

Automatic generation of the use case model

Supervisor: Marinos Georgiades (marinos dot georgiadis at ut dot ee)

The main goal of this endeavor is to build a software tool that will automate the creation of use case diagrams and specifications from a set of predefined use case patterns. These patterns include specific use case types (e.g., *Create information object [IO]*, *Correct IO*, *Modify IO state*, *Cancel IO*, *Archive IO*, *Complete IO*, *Read IO*) and interconnections (through *include*, *extend* and *inheritance* relationships), at the context level, together with specific types and sequence of actions (e.g., *request*, *prompt*, *add*, *delete*, *choose*, *calculate*, *submit*, *validate*, *save*, *confirm*, and *notify*) for each use case type, at the content level. The patterns are described in a published paper [1], and additional relevant information will be provided by the first author of the paper, Marinos Georgiades.

The existing patterns already provide a use case formalization that helps in reducing the time to identify and specify requirements, in diminishing redundancies, inconsistencies and omissions, and, generally, in producing better requirements. Automating formalization will substantially improve speed and quality, and ease the human burden of the requirements engineering process.

The software tool to be developed will initially enable the user (e.g., software engineer, systems analyst) to specify the patterns and have them stored in an appropriate format, and, based on a number of rules, it will subsequently provide automatic creation of use case diagrams and specifications. The tool should also allow editing the use case model. Possible further or future implementation will involve the automation or semi-automation of (i) analyzing nouns and verbs found in the use case specifications and (ii) generating the analysis classes/model.

References

[1] Georgiades, M. and Andreou, A. 2013. Patterns for Use Case Context and Content. In *Proceedings of the 13th International Conference on Software Reuse (ICSR)*, (Pisa, Italy). Springer 2013 Lecture Notes in Computer Science, Volume 7925, 267-282.