PhD Thesis Review: Software Metrics Refinement

by Adrian Mihai VIŞOIU

In the domain of software quality there are numerous quality systems. Scientific literature describes characteristics identified for software products. Each characteristic is determined by attributes. For characteristics and attributes metrics are defined and indicators are built in order to assess aspects of the product. Estimation models for characteristics are necessary, in order to determine future levels of characteristics based on current recorded data, or to identify factors that influence certain characteristics.

The objective of the thesis is building methods and techniques for software metrics estimation models refinement.

The necessity of this approach is given by the difficulties of the users to permanently collect and process a large variety of data series for the software products developed by companies. Also, the effort is intensive when estimating product behavior with respect to the quality of aggregated information given by software metrics indicators. The laborious characters of methods that need to be applied to complex models in order to make results comparable with other usual metrics, in order to compare products, highlight the necessity for this approach. There is also a disproportion between the effort for obtaining results and the quality of the results.

The paper consists of twelve chapters, gradually presenting the approached theme.

The chapter entitled *The State of the Art in Research* presents results obtained by various authors from the domain's literature, regarding metrics refinement and related domains. Theoretical results obtained by scientists are presented. Research directions are identified and the necessity for metrics refinement is demonstrated.

The chapter entitled *Software Quality* presents quality characteristics for software products and associated metrics. The quality characteristics are grouped in international standards. Systems of indicators are described and correlations between indicators lead to their elimination. Properties are assessed for a set of widely used indicators. Ways of increasing software quality are identified.

In the chapter entitled *Refinement of Software Metrics Estimation Models*, elements of model architecture are presented. When developing estimation models for certain characteristics it is desired that the analytical expressions to be simple taking into account the number of operands and operators, in order to be easy to collect the necessary data, to operate, to understand and to interpret. A model developed for software metrics estimation have a degree of complexity given by the structure of its analytical expression. Complexity indicators for model assessment are proposed. Performance criteria used in the refinement process are defined and applied to order model lists. An aggregated indicator that takes into account both the complexity of a model and its performance is defined and used in this context. The objectives of model refinement are defined and ways of obtaining refinement. Refinement is the process to obtain new models with reduced complexity from the initial models and comparable performance. Refinement is also using performance criteria for filtering automatically generated model lists and retaining the best models.

The chapter entitled *Classic Methods Refinement* presents intuitive refinement methods and methods based on model generation. Intuitive methods use specialists' experience in building models to find ways of reducing the complexity of a large initial model. Such methods include: refinement through elimination of variables which is based on a top-down approach by eliminating combinations of independent variables to obtain the refined model, refinement through complexity decrease which is another proposed technique based on reducing nonlinearities to simplify the model. Further refinement using model generation is described. Model generators are software instruments for obtaining models from a certain model class given the list of variables, the model structure, existence restrictions and datasets. For each class a model generator is developed as a software module. Each dataset contains data series for the recorded variables. The endogenous variable is specified and the generator builds analytical expressions using influence factors, coefficients, simple operators and functions. For each model structure, coefficients are estimated and a performance indicator is computed. The resulting model list is ordered by the performance indicator. The analyst chooses between the best models an appropriate form that later will be used in estimating the studied

characteristic. Linear model generators take as input a dataset containing a number of independent variables and a dependent variable and produce linear models combining influence factors. Standard nonlinear model generators use predefined analytical forms for generating models. General nonlinear model generators build automatically analytical expressions containing influence factors. Analytical expressions of models are generated directly in polish form using a backtracking based algorithm with restrictions. The nonlinear model generator is suitable for modeling as the phenomena do not always follow linear laws. The linear models generators with delayed arguments allow the elaboration of constructions which permit the modeling of the multiple stimulation effects which are found on short term in influences from all the sets. Model generators are important instruments for the different refinement methods, but also generally for model design.

The chapter entitled *Neural Network Based Refinement Techniques* presents neural networks used for estimation and methods to achieve variables lists refinement. When used for regression analysis, neural networks are considered nonlinear models used estimate the level of a dependent variable given the values for the independent variables. A feed-forward multilayered network with backpropagation learning algorithm is taken into account. It is shown by proposed indicators that after establishing the outputs and hidden layer size, further model refinement is necessary through reducing the number of inputs. When the inputs are all normalized in the activation of a certain hidden unit each input has its own weight which shows the degree of influence for that activation. If a weight is greater than another that points to the same hidden unit then the corresponding input has more significance in the activation of that unit. The technique for ranking the input variables according to their influence in a model aims decreasing the number of variables. A set of indicators is developed to assess the influence of factors at input-to-hidden level. The result of applying this technique is a reduced list of variables while the performance of the estimation remains comparable. Another proposed technique aims to improve the precision of selection and records the results of the ranking for several runs. The average values of computed indicators are taken into account when ordering variable list removing the influence of random aspects of neural network learning algorithm.

The chapter entitled *Metrics Refinement Techniques based on Genetic Algorithms* presents elements related to gene expression programming and proposes two techniques to obtain model structure refinement. These algorithms have a very specific way of representing analytical expressions of models. A chromosome is a linear structure of fixed length made up of genes. The role of the chromosome is to code an analytical expression. An initial population of chromosomes suffers iteratively the effect of genetic operators in order to produce a best model for the given dataset. An identified issue in analytical expression generation using gene expression programming is the building of apparent high complexity expressions in contrast with the objective of model refinement. Two methods based on genetic algorithms are proposed. A first method chooses the most appropriate model structure based on multiple runs of the genetic algorithm. When using model generators, the analyst must pay attention to the distribution of generated model structures in order to find patterns that indicate a model structure is more fit than others for the purpose of the research. When running the algorithm for several times, using the same dataset, it is observed that the number of generated structure types is small, the algorithm having a stronger preference for generating models from certain structures than from others. The structure that was the most generated by the algorithm is chosen to become an estimation model. The other proposed technique uses the aggregated performance indicator, previously presented in the thesis, as selection criterion for best individuals, in order to force the algorithm to produce good performance models that also have a small complexity. The experimental results for both methods show that refinement has been successfully achieved.

The chapter *Software for metrics refinement* describes aspects that the product must comply and base architecture. Implementation elements are detailed showing the peculiarities and practical aspects that differentiate from the theoretical approach.

The chapter entitled *Refinement of IT&C* projects proposes and defines quality characteristics for projects, proposes indicators for assessment, and introduces a technique for project metrics refinement.

The chapter entitled *Refinement Process Validation* presents two approaches to metrics validation. The experimental validation observes and records software product behavior over time and compares estimated values with actual ones. The structural validation of software metrics refers to validation of working hypotheses and validation of indicator properties.

The thesis includes a chapter entitled *Original solutions and Research Results Dissemination* that comes to clearly establish the major contributions of the author that differentiate from what is already found in the scientific literature. Also, the author presents the ways the scientific achievements were disseminated.

The thesis ends with *Conclusions* where it is shown that the proposed objectives were achieved, and directions for further research are identified.

The *Bibliography* contains references to relevant papers from studied and related domains.

Original contributions of the author that come out of the thesis are:

- the proposal of model complexity indicators; the complexity indicator permits objective comparison between analytical expressions regarding the number of operands and operators; such indicators are used alone or along with statistical performance indicators for model list ordering for choosing the refined model;
- the proposal of an aggregated performance indicator for a model, that takes into account both statistical quality and complexity of that model; the two criteria are balanced through importance coefficients that allow the analyst to specify proportions between the two aspects; the aggregated performance indicator is widely used in the thesis, for selecting models in both classical refinement methods and modern ones;
- the definition of a contextual framework for software metrics refinement, where models are treated structurally; the aim is to obtain model refinement; along with this concept other concepts arise and are integrated in the context: variable list refinement, structure refinement, model generation;
- in the thesis, there are theoretically described and experimentally verified refinement methods based on software instruments that are the model generators; the thesis presents model generators of different types: linear model generators, nonlinear model generators, lagged variables linear model generators; model generators are instruments included in a broader software framework, which is the model base;
- the thesis defines and describes classical refinement methods containing original approaches: the refinement through weight, the method of relevant coefficients, refinement through reducing nonlinearities, refinement through accelerated decrease in the number of variables;
- an original contribution is given by the definition and the implementation of refinement techniques using neural networks; an indicator system is defined to assess input importance and a method is proposed for reducing variable lists based on the computed indicators; another method of reducing variable lists is based on iterative runs of the algorithm in order to improve precision;
- two methods for model refinement are presented, based on genetic algorithms; the first method obtains a refined model based on the distribution of generated model structures using a genetic algorithm; the most frequently generated model structure is chosen in the analysis; the second method uses the aggregated performance indicator as selection criterion in the genetic algorithm; this way the algorithm is forced to choose individuals with both good statistical performance and low complexity;
- two validation techniques are presented: experimental validation and structural validation; algorithms are proposed, indicators are defined for these methods;
- a software structure is defined; the software structure implements the described techniques; principles and characteristics for this product are discussed;
- regarding the IT&C projects, a set of project quality characteristics is proposed, indicators are built for each characteristic; a technique for project metrics refinement is proposed in the corresponding chapter.

Future research paths are identified by the author: the techniques must be validated at large scale, in practice. Methods must be extended and new methods must be added to the existing list in order to increase the generality of the approach. The author is engaged in activities that aim to expand current research.

Prof. dr. Ion Ivan, PhD