Interrelating Use Cases and Associated Requirements by Links

An Eye Tracking Study on the Impact of Different Linking Variants on the Reading Behavior

Oliver Karras
Leibniz Universität Hannover
Software Engineering Group
Hannover, Germany
oliver.karras@inf.uni-hannover.de

Alexandra Risch
Leibniz Universität Hannover
Software Engineering Group
Hannover, Germany
alexandra.risch@se.uni-hannover.de

Kurt Schneider
Leibniz Universität Hannover
Software Engineering Group
Hannover, Germany
kurt.schneider@inf.uni-hannover.de

ABSTRACT

[Context:] The descriptions of interactions and system functions are two of the most important artifact types in requirements specifications. Their common notations are use cases and requirements which are related to each other. There are different variants to link a use case with its associated requirements due to a wide variety of use case templates. The main purpose of all linking variants is to highlight the interrelationships between use cases and requirements. Besides considering both artifacts for themselves, a reader needs to interrelate them to achieve a high understanding of the overall content. [Objective / Method:] Due to the effort to create and maintain links, we investigated the impact of different linking variants on the reading behavior in an eye tracking study with 15 subjects. [Results:] Our findings indicate that all investigated variants cause comparable visual effort and share the most frequent sequential reading pattern. In all cases, the use case was read first and then the requirements. Nevertheless, the different variants result in divergent reading behaviors. Especially, links embedded in the table of a use case description significantly increase the number of attention switches from the use case to the requirements. [Conclusion:] These attention switches represent the reading behavior of interrelating the use case and the associated requirements which only occurred in case of the most detailed linking variant.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in visualization**; *Visualization design and evaluation methods*;

KEYWORDS

Links, reading behavior, eye tracking, visual effort, interrelationship

ACM Reference Format:
Oliver Karras, Alexandra Risch, and Kurt Schneider. 2018. Interrelating Use Cases and Associated Requirements by Links: An Eye Tracking Study on the Impact of Different Linking Variants on the Reading Behavior. In *EASE'18: 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, June 28–29, 2018, Christchurch, New Zealand*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3210459.3210460

EASE'18, June 28–29, 2018, Christchurch, New Zealand

ACM ISBN 978-1-4503-6403-4/18/06…$15.00
https://doi.org/10.1145/3210459.3210460

1 INTRODUCTION

Requirements specifications contain several artifact types such as descriptions of interactions or system functions which are two of the most important artifact types [2, 21, 22]. Gross and Doerr [21, 22], as well as Ahrens et al. [2], found that different roles in software development, such as software architects and developers, especially focus on these descriptions when reading a specification. Common notations to represent interactions and functionalities are *fully dressed use cases* and natural language requirements [14, 19].

Fully dressed use case templates are mainly used since their predefined structure prescribes a number of elements, such as trigger, precondition, basic sequence, or extensions. Anda et al. [5] compared the use of different guidelines for writing use cases, including use case templates. The authors showed that templates are significantly more useful to write high-quality use cases, which are easy to understand by readers, than guidelines without specific details on how to document a use case [5].

Jacobson's original use case template [26] consisting of natural language descriptions embedded in a table was incomplete [16]. Coleman [16] argued that this incompletion and the absence of a UML standard resulted in a wide variety of use case templates since users were likely to invent their own format. He proposed a UML compatible template close to best practice, more complete and less ambiguous [16]. This template should be easier to use. Therefore, he included a field for non-functional requirements since use cases often involve additional information that does not fit typical use case fields [55]. However, this idea contributed to the proliferation of use case variants. Users invented further formats to add any associated information that supports a reader's understanding or are valuable for the project [14, 28]. All supplementary information should help to keep a use case precise and readable [1].

The diverse developed use case templates share typical fields such as title, preconditions, or basic sequence [5]. However, many templates include various linking variants to refer to different associated information such as functional [3], non-functional [16], or special requirements [28]. Based on literature, we identified three widely used linking variants. Besides *no linking* [5, 29, 30, 43], templates mainly include an *additional field* to list all associated requirements [6, 16, 25, 28, 34, 58] or when using a requirements management tool, e.g. DOORS, *integrated links* in typical fields refer to the associated requirements [14, 54, 55]. The main purpose of all the linking variants is to highlight the interrelationships between a use case and its associated requirements. Therefore, created links are intended to cause a specific reading behavior. A reader should follow the links to interrelate both artifacts in order to achieve a high understanding of the overall content.

Regardless of the particular linking option, creating and maintaining links in a use case cause effort. According to Wiegers [55] as well as Robertson and Robertson [38], links are mainly realized by adding the labels of the associated requirements to a use case. These labels consist of unique and persistent identification numbers.

However, this kind of link is troublesome. Basirati et al. [9] performed a case study to understand changes in use cases. Their results showed that links based on identification numbers are one specific source of risky, dispersed changes [9]. "These types of numbered references are very hard to maintain and can easily lead to wrong references" [9, p. 360]. Due to the effort to create and maintain links, we investigate the impact of the three widely used linking variants (*no linking*, *additional field*, and *integrated links*) on the reading behavior. Our research question is:

Research question:
Does the linking variant of a use case and its associated requirements influence the reading behavior in terms of visual effort and intended way of interrelating both artifacts?

We conducted an eye tracking study in a between-subjects experiment to compare the three linking variants. Each linking variant was applied to the same use case and requirements so that all used materials differ only in the respective linking option from each other. We used eye tracking since this technology enables us to capture and analyze the individual reading behavior of our subjects in detail. Based on the collected eye tracking data, we investigated the visual effort and the reading behavior of each subject for the particular linking variant. This paper contributes the following insights:

(a) The above-mentioned linking variants showed no impact on the reading behavior in terms of visual effort. We measured three metrics for visual effort based on the number of fixations, the duration of fixations, and the duration of fixations and saccades. All three metrics indicated that all variants cause comparable visual effort.
(b) In all linking variants, subjects would first read the use case and then the requirements. Nevertheless, our results show that especially the *integrated links* have an impact on the reading behavior in terms of intended way of interrelating both artifacts. We analyzed the reading behavior based on scan-paths. Our analyses indicated that a reading behavior only includes interrelating both artifacts in case of visualized links. However, only *integrated links* differ significantly from *no linking* with respect to this specific aspect of reading.

This paper is structured as follows: Section 2 discusses related work. Section 3 reports our study and documents its findings, which are discussed in section 4. Section 5 concludes the paper.

2 BACKGROUND

2.1 Linking Use Cases and Requirements

The original use case template was proposed by Jacobson [26]. Coleman [16] extended this template due to the problem that the original proposal was incomplete. He added a field for non-functional requirements. Based on Coleman's idea [16], further adapted use case templates were invented to add any associated information to a use case that improves a reader's understanding [14, 28]. We considered different research papers and books that applied use cases in their approaches and also provided a respective template. Based on this literature, we identified the three major linking variants: *no linking*, *additional field*, and *integrated links*.

2.1.1 No Linking. This variant does not consider any linking between a use case and its associated requirements. The corresponding templates do not provide any option to perform linking. The template provided by Anda et al. [5] contains only the most typical fields of a use case such as summary, preconditions, and basic sequence. Kujala et al. [29] also offered a use case template consisting of the most common fields. Schneider and Winters [43] included a use case template that does not consider any option to link a use case with its associated requirements. Kulak and Guiney [30] also provided a template without any linking option to requirements.

2.1.2 Additional Field. This variant provides an additional field to link a use case and its associated requirements. As already mentioned, Coleman [16] was the first who included an additional field in his proposed template to link a use case with its associated non-functional requirements. Araujo and Coutinho [6] also provide a template for a use case that lists all non-functional requirements that affect the use case. Besides the term of "non-functional requirements", various other terms were used to label an additional field by different researchers [25, 28, 34, 58]. Paech and Kohler [34] included a template that has an additional column for "quality requirements". Jaaksi [25] offered a template that focused on "usability requirements". The Rational Unified Process (RUP) uses a template with an included field for "special requirements" [28]. Zhou et al. [58] modified a use case template of the restricted use case modeling approach. They added a column called "feature requirement" to specify the requirement that a particular feature is associated with.

2.1.3 Integrated Links. This variant includes links to requirements in typical fields of a use case. This linking is often associated with the use of a requirements management tool, such as DOORS. Wiegers [54, 55] explained that several requirements management tools support such a detailed linking. Use cases are often linked to requirements by using uniquely and persistently labels for requirements [55]. For example, DOORS provides several levels of linking a use case with its associated requirements. Besides an additional field [14], DOORS enables the creation of links in any field.

Regardless of the particular linking option, a reader should interrelate a use case and its associated requirements to achieve a high understanding of the overall content. Therefore, all linking variants are intended to result in a similar way of reading. However, the three options cause a different amount of effort to create and maintain links. Therefore, the question comes up whether all linking variants really result in a similar reading behavior.

2.2 Reading in Software Engineering

Reading software development artifacts is one primary task in software engineering. Software reading is defined as the process by which a reader gets an understanding of the information encoded in a software development artifact sufficient to accomplish a particular task [48]. The particular task is related to the purpose of reading such as gaining knowledge, detecting defects, or implementing a design [59]. There are various software reading techniques to support a reader during his consideration of a software development artifact. A software reading technique is a sequence of steps for the individual analysis of a textual document to achieve understanding needed for a particular task. These steps are a set of instructions that guide a reader how to read a software development artifact, which areas to focus on, and which problems to look for [48, 59]. Common reading techniques are *ad-hoc reading* (AHR) [36], *checklist-based reading* (CBR) [17], *defect-based reading* (DBR) [36], *perspective-based reading* (PBR) [8], and *usage-based reading* (UBR) [50]. These reading techniques are used in particular for inspections of software development artifacts, e.g. requirements specifications, which require efficient and effective reading in order to detect defects.

AHR implies that a reader considers an artifact based on his own skills and knowledge. There is no specific method to guide a reader in order to understand the artifact. Therefore, the effectiveness of AHR strongly depends on the individual reader [36]. CBR provides a checklist which a reader uses during the consideration of a software development artifact. Such a checklist often consists of questions whose answer should help a reader to focus on specific details of an artifact [17]. DBR has the main idea of distributing different aspects of an artifact to different readers. Thus, each reader only focuses on specific aspects of an artifact while inspecting the whole document [36]. PBR utilizes different perspectives such as tester, designer or user to read a software development artifact. The different perspectives help to reduce overlap of reading by several readers since the different perspectives should focus on specific details related to the respective perspective [8]. UBR focus on the utilization of use cases to guide a reader during the consideration of an artifact. Thus, the respective document is read from the user's point of view which should support an efficient understanding in terms of user needs [50]. Besides all these reading techniques for single software development artifacts, Travassos et al. [52] proposed the *traceability-based reading* (TBR) technique to read two related software development artifacts. TBR focuses on the inspection of either the consistency of two design documents or the correctness and completeness of one design and one requirements document. However, this reading technique does not consider the comparison of two linked requirements artifacts such as *fully dressed use cases* and natural language requirements.

There is no significant conclusion on which reading technique is the best for inspecting a software development artifact. On the one hand, some empirical investigations showed that CBR, which is the most used reading technique in the software industry, is not more efficient than AHR. As for DBR, PBR, and UBR, these investigations achieved slightly better performance than CBR and AHR [8, 35, 36, 51]. On the other hand, Halling et al. [23] reported the opposite finding that CBR is better than PBR. Several other studies also showed that the different reading techniques have no significant difference [20, 33, 41].

Reading is primarily an individual effort [59]. This explains the diverging results among the empirical studies. A reader's individual performance is more dominant than the reading technique itself [53]. Hence, instead of focusing on the applied reading technique itself, there is a need for a better understanding of a reader's individual reading behavior. Therefore, we decided to investigate the impact of the three different linking variants between a use case and its associated requirements by focusing on the reading behavior of a respective reader. An objective way to characterize reading behavior is the use of eye tracking since the individual way of reading appears in a reader's eye movements. Thus, eye tracking is a powerful technology that offers multiple objective metrics to assess reading behavior in terms of visual effort and intended way of interrelating both artifacts [45].

2.3 Eye Tracking in Software Engineering

Eye tracking technology became increasingly accepted as a useful tool for empirical studies that analyze reading behavior [45]. Especially, the comparison of alternative designs and layout representations were intensively investigated to understand comprehension tasks and reading behavior.

Yusuf et al. [57] investigated the impact of several characteristics such as layout, color, and stereotypes on the comprehension of UML class diagrams. Cepeda and Guéhéneuc [13] analyzed different UML presentations of design patterns with respect to developers' comprehension. Sharif and Maletic [47] compared orthogonal and multi-clustered representations of UML class diagrams. They investigated the impact of different layouts on the comprehension of design patterns by developers. Sharafi et al. [44] focused on the efficiency of graphical vs. textual representations of a specific notation so-called TROPOS for modeling and presenting software requirements. Santos et al. [42] evaluated the effect of layout guidelines for i* goal models on novice stakeholders' ability to understand and review such models. Karras et al. [27] compared the original task board design with three customized ones. They investigated the impact of the different design alternatives on developers' work with and comprehension of a task board. Bednarik and Tukiainen [10] proposed a visualization technique for source code. They analyzed how developers use the different perspectives of normal and visualized source code interchangeably. Busjahn et al. [12] investigated the difference between source code and natural text reading with an experiment. Sharafi et al. [46] compared different representation styles of source code identifiers. They analyzed the recalling of the names of identifiers by considering the different strategies deployed by men and women. Binkley et al. [11] also focused on the impact of identifier styles on the code comprehension of developers. Romero et al. [39] analyzed the use of different representations by developers while performing debugging tasks and the impact of these representations on the developers' performance. Romero et al. [40] extended their previous work by characterizing the developers' strategies in debugging tasks based on the level of focus attention and representations use as well as in terms of the general reasoning strategy. Heymady and Narayanan [24] also studied the effectiveness and role of different representations used during source code debugging. Ali et al. [4] applied eye tracking to understand how developers verify requirements traceability links. They identified and ranked developers' preferred source code entities to define two weighting schemes to recover traceability links.

Almost all the previously mentioned eye tracking studies are included in the results of a systematic literature study on the usage of eye tracking in software engineering by Sharafi et al. [45]. Sharafi et al. [45] identified 36 relevant papers of which 86.2% were published in the last 12 years. The major three research topics of these 36 papers are *code comprehension* (12 papers), *model comprehension* (10 papers), and *debugging* (9 papers). Thus, the majority of studies mainly used source code and graphical models as objects of investigation. Only two of the 36 papers included English texts as an object of investigation in their study [45].

Although 95% of requirements documents are written in common and structured natural language, e.g. templates or forms [32], there are only a few eye tracking studies which address the comprehension of such textual representations so far.

Ahrens et al. [2] conducted an eye tracking study to analyze how software specifications are read. They identified similar patterns between paper- and screen-based reading. The results contribute awareness by considering the readers' interests based on how they use a specification. Gross and Doerr [21] performed an explorative eye tracking study to investigate software architects' information needs and expectations from a requirements specification. The results provide first insights into the relevance of certain artifact types and their notational representations. Gross and Doerr [22] extended their eye tracking study by analyzing information needs and expectations of usability experts. Based on the findings, the authors introduced the idea of a view-based requirements specification to fulfill needs of different roles in the software development.

To the best of our knowledge, Ahrens et al. [2] as well as Gross and Doerr [21, 22] performed the only eye tracking studies that focus on the investigation of textual representations of requirements engineering artifacts so far. Eye tracking is mainly used to compare design alternatives in order to investigate visual effort and reading behavior. Therefore, the use of eye tracking is ideal to investigate the three different linking variants between a use case and its associated requirements with respect to their visual effort and intended way of interrelating both artifacts.

3 EYE TRACKING STUDY

We aligned our study by following the recommendations for experimentation in software engineering by Wohlin et al. [56]. Therefore, we applied the goal definition template to ensure that the important aspects of our experiment are well-defined.

Goal definition:
We *analyze* the three different linking variants between a use case and its associated requirements *for the purpose of* evaluating their impact on the reading behavior *with respect to* visual effort and intended way of interrelating both artifacts *from the point of view of* developers *in the context of* using undergraduate and graduate students of computer science who need to understand both artifacts and their interrelationships.

3.1 Experimental Design

We selected a design science perspective resulting in a brain-based IT artifact evaluation due to our goal definition. Figure 1 illustrates the abstract and concretized process for this evaluation approach proposed by Riedl and Léger [37]. We compare design alternatives which have an impact on the brain activity by leading up to a behavioral intention that can be observed in the concrete behavior of a subject.

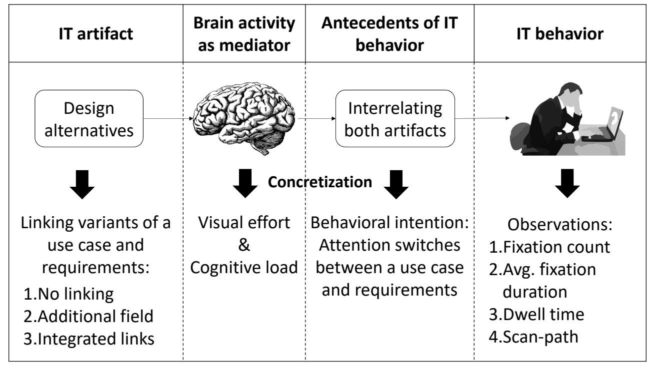

Figure 1: Brain-based IT artifact evaluation [37]

3.1.1 Hypotheses. Our study aims at answering our research question (see section 1). Thus, we test the following null hypotheses:

$H1_0$: There is no significant difference in the reading behavior of developers in terms of visual effort between the three linking variants while reading a use case and requirements to understand both artifacts and their interrelationships.

$H2_0$: There is no significant difference in the reading behavior of developers in terms of intended way of interrelating both artifacts between the three linking variant while reading a use case and requirements to understand both artifacts and their interrelationships.

Each corresponding alternative hypothesis $Hi_1, i \in \{1, 2\}$ claims that the respective difference exists. In case of such a difference, a post hoc pairwise comparisons test is necessary to determine which pairs of conditions differ significantly from one another.

3.1.2 Material. We selected a specification of a software project that developed a software to maintain students, lecturers, and lectures of an institute of sinology at a different university. Specifically, we used the use case "Enter a lecturer" and a subset of 22 requirements (11 functional and 11 quality requirements). 5 functional and 4 quality requirements were related to the use case. The use case follows the one-column template of Cockburn [14, p. 121]. It contains the 13 predefined fields by Cockburn [14] and one additional field for the links to the associated requirements. Based on this use case, we created the other two linking variants containing only the 13 predefined fields. Thus, all subjects got materials with the same content but different linking variants. Figure 2 shows exemplary snippets of the linking options *additional field* (see Figure 2a) and *integrated links* (see Figure 2b) as well as some of the requirements used (see Figure 2c). The links are realized by the enumeration of the respective labels of the associated requirements (see Figure 2, yellow markers). The variant *no linking* does not contain any links.

In total, we created a PDF file consisting of two pages for each linking variant. The first page contains the particular use case and the second page contains the 22 requirements. We divided the use case and the requirements on separate pages due to the fact that our eye tracking system does not support scrolling and both artifacts on one page would have been unreadable. The eye tracking system gathers the data for each page separately. This technical constraint does not impact the determination of the visual effort. However, we

Technology	None
Associated requirements	[R101], [R103], [R107], [107.1], [R108], [R401], [R502], [R601], [R602]

(a) Snippet of the use case description with an *additional field*

Guarantee	The client system will not be affected. A consistent state of the database will be guaranteed. [R602]	
Case of success	System saves the entered data of a lecturer in the database-	
Trigger	User selects the function "Enter a lecturer"	
Basic sequence	1	User selects the function "Enter a lecturer"
	2	System shows a window for entering the data
	3	User enters the data [R107], [R107.1], [R108]

(b) Snippet of the use case description with *integrated links*

ID	Requirements
	Functional requirements
R101	The system shall provide the user with the ability to enter a lecturer.
…	…
R107	The data set of a lecturer shall at least consist of surname, forename, and gender.
R107.1	The data set of one lecturer should contain additionally the degree, address, email address, phone number, consultation-hour, and room.
R108	The data element gender has the characteristic f for female or m for male.
…	…
	Integrity
R601	The system shall require a login with user name and password to ensure that only authorized user access the data.
R602	The database shall be consistent all time.

(c) Snippet of used requirements

Figure 2: Exemplary snippets of the used materials

needed to determine the scan-paths over both artifacts manually since our eye tracking system can only calculate a scan-path for a single page and not for a whole PDF document.

3.1.3 Independent, Dependent, and Mitigating Variables. We performed a between-subjects experiment with 3 groups, each with 5 subjects. Our independent variable is the applied linking variant with three levels: *no linking*, *additional field*, and *integrated links*.

The dependent variables are the reading behavior in terms of visual effort and intended way of interrelating both artifacts. We used three metrics for the visual effort based on the *fixation count*, the *average fixation duration*, and the *dwell time*. Besides the frequently used metrics *fixation count* and *average fixation duration* [45], we also decided to consider the *dwell time* since this time is the sum of the durations from fixations and saccades which represent actual reading time due to eye movements. For the intended way of interrelating both artifacts, we applied scan-path analysis. We focused on *attention switching frequency* between the use case and the requirements to describe the efforts to interrelate the two artifacts. We also applied sequential pattern mining [7] to identify *most frequent sequential reading patterns*. Table 1 lists all metrics.

We gathered the data about fixations, saccades, and scan-paths based on a set of 35 respectively 36 areas of interest[1] (AOIs). Each use case field and each requirement have been defined as an own AOI to get detailed information about the subjects' reading behavior.

The mitigating variables that might impact on the effect of the independent variable on the dependent variables are the level of study and the level of knowledge about use cases of our subjects.

[1]An area of interest is a relevant element in the used material.

Table 1: Overview of the applied metrics

Metrics for visual effort	
Fixation count	Number of all fixations in AOI
Avg. fixation duration	Average time of all fixations in AOI
Dwell time	Sum of durations from all fixations and saccades in AOI
Metrics for intended way of interrelating both artifacts	
Attention switching frequency	Number of attention switches between AOIs
Most frequent sequential patterns	Patterns of the most frequent subsequences within all scan-paths

3.1.4 Subjects' Demography. The study subjects were 15 volunteers, 10 undergraduate and 5 graduate students of computer science. The subjects were between their 3rd and 5th academic year, with an average academic year of 4. All of them were close to their graduation and had a similar basic level of knowledge with respect to use cases. We determined their level of knowledge based on the assessment of the statement "I have a lot of experience with use cases." on a Likert scale ranging from "strongly disagree (1)" to "strongly agree (4)". 2 subjects rated with "strongly disagree", 9 with "disagree", and 4 with "agree". 11 out of 15 subjects disagreed with the statement. Therefore, the subjects' level of knowledge about use cases was low and they formed a quite homogeneous group. This quite consistent perspective of developers helped us to encounter the mitigating variables since the level of knowledge of a subject has a strong influence on his behavior and thus the eye tracking data [49]. We randomly assigned the subjects to one of the three groups by only ensuring that the undergraduate and graduate students were distributed among all groups.

3.1.5 Setting and Procedure. We conducted the experiment in a small, quiet room. A 24-inch TFT screen with a resolution of 1920×1280 was used to show the use case and the requirements. The subject was seated approximately 60 cm away from the screen in a comfortable chair with armrests but without wheels to maintain the same seating position during the whole experiment. The subject got a keyboard to navigate between the use case and the requirements. The experimenter sat on the left side of the subject at a 90 degrees angle with a 15-inch laptop running the experimentation software. The experimenter used the laptop to conduct the experiment and to observe the subject. The 24-inch TFT screen, the keyboard, and the eye tracker were connected to the laptop.

Before running the experiment, we briefly described the experimental procedure and the eye tracking system, e.g. how it works and which data is gathered. Afterwards, the subject signed the declaration of consent to participate in our study and filled out a pre-questionnaire to gather the basic information such as level of study and knowledge. We explained the task of reading a use case and requirements to understand them and their interrelationships. We also provided a guide sheet with the task and scenario description. Each subject was put in the situation of being a developer in an ongoing project who has to understand both artifacts and their interrelationships to implement the use case.

For each subject, we first calibrated the eye tracking system with a 5-point calibration. After the successful calibration, we started the experiment, presented the respective PDF file, and collected the eye tracking data. The subject was free to decide how he read and navigate between the use case and the requirements. We set a time limit of 10 minutes for the reading task. At the end of the experiment, the subject filled out a post-questionnaire. The whole experiment took 20 minutes on average to complete.

3.1.6 Eye Tracking System. We used the RED250mobile eye tracker of SensoMotoric Instruments (SMI)[2] which supports fully automated image processing based on contact-free eye tracking and head movement compensation. It uses a binocular smart automatic tracking mode with a sampling rate of 250 Hz to capture information such as fixations and saccades. The experiment planning, capturing, and analysis was done with the software provided by SMI. The *Experiment Center* software supports the design, planning, and execution of an eye tracking experiment. The *BeGaze* software provides functions to visualize and analyze the eye tracking data which can be exported in CSV files for further statistical analysis.

3.2 Analysis and Results

We divide the analysis of the reading behavior into two parts corresponding to the two null hypotheses. First, we report the analysis and results with respect to the visual effort. Second, we present the analysis details of the intended way of interrelating both artifacts.

3.2.1 Visual Effort. We used three metrics for visual effort calculation based on different eye tracking data: number of fixations, duration of fixations, and duration of fixations and saccades. Table 2 shows the calculated overall fixation count, overall average fixation duration [ms], and overall dwell time [min] of all AOIs in the use case (*UC*), in the requirements (*RQ*), and in total ($\sum_i, i \in \{1, 2, 3\}$) of each subject. The results are grouped by the three linking variants. For each of the three dependent variables, we checked whether the data is normally distributed by applying the *Shapiro-Wilk* test. In case of a normal distribution, we performed a one-way *ANOVA* test otherwise a *Kruskal-Wallis* test on the respective in total data (see Table 2, $\sum_i, i \in \{1, 2, 3\}$) with the significance level $p = 0.05$.

We performed a *Kruskal-Wallis* test on the overall fixation counts (see Table 2, $\sum_1$). The test indicated that there was no statistically significant difference between the overall fixation counts by the different linking variants ($\chi^2 = 0.74, p = 0.69, \eta^2 = 0.05$), with a mean rank of 8.8 for *no linking*, 6.6 for *additional field*, and 8.6 for *integrated links*. Hence, we cannot reject the null hypothesis $H1_0$. Based on the overall fixation count, **there is no significant difference in the reading behavior in terms of visual effort between the three linking variants.**

The overall average fixation duration (see Table 2, $\sum_2$) was analyzed with a one-way *ANOVA* test. The analysis of variance showed that the effect of linking variant on the overall average fixation duration was not significant, $F(2, 12) = 1.51, p = 0.26, \eta^2 = 0.20$. $H1_0$ cannot be rejected. Based on the overall average fixation duration, **there is no significant difference in the reading behavior in terms of visual effort between the three linking variants.**

We investigated the overall dwell time (Table 2, $\sum_3$) with a *Kruskal-Wallis* test. The test yielded no statistically significant difference between the overall dwell times by the different linking variants ($\chi^2 = 0.56, p = 0.76, \eta^2 = 0.04$), with a mean rank of 8.4 for *no linking*, 6.8 for *additional field*, and 8.8 for *integrated links*. We cannot reject $H1_0$. Based on the overall dwell time, **there is no significant difference in the reading behavior in terms of visual effort between the three linking variants.**

3.2.2 Intended Way of Interrelating Both Artifacts. Reading behavior can be described based on the scan-path[3] of a reader. We achieved very detailed scan-paths due to the 13 respectively 14 AOIs over the entire, respective use case and the 22 AOIs covering all requirements. Links between a use case and its associated requirements highlight their interrelationships. These links are intended to cause a specific reading behavior. A reader should follow the links to interrelate the associated elements in order to understand their interrelationship. Therefore, the scan-paths provide information

[2] https://www.smivision.com/

[3] A scan-path is a sequence of visited AOIs arranged in the chronological order.

Table 2: Experiment results for visual effort – fixation count, average fixation duration [ms], and dwell time [min]

Treatment		No linking					Additional field					Integrated links				
Subject		P01	P04	P07	P10	P13	P02	P05	P08	P11	P14	P03	P06	P09	P12	P15
Fixation count	UC	721	525	188	410	594	204	295	543	700	1457	503	747	647	363	908
	RQ	479	570	325	533	535	162	416	240	313	1145	349	451	266	350	642
	$\sum_1$	1200	1095	513	943	1129	366	711	783	1013	2602	852	1198	893	713	1550
Average fixation duration	UC	138.2	138.1	105.6	142.6	133.4	111.6	158.8	128.3	144.6	114.2	113.4	130.8	109.3	109.7	116.6
	RQ	137.6	124.0	113.4	152.9	150.5	113.5	170.1	140.9	147.7	120.6	117.9	153.1	98.9	112.4	119.4
	$\sum_2$	137.9	131.1	109.5	147.7	141.9	112.5	164.5	134.6	146.2	117.4	115.6	141.9	104.1	111.0	118.0
Dwell time	UC	01:59	01:35	00:46	01:12	01:39	00:40	00:54	01:35	02:01	04:09	01:23	02:11	02:03	01:03	02:52
	RQ	01:22	01:32	00:52	01:34	01:37	00:28	01:20	00:41	00:55	03:08	00:58	01:18	00:44	00:56	01:53
	$\sum_3$	03:21	03:07	01:38	02:46	03:16	01:08	02:14	02:16	02:56	07:17	02:21	03:29	02:47	01:59	04:45

on how a subject reads and switches between the use case and the requirements. We focused on attention switching frequency and sequential pattern mining to investigate the reading behavior.

We used the attention switching frequency between all AOIs of the use case and requirements since the number of attention switches represents the subjects' efforts to interrelate both artifacts. Figure 3 represents the attention switches between the use case and the requirements of each subject grouped by the linking variants. We also determined for each attention switch whether it interrelates the last considered use case field with its associated requirement (see Figure 3, black markers). For example, the first attention switch of subject P06 from the use case (dark gray) to the requirements (bright gray) was caused by a link (black marker). In case of *no linking*, we considered whether the subjects switched from a use case field to the associated requirement on their own. The overall attention switching frequency of *integrated links* with a mean of 13.4 overall attention switches is larger compared to the other two linking variants. Whereas *no linking* has on average 2.0 overall attention switches, *additional field* has an average of 5.2 overall attention switches. Considering the directed attention switching frequency from the use case to the requirements, the average number of attention switches of *integrated links* ($M = 7.0$) is larger than the average number of attention switches of the *additional field* variant ($M = 2.8$). The *no linking* group has the smallest average number of attention switches ($M = 1.2$).

We checked the directed attention switching frequency from the use case to the requirements for normal distribution with the *Shapiro-Wilk* test since the links were only defined in this direction. The data were not normally distributed. Therefore, we performed a *Kruskal-Wallis* test with a significance level of $p = 0.05$. The test showed that there was a statistically significant difference between the directed attention switching frequencies by the different linking variants ($\chi^2 = 8.21, p = 0.02, \eta^2 = 0.55$), with a mean rank of 3.9 for *no linking*, 8.1 for *additional field*, and 12.0 for *integrated links*. Hence, we can reject the null hypothesis $H2_0$. Based on the directed attention switching frequency, **there is a significant difference in the reading behavior in terms of the intended way of interrelating both artifacts.** According to Cohen [15], the effect size value ($\eta^2 = 0.55$) indicated a large practical relevance. The post hoc pairwise comparisons test using the *Bonferroni-Dunn* test showed that the mean score for the *integrated links* condition ($M = 7.0, SD = 4.1$) was significantly different than the *no linking* condition ($M = 1.2, SD = 0.4$). However, the *additional field* condition ($M = 2.8, SD = 1.6$) did not significantly differ from the *integrated links* and *no linking* conditions. Summarized, we can say that these results show that *integrated links* have an effect on the reading behavior in terms of interrelating both artifacts. **Our results indicate that only the most detailed linking variant results in a reading behavior that includes interrelating both artifacts more intensively.**

We also calculated the ratio between attention switches from a use case field to its associated requirement and all attention switches from the use case to the requirements. The subjects of the *no linking* variant achieved a ratio of 0.0% since they did not match any use case field and associated requirement on their own. Whereas the *additional field* variant resulted in an average ratio of 22.0%, the *integrated links* option achieved an average ratio of 49.4%. In case of *integrated links*, the links caused on average every second attention switch from the use case to the requirements.

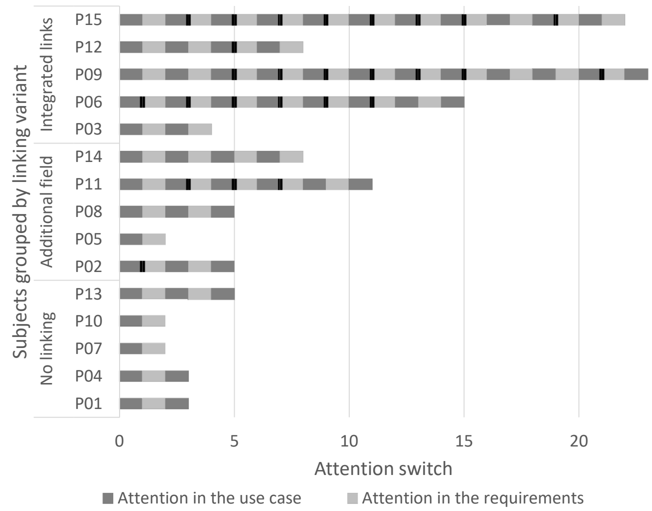

Figure 3: Attention switches

Besides the attention switching frequency, we also applied sequential pattern mining [7] on the scan-paths to identify most frequent sequential patterns in the subjects' reading behavior.

Sequential pattern mining requires sequences of symbols from a fixed itemset. However, if we use the 35 respectively 36 AOIs as an itemset in case of only 5 subjects per group the resulting sequences of the captured scan-paths are too divergent. As a consequence, they do not share any sequential pattern with a sufficient support by the sequences. Therefore, we decided to simplify the sequences by defining a smaller itemset. We defined the itemset $I = \{UCR, RQR, INT\}$ that describes the possible reading options. Either the subject reads in the use case (UCR) respectively in the requirements (RQR) or the subject interrelates (INT) both artifacts to understand their interrelationships. Three raters classified each visited AOI of each scan-path on their own as one of the three options. This classification could not be done by our subjects since they were no longer available at the time of the data analysis. We evaluated the reliability of the raters' classification by using *Fleiss' kappa* [18]. *Fleiss' kappa* is a measure of the agreement between a fixed number of raters greater than two, where agreement due to chance is factored out. The calculated *Fleiss' kappa* value was 0.85 which shows an almost perfect raters' agreement according to Landis and Koch [31]. We used three raters to achieve for each AOI a majority decision since an AOI can only be either UCR respectively RQR or INT. Based on the classification, we derived the simplified sequences by aggregating successive AOIs with the same label as one symbol of the respective label similar to Uwano et al. [53].

Figure 4 shows the resulting sequences which we used for the sequential pattern mining. For example, subject P12 reads the use case, then the requirements, and interrelates both artifacts at the end. The resulting sequence is encoded as $\langle UCR, RQR, INT \rangle$.

We performed the sequential pattern mining on the sequences presented in Figure 4 by using the *SPAM* algorithm [7] for each linking variant. A sequential pattern is a subsequence which appears in

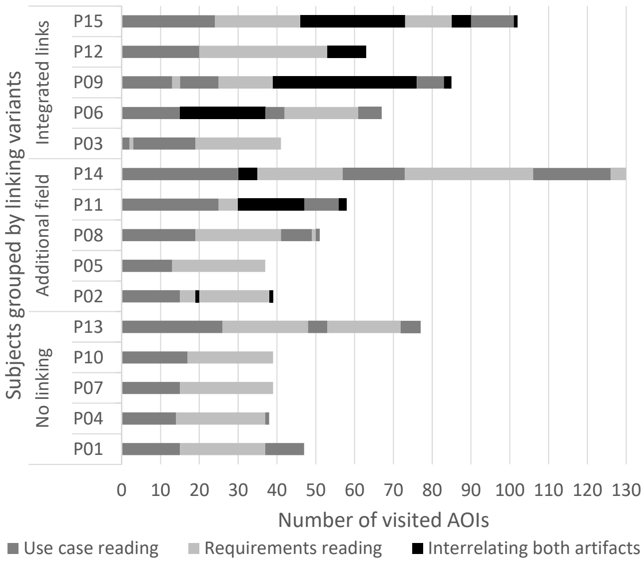

Figure 4: Simplified sequences of scan-paths

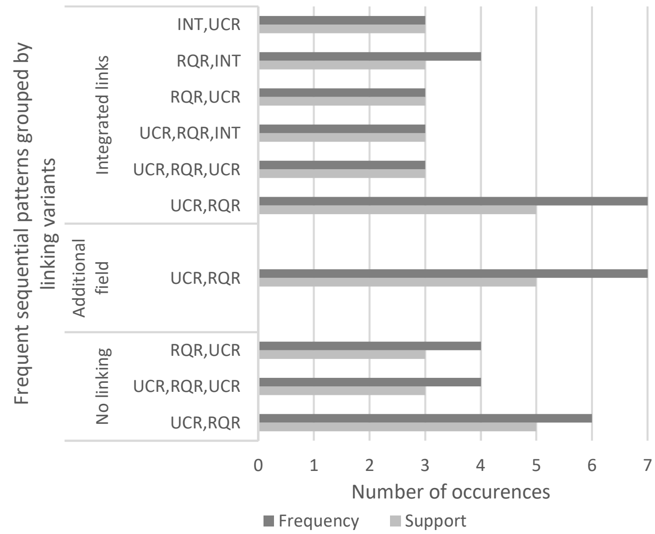

Figure 5: Frequent sequential reading patterns

at least a specified minimal number of sequences. These sequences support the identified sequential pattern. We selected a minimal support of 3 which means we decided that a frequent sequential pattern should appear in more than 50% of the analyzed sequences. Figure 5 shows the identified frequent sequential reading patterns grouped by the linking variants. All three linking variants share the most frequent sequential reading pattern $\langle UCR, RQR \rangle$ with a support of 5 per group and an overall frequency of 20. Interrelating both artifacts (INT) only appears as part of sequential reading patterns of the *integrated links* group in the subsequences $\langle UCR, RQR, INT \rangle$, $\langle INT, UCR \rangle$, and $\langle RQR, INT \rangle$ all with a support of 3 and a frequency of 3 respectively 4. The most frequent sequential reading pattern that is shared by all linking variants is reading the use case first and then the requirements. Interrelating both artifacts only occurred in case of visualized links. However, we have to restrict this finding by emphasizing that interrelating the two artifacts is only part of frequent sequential reading patterns in case of *integrated links*.

3.3 Interpretation

Our findings provide interesting insights with respect to the impact of the three linking variants on the reading behavior. All linking variants cause a comparable visual effort and share the most frequent sequential pattern of reading the use case first and then the requirements. However, only the *integrated links* result in the intended way of interrelating both artifacts according to the directed attention switching frequency and frequent sequential patterns.

Developers' reading behavior in terms of visual effort does not differ between the three linking variants. All three analyzed metrics for visual effort, which are based on the different measures, show no significant difference. Therefore, adding links consisting of unique and persistent identifiers that correspond to labels of requirements does not increase a reader's visual effort. This result is plausible since all three linking variants result in visual representations of a use case that only differ slightly. These minimal differences in the representation caused by the links do not impede the perception and thus the reading of the two artifacts.

Although there is no difference in the visual effort, we identified a significant difference in terms of the intended way of interrelating both artifacts. The overall attention switching frequency increases in case of visualized links compared to *no linking*. This finding, however, is restricted since we only identified a statistically significant difference of the directed attention switching frequency between the *integrated links* and *no linking* condition. The *integrated links* variant ($M = 7.0$) leads to 82.9% more directed attention switches from the use case to the requirements than the *no linking* option. **Therefore, the *integrated links* variant is the only linking option that results in more efforts by a reader to interrelate both artifacts.** The sequential pattern analysis supports this finding since we only identified interrelating both artifacts as a part of the frequent sequential reading patterns of the *integrated links* variant. Hence, the intended way of interrelating both artifacts is only verifiable in case of the most detailed linking variant. As an answer to our research question, we can summarize:

Answer to research question:
The three linking variants do not impact on the reading behavior in terms of visual effort since they cause comparable visual effort. They also share the most frequent sequential pattern of reading the use case first and then the requirements. Nevertheless, only the *integrated links* result in a reading behavior that includes interrelating both artifacts more intensively.

3.4 Threats to Validity

We considered threats to construct, external, internal, and conclusion validity according to Wohlin et al. [56].

3.4.1 Construct Validity. We used the same use case and requirements for all three linking variants. Thus, we have a mono-operation bias since we only used one dataset for the material of our experiment. As a consequence, the used material does not convey a comprehensive overview of the complexity in practice. Nevertheless, the material is from a real software project. Therefore, we expected that the selected material represented a sufficient realistic complexity for the subjects. We accepted this threat of a mono-operation bias to achieve a better comparability of our measurements. The analysis of reading behavior required an exact measuring. However, people are afraid of being evaluated and thus they are inclined to convey the impression of being better than they really are. This human tendency endangered the outcome of our experiment. We counteracted this threat to validity by using eye tracking for objective measurements of the subjects' behavior beyond doubt. Particularly, we used a contact-free eye tracker which can compensate head movements to mitigate the presence of the eye tracker. As a result, we created a more natural behavior by our subjects since they sat as usual in front of a monitor while working with digital documents. The single use of eye tracking caused a mono-method bias. All measurements are based on the eye tracking data and thus only allow a restricted explanation of our findings. We mitigated this threat by using multiple measures for visual effort and interrelating both artifacts in order to cross-check them. We focused on these objective measures instead of subjective ones since objectives measures are easier to reproduce and thus more reliable. We also used a post-questionnaire for subjective measures. However, these answers were not as convincing as the eye tracking data. The subjects did not remember the rationales behind their behavior and decisions due to the time between reading the artifacts and answering the post-questionnaire. The given task of reading a use case and requirements in combination with an eye tracker caused an interaction of testing and treatment. The use of an eye tracker and reading materials implies to analyze the reading behavior. Therefore, the subject could be aware of how to read the artifacts. This threat to validity is difficult to mitigate. However, we did not mention the existence and comparison of different linking variants. Thus, the subjects' attention was not focused on the links and each subject decided to use the links on his own.

3.4.2 External Validity. The selected subjects, as well as the data from a real software project, produced a good level of realism. The undergraduate and graduate students were close to their graduation. Thus, they are future developers which represent one major role that intensively works with use cases and requirements. Our subjects formed a homogeneous group due to a similar level of knowledge, experience, and age. However, the subjects' homogeneity restricts the generalizability of our results. The artificial environment and the experimental setting also endangered the external validity. Understanding the artifacts had no true pragmatic value for the subjects since none of them had a genuine working task. Future evaluations should be done on real industry projects with developers and other roles that truly work with these artifacts.

3.4.3 Internal Validity. We had three different groups due to the selected between-subjects design. These groups caused interactions with selection since different groups have a different behavior. However, we consciously decided to apply this design in order to use the same material for all groups. Thus, we counteracted possible learning effects. Furthermore, the use of eye tracking is time-consuming and exhausting for the subjects. A single session with one subject required as much as 20 minutes for reading one use case and the requirements of one particular linking variant. A session with all three linking variants is not reasonable for the subjects.

3.4.4 Conclusion Validity. Eye tracking was used to improve the reliability of our results since the use of objective measures is easier to reproduce and more reliable than subjective ones. However, eye tracking data, visual effort, and cognitive load are influenced by different factors such as knowledge, experience, and age which are difficult to control. We counteracted this threat to validity by only selecting students as subjects that were close to their graduation and with a similar level of knowledge. Hence, the subjects form a more homogeneous group which counteracts the threat of erroneous conclusions. Therefore, we mitigated the risk that the variation due to the subjects' random heterogeneity is larger than due to the investigated linking variants. Nevertheless, there is still a risk that we drew an erroneous conclusion due to the low statistical power caused by the small sample size of 15 subjects.

4 DISCUSSION

This work investigates the impact of different linking variants of a use case and its associated requirements on the reading behavior.

Use cases and requirements are two of the most important artifacts in requirements specifications. Several roles, such as software architects and developers, focus on these descriptions when reading a specification. A use case often refers to its associated requirements to support a reader's understanding of the artifacts and their interrelationships. A wide variety of use case templates exist that essentially offer three different linking variants: *no linking*, *additional field*, and *integrated links*. In general, a reader should interrelate a use case and requirements on his own to understand their interrelationships. Created links are intended to support a reading behavior that includes interrelating both artifacts. However, creating and maintaining links consisting of identification numbers cause effort. Such links are troublesome since they lead to risky, dispersed changes that increase the maintenance costs of a specification. In an eye tracking study, we analyzed the impact of the different linking variants on the reading behavior in terms of visual effort and intended way of interrelating both artifacts.

We found the insights that all three linking variants cause comparable visual effort and share the most frequent sequential pattern of reading the use case first and then the requirements. However, only *integrated links* have an impact on the reading behavior in terms of interrelating both artifacts as an inherent part of its frequent sequential reading patterns.

Our explanation of these insights is based on the *Cognitive Load Theory* by Sweller et al. [49]. The total cognitive load imposed by the used materials consists of the intrinsic and the extraneous cognitive load. While the intrinsic cognitive load depends on the nature of the materials in terms of difficulty and complexity, extraneous cognitive load depends on the representation and design of the materials. Sweller et al. [49] stated that a reader needs to process all provided materials together in order to achieve a high understanding of the overall content and its interrelationships. A successive processing of the particular artifacts only enables the understanding of the single materials but not of their interrelationships. Therefore, interrelating all provided materials is an essential part of the reading behavior in order to process and understand them.

In consideration of the *Cognitive Load Theory* [49], our used materials (*fully dressed use case* and natural language requirements) have a high element interactivity since they are strongly associated and represented in a split presentation. This high element interactivity results in a high total cognitive load. The intrinsic cognitive load is the same for all our subjects since we used the same use case and requirements for all three linking variants. Nevertheless, this load is high due to complex interrelationships between the use case and the requirements. Our observations support this assumption because no subject of the group *no linking* interrelated any use case field and corresponding associated requirement on his own. The extraneous cognitive load needs to be different between the three groups due to the distinct linking representations. One would assume that the visualized links increase the extraneous cognitive load since they need to be processed and understood. However, our results show no significant difference in the reading behavior in terms of visual effort. Hence, the visualized links do not impact on the extraneous cognitive load negatively. Instead, the physical integration of links improves the representation by supporting the element interactivity and facilitating the split-attention. The integration of links externalizes knowledge about associated elements. Thus, the links also reduce the intrinsic cognitive load since they simplify the identification of interrelationships between the use case fields and requirements. *Additional field* and *no linking* lack this explicit externalized knowledge. In contrast to the *integrated links*, the *additional field* variant is not different from the *no linking* variant. The knowledge about requirements that are associated with the whole use case is not sufficient for a reader to identify which particular use case fields refer to specific associated requirements. As a consequence, interrelating the two artifacts is not a trivial task and requires a lot of working memory resources which result in a high total cognitive load. According to our findings, only *integrated links* achieve additionally available working memory resources due to the externalized knowledge visualized by the physically integrated links. These idle resources can be used by a reader to interrelate both artifacts. Our results support this interpretation since interrelating both artifacts is only part of the frequent sequential patterns of the *integrated links* variants. Furthermore, the efforts to interrelate both artifacts in terms of directed attention switching frequency is only significantly different between the *no linking* and *integrated links* condition with an average increase of 82.9% attention switches in case of the most detailed linking variant.

The benefit of our results for the practice is the insight that the reading behavior can be influenced by the particular linking variant to achieve interrelating artifacts. If interrelating a use case and its associated requirements is necessary *integrated links* are the best option. An *additional field* does not provide the same effect. Instead, listing all links in an *additional field* leads to the same reading behavior as *no linking*. The effort to create and maintain *integrated links* should be comparable to the effort of an *additional field* since the widespread digital maintenance of links is independent of the position of a link within a use case. Therefore, the *integrated links* option is the preferred linking variant.

For the future research, our findings emphasize the need to invest more efforts for improved support of creating and maintaining *integrated links*. Suitable methods are necessary to encounter the problem of integrated links based on identification numbers that are one specific source of risky, dispersed changes of a use case. Thus, we can reduce maintenance costs and support readers by interrelating artifacts of a requirements specification more easily.

All in all, we conclude that the particular linking variant impacts on the reading behavior. Even though all three major linkings variants cause comparable visual effort and share the most frequent sequential reading pattern, only the *integrated links* caused the intended way of interrelating two artifacts according to our results.

5 CONCLUSION

The particular linking variant of a use case with its associated requirements impacts on the reading behavior.

Linking of use cases and requirements is mainly realized by using identification numbers that correspond to the labels of requirements. Besides *no linking*, these labels are either enumerated in an *additional field* or represented as *integrated links* in typical use case fields. Regardless of the applied linking variant, a reader should interrelate a use case and requirements to understand both artifacts for themselves and their interrelationships. However, creating and maintaining links is troublesome since they cause effort and can easily lead to risky, dispersed changes of use cases.

We performed an eye tracking study to investigate the previously mentioned linking variants and their impact on the reading behavior in terms of visual effort and intended way of interrelating both artifacts. Based on our results, we identified that all three linking variants cause comparable visual effort. Thus, adding links to a use case does not impede its reading. All investigated linking variants also share the most frequent sequential reading pattern. Regardless of the linking variants, all subjects read the use case first and then the requirements. However, we identified a statistically significant difference between the directed attention switches by *integrated links* and *no linking*. These results show that only *integrated links* cause more attention switches from a use case to the requirements which represent increased efforts to interrelate both artifacts. The scan-path analysis also showed that interrelating both artifacts was only part of frequent sequential patterns in case of *integrated links*.

Our work points to the conclusion that all linking variants do not impede the reading of the two artifacts for themselves. However, interrelating a use case and its associated requirements to achieve a high understanding of the overall content and its interrelationships is only supported by the detailed integration of links. Due to the effort of creating and maintaining links we recommend to prefer on the most detailed linking variant *integrated links*.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under Grant No.: 289386339, ViViReq (2017 – 2019). We follow ethical guidelines of the central ethics committee of our university. The committee regulates subject information and rights. We have to archive our data internally for future reference since recognizable persons should not be visible on distributed video.

REFERENCES

[1] Steve Adolph, Alistair Cockburn, and Paul Bramble. 2002. *Patterns for Effective Use Cases*. Addison-Wesley Longman Publishing Co., Inc.

[2] Maike Ahrens, Kurt Schneider, and Stephan Kiesling. 2016. How Do We Read Specifications? Experiences from an Eye Tracking Study. In *22nd International*

Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ). Springer International Publishing, 301–317.

[3] Ian F. Alexander and Maiden Neil. 2004. *Scenarios, Stories, Use Cases: Through the Systems Development Life-cycle*. John Wiley & Sons.

[4] Nasir Ali, Zohreh Sharafl, Yann-Gaël Guéhéneuc, and Giuliano Antoniol. 2012. An Empirical Study on Requirements Traceability Using Eye-Tracking. In *28th IEEE International Conference on Software Maintenance (ICSM)*. IEEE, 191–200.

[5] Bente Anda, Dag I. K. Sjøberg, and Magne Jørgensen. 2001. Quality and Understandability of Use Case Models. In *15th European Conference on Object-Oriented Programming (ECOOP)*. Springer-Verlag, 402–428.

[6] Joao Araujo and Paulo Coutinho. 2003. Identifying Aspectual Use Cases Using a Viewpoint-Oriented Requirements Method. In *Early Aspects 2003: Aspect-Oriented Requirements Engineering and Architecture Design*. 1–6.

[7] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. 2002. Sequential Pattern Mining Using a Bitmap Representation. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 429–435.

[8] Victor R. Basili, Scott Green, Oliver Laitenberger, Filippo Lanubile, Forrest Shull, Sivert Sørumgård, and Marvin V. Zelkowitz. 1996. The Empirical Investigation of Perspective-Based Reading. *Empirical Software Engineering* 1, 2 (1996), 133–164.

[9] Mohammad R. Basirati, Henning Femmer, Sebastian Eder, Martin Fritzsche, and Alexander Widera. 2015. Understanding Changes in Use Cases: A Case Study. In *23rd IEEE International Requirements Engineering Conference (RE)*. 352–361.

[10] Roman Bednarik and Markku Tukiainen. 2006. An Eye-Tracking Methodology for Characterizing Program Comprehension Processes. In *Symposium on Eye Tracking Research and Applications (ETRA)*. ACM, 125–132.

[11] Dave Binkley, Marcia Davis, Dawn Lawrie, Jonathan I Maletic, Christopher Morrell, and Bonita Sharif. 2013. The Impact of Identifier Style on Effort and Comprehension. *Empirical Software Engineering* 18, 2 (2013), 219–276.

[12] Teresa Busjahn, Carsten Schulte, and Andreas Busjahn. 2011. Analysis of Code Reading to Gain More Insight in Program Comprehension. In *11th Koli Calling International Conference on Computing Education Research*. ACM, 1–9.

[13] Gerardo Cepeda Porras and Yann-Gaël Guéhéneuc. 2010. An Empirical Study on the Efficiency of Different Design pattern Representations in UML Class Diagrams. *Empirical Software Engineering* 15, 5 (2010), 493–522.

[14] Alistair Cockburn. 2001. *Writing Effective Use Cases*. Addison-Wesley Reading.

[15] Jacob Cohen. 1988. Statistical Power Analysis for the Behavioral Sciences.

[16] Derek Coleman. 1998. A Use Case Template: Draft for Discussion. *Use Case Template Guidelines* (1998).

[17] Michael Fagan. 2002. Design and Code Inspections to Reduce Errors in Program Development. 575–607.

[18] Joseph L. Fleiss. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin* 76, 5 (1971), 378–382.

[19] Samuel A. Fricker, Rainer Grau, and Adrian Zwingli. 2015. Requirements Engineering: Best Practice. In *Requirements Engineering for Digital Health*. Springer International Publishing, 25–46.

[20] Pierfrancesco Fusaro, Filippo Lanubile, and Giuseppe Visaggio. 1997. A Replicated Experiment to Assess Requirements Inspection Techniques. *Empirical Software Engineering* 2, 1 (1997), 39–57.

[21] Anne Gross and Joerg Doerr. 2012. What do Software Architects Expect from Requirements Specifications? Results of Initial Explorative Studies. In *1st IEEE International Workshop on the Twin Peaks of Requirements and Architecture*. 41–45.

[22] Anne Gross and Joerg Doerr. 2012. What You Need is What You Get! The Vision of View-based Requirements Specifications. In *20th IEEE International Requirements Engineering Conference (RE)*. 171–180.

[23] Michael Halling, Stefan Biffl, Thomas Grechenig, and Monika Kohle. 2001. Using Reading Techniques to Focus Inspection Performance. In *27th EUROMICRO Conference*. 248–257.

[24] Prateek Hejmady and N. Hari Narayanan. 2012. Visual Attention Patterns During Program Debugging with an IDE. In *Symposium on Eye Tracking Research and Applications (ETRA)*. ACM, 197–200.

[25] Ari Jaaksi. 1998. Our Cases with Use Cases.

[26] Ivar Jacobson. 1993. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Pearson Education India.

[27] Oliver Karras, Jil Klünder, and Kurt Schneider. 2017. Is Task Board Customization Beneficial?. In *International Conference on Product-Focused Software Process Improvement*. Springer, 3–18.

[28] Philippe Kruchten. 2004. *The Rational Unified Process: An Introduction*. Addison-Wesley Professional.

[29] Sari Kujala, Marjo Kauppinen, and Sanna Rekola. 2001. Bridging the Gap Between User Needs and User Requirements. *Advances in Human-Computer Interaction I*, 45–50.

[30] Daryl Kulak and Eamonn Guiney. 2012. *Use Cases: Requirements in Context*. Addison-Wesley.

[31] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* (1977), 159–174.

[32] Luisa Mich, Mariangela Franch, and Pierluigi Novi Inverardi. 2004. Market Research for Requirements Analysis Using Linguistic Tools. *Requirements Engineering* 9, 1 (2004), 40–56.

[33] James Miller, Murray Wood, and Marc Roper. 1998. Further Experiences with Scenarios and Checklists. *Empirical Software Engineering* 3, 1 (1998), 37–64.

[34] Barbara Paech and Kirstin Kohler. 2004. Task-Driven Requirements in Object-Oriented Development. *Kluwer International Series in Engineering and Computer Science* (2004), 45–68.

[35] Adam A. Porter and Lawrence G. Votta. 1998. Comparing Detection Methods for Software Requirements Inspections: A Replication Using Professional Subjects. *Empirical software engineering* 3, 4 (1998), 355–379.

[36] Adam A. Porter, Lawrence G. Votta, and Victor R. Basili. 1995. Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment. *IEEE Transactions on Software Engineering* 21, 6 (1995), 563–575.

[37] René Riedl and Pierre-Majorique Léger. 2016. *Fundamentals of NeuroIS*. Springer.

[38] Suzanne Robertson and James Robertson. 2012. *Mastering the Requirements Process: Getting Requirements Right*. Addison-Wesley.

[39] Pablo Romero, Richard Cox, Benedict du Boulay, and Rudi Lutz. 2002. Visual Attention and Representation Switching During Java Program Debugging: A Study Using the Restricted Focus Viewer. *Diagrammatic Representation and Inference* (2002), 323–326.

[40] Pablo Romero, Benedict du Boulay, Richard Cox, and Rudi Lutz. 2003. Java Debugging Strategies in Multi-Representational Environments. In *15th Annual Workshop of the Psychology of Programming Interest Group (PPIG)*. 421–434.

[41] Kristian Sandahl, Ola Blomkvist, Joachim Karlsson, Christian Krysander, Mikael Lindvall, and Niclas Ohlsson. 1998. An Extended Replication of an Experiment for Assessing Methods for Software Requirements Inspections. *Empirical Software Engineering* 3, 4 (1998), 327–354.

[42] Mafalda Santos, Catarina Gralha, Miguel Goulão, João Araújo, Ana Moreira, and Joao Cambeiro. 2016. What is the Impact of Bad Layout in the Understandability of Social Goal Models?. In *24th IEEE International Requirements Engineering Conference (RE)*. 206–215.

[43] Geri Schneider and Jason P. Winters. 2001. *Applying Use Cases: A Practical Guide*. Pearson Education.

[44] Zohreh Sharafi, Alessandro Marchetto, Angelo Susi, Giuliano Antoniol, and Yann-Gaël Guéhéneuc. 2013. An Empirical Study on the Efficiency of Graphical vs. Textual Representations in Requirements Comprehension. In *21st International Conference on Program Comprehension (ICPC)*. 33–42.

[45] Zohreh Sharafi, Zéphyrin Soh, and Yann-Gaël Guéhéneuc. 2015. A Systematic Literature Review on the Usage of Eye-Tracking in Software Engineering. *Information and Software Technology* 67 (2015), 79–107.

[46] Zohreh Sharafi, Zéphyrin Soh, Yann-Gaël Guéhéneuc, and Giuliano Antoniol. 2012. Women and Men – Different but Equal: On the Impact of Identifier Style on Source Code Reading. In *20st International Conference on Program Comprehension (ICPC)*. 27–36.

[47] Bonita Sharif and Jonathan I. Maletic. 2010. An Eye Tracking Study on the Effects of Layout in Understanding the Role of Design Patterns. In *IEEE International Conference on Software Maintenance (ICSM)*. 1–10.

[48] Forrest Shull. 2002. Software Reading Techniques. *Encyclopedia of Software Engineering* (2002).

[49] John Sweller, Paul Ayres, and Slava Kalyuga. 2011. *Cognitive load Theory*. Vol. 1.

[50] Thomas Thelin, Per Runeson, and Björn Regnell. 2001. Usage-Based Reading – An Experiment to Guide Reviewers with Use Cases. *Information and Software Technology* 43, 15 (2001), 925–938.

[51] Thomas Thelin, Per Runeson, and Claes Wohlin. 2003. An Experimental Comparison of Usage-Based and Checklist-Based Reading. *IEEE Transactions on Software Engineering* 29, 8 (2003), 687–704.

[52] Guilherme Travassos, Forrest Shull, Michael Fredericks, and Victor R. Basili. 1999. Detecting Defects in Object-Oriented Designs: Using Reading Techniques to Increase Software Quality. In *14th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. 47–56.

[53] Hidetake Uwano, Masahide Nakamura, Akito Monden, and Ken-ichi Matsumoto. 2006. Analyzing Individual Performance of Source Code Review Using Reviewers' Eye Movement. In *Symposium on Eye Tracking Research and Applications (ETRA)*. ACM, 133–140.

[54] Karl E. Wiegers. 1999. Automating Requirements Management. *Software Development* 7, 7 (1999), 1–5.

[55] Karl E. Wiegers. 2003. Software Requirements: Practical Techniques for Gathering and Managing Requirement Through the Product Development Cycle. *Microsoft Corporation* (2003).

[56] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering*. Springer.

[57] S. Yusuf, H. Kagdi, and J. I. Maletic. 2007. Assessing the Comprehension of UML Class Diagrams via Eye Tracking. In *15th IEEE International Conference on Program Comprehension (ICPC)*. 113–122.

[58] Jiale Zhou, Yue Lu, Kristina Lundqvist, Henrik Lönn, Daniel Karlsson, and Bo Liwång. 2014. Towards Feature-Oriented Requirements Validation for Automotive Systems. In *22nd IEEE International Requirements Engineering Conference (RE)*. 428–436.

[59] Yang-Ming Zhu. 2016. *Software Reading Techniques*. Springer.