한국스마트미디어학회 & 한국전자거래학회
2016 추계학술대회
장소: 광주시청자미디어센터 / 호남대학교
일시: 2016. 10. 28 (금) ~ 29 (토)

PROCEEDINGS
논문집
<table>
<thead>
<tr>
<th>논문 발표순서 (Poster Session 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P35 283Page</strong> 제목: 라즈베리파이 기반 농업용 스마트 미러 설계</td>
</tr>
<tr>
<td>저자: 신상훈, 김찬중, 이성근(순천대)</td>
</tr>
<tr>
<td><strong>P36 286Page</strong> 제목: 아두이노를 이용한 알약 알림 시스템</td>
</tr>
<tr>
<td>저자: 김환희, 류건웅, 이대우, 이성근(순천대)</td>
</tr>
<tr>
<td><strong>P37 289Page</strong> 제목: 중소 의료기관 내 악성코드 탐지 특징과 관련 보안위협 현황 확인</td>
</tr>
<tr>
<td>저자: 홍승균, 문창희, 김준영, 정재호(구)에브리존-기술연구소</td>
</tr>
<tr>
<td><strong>P38 293Page</strong> 제목: 변형된 스테이너 시스템을 이용한 분산 저장 부호의 설계</td>
</tr>
<tr>
<td>저자: 카림하륜, 박호성(전남대)</td>
</tr>
<tr>
<td><strong>P39 295Page</strong> 제목: 이중 태양광 시스템의 모델 변환을 위한 메타모델에서 자동 트리 모델 생성</td>
</tr>
<tr>
<td>저자: 손현승, 김영철(홍익대)</td>
</tr>
<tr>
<td><strong>P40 299Page</strong> 제목: 빅데이터 시각화 방법과 사례분석</td>
</tr>
<tr>
<td>저자: 문희정(호남대)</td>
</tr>
<tr>
<td><strong>P41 302Page</strong> 제목: 360도 비디오 서비스 기술을 활용한 개인 적성검사기반 기상작업체험 서비스 플랫폼 구축</td>
</tr>
<tr>
<td>저자: 하태진, 김국정(비온사이노베이트), 조승원, 오충근(레몬소프트), 박정민(조선아공대)</td>
</tr>
<tr>
<td><strong>P042 305Page</strong> 제목: A Study of Hacking Identification and Measurement of the Effect</td>
</tr>
<tr>
<td>저자: Daehwan Ahn, Byungjoon Yoo(서울대)</td>
</tr>
<tr>
<td><strong>P43 308Page</strong> 제목: 학교의 학생운영을 위한 침실 시뮬레이터 구현 방법에 관한 연구</td>
</tr>
<tr>
<td>저자: 민병국, 정희자, 이현준(비온사이노베이트), 박정민(조선아공대), 김남호(호남대)</td>
</tr>
</tbody>
</table>
이중 태양광 시스템의 모델 변환을 위한 메타모델에서 자동 트리 모델 생성

손현승, 김영철
홍익대학교 소프트웨어공학연구실
e-mail: {son, bob}@selab.hongik.ac.kr

An Automatic Tree Model Generation from Metamodel for Model Transformation of Heterogenous Photovoltaic System

Hyun Seung Son, R. Young Chul Kim
SE Lab, Dept. of Computer Information Compuctions, Hongik University

요 약

기존의 다른 형의 언어 XML 스키마, 데이터베이스 스키마, EBNF 등과는 다르게, 메타모델은 모델을 정의할 수 있는 언어로써 객체지향 벡터나무를 사용한다. 객체지향 기법은 속성과 메서드가 결합된 클래스를 사용할 수 있도록 하여, 메서드 단위의 재사용보다 직접 오버라이딩과 오버로딩을 통한 확장이 용이하다. 그러나 설계의 상속 개념은 포트의 길이에 따라서, 특수 클래스로부터 상속받은 타입이 무엇인지, 또는 클래스간의 연관성 파악이 어려운데다, 모델 변환에서 메타모델의 구조를 파악하지 못할 경우 변환 규칙 작성이 어려워진다. 이를 위해, 메타모델을 트리 모델로 자동생성 방법을 제안한다. 이 트리 모델은 모델 변환에서 각 모델의 입력 요소가 되는 메타모델의 구조를 트리로 표현함으로써, 모델 변환 시 메타모델 정보의 누락 방지와 메타모델의 구조를 쉽게 파악하고 모델 변환 규칙 작성의 오류를 줄일 수 있다. 현재 이중 태양광 시스템에 적용하고자 한다.

1. 서론

MDA/MDD(Model Driven Architecture/Model Driven Development) [1-2]는 소프트웨어 개발에서 플랫폼 독립 모델과 종속 모델을 분리하여 독립 모델을 재사용하여 이 중 플랫폼을 개발할 수 있는 아이디어로 제공한다. MDA/MDD는 독립모델과 종속모델로 분리하고 두 모델 간의 차이를 모델 변환 기법으로 자동화한다. 그러므로 자동화 도구는 필수이다.


앞에서 살펴본 것과 같이, 모델 변환 기법에서 모델과 모델을 변환시킬 수 있는 규칙만 중요할 것 같지만 입력과 출력을 정의하는 메타모델도 중요하다.


그러나 설계 문서로 그러진 상속 개념은 노트의 질이가 길음을 수록 부모 클래스로부터 상속받은 타입이 무엇인지에 대한 클래스와 연관되어 있는 것에 따라가 어렵다. 또한 메타모델의 규모가 커질수록 그 복잡도는 점점 커진다.

메타모델 변환에서 메타모델의 구조를 파악하기 어려운 경우 변환 규칙 작성이 어렵다. 그 이유는 메타모델 구조를 동

* 이 논문은 2015년 교육부와 한국연구재단의 지역혁신창조인력양성사업의 지원을 받아 수행된 연구임(NRF-2015H1C1A035548)
해서 만 모델의 정보를 가져올 수 있기 때문이다. 이 문제를 해결하기 위해 본 논문에서는 메타모델을 트리 모델로 자동생성 하는 방법을 제안한다. 이 트리 모델은 모델 변환에서 각 모델의 입력 요소가 되는 메타모델의 구조를 트리로 표현함으로써, 모델 변환시 메타모델 정보가 누락되는 것을 방지하고 메타모델의 구조를 쉽게 파악할 수 있어 모델 변환 규칙 작성의 오류를 줄일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 모델 변환 기법에 대해 소개하고 3장에서는 트리 모델 자동 생성 방법을 언급한다. 4장에서는 제안한 내용을 실제 메타모델에 적용하고 5장에서는 결론 및 향후 연구에 대해 언급한다.

2. 관련 연구
모델로 모델(M2M, Model to Model) 변환은 모델을 입력하여 모델을 출력하는 방법으로 그림 1과 같은 과정으로 메타모델, 변환 규칙, 변환 엔진으로 구성된다[3]. 모델에서 모델 변환을 위해서는 입력과 출력의 양쪽 메타 모델이 정의되어 있어야한다. 여기서 메타모델은 모델정보를 읽어올 수 있는 스키마 역할을 한다. 다음으로 변환 규칙을 정의하는데, 이는 각 모델 요소에 대한 메타 요소로 어떻게 변형되어야 하는지 그 규칙을 기술한다. 즉, 모델의 추가, 삭제, 수정에 관한 내용이 있다. 모델 변환 엔진은 모델 변환 규칙을 실행해 입력 모델의 변환 규칙에 따라 출력 모델로 변환한다.

![그림 1] 모델로 모델로 변환의 메커니즘


3. 트리모델로 자동 생성 방법

![그림 2] 트리 모델 생성기의 구조

트리 모델 생성기는 그림 2와 같이 메타모델을 입력받고 트리 모델로 변환한다. EMF를 기반으로 메타모델을 처리하기 때문에 Ecore 메타모델을 사용하고 본 논문에서 제안한 트리 모델을 메타모델로 사용한다.

메타모델의 레이어는 그림 3과 같이 상속구조에 대한 모든 정보를 볼 수 있는 것이 아니라 평행적인 구조로 자기 자신의 상위 클래스만 확인할 수 있다. 그러므로 전체 구조를 파악하기 위해서는 일일이 추적해야 하는 설계 그림을 보면서 찾아야 한다.

![그림 3] ecore의 구조

트리 모델은 메타 모델의 평행적인 구조를 트리 구조로 변환한 것으로 트리 모델의 메타모델은 그림 4와 같다. TreeModel은 최상의 노드로 트리 모델 자체 이름을 저장할 수 있다. TreeModel의 서브 노드는 Node로 메타모델의 부모 노드인 parent와 자식 노드인 subNode를 가지고 있다. 그리고 Node에는 속성과 태그에 해당하는 Attribute와 Type을 가지고 있어 메타모델의 속성과 태그를 표현할 수 있다.

![그림 4] 트리 모델(TreeModel)의 메타모델

트리 모델의 표현은 아래와 같이 다중 관계를 표현하는 [0..*]와 메타모델의 이름, 태그, 속성을므로 이루어져 있다.
속성을 둘로 클래스로부터 상속받는 모든 속성이 표현되어 별도의 추가가 없어야도 관련된 속성을 한 번에 파악 가능하다.

그림 5의 메타모델을 TreeModelGenerator로 트리 모델을 생성하면, 그림 6과 같은 형태의 트리 모델을 만들 수 있다. 생성된 트리 모델을 살펴보면, Package가 루트 노드인 것을 메타모델 설계 모델을 찾아보지 않아도 쉽게 알 수 있다. 메타모델은 루트 노드를 루트에 표시하지 않으면 찾기 어렵다. 또한 Package 하부에 Class, PrimitiveDataType, Association이 있음 수 있다는 것을 트리 구조로 쉽게 파악할 수 있다.

1) 입력된 메타모델을 너비 우선 탐색을 수행
2) 입출력에 따른 서브 노드를 생성
3) 추상 클래스일 경우: 모든 상위 클래스의 속성과 참조 탐색을 가진 노드 생성
4) 그렇지 않은 경우: 현재 노드의 속성과 참조 탐색을 가진 노드 생성

5. 결론
메타모델은 모델을 정의할 수 있는 언어로 객체지향 기법을 사용한다. 객체지향 기법은 클래스 단위로 제작이 가능하고 재사용화가 깊이 포함된다. 그러나 본 논문에서는 이 문제를 중점적으로 해결하기 위해 메타모델을 트리 모델로 자동 생성 모델 제안한다. 이 트리 모델은 메타모델의 구조를 트리로 표현함으로써, 메타모델의 구조를 쉽게 파악할 수 있게 모델 변환시 누락되는 정보를 방지하고자 한다. 이론적游戏玩家에 적응하고자 트리 모델을 이용한 모델 변환 언어를 연구 중이고 이를 확산하고자 한다.

참고 문헌


