Next Release Tool

Fatma Başak Aydemir, Dagmawi Neway Mekuria, Paolo Giorgini, John Mylopoulos

University of Trento, Italy
{aydemir, mekuria, pg, jm}@disi.unitn.it

Abstract. In this paper we present Next Release Tool (NRT), a software tool that supports modeling and reasoning to solve the next release problem. NRT supports a goal-oriented modeling language and uses an underlying satisfiability modulo theorem (SMT) solver with optimization capabilities.

Keywords: next release problem, requirements engineering, qualitative analysis, goal modeling

1 Introduction

Evolution of software products is maintained through releases where new requirements are gathered for certain periods of time, and a selection is made based on various criteria such as cost of implementation and value to the customer. We refer to the problem of selecting the optimal set of requirements for the next release as *the next release problem* (NRP) following the literature [1].

In the literature NRP has been formulated as either a single [1] or multi-objective [2] optimization problem and many search-based software engineering techniques have been adopted towards the solution. The search-space is constituted by flat structures where each node is assigned a set of numeric values corresponding to a set of objectives. Such approaches have two major drawbacks: *(i.)* the flat structure is not expressive enough to capture the hierarchical structure representing 'why' relationship between higher and lower level requirements, and other relations exist between them, *(ii.)* it is difficult to estimate the absolute quantitative values for the objectives. Modern software development techniques rely on relative estimation rather than absolute values. In order to overcome these challenges, we are interested revisiting NRP using a goal based approach, which is a more expressive method of modeling requirements than using flat, isolated requirement nodes and reason on these models using optimization modulo theory (OMT) techniques which are powerful reasoning procedures that have never been applied to NRP domain.

In this paper we present Next Release Tool (NRT), a software tool that supports goal-based modeling and reasoning to solve the next release problem. NRT supports a goal-oriented modeling language and uses an underlying satisfiability modulo theorem (SMT) solver with optimization capabilities.

2 Conceptual Framework

Carlshamre et al. [3] report that industrial experts report six interdependencies exist among the requirements considered for the next release. We propose a goal-oriented conceptual model based on these interdependencies [4]. Definitions given below are tailored for NRP.

Goal. A goal is the propositional content of an intention and represents a requirement for a software product. Goals that are marked as *mandatory* have to be included in the solution set of requirements for the NRP, whereas *optional* goals may be excluded. If there is an implemented functionality whose manifestations satisfy certain goals, such goals are marked as *implemented*.

Refinement. A refinement connects a set of edges from a set of goals (children) to another goal (parent). A parent goal may have multiple refinements, each of which represent an alternative solution for the goal. A refinement may have a single outgoing edge to a parent goal.

Cost contribution. A cost-contribution relation exists between two requirements (represented as goals) when the implementation of a requirement has an effect of the implementation cost of the other requirement. This effect can be positive or negative and the relation is assigned a penalty or reward, respectively. The impact of the effect may be set by the analyst.

Customer value contribution A customer value contribution exists between two requirements when the implementation of one requirement increases or decreases the customer value of another requirement. In case of an increase, the relationship introduce a reward for the solution that includes the link, on the other hand, a decrease introduce a penalty for the solution that includes the link.

Precedence. Temporal constraints between the implementation of two requirements are represented via a precedence relationship where the source requirement has to be implemented before the target requirement. In such cases, the solutions that include the target but not the source are not valid.

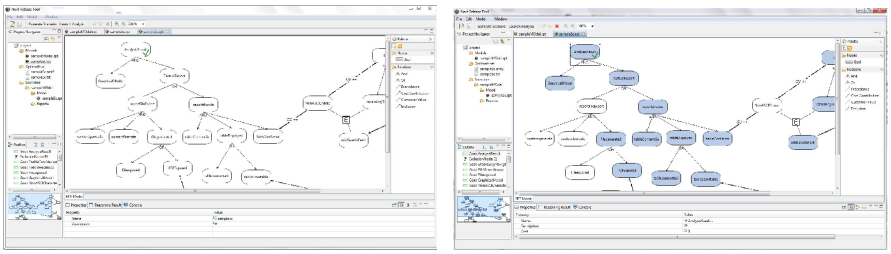
Exclusion. An analyst might decide that for certain reasons, such as the marketing strategies, some requirements should not be included together in the same release. In this case, the goal nodes that represent those requirements are connected with an exclusion relationship where the solutions that include all of these goals are not valid.

Given a model constructed by the aforementioned concepts and relationships, the *next release problem* is to find a sub-model where *(i)* all mandatory goals are satisfied, *(ii)* constraints introduced by precedence and exclusions relationships are respected, *(iii)* penalties introduced by the cost and customer value contributions are minimized, *(iv)* rewards introduced by the cost and customer value relationships are maximized.

3 Next Release Tool

The demo of Next Release Tool aims at allowing users to model requirements for the next release problem and find the optimal solution given the relationships

among the requirements. The requirements modeling language is a goal-oriented language whose main constructs and relationships are described in the previous section. The tool is available for Windows, MAC OSX, and Linux platforms and is distributed at `https://www.nextrelease.eu`.



(a) NRT with an input model (b) NRT with a highlighted solution

Fig. 1: Screenshots of Next Release Tool

Modeling. Next Release Tool allows users to import existing models as well as to build new models from scratch. Users are presented with a palette that contains shapes for concepts and various edges for different relationships and refinements. Users can drag the shapes from the palette to the white canvas at the center of the tool.

Consistency Checking. At any time point during the modeling phase, users can check whether their models are well-formed. In case of a problem, the tool lists the problematic parts of the model at the bottom. The tool also does not allow certain illegal actions during modeling, such as refining a goal into itself.

Automated Encoding. Once the modeling phase is completed, the tool automatically encodes the model into OMT formulas. The encoding differs slightly for different types of analyses. The user is asked about their preferred method of reasoning in a dialog window.

Reasoning Results Next Release Tool relies on Optimathsat OMT solver [5] for the reasoning. The tool feeds the solver with the encoding, and retrieves the result. The result provided by the solver is parsed by the tool and the model is highlighted with the optimal solution given by the solver. A written report in PDF file format is also provided. Models can be exported in various file formats such as pdf of jpg. In case of no solution, that is, the model is unsatisfiable, the user is informed with a message.

4 Demo

The demo of Next Release Tool aims at allowing users to model requirements for the next release problem and find the optimal solution given the relationships among the requirements.

Modeling. For the modeling case, we present the participants with the English-language description of three different requirements models and their corresponding proposed requirements for the next release. The first model is the requirements model for a modeling software, the second model is the requirements model of a software that supports training humans for the emergency management, and the final model is the requirements model for the air traffic management. Interested participants are encouraged to build models from scratch or complete partial models. Complete models drawn by experts will be shown after a given time.

Reasoning. For the analysis part, the participants are asked to find an optimal solution given the complete models built by experts. After a given time, we demonstrate the analysis results of the tool. For the second part of the reasoning demo, we ask the participants to modify the models, add or remove relations, add or remove mandatory goals, or change the weight of the existing relations and find the optimal solutions. Their answers are then compared with the results obtained by the tool and how their changes on the model affect the optimal solution is shown. The solution reports, models where the solution is highlighted, and the input models are exported into different file formats and shared with the participants upon their request.

5 Conclusions

Our work on NRT is ongoing as part of our current research. The current version of the tool is a result of an iterative development process, where the release of internal versions of the tool has been followed by evaluation activities.

Future work about NRT includes *(i)* expanding the conceptual framework to include multiple stakeholders and *(ii)* adding multiple views for global and local perspectives on the model.

Acknowledgements. This research was partially supported by the ERC advanced grant 267856 'Lucretius: Foundations for Software Evolution'.

References

1. Bagnall, A., Rayward-Smith, V., Whittley, I.: The next release problem. Inf. Softw. Technol. **43**(14) (December 2001) 883–890
2. Zhang, Y., Harman, M., Mansouri, S.A.: The multi-objective next release problem. Proc. 9th Annu. Conf. Genet. Evol. Comput. - GECCO '07 (2007) 1129–1136
3. Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Natt och Dag, J.: An industrial survey of requirements interdependencies in software product release planning. In the Proceedings of the Fifth IEEE International Symposium on Requirements Engineering (2001) 84–91
4. Aydemir, F.B., Mekuria, D.N., Giorgini, P., Mylopoulos, J.: Solving the next release problem through qualitative reasoning. to appear
5. Sebastiani, R., Trentin, P.: Pushing the envelope of optimization modulo theories with linear-arithmetic cost functions. In the Proceedings of the 21st Conference on Tools and Algorithms for the Construction and Analysis of Systems (2015) 335–349