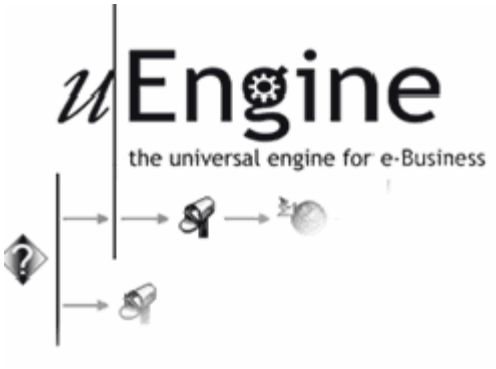

김보상, uEngine consultant
uEngine 오픈소스 프로젝트 2008-11-04

Process Monitoring Toolkit 사용자 가이드

ver 1.1

1. PMT 소개

1.1. PMT 이란?

Process Monitoring Toolkit의 약자이며, 업무 프로세스의 흐름을 프로세스 플로우차트 형식으로 출력하여 진행 상태을 보여주는 모니터링 도구입니다.

1.2. PMT의 도입 목적

Workflow 기능을 가지고 있는 제품이 아니더라도 ERP나 그룹웨어 시스템에서도 업무 흐름(프로세스)는 존재합니다. 하지만 업무 프로세스의 진행상황을 알고 싶을 때에 현재 사용하고 있는 시스템에서 프로세스의 모니터링 전용 화면이 없으면 각 건에 대해서 모두 조회를 해야 하거나 담당자에게 전화를 해서 보고를 받아야 합니다. 이런 경우에 PMT는 해당 프로세스의 업무진행 상태를 한눈에 볼 수 있도록 가시화 시켜줌으로써 업무의 효율과 프로세스의 정체된 단계를 쉽게 알 수가 있어서 비용 절감 효과를 얻을 수 있습니다.

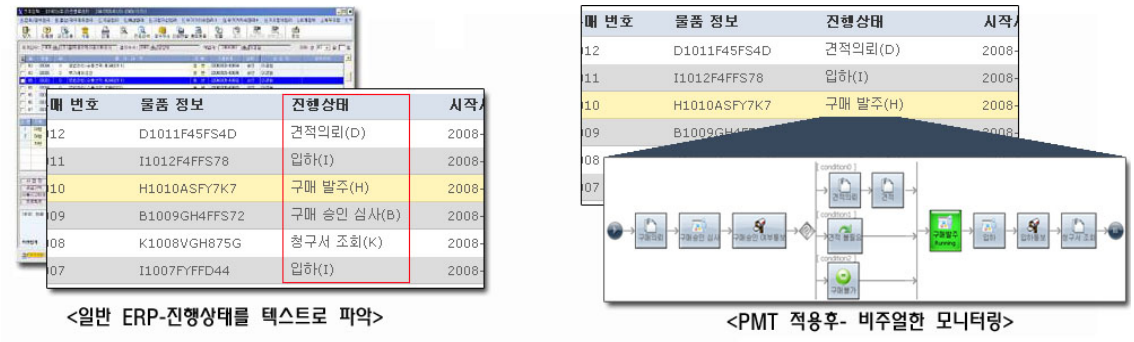

그림 1. 업무 모니터링 화면 비교

2. 프로세스 디자이너(Process Designer)

업무 프로세스를 그리기 위한 모델링툴로서 프로세스 디자이너를 이용하여 프로세스 정의를 만들게 되는데 완성 된 정의를 저장하면 XML형식의 프로세스 정의 파일(.XPD)이 생성됩니다. 그 정의 파일을 로드하여 플로우차트를 출력하게 됩니다.

2.1. 실행

프로세스 디자이너를 실행하기 위해서는 java JDK 1.6 을 포함한 상의 버전이 설치되어 있어야 하며, 윈도우 환경 변수에 JAVA_HOME을 설정해야 합니다.

그럼 준비가 끝났으면 프로세스 디자이너 아이콘을 클릭하여 **<그림 3>**과 같이 정상적으로 실행이 되는데 확인합니다.

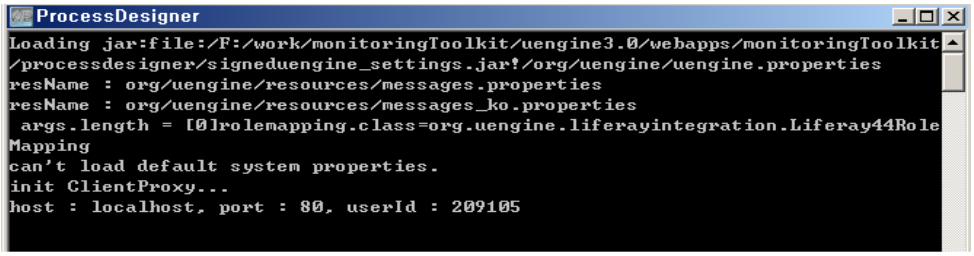

그림 2. 프로세스 다자이너 실행 콘솔

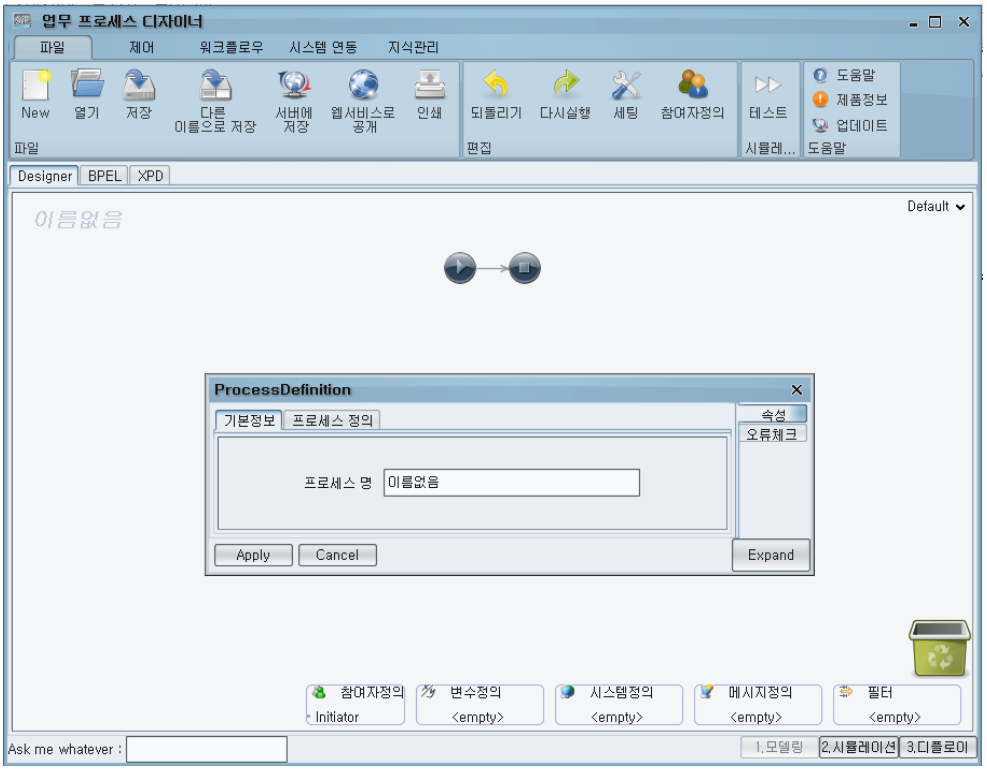

그림 3. 프로세스 디자이너 실행 화면

2.2. 액티비티(Activity)

프로세스 디자이너의 상단 영역에는 여러 종류의 Activity Type 들로 구성되어 있고 이 영역을 액티비티 타입 팔렛트라고 합니다. 이 중에서 문서 및 사람업무 액티비티 그룹에 소속된 [일반사람업무] Activity 를 클릭하거나 프로세스 플로우 차트 영역으로 Drag & Drop 하면

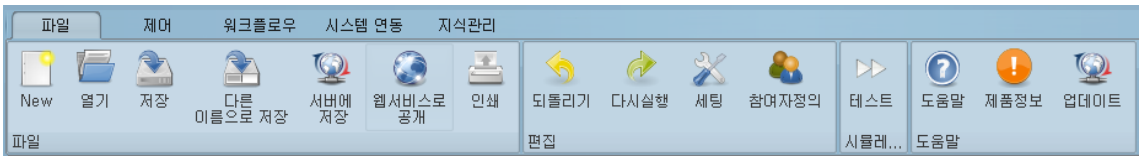

그림 4. 액티비티 팔렛트

<그림 5>와 같이 선택한 Activity가 프로세스 디자이너의 중앙의 프로세스 플로우차트영역에 추가됩니다.

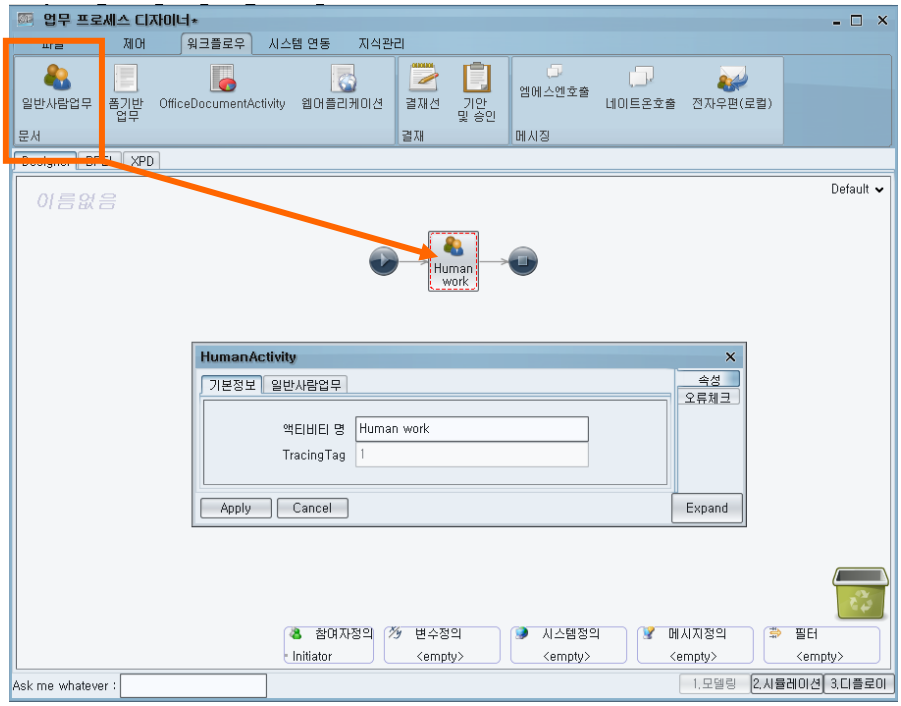

그림 5. 액티비티 추가

추가된 액티비티를 삭제할 때는 삭제하고자 하는 액티비티가 선택된 상태에서 [Delete] 키를 누르거나 오른쪽 마우스버튼을 클릭하여 삭제 할 수 있으며 오른쪽 하단에 있는 휴지통으로 Drag & Drop 해도 삭제가 가능합니다.

2.3. 액티비티 속성

선택된 액티비티에 속성을 설정하기 위해서 프로세스 플로우 차트에 추가된 일반사람업무 액티비티를 클릭하면 속성창이 활성화 되며 해당 액티비티 속성을 설정 할 수 있으며 액티비티 속성을 저장하려면 [Apply]를 클릭합니다.

액티비티명: 액티비티 이름이며 사용자 입력.
TracingTag: 액티비티 고유번호(숫자)로 자동으로 발급받으며 중복불가.
StatusCode: 액티비티 상태코드(문자열)이며 사용자 입력, 중복가능.

TracingTag는 액티비티의 ID라고 생각하시면 되고 액티비티별 지칭을 위해서 사용되고, StatusCode는 진행상태 전위를 위한 코드로서 병렬흐름과 같은 경우에는 동시에 진행이 이루어 지기 때문에 중복해서 사용할 수 있습니다.

3. 프로세스 플로우차트(Process Flowchart)

업무프로세스 진행 상태를 플로우차트 형식으로 화면에 보여주기 때문에 가시성이 높으며, 각 액티비티의 상세 정보를 풍선도움말 형식으로 출력해주고 업무프로세스가 한눈에 들어오기 때문에 업무프로세스의 빠른 이해를 돕습니다.

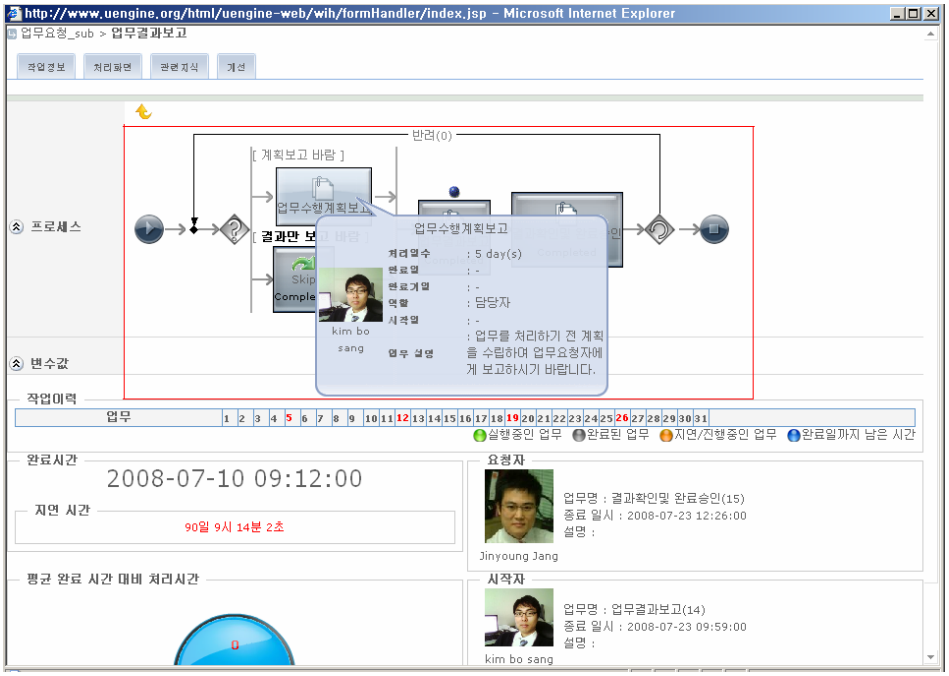

그림 6. 프로세스 플로우차트 적용화면

3.1. 기능

3.1.1. 비즈니스 상태

해당 단계(액티비티)의 비즈니스 상태를 출력문구를 입력할 수 있는 기능입니다.

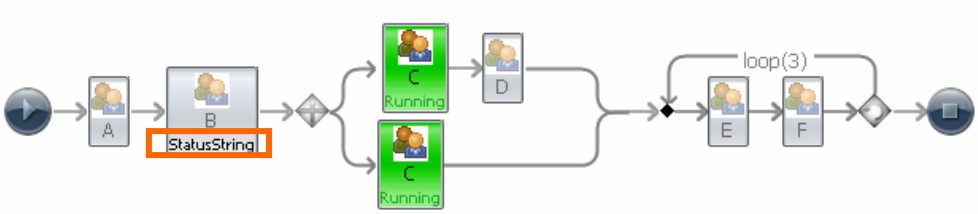

그림 7. 비즈니스 상태 출력

3.1.2. 풍선 도움말

해당 단계(액티비티)로 마우스를 이동시키면 풍선 도움말를 보여줍니다. 기본적인 레이아웃은 왼쪽에 담당자의 사진과 이름을 출력하며 오른쪽에는 액티비티의 상세 정보를 출력하게 되어 있습니다.

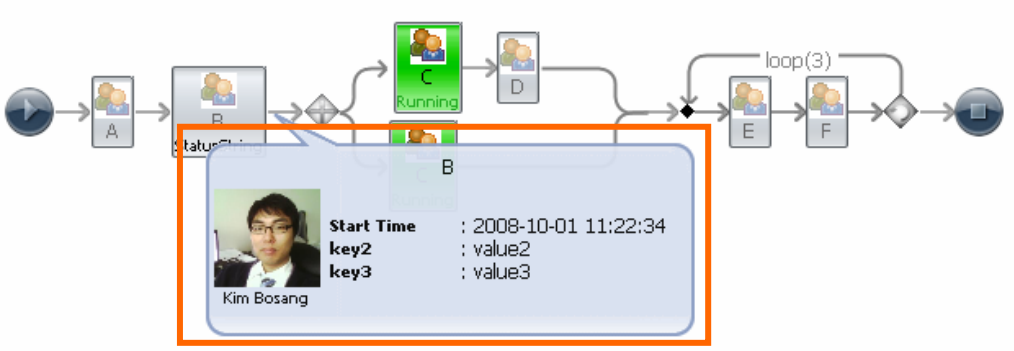

그림 8. 풍선도움말

3.1.3. 실행 단계

현재 실행 단계를 표시하는 기능입니다. 이 기능은 모델링 단계에서 입력한 상태코드(statusCode)를 이용하게 되며, 아래 **<그림 9>**에서는 각 액티비티 이름과 상태코드가 동일하게 설정되어 있으며 실행단계가 C단계에서 D단계로 바뀌는 것을 보여주고 있습니다.

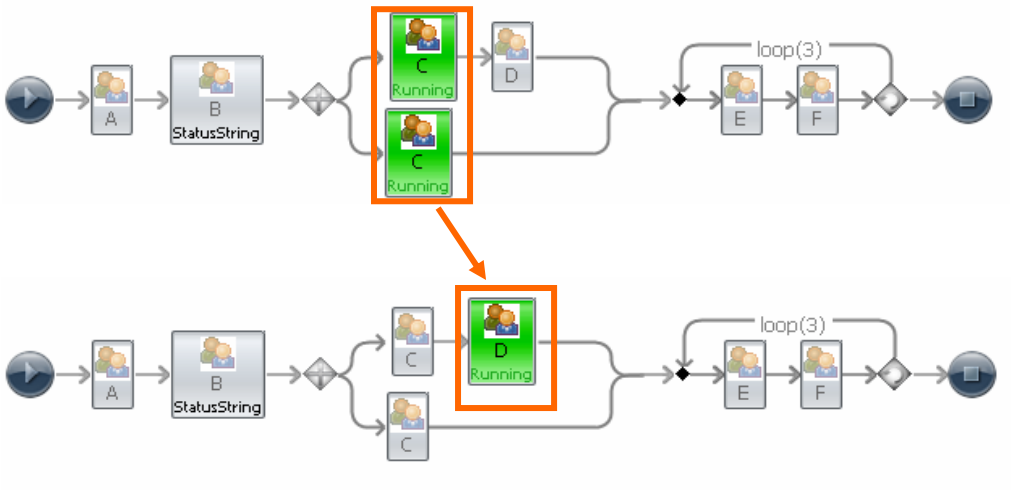

그림 9. 실행 단계 전위

3.1.4. 액션

액티비티를 클릭하였을 경우 스크립트를 실행할 수 있는 기능입니다. 아래 **<그림 10>**에서는 B액티비티를 클릭하였을 경우 'alert()' 스크립트를 실행하도록 한 것입니다.

그림 10. 액티비티 액션

3.1.5. 반복

프로세스 내에 반복적인 업무가 있을 경우에 **<그림 11>**와 같이 'loop(3)'과 같이 반복 횟수를 나타낼 수가 있으며 'loop'는 LoopActivity의 액티비티명입니다.

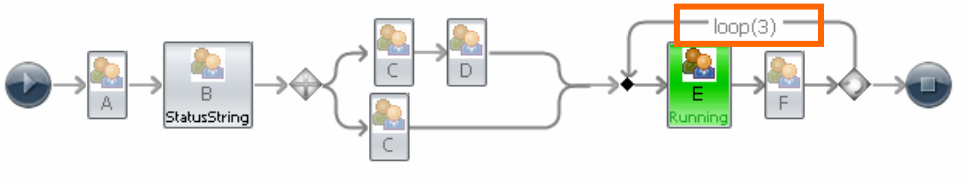

그림 11. 반복 횟수 표시

3.1.6. 세로 플로우차트

업무 프로세스가 긴 경우에는 가로 보다는 세로로 출력해서 모니터링을 하면 가시성을 높일 수가 있습니다.

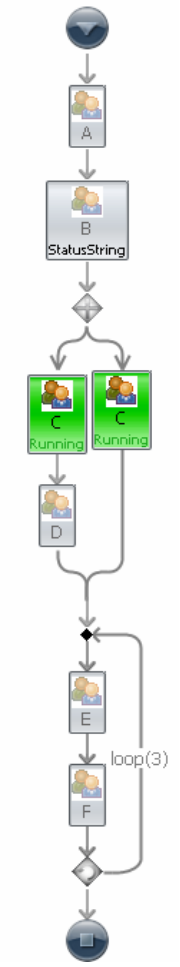

그림 12. 세로 플로우차트

4. 구현 방법

3 장에서 알아본 플로우차트 기능들을 샘플코드를 통하여 어떻게 구현을 해야하는지 알아보도록 하겠습니다.

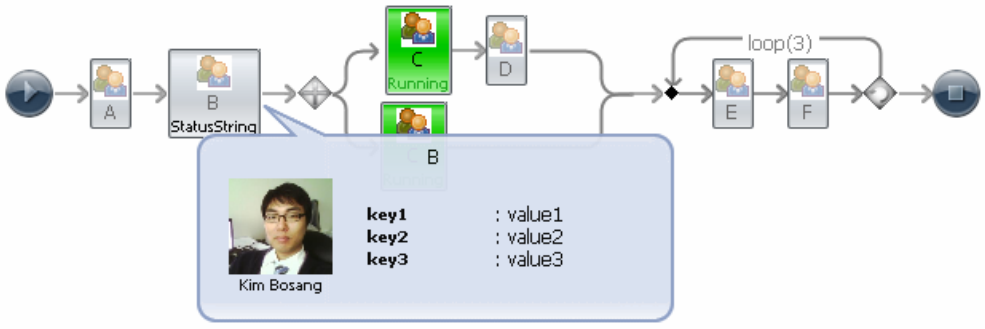

그림 13. 샘플 프로세스 플로우차트

4.1. 모델링

각 액티비티이름과 상태코드를 동일하게 설정(단계 표시를 쉽게 하기 위해서)하고 저장버튼을 클릭하여 프로세스 정의 파일을 특정 폴더에 저장을 합니다.

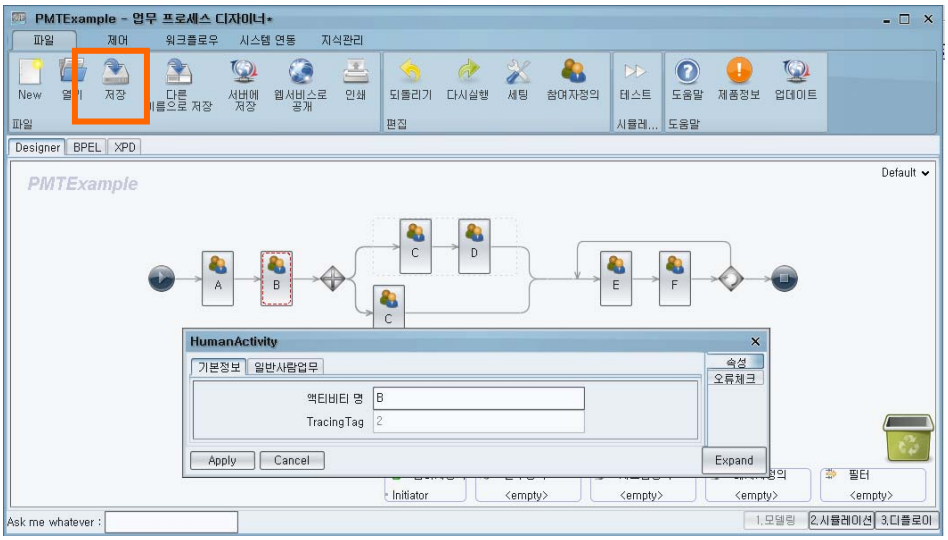

그림 14. 예제 프로세스 모델링

4.2. 샘플 소스코드(JSP)

```
<%@page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8" %>

<%@include file="common/dojoHeader.jsp"%>

<%@ page import="org.uengine.util.MonitoringOption"%>
<%@ page import="org.uengine.util.ProcessMonitoringToolkit"%>
<%@ page import="java.util.*"%>

<%
        String html="";
        String flowchart="";
        String statusCode="C";   //StatusCode가 'C'단계를 실행단계로 설정.
        String filePath = request.getRealPath("XPD/PMTExample.XPD"); //프로세스 정의 파일(XPD) 경로

         // 플로우차트 기본 스타일 옵션 설정
        Hashtable options = new Hashtable();
        options.put("decorated", "yes");   // 액티비티 배경이미지 유무
        options.put("nowrap", "yes");      // 액티비티명 내려쓰고 방지
```

```
		options.put("locale", "ko");		// 언어 설정, 영어는 'en'
		options.put("horizontal", new Boolean(true));	// 가로 플로우 차트, 세로는 'vertical'
		options.put("web_context_root", "/monitoringToolkit"); // 웹 콘텍스트 루트

		try{
			MonitoringOption monitoringOption = new MonitoringOption(); //플로우 차트 옵션 클래스

			monitoringOption.setFlowchartOptions(options); // 기본 스타일 옵션 설정

			monitoringOption.setBusinessStatus("2","StatusString"); //비즈니스 상태 문구 입력

			monitoringOption.setPortraitURL("2","Kim Bosang", "images/portrait/kbs.JPG"); // 담당자 정보

			monitoringOption.addActivityDetail("2", "key1", "value1"); // 액티비티 상세정보
			monitoringOption.addActivityDetail("2", "key2", "value2");
			monitoringOption.addActivityDetail("2", "key3", "value3");

			monitoringOption.setOnClickAction("2","alert('- B Activity -');"); //액티비티 액션 스크립트

			monitoringOption.setStatusCode(statusCode); //상태코드 중복 입력 가능

			monitoringOption.setRepeatCount("8","2"); // 반복 횟수 표시

			// 플로우 차트 HTML소스 얻기
			html = ProcessMonitoringToolkit.createFlowChart(filePath,monitoringOption);
		}catch(Exception e){
			throw new Exception(e);
		}
%>

<html>
<head>
<LINK href="style/uengine.css" type=text/css rel=stylesheet>
</head>
<body onload="enableTooltips();drawLoopLines(); " > // onload시에 풍선도움말과 라인 출력
<%=html%>
```

```
</body>
</html>
```

#MonitoringOption class

```
public void setFlowchartOptions(Map flowchartOptions)
public void setPortraitURL(String tracingTag,String userName, String portraitURLString)
public void setBusinessStatus(String tracingTag, String status)
public void setOnClickAction(String tracingTag, String javascript)
public void addActivityDetail(String tracingTag, String key, String value)
public void setRepeatCount(String tracingTag, String countMessage)
```

위 샘플에서는 한 단계(tracingTag=2)에 한하여 옵션을 적용해 보았는데 이와 같은 방법으로 각 단계(액티비티)의 모니터링 옵션을 모두 설정을 해주면 플로우차트는 완성이 됩니다.