

다중 테넌트 지원을 위한 액티비티 기반의 워크플로우 설정 방법

정문영, 원희선, 허성진
한국전자통신연구원 소프트웨어연구부문 클라우드컴퓨팅연구부 SW 서비스연구팀
e-mail : {mchung, hswon, sjheo}@etri.re.kr

A Method for Workflow Configuration based on Activity

Moonyoung Chung, Hee Sun Won, and Sung Jin Hur
SW Service Research Team, ETRI

요 약

본 논문에서는 다중 테넌트를 지원하기 위한 액티비티 기반의 워크플로우 설정 방법을 제안한다. 이를 위해서, 다중 테넌트가 하나의 워크플로우 인스턴스를 사용할 때, 액티비티 기반으로 워크플로우를 테넌트에 맞게 설정하는 방법과, 워크플로우 액티비티를 별도의 메타데이터로 관리하는 방법을 제공한다.

1. 서론

신뢰성 있는 대용량 인터넷 액세스 기술의 발전과 서비스 지향 아키텍처 기술, 패키지 소프트웨어의 유지보수 비용이 높아짐에 따라, SaaS 라는 새로운 방식의 웹기반 서비스가 빠르게 성장하고 있다. 기존에는 같은 서비스를 여러 테넌트(기업)의 사용자에게 제공하는 데에 많은 비용을 들이고 있는데, 이를 줄이기 위해서 SaaS 환경에서는 하나의 서비스를 웹기반 다중 테넌트 지원 방식으로 제공하고 있다. 또한, 기존에는 한 테넌트에게 서비스를 제공하기 위해서는 소프트웨어, 하드웨어 등을 포함하여 전용 리소스를 그 테넌트에 할당하였으나, SaaS 환경에서는 하나의 소프트웨어 인스턴스만 관리하며 여러 테넌트가 이를 공유하여 사용할 수 있게 하여 관리 비용 감소, 리소스 구축 비용 감소 등 막대한 경제적인 효과를 얻을 수 있다.

다중 테넌트가 하나의 소프트웨어 인스턴스를 공유해서 사용하면서도 마치 한 그룹당 할당된 리소스를 사용하는 것처럼 제공하기 위해서 많은 기술들이 제안되고 있는데, 그 예로 다중 테넌트 지원 메타데이터 기술, 워크플로우 승인 기술 등이 있다. 그 중 하나의 소프트웨어 인스턴스를 공유하면서도 각 테넌트가 그 테넌트에 할당되고 컨피규레이션된 고유의 서비스를 사용하는 것처럼 서비스를 제공하는 기술이 매우 중요하여 다중 테넌트 메타데이터 연구에서 스키마 매핑, 공유 스키마 등 다수의 기술이 제안되고 있다.

그러나, 다중 테넌트 지원 워크플로우 관련 기술에서는 워크플로우 승인 기술 등 제한된 영역에서만 컨피규레이션 방법이 제공되고 있다. 워크플로우를 지원하는 기존의 SaaS 환경에서는 단순히 다중 테넌트가 워크플로우 엔진을 온라인 방식으로 이용할 수 있도록 하고 있으며, 정의된 하나의 워크플로우 인스턴스를 설정하여 사용할 수 있는 기능을 지원하지 않는다. 따라서, 정의된 워크플로우를 그대로 사용하거나, 새로운 워크플로우를 정의하여 사용하여야 한다. 이 경우, 하나의 워크플로우 인스턴스를 다중 테넌트에게 제공하는 것에 비해 리소스 공유나 리소스 관리 측면에서 고비용, 저효율이 될 수밖에 없다.

본 논문에서는 위와 같은 기존 기술의 문제점을 해결하기 위해서, 다중 테넌트 지원 환경에서 워크플로우를 테넌트 환경에 맞게 설정하는 방법을 제안한다. 이를 위해서, 다중 테넌트가 하나의 워크플로우 인스턴스를 사용할 때, 액티비티 기반으로 워크플로우를 테넌트에 맞게 설정하는 방법과, 워크플로우 액티비티를 별도의 메타데이터로 관리하는 방법을 제공한다.

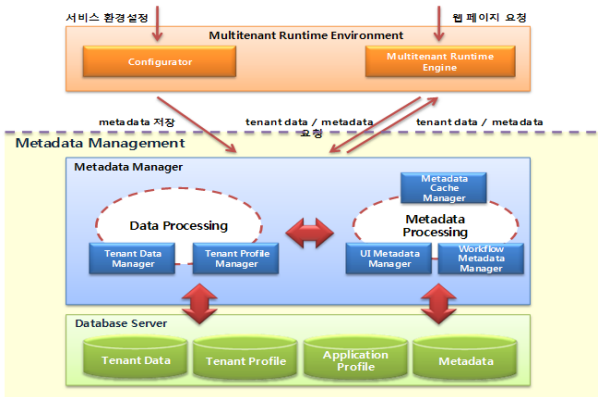
2. SaaSpia 플랫폼과 워크플로우 설정

SaaSpia 플랫폼은 다양한 응용 프로그램을 온라인으로 서비스할 수 있는 SaaS 플랫폼으로, SaaSpia 플랫폼에서 제공하는 응용 프로그램은 사용자가 설정 기능을 통해 직접 요구사항을 반영할 수 있는 고객 맞춤형 서비스로 제공된다. 이를 위해서, SaaSpia 플랫폼은 응용 프로그램을 통해 사용하는 테넌트의 데이터와 응용 프로그램을 구성하는 UI 에 대한 정보를 메타데이터화 하여 관리한다.

테넌트의 사용자가 응용 프로그램을 사용하면서 특정 페이지를 요청하면, 그 테넌트의 접근 권한을 확인하고 요청한 페이지에 대한 메타 데이터 정보를 통

¹ 본 연구는 지식경제부 및 정보통신연구진흥원의 IT 산업원천기술개발사업의 일환으로 수행하였음. [2009-S-033-01, 중소기업 SW 서비스를 위한 SaaS 플랫폼 개발]

해 필요한 UI 를 동적으로 구성하고, 스키마와 데이터를 읽어오게 된다. 반대로, 특정 페이지의 UI 구성이나 스키마를 설정하면, 그 테넌트의 메타데이터로 저장된다.

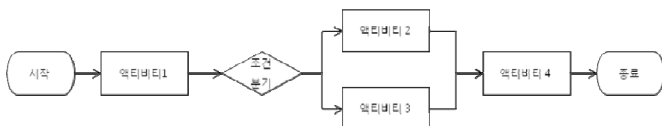


(그림 1) 다중 테넌트 메타데이터 관리 시스템 구조

워크플로우 설정 및 실행도 메타데이터를 통해 이루어진다. 응용 프로그램에서 제공하는 워크플로우를 설정하면, 설정 정보들은 그 테넌트의 메타데이터로 저장되며, 워크플로우 실행 시 메타데이터를 통해 설정된 워크플로우를 동적으로 생성하여 실행하게 된다.

3. 액티비티 기반 워크플로우 설정

이 장에서는 논문에서 제안하는 액티비티 기반의 워크플로우 설정 방법을 설명한다. 아래 그림 2 는 일반적으로 실행될 수 있는 워크플로우의 예를 보여준다. 워크플로우는 다양한 액티비티들로 구성되며, 일반적으로 액티비티는 제어, 할당, 시스템 연동, 반복 등의 다양한 타입으로 정의할 수 있다.



(그림 2) 워크플로우의 예

본 논문에서는 테넌트가 워크플로우를 설정할 때, 액티비티 단위로 설정하는 방법을 제안한다. 기존의 워크플로우를 설정하는 방법에서는 원래의 워크플로우를 사용자의 요구 사항에 맞게 설정하여 변경된 워크플로우 전체를 저장하여 사용하기 때문에, 사용자의 수에 따라 워크플로우의 복사본이 생성된다. 이 경우, 워크플로우 개발자가 일정 액티비티를 수정하려고 할 때 모든 워크플로우 복사본에서 해당 액티비티를 수정해야 하는 어려움이 있다. 따라서, 본 논문에서는 테넌트는 액티비티 단위로 워크플로우를 설정하고 설정 정보만 메타데이터로 관리하는 방법을 제안하여 워크플로우의 복사를 막고, 유지 보수 비용을 줄일 수 있다.

그림 3 에서는 워크플로우가 XML 파일로 직렬화하여 저장되는 예를 보여준다. 그림에서 이탤릭체로

표시된 부분은 하나의 SQL 액티비티를 저장한 모습이다. 그림에서 보는 것과 같이 워크플로우는 액티비티 단위로 구분되어 저장되므로, 이를 런타임 시로 대치해서 실행할 때 특정 액티비티를 다른 액티비티로 대체해서 실행하는 것이 쉽다.

```
<ProcessDefinition>
  <serviceDefinitions/>
  .
  .
  <SQLActivity>
    <sqlStmt>
      select sum(A.amount)
      from (select FLOOR(NVL(field_b,0)) - FLOOR(NVL(field_c,0)) AS amount
            from table_1
            where field_a = 'tenant_A') A
    </sqlStmt>
    <connectionFactory class="DataSourceConnectionFactory">
      .
      .
    </connectionFactory>
    <parameters>
      <oxy.uengine.kernel.ParameterContext>
        <argument>
          <text>field_a</text>
          <located%xls/>
        </argument>
        <variable reference="/processVariableDescriptors/ProcessVariable[2]"/>
        <transformMapping/>
        </oxy.uengine.kernel.ParameterContext>
        <transformMapping/>
        </oxy.uengine.kernel.ParameterContext>
      </parameters>
    </SQLActivity>
    .
    .
  </ProcessDefinition>
```

(그림 3) 워크플로우의 저장 예

따라서, 테넌트의 사용자에게 액티비티 단위로 설정할 수 있는 설정 도구를 제공하여, 액티비티 단위의 설정이 되면 그 메타데이터는 그림 4 의 예와 같은 액티비티 메타데이터 테이블에 저장한다. 워크플로우와 액티비티는 각각 할당된 식별자를 가지고 있으며, 테넌트도 식별자를 가지고 있다. 테넌트가 워크플로우의 액티비티를 설정하면, 메타데이터 테이블에 해당 워크플로우, 액티비티, 테넌트 식별자와 함께, 변경된 액티비티에 대한 URI 를 저장한다. 후에 사용자가 워크플로우를 실행하면, 원래의 워크플로우와 사용자가 속한 테넌트의 메타데이터 정보를 이용하여 동적으로 테넌트에 맞춤화 된 워크플로우를 구성하여 실행시켜 준다.

activity metadata table

workflow_id	activity_id	tenant_id	resource_uri

(그림 4) 워크플로우의 메타데이터 스키마의 예

4. 결론

본 논문에서는 액티비티 기반으로 워크플로우를 설정하는 방법 및 테넌트별로 설정한 데이터를 메타데이터로 저장하고 관리하는 방법을 제안하였다. 이를 이용하며, 다중 테넌트가 저비용, 고효율로 하나의 워크플로우 인스턴스를 테넌트에 전용으로 할당된 것과 같이 사용할 수 있다.

참고문헌

[1] <http://www.uengine.org>
 [2] 구경이, 이상민, 원희선, 허성진, “SaaS피아 플랫폼에서의 메타데이터의 지원,” 제 34 회 한국정보처리학회 춘계학술대회 논문집 제 17 권 2 호, 2010.11.
 [3] 정문영, 양경아, 구경이, 원희선, 허성진, “프로세스 기반 원산지관리시스템”, 제 34 회 한국정보처리학회 춘계학술대회 논문집 제 17 권 2 호, 2010.11.