
Reference: Model Quality Assurance – AIME (QSE:CSEK)

Topic: Automatic Identification of Model Elements in Text

LVA-Type: Praktikum, Bakk-/Diplomarbeit

Start: Ab sofort

End: Nach Vereinbarung

Contact: Marta Sabou (marta.sabou@ifs.tuwien.ac.at)

Dietmar Winkler (dietmar.winkler@tuwien.ac.at)

Stefan Biffl (Stefan.Biffl@tuwien.ac.at)

Background

During the design of software systems, a variety of models are created in the process of transforming the requirements and/or specification of the desired system into the corresponding software. These models include Extended Entity Relationship (EER) diagrams or UML model variants for designing databases and software system structures and behavior. The tasks of creating such models from software specifications and their subsequent verification to ensure their quality, i.e., through software model inspection [1], are cognitively intense tasks, that require significant time and effort investment from software engineers.

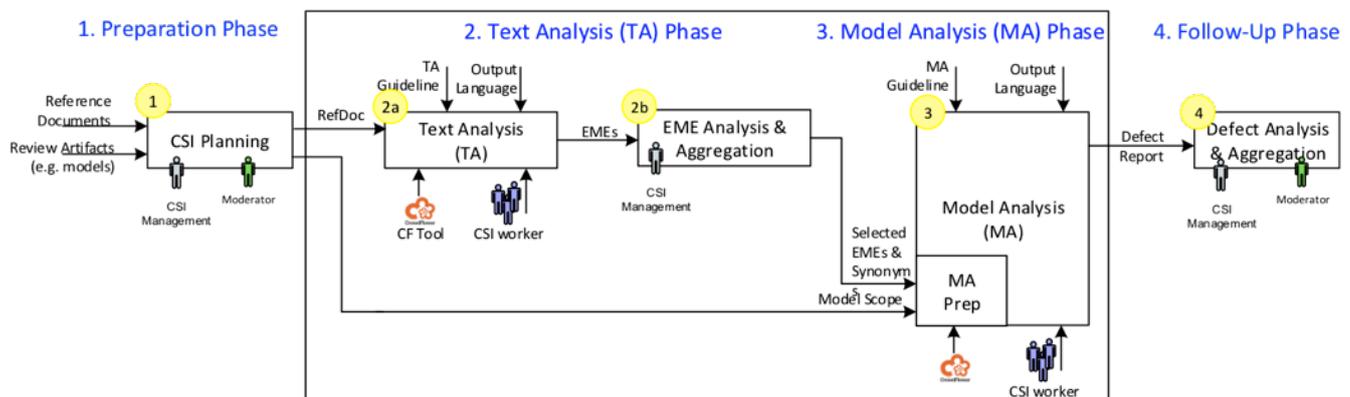


Figure 1: Crowdsourcing-based Software Inspection (CSI) process

To address this issue, we defined and introduced a novel Crowdsourcing-based Software Inspection (CSI) process, previously described in [1,2,3] and shown in Figure 1. A key element of the CSI process is the concept of Expected Model Element (EME). EMEs represent the main building blocks of software engineering models and include entities, attributes, and relationships, that are present in system requirements specifications (i.e., reference documents) and need to be modeled in the corresponding software model, e.g., in an EER diagram. The identification of the EMSs is currently achieved through crowdsourcing in the *Text Analysis* phase of the CSI process, however, this approach has several limitations in terms of being too time-consuming and resulting into EME suggestions that are challenging to combine.

Tasks

The key task is to provide methods for the semi-automatic identification of Model Elements in a textual system specification documentation. The work should

investigate the optimal mix of approaches based on natural language processing (NLP), crowdsourcing and expertsourcing. The GATE NLP infrastructure will be used to process the textual document and crowd-or expertsourcing will be implemented with the Crowdflower framework.

Experience and skills needed

The following preconditions are recommended, but could also be learned during the thesis itself.

- Java programming
- Crowdsourcing (CrowdFlower)
- Natural language processing (GATE framework)
- Software engineering
- Defect detection in software engineering models

References

[1] Winkler D., Sabou M., Petrovic S., Carneiro G., Kalinowski M., Biffi S.: Investigating Model Quality Assurance with a Distributed and Scalable Review Process, In: Proceedings of the 20th Ibero-American Conference on Software Engineering, Experimental Software Engineering (ESELAW) Track, Springer, Buenos Aires, Argentina, 2017.

[2] Winkler D., Sabou M., Petrovic S., Biffi S., Kalinowski M., Carneiro G.: Improving Model Inspection with Crowdsourcing, In: Proceedings of the 4th International Workshop on Crowdsourcing in Software Engineering, ACM/IEEE International Conference on Software Engineering (ICSE), Buenos Aires, Argentina, 2017.

[3] Winkler D., Sabou M., Petrovic S., Biffi S., Kalinowski M., Carneiro G.: Improving Model Inspection Processes with Crowdsourcing: Findings from a Controlled Experiment. In: Proceedings of the 24th European System, Software and Service Process Improvement and Innovation (EuroSPI), 2017