# Automatic Requirement Sentence Extractions from Byproducts based on Heterogeneous Issue Tracking models in Software Visualization Environment

Woo Sung Jang, Kidu Kim, So Young Moon, Young Soo Kim, R. Young Chul Kim

## Abstract

- We work on the test case generation method with requirement sentences, which tries to guarantee 100% coverage with minimal test cases.
- The problem requires manual input of requirement sentences. To solve this problem, we propose automatic requirement sentence extraction from byproducts on issue tracking in our visualization system.
- The practitioners easily manage the requirements using the Issue Tracking System in the software visualization environment. As a result, this method supports the automatic generation of test cases from requirements in the practical environment.

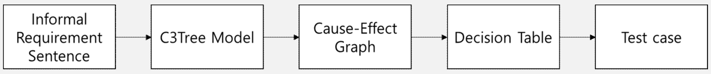
## Effect

- The requirement sentences are automatically extracted from heterogeneous ITS tools within a software process visualization environment.
- The requirement sentences in the ITS DB are automatically entered into the test case automatic generator.
- New ITS tool easily added through only metamodel and metamodel transformation rules add.
- The source code of the software is not modified.
- Maintenance of the software is increased.

## Related Studies

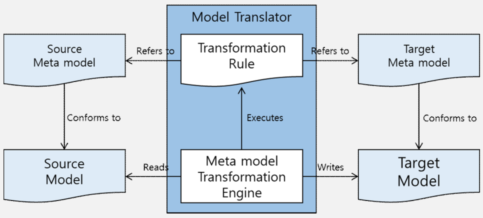
### 1. Test case automatic generation with informal requirement sentence

- Test cases generation method from model-based requirements
- Guaranteed 100% coverage with minimal test cases

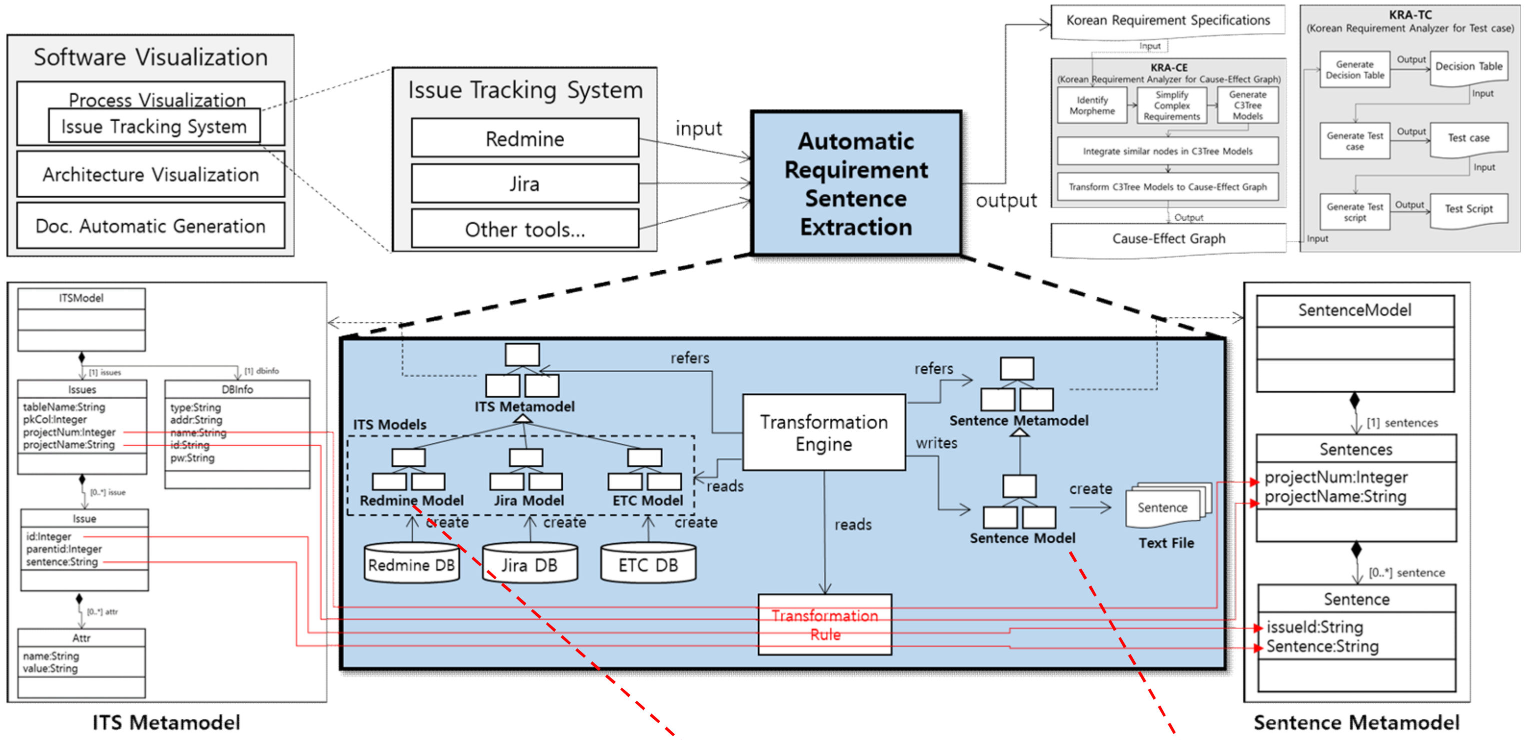


### 2. Model to Model Translation Mechanism

- Model automatic transformation engine based on metamodel
- Automatically transforms models generated based on metamodels using a model transformation language



## Automatic Requirement Sentence Extraction Method



## Process

1. ITS models (Redmine Model, Jira Model, etc.) are created heterogeneous ITS DBs with issue information.
2. The ITS Metamodel is designed from the meta information of each heterogeneous ITS model.
3. The Sentence Model is a model to store natural language sentence information.
4. The Sentence Metamodel is designed from the meta information of the Sentence Model.
5. A transformation engine automatically transforms ITS models (Redmine Model, Jira Model, etc.) based on ITS Metamodel into Sentence Model based on Sentence Metamodel using Transformation Rule.
6. Transformation Rule defines the rule to change each element of ITS Metamodel to an element of Sentence Metamodel.
7. Natural language sentences are extracted from the sentence model.

## Case Study

(a) XMI Code of Redmine Model

```
<itsmodel>
<dbinfo type="redmine" addr="…" name="redmine" id="root" pw="…"/>
<issues tablename="issue" pkcol="4" projectnum="1" projectname="project1">
  <issue id="1" parentid="" sentence="If a input then…" />
  <issue id="2" parentid="" sentence="If b input then…" />
</issues>
</itsmodel>
```

Automatic Transformation

(b) XMI Code of Sentence Model

```
<sentencemodel>
  <sentences>
    <sentence issueid="1" sentence="If a input then …"/>
    <sentence issueid="2" sentence="If b input then …"/>
</sentences>
</sentencemodel>
```