Metaheuristic Aided Software Features Assembly

José del Sagrado and Isabel M. del Águila and Francisco J. Orellana

1 INTRODUCTION

Optimization and meta-heuristic techniques have found wide application in most areas of engineering. Software engineering is one of such areas [8]. We strongly believe that having these techniques available in a CASE (Computer-Aided Software Engineering) tool would be considerably helpful for any software developer, increasing their ease of use even for non-expert users. One important issue addressed during a software development project is the determination of which features or requirements should be covered by the product that is being constructed [3]. In most software projects, all customer demands cannot be fulfilled within reasonable time and resource constraints and must hence be limited in some way [2]. This limitation is performed by means of prioritization of the candidate requirements and selection of the best subset of requirements according to the resources available. This problem, known as the next release problem (NRP) [1], has been widely addressed applying metaheuristic optimization techniques [1, 6, 5].

The CASE tools that are related to requirement analysis stage are called CARE (Computer Aided Requirement Engineering) tools [10]. InSCo Requisite [11] is an academic web CARE tool, which aids in requirement management. The fact of having the possibility of make changes to the tool, gives us an exceptional opportunity to afford the integration of AI techniques in a CARE tool. The use of metaheuristic techniques constitute a valuable aid for experts who must decide what is the set of requirements that has to be considered in the next development stages when they face contradictory goals. InSCo-Requisite embed a new functionality called Metaheuristic Aided Software Features Assembly (MASA), in order to take advantage of optimization in a software development project. The rest of the paper is structured as follows. Section 2 summarizes the problem description. In Section 3 we describe MASA workflow, paying special attention to its different stages and to the selection of a solution for its development. Finally, in Section 4 we present the obtained conclusions.

2 REQUIREMENT SELECTION

Requirements collect the needs or conditions that have to be met for the software product. Task related to requirement management have a very different nature. Some of them are related to decision making about quality, risk or viability of requirements. One of these task is requirement triage, that can be defined as the process of determining which requirement (from those gathered from customers) a product or release should satisfy given available personnel, time and other resources [3]. When we face with NRP, it is assumed that there is a set of customers, \( C = \{c_1, c_2, \ldots, c_m\} \) and a set of possible software requirements, \( R = \{r_1, r_2, \ldots, r_n\} \). The set \( R \) is the master list of all functionality agreed with customers. All customers are not equally important for a given project, each one has a weight, \( W = \{w_1, w_2, \ldots, w_m\} \). Each customer has to assign a value \( v_{ij} \) to each requirement in \( R \), i.e. the level of priority that customer \( c_i \) assigns to requirement \( r_j \). Thus, for a given a requirement \( r_j \) its score, \( s_j \), is defined as the weighted sum of its values as: \( s_j = \sum_{i \in C} w_i v_{ij} \). The set of scores will be denoted as \( S = \{s_1, s_2, \ldots, s_n\} \). In addition, each \( r_j \) has an associated effort \( e_j \) measuring its software development effort. \( E = \{e_1, e_2, \ldots, e_n\} \) is the set of efforts. The problem is to select a subset of requirements \( \hat{R} \subseteq R \) to be included and developed in the next software release, which maximizes the score (the total satisfaction of the customer) and minimizes the total effort needed to develop it, within the effort limit \( B \) established for the project:

\[
\begin{align*}
\text{Maximize } & \text{ sat}(\hat{R}) \\
\text{Minimize } & \text{ eff}(\hat{R}) \\
\text{subject to } & \text{ eff}(\hat{R}) \leq B
\end{align*}
\]

where the satisfaction (sat) and development effort (eff) of a subset of requirements \( \hat{R} \) can be computed, respectively, as: \( \text{sat}(\hat{R}) = \sum_{j \in \hat{R}} s_j, \text{eff}(\hat{R}) = \sum_{j \in \hat{R}} e_j \), using \( j \) as an abbreviation for \( r_j \). This problem is known to be NP-hard [1].

Numerous metaheuristics algorithms have been applied to NRP (see [6]). MASA uses three multi-objective algorithms, i.e. Greedy Randomized Adaptive Search Procedure [7], Non-Dominated Sorting Genetic Algorithm [4] and Ant Colony System [9] to solve NRP extending InSCo-Requisite capabilities.

3 SOFTWARE FEATURES ASSEMBLY

The contract between customers and developers collects only the set of requirements that has been selected in order to be included in the development of a software product. Metaheuristics techniques are able to automatically find solutions to NRP, but if we are placed at a commercial level in a software development project, there are other relevant factors for the solution, (as, for example, market opportunity), which can hardly be modelled inside a search problem.

\[1 \text{ Department of Languages and Computation, University of Almería, Spain, email: jsagrado@ual.es} \]
MASA architecture is depicted in Figure 1. Its workflow combines classical approaches for requirement analysis stages (stage 1) with metaheuristics search techniques (stage 2). The set of candidate solutions obtained by these techniques constitute a first approach (as they do not take into account commercial considerations or opportunities) based on NRP, on the way to select the software features assembly that will be finally developed. Requirement managers analyze all these solutions with the aid of quality indicators that are graphically represented in InSCo-Requisite (stage 3).

The goal of stage 1 is to gather all requirements from customers together with the necessary information in order to manage the software development project. At this level, InSCo-Requisite allows groups of clients/users to work in cooperation through Internet in order to define the pull of initial requirements. When the administrator enrolls a user, \( c_j \), in a project, assigns a value \( w_j \) that represents the importance of this user for the company and the project. The tool guides the requirements management by means of templates and associated diagrams. Functional requirements, which are located at the lowest level of abstraction are modelled using scenarios and use cases. The set \( R \) of all these requirements represents the software functionalities required by users. Also in this stage, users have to perform the task of estimating the benefit \( v_{ij} \) that provides to her/him the completion of each functional requirement, even if the requirement has been proposed by other users. As a result, we obtain a database that contains both the requirements as all data that define NRP.

Once the software features are identified they have to be assembled into a solution (stage 2). Metaheuristics search techniques are in charge of this task and find a set of non-dominated solutions for the NRP. Users can access to the metaheuristic assemble of software features from the main interface of InSCo-Requisite by clicking on the MASA option. Then the user selects one of the three metaheuristic techniques offered (i.e. GRASP, NSGA-II or ACS). Next, the algorithm computes a set of non-dominated solutions or valid alternatives between which to decide.

Requirement managers analyze all these solutions (stage 3), with the aid of quality indicators that are graphically represented in InSCo-Requisite. In order to compare two solutions, besides the number of requirements in a solution, the set of requirements that comprises it, its satisfaction and effort, it worthwhile to have a measure of the amount covered by the solution with respect to everything raised by the client. For a given client \( c_i \), the coverage measure associated to a solution \( \hat{R} \) is defined as

\[
\text{cov}_{ij}(\hat{R}) = \frac{\sum_{j \in R} w_j \cdot v_{ij}}{\sum_{j = 1}^{n} w_j \cdot v_{ij}}
\]

(MASA can compare visually two of the solutions obtained on the previous stage. MASA uses histograms to represents, the coverage measure of the solution and the support provided by each client to each requirement included in the solution.

4 CONCLUSIONS

In order to determine which features should be included in a software product that is being developed, we have used three metaheuristic search algorithms. These techniques have been embedded seamlessly in a computer aided software engineering tool, InSCo-Requisite, as a new functionality called MASA. MASA allows requirement managers to make a what-if analysis using the candidate requirements. The assembly process of the requirements is done in three stages. The first gathers candidate requirements, the second finds solutions to the problem using metaheuristic algorithms and the third analyses the solutions found and selects one of them for its development. MASA provides a graphical representation of quality indicators in order to assist project managers. We have shown how Artificial Intelligence techniques can be used as an aid for software developers, but it is not enough to give solutions, it is also necessary to offer other analysis tools in order to facilitate the exploration or exploitation of the results provided by AI techniques.

ACKNOWLEDGEMENTS

This research has been funded by Spanish Ministry of Education, Culture and Sport under project TIN2010-20900-C04-02.

REFERENCES