테스트 프로세스 tailoring을 위한 방안

윤회진*, 최병주
이화여자대학교 컴퓨터학과

Tailoring Technique for the Test Process

Hojin Yoon, Byoungiu Choi
Computer Science & Engineering, Ewha Womans University,

요 약

표준에 정의된 내용을 가지고, 각 프로세스와 도메인에서 사용할 수 있는 수준의 프로세스를 구축 하는일은 쉽지 않다. 표준에 근거하여, 어떤 부분을 어떻게 tailoring하여 하는지에 대한 기준이 없기 때문이다. 따라서 표준은 표준대로 본문을 작성하더라도 실제 프로세스를 제공하려는 경우에, 표준과 실제 프로세스의 경로가 적절함을 확인하는 것이 필요하다. 따라서 표준을 각 프로세스에서 이용하기 위해서는 개발 도메인에 맞도록 tailoring이 요구 된다.

그러나 실제 도표에 정의된 내용을 가지고, 각 프로세스와 도메인에서 사용할 수 있는 수준의 프로세스를 tailoring하는 것은 쉽지 않다. 표준에 근거하여, 어떤 부분을 어떻게 tailoring하든지 하는지에 대한 기준이 구체적이지 않기 때문이다. 따라서 표준은 표준대로 본문을 작성할 뿐, 실제 프로세스의 경로에 게시물 사용하기 어렵다.

본 논문에서는 표준에 대한 tailoring 이檢查의 결과를 제출하기 위해, 콤포넌트 묶음이 구성하는 콤포넌트를 구성하고 composite하려는 작업을 하는 콤포넌트를 구축하는 과정에, 표준을 tailoring하여 특정 도메인에 맞는 프로세스를 구축하는 본 논문의 목적과 일치한다. 따라서 본 장에서는 콤포넌트 기반 개발에 대해 기술한다.

1. 콤포넌트 기반 개발

프로세스 표준들은 실제 개발 도메인에 맞도록 tailoring하기 위해, 본 논문은 콤포넌트 개념을 이용한다. 이미 작성된 프로세스를 새로운 목적으로 customize하고 composite하여, 새로운 기능을 하는 소프트웨어로 구축하는 과정이, 표준을 tailoring하여 특정 도메인에 맞는 프로세스를 구축하는 본 논문의 목격과 일치한다. 따라서 본 장에서는 콤포넌트 기반 개발에 대해 기술한다.

2. 콤포넌트 기반 개발

기본 단위가 되는 콤포넌트에 대한 정의는 다음과 같다. 이러한 다양한 정의들이 다음을 공통적으로 만드는 노드의 요소들로 구성한다. 콤포넌트는 요소들 간에 자율적으로 결합될 수 있도록, 인터페이스에 대한 명시적으로 정의된 요소를 갖는다. 콤포넌트 그 자체를 정의하지 않고 다른 콤포넌트들와 composition될 수 있도록, 그의 특성을 customize할 수 있다. 소프트웨어를 통하고, 인터페이스를 통해 콤포넌트를 다양하게 다룰 수 있다. 따라서 콤포넌트는 표1과 같은 요소들로 구성된 하나의 파키지라고 볼 수 있다[4].


1. 콤포넌트 기반 개발

프로세스 표준들은 실제 개발 도메인에 맞도록 tailoring하기 위해, 본 논문은 콤포넌트 개념을 이용한다. 이미 작성된 프로세스를 새로운 목적으로 customize하고 composite하여, 새로운 기능을 하는 소프트웨어를 구축하는 과정이, 표준을 tailoring하여 특정 도메인에 맞는 프로세스를 구축하는 본 논문의 목적과 일치한다. 따라서 본 장에서는 콤포넌트 기반 개발에 대해 기술한다.

2. 콤포넌트 기반 개발

기본 단위가 되는 콤포넌트에 대한 정의는 다음과 같다. 이러한 다양한 정의들이 다음을 공통적으로 만드는 노드의 요소들로 구성한다. 콤포넌트는 요소들 간에 자율적으로 결합될 수 있도록, 인터페이스에 대한 명시적으로 정의된 요소를 갖는다. 콤포넌트 그 자체를 정의하지 않고 다른 콤포넌트들와 composition될 수 있도록, 그의 특성을 customize할 수 있다. 소프트웨어를 통하고, 인터페이스를 통해 콤포넌트를 다양하게 다룰 수 있다. 따라서 콤포넌트는 표1과 같은 요소들로 구성된 하나의 파키지라고 볼 수 있다[4].

608
CBD는 크게 3개의 view를 갖는데[5]. 이미 만들어진 프로세스를 이용하여 새로운 소프트웨어를 구성하는 integrator관점, 사용자들이 생성할 수 있는 프로세스를 만드는 vendor관점 그리고 vendor가 만들어낸 프로세스를 integrator가 효과적으로 이용할 수 있도록 도와주는 기능을 하는 broker를 그룹으로 한다. 이 가운데 특히 integrator의 경우에서 CBD를 보기 어려우며, 완벽한 CBD가 되려면 이 세 가지 모두 고려되어야 한다.


3. 테스트 프로세스 콤포넌트 개발


자세화 모델링 결과, 테스트 프로세스의 활동은 기존의 작업은 캐레스로, 캐레스는 입의어로서 변환되며, 그 외에도 해당 활동의 대응이 있는 입력이거나 시동을 클레어로 존재한다. 각 절차를 수행하기 위한 기업들은 오버에이션 내에 위치하고, 특정 도메인을 위한 기업들은 클래식로 존재하여 customization을 통해 프로세스 콤포넌트를 개발한다. 테스트 프로세스를 이루는 시동과 활동들을 클래식으로 표현하여 클래식으로 특성을 구현하고, 클래식과 오버에이션의 상호 순서는 순서로 표현한다. 클래식에 드래그아웃 사전 순서에서 낮은 클래식과 높은 순열을 갖는 영역을 쌓아 하나의 프로세스로 정의한다. 이로 정의되는 것은 기본적인 소프트웨어의 콤포넌트를 `자세화 프로세스'라고 할 수 있다. 프로세스 콤포넌트는 표1의 제가지 구성요소를 표 2와 같은 내용으로 specialize하여 정의한다.

표 1 프로세스 콤포넌트의 구성요소

<table>
<thead>
<tr>
<th>프로세스 콤포넌트 구성요소</th>
<th>프로세스 콤포넌트 구성요소</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provided interfaces</td>
<td>프로세스의 제공자에 서비스를 제공하는 작업</td>
</tr>
<tr>
<td>Required interfaces</td>
<td>프로세스의 사용자에 서비스를 제공하는 작업</td>
</tr>
<tr>
<td>External specification</td>
<td>프로세스의 제공자가 서비스의 제공에 필요한 작업</td>
</tr>
<tr>
<td>Executable code</td>
<td>프로세스의 실행할 수 있는 코드</td>
</tr>
<tr>
<td>Design</td>
<td>프로세스의 구조</td>
</tr>
</tbody>
</table>

이로 프로세스 콤포넌트의 소스코드는 순서로 따라 콤포넌트를

이주는 클러스터의 오픈이더이션들에 수행될 수 있도록 구성되어 있는 프로세스였으며, 그에 대한 실행되서는 소스코드 프로세스를 제공하여 사용자가 직접 수행할 수 있도록 자연스럽게 기술된 데모 프로세스였으며, 이는 단순한 테스트 프로세스의 자동화를 위해 실제 기계에서 일하기도 기계와 기계의 모델을 전부 하는 몇가지 방법들을 설명한다.

그림 1 'Integration/TestDesign' 프로세스 콤포넌트

프로세스 콤포넌트는 특정에 따라 그림 2의 캐레스 세 가지 영역들을 갖는다. 표준에 따라 반드시 수행되어야 하는 부분으로 구성이 가능한 방법론의 영역과, 특정 도메인을 위한 기업의 설정이나 도메인에 요구하는 특정 작업들을 정의할 `클러스터' 영역, 그리고 다른 프로세스를 콤포넌트 또는 클러스터 서비스를 포함하여 하는 부분인 '인프라'의 영역이다.

그림 2 프로세스 콤포넌트의 구조

위에서 정의한 7개의 프로세스 콤포넌트는 표준에 따라 테스트 프로세스에서 수록한 프로세스 콤포넌트로서, 이들은 다양한 모데니의
특성을 표현한 플러그인을 갖는 플러그인은 프로세스 콤포넌트의 정의에서 도메인의 특성에 따라 수행할 수 있는 구체적인 기법과 수행의 수준을 달리고 있다. 플러그인은 콤포넌트에 대해 도메인의 특성을 반영하도록 customize한다. 본 논문에서 정의한 프로세스 콤포넌트들은 표 3과 같은 특성을 갖는 플러그인들로 구성된다.

표 3. 프로세스 콤포넌트의 plug-in

| 프로세스 콤포넌트 | 특성
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitTestDesign</td>
<td>Component plug-in</td>
</tr>
<tr>
<td>IntegrationTestDesign</td>
<td>Component plug-in</td>
</tr>
<tr>
<td>TestModeling</td>
<td>Component plug-in</td>
</tr>
</tbody>
</table>

4. 테스트 프로세스 콤포넌트 적용

3장에서는 CB의 vendor 탐색에서 프로세스 콤포넌트를 개발하였다. 프로세스 콤포넌트를 특정 환경에서 이용하기 위해서는 이미 개발된 콤포넌트들을 CB의 integrator로 도메인에 맞춘 방법이 요구된다. 콤포넌트는 다음 방법에 customiztion과 composition에 있다.

그림 1의 ‘IntegrationTestDesign’ 프로세스 콤포넌트의 경우, 그림 3에서처럼 캡슐화된 클래스를 통해 테스트하는 특정 기법을 갖는 ‘InterClass’TestDesign’ 플러그인을 프로세스 콤포넌트의 인터페이스에 연관관계로 연결하거나(그림 3의 B), 인터페이스로 공개된 오브젝트와 연결하여 실행할 수 있다.

그림 3의 A는 CB에서 제안한 프로세스 콤포넌트의 인터페이스를 사용하기 위해서는 표준의 테스트 프로세스를 도메인의 특성에 적합한 테스트 프로세스로 tailoring할 수 있다. 따라서 사용자는 프로세스 콤포넌트의 인터페이스를 사용할 때에는 그러한 필요성을 고려하여야 하며, 의뢰자에서의 요구를 이해하고 크로스모델링의 방법에 대해서는 구체적으로 tailoring한 테스트 프로세스를 구축할 수 있다.

그림 3의 B는 CB에서 제안한 프로세스 콤포넌트의 인터페이스를 사용하기 위해서는 표준의 테스트 프로세스를 도메인의 특성에 적합한 테스트 프로세스로 tailoring할 수 있다. 따라서 사용자는 프로세스 콤포넌트의 인터페이스를 사용할 때에는 그러한 필요성을 고려하여야 하며, 의뢰자에서의 요구를 이해하고 크로스모델링의 방법에 대해서는 구체적으로 tailoring한 테스트 프로세스를 구축할 수 있다.

그림 3의 B는 CB에서 제안한 프로세스 콤포넌트의 인터페이스를 사용하기 위해서는 표준의 테스트 프로세스를 도메인의 특성에 적합한 테스트 프로세스로 tailoring할 수 있다. 따라서 사용자는 프로세스 콤포넌트의 인터페이스를 사용할 때에는 그러한 필요성을 고려하여야 하며, 의뢰자에서의 요구를 이해하고 크로스모델링의 방법에 대해서는 구체적으로 tailoring한 테스트 프로세스를 구축할 수 있다.

그림 3의 B는 CB에서 제안한 프로세스 콤포넌트의 인터페이스를 사용하기 위해서는 표준의 테스트 프로세스를 도메인의 특성에 적합한 테스트 프로세스로 tailoring할 수 있다. 따라서 사용자는 프로세스 콤포넌트의 인터페이스를 사용할 때에는 그러한 필요성을 고려하여야 하며, 의뢰자에서의 요구를 이해하고 크로스모델링의 방법에 대해서는 구체적으로 tailoring한 테스트 프로세스를 구축할 수 있다.

그림 3의 B는 CB에서 제안한 프로세스 콤포넌트의 인터페이스를 사용하기 위해서는 표준의 테스트 프로세스를 도메인의 특성에 적합한 테스트 프로세스로 tailoring할 수 있다. 따라서 사용자는 프로세스 콤포넌트의 인터페이스를 사용할 때에는 그러한 필요성을 고려하여야 하며, 의뢰자에서의 요구를 이해하고 크로스모델링의 방법에 대해서는 구체적으로 tailoring한 테스트 프로세스를 구축할 수 있다.

그림 3의 B는 CB에서 제안한 프로세스 콤포넌트의 인터페이스를 사용하기 위해서는 표준의 테스트 프로세스를 도메인의 특성에 적합한 테스트 프로세스로 tailoring할 수 있다. 따라서 사용자는 프로세스 콤포넌트의 인터쁘니스를 사용할 때에는 그러한 필요성을 고려하여야 하며, 의뢰자에서의 요구를 이해하고 크로스모델링의 방법에 대해서는 구체적으로 tailoring한 테스트 프로세스를 구축할 수 있다.

그림 3의 B는 CB에서 제안한 프로세스 콤포넌트의 인터페이스를 사용하기 위해서는 표준의 테스트 프로세스를 도메인의 특성에 적합한 테스트 프로세스로 tailoring할 수 있다. 따라서 사용자는 프로세스 콤포넌트의 인터페이스를 사용할 때에는 그러한 필요성을 고려하여야 하며, 의뢰자에서의 요구를 이해하고 크로스모델링의 방법에 대해서는 구체적으로 tailoring한 테스트 프로세스를 구축할 수 있다.