University of Passau

Department of Informatics and Mathematics

Master’s Thesis

Visualization of Performance-Influence Models

Author:

Rima Celita Lewis

May 16, 2019

Advisors:

Dr.-Ing. Sven Apel
Chair of Software Engineering I

Christian Kaltenecker
Chair of Software Engineering I

Lewis, Rima Celita:
Visualization of Performance-Influence Models
Master's Thesis, University of Passau, 2019.

Abstract

Configurable software systems have various configuration options such as encryption and compression. These configuration options and their interactions may have an influence on the non-functional properties of the system like performance or energy consumption. Performance-influence models are used to interpret and understand the performance influences of certain configuration options or interactions on the non-functional properties of the system. However, the increasing complexity of performance-influence models for complex configurable software systems turns the interpretation into a difficult task. To overcome this issue, we present a tool to visualize the performance-influence models to make it easier to interpret them. For this thesis, we select 3 different visualization techniques. To demonstrate their usefulness, we validate these visualization techniques by performing an interview. In general, we find that one of the visualization techniques outperformed the others in interpreting performance-influence models.

Contents

List of Figures

List of Tables

1. Introduction

Modern software systems provide a multitude of configuration options. Configuration options define the functionality of a configurable software system. Configuration options and their interactions often have an influence on the non-functional properties of the system, such as performance or energy consumption. To identify the performance influence of certain configuration options or interactions on the non-functional properties of the system, we use performance-influence models [SGAK]. Users or developers need to analyze performance-influence models in order to find the relevant configuration option or interaction of the configurable software system. Performance-influence models can also be used to compare the influences of two different non-functional properties on the configurable software system. This aids to configure the software in a way that affects its performance positively. The performance-influence models can become quite complex as the set of configuration options and interactions that influence performance increases. The increasing complexity of performance-influence models makes analyzing them a difficult task. The solution we use in our thesis is to present 3 different visual representations of performance-influence models.

We present 3 different visualization techniques to analyze performance-influence models, (1) the radar plot, (2) the text plot, and (3) the ratio plot. The Radar and the text plot, present the actual performance of the configuration option or interaction, whereas the ratio plot presents the relative performance of the configuration option or interaction with respect to the total performance of the system. To assess the quality of these visualization techniques, we perform an interview. The interview consists of questions with a combination of performance-influence models with different levels of complexity and different visualization techniques. The results from the interview aid us to evaluate our research questions. We have selected the following research questions to evaluate our thesis.

RQ1: Can we use the visualization techniques to identify the relevant properties of one performance-influence model?

RQ2: Can we use the visualization techniques to compare two performance-influence models?

RQ3: How good can the visualizations be used to compare a high number of performance-influence models and a high number of terms?

RQ4: What are the differences when considering the scalability of many performance-influence models?

The research questions mentioned above are evaluated for different complexity levels of performance-influence models.

The structure of this thesis is as follows:

In Chapter 2, we introduce terms used throughout the thesis for a better understanding of the reader. We explain configurable software systems and its non-functional properties, performance-influence models and how we derive them. We also introduce different visualization techniques.

We present the research questions selected to validate this thesis in Chapter 3. We introduce the process of the interview and how it is designed to help us assess the research questions.

The answers obtained in the interview are presented and discussed in Chapter 4. We discuss our findings from the presented results and answer our research questions.

The work related to the visualization of performance-influence models is discussed in Chapter 5.

In Chapter 6, we present the conclusion of our thesis and a summary of our findings with respect to the research questions.

We present further improvements for the visualizations of the performance-influence models tool in Chapter 7. The improvements are based on the general feedback given by the interviewees in the interview process.

2. Background

In this chapter, we introduce the general terms and definitions that help the reader to understand the remainder of the work. In Section 2.1, we provide a detailed description and definition of the configurable software system along with their use cases. In Section 2.2, we describe the performance and how it is used in this thesis, followed by an example of a performance-influence model and how performance-influence models are derived. In Section 2.3, we describe different visualizations techniques for performance-influence models.

Section 2.1: Configurable Software System

A software or a system with different configuration options is called a configurable software system [ABKS13]. A configuration option offers a specific functionality to the system. For instance, a configuration option `compression` introduces a functionality to compress files. A configuration option could belong to an `alternative` group or an `or` group. The alternative group implies that a configuration option should be selected with exactly one of several algorithms available for that functionality. Whereas, or group implies that the configuration option should be selected with at least one of several algorithms available for that functionality. For instance, the configuration option compression has several algorithms like `AES, 3DES, SHA1`. Furthermore, configuration options can be optional or mandatory. Optional configuration options can be either selected or deselected, whereas mandatory options are always be selected. Configuration options that can be selected or deselected are called binary configuration options.

Optional configuration option also means that several of these configuration options can be selected at the same time, but oftenly not all combinations of the configuration options selected together are valid. The validity of the combinations depends on constraints. A constraint is a limitation on how the configuration options can be selected in combination with other configuration options. For instance, two configuration options can be mutually exclusive. When at least one of these constraints is not fulfilled, in a combination, the combination is invalid. The valid combinations of the configuration options is called a 'configuration'. The configuration options and the constraints among them are described in a variability model.

Formally, we denote $\mathcal{O}$ as the set of all configuration options and C as the set of all valid configurations. Any configuration c $\in C$ is a valid configuration, which is a function c: $\mathcal{O} \to \{0, 1\}$. That is, c(o) = 1, if the configuration option o $\in \mathcal{O}$ is selected in the configuration c $\in$ C and c(o) = 0 otherwise [SGAK].

Mandatory configuration options are the ones that are present in all valid configurations. Mostly, a mandatory feature is the core functionality of the software system and it is called 'base' functionality.

In addition to binary configuration options, numeric configuration options are not restricted to the binary selection {0, 1}, but on any other arbitrary set of numeric values. In this thesis, we only focus on binary configuration options.

In addition to functional properties (i.e., configuration options), we also have non-functional properties (NFPs) [SKK+] of a software system. In our thesis, we consider the non-functional property 'performance' of the system, which is based on the selected configuration. An example where a user or developer might be interested in the performance of the system are to produce *High-performance computer architecture* [KKSS18] or to produce *High-performance graphics* [AAJ+19]. Performance is explained in detail in the following section.

Section 2.2: Non-functional properties

The non-functional property that we mainly consider in our thesis is performance. Performance is considered as execution time in this thesis.

General equation of performance is shown below:

$$Performance = Execution\ Time$$

A user or developer of a configurable system is mainly interested in knowing the most relevant configuration and in knowing how the selected configuration options influence the total performance or execution time of the configuration. Besides, a developer of the configurable system is also interested in the performance evolution of the system [JSS+]. An evolution or a revision of software is a newer version of the same software with new or modified configuration options.

Configuration options not only have an influence on the execution time individually but when several configuration options are selected together, they may interact with each other, which is called an `interaction`. For example, a database management system like `MySQL` has the configuration options `encryption` and `compression`, they interact with each other when both are selected. This is because `compression` compresses the data and `encryption` now has a smaller size of data to encrypt, which influences the execution time negatively. However, in the case where `compression` is not enabled, `encryption` must encrypt the original size of the data which might influence execution time positively.

Execution time of an interaction is denoted by c(A)· c(B), where A, B $\in \mathcal{O}$ that form an interaction, c(A) being the individual execution time by the configuration option A. An interaction must have two or more participating configuration options.

For a given configuration of the software system, we need to compute its total execution time. To derive the execution time of a configuration, we use the tool '`SPL Conqueror`'. `SPL Conqueror` takes as input a variability model. They also depend on measurements of valid configurations to perform machine learning (multiple linear regression in combination with feature forward selection) on it and produces a set of valid *performance-influence models* as an output [SGAK]. '$\prod$' indicates performance-influence model of the system, denoted by $\prod: C \to \mathbb{R}$.

A performance-influence model is defined as follows:

$$\prod(c) = \beta_0 + \sum_{i \in \mathcal{O}} \beta_i \cdot c(i) + \sum_{i..j \in \mathcal{O}} \beta_{i..j} \cdot c(i)..c(j) \tag{2.2.1}$$

In the performance-influence model, β_0 represents constant execution time of the base functionality which is the root execution time shared by all configurations. $\sum_{i \in \mathcal{O}} \beta_i \cdot c(i)$ represents the influence of the configuration options on the performance on the system, with $\beta_i \cdot c(i)$ being the influence of individual configuration option and $\sum_{i..j \in \mathcal{O}} \beta_{i..j} \cdot c(i)..c(j)$ represents the influence of the interactions among two or more configuration options on the system with $\beta_{i..j} \cdot c(i)..c(j)$ being the execution time given by each interaction.

Every performance-influence model consists of a set of terms. Each term indicated by π_n, consists of the participating configuration option or interaction and a coefficient. This coefficient indicates the execution time that the configuration option or interaction contributes towards the total execution time of the system.

In the following example, we assume that all the configuration options are optional, none of them are mandatory. Hence, the set of configuration options is as follows $\mathcal{O}$ = {A, B, C }.

An example for performance-influence model:

$$\prod(c) = \overbrace{\underbrace{3}_{Coeff.} \cdot \underbrace{c(A)}_{Option}}^{\pi_1} + \overbrace{\underbrace{5}_{Coeff.} \cdot \underbrace{c(B)}_{Option}}^{\pi_2} + \overbrace{\underbrace{2}_{Coeff.} \cdot \underbrace{c(C)}_{Option}}^{\pi_3} - \overbrace{\underbrace{4}_{Coeff.} \cdot \underbrace{c(A) \cdot c(B)}_{Interaction}}^{\pi_4} \tag{2.2.2}$$

In the above given example there are four terms, the first term π_1, has the participating configuration option A, and it contributes 3 units towards the total execution time of the system. Units can be any measurement of time like milliseconds or seconds. π_2 has the participating option B and it contributes 5 units towards the total execution time of the system. Similarly, π_3 has configuration option C that contributes 2 units towards the execution time of the system. π_4 has two configuration options A and B, which when selected together interact with each other, hence it is called an interaction. This interaction contributes 4 units towards the total execution time of the system. The negative sign before the coefficient indicates that it decreases the execution time of the system, whereas the first 3 terms increase the execution time of the system which is indicated by the plus sign.

The performance-influence model shown in Equation 2.2.2 is simple. In practice, these models are quite complex with a large number of configuration options and interactions.

An example for complex performance-influence model:

$$\begin{aligned}\prod(c) =&81.85 \cdot c(A) - 55.67 \cdot c(B) - 27.97 \cdot c(C) + 195.14 \cdot c(C) \cdot c(D) - 50.22 \cdot c(E)\\ &- 40.35 \cdot c(F) + 164.68 \cdot c(F) \cdot c(D) + 3.41 \cdot c(G) + 232.77 \cdot c(C) \cdot c(D) \cdot c(H)\\ &+ 8.68 \cdot c(H) + 9.37 \cdot c(E) \cdot c(D) + 17.30 \cdot c(C) \cdot c(I) - 6.49 \cdot c(B) \cdot c(J)\\ &- 0.39 \cdot c(K) - 5.04 \cdot c(L) \cdot c(M) - 9.07 \cdot c(E) \cdot c(M) - 5.04 \cdot c(L) \cdot c(M)\\ &- 9.07 \cdot c(E) \cdot c(M) - 5.12 \cdot c(K) \cdot c(E) - 3.82 \cdot c(F) \cdot c(G)\\ &- 17.20 \cdot c(C) \cdot c(D) \cdot c(I) - 11.60 \cdot c(H) \cdot c(I) + 13.90 \cdot c(H) \cdot c(I) \cdot c(D)\\ &+ 13.90 \cdot c(H) \cdot c(I) \cdot c(D) - 48.34 \cdot c(N) + 35.77 \cdot c(N) \cdot c(D)1.58 \cdot c(I)\\ &- 20.09 \cdot c(N) \cdot c(D) \cdot c(K) - 55.04 \cdot c(I) \cdot c(O) + 98.02 \cdot c(I) \cdot c(O) \cdot c(D)\\ &- 6.98 \cdot c(N) \cdot c(I) - 2.19 \cdot c(K) \cdot c(B) - 5.81 \cdot c(H) \cdot c(E)\end{aligned}$$

This performance-influence model has more than 50 configuration options and interactions. This model is certainly difficult to make conclusions. It can be tedious and time consuming. To overcome this issue, this work aims at visualizing performance-influence models, which eases the interpretation of performance-influence models. Once we have the visualization, we can identify the most relevant configuration option or interaction. In the following section, we explain different visualization techniques, which can be used to visualize the performance-influence models.

Section 2.3: Visualizations

Visualization is an effective way of communicating a message [IGJ+]. Complex performance-influence models can be difficult to interpret by only looking at the data. Hence, we rely on the pictorial representation of displaying information. A user or developer would like to compare different NFPs like performance and energy consumption. However, these NFPs have different unit ranges. To overcome this issue, we normalize the performance-influence models. For instance we use the performance-influence model as in Equation 2.2.2

$$\prod_1(c) = 3 \cdot c(A) + 5 \cdot c(B) + 2 \cdot c(C) - 4 \cdot c(A) \cdot c(B) \tag{2.3.1}$$

and a second performance-influence model of a configurable software system.

$$\prod_2(c) = -6 \cdot c(A) + 2 \cdot c(B) + 3 \cdot c(A) \cdot c(B) \tag{2.3.2}$$

In the second performance-influence model, $\prod_2$ as given in Equation 2.3.2, configuration option C is not present, this is because it does not influence the performance of the system in any way. Hence, the performance influence by configuration option C is zero.

To normalize the performance-influence models, we first compute the β_{max} per influence model, it is the absolute highest value among all the β values. It is denoted mathematically as below:

$$\beta_{max} = max|\ \beta_0, \beta_{i'}, \beta_{i..j} : i', i..j \in \mathcal{O}\ | \tag{2.3.3}$$

To normalize the performance-influence models, we divide the coefficient of every term per model with the β_{max} of the corresponding model.

$$\prod(c) = \frac{\beta_0}{\beta_{max}} + \sum_{i \in \mathcal{O}} \frac{\beta_i}{\beta_{max}} \cdot c(i) + \sum_{i..j \in \mathcal{O}} \frac{\beta_{i..j}}{\beta_{max}} \cdot c(i)..c(j) \tag{2.3.4}$$

For the first performance-influence model in Equation 2.3.1, we have $\beta_{max} = 5$, which is the absolute highest value. The normalized performance-influence model is shown below.

$$\prod_1(c) = 0.6 \cdot c(A) + 1 \cdot c(B) + 0.4 \cdot c(C) - 0.8 \cdot c(A) \cdot c(B) \tag{2.3.5}$$

Similarly, for the second performance-influence model in Equation 2.3.2, we have $\beta_{max} = 6$, which is the absolute highest value. The normalized performance-influence model is shown below.

$$\prod_2(c) = -1 \cdot c(A) + 0.34 \cdot c(B) + 0.5 \cdot c(A) \cdot c(B) \tag{2.3.6}$$

We then use these normalized models as input for the visualizations. We selected three different visualizations, namely '*the Radar Plot*', '*the text Plot*' and '*the ratio Plot*'.

2.3.1 The Radar Plot

The radar plot as shown in Figure 2.1, is one way of visually representing the performance-influence models. We use the normalized performance-influence models given in Equation 2.3.5 and Equation 2.3.6 for this example. The outer circle indicates +1, the most inner circle indicates -1, and the middle circle indicates 0, which corresponds to negative influence, positive influence and zero or no influence respectively. The underlying grey axis lines indicate configuration options A, B, C and the interaction A · B. The performance influence from each configuration option and interaction are plotted on corresponding axis lines for every performance-influence model.

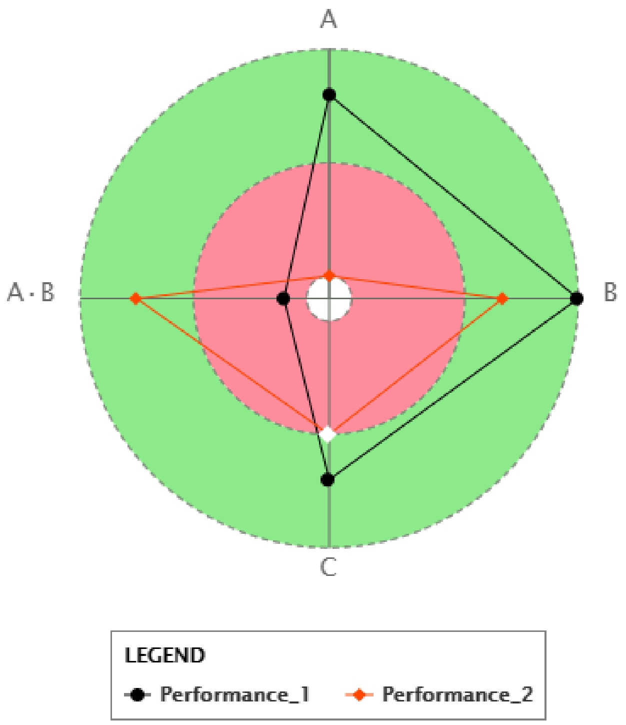


Figure 2.1: The radar Plot visualization for two performance-influence models, $\prod_1$ and $\prod_2$ with configuration options A, B, C and interaction A·B.

For the sake of simplicity, for this visualization and all other visualizations that follow, we have used A as a short-hand form for c(A). Similarly, for other configuration options and interactions.

If the performance contributed by a configuration option or interaction is 0, then it is indicated by an empty white marker symbol on the middle circle as seen by the configuration option C for the second performance-influence model.

In Figure 2.1, the first performance-influence model, $\prod_1$ is plotted in black and the second performance-influence model, $\prod_2$ in red. We can see from the visualization concerning first performance-influence model, that configuration options A, B and C are plotted in the green area, which means they increase the performance of the system. The interaction A·B is marked in the red area of the visualization, which indicates that it decreases the performance of the system because, in real world systems, there are configuration options with a small but relevant influence on an NFPs.

Similarly, for the second performance-influence model, we can see that the configuration option A lies in the red area of the visualization indicating that it decreases the performance of the system. Whereas, the configuration options B and the interaction A·B are plotted in the green area of the visualization, which indicates that they increase the performance of the system. Configuration option C has no influence on the performance of the system; hence it is marked with a different marker symbol that stands out from others. This is done to immediately recognize the configuration options that cause no influence on the performance of the system.

From Figure 2.1, we can infer that option B causes the highest influence on the performance for the first performance-influence model and for the second performance-influence model, configuration option A causes highest influence on the performance of the system, though they both lie in green and red area respectively.

2.3.2 The Text Plot

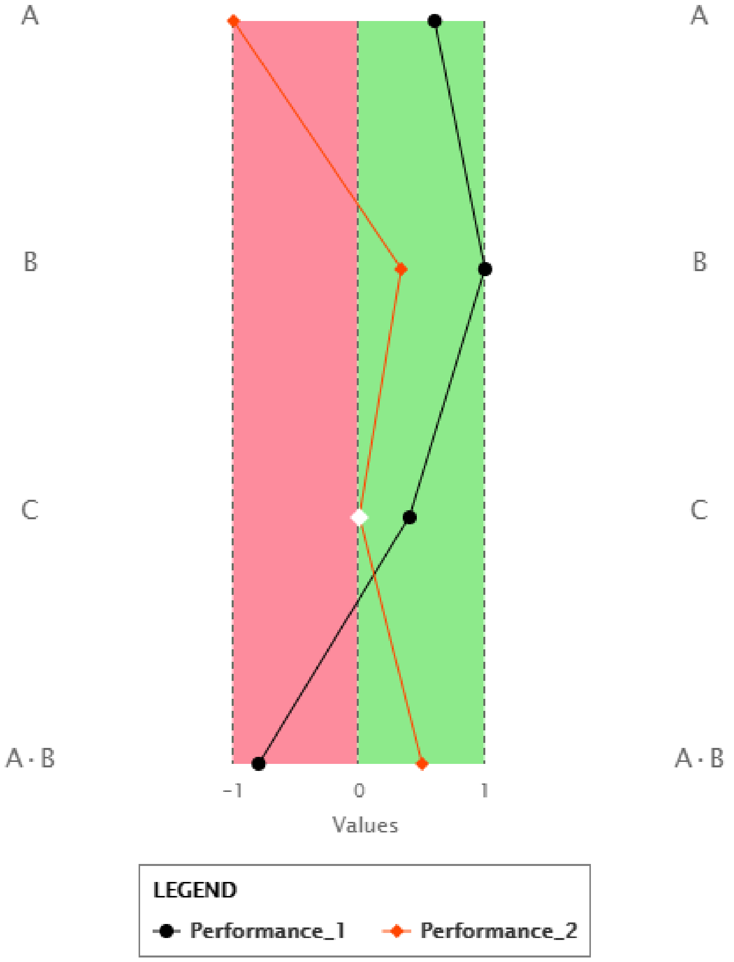


Figure 2.2: The text plot visualization for two performance-influence models, $\prod_1$ and $\prod_2$ with configuration options A, B, C and interaction A·B.

The text plot as shown in Figure 2.2, is the visual representation of the normalized performance-influence model in Equation 2.3.5 and Equation 2.3.6. As shown in the Figure 2.2, the left border line indicates -1, the middle line indicates 0 and the right line indicates +1, corresponding to positive, zero, and negative influence, respectively.

This visualization technique and the color coding is perceived in a way similar to that of the radar plot. The black line indicates the first performance-influence model, $\prod_1$ and the red line indicates the second performance-influence model, $\prod_2$.

From this visualization, we can easily infer that the configuration options plotted on the left and right lines are the ones that make highest influence on the system.

For instance, for the first performance-influence model configuration option B makes highest influence on the performance of the system since it lies on the right line of the visualization. Similarly, for the second performance-influence model configuration option A makes the highest influence on the performance of the system, since it lies on the left line of the visualization.

The configuration option or interaction are marked on both the sides of the plot, hence this plot has the advantage of looking at the performance-influence models textually. This plot is easier for comparison of models, since the configuration options are vertically aligned than the radial form in the radar plot.

2.3.3 The Ratio Plot

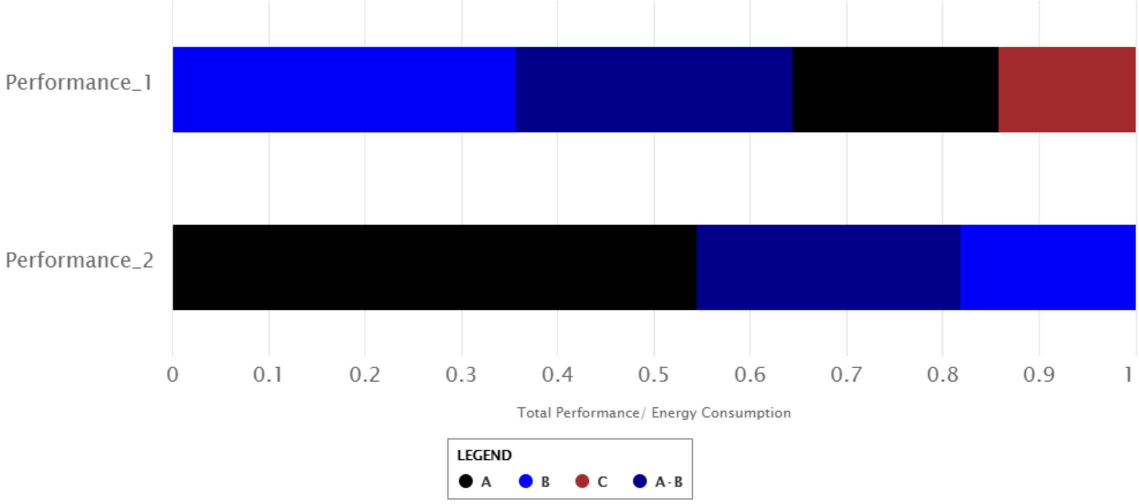


Figure 2.3: The ratio plot visualization for two performance-influence models, $\prod_1$ and $\prod_2$ with configuration options A, B, C and interaction A·B.

Normalization: For the ratio plot, we normalize the performance-influence models in a different way. We calculate the total performance of the system, it is the sum of all of the β terms and is denoted mathematically as below:

$$\beta_{total} = \beta_0 + \sum_{i \in \mathcal{O}} \beta_i + \sum_{i..j \in \mathcal{O}} \beta_{i..j} \tag{2.3.7}$$

After we obtain the β_{total}, we normalize the performance-influence models, we divide the coefficient of every term per model with the β_{total} of the corresponding model.

$$\prod(c) = \frac{\beta_0}{\beta_{total}} + \sum_{i \in \mathcal{O}} \frac{\beta_i}{\beta_{total}} \cdot c(i) + \sum_{i..j \in \mathcal{O}} \frac{\beta_{i..j}}{\beta_{total}} \cdot c(i)..c(j) \tag{2.3.8}$$

I.e, we calculate the performance of individual configuration option or interaction with respect to the total performance to obtain the relative performance influence of the configuration options and interactions.

For the performance-influence model in Equation 2.3.1, we have the total performance i.e, $\beta_{total} = 14$ units and we divide coefficient of every term by 14 to obtain the relative performance influence.

$$\prod_{1}(c) = 0.214 \cdot c(A) + 0.357 \cdot c(B) + 0.142 \cdot c(C) - 0.285 \cdot c(A) \cdot c(B) \tag{2.3.9}$$

Similarly, for the performance-influence model in Equation 2.3.2, we have the total performance influence i.e, $\beta_{total} = 11$ units and the corresponding normalized performance-influence model is shown below:

$$\prod_{2}(c) = -0.545 \cdot c(A) + 0.181 \cdot c(B) + 0.272 \cdot c(A) \cdot c(B) \tag{2.3.10}$$

Visualization: After obtaining the normalized performance-influence models, we sort the terms in descending order starting from the left side of the visualization regardless of the positive or the negative sign of the term. Hence the first bar represents the configuration option or interaction that makes the highest influence on the total performance of the system, the second bar makes the next highest influence on the total performance of the system and so on. As shown in the Equation 2.3.10, the configuration option C for the second performance-influence model has no relevant performance influence, and therefore it does not appear in the ratio plot.

The Ratio plot as shown in Figure 2.3, is the visual representation of the performance-influence models in Equation 2.3.9 and Equation 2.3.10. The ratio plot represents the general influence that a configuration option or interaction has on the performance of the system, whereas the radar plot and the text plot show the positive and negative influence that a configuration option or interaction has on the performance of the entire system.

We can see from the first performance-influence model, $\prod_1$ that configuration option B makes the highest influence and configuration option C makes the lowest influence on performance, regardless of it being positive or negative

Similarly, for the second performance-influence model, $\prod_2$ we can see that the configuration option A makes the highest influence on the performance of the system, regardless of it being positive or negative. Configuration option B makes the least influence on the performance of the system and configuration option C makes no influence on the performance of the system. Hence, it does not appear in the visualization.

An advantage of this plot would be to check the overall influence a configuration option or interaction has on the performance of the system, irrespective of it being a positive or negative influence.

3. Methodology

In this chapter, we present the research questions and the methods to evaluate the research questions. In Section 3.1, we introduce the research questions and the motivation behind them. In Section 3.2, we introduce the design, interview procedure, and the interview structure. Also, how the interview can be used to evaluate the research questions. In the interview, we present different visualization techniques and therefore, assess the quality of these visualization techniques for different use cases by conducting an interview. We also present the dependent and independent variables that affect the interview.

Section 3.1: Research Questions

To assess the quality of the visualizations with respect to different use cases, we formulate different research questions. Research questions are selected based on the number of performance-influence models they involve. For this thesis, we have selected 4 research questions each considering use cases one performance-influence model, two performance-influence models and the final two belonging to multiple performance-influence models.

3.1.1 One performance-influence model

Research questions considering one performance-influence model are selected to evaluate the visualizations when a user is interested in knowing only the influence of the configuration options and interactions among them of one non-functional property of the software. The information is useful for selecting or deselecting configuration options to optimize the behavior of the system with respect to the non-functional property.

RQ1: Can we use the visualization techniques to identify relevant configuration options or interactions of one performance-influence model?

When our focus is on one performance-influence model, our area of interest is on the terms that cause the highest performance increase or decrease. This indicates that the corresponding configuration option or interaction is making an influence on the performance that is higher compared to other configuration options or interactions.

If a term has the highest positive influence on the configurable software system, it indicates that the corresponding configuration option increases the performance of the system much more than other configuration options or interactions. Similarly, if a term has the highest negative influence, it indicates that the corresponding configuration option decreases the performance of the system much more than other configuration options or interactions.

Since performance-influence models have different levels of complexities, we further investigate these levels of complexities as an additional factor. Therefore, we have sub-questions with simple and complex complexity.

RQ1.1: Can we use the visualization techniques to identify relevant relevant configuration options or interactions of one simple performance-influence model?

For this sub-question, we consider the use case of simple performance-influence models. Simple performance-influence model does not contain interactions among configuration options. This makes it easier to identify the configuration option that causes the most relevant influence on the performance of the entire system.

RQ1.2: Can we use the visualization techniques to identify relevant configuration options or interactions of one complex performance-influence model?

For complex performance-influence models, we consider interactions along with configuration options. Hence, finding out the most relevant configuration option or interaction is more complex than the simple performance-influence model use case.

3.1.2 Two performance-influence models

Research questions based on two performance-influence models are selected to evaluate the visualizations when a user wants to compare the models. For instance, a comparison of two revisions or a comparison of two different NFPs of software.

When we consider two performance-influence models, we usually want to compare performances between them. Comparison for instance, is done over two revisions of the software system. Many times, a revision of a software needs to be compared to its earlier or next revision to determine how the two revisions differ in terms of its performance. Our area of interest here is to find out the similarities or differences in the two revisions.

RQ2: Can we use the visualization techniques to compare two performance-influence models?

Comparison can also be done in terms of two different NFPs. For instance, if a user is interested in knowing how a configuration of a software affects the energy consumption and performance, he can use an energy-influence model and a performance-influence model in the visualization to compare them. As in RQ1, we formulate the following sub-questions for different levels of complexities of the performance-influence model.

RQ2.1: Can we use the visualization techniques to compare two simple performance-influence models?

For the simple use-case, we do not consider interactions in our performance-influence models.

RQ2.2: Can we use the visualization techniques to compare two complex performance-influence models?

For the complex use case, we consider interactions among the configuration options.

3.1.3 Many performance-influence models

Finally, research questions based on more than two performance-influence models are selected to evaluate visualizations, when a user is interested in comparing several revisions of a configurable software system or comparing multiple different NFPs.

With a new revision, configurable software system adds new configuration options, performance-influence models scale to reflect the influence of newly added configuration options on the performance of the system. Scaling can be of two types; addition of new configuration options, addition of new performance-influence models.

RQ3: How good can the visualizations be used to compare a high number of performance-influence models and a high number of terms?

When we have many performance-influence models, our area of interest is in finding a pair or more of performance-influence models that share a large set of influences. This visualization also helps to notice outliers among performance-influence models.

RQ3.1: How good can the visualizations be used to compare a high number of simple performance-influence models and a high number of terms?

For the simple use case, we considered that at least one pair of performance-influence models shared a set of influences.

RQ3.2: How good can the visualizations be used to compare a high number of complex performance-influence models and a high number of terms?

For the complex use case, we considered that at more than a pair of performance-influence models shared a similar set of influences.

RQ4: What are the differences with respect to visualization techniques regarding many performance-influence models?

With this research question, we evaluate how robust the visualization techniques are and how do the visualizations scale with respect to the addition of configuration options or addition of performance-influence models? We evaluate if the visualizations of many performance-influence models can be inferred with the same ease as that of visualizations with fewer performance-influence models.

RQ4.1: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of terms?

This sub-question evaluates the scalability of performance-influence models with respect to the addition of configuration options or interactions.

RQ4.2: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of models?

This sub-question evaluates the scalability of performance-influence models with respect to the addition of performance-influence models.

Section 3.2: Design and Interview

3.2.1 Realization

In this section, we discuss on how we plan on getting the answers to the research questions and as a method to acquire the answer to the research questions, we conduct an interview with several participants [DSC14]. We only considered participants that had prior knowledge either in performance-influence models or configurable software systems. We had 9 participants answering the interview with the help of the performance-influence model tool. 5 participants belong to the Faculty of Computer Science and Mathematics, Passau, while 4 belonged to Bauhaus-University of Weimar. The invites were sent by using a mail address from the University of Passau.

We interviewed the participants to answer a questionnaire with several questions based on visualization of performance-influence models. The questionnaire is attached in the Appendix:A.3. The interview consisted of an initial warm-up phase, where the interviewee is given a brief explanation of performance-influence models, configurable software systems, configuration options, interactions, and the visualization techniques. The warm-up phase is conducted so that we reduce bias between the interviewees who have not seen the visualizations before from those who have. During this phase, the interviewee also got to answer a question using different visualization techniques as a try-out question.

The questions are categorized based on the number of performance-influence models considered in the questions. There are three sections; one performance-influence model, two performance-influence models, and many performance-influence models. Each section has several questions, each question includes a different performance-influence model for each of the visualization techniques; the radar plot, the text plot and the ratio plot. Questions considering each section are used to evaluate research questions that correspond to the same section. For instance, questions (Q1, Q2...) considering one performance-influence models are used to evaluate the research questions (RQ1, RQ1.2...) considering one performance-influence model.

Every question has a difficulty rating indicating how easy or difficult it is for a user to derive the answer to the question. The difficulty rating is done using Likert scales, they are used as sanity check measure to quickly evaluate whether a claim or the result of a calculation can possibly be true.

If the questions have a definite answer, an area to fill in the answer is provided. Also, each question has a comment or feedback section to fill in, in the case when the user has possible ideas for improvement of the visualizations or the tool.

RQ1: Can we use the visualization techniques to identify relevant configuration options or interactions of one performance-influence model?

> **RQ1.1**: Can we use the visualization techniques to identify relevant configuration options or interactions of one simple performance-influence model?
>
> **RQ1.2**: Can we use the visualization techniques to identify relevant configuration options or interactions of one complex performance-influence model?
>
> To evaluate these research questions, we selected the following questions in our questionnaire.
>
> **Q1: Which is the most relevant configuration option or interaction?**
>
> When we have visualization with only one performance-influence model, our interest is on the configuration options or interactions that stand out or the ones that make the highest influence on performance.
>
> **Q2: Which is the configuration option or interaction that leads to the highest performance increase or decrease?**
>
> A user would like to know which of the configuration options or interactions increases or decreases the performance the most. The user then has the choice to either deselect or select the corresponding configuration options or interactions to improve the performance of the system.

RQ2: Can we use the visualization techniques to compare two performance-influence models?

RQ2.1: Can we use the visualization techniques to compare two simple performance-influence models?

RQ2.2: Can we use the visualization techniques to compare two complex performance-influence models?

To evaluate these research questions, we selected the following questions in our questionnaire.

Q3: Which is the configuration option or interaction where the performance-influence models differ the most?

When two performance-influence models are concerned, we compare two different revisions of the software. A user is interested if the current revision of the software that has configuration options or interactions that performs very differently than its previous revision. He can infer if that configuration options or interactions caused a performance spike or improved the performance of the software.

Q4: Which is the configuration option or interaction where the performance-influence models are the most similar?

A user is also interested if the current revision of the software has configuration options or interactions that performs similar to its previous revision. He can infer if that configuration options or interactions is stable or the new revision of software did not affect the performance of these configuration options or interactions.

RQ3: How good can the visualizations be used to compare a high number of performance-influence models and a high number of terms?

RQ3.1: How good can the visualizations be used to compare a high number of simple performance-influence models and a high number of terms?

RQ3.2: How good can the visualizations be used to compare a high number of complex performance-influence models and a high number of terms?

To evaluate these research questions, we selected the following questions in our questionnaire.

Q5 & Q6: which pair of performance-influence models share a large set of influences?

With many performance-influence models, we validate the scalability of the models. A user is interested to know which models among the many, share a maximum set of influences.

Question 5 corresponds to scalability with respect to terms and question 6 corresponds to scalability with respect to models.

RQ4: What are the differences with respect to visualization techniques regarding many performance-influence models?

RQ4.1: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of terms?

RQ4.2: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of models?

To evaluate these research questions, we rely on correctness and time taken for question 5 and question 6. We present the results using box plot with co-ordinates; time vs simple and complex use-case for each of the visualization technique. We also use Mann-Whitney U test for this research question.

3.2.2 Independent Variables

An independent variable is a variable that is manipulated, to test its effects on dependent variables.

Simple vs Complex Performance-influence Models

Every question in the questionnaire is divided into two sections, namely 'Simple' and 'Complex'. With this variable, we aim at considering how good the visualizations scale and how the perception or time taken by the interviewee to answer the question changes with respect to the complexity of the model.

Visualization Techniques

We have 3 visualization techniques; the radar plot, the text plot and the ratio plot. Every question was presented with each of the visualization techniques. With this variable, we aim to find which visualization technique is best to answer a given question. This variable has a direct impact on the difficulty rating.

3.2.3 Dependent Variables

Also, these are variables that are used as a factor to measure the outcome of the interview, namely

Correctness

This dependent variable refers to the correctness of the answer. Correctness is measured in percent and indicates the relative number of participants that could answer the question correctly.

Time

When a participant was answering questions from the questionnaire, a timer was started at the start of each question and ended as soon as the participant had filled in his response to the question. This was done to track the time required by each question. This helped us to support the analysis of how suitable some visualization techniques are to answer certain questions.

Mann-Whitney U Test: This test helps us to determine if one visualization technique is significantly better than other visualization techniques when time is considered as a factor [GS18].

Difficulty rating

Difficulty rating in our thesis is assessed using `Likert scales`. This scale helps us determine the difficulty perception of the interviewee based on the visualization technique and the complexity of the performance-influence model.

Pearson Correlation Coefficient: This coefficient helps us to determine if the time measurements of each question are linearly or non-linearly co-related to the corresponding difficulty ratings. This coefficient is computed for each question, for different complexities.

4. Evaluation

In this chapter, we present and discuss the results of the interview. We evaluate the research questions from the results obtained from the interview. We assess the results by using the correctness of the answers, the time needed and the difficulty rating giving by the interviewees. Correctness is presented in Section 4.1 and time measurements is presented in Section 4.2. We also present the results of difficulty rating in Section 4.3. We answer our research question from the presented results in Section 4.4. In Section 4.5, we discuss our findings with respect to the results. We also present certain factors in Section 4.6, that might have influenced our interview unfavorably. Finally, in Section 4.7, we present feedback given by the interviewees on how the visualization of performance-influence model tool can be improved.

Section 4.1: Correctness

We check the answers given by the interviewees to the questions asked in the questionnaire. We evaluate how often an interviewee answered the question correctly. The evaluation is done with respect to each question and each use case; simple and complex. Table 4.1 presents the relative number of correct answers given for each use case with its summary per question.

Section 4.2: Time Measurements

The second factor to evaluate the research questions, is by taking time measurements.

Time is measured on how long an interviewee took to answer a question. This is done to compare if an interviewee took a long time to answer a complex use case than a simpler one or to check if one of the visualization techniques was more time consuming than others.

We will conduct the *Mann–Whitney U test* in this section.

Input	Use Case	Radar Plot	Text Plot	Ratio Plot
Q1	Simple	100%	100%	88.8%
	Complex	100%	100%	77.7%
	Summary	100%	100%	83.83%
Q2	Simple	70%	100%	60%
	Complex	100%	100%	NA
	Summary	83.34%	100%	60%
Q3	Simple	100%	100%	100%
	Complex	22.23%	100%	66.66%
	Summary	61.11%	100%	66.66%
Q4	Simple	100%	88.89%	22.23%
	Complex	88.89%	88.89%	55.55%
	Summary	94.45%	88.89%	26.66%
Q5	Simple	88.89%	88.89%	33.34%
	Complex	33.34%	100%	88.89%
	Summary	77.77%	94.44%	61.11%
Q6	Simple	33.34%	22.22%	88.88%
	Complex	55.55%	77.77%	77.77%
	Summary	61.11%	50%	83.33%

Table 4.1: The relative number of correctly answered questions in the interview for the different visualization techniques across the different questions and complexities.

Input	Radar Plot &lt; Text Plot	Text Plot &lt; Ratio Plot	Ratio Plot &lt; Radar Plot	Radar Plot &gt; Text Plot	Text Plot &gt; Ratio Plot	Ratio Plot &gt; Radar Plot
Q1 Simple	0.535	0.638	0.5	0.5	0.395	0.535
Q1 Complex	0.535	0.464	0.604	0.5	0.570	0.429
Q2 Simple	0.855	**0.010**	0.953	0.165	0.991	**0.055**
Q2 Complex	0.811	–	–	0.213	–	–
Q3 Simple	0.5	**0.004**	0.994	0.535	0.996	**0.006**
Q3 Complex	0.982	**0.004**	0.701	**0.021**	0.996	0.329
Q4 Simple	0.395	**0.001**	0.999	0.638	0.998	**0.000**
Q4 Complex	0.638	**0.006**	0.996	0.395	0.994	**0.004**
Q5 Simple	0.973	0.055	0.701	**0.031**	0.953	0.329
Q5 Complex	0.429	0.125	0.670	0.604	0.891	0.361
Q6 Simple	0.464	0.760	0.329	0.570	0.268	0.701
Q6 Complex	0.535	0.188	0.811	0.5	0.834	0.213
Q1 Simple & Complex	0.581	0.493	0.544	0.430	0.581	0.468
Q2 Simple & Complex	0.920	**0.004**	0.973	0.084	0.995	**0.030**
Q3 Simple & Complex	0.894	**0.000**	0.976	0.111	0.999	**0.025**
Q4 Simple & Complex	0.518	**0.000**	0.999	0.493	0.999	**0.000**
Q5 Simple & Complex	0.899	**0.028**	0.559	0.105	0.973	0.454
Q6 Simple & Complex	0.455	0.369	0.665	0.556	0.642	0.346

Table 4.2: Mann-Whitney U test results for the different visualization techniques across the different questions and complexities.

4.2.1 Mann–Whitney U test

We use the Mann-Whitney U test to determine if one sample is significantly better than another sample when time is considered as a factor. A sample is significantly better than another if the p value of the test is less than 0.05. Table 4.2, presents the results for Mann–Whitney U test. The significant values are marked in bold.

In our case, we determine if one group of time measurement is significantly better than another group of time measurement. We compare time measurement groups between the radar plot, the text plot, and the ratio plot.

Here we try to find, if a visualization technique consumed significantly less time than other visualization techniques. For instance, radar plot < text plot implies if the radar plot is better than the text plot if the Mann–Whitney U test value is less 0.05. Similarly, radar plot > text plot implies if the radar plot is worse than the text plot.

Section 4.3: Difficulty Rating

The difficulty rating in our thesis is assessed using Likert scales. This scale helps us determine the difficulty perception of the interviewee based on the visualization technique and the complexity of the performance-influence model.

The difficulty rating is considered as a factor for the evaluation of research questions. The interviewee selects the difficulty rating on how easy or difficult it was to answer a question. This rating corresponds to the time taken by the interviewee to answer the question. Table 4.3, we present the average difficulty ratings and their standard deviation.

4.3.1 Pearson Correlation Coefficient

We compute the Pearson correlation coefficient [Pea96] with difficulty ratings and time measurements as input. This coefficient is used to determine the measure of linear correlation between the 2 inputs. Table 4.4, presents the linear correlation values for the same.

The coefficient ranges from -1 to +1. A value of less than 0.5 implies a weak correlation, a value between 0.5 - 0.8 implies medium correlation and value greater than 0.8 implies a strong correlation. A positive coefficient implies that both inputs increase or decrease linearly and a negative coefficient implies that both increase or decrease inversely.

From Table 4.4, we can see that the only strong correlation coefficient is between the time measurements and difficulty ratings of Q3 for a simple use case with the radar plot as the visualization technique.

Time measurements and difficulty ratings are also plotted for each interviewee, for each question. This is done so as to look at the correlation between them graphically. These plots are presented in Appendix:A.2

	Radar Plot	Text Plot	Ratio Plot
Input	Mean ± SD	Mean ± SD	Mean ± SD
Q1 Simple	1.3 ± 0.471	1.5 ± 0.496	1.2 ± 0.415
Q1 Complex	2.4 ± 0.831	2.2 ± 0.628	1.3 ± 0.471
Q2 Simple	1.5 ± 0.684	1.5 ± 0.831	4.2 ± 1.030
Q2 Complex	1.8 ± 0.566	2.0 ± 0.471	–
Q3 Simple	1.5 ± 0.496	1.6 ± 0.666	3.1 ± 1.286
Q3 Complex	2.8 ± 1.099	2.1 ± 0.314	3.4 ± 1.065
Q4 Simple	1.6 ± 0.471	1.7 ± 0.415	3.7 ± 0.916
Q4 Complex	2.4 ± 0.496	1.8 ± 0.314	3.5 ± 0.955
Q5 Simple	2.6 ± 1.054	2.2 ± 0.785	4.3 ± 0.942
Q5 Complex	2.8 ± 0.993	1.7 ± 0.628	4.4 ± 0.684
Q6 Simple	2.7 ± 0.628	3.0 ± 0.666	3.6 ± 1.054
Q6 Complex	2.3 ± 0.471	2.5 ± 0.955	4.2 ± 1.030
Q1 Simple & Complex	1.8 ± 0.874	1.8 ± 0.657	1.2 ± 0.447
Q2 Simple & Complex	1.7 ± 0.650	1.7 ± 0.711	4.2 ± 1.030
Q3 Simple & Complex	2.2 ± 1.082	1.8 ± 0.566	3.2 ± 1.192
Q4 Simple & Complex	2.0 ± 0.621	1.8 ± 0.372	3.6 ± 0.942
Q5 Simple & Complex	2.7 ± 1.030	2.0 ± 0.745	4.3 ± 0.825
Q6 Simple & Complex	2.5 ± 0.598	2.7 ± 0.853	3.9 ± 1.078

Table 4.3: Average difficulty rating and their standard deviations for all interviewees. The mean and standard deviations are computed for each different visualization techniques across the different questions and complexities.

	Radar Plot	Text Plot	Ratio Plot
Input	Coefficient	Coefficient	Coefficient
Q1 Simple	-0.013	0.216	0.711
Q1 Complex	0.637	0.403	-0.218
Q2 Simple	0.384	-0.280	-0.019
Q2 Complex	0.753	0.576	–
Q3 Simple	0.877	0.566	0.506
Q3 Complex	0.163	0.509	0.119
Q4 Simple	0.350	0.347	0.626
Q4 Complex	-0.268	0.373	-0.280
Q5 Simple	0.289	0.493	0.401
Q5 Complex	0.618	0.773	-0.133
Q6 Simple	-0.145	-0.268	0.519
Q6 Complex	0.293	0.178	0.487

Table 4.4: Pearson correlation coefficient for the different visualization techniques across the different questions and complexities.

Section 4.4: Results

We now answer the research questions by using the results from the correctness values, the time measurements, and the difficulty ratings.

We also use Pareto front plots for each research question. So far we have considered the two evaluation factors; correctness and time separately. Using Pareto front plots, we combine these two factors. These plots are plotted with time vs correctness. They help us to determine which visualization technique is the best when considering both these evaluation factors. Visualization is considered the best when it takes the lowest amount of time to answer a question with the the highest relative number of correct values.

One Performance-Influence Model

RQ1: Can we use the visualization techniques to identify relevant configuration options or interactions of one performance-influence model?

We consider Q1 and Q2 for this research question.

RQ1.1: Can we use the visualization techniques to identify relevant configuration options or interactions of one simple performance-influence model?

For this sub-question, we only consider the simple use case for Q1 and Q2.

Correctness: From Table 4.1, we can infer that all the interviewees were able to answer Q1 and Q2 correctly when presented with the text plot as the visualization technique. Whereas, the ratio plot did not perform as good as the text plot for both Q1 and Q2.

Time Measurements: From the results presented in Table 4.2, we can see that only for Q2, the text plot proved to be significantly better than other visualization techniques.

Difficulty Ratings: From the difficulty ratings presented in Table 4.3, we can note that the average difficulty ratings for Q1 and Q2 are below 2.4 with an exception of the ratio plot for Q2. Where it has an average rating of 4.2, but also with a higher standard deviation.

Q1: From Figure 4.1, for the simple use case the text plot is much better than other visualization techniques. We can also notice that the ratio plot takes slightly less time, but does not guarantee perfect correctness results.

Q2: From Figure 4.2, for the simple use case, the text plot is much better both in terms of time and correctness.

Summary of RQ1.1: The answer to this sub-question is **yes**, with the text plot being the choice of visualization.

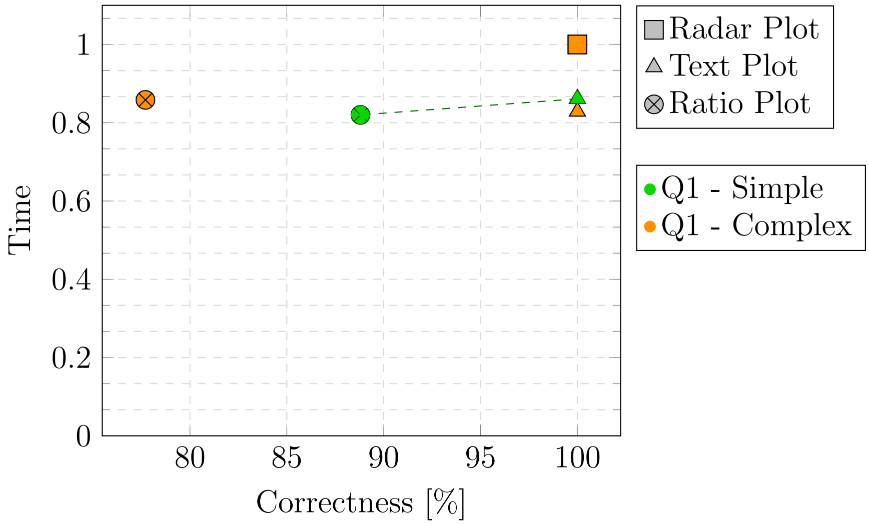


Figure 4.1: The correctness and time values for Q1 (simple and complex). The Pareto front for the different complexities is drawn by using a green and orange line respectively. The visualizations on the Pareto front for Q1 - Simple is text and the ratio plot and for Q1 - Complex is the text plot.

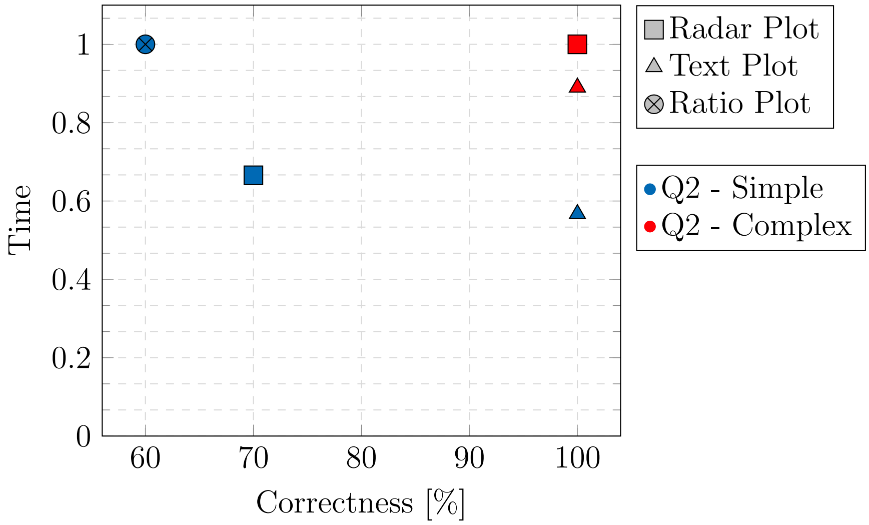


Figure 4.2: The correctness and time values for Q2 (simple and complex). The Pareto front for the different complexities is drawn by using a blue and red line respectively. The visualizations on the Pareto front for Q2 - Simple is the text plot and for Q2 - Complex is the text plot.

RQ1.2: Can we use the visualization techniques to identify relevant configuration options or interactions of one complex performance-influence model?

For this sub-question, we only consider the complex use case for Q1 and Q2.

Correctness: From Table 4.1, we can infer that all the interviewees were able to answer the Q1 and Q2 correctly for both the radar plot and the text plot.

Time Measurements: From the results presented in Table 4.2, we can see infer that none of the visualization technique proved to be significantly better than the other visualization techniques.

Difficulty Ratings: From the difficulty ratings presented in Table 4.3, we can infer that all of the 3 visualization techniques had an average difficulty rating below 2.4

Q1: From Figure 4.1, we can notice that for the complex use case, both the radar plot and the text plot perform equally better in terms of correctness. However, the text plot performed slightly better in terms of time. Whereas, the ratio plot did not perform as good as text and the radar plot.

Q2: From Figure 4.2, for the complex use case, the text plot is slightly better than the radar plot both in terms of time and correctness. The ratio plot was not considered for this question, since the ratio plot displays the general influence of a configuration option or interaction, also that the ratio plot does not display configuration options with zero or no influence.

Summary of RQ1.2: The answer to this sub-question is **yes**, with the text plot being the choice of visualization.

Summary of RQ1: Hence, the answer to this research question is **yes**, with the text plot being the considerable choice of visualization, since it has better results when considering time, correctness, and difficulty rating factors.

Two Performance-Influence Models

RQ2: Can we use the visualization to compare two performance-influence models?

We consider Q3 and Q4 for this research question.

RQ2.1: Can we use the visualization to compare two simple performance-influence models?

For this sub-question, we consider only the simple use case for Q3 and Q4.

Correctness: From Table 4.1, we can infer that for Q3 all the interviewees answered correctly when presented with any of the 3 visualization techniques, whereas for Q4 only the radar plot performed better than the text plot and the ratio plot.

Time Measurements: From the results presented in Table 4.2, we can see that the text plot performed significantly better in terms of time measurements.

Difficulty Ratings: From the difficulty ratings presented in Table 4.3, we can see that the ratio plot has on an average higher difficulty rating than the text plot and the radar plot.

Q3: From Figure 4.3, for the simple use case, we can infer that both the text plot and the radar plot are equally better when both factors; correctness and time are considered. We also see that both these visualization techniques produce perfect correctness results.

Q4: From Figure 4.4, for the simple use case, the radar plot is better in terms of correctness measurements. But, from Table 4.2, we know the text plot does significantly better than other visualization techniques. This implies that even though interviewees took less time to answer the text plot, the results were not 100% correct.

Summary of RQ2.1: Therefore, the answer to this sub-question is **yes**, with the text plot being the considerable choice of visualization.

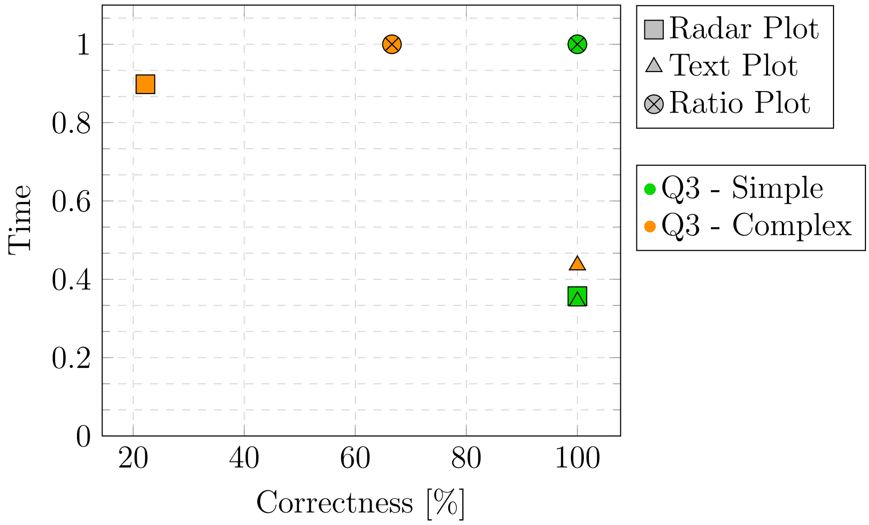


Figure 4.3: The correctness and time values for Q3 (simple and complex). The Pareto front for the different complexities is drawn by using a green and orange line respectively. The visualizations on the Pareto front for Q3 - Simple is the radar and the text plot and for Q3 - Complex is the text plot.

RQ2.2: Can we use the visualization to compare two complex performance-influence models?

For this sub-question, we consider only the complex use case for Q3 and Q4.

Correctness: From Table 4.1, we can infer that for Q3, all the interviewees answered correctly when presented with the text plot. Whereas, for Q4 none of the visualization techniques gave perfect correctness answers.

Time Measurements: From the results presented in Table 4.2, we can see that the text plot performed significantly better in terms of time measurements.

Difficulty Ratings: From the difficulty ratings presented in Table 4.3, we can see that the ratio plot has on an average higher difficulty rating than the text plot and the radar plot.

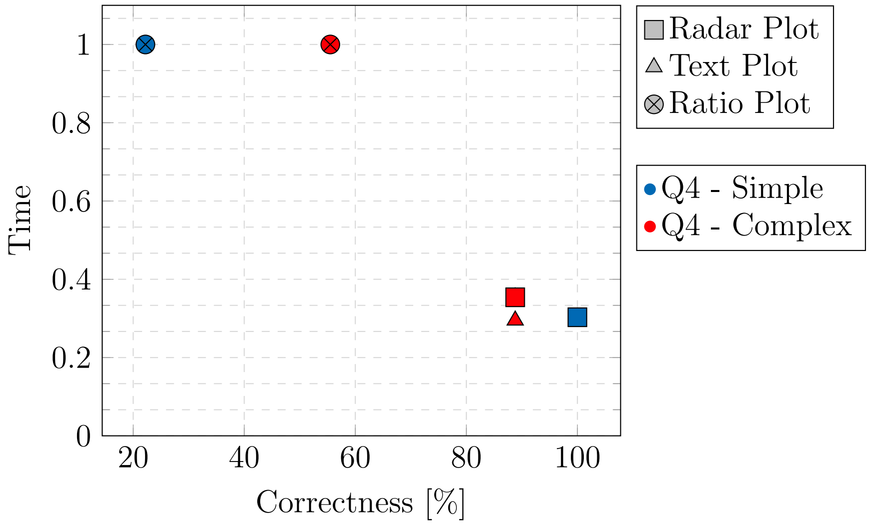


Figure 4.4: The correctness and time values for Q4 (simple and complex). The Pareto front for the different complexities is drawn by using a blue and red line respectively. The visualizations on the Pareto front for Q4 - Simple is the radar plot and for Q4 - Complex is the text plot.

Q3: From Figure 4.3, for the complex use case, we can infer that both the text plot outperformed other visualization techniques when both factors; correctness and time are considered.

Q4: From Figure 4.4, for the complex use case, we can infer that the text plot is slightly better than the radar plot both in terms of time and correctness factors, even though it did not lead to perfect correctness results.

Summary of RQ2.2: Therefore, the answer to this sub-question is **yes**.

Summary of RQ2: Hence, the answer to this research question is **yes**, with the text plot being the considerable choice of visualization technique, since it has better results when considering time, correctness, and difficulty ratings.

Many Performance-Influence Models

RQ3: How good can the visualizations be used to compare a high number of performance-influence models and a high number of terms?

We consider Q5 and Q6 for this research question.

Q5 corresponds to scalability with respect to the addition of performance-influence models and Q6 corresponds to scalability with respect to the addition of configuration option or interactions.

RQ3.1: How good can the visualizations be used to compare a high number of simple performance-influence models and a high number of terms?

For this sub-question, we only consider the simple use case for Q5 and Q6.

Correctness: From Table 4.1, we can infer that for both Q5 and Q6, none of the visualization techniques lead to perfect correctness answer. Although, the text plot performed comparatively better for Q5 and the ratio plot performed comparatively better for Q6.

Time Measurements: From the results presented in Table 4.2, none of the visualization techniques performed significantly better in terms of time measurements.

Difficulty Ratings: From the difficulty ratings presented in Table 4.3, we can infer that the ratio plot has higher difficulty ratings for Q5 and Q6.

Q5: From Figure 4.5, for the simple use case, we can see that the text plot is a better choice of visualization technique when considering both time and correctness factors.

Q6: From Figure 4.6, for the simple use case, we can infer that if time is considered as a factor, the radar plot performs comparatively better than other visualization techniques. However, when correctness is considered as a factor, the ratio plot performs better. Even though both these plots do not lead to perfect correctness results.

Summary of RQ3.1: The answer to this sub-question is **yes**, with the text plot as the best visualization type for performance-influence models with many terms and the ratio plot as the best visualization for many performance-influence models.

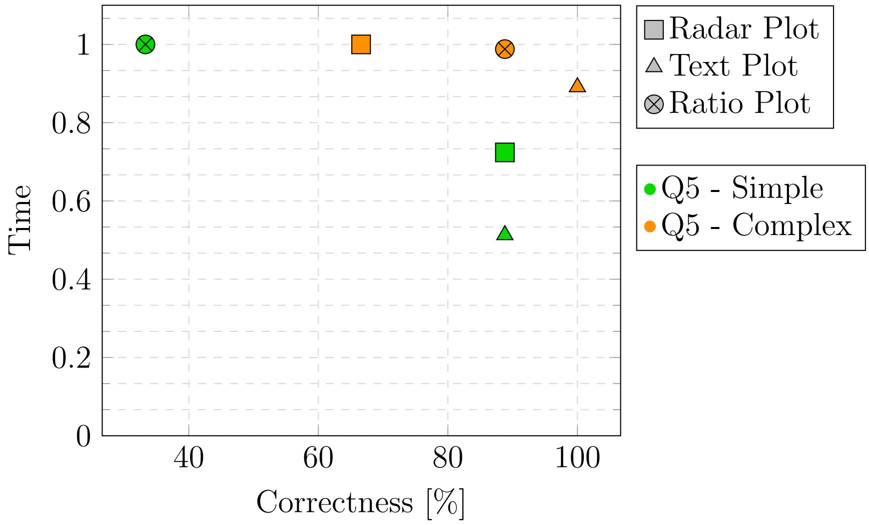


Figure 4.5: The correctness and time values for Q5 (simple and complex). The pareto front for the different complexities is drawn by using a green and orange line respectively. The visualizations on the pareto front for Q5 - Simple is the text plot and for Q5 - Complex is the text plot.

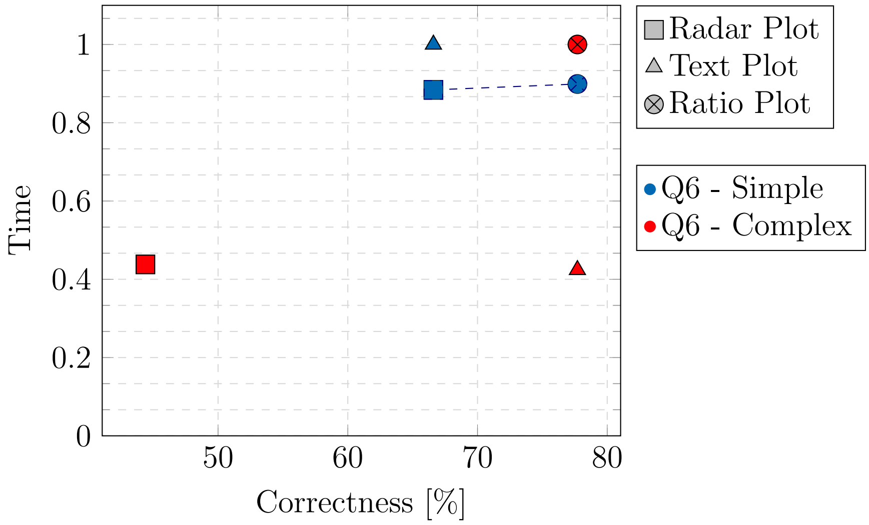


Figure 4.6: The correctness and time values for Q6 (simple and complex). The Pareto front for the different complexities is drawn by using a blue and red line respectively. The visualizations on the Pareto front for Q6 - Simple is the radar and the ratio plot and for Q6 - Complex is the text plot.

RQ3.2: How good can the visualizations be used to compare a high number of complex performance-influence models and a high number of terms?

For this sub-question, we only consider the complex use case for Q5 and Q6.

Correctness: From Table 4.1, we can infer that for Q5, all the interviewees were able to answer the question correctly when presented with the text plot. However, for Q6, the ratio plot and the text plot performed equally good, even though both did not lead to perfect correctness results.

Time Measurements: From the results presented in Table 4.2, none of the visualization techniques performed significantly better in terms of time measurements.

Difficulty Ratings: From the difficulty ratings presented in Table 4.3, we can infer that the ratio plot has higher difficulty ratings for Q5 and Q6.

Q5: From Figure 4.5, for the complex use case, we can see that if both time and correctness factors are considered the text plot performs better than other visualization technique, with perfect correctness measurements.

Q6: From Figure 4.6, for the complex use case, we can infer that the text plot performs considerably better than the ratio plot in terms of time measurements, and better than the radar plot in terms of correctness measurements.

Summary of RQ3.2: The answer to this sub-question is **yes**, with the text plot as the best visualization type for both performance-influence models with many terms and for multiple performance-influence models.

Summary of RQ3: Therefore, the answer to this research question is **yes**, with the text plot being the considerable choice of visualization technique when scalability with respect to performance-influence models is considered, since it has better results when considering both time and correctness factors and when scalability with respect to configuration options is considered, the ratio plot is the best visualization technique in general.

RQ4: What are the differences with respect to visualization techniques regarding many performance-influence models?

For the above research question and its sub-questions, we use Mann-Whitney test results and box-plot visualizations.

Mann-Whitney U Test: For this test, we use the difficulty ratings for each visualization technique and for each use case. For instance, for Q5 and the radar plot, the inputs are the difficulty ratings of the radar plot for the simple use case and the radar plot for the complex use case of Q5.

Input	**Radar Plot**	**Text Plot**	**Ratio Plot**
Q5	0.274	0.887	0.5
Q6	0.938	0.947	0.151

Table 4.5: Mann-Whitney U Test for the different visualization techniques and complexities for Q5 and Q6.

Box Plots: The box plot has the coordinates time vs simple, complex use cases per visualization type. For instance, for the radar plot, we calculate the average time taken per question, when presented with the radar plot with a simple use case and similarly for the complex use case.

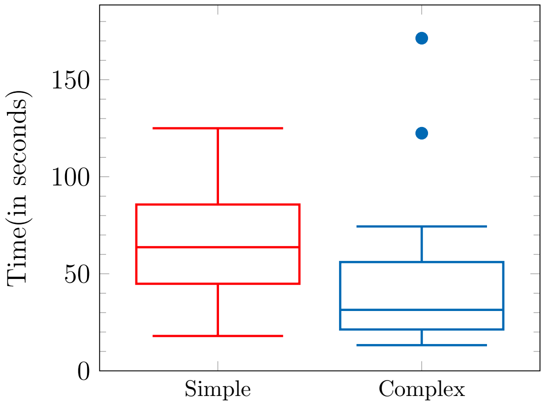


Figure 4.7: Scalability of the radar plot

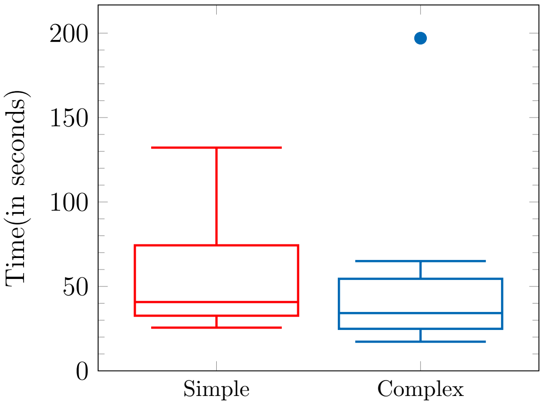


Figure 4.8: Scalability of the text plot

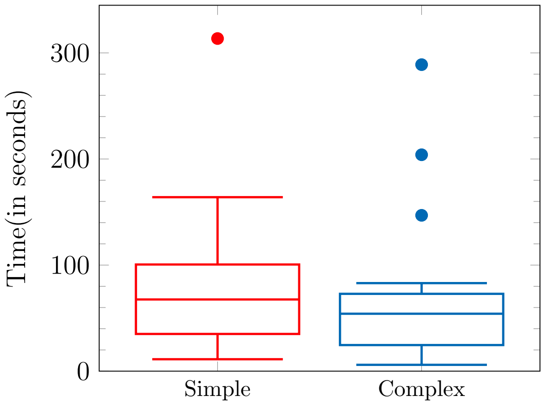


Figure 4.9: Scalability of the ratio plot

Radar Plot: From Figure 4.7, We can infer that for a simple use case with the radar plot, an interviewee on an average took 63 seconds to answer the complex performance-influence model question. We can also notice that the distribution is normal, implying that 50% of the values are below the median and 50% of values are above the median.

Similarly, for a complex use case, an interviewee takes on an average of 31 seconds to answer the complex performance-influence model question. We can also see that most of the time measurements are relatively shorter than the time measurements for the simple use case.

Text Plot: From Figure 4.8, We can infer that for a simple use case an interviewee takes approximately 40 seconds to answer the complex performance-influence model question regarding the text plot.

Similarly, for a complex use case, an interviewee took a minimum of 34 seconds to answer the question.

The ratio Plot: From Figure 4.9, we can notice that for simple use, an interviewee on an average takes 67 seconds for the complex use case to answer the question. We can also notice that the distribution is normal, implying that 50% of the values are below the median and 50% of values are above the the the median.

Similarly, for the complex use case, an interviewee took on an average of 54 seconds to answer the question, with most of the time measurements being relatively shorter, with an exception of some outliers.

RQ4.1: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of terms?

For this sub-question, we consider only Q5 since Q5 deals with scalability with respect to a number of terms.

We will evaluate this research question with a hypothesis. The hypothesis we consider is as follows:

Hypothesis: There are significant differences among the visualization techniques considering many performance-influence models having a number of terms

From the Mann-Whitney U test in Table 4.5, for Q5 we can infer that none of the visualization techniques have significant differences between the simple and the complex use case. Hence, we reject the hypothesis.

Hypothesis: Rejected

RQ4.2: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of models?

For this sub-question, we consider only Q6 since Q6 deals with scalability with respect to a number of models.

We will evaluate this research question with a hypothesis. The hypothesis we consider is as follows:

Hypothesis: There are significant differences among the visualization techniques considering many performance-influence models having a number of models

From the Mann-Whitney U test in Table 4.5, for Q6 we can infer that none of the visualization techniques have significant differences between the simple and the complex use case. Hence, we reject the hypothesis.

Hypothesis: Rejected

Summary of RQ4: From all the 3 box plots, we can infer on an average, a complex use case takes less amount of time than simple use case to answer a question regarding many performance-influence models, and hence the complex use case seems better in terms of time.

In general, to answer this research question, there are no significant differences in the visualization techniques regarding many performance-influence models.

Section 4.5: Discussion

In this section we discuss our observations and findings from the presented results.

One Performance-Influence Model

RQ1: Can we use the visualization techniques to identify relevant configuration options or interactions of one performance-influence model?

To evaluate this research question, we used the time measurements, relative correctness values and the difficulty ratings for Q1 and Q2.

RQ1.1: Can we use the visualization techniques to identify relevant configuration options or interactions of one simple performance-influence model?

For this sub-question, we considered the simple use case of Q1 and Q2.

Taking into consideration all the 3 factors; time measurements, correctness, and difficulty ratings the text plot performs considerably better than the radar plot and the ratio plot.

The text plot is better at finding the most relevant configuration option or interaction because it visualizes the data in a vertically aligned format, which means all the configuration options and interactions are visualized one below the other. Hence, the interviewee's perception is better towards the text plot and therefore, towards the vertically aligned format.

The ratio plot performs the worst and this can be seen in Table 4.3, where the average difficulty ratings are the highest. This is due to the fact that Q2 was impossible to answer with the ratio plot. However, some interviewees got a different perception and they gave a lower difficulty rating and hence, the higher deviation from the mean.

From the Table 4.4, we see that there is no strong correlation for any visualization technique. This could be due to the fact that the interviewees are looking at the visualizations for the first time and hence, their perception towards difficulty rating and time measurements do not match.

The radar plot visualizes the data in the form of bars, which does not aid the interviewee's perception when finding out the most relevant configuration option or interaction.

RQ1.2: Can we use the visualization techniques to identify relevant configuration options or interactions of one complex performance-influence model?

For this sub-question, we considered the complex use case of Q1 and Q2.

Considering all the 3 factors; time measurements, correctness and the difficulty ratings, again the text plot outperforms the other visualization techniques. The reasoning behind this is the same as given in the previous research question. Whereas, the ratio plot again proves to be the unfavorable choice of visualization technique. Since Q2, the simple use case was impossible to answer with the ratio plot, we excluded the ratio plot from the complex use case.

Two Performance-Influence Models

RQ2: Can we use the visualization to compare two performance-influence models?

To evaluate this research question, we used the time measurements, relative correctness values and the difficulty ratings for Q3 and Q4.

RQ2.1: Can we use the visualization to compare two simple performance-influence models?

For this sub-question, we considered the simple use case of Q3 and Q4.

Taking into consideration all the 3 factors; time measurements, correctness, and difficulty ratings the text plot performs better than the radar plot and the ratio plot. This is because the text plot displays the configuration option or interaction of both the performance-influence models next to each other, and it is visually easier to notice the differences or similarities between them.

The radar plot also displays the configuration option or interaction of both performance-influence models next to each other, but they are arranged in a radial form, which is not straight forward for comparisons.

Comparing performance-influence models in the ratio plot is difficult since the configuration options and the interactions are not in the same order for all the performance-influence models. This requires the user to find the corresponding configuration option or interactions to make a comparison, which influences the time measurements and the difficulty rating negatively.

RQ2.2: Can we use the visualization to compare two complex performance-influence models?

For this sub-question, we considered the complex use case of Q3 and Q4.

We notice it again that the text plot is better than the ratio plot or the radar plot. Table 4.3, reflect the difficulty levels of the 3 visualization techniques. This implies that the ratio plot is most difficult, the radar plot is moderately difficult and the text plot being the easiest one. The reason being that two performance-influence models are easier and faster to compare when they are visualized side-by-side as in the text plot. When they are visualized in the ratio plot, the same configuration option or interaction is not plotted side-by-side, their order differs based on their general performance. Hence, a comparison of the text plot is easier than on the ratio plot.

The radar plot does moderately good at the comparison since data is plotted in a circle than in a vertically aligned format.

The text plot is better since the comparison of performances in a vertically aligned format is much easier and faster than on the radar and the ratio plots.

The ratio plot took a relatively high time than other visualization techniques. This is because the connection between the influence of each configuration option or interaction between both the performance-influence model has to be established first. And doing so in the ratio plot was time consuming because of the order of the configuration option or interaction of each performance-influence models were not the same. This could be improved by sorting the terms of the performance-influence models in the same order.

Many Performance-Influence Models

RQ3: How good can the visualizations be used to compare a high number of performance-influence models and a high number of terms?

To evaluate this research question, we used the time measurements, relative correctness values and the difficulty ratings for Q5 and Q6.

Q5 is with respect to many performance-influence models with a high number of terms and Q6 is with respect to many performance-influence models with a high number of models.

RQ3.1: How good can the visualizations be used to compare a high number of simple performance-influence models and a high number of terms?

For this sub-question, we considered the simple use case of Q5 and Q6.

For scalability with a high number of terms and considering all the 3 factors; time measurements, correctness, and the difficulty rating, we can infer that the text plot is a better choice of visualization. This is due to the fact that the comparison between configuration options and interactions is easier when they are arranged next to each other for multiple performance-influence models.

However, for scalability with a high number of models, we can infer that the ratio plot is a better choice of visualization. From Table 4.3, for Q6 we can notice that the ratio plot has consistently higher difficulty ratings, and Table 4.1, tells us otherwise. This implies that even though the interviewees took a long time for the ratio plot, it leads to the comparatively higher correct answer than the radar plot or the text plot which took a lower time. This is because finding the corresponding configuration option or interaction in multiple performance-influence models is time consuming. Although, since the configuration options and interactions in the ratio plot are color coded, it aids the interviewees to make quick comparisons. The width of the bars in the ratio plot determines the amount of relative performance influence they contribute, which also aids the interviewees.

RQ3.2: How good can the visualizations be used to compare a high number of complex performance-influence models and a high number of terms?

For this sub-question, we considered the complex use case of Q5 and Q6.

For the complex use case, we find the same conclusion as that of a simple use case. I.e., the text plot is better at comparing the higher number of terms and the ratio plot is better at comparing the higher number of performance-influence models, but with a lesser number of terms.

We already know that for comparing several performance-influence models, the text plot is the best choice of visualization, since data is easier and also faster to perceive when displayed in a vertically aligned format.

For comparing many configuration options among several performance-influence models, the ratio plot proved to be better than text and the radar plot. The reason can be that when many terms are introduced, the visualization for the radar and text can get quite complex and hard to read and compare. Whereas, for the ratio plot the bars are spread out, helping the user to do comparisons. The comparisons using the ratio plot can be time consuming, but they lead to the highest correct answers than the other two visualizations.

RQ4: What are the differences with respect to visualization techniques regarding many performance-influence models?

For this research question, we introduced a hypothesis which considered that there are significant differences between the visualization techniques with many performance-influence models.

RQ4.1: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of terms?

Also, the box plots, do not show significant differences among the visualization techniques. The reason can be that the interviewee's perception almost similar for both the simple and the complex use case and for all the visualization techniques.

RQ4.2: What are the differences with respect to visualization techniques regarding many performance-influence models having a number of models?

The hypothesis is rejected for the same reason as given in RQ4.1.

Section 4.6: Threats to Validity

In this section, we present different factors that could affect the validity of the results. There can be several factors that influence the interviewee in a way that might lead to incorrect conclusions. These factors are divided into internal threats and external threats.

The perception of the interviewees is one of the internal threats to validity that biases the time measurements. It occurs when the interviewee re-reads the question which results in additional time that is not needed to answer the question, but to understand the question itself. A probable solution to this issue could that the interviewer presents an example for every new question that appears so that the interviewee understands the question well in advance.

Another internal threat can be the expectation the interviewer has with the interviewees, since there is no community, all the questions are selected in a way that a normal user of the tool would ask. A possible solution used in this interview is to get feedback on the questions asked in the interview, to know if they were meaningful and if the interviewee had any new questions or use cases.

The third internal threat is when an interviewee is answering a question with regard to the ratio plot visualization. The interviewee at some point will conclude that the ratio plot is time consuming than the radar and the text plot, or he will conclude that the said question can be answered easily with the help of radar plot or the text plot and he totally gives up on the ratio plot, which affects the time measurement part of our evaluation in a negative way. To resolve this issue, we use the difficulty ratings.

The time measurement which does not match our assumption with respect to the difficulty rating could be another internal threat. For instance, if the time taken by a certain visualization is very low, we assume that this visualization technique is easier to answer, but the corresponding difficulty rating could be high indicating that the interviewee blindly tried to answer the question.

The setup of our study could be another possible internal threat. Since we kept the order of the visualizations for each question the same, the time for understanding the question is mainly consumed in the first plot; that is, the radar plot.

One of the external threats is with regard to the community. The interview is conducted without any community behind it and hence, we cannot really generalize it. However, we tried to avoid this threat by asking the interviewees whether they could imagine other useful use cases.

Section 4.7: General Feedback

The visualizations presented in this thesis can still be enhanced with additional functionalities that will help the user of the tool to improve their perception. Therefore, we have asked our interviewees for possible feedback. The feedback is categorized as below, in an order where a functionality with a higher number of requests is at the top of the list.

- **The ratio plot with positive and negative influences:** 33% of interviewees suggested that the ratio plot be shown with positive and negative influences. The current ratio plot displays only the performance in general, and it would not help in answering the question if a positive or negative influence has to be identified. Hence the suggestion was that we have an axis in the middle, and then all the configuration options which give a positive effect are to be placed on the top of the axis and all the configuration options which give a negative effect be placed below the axis.
- **The ratio plot without sorting functionality** : 33% of interviewees mentioned that questions regarding the ratio plot needed them to compare each bar with the same in all other groups, and since the bars in each group were sorted according to their relevance it was difficult for them to find the corresponding bar in other groups. Hence, the suggestion was to remove the sorting functionality to let the bars appear in the same order in all the groups, which is very easy to compare.
- **Sorting Feature:** 33% of the interviewees suggested that the configuration options/interactions in the radar plot and the text plot be displayed in a sorted order according to their relevance, which would then help to find the most relevant option easily.
- **The ratio plot bars with contrasting colors:** Sometimes the adjacent bars in the ratio plot had colors that led to misunderstandings. Hence the feedback was to have adjacent bars with contrasting colors so that they do not look similar and can be identified easily. 22% of interviewees had this suggestion for the ratio plot.
- **Filter functionality:** 22% of the Interviewees suggested having a filter functionality on top of each visualization, where certain configuration options are filtered out whose difference falls beyond a certain selected threshold from the filter. By doing this a potential user of the tool can eliminate configuration options from the visualization that he is not interested in.
- **Consistent Visualizations:** 22% of the interviewees were confused whether the markup on hover of data points showed a group name or a configuration option name. This can be avoided by having a consistent way of representing group names and configuration option names across all three visualizations.
- **Invert red and green colors:** 11% of interviewees were confused as to why positive performance influences were marked in red, whereas negative

performance influences were marked in green. They suggested marking green for positive influence and vice versa, which would be an easy visual aid in knowing positive or negative influence.

- **Provide markup** : 11% Interviewees gave feedback on making data points stand out easily if the performance they contribute is either 0 or 1. Probably by using a different marker symbol in the visualization to show if the performance is 0 or 1. If the performance of a configuration option is 0 it is shown by an empty white marker symbol.

- **Select 2 data point to show their difference** : 11% of interviewees suggested that for the radar plot and the text plot, there should be a functionality wherein the user can select 2 data points and potentially know the difference between them. This suggested holds for more than one performance-influence model.

- **The ratio plot - sort only one group by relevance :** 11% of interviewees wanted the visualization of the ratio plot in a slightly different way. Their suggested was to display only one of the group's example:Group A, in sorted order, and all groups to follow the order of Group A. This would help in comparing the bar without having to look in the corresponding groups. This applies to visualizations with more than one performance-influence model.

5. Related Work

To the best of our knowledge, visualization specifically based on performance-influence models has not been an area of research up until now, although we have studies related to performance visualization techniques in recent past that aids developers and analysts in detecting flaws in the performance of the software by visualizing performance graphically. Visualization of performance also aids in improving the time and energy efficiency of the software.

The paper by Isaacs *et al.* [IGJ+] mainly focuses on the current state of the art of performance visualization. This paper surveys existing work on performance visualization, and categorize the goals that these performance visualization techniques can answer. The purpose of this survey is to introduce state of the art for information visualization, which aids domain experts in exploring tools and methods to analyze their data. The results from the survey are organized into areas depending on which the visualization is constructed and describe the state of art research for each area. However, the number of domain experts available is small, which made it difficult to conduct a usability study. Therefore, an alternative approach required a small number of visualization and domain experts, which were more feasible for a usability study.

Another research by Haynes *et al.* [HCR01] presents a 3-D visualization tool to visually represent the performance data from a large scale cluster for analyzing. Differing from the study by Isaacs et al., Haynes focused on a particular visualization technique and the visualization displays data in the context of complex cluster interconnection topologies. It aids analysts to discover the cause of issues ranging from communication bottlenecks to hardware errors. While Isaacs et al., did not conduct a usability study, Haynes evaluated the effectiveness of the visualization tool on clusters in two separate instances. The first instance of usability shows that using visualization can minimize the time taken to diagnose hardware problems in a large system. The second instance of usability demonstrates that visualization can provide insight for understanding systems and job performance.

A similar study by Müller *et al.* [MKJ+] presents scalability studies on performance analysis tool Vampir and VampirTrace. Unlike in the study by Haynes *et al.*, Müller focuses on scalability studies of dedicated tools in the Vampir tools family. The usability study and analysis of these tools are done on real applications taken from the SpecMPI benchmark suite. Vampir is a well known performance analysis framework. Vampir and VampirServer provide an interactive visualization of dynamic program behavior. They depend on loading the trace data to the main memory completely to begin the performance analysis session. The evaluation with VampirServer displayed good results, provided enough distributed memory. Hence, the software architecture is suitable for distributed memory platforms.

A similar work done by Bhatele *et al.* [BGI+12], aims at visualizing performance data of large scale adaptive applications. In comparison to studies by Haynes *et al.* and by Müller *et al.* this study presents a scalable visualization technique that combines hardware and communication data providing an extensive diagnosis of detailed data collected from a dynamically structured AMR library. In addition to performance measurements, the visualization assisted in the diagnosis of a scalability problem that caused a bottleneck in the AMR library. The evaluation showed that the mitigation strategy improves the performance of the AMR library by 22% for a 65,536 core run on a Blue Gene/P system.

A study which used a questionnaire as a method of evaluation of research is by Herman *et al.* [LHa16], where they identify potential differences between the performance of user towards static perspective views and interactive 3-D visualizations. Experimental tools based on web frameworks were used for this test. To identify the differences an initial questionnaire, and several training and experimental tasks were conducted. The duration of tasks (time measurements), the correctness of answers, errors and subjective evaluation of the difficulty of individual tasks were evaluated and analyzed. The results from the questionnaire and other experiments suggested that in general, the participants working with static perspective views reached better results with fewer errors and were also faster compared to interactive 3-D visualizations.

6. Conclusion

Performance-influence models are used to interpret the influence of configuration options like encryption on non-functional properties like the performance of a configurable software system. With the increasing complexity of performance-influence models, it is difficult to interpret them in their original representation. Therefore, we presented different visualization techniques to ease the interpretation of performance-influence models.

We selected 3 visualization techniques; the radar plot, the text plot, and the ratio plot. To assess the quality of these visualization techniques we conducted an interview.

Research questions were selected based on the number of performance-influence models the visualizations included. The visualizations could perform differently regarding the number of performance-influence models and, thus, we investigate how they perform with one performance-influence model, two performance-influence models, and many performance-influence models.

From the interview and their results, we found out that the text plot outperformed the radar plot and the ratio plot. The text plot presents the performance influence in a vertically aligned format and hence it is easier to perceive data than on the radar and the ratio plot.

7. Future Work

In this section, we describe ways to improve the visualization techniques presented in this work. The general feedback shown in Section 4.6 forms as a base for our future work. We describe the most important feedback and provide mockups for each of them.

Ratio plot with positive and negative influences: The Ratio plot does not distinguish between positive and negative influences of configuration options. Hence, 33% of the interviewees suggested to improve the ratio plot by having an axis separating configuration options and interactions that indicate positive and negative influence. Figure 7.1, presents the current state of the ratio plot and Figure 7.2, presents the mockup of the ratio plot with the positive and negative influences.

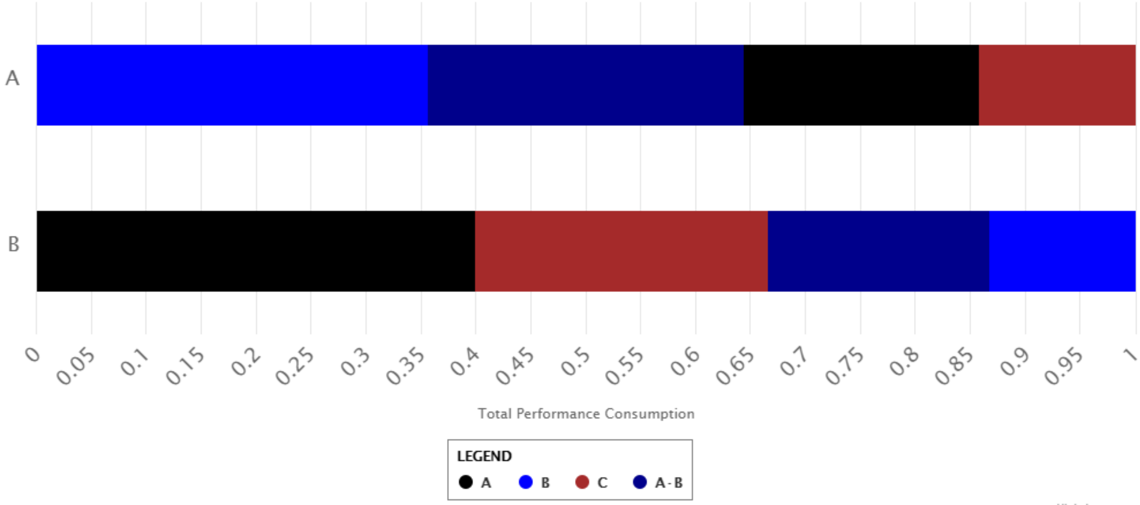


Figure 7.1: Mockup of the ratio plot with configuration options A and B and interaction A · B of 2 performance-influence models A and B

Ratio plot without sorting functionality: Another improvement on the ratio plot would be to show all configuration options and interactions in the same order

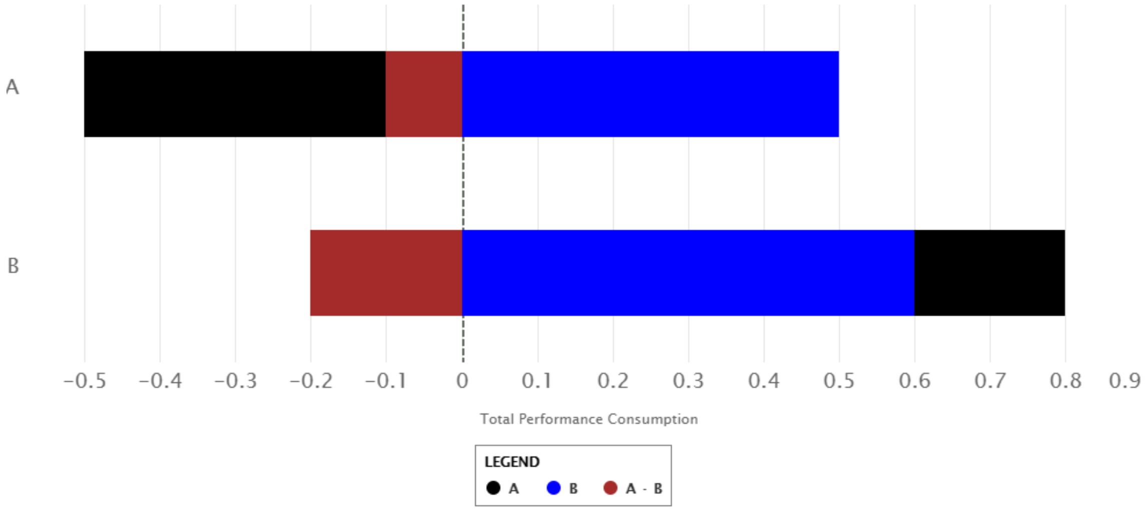


Figure 7.2: Mockup of the ratio plot with configuration options A and B and interaction A · B of 2 performance-influence models A and B. For performance-influence model A, the interaction A · B and configuration option A are on left side on the axis and configuration option B is on the right side of the axis, indicating negative and positive influences respectively. Similarly, for performance-influence model B, the interaction A · B is on the left side of the axis and configuration options A and B are on the right side of the axis

for all performance-influence models. Currently, we display the configuration options and interactions in a sorted order from highest to lowest relative performance influence for each performance-influence model. However, sorting the configuration options and interactions makes the comparison between performance-influence models more difficult. Figure 7.3, presents the mockup of the ratio plot with the feature and Figure 7.2, presents the current state of the ratio plot.

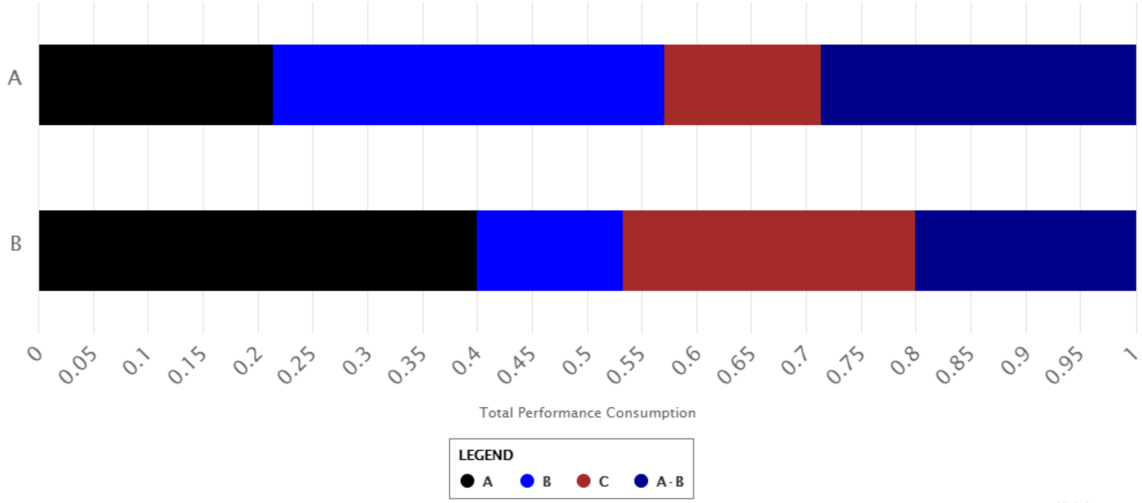


Figure 7.3: Mockup of the ratio plot with configuration options A and B and interaction A · B of 5 performance-influence models all displayed in the same order

Sorting Feature: An improvement suggested for the radar and the text plot was to present the configuration options and interactions in an order from highest to lowest relative performance influence. This would help the users perception in finding the most relevant configuration option or interaction easily. Figure 7.4, presents the current state of the radar plot and Figure 7.5, is a mockup of the revised the radar plot.

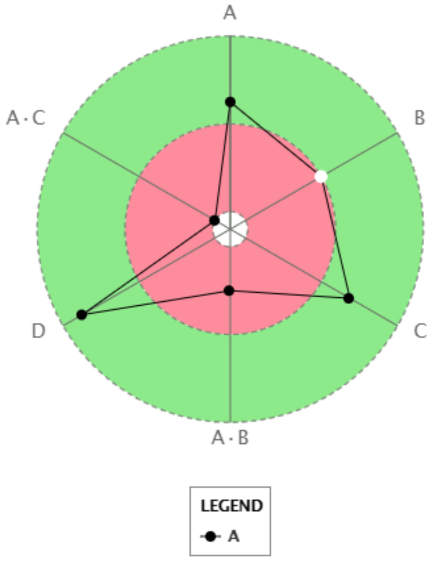


Figure 7.4: Radar plot with configuration options and interactions displayed without a definite order.

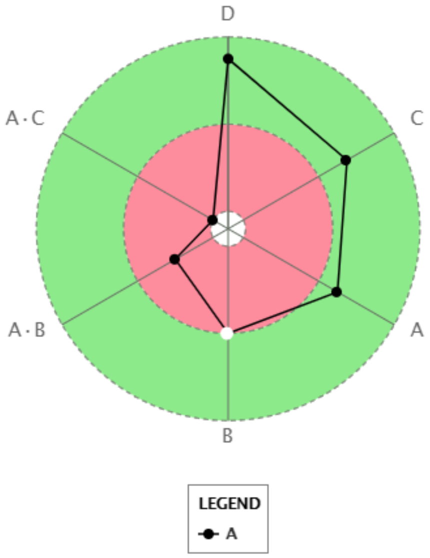


Figure 7.5: Radar plot with configuration options and interactions sorted according to the relative performance influence they contribute.

A. Appendix

Section A.1: Folder Structure of the CD

In this thesis, we introduced and presented 3 different visualization techniques that aids developers and analysts to interpret complex performance-influence models.

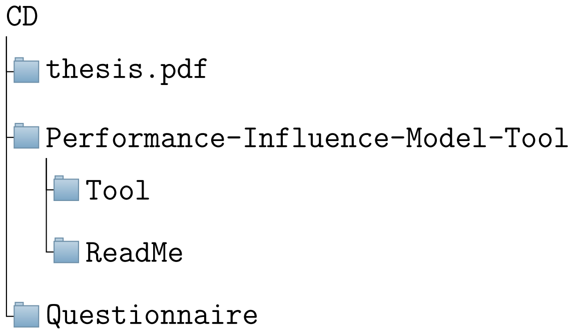


Figure A.1: Directory listing of the contents of the CD

thesis.pdf : Contains the thesis in PDF format.

Performance-Influence-Model-Tool : This folder contains the 'Visualization of Performance-Influence Model' tool. It is is written in javascript and angularJs framework. It also contains a ReadMe file to aid the reader to setup the tool.

Questionnaire : Contains the questionnaire used for the interview in PDF format.

Section A.2: Difficulty Rating vs Time measurements correlation

Sanity check is made to ensure that the assumption made from the results of difficulty rating hold true. To conduct the sanity check, we plot graphs with difficulty rating against time taken to answer a question, per interviewee.

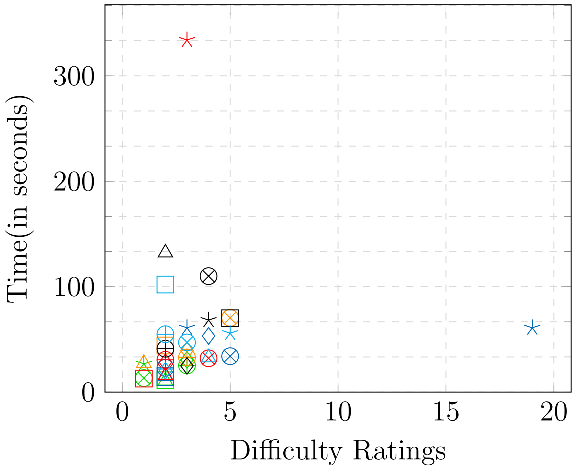


Figure A.2: Difficulty rating and time measurements for interviewee 1

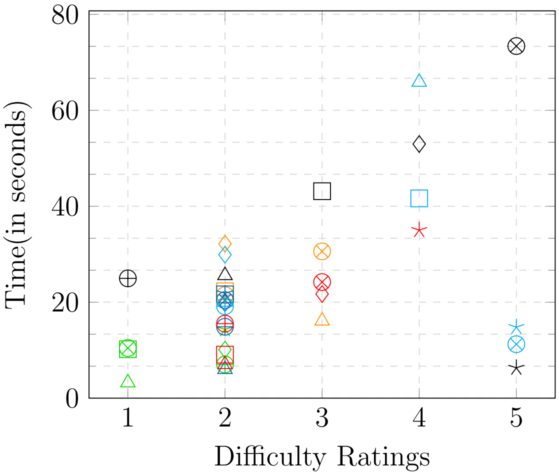


Figure A.3: Difficulty rating and time measurements for interviewee 2

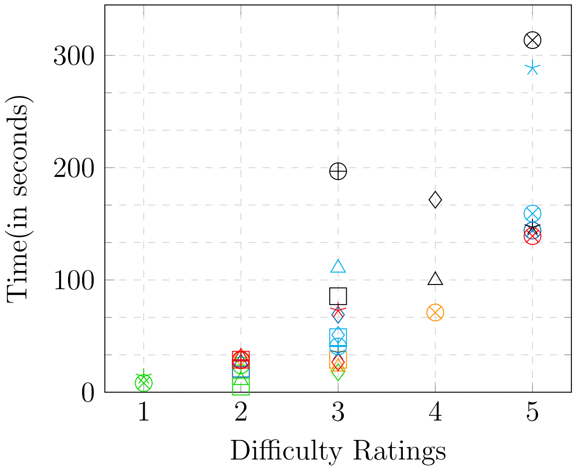


Figure A.4: Difficulty rating and time measurements for interviewee 3

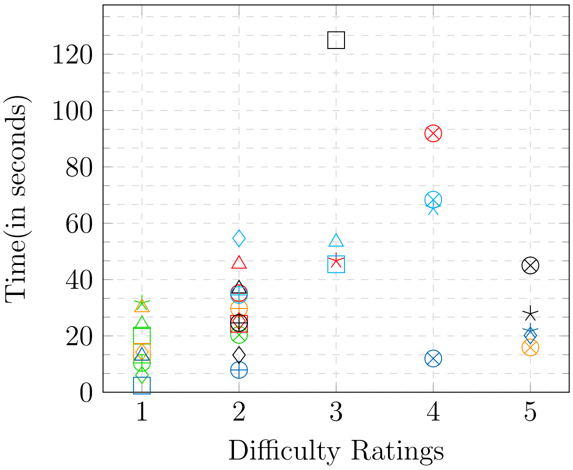


Figure A.5: Difficulty rating and time measurements for interviewee 4

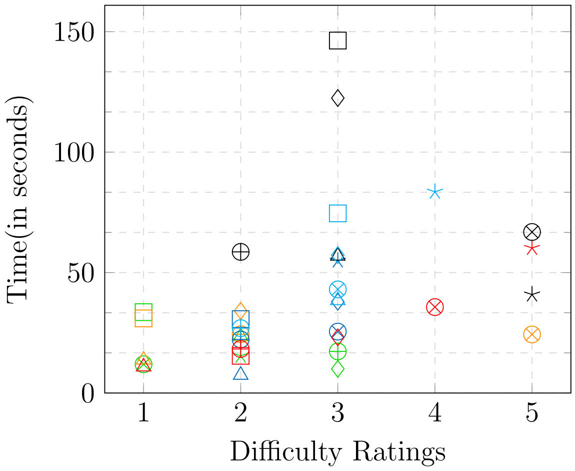


Figure A.6: Difficulty rating and time measurements for interviewee 5

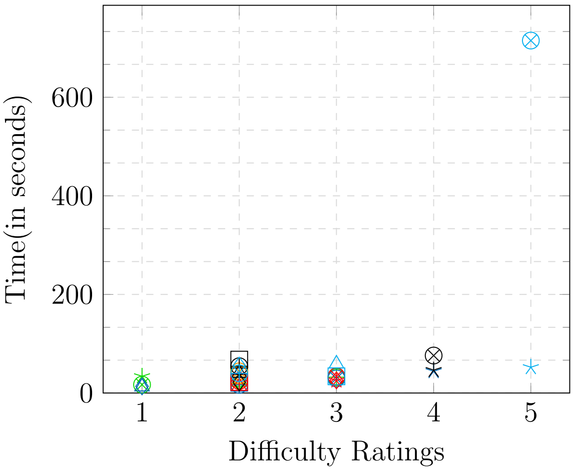


Figure A.7: Difficulty rating and time measurements for interviewee 6

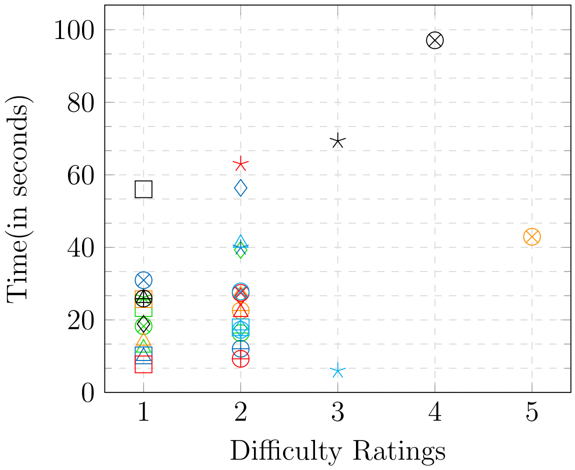


Figure A.8: Difficulty rating and time measurements for interviewee 7

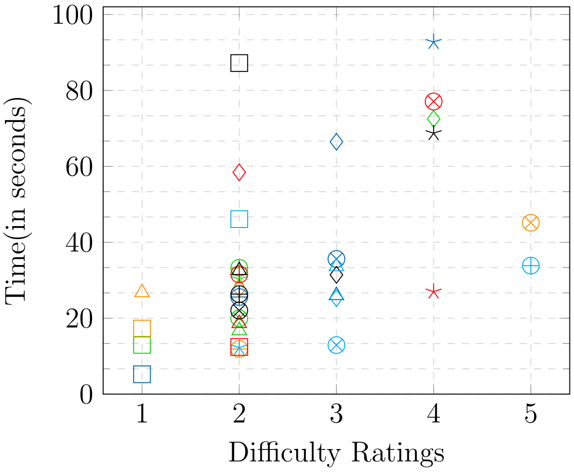


Figure A.9: Difficulty rating and time measurements for interviewee 8

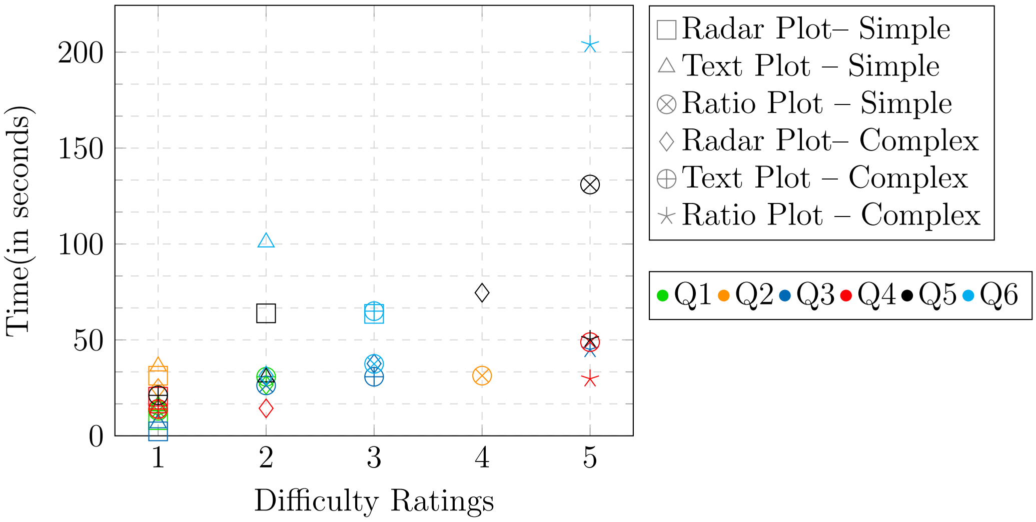
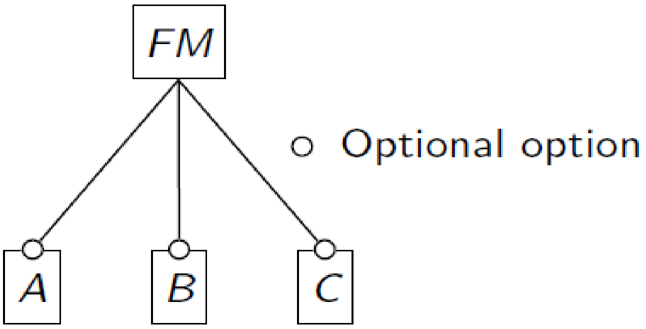


Figure A.10: Difficulty rating and time measurements for interviewee 9

Section A.3: Questionnaire

Performance-Influence Models

Warm-up Phase

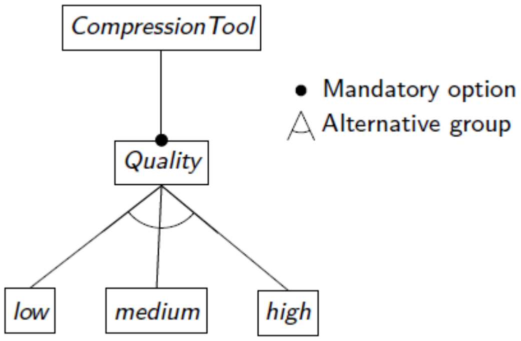


- Options: $\{A, B, C\} \equiv \mathcal{O}$
- Configurations: $\mathcal{C}$ is the set of all configurations, where $c \in \mathcal{C},\ c : \mathcal{O} \to \{0, 1\}$
  Example: $c(A) = 0$ iff option A is deselected in configuration c.
- Performance-Influence Model: $\pi : \mathcal{C} \to \mathbb{R}$

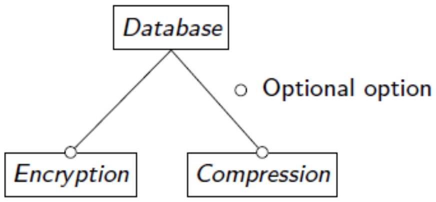
Examples:

$$\pi_1(c) = \overbrace{\underbrace{3}_{coeff.} \cdot \underbrace{c(A)}_{option}}^{Term1} \overbrace{\underbrace{+\,6}_{coeff.} \cdot \underbrace{c(B)}_{option}}^{Term2} \overbrace{\underbrace{+\,0}_{coeff.} \cdot \underbrace{c(C)}_{option}}^{Term3} \overbrace{\underbrace{-\,3}_{coeff.} \cdot \underbrace{c(A) \cdot c(B)}_{options}}^{Term4}$$

$$\pi_2(c) = 1 \cdot c(A) + 5 \cdot c(B) - 6 \cdot c(C) + 7 \cdot c(A) \cdot c(B)$$



$$\pi(c) = -3 \cdot c(low) + 5 \cdot c(high)$$



$$\pi(c) = 5 \cdot c(Compression) + 5 \cdot c(Encryption) \\ - 3 \cdot c(Compression) \cdot c(Encryption)$$

One Performance-Influence Model

Model 1: Simple Performance Model

- Radar Chart
  1. Which is the most relevant configuration option/interaction?

  Answer:

  How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  1. Which is the most relevant configuration option/interaction?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  1. Which is the most relevant configuration option/interaction?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

One Performance-Influence Model

Model 2: Complex Performance Model

- Radar Chart
  1. Which is the most relevant configuration option/interaction?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  1. Which is the most relevant configuration option/interaction?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  1. Which is the most relevant configuration option/interaction?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

One Performance-Influence Model

Model 1: Simple Performance Model

- Radar Chart

2. Which is the configuration option/interaction that leads to highest performance increase or decrease?

Answer: Option that increases performance :

Option that decreases performance :

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot

2. Which is the configuration option/interaction that leads to highest performance increase or decrease?

Answer: Option that increases performance :

Option that decreases performance :

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot

2. Which is the configuration option/interaction that leads to highest performance increase or decrease?

Answer: Option that increases performance :

Option that decreases performance :

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

One Performance-Influence Model

Model 2: Complex Performance Model

- Radar Chart
  2. Which is the configuration option/interaction that leads to highest performance increase or decrease?

Answer: Option that increases performance :

Option that decreases performance :

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  2. Which is the configuration option/interaction that leads to highest performance increase or decrease?

Answer: Option that increases performance :

Option that decreases performance :

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Two Performance-Influence Models

Model 1: Simple Performance Model

- Radar Chart
  1. Which is the configuration option/interaction where the performance-influence models differs the most?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  1. Which is the configuration option/interaction where the performance-influence models differs the most?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  1. Which is the configuration option/interaction where the performance-influence models differs the most?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Two Performance-Influence Models

Model 2: Complex Performance Model

- Radar Chart
  1. Which is the configuration option/interaction where the performance-influence models differs the most?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  1. Which is the configuration option/interaction where the performance-influence models differs the most?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  1. Which is the configuration option/interaction where the performance-influence models differs the most?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Two Performance-Influence Models

Model 1: Simple Performance Model

- Radar Chart
  2. Which is the configuration option/interaction where the performance-influence models are most similar?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  2. Which is the configuration option/interaction where the performance-influence models are most similar?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  2. Which is the configuration option/interaction where the performance-influence models are most similar?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Two Performance-Influence Models

Model 2: Complex Performance Model

- Radar Chart
  2. Which is the configuration option/interaction where the performance-influence models are most similar?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  2. Which is the configuration option/interaction where the performance-influence models are most similar?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  2. Which is the configuration option/interaction where the performance-influence models are most similar?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Many Performance-Influence Models

Model 1: Simple Performance Model

- Radar Chart
  1. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  1. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  1. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Many Performance-Influence Models

Model 2: Complex Performance Model

- Radar Chart

1. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot

1. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot

1. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult it is to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Many Performance-Influence Models

Model 1: Simple Performance Model

- Radar Chart
  2. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult is it to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  2. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult is it to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  2. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult is it to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Many Performance Influence Models

Model 2: Complex Performance Model

- Radar Chart
  2. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult is it to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Text Plot
  2. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult is it to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

- Ratio Plot
  2. Which pair of performance-influence models share a large set of influences?

Answer:

How easy/difficult is it to derive the answer?

Very Easy	Easy	Neither	Difficult	Very Difficult

Comments:

Feedback on the Interview

Were the questions in this interview meaningful?

Strongly Agree	Agree	Neither	Disagree	Strongly Disagree

Can you think about other use cases where visualizing performance-influence models is useful?

Bibliography

[AAJ+19] Muhammad A. Awad, Saman Ashkiani, Rob Johnson, Martin Farach-Colton, and John D. Owens. Engineering a high-performance GPU b-tree. pages 145–157. ACM, 2019. (cited on Page 4)

[ABKS13] Sven Apel, Don S. Batory, Christian Kästner, and Gunter Saake. *Feature-Oriented Software Product Lines - Concepts and Implementation.* Springer, 2013. (cited on Page 3)

[BGI+12] Abhinav Bhatele, Todd Gamblin, Katherine E. Isaacs, Brian T. N. Gunney, Martin Schulz, Peer-Timo Bremer, and Bernd Hamann. Novel views of performance data to analyze large-scale adaptive applications. page 31. IEEE/ACM, 2012. (cited on Page 44)

[DSC14] Don A Dillman, Jolene D Smyth, and Leah Melani Christian. *Internet, phone, mail, and mixed-mode surveys: the tailored design method.* John Wiley & Sons, 2014. (cited on Page 16)

[GS18] Przemyslaw Grzegorzewski and Martyna Spiewak. The mann-whitney test for interval-valued data. volume 642 of *Advances in Intelligent Systems and Computing*, pages 188–199. Springer, 2018. (cited on Page 19)

[HCR01] Rena A. Haynes, Patricia Crossno, and Eric Russell. A visualization tool for analyzing cluster performance data. pages 295–302. IEEE Computer Society, 2001. (cited on Page 43)

[IGJ+] Katherine E. Isaacs, Alfredo Giménez, Ilir Jusufi, Todd Gamblin, Abhinav Bhatele, Martin Schulz, Bernd Hamann, and Peer-Timo Bremer. State of the art of performance visualization. (cited on Page 6 and 43)

[JSS+] Guoliang Jin, Linhai Song, Xiaoming Shi, Joel Scherpelz, and Shan Lu. Understanding and detecting real-world performance bugs. pages 77–88. (cited on Page 4)

[KKSS18] Michael Kenzel, Bernhard Kerbl, Dieter Schmalstieg, and Markus Steinberger. A high-performance software graphics pipeline architecture for the GPU. *ACM Trans. Graph.*, 37(4):140:1–140:15, 2018. (cited on Page 4)

[LHa16] Z. Stachoň a L. Herman a, *. Comparison of user performance with interactive and static 3d visualization – pilot study. 2016. (cited on Page 44)

[MKJ+] Matthias S. Müller, Andreas Knüpfer, Matthias Jurenz, Matthias Lieber, Holger Brunst, Hartmut Mix, and Wolfgang E. Nagel. Developing scalable applications with vampir, vampirserver and vampirtrace. pages 637–644. (cited on Page 43)

[Pea96] Karl Pearson. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318, 1896. (cited on Page 23)

[SGAK] Norbert Siegmund, Alexander Grebhahn, Sven Apel, and Christian Kästner. Performance-influence models for highly configurable systems. pages 284–294. (cited on Page 1, 4, and 5)

[SKK+] Norbert Siegmund, Sergiy S. Kolesnikov, Christian Kästner, Sven Apel, Don S. Batory, Marko Rosenmüller, and Gunter Saake. Predicting performance via automated feature-interaction detection. pages 167–177. (cited on Page 4)

Eidesstattliche Erklärung:

Hiermit versichere ich an Eides statt, dass ich diese Masterarbeit selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe und dass alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind, sowie dass ich die Masterarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

I hereby certify that this master's thesis has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged. It has not been submitted in any other application for a degree.

Rima Celita Lewis

Passau, den 16. Mai 2019