CMSEBA 2014 Workshop Report

Richard F. Paige[1], Marouane Kessentini[2], Philip Langer[3], and Manuel Wimmer[3]

[1] University of York, UK
richard.paige@york.ac.uk
[2] University of Michigan, USA
marouane@umich.edu
[3] Vienna University of Technology, Austria
{langer,wimmer}@big.tuwien.ac.at

Abstract. This report is a summary of the First International Workshop on Combining Modelling with Search- and Example-Based Approaches (CMSEBA 2014) held at the MoDELS 2014 conference. The workshop brought together researchers from model-driven engineering and search-based software engineering. The major aims of CMSEBA 2014 were to identify motivations, problems, and requirements for combining modeling with search- and example-based approaches as well as to present different proposals supporting different kinds of model engineering tasks and how models can help as a base technology in search-based software engineering.

1 Introduction

Models are an abstraction of a problem under scrutiny and have been crucial components in engineering disciplines for millennia. They play a central role in all aspects of software engineering. The fundamental premise behind approaches and methodologies to software engineering that exploit modelling is that raising the level of abstraction at which a system is developed enables the creation of the large-scale and complex software systems being produced today [1].

Search-based software engineering (SBSE) [2] is a software development practice which focuses on couching software engineering problems as optimisation problems and utilising metaheuristic techniques to discover near optimal solutions to those problems. SBSE has been applied to a wide variety of engineering problems, including requirements management, software testing, and capability management.

Examples play a key role in the human learning process. There exist numerous theories on learning styles in which examples are used. Thus the idea of using examples to derive programs has a long tradition in computer science [10]. Like many other domains of software engineering, the modelling community is currently concerned with the use of examples, such as traceability information and different kind of models, to search for solutions that fall within a specified acceptance margin to solve specific problems.

We believe that SBSE approaches and example-based approaches to software engineering offer innovate ways in which to better discover, manage, and evaluate models in software engineering. Furthermore, we believe that the example-based and SBSE communities would benefit from state-of-the-art modelling practices in order to evaluate, compare, and improve different example-based and search techniques. Thus, the goal of the Combining Modelling with Search- and Example-Based Approaches (CMSEBA) workshop is to discover opportunities for different ways SBSE and example-based techniques can be combined with modelling, and aims to stimulate research and to grow the community working in this area. CMSEBA 2014 was organized as a half-day workshop and was co-located with the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014) in Valencia, Spain on September, 28th, 2014. This report documents the various presentations as well as the enthusiastic and intense discussions.

The remainder of this report presents the background of this workshop as well as the outcome of the moderated presentation sessions. In Section 2, we recall the history of the workshop as well as the workshop topics of interest. In Section 3, we summarize the presentations, while in Section 4 we summarize the individual discussions of the presentations as well as the overall discussions regarding the combination of modeling, examples, and search-based techniques. Finally conclusions are presented in Section 5 with an outlook on future efforts to further establish and advance the modeling community with respect to the application of examples in combination with search-based techniques.

2 Workshop History and Organisation

The CMSEBA workshop is a synthesis and integration of two workshops that have previously run successfully: the International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE'13; which was an ICSE'13 workshop in San Francisco; for more information see `http://www.cs.york.ac.uk/es/cmsbse`) and the International Workshop on Model-driven Engineering By Example (MDEBE'13; which was a MoDELS'13 workshop in Miami; for more information see `http://mdebe2013.big.tuwien.ac.at`). CMSBSE'13 focused specifically on combining search and modelling while MDEBE'13 focused specifically on combining example-based techniques and MDE. Merging both workshops resulted in running a broader workshop that takes into account modelling (and not just MDE), search techniques and example-based techniques explicitly. Having this broader scope should lead to identification of greater synergies, while building on the successes of previous workshops.

The specific contribution of CMSEBA is to propose an open discussion space which invites contributions to the following topics:

- Search or example-amenable representations of models
- Domain-specific search operators: Are the standard genetic operators sufficient? Do we need problem-specific operators?
- Models of techniques: Can modelling techniques be used to evaluate and compare different example-based or metaheuristic techniques?
- Metaheuristic or example-based techniques specific to modelling: Due to the complex nature of models, are current search or example-based techniques enough? Do we need new techniques that are tailored to manage the complexity of models?

- Comparisons of standard metaheuristic techniques against bespoke modelling-specific metaheuristic techniques
- Applications of search to modelling problems
- Integrations of search techniques that are enabled through the use of modelling
- Novel algorithms for search when applied to modelling formalisms
- Search techniques applied to metamodels
- Machine learning applied to modelling
- Benchmarking of example-based techniques applied in MDE
- Prediction models for modelling problems
- New MDE problems that have not been tackled by previous example-based approaches
- Learning from model repositories
- Solving case studies by applying by-example approaches
- Search techniques applied to operations on models (e.g., model transformations): Examples include white or black box testing, test case generation
- Optimising models at runtime: Can we efficiently use metaheuristic techniques to optimise models that are used at runtime?
- Proposals of benchmark modelling case studies against which different metaheuristic techniques can be evaluated
- Tool papers related to combining examples, search, and modelling
- Position statements on ways in which modelling, examples or search could be combined

3 Papers and Presentations

The papers and the associated presentations are available on the workshop web pages (`http://www.cs.york.ac.uk/es/cmseba`).

Crepe Complete: Multi-objective Optimization for Your Models *by Dionysios Efstathiou, James R. Williams, Steffen Zschaler*. This contribution discusses an extension of Crepe [8], initially designed as a framework for single-objective optimization based on Epsilon, to support multi-objective optimisation. In particular, the authors provide an experimental comparison between an optimization problem implemented in Crepe and in native Java concerning the quality of found solutions and execution time needed. For a post-workshop update on the comparison results we refer the interested reader to Section 4 of this report.
Graph Query by Example *by Gabor Bergmann, Abel Hegedüs, György Gerencser, Daniel Varro*. Model Transformation By-Example (MTBE) approaches [4, 7, 9] have been the first by-example approaches in the field of MDE and have been the first target within the MDE field which was tackled with search-based techniques [3]. The work by Bergmann et al. discusses a specific kind of MTBE, namely how to derive graph queries by example. In particular, Bergmann et al. present an extension to the EMF-IncQuery model query tool [6] that allows the user to define how the result of a query should look like by using an example rendered in the concrete syntax of the modeling languages. Based on this input, a general query is derived which is then used to search for all such occurrences of the general pattern behind the defined example.

Design-Space Exploration in MDE: an initial Pattern Catalogue *by Ken Vanherpen, Joachim Denil, Paul De Meulenaere, Hans Vangheluwe*. Design-space exploration is becoming more and more important in the area of MDE. However, a multitude of Design-Space Exploration (DSE) techniques exist in literature which makes choosing the most appropriate technique for a specific problem a challenging task. Furthermore, the integration of DSE techniques into MDE approaches is still not mature enough, but highly needed to provide a higher degree of automation in the design process. The paper by Vanherpen et al. follows the idea of software design patterns to document how DSE techniques are used in MDE. By following this approach, they introduce an initial pattern catalogue to categorise and characterize the current state-of-the-art of embedding different DSE techniques in MDE.

EXEMPLAR: an Experimental Information Repository for SBSE Research *by Jose Antonio Parejo, Sergio Segura, Pablo Fernandez, Antonio Ruiz Cortes*. Experiments are an important cornerstone of software engineering research, and thus, an increasing amount of studies are using this kind of instrument. However, this leads directly to the need of replication and review of the experimental setups, executions, and results. The idea behind EXEMPLAR (EXpErimets Management PLAtfoRm) is to provide this kind of information by lab-packs which are hosted in the EXEMPLAR online platform. This platform provides not only a systematic way to represent experiment information, but also checks for the validity of the experiments as well as powerful search capabilities. The platform supports two modeling languages for providing experimental descriptions, namely SEDL and MOEDL, which are presented by Parejo et al.

Efficient Model Querying with VMQL *by Vlad Acretoaie, Harald Störrle*. Acretoaie and Störrle consider model queries using the Visual Model Query Language (VMQL) [5] which allows to develop model queries in a by-example manner using the concrete syntax of the modeling languages. In particular, Acretoaie and Störrle present a study of the efficiency of such model queries for ad-hoc model querying scenarios, i.e., scenarios where the user is exploring the model interactively by using queries which of course should provide the results immediately. The results of performance evaluations over a set of different models with different sizes indicate that VMQL is usable for ad-hoc model querying.

4 Discussions

Each paper presented at CMSEBA'14 led to a short discussion session, that both addressed technical issues related to the paper, and also more general themes. The workshop organisers attempted to stimulate the discussion to help tease out recurring themes, and to potentially identify interesting and profitable directions for future research.

One of the key topics for discussion came about as a result of Benoit Baudry's invited keynote, *Searching models for proactive software diversification.* A key element of this talk was the use of *novel* search criteria – in this case, measures of diversity in a software landscape. Diversity, in general, is the notion that a population may be (informally) more "survivable" if there is more variability within it. In a software context, software *monocultures* – such as through use of OS, networking protocols, APIs – have been identified as a particularly significant source of risk (e.g., in terms of exposure to security vulnerabilities). Baudry's keynote discussed software diversity and in particular ways in which diversification could be treated as a search problem, enabled by modelling. This led to significant discussion on topics such as:

- different novel search criteria beyond diversity;
- the feasibility of expressing novel search criteria in tractable, efficient fitness functions;
- the value of domain-specific languages and modelling languages in expressing search criteria in effective ways;
- more generally, novel applications of search, inspired by work on software diversification.

It was suggested during the discussion that search and modelling might be effectively combined to help identify problematic software monocultures (as there may be value in a software monoculture in certain circumstances).

Another general topic of discussion was a result of Zschaler et al's talk on Crepe Complete, which presented a generic implementation/representation (based on Eclipse EMF) for models that were amenable to search. The talk showed how Crepe Complete could be used for different search problems, but also indicated that it may have performance problems – this was the classical generic-vs-bespoke design issue that arises in search-based software engineering (and other disciplines). Because the performance of the generic model representation was quite poor compared with bespoke implementations, questions were asked about the practicality and feasibility of developing a performant generic implementation, and whether the cost-benefit argument for generic implementations vs bespoke implementations was skewed towards the bespoke. However, after the workshop, it was discovered that the performance results for the generic implementation were in fact *flawed*: the authors of the Crepe Complete noted:

> The results in the paper showed that Crepe performed worse than the bespoke implementation on two fronts: speed and solution quality. Since the workshop (October 2014) we have been able to improve both. We devised a more optimal encoding of the problem, which speeds up Crepe by up to a factor of 2. Furthermore, we discovered a bug in the Crepe implementation of the fitness function which severely, and inaccurately, punished Crepe solutions. Experiments are still underway to determine how much Crepe has improved and we plan to publish an updated version of this work, including a discussion on the challenges of developers working with domain experts when encoding problems for search.

As such, the question of whether generic-vs-bespoke implementations of model representations and fitness functions are to preferred is still very much open.

The next topic to be discussed was the use and importance of *patterns* in defining search problems. Some search problems have recurring characteristics, and search solutions definitely have recurring characteristics. Can these patterns be formalised and expressed in such a way so as to inform, for example, decision making related to whether a generic representation may be suitable for a particular problem?

Finally, there was discussion related to the importance of repositories of modelling and search problems (e.g., with examples of queries, constraints, transformations etc) that could be used to run repeatable experiments. The attendees agreed that this was extremely important and valuable, and perhaps existing repositories such as `github` could be used to support this.

The audience finally noted that the workshop had proven to be extremely valuable, insightful (and intense) and hoped to have a follow-up workshop at a future MoDELS conference.

5 Conclusion

As noted, at the end of the workshop many participants agreed that there is a need for further work on the foundations as well as application of modeling with examples as well as to further elaborate on the usage of search-based techniques in MDE. A future research line for an upcoming CMSEBA workshop (e.g., at MoDELS 2015) may include proposals for development of repositories of standardised examples (e.g., to support sharing of experiments and experimental results), as well as experiments on novel fitness functions and search criteria.

Acknowledgement

The authors would like to thank all the authors and participants of CMSEBA 2014 for their contributions.

References

1. J. Bézivin, R. F. Paige, U. Aßmann, B. Rumpe, and D. Schmidt. Manifesto - Model Engineering for Complex Systems. *CoRR*, abs/1409.6591, 2014.
2. M. Harman, S. A. Mansouri, and Y. Zhang. Search-based software engineering: Trends, techniques and applications. *ACM Comput. Surv.*, 45(1):11:1–11:61, 2012.
3. G. Kappel, P. Langer, W. Retschitzegger, W. Schwinger, and M. Wimmer. Model transformation by-example: A survey of the first wave. In *Conceptual Modelling and Its Theoretical Foundations*, volume 7260 of *LNCS*, pages 197–215. Springer, 2012.
4. M. Kessentini, H. A. Sahraoui, and M. Boukadoum. Model Transformation as an Optimization Problem. In *Proc. of MoDELS*, volume 5301 of *LNCS*, pages 159–173. Springer, 2008.
5. H. Störrle. VMQL: A visual language for ad-hoc model querying. *J. Vis. Lang. Comput.*, 22(1):3–29, 2011.
6. Z. Ujhelyi, G. Bergmann, Á. Hegedüs, Á. Horváth, B. Izsó, I. Ráth, Z. Szatmári, and D. Varró. EMF-IncQuery: An integrated development environment for live model queries. *Sci. Comput. Program.*, 98:80–99, 2015.
7. D. Varró. Model Transformation by Example. In *Proc. of MoDELS*, volume 4199 of *LNCS*, pages 410–424. Springer, 2006.
8. J. R. Williams. *A Novel Representation for Search-Based Model-Driven Engineering*. PhD thesis, University of York, 2013.
9. M. Wimmer, M. Strommer, H. Kargl, and G. Kramler. Towards Model Transformation Generation By-Example. In *Proc. of HICSS*. IEEE, 2007.
10. M. M. Zloof. Query-by-example: A data base language. *IBM systems Journal*, 16(4):324–343, 1977.