

Identifying Duplicate Functionality in Textual Use Cases by Aligning Semantic Actions

(SoSyM Abstract)

Alejandro Rago^{*†1}, Claudia Marcos^{*‡2}, J. Andrés Díaz-Pace^{*†3}

^{*}*Instituto Superior de Ingeniería de Software (ISISTAN-UNICEN)*

Tandil, Buenos Aires, Argentina

[†]*CONICET, Argentina*

[‡]*CIC, Buenos Aires, Argentina*

^{1 2 3}{arago, cmarcos, adiaz}@exa.unicen.edu.ar

Abstract—Developing high-quality requirements specifications often demands a thoughtful analysis and an adequate level of expertise from analysts. Although requirements modeling techniques provide mechanisms for abstraction and clarity, fostering the reuse of shared functionality (e.g., via UML relationships for use cases), they are seldom employed in practice. A particular quality problem of textual requirements, such as use cases, is that of having duplicate pieces of functionality scattered across the specifications. Duplicate functionality can sometimes improve readability for end users, but hinders development-related tasks such as effort estimation, feature prioritization and maintenance, among others. Unfortunately, inspecting textual requirements by hand in order to deal with redundant functionality can be an arduous, time-consuming and error-prone activity for analysts. In this context, we introduce a novel approach called *ReqAligner* that aids analysts to spot signs of duplication in use cases in an automated fashion. To do so, *ReqAligner* combines several text processing techniques, such as a *use-case-aware classifier* and a customized algorithm for *sequence alignment*. Essentially, the classifier converts the use cases into an abstract representation that consists of sequences of semantic actions, and then these sequences are compared pairwise in order to identify action matches, which become possible duplications. We have applied our technique to five real-world specifications, achieving promising results and identifying many sources of duplication in the use cases.

I. INTRODUCTION

This is an extended abstract of the journal article with the same name, for the MODELS 2015 Conference [1].

In mostly any software development, getting a clear understanding of system requirements and describing them in an accurate and unambiguous manner is a necessity. A deficient analysis and documentation of requirements usually has negative effects downstream, compromising the success of a project. Requirements are generally specified using textual documentation, such as use case specifications. In spite of existing guidelines for writing use cases, the harsh reality is that the actual use cases generated in software projects do not often meet the standards of what it is considered a “good” use case model. We argue that the duplication of functionality in use case specifications is a major problem in software development and that analysts should deal with it since early stages of the development.

Duplicating functionality is the action of repeating some interactions between the system and actors across multiple documents. Several factors contribute to this duplication phenomenon, namely: applying use cases in large projects with many requirements, having inexperienced analysts documenting the requirements, having requirements that change very often, or abusing of copy/paste features in text processors, among others. Although duplication is not always a quality defect, and it might be there for the sake of readability of non-technical stakeholders, the issues entailed to lack of modularity and abstraction can have a profound (negative) effect on the developers conducting activities such as effort estimation, project planning, architectural design, change impact analyses and evolution management.

In this context, we present a novel tool called *ReqAligner* that aids analysts in the search of duplicate functionality in textual use cases by means of advanced text processing techniques. *ReqAligner* implements an algorithm that permits to quickly discover duplicate functionality by combining state-of-the-art *Natural Language Processing* (NLP) and *Machine Learning* (ML) techniques with *Sequence Alignment* (SA) algorithms (typically used in the bioinformatic domain). The tool also supports the process of improving requirements by guiding the analyst in the visualization of behavior that appears to have duplication across a set of use cases and recommending alternatives to resolve such duplication in the form of use case refactorings that use UML relationships (e.g., inclusion, extension, inheritance, among others).

In order to evaluate our approach, we have performed a series of experiments in five publicly available case-studies. These experiments were developed to address questions related to: i) whether *ReqAligner* works well in real-world specifications, and ii) whether the recommendations provided to analysts are helpful for improving the use cases. The results obtained so far are encouraging, as our approach has correctly detected most duplication problems and provided helpful refactoring solutions for the affected use cases.

REFERENCES

- [1] A. Rago, C. Marcos, and A. Díaz-Pace, “Identifying duplicate functionality in textual use cases by aligning semantic actions,” *Software and Systems Modeling*, August 2014.