MASTER

An evaluation of MOMoT

van Lier, B.J.J.

Award date:
2017

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
An evaluation of MOMoT

A thesis submitted in partial fulfillment of
the requirements of a Business Information
Systems Degree

B.J.J. van Lier - 0770065 -
b.j.j.v.lier@student.tue.nl

Supervisors:
dr. P.M.E. Van Gorp
dr. N. Sidorova
dr. M. Razavian

Eindhoven, February 2017
Abstract

Marrying Optimization and Model Transformations (MOMoT) is a technique proposed by Fleck, Troya and Wimmer (2015). It combines model transformations from the Model-driven Engineering (MDE) domain with optimization algorithms from the Search-based Software Engineering (SBSE) domain. MOMoT is inspired from the Software Engineering field, and the evaluation of the tool in this context looks promising. This thesis presents a pilot study that aims to evaluate the use of MOMoT in the Industrial Engineering domain, in the context of advising academic decision makers about the inclusion of model-based methods in the curriculum. In this context, MOMoT is viewed as a model-based optimization tool. It is compared with the Multi-objective Evolutionary Algorithms (MOEA) framework, which is a code-based optimization tool. The Method Evaluation Model (MEM) (Moody, 2003) is applied to assess Industrial Engineering students’ attitudes towards the use of MOMoT and the MOEA framework. The results of applying MEM do not enable statistically significant conclusions to be drawn due to low participation in the research. Nevertheless, this pilot study is successful as it shows that the methodology proposed in this thesis can successfully be applied in the evaluation of the attitudes towards use. Hence, it is recommended to repeat this research on a large scale to obtain sufficient results for providing well-founded advice to decision makers in the academic community.
1 Introduction

The inspiration for this research was provided by the work of Martin Fleck (Fleck, Troya and Wimmer (2015); Fleck, Troya and Wimmer (2016); Fleck (2016)). In his PhD thesis, Fleck introduces MOMoT, which is not only an acronym for “Marrying Optimization and Model Transformations”, but also the name of a tool that aims to achieve exactly that. This section starts by providing the theoretical background of MOMoT. Then, the research presented in this thesis is introduced and motivated.

1.1 Background

The combination of optimization and model transformation is in essence the combination of two distinct disciplines, namely Model-driven Engineering (MDE) and Search-based Software Engineering (SBSE). The synergies of those fields, and thereby the foundations of MOMoT, were presented in a paper by Kessentini, Langer and Wimmer (2013). MOMoT is the implementation of the approach proposed in that paper.

Mens and Van Gorp (2006) describe MDE as “a discipline in software engineering that relies on models as first class entities and that aims to develop, maintain and evolve software by performing model transformations”. Various authors (Atkinson and Kühne (2003); Schmidt (2006)) consider MDE as the next step in increasing the level of abstraction at which software is written. These abstractions aim at enabling the development of increasingly complex software systems. MDE is a promising effort in that direction, as it proposes a visual approach to software development.

Model transformations are the main challenge in this discipline (Sendall and Kozaczynski, 2003). These are at the same time the primary enabling technology of MDE. Relevant definitions with respect to model transformations are proposed by Mens and Van Gorp (2006): “A model transformation is the automatic generation of one or multiple target models from one or multiple source models, according to a transformation definition. A transformation definition is a set of transformation rules that together describe how a model in the source language can be transformed into a model in the target language. A transformation rule is a description of how one or more constructs in the source language can be transformed into one or more constructs in the target language.” An example of a class diagram can be found in Figure 1. This class diagram contains the basis for the source and target models. An example transformation rule is shown in Figure 2. These examples are typical for the software engineering domain.

Search-based Software Engineering as a field was defined by Harman and Jones (2001). At the time, it was described as a rapidly growing field of research, proposing a search-based approach to software engineering problems. This approach enables the use of optimization techniques to solve problems that were previously solved manually. As such, better solutions can be found faster.

A typical example of a relevant problem from the software engineering domain is the division of

![Figure 1: Example class diagram. Image taken from Fleck, Troya and Wimmer (2016)](image-url)
features over classes, with the goal of optimizing the conflicting objectives coupling and cohesion. Another example is automatic refactoring to improve software design. Refactoring can be automated by searching for optimal designs. Yet, searching is costly, and should thus be performed with as little steps as possible. Hence, a constraint for a search could be a maximum number of steps.

Both MDE and SBSE emerged in the context of software engineering, and thus primarily focus on problems in this field. That does not mean that other fields cannot benefit from these developments. Search-based optimization is applicable in many fields, and visual model transformations can enable non-expert programmers to develop software that would otherwise have been out of their reach. Moreover, MDE and SBSE are not only promising fields individually. As claimed by Kessentini, Langer and Wimmer (2013), synergies exist between the two, suggesting that a combination of these fields is potentially valuable as well.

1.2 Motivation

Kessentini, Langer and Wimmer (2013) and Fleck, Troya and Wimmer (2015) describe the synergies between MDE and SBSE from the software engineering perspective. Here, a different view on model-driven techniques, and on MOMoT in particular, is proposed. Fleck, Troya and Wimmer (2015) explain the combination of optimization and model transformations from the MDE domain, primarily focusing on MDE problems. Instead, it is also possible to consider MOMoT as a visual optimization tool. The models involved are in fact executable models which are not domain specific. Such models can also be applied for problems outside the software engineering domain. An example of an application outside of the software engineering domain is shown in Figures 3 and 4. The class diagram and transformation rule are part of the implementation of the stack problem, which is the example case used in this research. Although the case is used by Fleck, Troya and Wimmer (2015), it originates from the logistics domain.

Viewing MOMoT as a visual optimization tool can be very interesting from a business analysts' perspective. Their field is likely to profit from the use of such a tool, as business analysts aim to improve business processes. In order to achieve this goal, a broad set of methods is available, one of which is the use of software. Applying software is not only supporting business processes, it also

Figure 3: Example class diagram for a problem outside of the software engineering domain. Image taken from Fleck, Troya and Wimmer (2015)
enables the analysis of those processes, thereby resulting in possible improvements. However, as business analysts have an extensive toolbox available to them, they do not have the same skills as professional software developers. Their programming skills are mostly on a basic level, sufficient to understand the functionality of a tool, but not enough to implement a case in the tool. This means that they cannot profit from the tool’s functionality. This is a logical consequence of the broad orientation of the field. Yet, it means that tools which require complex coding can often not be used by business analysts, even though the functionality of those tools can be very useful to them.

From this perspective, it could be beneficial to let business analysts work with model-driven tools. Perhaps, business analysts can benefit more from these tools than software engineers. With professional programming skills, model-driven development tools only become useful when complexity of the software gets so high, that taking a higher level of abstraction in defining it becomes beneficial. For business analysts on the other hand, model-driven tools can also be useful for relatively straightforward problems, as their limited programming skills restrict them from implementing such problems in code.

Thus, although only introduced recently, MOMoT is a state-of-the-art tool with many potential applications. Up to now, several aspects of MOMoT have been evaluated by Fleck (2016), leading to promising results. The fact that empirical evaluation is included in Fleck (2016), is quite exceptional in the field of software engineering. It has repeatedly been shown that research in this field often lacks any formal evaluation of its results (Tichy et al. (1995); Zelkowitz and Wallace (1997); Zelkowitz and Wallace (1998)).

The evaluation provided in Fleck (2016) is primarily performed from the software engineering perspective, assessing the suitability of MOMoT for typical software engineering problems. However, taking the above into account, it could be interesting to perform an evaluation of MOMoT that takes the business perspective, putting the use of MOMoT by business analysts central. Although it is expected that business analysts can profit from the use of MOMoT, it is likely that professionals will take significant time to get used this tool. A common phenomenon is that professionals tend to be biased towards the tool they are working with on a daily basis. This is known as confirmation bias, which is reviewed extensively by Nickerson (1998). The visual approach that is proposed by MOMoT differs significantly from other optimization approaches, and therefore confirmation bias is likely to occur if professional business analysts are asked to evaluate MOMoT. Students are far less likely to be biased in this sense, because they have not been working with any tool for longer periods of time. In fact, students are introduced to new tools on a regular basis, and as such they will be more open to other new tools.

In particular, this research focuses on Industrial Engineering students, as they possess several convenient characteristics. Next to the lower likelihood of confirmation bias, this group of students can be considered the business analysts of the future. Furthermore, the environment in which this research is carried out, supports the use of MOMoT by future business analysts. Industrial Engineering students are available in large numbers at Eindhoven University of Technology (TU/e). In addition, a significant part of these students has followed the course Algorithmic Programming for Operations Management, which is an introductory Java course. Hence, it can be assumed and checked that the students who followed this course all have similar programming experience.

In this thesis, attitudes towards use are studied. Although the concept of attitudes can be explained in various ways, there is a very particular reason for using it in this context. Attitudes
towards use are part of the Technology Acceptance Model (TAM). They have been proven to be related to the intention to use (Davis, 1989). In fact, TAM is the most prominent model when it comes to identifying factors that explain the use of software (Lee, Kozar and Larsen (2003); King and He (2006)). Hence, attitudes can provide valuable information about the use of MOMoT by future business analysts.

As this research involves students as participants, the focus is not directly on a business audience. Instead, the primary target audience for this research is the academic community, in particular the decision makers about the curricula of students in the field of Industrial Engineering. Those people determine what the business analysts of the future are learning, and as such they are shaping the next generation of professionals. This research is a pilot to assess the use of MOMoT by students on a small scale. Therefore its results can be interesting to decision makers in the academic community. Of course, implicitly a business audience can also profit in the long-term, as better educated students will result in a better educated workforce in the future. If the results of this pilot are promising, continuing this line of research on a larger scale should certainly be considered.

Given all of the above, a research question has been formulated. It aims to include all the aspects that are mentioned, and sets the scope for the research:

What are Industrial Engineering students’ attitudes towards the use of a model-driven optimization technique, compared to a traditional technique?

The model-driven technique in this case is MOMoT. The traditional technique with which it will be compared, is the Multi-objective Evolutionary Algorithms (MOEA) framework, which is in fact one of the building blocks of MOMoT.

In the next chapter, the methodology used in this research is described in detail. The starting point for the methodology is TAM. Several extensions of TAM are considered. One of those extensions is selected to guide the further definition of this research. The methodology section ends with the details of the experiment and survey which are performed to collect data regarding students’ attitudes. Chapter 3 contains the results of the pilot. A discussion of these results follows in Chapter 4. Lastly, conclusions are drawn in Chapter 5.
2 Methodology

This thesis builds on several well-established academic sources in the definition of the methodology. First, a model is selected from the literature. Then, research methods are discussed and their suitability for this study is assessed. Next, the detailed study design is covered, for which the Goal Question Metric (GQM) approach is applied. This approach facilitates the identification of the appropriate metrics to achieve the goal of the research. Lastly, a summary of the methodology is provided.

2.1 Theoretical model

The first step in the definition of the methodology for this research concerns the selection of a model from theory. This model serves as an outline for the detailed study design. As stated in the previous section, this research is motivated using TAM. Therefore, TAM literature is the starting point regarding the selection of a theoretical model.

2.1.1 Technology Acceptance Model

Davis (1989) was one of the first to successfully identify factors that explain the usage of a technique. The model he developed is known as the Technology Acceptance Model (TAM), and it is based on the Theory of Reasoned Action (TRA) Ajzen and Fishbein (1980). TAM states that perceived usefulness and perceived ease of use are determinants of usage of a technique. As such, these two factors are suitable as a measurement scale to predict and explain usage.

The primary aim of TAM is to provide a validated alternative for the many unvalidated measures that were being used in Software Engineering practice at the time. To ensure the validity of the TAM constructs, the model was validated in multiple steps (Davis, 1989). These steps are included here to confirm that TAM is a solid model. In the first step of the validation of TAM, existing literature was analyzed to explain the use of perceived usefulness and perceived ease of use. Based on this analysis, 14 items were selected for both constructs. The validity of those items was tested in a small pilot study. As a result, the number of items for each construct was reduced to 10. Then, a field study was performed in which 112 participants evaluated two techniques. Based on the results of this study, the number of items was further limited to six per construct. A second study, which was a laboratory study with 40 participants, was performed to validate the items. Altogether, the results convincingly show that both perceived usefulness and perceived ease of use are determinants of computer usage.

TAM is further described by Davis, Bagozzi and Warshaw (1989). It is stated that actual use is a direct result of the behavioral intention that (potential) users have towards using a technology. Behavioral intention, in turn, is explained by attitudes towards use, combined with perceived usefulness and perceived ease of use. Figure 5 illustrates the Technology Acceptance Model.

![Figure 5: The Technology Acceptance Model (Davis, Bagozzi and Warshaw, 1989)](image)

An evaluation of MOMoT
usefulness. Attitudes towards use are formed by the combination of perceived usefulness and perceived ease of use. Lastly, perceived usefulness and perceived ease of use are both the result of a set of (unidentified) external variables. The complete model is shown in Figure 5.

The focus on perceived usefulness and perceived ease of use was not new when TAM was introduced. Davis (1989) mentions that these factors were related to self-predicted use in other research. Davis, Bagozzi and Warshaw (1989) take this one step further by validating the relation of those factors to actual system use. Whereas the relation with self-predicted use could be shown by asking participants to fill in a questionnaire, the proof of the relation to actual use required a more advanced study. A longitudinal study with 107 full-time MBA students was performed. After a one-hour introduction to a system they were asked to fill in a questionnaire to measure perceived usefulness and perceived ease of use. Next, they were given the opportunity to use the evaluated techniques. After a period of 14 weeks, the participants were asked to the extent to which they were using the techniques in practice. Furthermore, they were again asked to fill in a questionnaire measuring both TAM constructs. The results of this study confirmed the hypothesized influence of perceived usefulness and perceived ease of use on actual usage in practice.

2.1.2 Extensions of TAM

As it is a well-known model, TAM has been the topic of many scientific publications over the years. Adams, Nelson and Todd (1992) provide a replication of the research by Davis (1989), largely confirming the original results. Meta-analyses regarding TAM are performed by Lee, Kozar and Larsen (2003) and King and He (2006), both confirming the value of TAM. Venkatesh and Davis (2000) and Venkatesh et al. (2003) provide revisions of TAM, which are called TAM2 and Unified Theory of Acceptance and Use of Technology (UTAUT), respectively. These revisions have been considered for use in this study. In general, these more recent alternatives are more explicit
taking the perspective of the professional working with the technique under consideration. This means that the questionnaires involved in those methods are explicitly assuming that the person who is answering the questions is evaluating the technique with the idea of using it in his/her job. This research is aimed at assessing the attitudes of students. These students are not in a professional environment. They are not using any of the tools involved on a professional basis. As such, a set of questions that assumes that the people answering them are professionals has to be adjusted to ask them to a student. The questions that relate to perceived ease of use in the Technology Acceptance Method do not explicitly require professionals as respondents. The questions that measure perceived usefulness have to be refined. As the original TAM questions require the least changes, TAM is preferred over its more recent alternatives.

Another more recent extension of TAM is provided by Moody (2003), who proposed the Method Evaluation Model (MEM). This model is shown in Figure 6. Perceived usefulness and perceived ease of use are summarized as perceived efficacy. The most important difference compared to TAM is the addition of actual efficiency and actual effectiveness as predictors of perceived ease of use and perceived usefulness, respectively. These constructs are summarized as actual efficacy, or task performance. According to Moody (2003), efficiency is related to the effort required to complete a task, while effectiveness is related to the quality of the result. This is shown in Figure 7.

An important difference between MEM on the one hand and TAM2 and UTAUT on the other is that MEM uses the questionnaire proposed by Davis (1989). Only minor changes are made to the questions to make them suitable for the specific context of MEM. As such, MEM is more suitable to use in a context in which the questionnaire is filled in by students.

The validation of MEM consisted of both a laboratory experiment and a field experiment. The laboratory experiment was performed with 41 participants, who were all students in their final year of the study Information Systems. Three methods were compared. In the field experiment 21 experienced professionals participated, and only the method performing best in the laboratory experiment was included. The results of the two experiments were compared. Both experiments included a post task survey to measure perceived efficacy. Actual efficacy was only measured in the laboratory experiment. According to Moody (2003) it is meaningless to measure actual efficacy when only one method is included, as this metric is only meaningful when multiple methods are compared.

The conclusion is drawn that MEM can be useful in the comparison of the efficacy of different methods and the evaluation of the likelihood of adoption of a method in isolation (Moody, 2003). It is stated that the focus in research is often on the task performance of a method. However, the likelihood of adoption is argued to be at least as important. As already stated by Davis, Bagozzi and Warshaw (1989), a useful system does not provide any value if it is not used in practice. Similarly, attitudes towards the use of a system should not be expressed solely on the task performance of that system. Perceptions should be taken into account as well. Therefore, MEM is adopted to guide the further definition of this research.
2.2 Research methods

The software engineering domain largely lacks the use of empirical methods in the evaluation of its methods. Instead, normative methods are mainly used (Wynkoop and Russo, 1997). Moody (2003) acknowledges this issue. In fact, it is the primary motivation of his research. As a result, MEM relies on the application of empirical research methods. Therefore, this section provides an overview of such methods.

This research can be placed in the field of Empirical Software Engineering. Within this field, several research methods that can be used to observe a software engineering phenomenon are available. There is no clear consensus as to which specific methods are available, and therefore the view of several authors is presented.

Easterbrook et al. (2008) mention the following methods:

- Case studies (both exploratory and confirmatory): this type of research is “aimed at investigating contemporary phenomena in their context” (Runeson & Höst, 2009).

- Survey research: according to Wohlin et al. (2000), data in survey research is mostly gathered through the use of interviews and questionnaires. Surveys are typically applied to review a technique that has been used in a real-world setting for a while. Nevertheless, survey research is also possible in an artificial context.

- Controlled experiments (including quasi-experiments): an investigation of a testable hypothesis where one or more independent variables are manipulated to measure their effect on one or more dependent variables (Easterbrook et al., 2008). This is the most suitable type of research in an artificial context. Experiments require a high degree of control over the environment, which is typically found in an artificial context.

- Ethnographies: “focus on the sociology of meaning through field observation. The goal is to study a community of people to understand how the members of that community make sense of their social interactions.” (Robinson, Segal and Sharp (2007), cited in Easterbrook et al. (2008))

- Action research: the distinctive characteristic of this type of research is that it aims to “influence or change some aspect of whatever is the focus of the research” (Robson, 2002). It is comparable to a case study, as a phenomenon is studied in its context. However, this type of research is not only about observation. It aims for change, which in practice comes down to aiming for improvement of the current setting.

The list provided by Runeson & Höst (2009) is almost the same, with the only difference that ethnographies are not included:

- Case studies
- Survey research
- Experiments
- Action research

The reason that the authors give for this, is that they “prefer to consider ethnographic studies as a specialized type of case studies”. Wohlin et al. (2000) further limit the number of methods. They only mention the, in their view, major strategies:

- Case studies
- Survey research
- Experiments
2.3 Detailed study design

In previous sections, models and methods relevant for this research have been discussed. This section specifies the research on a more detailed level, using the Goal Question Metric (GQM) approach. GQM prescribes that the goals of a study are made explicit first. These goals are then translated into concrete measures (Basili, Caldiera and Rombach (1994); Van Solingen and Berg-hout (1999)). The three parts of the approach are consecutively on the conceptual, operational, and quantitative level. All these parts are covered in this section.

2.3.1 Goal

The goal is the conceptual part of the GQM approach. It connects best to Section 1. In fact, it can be considered as a summary of everything that has been presented about the research up to now.

According to Basili, Caldiera and Rombach (1994), a well-defined goal has multiple dimensions:

- **Purpose:** what is this research aiming to achieve?
- **Object:** which product, process or resource is being studied?
- **Quality issue:** what effect is studied?
- **Viewpoint:** from which point-of-view are the above aspects considered?

This research assesses Industrial Engineering students’ attitudes towards the use of MOMoT and the MOEA framework. Therefore, the point-of-view is that of Industrial Engineering students. As mentioned in Section 1, these students are considered to be the business analysts of the future. Consequently, decision makers in the academic community are the primary audience, in particular those in the field of Industrial Engineering. The purpose of this study is evaluation, specifically the evaluation of MOMoT as a graphical optimization tool.

A goal definition template is used to ensure that all relevant aspects are covered, and to summarize the choices made in this section. The goal definition template, taken from Wohlin et al. (2000), is based on GQM. Next to the dimensions mentioned above, it also includes some information about the context of the research. Together, these aspects form the fundamentals of the research:

```
Analyze MOMoT and the MOEA framework
for the purpose of evaluation
with respect to the attitudes towards use
from the point of view of the Industrial Engineering student
in the context of supporting academic decision makers in the development of curricula
in the Industrial Engineering domain.
```

The students involved are assumed not to be familiar with MOMoT or the MOEA framework. It has been mentioned that this makes them suitable participants for this research, but at the same time, people cannot be asked to evaluate something they do not know. Therefore, the participants have to be introduced to the tools. Below, the various options for such an introduction are described.

Students can be provided with a problem description, for which they are asked to provide a solution by using the techniques. The advantage of this approach is that a realistic setting is simulated as closely as possible. However, based on the author’s experience with the techniques, it is very unlikely that the students will get a representative view of the techniques within the short time in which they will work with them. They are very unlikely to be able to implement an example case successfully, and as such, the focus will only be on the development aspect of the techniques, and not on the results they provide.
An alternative is to provide the students with a fully functioning implementation. The implementation will run without errors, thus immediately providing access to the results provided by the tools, without having to go through the trouble of setting it all up. Although this provides a good view on the results, development is ignored in this option. Moreover, this is not a very realistic setting.

The before-mentioned options are two extremes in terms of experiencing development versus observing results of the techniques. In between, there is an alternative that provides both. Students will be provided with a functioning implementation, in the sense that it does not produce runtime errors. However, incorrect output will be returned, which is caused by a logic error (bug) in the implementation. The students will be asked to find this bug and consequently fix it, thus focusing on corrective maintenance. They are provided with correct example output, which they can compare with the output generated by the implementation with the bug. Not only does this particular way of presenting the techniques provide the opportunity to experience both development and results. It also allows the researcher to focus on particular parts of the techniques, by inserting bugs on strategic places. Furthermore, corrective maintenance is a very common activity in software engineering. According to arthur1988software (arthur1988software), more than three-fourths of the costs of software over the entire life cycle are caused by maintenance activities. Corrective maintenance is part of those activities, and as such it is a suitable basis for the evaluation of the tools.

This particular approach is not only motivated from the rational argument provided above. Fleck (2016) also focuses on corrective maintenance to assess the usability of the functionality that MOMoT provides. Usability is operationalized using the time it takes to fix the bugs and the degree of difficulty that the participants experience when fixing the bugs. In the context of MEM, bug fixing time can be considered as a measure of actual efficacy, while the degree of difficulty that participants experience can be considered as a measure of perceived efficacy. These measures are in fact very similar to the measures that Moody (2003) uses in the validation of MEM. As becomes clear in Section 2.3.3, the metrics used in this thesis are to a large extent inspired by Moody (2003), and thus show substantial similarities to the metrics used by Fleck (2016). The focus on bug fixing can therefore be explained both rationally and from literature.

2.3.2 Question

On the operational level, questions are asked to support the goal of the research in a quantifiable way. In Section 1, a research question has been provided. This question is the basis for the goal defined above. The questions provided in this section operationalize that goal.

MEM is used as the theoretical model in this research, as it promotes a balanced view on attitudes, incorporating both actual and perceived efficacy. To emphasize the importance of both types of efficacy, two questions are asked:

*What is Industrial Engineering students’ actual efficacy regarding the use of MOMoT, compared to the MOEA framework?*

*What is Industrial Engineering students’ perceived efficacy regarding the use of MOMoT, compared to the MOEA framework?*

In line with the existing evaluation of MOMoT, it is hypothesized that MOMoT performs better on both aspects. As stated by MEM, actual efficacy consists of actual efficiency and actual effectiveness. Perceived efficacy is a combination of perceived ease of use and perceived usefulness. All these constructs are implicitly referred to in the questions presented here. These questions might not reflect all the questions that decision makers in the academic community might have regarding the use of MOMoT by students, and more in general regarding the use of graphical (optimization) methods in software engineering. However, the scope is set to these two questions since only limited time is available for the execution of this research.
2.3.3 Metric

The last step of GQM is the selection of appropriate metrics for the questions defined in the previous section. A metric is considered appropriate if a change of its value reflects a change in the construct it represents. A large variety of data can be gathered from the interaction of students with the techniques. An important condition to be met is that the process of collecting this data does not significantly influence the experience of using the techniques, as this could in the end influence the evaluation. The metrics for both questions are discussed separately below, largely adopting the metrics proposed by Moody (2003).

**Actual efficacy** Context-specific metrics have to be developed to measure actual efficiency and actual effectiveness, as it is impossible to formulate metrics that are meaningful in every context (Moody, 2003). Next to the metric itself, it is important to determine which of the empirical methods discussed in Section 2.2 will be used to collect the data for that metric. Students are introduced to MOMoT and the MOEA framework and use these techniques with respect to corrective maintenance, also known as bug fixing. This means that a research method that observes a technique in its context is not suitable. Furthermore, actual efficacy cannot be measured by collecting subjective data. This leaves an experiment as the only suitable research method.

**Experimentation in software engineering** Experiments in general are characterized by a high level of control in comparison to the other methods. The combination of subjects and treatments is randomized. Multiple levels of one or more variables are applied, while at the same time keeping all other variables the same. Altogether, experimentation is the most formal research method presented in this thesis (Wohlin et al., 2000).

The approach proposed by Wohlin et al. (2000) is used, as it offers (probably) the most complete and comprehensive guide for conducting experiments in the software engineering domain. It prescribes to cover the following aspects before performing the experiment:

- Context selection
- Hypothesis formulation
- Variables selection
- Selection of subjects
- Experiment design
- Instrumentation
- Validity evaluation

All these steps are treated below.

**Context selection** Four dimensions are of importance in this respect:

- Off-line vs. on-line
- Student vs. professional
- Toy vs. real problems
- Specific vs. general

These dimensions together summarize the context of the experiment. This experiment is off-line, as it is not performed during execution in a real-life setting of any of the tools involved. Students will be used as subjects in the experiment. Furthermore, the problem that will be used in the experiment can be considered as a toy problem. Although it might be a simplification of a real
problem, the fact that it is simplified makes it a toy problem. The choices in the other dimensions makes this experiment rather specific, meaning that its validity will be limited to the specific context in which it is performed.

Wohlin et al. (2000) states that the use of students instead of professionals limits the ability to generalize the results of an experiment. Yet, Höst, Regnell and Wohlin (2000) showed that research with students can provide results which are not significantly different from the same research with professionals. Even though this result was found in a specific setting, the conclusion is drawn that it is valid for many other settings in the software engineering domain as well. Therefore, the ability to generalize the results of this research to a professional setting might not be as limited as Wohlin et al. (2000) suggest.

Hypothesis formulation

In line with the existing evaluation and the goals of MOMoT, it is expected that MOMoT performs better than the MOEA framework. The hypothesis is formulated as follows:

\[ H_0: \text{There is no difference in actual efficacy when fixing bugs in MOMoT, compared to a native encoding of the same problem in the MOEA framework} \]

\[ H_1: \text{Actual efficacy is higher when fixing bugs in MOMoT, compared to a native encoding of the same problem in the MOEA framework} \]

With the data that the experiment will provide, it can statistically be determined whether the null hypothesis can be rejected.

Variables selection

Two types of variables can be distinguished in an experiment context: dependent and independent variables. The latter are the variables that can be controlled and influenced by the experimenter. One of the independent variable is the implementation of the case in which the bugs have to be found and fixed: either model-based (MOMoT) or code-based (the MOEA framework). Other independent variables which are taken into account, are the case in which bugs have to be found, the bugs themselves, the programming experience of the subjects, the average grade of the subjects (since they are students) and in which phase of their study they are (Bachelor/Master).

Dependent variables are the variables that will be measured to determine whether changing one or more of the independent variables results in an effect. In this case two dependent variables are used. To measure actual efficiency, the time it takes to find and fix bugs is used. By using this metric, this research follows Fleck (2016) and Moody (2003). Time is one of the measures that Fleck (2016) uses in the evaluation of MOMoT to assess the difficulty that users experience in completing a certain task. Moreover, Moody (2003) uses time for the exact same purpose as is proposed here, namely as a measure for actual efficiency. The number of bugs fixed after a certain time period is used as a measure of actual effectiveness. This is also largely similar to the measure used by Moody (2003).

Selection of subjects

The subjects in this experiment are students. The choice was made not to invite just any student in the university to participate. Instead, participation was limited to people who followed the course Algorithmic Programming for Operations Management during the academic year 2016-2017. This course is specifically aimed at providing an introduction to programming with Java to students who follow the Bachelor Industrial Engineering. As a result, the subjects will have very similar backgrounds. Additionally, it can safely be assumed and checked that the students are all novice to MOMoT, as this is a very recent technique that is not being used on a large scale (yet). This does of course not mean that all subjects are the same; this is impossible. Yet, when the subjects are randomly assigned to the tools that are compared in this study, the differences that remain between them will be dealt with as well as possible. Potential participants will be introduced to this research during the course by means of a presentation, and then they are free to choose whether they want to participate or not. In that sense,
convenience sampling is applied; the most convenient people out of the ‘population’ of inexperienced programmers among Industrial Engineering students is taken. Students who subscribe, will be asked what study they are currently in (Bachelor or Master, and which field), what their average grade is, and how they rate their own programming experience. The aim of this is to get a good view of how the values of the independent variables are spread.

**Experiment design**  The experiment design is not only relevant as it states how the experiment shall be conducted. Connected to the chosen design is the set of statistical methods that can later be used to analyze the results. To determine what design should be used, it should first be determined which independent variables will be kept constant, and which will be changed. The amount of variables in the latter category, which are known as factors in the experiment, is one of the defining characteristics of a design type. The other characteristic is the number of different values (treatments) that will be used for each factor.

The hypothesis, which is stated above, contains the answer to the question which independent variable will be changed. Only the implementation of the case in which bugs have to be found and fixed will be altered; all other independent variables are either kept constant, or are randomized over the treatments. There are two treatments: subjects can work with MOMoT or with the MOEA framework. Summarizing, the experiment design should be suitable for an experiment with one factor that has two treatments. Note that in some experimental settings, in which the number of possible combinations of treatments is high, the experimenter might consider not to include all possible combinations of treatments in the experiment. However, in this case, this is not relevant as there are only two treatments.

The next aspect to cover is the division of subjects over the treatments. A simple strategy would be to expose all subjects to both treatments. However, as Fleck (2016) remarks, this could yield undesired results. The main problem that Fleck names in this regard is the occurrence of the learning effect. This effect can be described as a subject performing differently in an experiment because of the experience that this subject gathered as a result of already having been exposed to another treatment. In this case, the learning effect can be prevented by exposing the subjects to only one of the two treatments. Hence, the subjects will either work with MOMoT, or with the MOEA framework, and not with both.

In fact, there is a large amount of literature on experiment design, and the division of subjects over the treatments is an important subject in this field. Wohlin et al. (2000) mention three general design principles:

- **Randomization:** the combination of subjects and treatments has to be randomized to average out the effect of a nuisance factor (unwanted source of variation that can influence the dependent variable Kirk (1982). This is a requirement of many statistical methods that can be used for analyzing the data.

- **Blocking:** if the presence of a nuisance factor is known and measurable, subjects with similar levels of the nuisance factor can be combined in blocks. As the nuisance factor will thus be similar within a block, it will not influence the dependent variable.

- **Balancing:** an equal number of subjects is assigned to all treatments of a factor. This simplifies and strengthens the statistical analysis.

Those independent variables related to the subjects that were identified in the section on variable selection, are all potential nuisance variables, which can as a result cause unwanted variation in the results. To make sure that any nuisance variable is taken into account, all of those variables are measured beforehand. The subjects will be asked to judge their own programming experience when they subscribe to participate in the experiment. The scale will contain 5 steps, and range from ‘Novice: never programmed’ to ‘Expert: high programming experience in multiple languages’. As the subjects will all be students from the Algorithmic Programming for Operations Management course at TU/e, it is expected that the level will in general be just above the novice
level. Furthermore, the subjects will be asked which Bachelor and Master (if applicable) they are in, and what their average grade is up to this point.

If no large variation is found in the answers to those questions, blocking is not necessary, as there will not be a nuisance variable that needs to be dealt with. In this situation, a completely randomized design will be applied, meaning that every subject is randomly assigned to one of the two treatments (randomization), while at the same time assigning an equal number of subjects to every treatment combination (balancing). In case that less than 20 students subscribe, and the number is uneven, participation is preferred over balancing, meaning that one of the treatments is applied to one more person. An example of a completely randomized design is shown in Figure 8.

If it turns out that there is a nuisance variable, a randomized block design will be used. Every block will contain two subjects, such that the value of the nuisance variable is as equal as possible within each block. The subjects in a block will each be randomly assigned to a treatment, such that every treatment occurs exactly once in each block. An example is shown in Figure 9.

**Instrumentation** The next step in experiment planning is the selection of instruments that will be used in the execution of the experiment. Three categories of instruments are identified by Wohlin et al. (2000):

**Objects** The object in this experiment is the case that is used, which is the stack example case. This case has been implemented both in MOMoT and the MOEA framework by Fleck (2016), who uses it among other cases in his evaluation of MOMoT. The case itself is not from the
software engineering domain. In fact, it is a typical Industrial Engineering problem. As such, it fits in the context of this research very well.

The problem starts with a number of stacks which all have an initial load. The goal is to divide the loads over the stacks in such a way that the loads are equal, or at least as equal as possible with the given initial loads. There is a constraint regarding the load movements that are allowed: every stack has one left and one right neighbor, and loads can only be transferred to these neighbors. If stack A is the right neighbor of stack B, stack B must be the left neighbor of stack A. The objectives are to minimize the standard deviation of the stack loads (minimal standard deviation means that the stack loads are as equal as possible) and to minimize the number of load moves. These objectives are conflicting, which means that there is not just one optimal solution. As such, the stack problem is suitable to be solved with algorithms from the MOEA framework.

The starting point for creating the set of bugs to be used is MOMoT, as it uses the MOEA framework in the background, but combines the functionality of this framework with functionality from the Eclipse Modeling Framework\(^1\) (EMF) and Henshin\(^2\) (Arendt et al., 2010). The goal is to show as much of the aspects of MOMoT as possible, by inserting bugs in various, strategic parts of the implementation. The bugs are always presented to the subjects in the same order, to enable comparison of the results for the individual bugs. The inserted bugs and the order in which they are presented are:

- Wrong number of algorithm runs: 300 runs instead of 30 runs
- Wrong number of initial stacks: 6 stacks instead of 5 stacks
- Wrong algorithm used: Epsilon MOEA instead of NSGA II
- Wrong load shift: shift loads from a stack to the same stack, instead of to a neighbor stack
- Wrong goal for one of the objectives: maximize instead of minimize the number of moves

The number of algorithm runs, the algorithm used in the search and the objective of the search are all set in the MOMoT search configuration file. The number of initial stacks is changed in the instantiation of the class model, while the load shift is adjusted in the model transformations. Together with the class model itself, these three parts represent the most important differences in the definition of a search between MOMoT and the MOEA

\(^1\)http://www.eclipse.org/modeling/
\(^2\)http://www.eclipse.org/henshin/

---

An evaluation of MOMoT 15
framework. Therefore, the bugs have been inserted in those parts. Inserting a bug in the class model has also been considered. However, it is almost impossible to insert a relevant bug in the class model while leaving everything else the same, and at the same time making sure that no runtime errors occur. Moreover, a bug in the class model is inevitably much more complex than the bugs proposed in the list above, and thus not suitable for an audience consisting of students who only had a very limited introduction to the tools. Therefore, the choice was made to focus on the other important parts of MOMoT.

**Guidelines** This research is performed with students who participate on a voluntary basis, and thus can not be expected to invest more than just a few hours of their time. If more time would be asked from the students, participation will most certainly drop as there is no reward for the extra time. Of course, the introduction to the techniques can be considered valuable, but a longer introduction does not add much value anymore. Therefore, the choice was made to invite students for a session that takes between two and three hours, depending on the time it takes to fix the bugs.

Every subject will only be exposed to one treatment. Using the list of subscriptions, the subjects are randomly divided over the treatment combinations. Before the subjects can start finding and fixing bugs, they will first get an introduction to MOMoT and MOEA. This will be limited to maximum one hour: both topics will receive about half of the time, to make sure that neither method benefits unequally from this introduction.

After the introduction, each subject is assigned to a computer, on which the correct treatment is loaded. For every treatment, there will be a case description, including specific information on how the results should look. The subjects read this, after which they can run the tool including the bug. Every treatment will consist of multiple versions of the same implementation, each with their own bug. This way, the time it takes to find and fix every bug can be recorded separately.

**Measurement instruments** The measurement instruments necessary in this experiment are a tool for recording the time that each subject takes to find and fix each bug, and a tool to record the number of fixed bugs. An option for recording the time is to install a stopwatch, with lap recording function, on each computer. This way, subjects can record the time themselves. However, this is not a very reliable method. It would be better to include keep track of the time automatically, by saving the time whenever a correct run is performed. The test functionality available in Java could be used for this. An advantage of this option it that the Java test functionality can also keep track of the number of fixed bugs. If this option is chosen, it is important to make sure that the test files are either not visible, or do not provide hints to where the bug is located. Another problem is that, in this setting, subjects will need to use testing functionality, which changes the experience of use. Therefore, the choice is made to record the interaction of subjects with the tools in a screencast. These screencasts will be analyzed afterwards, to determine the time it took to fix the bugs and the number of bugs that were fixed. Moreover, these screencasts contain much more information, and thus can be used to analyze other aspects in the future.

**Validity evaluation** This section is aimed at identifying and, if possible, preventing threats to validity before they occur. It might look like this is the first time that the validity of the experiment results is considered. The opposite is true: almost every design choice that is included in this document, has been made with the aim of maximizing validity. However, good validity is not an optimization problem with just one objective. Validity consist of several aspects, and good validity is only achieved if the results score sufficiently on all aspects. Yet, an increase on one aspect can lead to a decrease on another. In that sense, maximizing validity is a multi-objective optimization problem, hence implying that there is not just one optimal solution.

Before presenting how maximum validity is to be achieved for this experiment, the different types of validity are introduced. These are taken from Cook and Campbell (1979). They distinguish the following four types/aspects of validity:
• Conclusion validity: to what extent are we able to draw a correct conclusion regarding the relation between the factor(s) and outcome of the experiment?
• Internal validity: to what extent are we sure that the relation between treatment and outcome is actually a causal relation, and not a result of another independent variable that we missed or over which we have no control?
• Construct validity: to what extent do the factor(s) and the outcome reflect the theoretical cause and effect constructs that were used when defining the experiment?
• External validity: to what extent can the experiment results be generalized?

Cook and Campbell (1979) provide guidance in balancing the different types of validity, and Wohlin et al. (2000) adopt this view. An order of importance of the different types is proposed. It is based on the purpose of the experiment. For this experiment, which fits in the category of applied research, internal validity is considered most important, followed by external validity, then construct validity and lastly conclusion validity.

Wohlin et al. (2000) provide a checklist with a number of possible threats to validity, specific for every type. This makes it very convenient to check to which threats this experiment might be vulnerable. Here, only the relevant threats that are not dealt with, implicitly or explicitly, in other parts of this document, are mentioned.

A possible threat to internal validity of the experiment is the occurrence of compensatory rivalry or its opposite, resentful demoralization. Compared to the terms themselves, the mechanism that is behind them is relatively straightforward. Subjects receiving a treatment that is considered less desirable, can be motivated to show that this treatment is not so undesirable as it looks: this is known as compensatory rivalry. Resentful demoralization occurs when these same subjects are instead discouraged by the undesirable treatment and thus actively try to make it look worse than other treatments.

MOMoT –the ‘new’ method –is compared to the native encoding in the MOEA framework –the ‘old’ method. Hence, this experiment contains a situation in which either threat might occur. As MOMoT builds on the MOEA framework, it is easy to consider it less desirable to work with the latter. This effect can be amplified by the characteristics of the methods. The students involved are to a certain extent used to the plain Java code that the MOEA framework uses, while the models used in MOMoT are new to them. As a result, participants working with the code-based method (the MOEA framework) could for example slow down their bug fixing progress. The difficulty with these threats is that they are almost impossible to measure. Therefore, it is important to prevent them from occurring. A logical way to do this is to be as objective as possible in the presentation of the two methods, meaning that no method is favored over the other when introducing them to the subjects.

In the section on context selection, the conclusion is drawn that this experiment is fairly specific. This means that great care should be taken when generalizing the results. As external validity is limited, the exact conditions of the research should be clearly stated when conclusions are drawn based on the results.

Ideally, the subjects in an experiment know just what they need to know to complete their task, thereby minimizing the risk that any information they have will influence the results. However, the subjects in this experiment are all volunteers, and they must be given some information in order to convince them to participate. Based on this information, they could attempt to guess the goal and the expected outcome of the experiment, and change their behavior as a result of this. This phenomenon is known as hypothesis guessing, and it poses a threat to construct validity. This threat is closely related to compensatory rivalry and resentful demoralization. To mitigate this threat as much as possible, it is once again important to be objective when introducing the experiment to the subjects.

A possible threat regarding conclusion validity is related to the notion of statistical power. Participation in this research is voluntary. If participation is not sufficiently high, hypothesis testing might not provide reliable results. Convincing students to participate should therefore have high priority. Various means of communication should be used to reach as many students as possible.
Furthermore, offering an inducement to the participants should be considered.

**Perceived efficacy** As opposed to the metrics for actual efficacy, the measures for perceived efficacy do not have to be defined for the particular context of this research. Instead, the measurement scale proposed by Davis (1989) can be used. Hence, perceived usefulness and perceived ease of use are measured by means of a survey. This is in line with the metrics used by Moody (2003). In the guidelines for the experiment that measures actual efficacy it is stated that students are invited for a session of maximum three hours. Davis et al. (1989) have shown that such a brief introduction (they gave participants a one-hour introduction) to a technique is sufficient to evaluate usefulness and ease of use successfully. In the section on actual efficacy, many aspects are covered that are also relevant for the measurement of perceived efficacy. The context in which data will be collected is described. Furthermore, the selection of subjects is incorporated. A significant part of the instrumentation and validity evaluation are also of importance here. Additionally, TAM has been discussed extensively in Section 2.1.1. As such, this section is kept short, only covering aspects that have not been covered in other parts of this thesis.

**Hypothesis formulation** Similar to actual efficacy, a hypothesis is formulated for the expected perceived efficacy. Again, it is hypothesized that MOMoT performs better than the MOEA framework, based on the existing evaluation of MOMoT:

\[ H_0: \text{There is no difference in perceived efficacy when fixing bugs in MOMoT, compared to a native encoding of the same problem in the MOEA framework} \]

\[ H_1: \text{Actual efficacy is higher when fixing bugs in MOMoT, compared to a native encoding of the same problem in the MOEA framework} \]

The results can be used to determine whether the null hypothesis can be rejected.

**Evaluation questions** Perceived usefulness and perceived ease of use are measured by asking the users of a tool a set of evaluation questions. In this case the users are the participants in the research. The evaluation questions have been defined but have to be partially adjusted to make them suitable for use in this research. In particular, the questions regarding usefulness need to be reformulated.

The original set of questions for TAM has been proposed and verified by Davis (1989). It is straightforward that the adjusted questions should be as close as possible to the original questions, as the latter is the only validated set of questions that is available. In line with Davis (1989) answers to the questions are in the format of a 7-point Likert scale. The full set of questions that is used in this research can be found in Appendix A. For completeness, an example of adjusting a question is included here. One of the original questions is the following:

\text{Using [technology X] in my job would enable me to accomplish tasks more quickly}

As can be seen, this question explicitly requires the respondent to use the tool in a professional setting. In the following formulation, this is not the case, while at the same time, preserving the content of the original question as much as possible:

\text{I think that using [technology X] in a job setting would enable employees to accomplish tasks more quickly}

**Instrumentation** Instrumentation has already been covered in the section on actual efficacy. Some additional guidelines and measurement instruments need to be discussed with respect to perceived efficacy.
**Guidelines** The first hour of a session is devoted to introducing the participants to the techniques by means of a presentation. After that, they apply the techniques on the example case introduced in the previous section. When all bugs have been fixed, or when the time runs out (maximum two hours for all the bugs) the participant is asked to fill in the questionnaire regarding perceived efficacy.

For working with the tools, a virtual machine on a remote server is used. This not only ensures that all participants work with the same hardware, it also means that the environment in which the tools are used can be controlled as much as possible. Furthermore, the tools and screencast software can be installed beforehand, thereby saving valuable time during the session.

**Measurement instruments** In general, there are two options to provide the evaluation questionnaire to the participants of the research: on paper, or digitally. The choice is made to provide it digitally, as eventually the results are needed in digital form to insert them in this thesis. By providing the questionnaire digitally, valuable time is saved. Google Forms\(^3\) is used, as it is easy to use, reliable, and offers the possibility to export the results directly to an Excel sheet. Two questionnaires are created: one for MOMoT, and one for the MOEA framework. During the sessions, the participants will be provided with the link to the correct questionnaire.

### 2.4 Summary

The methodology of this research is summarized in Figure 10. On the abstract level, TAM provided the inspiration for the goal of the research. As such, attitudes towards use are the key concept on that level. The goal is operationalized with questions regarding actual efficacy and perceived efficacy. Perceived efficacy originates from TAM, and it consists of perceived ease of use and perceived usefulness. MEM adds actual efficacy to that, which consists of actual efficiency and actual effectiveness. The metrics for actual efficacy are context-specific, and measurement is done by performing an experiment. The metrics for perceived efficacy are adopted from TAM, and are measured by means of a survey.

---

\(^3\)https://www.google.com/forms/about/

An evaluation of MOMoT
Figure 10: Graphical summary of the methodology
3 Results

The results section follows the order in which the data has been gathered during the research. This means a summary of the students who participated is presented first, followed by the results for actual efficacy, and finally the results of perceived efficacy.

3.1 Participants

During the last lecture of the Algorithmic Programming for Operations Management course, a presentation was given to introduce the research to the students. Following this presentation, students were given the opportunity to subscribe directly, or at a later moment via a Google Form. For those who missed the lecture, a video presentation was prepared, which was sent to all the students who were registered for the course. This way, as much attention as possible was drawn to the research.

Next to their name, email address, gender, Bachelor, Master (if applicable) and average grade, questions about programming skills and availability were included in the registration form. Regarding programming skills, five possible answers were included:

- Novice: never programmed
- Beginner: limited programming experience in one programming language
- Moderate: high programming experience in one language, limited experience in some other languages
- Expert: high programming experience in multiple languages

Based on the course content, it was expected that students would place themselves in the ‘Beginner’ category. With respect to availability, several time slots were suggested, of which the students could select multiple. These time slots were chosen strategically: on the end of the exam weeks, and during the beginning of the new quartile, as this is normally the least busy period for students.

Students were induced to participate by offering them lunch or beer, depending on the timeslot on which a session took place.

A total of seven students participated in the evaluation. Three of them evaluated the code-based method (the MOEA framework), while the other four evaluated the model-based method (MOMoT). This means that the experiment is not balanced. The data of these participants can be found in Tables 1 and 2.

Although it was expected that the students would mainly be from the Industrial Engineering Bachelor, it turned out that this was not always the case. Several students were already in their Master and had not followed the Industrial Engineering Bachelor at TU/e. The Bachelors ranged from Economics to Mechanical Engineering. However, all participants who did not follow an Industrial Engineering Bachelor, were in the Operations Management & Logistics Master at TU/e, or in a similar program at another university.

Except for one participant, everybody judged oneself as a beginning programmer. As there was only one participant indicating to have moderate programming experience, the only option was to apply randomization. Blocking would not have made a difference, as this requires two moderate programmers, which could be equally spread over the treatments.

This same person indicated 9 as average grade, while all other participants filled in 7. Again, it was not possible to apply blocking to take this into account. Therefore, randomization was used.

Two participants were females, the rest were males. This means that the females could be spread over the treatments, which has been done: one of them evaluated MOMoT, the other the MOEA framework.

The sessions took place on three days. This was the minimal number of days required to make it fit with all the students’ schedules.
Table 1: Participant data, part 1

<table>
<thead>
<tr>
<th>Evaluated technique</th>
<th>Gender</th>
<th>Bachelor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOMoT</td>
<td>Male</td>
<td>Economics @ UvA</td>
</tr>
<tr>
<td>MOMoT</td>
<td>Male</td>
<td>Industrial Engineering @ Anna University, Chennai, India</td>
</tr>
<tr>
<td>MOEA framework</td>
<td>Female</td>
<td>Industrial Engineering @ TU/e</td>
</tr>
<tr>
<td>MOMoT</td>
<td>Male</td>
<td>Industrial Engineering @ Avans Tilburg</td>
</tr>
<tr>
<td>MOMoT</td>
<td>Female</td>
<td>Logistiek &amp; economie @ NHTV</td>
</tr>
<tr>
<td>MOEA framework</td>
<td>Male</td>
<td>Mechanical Engineering @ Applied University of Utrecht</td>
</tr>
</tbody>
</table>

Table 2: Participant data, part 2

<table>
<thead>
<tr>
<th>Master</th>
<th>Average grade</th>
<th>Programming skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations Management &amp; Logistics @ TU/e</td>
<td>7</td>
<td>Beginner</td>
</tr>
<tr>
<td>Supply chain management @ KLU Hamburg</td>
<td>9</td>
<td>Moderate</td>
</tr>
<tr>
<td>I am still in my Bachelor</td>
<td>7</td>
<td>Beginner</td>
</tr>
<tr>
<td>Operations Management &amp; Logistics @ TU/e</td>
<td>7</td>
<td>Beginner</td>
</tr>
<tr>
<td>Operations Management &amp; Logistics @ TU/e</td>
<td>7</td>
<td>Beginner</td>
</tr>
<tr>
<td>I am still in my Bachelor</td>
<td>7</td>
<td>Beginner</td>
</tr>
<tr>
<td>Operations Management &amp; Logistics @ TU/e</td>
<td>7</td>
<td>Beginner</td>
</tr>
</tbody>
</table>

3.2 Results for actual efficacy

This section discusses the results for actual efficacy, starting with the results for actual efficiency which are shown in Table 3. Empty cells indicate that the bug was not found, either because the participant gave up or had to leave, or because the session ended. In Figure 11, the results for bugs 1, 2 and 3 are presented as boxplots. For bugs 4 and 5, this does not add any value, as for these bugs not enough data is available to create meaningful boxplots. Although neither of the tools enabled faster bug fixing for all bugs, the boxplots show that the times do not overlap for any of the bugs. This indicates that a significant difference might exist.

In Table 4, the results of statistical tests on the bug fixing times of bugs 1, 2 and 3 are shown. Statistically significant results are underlined. The student’s t-test is performed to assess whether the mean times it takes to fix each of the bugs differs significantly between implementations in MOMoT and the MOEA framework. This test assumes a normal distribution of the test data. To verify that this assumption holds, a Shapiro-Wilk test is performed. This test did not provide evidence that the data does not follow a normal distribution (α = 0.05). The student’s t-test revealed that it takes significantly longer to find and fix the second bug in the MOMoT implementation (α = 0.05).

It should be mentioned that the null hypothesis of the Shapiro-Wilk test is that normality holds for the given data. The fact that this hypothesis was not rejected, does not necessarily mean that the data actually follows a normal distribution. It is also possible that the amount of data simply does not provide enough evidence to reject the null hypothesis. As there are only three observations per bug for the MOEA framework, and four for MOMoT, there is reason to believe that normality only holds because of a lack of data. Therefore, a non-parametric alternative to the student’s t-test is used: the Mann-Whitney test. This test does not assume normality, and thus does not rely on the result of the Shapiro-Wilk test. In exchange, the power of this test is lower compared to the student’s t-test. This has an effect on the results: no significant differences are found when using the Mann-Whitney test (α = 0.05).

The results for actual effectiveness are shown in Table 5. As mentioned in the methodology, the used metric is defined as the number of bugs found in a specified period of time. The cut-off point
Table 3: Bug fixing times

<table>
<thead>
<tr>
<th>Bug</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOMoT</td>
<td>1:03</td>
<td>43:39</td>
<td>15:00</td>
<td>30:45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2:43</td>
<td>40:20</td>
<td>11:37</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1:48</td>
<td>20:45</td>
<td>33:27</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3:51</td>
<td>25:31</td>
<td>8:05</td>
<td>28:10</td>
<td></td>
</tr>
<tr>
<td>MOEA</td>
<td>26:34</td>
<td>6:55</td>
<td>4:29</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>18:17</td>
<td>19:18</td>
<td>0:34</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8:44</td>
<td>1:49</td>
<td>5:19</td>
<td>55:03</td>
<td></td>
</tr>
</tbody>
</table>

Figure 11: Boxplots for the bug fixing times of bugs 1, 2 and 3

Table 4: p-values of statistical tests on the bug fixing times

<table>
<thead>
<tr>
<th>Bug</th>
<th>Shapiro-Wilk test</th>
<th>Student’s t-test</th>
<th>Mann-Whitney test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MOMoT</td>
<td>MOEA</td>
<td>MOMoT</td>
</tr>
<tr>
<td>1</td>
<td>0.9331</td>
<td>0.9217</td>
<td>0.09297</td>
</tr>
<tr>
<td>2</td>
<td>0.3809</td>
<td>0.5492</td>
<td><strong>0.0291</strong></td>
</tr>
<tr>
<td>3</td>
<td>0.2117</td>
<td>0.3152</td>
<td>0.09227</td>
</tr>
</tbody>
</table>

That has been used is 75 minutes. Shortly after that, participants started quitting, either because they gave up or because they had to leave. Therefore, the decision was made to stop counting at that point.

It should be noted that the number of bugs found within 75 minutes does not necessarily match the data in Table 3. For example, all the bug fixing times on the fourth row summed results in a number lower than 75, suggesting that four bugs were found, while Table 5 states that only three bugs were found. This is caused by the way that bug fixing times were measured. In the screencasts, bug fixing time was measured from the point in time on which a participant ran the implementation with the bug for the first time, until the moment that the fix for the bug was applied. These actions were performed by all the participants for any of the bugs they fixed, and as such these are suitable measurement points. As a result, there is time left between two consecutive measurements. During this time, a participant was waiting for the results of running
a bug fix, waiting for the researcher to confirm that the bug was fixed correctly, and starting the implementation with the next bug. To measure the number of bugs found within 75 minutes, the screen casts were viewed at that exact point in time. It was checked which bug the participants were working on. As all participants worked on the bugs in the same order, it was straightforward to measure how many bugs had been fixed up to that point.

In Table 6, the results of statistical test performed on the number of bugs found are presented. Statistically significant results are underlined. The student’s t-test did not reveal a statistically significant difference between the number of bugs found ($\alpha = 0.05$). Moreover, applying the Shapiro-Wilk test showed that normality is rejected for both samples ($\alpha = 0.05$). However, this is most certainly due to the low amount of data. If, for example, one of the participants evaluating MOMoT would have found four bugs instead of three, normality would not have been rejected. Nevertheless, a Mann-Whitney test is also performed for the number of bugs found. This did also not reveal a statistically significant difference between the number of bugs found in MOMoT and the MOEA framework ($\alpha = 0.05$).

Table 5: Number of bugs found within 75 minutes

<table>
<thead>
<tr>
<th>Tool</th>
<th>Number of bugs found</th>
</tr>
</thead>
</table>
| MOMoT | 2  
|       | 3  
|       | 3  
|       | 3  |
| MOEA  | 3  
|       | 3  
|       | 4  |

Table 6: p-values of statistical tests on the number of bugs found

<table>
<thead>
<tr>
<th>Shapiro-Wilk test</th>
<th>Student’s t-test</th>
<th>Mann-Whitney test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOMoT</td>
<td>MOEA</td>
<td></td>
</tr>
<tr>
<td>0.001241</td>
<td>$&lt;2.2e-16$</td>
<td>0.2336</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2703</td>
</tr>
</tbody>
</table>

3.3 Results for perceived efficacy

The results for both constructs of perceived efficacy are shown in Figure 12. The results are presented in diverging stacked bar charts, as these have been shown to be the best method for presenting Likert scale results (Robbins and Heiberger, 2011). As mentioned before, a 7-point Likert scale is used. It is taken from Davis (1989). The following seven possible answers were included:

- Extremely unlikely (EU)
- Quite unlikely (QU)
- Slightly unlikely (SU)
- Not likely, not unlikely (NLNU)
- Slightly likely (SL)
- Quite likely (QL)
- Extremely likely (EL)
Behind each possible answer, its abbreviation used in Figure 12 is shown. The diverging stacked bar charts shown in Figure 12 contain counts for each possible answer, for all the questions. On the horizontal axis, the count is shown. On the vertical axis, the questions are shown in the same order as in Appendix A. Each bar starts from the middle, with the count of ‘Not likely, not unlikely’ in grey. Left of the middle, the counts of every degree of ‘unlikely’ are shown. The more red, the more unlikely. Similarly, the counts of every degree of ‘likely’ are shown right of the middle.

Figure 12 suggests that MOMoT and the MOEA framework score better on perceived usefulness than on perceived ease of use. Furthermore, MOMoT seems to be evaluated better than the MOEA framework on both aspects. According to TAM, this indicates that users have a better attitude towards the use of MOMoT. In the end, this means that MOMoT is more likely to be used in practice than the MOEA framework. Of course, the low amount of available data should again be taken into account. It is not unlikely that, with more data, other conclusions would be drawn.

Statistical analysis of the perceived efficacy data is not included. The amount of results is considered too low for this. The results of statistical analysis would most certainly be inconclusive, and as such would not provide insights that cannot be derived from Figure 12. When repeating this pilot study on a larger scale, statistical analysis should be performed. Based on the results, the hypothesis regarding perceived efficacy defined in the methodology section can then be tested. Similar to the actual efficacy data, relevant statistical tests for comparing Likert scale data are the student’s t-test and the Mann-Whitney test (De Winter and Dodou, 2010). These tests need numerical input, meaning that the answers to the questionnaire should be transformed to numerical values. This is done as follows:

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely, not unlikely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

The linear scale described here is taken from De Winter and Dodou (2010). The remainder of the analysis is exactly the same as for the actual efficacy. A statistical tool should be used to perform the student’s t-test and the Mann-Whitney test. In this research, R has been used. A link to the scripts used in this research can be found in Appendix C.

In summary, the model-based method (MOMoT) seems to perform better than the code-based method (the MOEA framework) with respect to both constructs of perceived efficacy. This slight preference for MOMoT is not confirmed by the results for actual efficacy. Two out of three bugs took more time to be found and fixed in MOMoT than they took in the MOEA framework implementation. In a time period of 75 minutes, more bugs seem to be found in the MOEA framework implementation. Therefore, the data does not suggest a clear preference for either one of the techniques.
Figure 12: Results for perceived efficacy
4 Discussion

Both the results and the research as a whole provide many new insights. The value of these insights is discussed here. The results are summarized, relevant threats to their validity are examined, and interesting follow-ups are outlined.

The results for perceived efficacy suggest that the expectation that MOMoT would be evaluated better by Industrial Engineering students was correct. On both perceived usefulness and perceived ease of use, MOMoT scores slightly better, suggesting that Industrial Engineering students prefer a model-driven optimization technique over a traditional technique. It should be noted that, even though MOMoT is evaluated relatively better, both MOMoT and the MOEA framework do not give rise to extreme opinions. This could be due to the lack of extreme opinions about the tools. Another explanation could be that students in general tend to have less extreme opinions towards a technique than a professional would have (Moody, 2003).

For two of the three bugs that were fixed by all participants, the results suggest that the MOEA framework is the preferred tool with respect to corrective maintenance. Minor proof was found that the second bug is on average found significantly faster in the MOEA framework implementation. For the other bugs, the differences were not significant. Regarding the number of bugs found in a certain period of time, the results also suggest that the MOEA framework performs slightly better. However, this suggestion can also not be confirmed statistically. Altogether, the hypothesis that actual efficacy is higher when using MOMoT instead of the MOEA framework, could not be statistically confirmed by the results of the experiment. Moreover, the results for actual efficacy suggest a slight preference for the MOEA framework, thereby contradicting the results found for perceived efficacy.

4.1 Threats to validity

Great care was taken in defining and performing this research. In an early stage, possible threats to validity were made explicit. Implicitly, result validity has been taken into account throughout the research. Nevertheless, some threats to the validity of the results are unavoidable. These threats are identified here. As stated in Section 2.3.3, Cook, Campbell and Day (1979) identify four types of validity. Moreover, they prioritize the types for different areas of research. It was determined that internal validity is most important for this research, followed by external validity, construct validity and conclusion validity. This section is structured according to this order.

4.1.1 Internal validity

It is possible that the implementations of the case that have been used, do not completely reflect a typical implementation of a case in either MOMoT or the MOEA framework. This is a threat to internal validity. As was mentioned before, the implementations have been made by Martin Fleck, who uses them in his evaluation of MOMoT. In particular, the MOMoT implementation is used for an evaluation of the applicability, overhead and search features. The MOEA framework implementation is solely used to analyze the overhead that MOMoT introduces in comparison to a case that is implemented in the MOEA framework directly. As such, this implementation does not necessarily need to adhere to any coding conventions. As long as it generates correct output, and does not do this significantly slower than a typical MOEA framework implementation of the same case, it can be considered sufficiently good.

To determine to what extent this threat is relevant, both implementations were analyzed before using them in the research. For the MOMoT implementation, it is reasonable to assume that it reflects a typical implementation in MOMoT fairly well, because it has been created by the developer of MOMoT. The MOEA framework implementation was analyzed more thoroughly. A guide for writing good Java code was used (Perry, 2016). This guide was found on the IBM developerWorks® platform, which provides tutorials for Java and many other technologies. The
metrics provided in the guide were applied on the MOEA framework implementation. The following best practices are mentioned by Perry (2016):

- Keep classes small: focus on doing a small number of things, doing those well
- Name methods carefully: a good name reveals the intention of the method
- Keep methods small: limit each method to a single job
- Use comments: comments help others, but also yourself, to understand your code better
- Use a consistent style: preferably use a coding standard, but in any case, be consistent
- Use built-in logging: prevent using print statements to record relevant actions of a program

Regarding the size of the classes and methods and consistency, the MOEA framework implementation provided by Fleck was compared to a similar implementation. Specifically, it was compared to the implementation of the knapsack problem, which is one of the examples provided by the developers of the MOEA framework. Both implementations consist of two classes: one describing the problem, and one configuring the search. The length of both sets of classes is similar. Moreover, the structure is the same, and thus the MOEA framework implementation provided by Fleck is consistent with the style used by the developers of the MOEA framework. The same holds for the methods used by Fleck. Furthermore, the naming of methods and variables is in line with the naming conventions used by the developers of the MOEA framework. Problem-specific methods and variables have names that clearly show their intent, as suggested by Perry (2016). Logging is mainly relevant in the development of an implementation, and as such it is not of importance here. The only negative note regarding the MOEA framework implementation provided by Fleck is the lack of comments. However, the implementation in MOMoT also lacks comments. Although this might not be the best practice in terms of coding, it should not affect one implementation differently than the other.

In all the communication to the students about the research the aim has been to be as objective as possible. The main reason for this has been provided in Section 2.3.3: by presenting the tools in an objective manner, an unjustified preference towards either one of the tools is prevented. However, when describing MOMoT it is impossible not to mention that it has been built on top of the MOEA framework, combined with two other existing frameworks. As a result, it is easy to think that MOMoT provides the same functionality as the MOEA framework, but is better due to the addition of functionality from other frameworks. This can lead to an unjustified preference towards MOMoT, which is known as resentful demoralization (Wohlin et al., 2000). Such a preference can influence the results for perceived efficacy. As stated above, the results for perceived efficacy show a slight preference for MOMoT. As promoted by MEM, actual efficacy is measured next to perceived efficacy. The results for actual efficacy suggest a preference for the MOEA framework. Hence, it could be the case that the preference for MOMoT suggested by the results for perceived efficacy is an unjustified preference. Although it is hard to define when a subjective preference for a tool is unjustified, the contradiction in the results shows how the measurement of two types of efficacy can lead to a better insight into attitudes towards the use of a tool.

This threat to internal validity is relevant for any comparison based on the TAM constructs, as the subjective judgment that TAM requires as input can always be distorted by unjustified preferences. To that end, MEM is not only applicable to this specific case. It can provide valuable insights in any context in which the use of a new tool is considered. Perceived efficacy assesses the likelihood of the tool being accepted by its intended users, while actual efficacy indicates whether it is desirable to use the new tool. The results of this research are a good example of the power of MEM.

The bugs introduced were selected with a clear goal: covering as much of the functionality of MOMoT as possible. Still, there is the risk that the process of finding and fixing those bugs does not sufficiently reflect the tools’ functionality, posing a threat to internal validity. In this specific case, none of the participants were able to find and fix all the bugs. This means that none of

---

4 http://moeaframework.org/
them have seen all the bugs, and thus all the aspects covered by the bugs. Out of five bugs, three were always found. After that, the majority of the participants got stuck when trying to fix the fourth. If all participants would have fixed all bugs, their impression of the tools might have been more equal. However, it is hard to estimate how long finding and fixing a bug costs at the time of inserting it. In case of repeating this research, it should be considered to make small changes to the bugs, with the aim of making them easier to fix. In that case, the discussion of possible bugs included in Section 2.3.3 should be taken into account.

4.1.2 External validity

Regarding external validity, it is important to highlight the point that was made earlier in this thesis. This research is carried out in an artificial setting with a toy problem, using students as participants. As such, it is rather specific, meaning that its results cannot be widely generalized. For that reason, this research aims explicitly at decision makers in the academic community, in particular those in the field of Industrial Engineering. Although a larger audience might be interested in the results of this research, the scope of this research has been kept modest, mainly because of the practical limitation that only limited time was available to perform this research.

4.1.3 Construct validity

The relevant threats to constructs validity have all been covered in Section 2.3.3.

4.1.4 Conclusion validity

An important threat to conclusion validity is the low number of participants in the research. Even though this is a pilot study, the number of participants is very low. As a result, not enough data is available to draw statistically significant conclusions, decreasing the conclusion validity. Because of this low number of participants, the choice was made to perform a non-balanced experiment. With seven participants and two treatments, it is obvious that a balanced experiment is not possible, but selecting six out of seven participants was considered worse, as every additional participant is valuable with such low numbers. The lesson learned is that voluntary participation poses a risk for validity of the results. Therefore, it is suggested to include participation in this research as a mandatory part of the following edition of the Algorithmic Programming for Operations Management course. A concrete description on how this should be done is provided in Appendix B. Another luxury that could not be afforded due to the low number of participants, is to select participants based on their availability. Random irrelevancies in the experiment setting are mentioned by Wohlin et al. (2000) as a possible threat to conclusion validity. Therefore, in the best case there would be only one session, in which all students are present. This way, these random irrelevancies would be equal for all the participants. However, again the inclusion of all available participants was preferred over minimizing this threat.

4.2 Follow-ups

Next to the results presented above, the process of performing this research led to many valuable insights about potential follow-ups. MEM states that actual efficiency is a determinant of perceived ease of use, and actual effectiveness is a determinant of perceived usefulness. However, Moody (2003) also admits that these are not the only determinants. It could therefore be interesting to specifically study the determinants of perceived usefulness and perceived ease of use. Such research could be valuable for MOMoT in particular, but also for TAM in general. When performing research in this direction, it should be
taken into account that there is already some literature available on this topic. An example can be found in the work of Venkatesh (2000), about the determinants of perceived ease of use. A small change in the research setting that could be interesting to consider, is to change the order in which the bugs are presented. In this research, the choice was made to always present the bugs in the same order. However, presenting them in a different order might change the experience of use significantly and thus might lead to different results.

Instead of changing the order of the bugs, it is also possible to change the bugs themselves as is suggested in Section 4.1. Such a change should not only aim at making the bugs easier to find. It should also intend to let the bugs reflect as much of the functionality of the tool as possible. In MOMoT, this can be done by inserting bugs in all the different parts: the model, the model instantiation, the model transformations and the search configuration. The bugs should not cause runtime errors. The search should run to completion and provide results. This provides the students with the opportunity to compare the output with correct output. This is a more interesting scenario than tracing the cause of a runtime error, because it requires the participants not only to understand the method and its syntax, it also requires them to understand the problem domain. As such, the extent to which a tool helps in creating an understanding of the problem is also assessed, and a more complete experience of use is offered.

This research applies quantitative methods to measure both types of efficacy. As actual efficacy relates to objective performance, quantitative methods are the only option. However, for measuring perceptions a wide range of qualitative methods is available, examples of which are interviews or focus groups. For TAM in particular, it has been argued that the use of qualitative methods should be considered (Vogelsang, Steinhüser and Hoppe, 2013).

In this study, the choice was made to aim for a sufficiently large participant group to be able to get convincing results from the quantitative methods. As it turned out, obtaining such a participant group is one of the most difficult things in a study where participation is voluntary. Although qualitative methods also require a certain number of participants, the results with with a small group are in general stronger than the results of quantitative methods in a similar setting, because more data can be collected from each participant. Therefore, an interesting alternative could be to repeat this research using qualitative research methods for the TAM the constructs.

This research is considered as a pilot study for the evaluation of the use of model-driven optimization techniques by Industrial Engineering students. For the future, the most important follow-up that is considered, is to repeat this research with a larger number of participants by including participation as a mandatory part of the Algorithmic Programming for Operations Management course at TU/e. A detailed description of how this should be done is presented in Appendix B. In the last edition of the course, almost 200 students were subscribed. Assuming that a similar number of students will subscribe next year, this would provide a sufficiently large pool of participants. The results of repeating this research on a large scale can provide very useful insights to decision makers in the academic community. In particular, it will give them the information necessary to decide on the possible inclusion of model-driven methods in Industrial Engineering curricula, and perhaps also in Software Engineering curricula. In this regard, future research should also take a broader scope. This study is limited to the assessment of Industrial Engineering students’ attitudes towards a model-driven optimization technique. This was considered as a proper scope for the time available to perform this research. However, academic decision makers’ concerns regarding model-based techniques might well range beyond that scope. MOMoT and the MOEA framework have in common that they are both optimization tools. An interesting question could be whether other categories of code-based tools are likely to profit from a model-based alternative. On the other hand, a comparison of different model-based methods could provide a set of requirements to define what makes a good model-based method. These are all fundamental questions, and answering them will clarify if it is desirable to include model-based methods in the curricula of the business analysts of the future.
5 Conclusions

The results of this research do not reveal a clear preference towards a model-based or a code-based optimization method. Participants’ perceptions are seemingly slightly more positive towards model-driven optimization techniques than they are towards traditional techniques. However, at the same time the results suggest that their performance is lower when using a model-based optimization technique. In that sense, the results are contradictory, and thus do not allow for a clear-cut view on Industrial Engineering attitudes towards the use of such techniques. None of the results could be supported by statistical evidence because of the low number of participants. However, this is a pilot study to assess the suitability of applying the proposed methodology. As such, statistically significant results would be valuable, but they are not crucial to the success of this research.

This research has shown that MEM can be successfully applied to evaluate Industrial Engineering students’ attitudes towards the use of a model-based and a code-based optimization technique. Context-specific metrics for actual efficacy have been defined. Combined with the measurement scales of TAM, a robust set of metrics is provided for the comparison of attitudes towards the use of MOMoT and the MOEA framework.

For the future, it is proposed that this research is repeated in the next edition of the Algorithmic Programming for Operations Management course at TU/e. Detailed instructions that intend to facilitate this process are provided in Appendix B. It is advised to include participation in the research as a mandatory part of the course. This will guarantee that the number of participants will be sufficiently high, implying that statistically significant conclusions can be drawn based on the results. Only then, a well-founded advice regarding the inclusion of model-based methods in academic education can be given to decision makers within the field of Industrial Engineering, and perhaps also within other fields.
Acknowledgments

This thesis is my last step towards the Master’s Degree in Business Information Systems. It presents the research that I have performed over the last half year. This research has been conducted within the Information Systems group of the Industrial Engineering & Innovation Sciences department of Eindhoven University of Technology, under the supervision of Pieter Van Gorp.

First and foremost, I would like to thank Pieter for the many useful comments he provided on my work, and for the many hours he spent on this project. He provided me with the topic for this research. His enthusiasm, support and knowledge on a large variety of fields repeatedly gave me the necessary inspiration for taking this research to the next level.

Feedback provided by Natalia Sidorova and Maryam Razavian yielded useful new views on my thesis. I would also like to thank both of them for being part of my graduation committee, together with Pieter.

When I ran into trouble with MOMoT, I could always contact Martin Fleck. He would help me to get back on track again, which I greatly appreciate.

An important group of people who deserve to be mentioned here, are the volunteers who participated in this research. Their participation enabled me to apply the proposed methodology.

Last but not least, I would like to thank all my friends and family for the implicit and explicit support they provided during the process of performing this research and writing this thesis.
References

Appendix A: Evaluation questions for the Technology Acceptance Model

**Perceived usefulness**

- I think that using X in a job setting would enable employees to accomplish tasks more quickly

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

- I think that using X in a job setting would increase performance of employees

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

- I think that using X in a job setting would increase productivity of employees

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

- I think that using X in a job setting would improve effectiveness of employees

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

- I think that using X in a job setting would make the work easier

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

- I think that using X in a job setting would be useful

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>
Perceived ease of use

- Learning to work with X would be easy for me

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely, not unlikely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- I would find it easy to get X to do what I want it to do

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely, not unlikely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- My interaction with X would be clear and understandable

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely, not unlikely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- I would find X to be flexible to interact with

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely, not unlikely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- It would be easy for me to become skillful at using X

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely, not unlikely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- I would find X easy to use

<table>
<thead>
<tr>
<th>Extremely unlikely</th>
<th>Quite unlikely</th>
<th>Slightly unlikely</th>
<th>Not likely, not unlikely</th>
<th>Slightly likely</th>
<th>Quite likely</th>
<th>Extremely likely</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B: Guidelines for performing this research in a course context

The research described in this thesis has been performed with students who participated on a voluntary basis. It has been considered to include participation as a part of the TU/e course Algorithmic Programming for Operations Management, as a mandatory part or as a bonus assignment. As this research was proposed when the course already started, this was not possible in this academic year. To turn the results of this pilot study into a set of results on which statistically significant conclusions can be drawn, it is suggested to include participation in this research as a mandatory part in the next edition of the course. All the necessary information to make this possible can be found in this appendix. Although this section focuses specifically on the Algorithmic Programming for Operations Management course taught on TU/e, the research can be carried out as part of similar courses in other universities as well.

Steps to be taken

The following steps are part of the execution of the research. These steps are described in detail in the remainder of this appendix:

- Preparation of the session(s)
- Running the session(s)
- Analysis and conclusions

Step 1: Preparation of the session(s)

The session(s) should be prepared carefully. This will minimize the occurrence of threats to validity.

**Presenting the research**  As participation in this research will be a mandatory part of the course, it is sufficient to mention the date on which the session will take place. A short introduction to the research is possible, but not necessary. Moreover, not introducing the subjects to the research before a session ensures the absence of bias towards one of the techniques. In the study guide, the relevance of this research in the context of the course has to be clarified.

**Planning the session(s)**  In this research, students from the TU/e course Algorithmic Programming for Operations Management are the participants. As this is an introductory Java course, it is best to plan the session(s) towards the end of the course, as working with the tools, and with the MOEA framework in particular, requires at the very least some basic programming skills. As the course assessment does not include a written exam, it is possible to reserve a 3-hour time slot during the exam weeks. During this time slot, either one big session with all the students on one location can be organized. From the point of view of the experiment, this is preferred. However, is doubtful whether this is possible in practice. Instead, multiple sessions could be organized at different locations during this time slot. If sessions during the exam week are considered unsuitable, the sessions could instead be organized during regular lecture hours of the course, preferably during the last instruction session.

**Case implementations**  Currently, one implementation of a case in both MOMoT and the MOEA framework has been verified to be suitable for use in this research. The most straightforward way to repeat this research in a course context is to use that case, with the same bugs. It is available online, in a virtual machine, which means that the effort of setting up the implementations can be saved. Another advantage of using this platform, is that the tools do not have to be

---

5http://is.ieis.tue.nl/staff/pvgorp/share/?page=Home
"ICMT 2016::Ubuntu12LTS_MOMoT_MOMoT_experiment.vdi"
installed on any other computers, thereby saving time during the session(s) or in the preparation.

**Introduction to the tools**  
The session(s) should start with an introduction to MOMoT and the MOEA framework, as the students are not familiar with these tools. It is most logical to explain the functionality of the MOEA framework first, after which MOMoT is introduced. In this research, this introduction was given as a mini-lecture, including examples of implementations of both tools. The slides of that mini-lecture are among the artifacts that belong to this thesis. A link to these artifacts is included in Appendix C.

**Measuring actual efficacy**  
In the sessions that were held for this research, the interaction of the students with the tools has been recorded in screencasts. From these screencasts, the time it takes to find and fix bugs and the number of bugs found were determined afterwards. However, this might not be the best option when performing the experiment with a larger group of participants, as the analysis of the screencasts is fairly time-consuming. Some other options were considered in Section 2.3.3, namely using the Java testing functionality or having the students keep the time and number of bugs found themselves. Out of these two, using the Java testing functionality might be a good option, but then again, setting this up takes considerable time too. As such, this method is worth considering with a large participant group. Setting up the screencasts can be done on the participants’ laptops, by the participants themselves, or it can be done on computers provided by the researcher. The latter option has the advantage that time is saved during the session, but the disadvantages that more time has to be spent on the preparation and that it is practically impossible to get a computer for every participant in the course. The screencast tool that has been used in this research is Screencast-O-Matic\(^6\), with a Site License\(^7\) for the Pro Recorder, as this allows for the use of the software on multiple computers.

**Evaluation questionnaire**  
The questions presented in Appendix B should be prepared for the students, one version for MOMoT and one for the MOEA framework. Note that the questions in Appendix B lack the names of the tools. A paper version of the questionnaire is possible, but a digital version saves a lot of work. The tool that has been used in this research is Google Forms\(^8\). PDF versions of the questionnaires used in this research are among the artifacts that belong to this thesis. A link to these artifacts is included in Appendix C.

**Step 2: Running the session(s)**

As an experiment is part of the research, a high degree of control should be maintained over the environment during the session(s). When multiple sessions are held, it is important to make sure that there are as little differences between them as possible. The overall structure has to be the same for all the sessions. It is advised to follow the structure presented in this thesis, meaning a session of maximum three hours, in which about one hour is devoted to an introduction to both tools, while during the remainder of the time students work with the tools. The participants work with the tools individually, and therefore it is best to create an exam-like environment.

**Step 3: Analysis and conclusions**

For the analysis of the results, the reader is referred to Section 3. The exact same structure can be used. The expectation is that, with more data, statistically significant results can and will be found.

---

\(^6\)https://screencast-o-matic.com/home
\(^7\)https://screencast-o-matic.com/gopro#prorecorder-sitelicense
\(^8\)https://www.google.com/forms/about/
Appendix C: Access to artifacts referred to in this thesis

In the thesis body several references to external artifacts are made. These include the video presentation and the email which were sent to the students, the screencasts made during the sessions with students, the complete results of the sessions in Excel sheets, the R scripts that were used for the statistical analysis, etcetera. All such artifacts have been collected in a Dropbox folder, which can be found by following the link below:
https://www.dropbox.com/sh/rgnckkkhttnzx8i/AADnECorwdxwV3qR7fEp65L4a?dl=0