즈니스 서비스로 구분하여 분석한다. 분석된 정보는 를 정보 분석 모듈에 전달되고 도메인이 가지고 있는 가변적 특성을 해석하는 비즈니스 도메인 룰과 업무를 처리하는데 필요한 작업 종류, 순서, 처리 조건을 해석하는 비즈니스 프로세스 룰로 분류되어 분석된다. 를 정보 생성 모듈은 분석된 를 정보를 취합하여 통신 모듈을 통해 에이전트 매니저에게 전달하거나 새로 생성되거나 변경된

를 정보를 프로세스 리파지토리에 저장한다. 룰은 하나 이상의 비즈니스 프로세스의 행위를 통제하는 규칙으로써 비즈니스 전략과 규칙의 가변성에 대해 유연하게 대응할 수 있도록 비즈니스 도메인 룰과 비즈니스 프로세스 룰로 분리할 필요가 있다. Advisor 에이전트는 가변성을 갖는 비즈니스 규칙 및 전략을 분리하여 비즈니스 프로세스의 확장성 및 유연성을 지원한다. 그러므로 프로세스 실행 시 필요한 지침을 제공받을 수 있기 때문에 조직은 프로세스의 정확한 관리가 가능해지고 업무 담당자는 업무 활동에 대한 정확성을 높일 수 있는 장점이 있다.

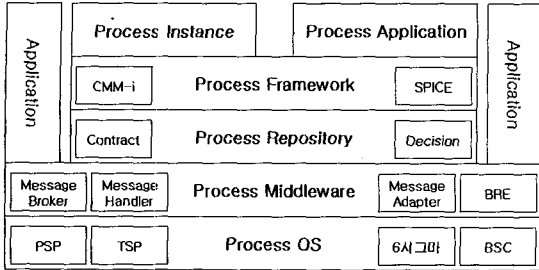
3.3 머큐리의 시스템 구조

머큐리를 활용하는 조직은 업무에 문혀 잘 들어나지 않는 프로세스의 가시성을 확보하여 업무의 워크플로우를 정확히 이해하고 관리할 수 있다. 이와 같은 프로세스 관리 환경을 지원하기 위해 머큐리는 <그림 5>의 개념 모델을 기반으로 한다. 프로세스의 통제성, 유연성, 민첩성, 신뢰성, 재사용성의 수준 향상과 관련된 컴포넌트는 <표 2>와 같으며 컴포넌트들의

<표 2> 프로세스 관리와 관련된 머큐리의 컴포넌트

특 성	정 의	주요 관련 컴포넌트
통제성	프로세스의 실시간 모니터링을 통하여 프로세스 실행 상태를 파악 및 관리하는 능력.	Process OS
유연성	IT 환경의 표준화 및 IT 신기술을 수용 또는 확장하여 시장 환경 및 고객 요구 사항에 대응하는 프로세스를 정확히 구현할 수 있는 능력.	Process Middleware
민첩성	시장 환경 및 고객 요구 사항에 대응하여 필요한 경우 프로세스를 정의·수정하고 새로운 비즈니스 규칙을 적용할 수 있는 능력.	Process Framework
신뢰성	요구 조건(시간, 수량, 품질 등)에 의해 정의된 프로세스(업무 규칙)가 일관된 결과물을 생산하는 능력.	Process Framework Process Repository
재사용성	프로세스 리파지토리에 축적된 업무 프로세스를 활용하여 유사한 업무에 프로세스를 적용하거나 반복적으로 사용하는 프로세스에 사용할 수 있는 능력.	Process Framework Process Repository

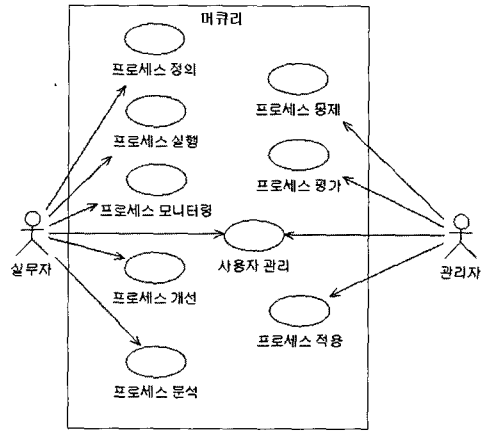
주요 역할 및 상호 관계는 <표 3>과 같다.



<그림 5> 머큐리의 개념 모델

머큐리를 구성하는 Process OS, Process Middleware, Process Repository, Process Framework 컴포넌트는 머큐리의 핵심 컴포넌트으로써 프로세스의 생성, 수행, 통합, 개선, 적용에 대한 역할과 책임(R&R: Role & Responsibility)을 담당한다. 머큐리의 서비스를 이용하는 사

용자와 머큐리간의 상호작용을 표현한 <그림 6>은 사용자 관점에서 머큐리의 활동을 설계한 것으로 머큐리가 사용자에게 지원하는 기능적 범위를 나타내고 있다.



<그림 6> 머큐리의 유스케이스 모델

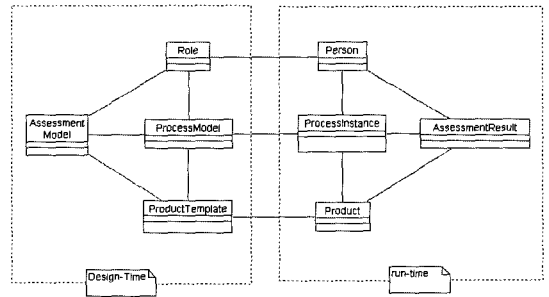
<표 3> 머큐리를 구성하는 컴포넌트 역할

컴포넌트	주요 내용
Process OS	머큐리에 의해, 생성에서부터 소멸되기까지의 모든 프로세스의 전체 라이프 사이클을 통제하고 프로세스와 관련된 자원 관리(PSP, TSP, 6시그마, BSC(Balanced Score Card))를 담당한다. Process Middleware와의 상호작용을 위한 인터페이스를 제공한다.
Process Middleware	Process OS 컴포넌트, Application에서 넘어오는 프로세스 관련 메시지들이나 요청 데이터들을 컴포넌트의 상호작용을 돕는 형태로 변환하여 해당 목적지와 연결시켜줌으로써 원활한 자원 이용을 위한 프로세스간의 상호작용을 돕는 통신 수단(예: 프로토콜, 데이터 포맷) 및 제반 환경 요소를 제공한다. 외부 서비스(Operating System)와의 통신을 담당하고 외부 서비스가 가지고 있는 제어권을 주고받음으로써 프로세스의 유연성 및 확장성을 높인다.
Process Repository	다양한 프로세스 컴포넌트가 안정적으로 저장된다. 프로세스 데이터의 일관성과 무결성을 보장하여 프로세스의 신뢰성 재사용성을 높인다.
Process Framework	도메인과 관련된 프로세스들의 구성 요소간의 유기적 협력관계를 정의하여 검증된 메커니즘을 제공하여 프로세스의 민첩성, 신뢰성, 재사용성을 높인다.
Process Component	도메인과 관련된 프로세스의 독립된 기능을 제공하는 소프트웨어 모듈.
Process Instance	Process Framework 또는 Process Component가 업무에 적용되어 실행중인 프로세스.
Process Application	프로세스가 적용되어 실행중인 업무.
Application	프로세스 실행 중 다양한 서비스를 제공하는 응용 프로그램으로 워드, 파워포인트, 아웃룩 등의 특정 업무 시스템을 의미한다.

머큐리는 조직의 목적 달성을 지원하기 위해 6시그마에서 제시하는 프로세스 개선 방법론인 DMAIC 프로세스의 장점을 지원한다. 6시그마의 프로세스 개선 방법론을 조직의 프로세스에 응용하기 위해서는 고유의 조직 문화 및 업무 특성에 맞게 선택적으로 흡수해야 할 필요가 있다. <표 4>는 머큐리가 사용자에게 지원하는 기능을 유스케이스 단위로 머큐리 활동을 상세화한 것이다.

머큐리가 프로세스 중심의 업무 활동을 지원하기 위해서는 유연하고 동적인 프로세스 통합 관리 기능을 제공해야 하므로 프로세스를 구성하는 요소들의 긴밀한 상호 작용을 보장해야 한다. 프로세스 정의 과정에서는 ProcessModel 객체를 중심으로 Role, ProductTemplate, Ass-

essmentModel 객체가 구조적으로 연결되어 통합 관리를 수행하며 프로세스 실행 과정에서는 ProcessInstance 객체를 중심으로 프로세스 실행을 담당하는 객체들과 필요한 정보를 주고받을 수 있게끔 머큐리가 지원한다<그림 7>.



<그림 7> 머큐리의 객체 모델

<표 4> 사용자에게 지원하는 머큐리 활동

분 류	머큐리 지원 기능	
프로세스 정의	프로세스 전략 및 목표 설정. 프로세스 범위 및 제약 설정. 프로세스 요구 사항(CPR: Critical Process Requirement). 파악 및 품질 결정 요소(CTQ) 설정. 프로세스 수락 기준설정.	프로세스 절차 수립. 프로세스 절차 별 수행 조건 수립. 프로세스 가시성 확보 방안 수립. 프로세스 공유 방안 수립. 프로세스 개선 절차 수립.
프로세스 실행	업무별 최적 프로세스 식별. 프로세스 인스턴스 생성.	수행 작업 결정 및 작업자에 통보. 작업과 연결된 애플리케이션 실행.
프로세스 모니터링	프로세스 진행 상황 파악. 프로세스 수행 시간 및 사이클 타임의 추적과 누적 관리. 프로세스 변경 사항추적 및 누적 관리. 프로세스 데이터 및 측정 데이터 유형 파악.	CTQ에 대한 인자(factor) 파악. CTQ에 대한 프로세스 현 수준 및 핵심 개선 영역 파악. 핵심 개선 영역에서의 잠재적 원인 도출. Vital Few 추적 및 관리.
프로세스 분석	개선 대상프로세스 식별. 핵심 원인 인자(Vital Few) 파악. 및 의미 분석.	프로세스 최적조건 도출. 프로세스 개선효과 분석. 프로세스 개선비용 산정.
프로세스 개선	불량 프로세스 삭제. Vital Few 중심의 프로세스 재구성.	프로세스 간소화 및 최적화. 프로세스 자동화로 인한 성능 향상.
프로세스 통제	모니터링 시스템 분석 및 점검. 프로세스 표준화 점검.	프로세스 준수 상황 점검.
프로세스 적용	개선 프로세스에 대한 관계자 교육. 개선 프로세스의 공유 및 확산.	애플리케이션 및 타 프로세스로의 전파.
프로세스 평가	BSC에 의한 프로세스 성과 평가. PSC(Process Score Card)에 의한 프로세스 품질 평가.	MBO(Management By Objectives)에 의한 목표 성취 평가.
사용자 관리	사용자의 사원 정보 관리. 사용자의 업무 영역 식별 및 관리.	사용자의 업무 진행 상황 관리.

프로세스 통합 관리 및 개선 기능을 제공하기 위해 객체들의 역할 및 상호 작용에 대한 시스템 관점에서의 정의는 <표 5>와 같다.

머큐리는 프로세스 정의 단계에 생성되는 객체를 머큐리의 Process Repository에 저장한다. Process Repository에 저장된 ProcessModel의 객체 정보에 의해 프로세스를 실행한다. 프로세스 실행은 ProcessInstance 객체로 실현한다. ProcessInstance 객체로부터 프로세스 수행 산출물(Product)이 생산되면 정의 단계의 AssessmentModel 객체가 갖고 있는 정보, 즉, PCB (Process Capability Baseline)에 의해 결정된 평가 기준과 Process OS 컴포넌트의 평가 전략 및 도구(PSP, TSP, 6시그마, BSC 등)에 의해 결정된 평가 방법을 이용하여 AssessmentResult 객체가 이를 평가한다.

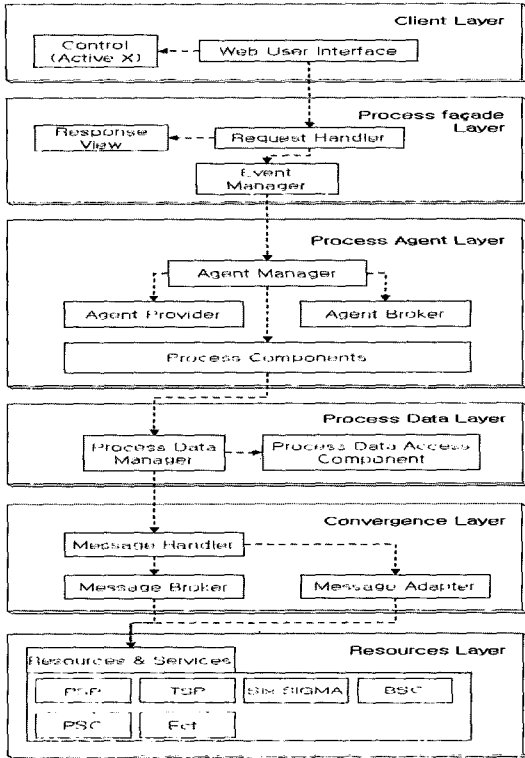
시장의 변화가 심한 현실 속에서, 조직의 전략은 시장의 변화에 맞춰가게 마련이다. 전략의 변화는 관련 업무를 생성, 변화, 소멸시킨

다. 이는 조직 프로세스의 변화를 의미하는 것이다.

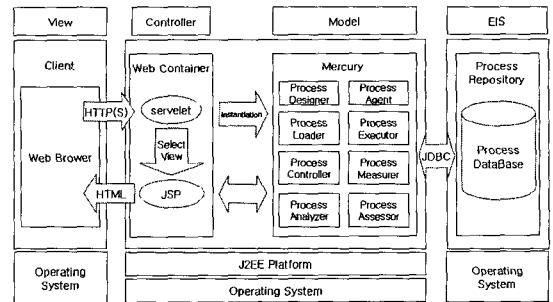
머큐리는 조직 내부의 다양한 구조적 변화에 민첩하게 대응하며 고효율, 저비용의 안정적 프로세스 운영을 지원하기 위해 머큐리는 내부적으로 레이어 구조를 가진다. 머큐리는 클라이언트 레이어를 제외하면 핵심 레이어 계층으로 5개의 계층을 가지고 있다. 레이어 방식은 엄격한 레이어(strict layering) 방식과 유연한 레이어(flexible layering) 방식 중에서 유연한 레이어 방식을 사용한다. 유연한 레이어 방식은 같은 레벨의 레이어와 하위 레벨의 어떤 레이어하고도 상호 작용이 가능하여 유연성이 증가된다. 머큐리의 각 레이어가 갖고 있는 구성 요소들이 서로 상호작용하는 논리적 모습은 <그림 8>과 같으며 <그림 8>에서 보여주는 머큐리의 레이어 구조 특징과 역할은 <표 6>과 같고 레이어 구성 요소들이 갖는 주요 특징과 역할은 <표 7>과 같다.

<표 5> 머큐리의 객체 정의

객 체	정 의
AssessmentModel	프로세스 정의 단계에서 생성되는 객체. 평가 기준 및 평가 방법에 대한 프로세스 평가 모델을 정의.
Role	프로세스 정의 단계에서 생성되는 객체. 프로세스에 대한 R&R(Role & Responsibility)을 갖고 있는 액터.
ProcessModel	프로세스 정의 단계에서 생성되는 객체. 도메인에 적용될 프로세스 모델을 정의. 입·출력 산출물(product) 정보를 정의하고 있다.
ProductTemplate	프로세스 정의 단계에서 생성되는 객체. 산출물의 내용에 대한 템플릿(template).
Person	프로세스 실행으로 결정되는 객체. 프로세스 실행 주체인 액터.
ProcessInstance	프로세스 실행으로 생성되는 객체. 프로세스 실행 상태에 대한 데이터를 가진다.
Product	프로세스 실행으로 생성되는 객체. 프로세스 실행 중 또는 실행으로 나오는 데이터에 관한 산출물.
AssessmentResult	프로세스 실행으로 생성되는 객체. 산출물(product)의 데이터에 대한 평가 데이터.



〈그림 8〉 머큐리의 레이어 아키텍처



〈그림 9〉 머큐리의 시스템 아키텍처

〈표 6〉 머큐리의 레이어 구조 특징 및 역할

레이어	레이어 설명
클라이언트(Client)	사용자와 머큐리와의 상호작용을 담당한다. 사용자의 요청을 머큐리에 전달하고 머큐리에서 처리한 결과를 사용자에게 전달한다.
프로세스 퍼사드 (Process façade)	사용자가 보는 화면의 흐름이나 구성 방식을 처리하며 클라이언트 레이어와 프로세스 레이어를 연결해 준다. 클라이언트 레이어에서 넘어온 요청을 검증하고 알맞은 형태로 변환하여 프로세스 에이전트 레이어로 전달하거나 프로세스 에이전트 레이어의 응답을 알맞은 형태로 변환하여 클라이언트 레이어로 전달한다.
프로세스 에이전트 (Process Agent)	프로세스 퍼사드에서 넘어온 데이터를 검증한다. 에이전트의 활동을 통해 사용자 업무 프로세스를 실행하거나 프로세스 프레임워크나 프로세스 컴포넌트에서 새로운 데이터를 만들어 프로세스 데이터 레이어 또는 외부 시스템에 전달한다.
프로세스 데이터 (Process Data)	프로세스 에이전트 레이어나 외부 자원에서 넘어오는 프로세스 관련 데이터를 알맞은 형태로 변환하여 저장하고 요청하는 프로세스 데이터를 해당 레이어에 맞게 변환하여 전달한다.
수렴(Convergence)	머큐리와 다양한 외부 자원을 연결한다. 머큐리의 외부 자원에 대한 물리적 연결과 데이터 전달 및 변환과 같은 상호작용을 담당한다.
자원(Resources)	머큐리에서 사용하는 모든 자원 및 서비스가 해당된다.

<표 7> 머큐리 레이어의 구성 요소 특징 및 역할

레이어	레이어 구성 요소	설 명
클라이언트 (Client)	Web User Interface	웹 사용자와 머큐리 사이에서 상호작용을 담당한다.
	Control (Active X)	사용자 인터페이스만으로는 제공할 수 없는 다양한 기능을 제공한다. Active X, Applet, Plug-In처럼 사용자 인터페이스 외부에 있는 프로그램이나 라이브러리가 해당된다.
프로세스 퍼사드 (Process façade)	Request Handler	클라이언트 레이어에서 넘어온 사용자 요청을 받는다. 머큐리의 처리 결과에 따라 적절하게 응답 뷰를 제공한다.
	Response View	머큐리의 응답을 사용자가 볼 수 있는 형태로 만든다.
	Event Manager	사용자 요청을 머큐리에서 처리할 수 있는 형식으로 바꾼다. 사용자 요청을 에이전트 매니저에게 전달한다.
프로세스 에이전트 (Process Agent)	Agent Manager	에이전트의 활동을 통제하며 사용자 요청을 처리할 수 있는 에이전트를 찾아 호출한다. 호출된 에이전트가 할당된 요구사항을 처리할 수 있도록 프로세스 컴포넌트와의 연결을 제어한다.
	Agent Provider	에이전트의 이름, 역할 및 활동 범위에 관한 정보를 가지고 있으며 이를 에이전트 매니저에게 제공한다.
	Agent Broker	에이전트들의 상호작용을 위한 통신 수단을 제공하고 이를 관리한다. 에이전트 매니저에게 에이전트간의 통신 정보를 제공한다.
	Process Components	표준 프로세스 프레임워크를 해석, 갱신, 실행하며, 표준 프로세스와 연동된 사용자 업무 프로세스를 통합 또는 재구성하여 그와 관련된 서비스들을 실행하고 서비스간의 데이터 흐름을 제어한다.
프로세스 데이터 (Process Data)	Process Data Manager	프로세스 에이전트 레이어에서 넘어온 요구 데이터의 성격을 파악하여 해당 컴포넌트로 연결시켜 준다.
	Process Data Access Component	프로세스 관련 데이터들이 저장되어 있는 프로세스 리파지토리에 접근할 수 있는 매커니즘을 제공한다. 데이터베이스 시스템의 의존성을 벗어나 프로세스 데이터에 대한 작업 환경을 독립적으로 구성해줌으로써 유지 관리를 쉽게 해준다.
수렴 (Convergence)	Message Handler	프로세스 데이터 레이어에서 넘어온 요청을 받아 내부 자원으로의 연결 또는 외부 서비스로의 연결을 결정한다.
	Message Broker	내부 자원의 이용을 위해 컴포넌트들의 상호작용을 돕는 통신 수단을 제공한다.
	Message Adapter	외부 서비스와의 데이터 호환을 지원하는 데이터 포맷과 다양한 통신 표준을 통해 외부 서비스의 정보를 교환할 수 있는 프로토콜을 제공한다.
자원 (Resources)	Resources & Services	PSP, 6시그마, BSC같은 프로세스 개선 및 평가와 관련된 정보들을 라이브러리, 모듈, 컴포넌트 형태로 제공한다.

<표 8> 머큐리의 MVC Model 역할

MVC Model	역 할
Model	프로세스 컴포넌트를 직접 호출하지 않으며 여러 자바빈(JavaBean)으로 구성되어 있다.
View	주어진 프로세스 데이터 모델로부터 웹컨텐츠를 생성한다.
Controller	프로세스 컴포넌트를 호출하고, 추출된 데이터를 전달한다. 프로세스 컴포넌트를 호출한 결과에 맞는 뷰를 표시할 모델을 생성한다.

<표 9> 머큐리의 기능별 주요 컴포넌트

컴포넌트	역 할
Process Designer	비즈니스 프로세스 설계를 담당한다.
Process Agent	PSP 에이전트와 Advisor 에이전트의 활동을 제어하고 에이전트의 활동 결과를 분석한다.
Process Loader	비즈니스 프로세스를 실행하기 위한 초기 설정을 담당한다.
Process Executor	정의된 비즈니스 프로세스를 해석하고 실행한다.
Process Controller	실행중인 비즈니스 프로세스들의 우선 순위, 자원 할당 등을 조정한다.
Process Measurer	실행중인 비즈니스 프로세스가 제공하는 데이터들을 측정한다.
Process Analyzer	비즈니스 프로세스의 효율성을 분석한다.
Process Assessor	비즈니스 프로세스의 적합성을 평가한다.

4. 구현 및 적용 사례

본 본문에서 제안한 프로세스 관리 모델에 대한 타당성 및 효율성을 확인하기 위해 프로토타입을 구현하여 금융 IT 업체인 S사 개발팀의 보험 상품 설계 시스템 개발 프로젝트에 적용해 보았다. 구현 환경은 마이크로소프트사의 윈도우 2000 환경에서 J2EE 5.0 플랫폼을 기반으로 JSP Model 2 아키텍처로 구축되었다. 웹 서버(HTTP Server)로는 아파치(Apache) 2.0.54 버전을 사용하고 Servlet 컴파일을 위해 아파치 톰캣(Apache Tomcat) 5.0.28 버전을 사용했다. 데이터베이스는 Oracle 8i를 사용했다. 개발 언어는 프로세스 컴포넌트 구현을 위해 JAVA 5.0 버전을 사용했으며 웹 환경 지원을 위해 JSP 2.1 버전을 사용했다.

4.1 프로세스 등록

S사 개발팀의 팀원은 머큐리에 접속하면서 머큐리에서 지원하는 프로세스 관련 서비스를 제공 받는다. 프로젝트와 관련된 실무자는 머큐리를 통해 신분을 확인받고 본인의 해당 업무를 확인 하게 된다. 머큐리는 Advisor 에이전트를 통해 팀원 각자의 업무에 해당하는 프로세스의 유무를 프로세스 리파지토리를 통해 확인한다. 해당 업무에 대한 프로세스가 부재일 경우, Advisor 에이전트는 사용자에게 프로세스 생성을 요구하며 프로세스가 존재할 경우 프로세스 리파지토리에서 정의된 프로세스를 로딩하여 프로세스를 실행하게 된다. Advisor 에이전트에 의해 정의된 프로세스는 Advisor 에이전트와 PSP 에이전트에 의해 해석되고 통

제된다. 그러므로 프로세스의 수행에 대한 정확성과 효율성을 극대화 할 수 있게 된다. <그림 10>은 머큐리가 지원하는 Advisor 에이전트와 PSP 에이전트의 자동화된 환경을 업무와 관련하여 전적으로 지원받기 위해 개발 실무자가 정의한 프로세스이다.

4.2 프로세스 실행

머큐리의 PSP 에이전트는 시간 기록을 업무 단위로 하며 분 단위로 기록한다. 또한 작업 도중의 인터럽트 시간과 작업 결과도 함께 추적한다. 업무에 투입된 시간 대비 단위 작업

1 문서 기본 정보
1.1 이름
Insurance Reengineering 프로젝트, 약칭 InsRE
1.2 문서 목적
Insurance Reengineering Project의 관리를 용이하게 한다.
1.3 범위
Insurance Reengineering Project의 관리에 한정한다.
1.4 유지 책임자
PM : 허준욱
2 프로젝트 개요
2.1 품식 명칭
<ul style="list-style-type: none"> Insurance Reengineering 프로젝트. 약칭 : InsRE
2.2 기간
<ul style="list-style-type: none"> 프로젝트 준비기간 : 2005년 2월 1일 - 2005년 3월 31일 프로젝트 수행기간 : 2005년 4월 1일 - 2005년 8월 19일
2.3 배경
<ul style="list-style-type: none"> 가입설계를 할때 기계막 심사를 같이 해 달라는 요구사항이 많음 (은행) 기계막 데이터를 오픈시스템으로 내려서 기계약관 관련된 업무를 오픈 시스템에서 처리함으로써 해서 메인 프레임의 부하를 경감시킬 필요성 대두.
2.4 목표
<ul style="list-style-type: none"> 삼성 생명에 운영 가능한 안정적 가입설계 시스템 인도 IS 실에 확실한 인수 인계
2.5 서브 프로젝트
<ul style="list-style-type: none"> /발자2단계 /상품개발 /콜마이그레이션
3 관리공정
<ul style="list-style-type: none"> /조직관리 /품질관리 /변경관리 /인수인계 /회의록
4 개발공정
<ul style="list-style-type: none"> 소프트웨어크래프트표준체계/용어 /로열티서버모니터링
5 결과물
5.1 프로젝트 기획서
<ul style="list-style-type: none"> ● 기획

<그림 10> 머큐리의 프로세스 정의

2.2 계획 대비 진척률 (프로세스 타이머)

○ 소요시간단위 : hour (예 : 30분 = 0.5 / 1시간 = 1.0)

작업	계획일정과 계획시간			진행일과 진행시간			담당자	진행률	지연사유
	시작일	완료일	소요시간	시작일	완료일	소요시간			
상품분석및일정회의	2005-05-02	2005-05-02	2.5시간	2005-05-13	2005-05-13	1시간	최인욱	100%	
콜데이터 작성	2005-05-13	2005-05-13	3.5시간	2005-05-13	2005-05-13	4시간	최인욱	100%	
콜데이터 리뷰	2005-05-13	2005-05-13	1.0시간	2005-05-13	2005-05-13	1시간	최인욱	100%	
홍표현식 작성	2005-05-13	2005-05-13	2.5시간	2005-05-13	2005-05-13	1시간	최인욱	100%	
홍표현식 단위테스트	2005-05-13	2005-05-13	0.5시간	2005-05-13	2005-05-13	0.5시간	최인욱	100%	
해약한금음 모듈 작성			-						전통형 상품이므로 불필요
툴 컴포넌트 수정	2005-05-16	2005-05-16	1.0시간	2005-05-16	2005-05-16	2시간	최인욱	100%	
DDS 맵핑	2005-05-16	2005-05-16	3.0시간	2005-05-16	2005-05-16	2.8시간	최인욱	100%	
개발서버 등록	2005-05-16	2005-05-16	0.5시간	2005-05-16	2005-05-16	0.5시간	최인욱	100%	
통합테스트			5.0시간						어린이건강과유사상품 (상품명만 수정) 불필요한 테스트
현업 검증	2005-06-03	2005-06-08	--	2005-06-03	2005-06-08	-	한혜진	100%	
운영체 반영	2005-06-08	2005-06-08	0.5시간	2005-06-08	2005-06-08	1.0시간	신무현	100%	
총계	최초시작일	최종완료일	총소요시간	최초시작일	최종완료일	총소요시간	담당자	계획대비달성률	
총소요시간	2005-05-02	2005-06-08	20	2005-05-13	2005-06-08	13.8			

<그림 11> 머큐리의 일정 관리 프로세스

달성 여부에 대해 나온 생산성 수치는 차후 개인의 업무 계획 수립에 많은 도움을 준다. 특히, 인터럽트 시간은 비효율성과 오류를 발생하여 업무의 흐름을 방해하기 때문에 개발 실무자는 자신의 인터럽트 시간을 인지하고 있는 것이 매우 중요하다. 따라서 PSP 에이전트는 개발 실무자가 자신의 업무 수행 능력 수준을 기준으로 시간 관리가 가능하게끔 팀원 각자의 이력 데이터 추적을 통한 시간 관리 환경을 지원하여 제 시간에 원하는 산출물을 획득하도록 한다. <그림 11>은 PSP 에이전트에 의해 개인 업무 시간에 대한 추적 및 관리를 지원받아 업무 효율성을 관리하고 있는 개발 실무자의 개인 일정 관리 프로세스이다.

머큐리는 PSP 에이전트가 제공하는 시간 로그를 추적하며 일정 관리를 할 수 있는 업무 환경을 만들어 준다. 이와 같은 환경을 토대로 개발 실무자는 업무 수행에 대한 방해 요소와 자신의 업무 패턴, 시간 활용 방식 등을 파악할 수 있다. <그림 12>는 개발 실무자의 업무와 관련 없는 인터럽트 시간을 체크한 화면이다.

중단 시간 로그

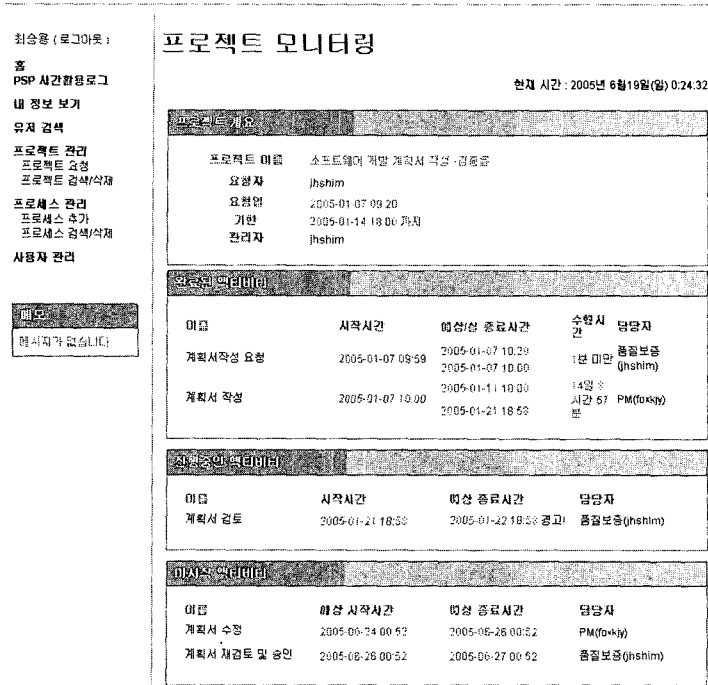
2004년2월17일 - 2004년2월25일 사이 소모된 모든 중단된 시간			
중단시작시간	중단종료시간	시간	비고
17:07	17:13	5분	인턴부
16:10	13:25	16:15	접당

위로

<그림 12> 머큐리의 인터럽트 시간 관리 화면

4.3 프로세스 모니터링

PSP 에이전트는 프로세스 진행 상황을 추적하여 Advisor 에이전트에게 프로세스 진척 상황 정보를 제공한다. Advisor 에이전트는 PSP 에이전트에서 제공되는 프로세스 진행 정보를 기반으로 개발 실무자에게 프로세스 수행 지침을 제공하게 된다. <그림 13>은 개발 실무자의 프로세스 진행 상황을 추적하여 본인에게 프로세스 진척 상황 정보를 보여주고 있다. 개발 실무자는 머큐리의 모니터링 정보를 근거로 현재의 프로세스 진행 위치와 문제점 발생 위치를 파악하여 업무 시간 조정 및 업무 순서 재배치를 하면서 정확한 프로세스 실행을 하게 된다.



<그림 13> 머큐리의 프로세스 모니터링

4.4 적용 평가

머큐리를 보험 상품 설계 시스템 개발을 위한 프로젝트 관리에 적용해 본 결과 총 보험 상품 25개 중에서 계획 시간보다 앞 당겨 개발이 완료된 상품이 20개로 나타났다<그림 14>. 보험 상품 25개에 대한 계획된 총 개발 시간은 363.5시간이었으나 개발에 소요된 시간은 328.2시간으로 35.3시간의 단축 효과를 보였다. 개발 실무자는 머큐리의 일정 관리 프로세스를 통해 생산성이 향상되어 계획 시간보다 개발을 앞당길 수 있었으며 개발사는 단축된 개발 시간으로 인해 개발 인력을 다른 프로젝트에 활용할 수 있어 효율적인 프로젝트 관리를 할 수 있었다. IT 업체가 지출하는 비용 중 인건비의 비중이 큰 만큼 개발 인력의 회전은 개발사의 비용 절감으로 나타났다. S 보험 회사는 단축된 개발 시간만큼 빨리 소비자에게 보험 서비스를

할 수 있어 수익 증가 효과를 보았다. 따라서 <그림 14>의 데이터를 통해 머큐리를 활용하면 프로세스의 통제성, 유연성, 민첩성, 신뢰성, 재사용성이 높아져 효율적인 프로젝트 관리가 가능함을 알 수 있다. 그러나 머큐리는 프로세스의 통제성, 유연성, 민첩성, 신뢰성, 재사용성의 정량적 측정에 대한 기술적 한계를 갖고 있다. 따라서 연구가 더 진행 되어 정량적 측정이 가능해지면 보다 체계적인 프로세스 관리 및 개선이 가능할 것으로 판단된다.

4.5 기대 효과

“프로세스는 스스로 정착되지 않는다.” 이는 프로세스를 정의한 후에는 반드시 의식적 노력이 수반되어야만 비로소 프로세스가 구현되고 조직의 ‘표준 실행 지침’으로 활용될 수 있다는 것을 의미한다. 조직으로부터 전략적 계획

1 Performance Data : 상용개정 및 등복

신용명	상용년/도	제척식	동기초요시간	차이시간	달성률	지연이유	비고
무배달상설연결보합(1.3)	개정로직(간단)	14시간	14시간	0 시간	100%		초과달성
무배달파워링크시연결 개정	개정로직(간단)	13시간	15시간	-2 시간	87 %	개발경험미숙	
무배달상설신번케어보합보합	개정로직(복잡)	24시간	19.5시간	4.5시간	123%		초과달성
무배달상설신번케어연결보합	개정로직(복잡)	22시간	14 시간	8시간	157%		초과달성
무배달상설변역연결보합	개정로직(복잡)	22.5시간	21.5시간	1시간	105%		초과달성
무배달상설변역연결보합_원가율	개정로직(복잡)	22.5시간	22.5시간	0시간	100%		초과달성
무배달상설저속보합	개정로직(간단)	14시간	8.5시간	5.5시간	165%		초과달성
월권연결 개정	개정로직(간단)	14시간	18시간	-4 시간	78 %	개발경험미숙	
연결저속상설카드연결_방카	개정로직(간단)	14시간	14시간	0 시간	100%		초과달성
연결저속상설카드연결_원원기일	개정로직(간단)	14시간	14시간	0 시간	100%		초과달성
연결저속상설카드연결	개정로직(간단)	14시간	9시간	5 시간	156 %		초과달성
무배달음주리호합(1.3)	개정요율(간단)	10시간	9.9시간	0.1시간	101%		초과달성
무배달유티슈의류보합보합(1.3)	개정요율(간단)	16시간	5.7시간	4.3시간	175%		초과달성
무배달브라보의류보합보합(1.3)	개정요율(간단)	10시간	6.6시간	3.4시간	152%		초과달성
무배달비추미일보합(1.4)	개정요율(간단)	10 시간	9.3 시간	0.7시간	108%		초과달성
무배달다미팩트상설보합	개정요율(간단)	8.5시간	8.5 시간	0 시간	100%		초과달성
무배달e-life상세보합(1.3)	개정요율(간단)	7시간	4.7시간	2.3시간	149%		초과달성
무배달월타입상세네타일상세보합	개정요율(간단)	10시간	9시간	1시간	111%		초과달성
무배달상설네타일상세보합	개정요율(간단)	10시간	10.6시간	-0.6시간	94%		미비한 지연으로 무시함
다미팩트월기_배정미일개정	개정요율(간단)	10시간	13시간	-3 시간	77%	DDS 변경으로인한 재작업	
분리월기_배정미일개정	개정요율(간단)	10시간	17시간	-7 시간	59%	DDS 변경으로인한 재작업	
다미팩트어원미일개정_신규	신규요율(간단)	20시간	13.8시간	6.2시간	145%		초과달성
어원미일개정_신규	신규요율(간단)	20시간	16.8시간	3.2시간	119%		초과달성
파워상세_신규	신규요율(간단)	20시간	18.8시간	1.2시간	106%		초과달성/통한데스트생략 바로그업결함
어원미일_신규	신규요율(간단)	20시간	14.5시간	5.5시간	138%		초과달성/통한데스트생략 바로그업결함

<그림 14> 머큐리를 활용한 프로젝트 관리 결과

이 세워지고 프로세스가 정의되어 결과적으로 조직에서 구현되기까지 프로세스 사용을 위한 제도화는 필수적으로 뒤따른다. 머큐리는 이를 위한 자동화된(intelligent) 환경을 제공하여 프로세스의 정의, 유지 및 관리, 개선을 위한 메타 프로세스(meta-process)에 대한 로드맵을 유연하게 설계할 수 있도록 해준다. 또한, CMMI, SPICE, 6시그마 등 관련 분야의 국제 표준을 조직 프로세스에 흡수하여 조정된 프로세스(tailored process)의 구축을 수월하게끔 지원하여 조직 문화의 전통을 훼손하지 않고 조직 혁신을 창출할 수 있는 기반을 제공한다. 따라서 머큐리 구축으로 국제 시장이 요구하는 표준을 조직 차원에서 관리하기가 용이해짐으로써 시장에서의 생존 확률을 높이는 효과를 기대할 수 있다.

또한, 머큐리는 가치사슬 전체에 걸쳐 전사적 프로세스 관리 및 개선을 지원하는 프로세

스 인지형(process-aware) 동적 플랫폼을 제공하기 때문에 머큐리를 구축한 기업은 다음과 같은 프로세스 경영이 가능해진다.

- 조직 전체 프로세스의 높은 가시성을 확보하게 되어 프로세스 관리 및 통제가 용이해진다.
- 조직 전체의 가치사슬을 측정, 분석하며 지속적으로 개선하고 최적화하는 능력을 갖추게 된다.
- 사후적 개선(reactive improvement)뿐만 아니라 선행적 개선(proactive improvement)도 가능한 프로세스 관리 및 개선 환경을 갖추게 된다.
- 프로세스 경영으로 인해 효율성, 민첩성, 투명성, 경쟁우위를 확보하게 된다.

5. 결론

본 논문에서는 프로세스 관리 및 개선의 방법으로 에이전트 기술과 프로세스 개선 방법론

인 6 시그마 프로세스를 적용하여 비즈니스 프로세스의 가시성 확보 및 자동화와 프로세스 중복 제거를 통해 프로세스의 통제성, 유연성, 민첩성, 신뢰성, 재사용성을 증가시켰다. 이에, 가치사슬 관점에서 조직에 최적화된 프로세스를 전개할 수 있게 했다. 또한, 프로세스 수행 활동에 대한 측정 도구로 PSP의 일정 관리 기법을 도입하고 이를 에이전트가 관리하도록 하여 개인 프로세스 역량을 측정하고 개선점을 식별하여 개인의 생산성이 향상되도록 했다. 향후, 비용 절감을 위한 측정 요소를 확장하여 일정과 통합된 프로세스 관리 모델 연구가 진행돼야 하고 프로세스의 통제성, 유연성, 민첩성, 신뢰성, 재사용성에 대한 정량적 측정 기술을 개발하여 통계 기반의 프로세스 관리와 개선 환경을 지원해야 한다. 또한, BRE, BPE를 통합하여 개선된 프로세스 관리 시스템을 구축 및 적용해야 한다.

참 고 문 헌

- [1] Majed Al-Mashari, et al, "Business Process Reengineering: a survey of international experience", BPMJ, Vol. 7, No. 5, pp. 437-455, 2001
- [2] Frank Leymann, Dieter Roller, "Business Process Management With FlowMark", Compton, pp. 230-234, 1994.
- [3] WfMC, "WfMC(Workflow Management Coalition) Standard Document", Technical Report, Workflow Management Coalition, November 1998.
- [4] Howard Smith, Peter Fingar, "Business Process Management: The Third Wave", Meghan-Kiffer, 2003.
- [5] Hammer M., "Reengineering the Corporation: A Manifesto for Business Revolution", HarperBusiness, January, 2004.
- [6] Mike Wooldridge, "Agent-based Software Engineering," IEEE Proceedings on Software Engineering, Vol. 144, No.1, pp 26-38, 1997
- [7] Humphrey, Watts S., "PSP: A Self-Improvement Process for Software Engineers", Addison-Wesley, 2005.
- [8] Mikel J. Harry, Richard Schroeder, "Six Sigma", Bantam Dell Pub Group, 1999.
- [9] Peter S. Pande, Robert P. Neuman, Roland R. Cavannagh, "The Six Sigma Way Team Fieldbook", McGraw-Hill, 2001.
- [10] N. R. Jennings, et al, "Agent Based Business Process Management", IJCIS, Vol. 5, No. 2/3, pp. 105-130. 1996.
- [11] Timothy J. Norman, et al, "Designing and Implementing a Multi-Agent Architecture for Business Process Management", Intelligent Agents III, pp. 261-276, 1997.
- [12] N. R. Jennings, et al, "Autonomous Agents For Business Process Management", Talyor & Francis, Vol. 14, No. 2, pp. 145-189. 2000.
- [13] Yuhong Yan, et al, "Integration of Workflow and Agent Technology for Business Process Management", CSCWD, The Sixth International Conference, pp. 420-426. 2001.
- [14] Humphrey, Watts S., "Introduction to the Personal Software Process", Addison-Wesley, 1999.
- [15] Humphrey, Watts S., "The Personal Software Process (PSP)", Technical Report CMU/SEI-2000-TR-022, 2000.
- [16] Matt Barney, Tom McCarty, "The New Six Sigma", Prentice Hall PTR, 2002.
- [17] Praveen Gupta, A. William Wiggenhorn,

- "Six Sigma Business Scorecard", McGraw-Hill, 2003.
- [18] Peter Fingar, Joseph Bellini, "The Real-Time Enterprise", Meghan-Kiffer, 2004.
- [19] Robert I. Carr, "Cost, Schedule, and Time Variances and Integration", Journal of Construction Engineering and Management, Vol. 119, No. 2, pp. 245-265, 1993.
- [20] David S. Christensen, "The Costs and Benefits of the Earned Value Management Process", Acquisition Review Quarterly, pp. 373-386, Fall 1998.
- [21] Giorgos Papavassiliou, et al, "Integrating Knowledge Modelling in Business Process Management", ECIS 2002, June 6-8, pp. 851-861. 2002.
- [22] Jim Tomayko, "Scientific Management Meets the Personal Software Process", IEEE Software, Vol. 20, No. 2, pp. 12-14, 2003.
- [23] Mala Murugappan, Gargi Keeni, "Blending CMM and Six Sigma to Meet Business Goals", IEEE Software, Vol. 20, No. 2, pp. 42-48, 2003.
- [24] Gary A. Gack, Kyle Robison, "Integrating Improvement Initiatives: Connecting Six Sigma for Software, CMMI, Personal Software(PSP), and Team Software Process(TSP)", SQP, Vol. 5, No. 4, pp. 5-13, 2003.
- [25] 신동일, 신동규, "워크플로우 관리 시스템의 설계 및 구현", 정보처리학회 논문지, 제7권, 제5호, pp. 1609-1618. 2000.
- [26] 송동현, 에이전트기반의 교수-학습 시스템 사례 및 활용에 관한 연구, 경기대학교 교육대학원 석사학위 논문, 2001
- [27] 신호준, e-비즈니스 응용 시스템 지원을 위한 에이전트 컴포넌트 개발에 관한 연구, 대구가톨릭대학교 대학원 박사학위 논문, 2003
- [28] 김계영, 국내기업의 지식경영 전략에 관한 연구, 원광대학교 경영대학원 석사학위논문, 2003
- [29] 고정욱, 지식경영의 조직 지식 저장 모형에 따른 성과 비교 분석: 에이전트 기반 모형을 이용하여, 한국과학기술원 석사학위 논문, 2004
- [30] 김정아, 황선명, 진영택, "룰 기반 컴포넌트 개발 기법 및 사례", 정보처리학회 논문지, 제12-D권, 제2호, pp. 275-282. 2005.

● 저 자 소개 ●



김 정 아 (JeongAh Kim)

1990년 중앙대학교 전자계산학과 졸업(석사)

1994년 중앙대학교 일반대학원 컴퓨터공학과 졸업(박사)

1996년 ~ 현재 관동대학교 컴퓨터교육학과 교수

관심분야 : Reuse, CBD, Product Line 공학, Ontology, 프로젝트 관리 및 프로세스 개선, etc.

E-mail : clara@kd.ac.kr



최 승 용 (SeungYong Choi)

2002년 관동대학교 컴퓨터교육과 졸업(학사)

2005년 관동대학교 일반대학원 정보통계학과 졸업(석사)

2006년 ~ 현재 성균관대학교 정보통신공학부 컴퓨터공학과 박사과정

관심분야 : CBD, Ontology, Agent, 프로젝트 관리 및 프로세스 개선 등, etc.

E-mail : boromi@gmail.com



배 제 민 (Jemin Bae)

1991년 중앙대학교 전자계산학과 공학사

1993년 중앙대학교 컴퓨터공학과 공학석사

1998년 중앙대학교 컴퓨터공학과 공학박사

1999년 ~ 현재 관동대학교 컴퓨터교육과 교수

2003년 ~ 현재 관동대학교 교육과학연구소 소장

관심분야 : 소프트웨어공학, 컴퓨터 교육

E-mail : gemini@kd.ac.kr