A Pilot Experiment to Assess Interactive OCL Specification in a Real Setting

Shaukat Ali[1], Muhammad Hammad[1], Hong Lu[1], Jan Nygård[3], Shuai Wang[1], Tao Yue[1, 2]

(The authors are alphabetically ordered)

[1]Simula Research Laboratory, Oslo, Norway, [2]University of Oslo, Oslo, Norway
[3]Cancer Registry of Norway, Oslo, Norway

{shaukat, hammad, honglu, shuai, tao}@simula.no
Jan.Nygard@kreftregisteret.no

Abstract. The Object Constraint Language (OCL) is a formal, declarative, and side-effect free language, standardized by the Object Management Group, for specifying constraints or queries on models specified in the Unified Modeling Language (UML). OCL was designed with the aim to bridge the gap between natural language and traditional formal languages requiring a strong mathematical background to understand and apply. OCL, along with UML, have been applied in practice for various purposes such as facilitating automated model-based testing. In most of such contexts of OCL, engineers with software engineering backgrounds specify OCL constraints. However, it is still a challenge for constraint authors (e.g., medical coders) who have no such background to apply OCL for other purposes (e.g., specifying medical rules). In this direction, in our previous work, we proposed a user-interactive specification framework, named iOCL, for facilitating OCL constraint specification and validation. The aim was to ease its adoption in practice in a wider application scope. In this paper, we present a pilot experiment that was conducted to assess the practical applicability of iOCL in the Cancer Registry of Norway with real users of iOCL in terms of specifying medical cancer coding rules with iOCL. Results of the pilot experiment showed that, with iOCL, time to specify OCL constraints can be significantly reduced as compared to specifying OCL constraints directly without the tool support. In addition, participants of the experiment found iOCL easy to use.

Keywords. Interactive OCL, Empirical Evaluation, Cancer Registry, Real World Application

1 Introduction

The Object Constraint Language (OCL) [5] is a standardized and declarative language, which is generally used to constrain models, for example, specified in the Unified Modeling Language (UML) [6]. Even though several tools exist for specifying, validating, evaluating, and solving OCL constraints (e.g., Eclipse OCL [8] and EsOCL [2]), modelers (researchers and practitioners) find it challenging to use OCL for specifying constraints.

To help the modelers in specifying OCL constraints, we proposed iOCL [10]—an OCL specification framework with the tool support that allows a modeler to specify constraints interactively. The underlying idea of iOCL is to guide a modeler to specify an OCL constraint interactively in the entire process of constraint specification, e.g., by presenting only relevant options to select at a given step in specification, pointing out problems in a constraint with potential solutions to fix them, and automatically correcting syntax errors, to mention but a few. In addition, iOCL integrates with other OCL tools including Eclipse OCL [8] for constraint validation and evaluation, and EsOCL [2] for solving a specified constraint. With EsOCL [2], a user gets an indication that whether the constraint specified by the modeler is solvable or not.

The iOCL tool was developed as part of an ongoing collaboration between Simula Research Laboratory [27] and the Cancer Registry of Norway (CRN) [3], where we are working together to improve the quality and productivity of cancer-related statistics produced by CRN [12]. We are applying model-based engineering in this project, where OCL constraints are used to specify medical cancer coding rules. iOCL was developed to help medical coders to specify medical rules. As a first step towards a large-scale experiment to assess the applicability of iOCL in the context of health registries, we report a pilot experiment that was conducted in CRN with four participants. We chose 10 medical cancer coding rules (specified in English text) to be defined on a domain model (in UML) and the participants were asked to specify these rules (one after another in the order of complexity) as OCL constraints using iOCL—a web-based application[1]. The time for specifying each constraint was automatically recorded on the server hosting the tool and was used for analyses reported in this paper. Before the experiment, a pre-lab questionnaire was conducted to solicit views of the participants about training given to them about iOCL. A post-lab questionnaire was conducted to solicit their views about the material used for the experiment and understandability/ease of use of iOCL.

We assessed iOCL from three perspectives: 1) Effort in terms of *Time* to specify OCL constraints, 2) Correlation of the complexity of the constraints (measured based on the metrics proposed in [11]) and *Time* to specify OCL constraints, and 3) Assessment of subjective opinions of the participants collected based on the pre and post-lab questionnaires. Based on the results, we can conclude that with iOCL, we can significantly reduce time to specify OCL constraints as compared to specifying OCL constraints without iOCL. In addition, based on subjective opinions of the participants, we can also conclude that they found iOCL easy to use. Moreover, open discussions of the experiment helped to improve the applicability and usability of iOCL.

This paper is organized as follows: Section 2 presents the background to understand the remaining sections of the paper. Section 3 presents the planning of our empirical evaluation. Section 4 provides results and analyses. Section 5 introduces the related work and we concluded the paper in Section 6.

[1] http://iocl.zen-tools.com

2 Background

In this section, we introduce the background knowledge related to this work, which includes: a brief description of the collaboration with CRN (Section 2.1), an introduction of the iOCL tool (Section 2.2) and an explanation of the metrics used to measure the complexity of OCL constraints (Section 2.3).

2.1 The MBE-CR Project with Cancer Registry of Norway (CRN)

CRN [3] gathers cancer-related information of cancer patients in Norway (e.g., diagnosis, treatment and relapse) from different medical entities (e.g., clinics, hospitals, and pathology laboratories) to support cancer research relying on cancer-related data and government to articulate public future healthcare policies. To validate the correctness of collected cancer data, a set of medical cancer coding rules (more than 1000) have been defined in CRN. At CRN, there are three key roles: 1) *Chief Medical Officers* who define and specify the cancer coding rules based on his/her medical domain knowledge; 2) *Medical Programmers* who are in charge of developing and implementing the specified cancer coding rules and 3) *Medical Coders* who use the cancer coding rules to validate the correctness of the collected cancer data.

A research project named as *MBE-CR* [12] has been established since 2015 between Simula Research Laboratory and CRN with the aim at employing model-based engineering to continuously improve the quality of the evolving automated cancer registry system and statistics it produces. In our previous work [1], we have proposed a model-based framework using UML to precisely capture domain knowledge of CRN (e.g., cancer patients, cancer messages) and use OCL to formally specify the cancer coding rules as constraints. While applying the model-based framework into the CRN's practice, we observed that it was challenging for the medical experts of CRN to manually specify the cancer coding rules as OCL constraints from scratch due to the lack of sufficient OCL knowledge. Therefore, we designed and developed iOCL [10], which will be briefly introduced in Section 2.2.

2.2 iOCL

The fundamental belief behind designing iOCL [10] is to interactively guide a modeler to specify OCL constraints (for instance, OCL operations) step by step during the constraint specification process. Precisely speaking, the core of iOCL includes three types of user operations, i.e., *selection operation*, *basic value input operation* and *text input operation*. The *selection operation* means that modelers can simply perform a selection from a list of valid options that are dynamically provided by iOCL at a given step when specifying a constraint. The *basic value input operation* denotes that modelers can input values of basic types (e.g., *Integer*, *Boolean*) based on their particular problems when specifying an OCL constraint. The *text input operation* permits to input free text without any restrictions.

To make the OCL constraint specification process easier and reduce the potential possibilities for modelers to make syntactic errors, we designed iOCL to maximize the usage of the selection operation and minimize the usage of the basic value input operation and text input operation. Notice that the selection operation does not require modelers holding OCL knowledge to a large extent since modelers can be always recalled and guided by the available valid options (e.g., OCL operations) provided by iOCL when specifying OCL constraints. In addition, iOCL is developed as a web application on the top of several existing tools, which consists of Eclipse Modeling Framework (EMF) [7], Eclipse OCL [8], Eclipse UML2 [9], and EsOCL [2]. More details on iOCL are presented in [10]. The key goal of this work is to conduct a pilot controlled experiment by involving medical experts from CRN with the aim to evaluate the applicability of iOCL for specifying cancer coding rules as OCL constraints.

2.3 Metrics to Measure the Complexity of OCL Constraints

A previous work reported in [11] defines the following four metrics to measure the complexity of OCL constraints. The first metric is the *maximum number of traversals in all the clauses of an OCL constraint* ($n_{traversals}$) that is defined to measure the maximum number of traversals from a context class to the farthest class on whose primitive attributes an OCL constraint is specified. The second metric is the *number of required attribute types* (n_{types}) to measure how many attribute types are required to specify an OCL constraint. There are in total four types of primitive attributes including *Boolean*, *Enumeration*, *Integer*, and *String*. The third metric is the *order of the complexity of the attribute types* ($O_{typesComplexity}$), which orders the complexity of the four attribute types, i.e., *Boolean*, *Enumeration*, *Integer*, and *String* from the least complex to the most complex in terms of specifying constraints. The fourth metric is the *number of clauses* ($n_{clauses}$) that measures the total number of clauses required for specifying an OCL constraint. More details with illustrations of these four metrics can be consulted in [11].

In our experiment, we ordered a given set of constraints from the least complex to the most complex based on the above-mentioned basic metrics [11]. We first ordered the constraints based on $n_{traversals}$ and if two constraints have the same $n_{traversals}$, we ordered the constraints based on n_{types}. When n_{types} is equal for two or more constraints, we further ordered the constraints based on $O_{typesComplexity}$ followed by using $n_{clauses}$.

3 Empirical Evaluation Planning

We present the planning of our pilot experiment in this section. Section 3.1 presents the overall objective of our pilot experiment and research questions and Section 3.2 presents the real-world case study and introduces the participants. Experiment materials are presented in Section 3.3. We present the experiment design in Section 3.4, whereas dependent variables are presented in Section 3.5.

3.1 Goal and Research Questions

The goal of our pilot experiment is to assess the practical applicability of iOCL in a real-world setting with real users. Accordingly, we would like to answer the following research questions.

RQ1: How much effort in terms of *Time* is required to specify OCL constraints with iOCL?

RQ2: Does the complexity of OCL constraints impact the time required to specify constraints with iOCL?

With the first research question, we wish to investigate effort in terms of time required to use iOCL for specifying OCL constraints of varying complexity by different users of variable experience and background in CRN. With the second research question, we are interested in studying the impact of the complexity of constraints on the time taken by users to specify OCL constraints with iOCL.

It is important to mention that we carefully checked both the syntactic and semantic correctness of the constraints specified by the participants and results show that all the participants specified each constraint fully correctly. Therefore, correctness of constraints is not considered in the design of the research questions.

3.2 Case Study and Participants

Our case study is a real system being implemented at CRN situated in Oslo. As part of the project, we have developed a domain model using UML capturing domain concepts, such as *Cancer Message, Cancer Case*, and *Cancer Patient*. More details related to this domain model can be consulted in [1]. We used the same domain model in this experiment for specifying medical cancer coding rules as OCL constraints.

We selected 10 cancer coding rules (written in English) of varying complexity to be specified using iOCL and the detailed cancer coding rules are shown in Table 1. More specifically, a *Cancer Message* refers to a medical record for a cancer patient from a particular medical entity (e.g., clinic hospitals) and each cancer message consists of a number of fields, e.g., *messageType* that denotes where a cancer message comes from (e.g., pathology laboratories) and *basis* indicating to what extent a patient can be diagnosed for getting a cancer.

Table 1: Ten Chosen Cancer Coding Rules

Rule No.	Cancer Coding Rules
1	Only a ***CancerMessage*** with its ***messageType*** equals to ‘D’ can have its ***basis*** equal to 90
2	If the ***basis*** of a ***CancerMessage*** is 83, then the ***messageType*** of the CancerMessage needs to be ‘K’ or ‘R’
3	The ***basis*** of a ***CancerMessage*** is from 32 to 39 and the ***messageType*** of the ***CancerMessage*** is ‘H’, then the ***surgery*** of this ***CancerMessage*** should be 96
4	If the ***topography*** of a ***CancerCase*** is ’42.0’ then the ***basis*** of the ***CancerCase*** should be one of these values: 33, 38, 45 or 47
5	If the ***basis*** of a ***CancerCase*** is 45, 46 or 47, then the ***surgery*** of the ***CancerCase*** should be 99; If the ***basis*** of a ***CancerCase*** is 2, then the ***surgery*** of the ***CancerCase*** should not be 10
6	If the ***surgery*** of a ***CancerCase*** is 14, 15, 16, 17, 18, 19, 25, 26, 28 or 29, then its ***topography*** should be 50.X
7	If the ***surgery*** of a ***CancerMessage*** is 35, then its ***basis*** should be one of these values: 57, 70,

	72, 74, 75, 76, 79 or 98
8	If the ***surgery*** of a ***CancerMessage*** is 7 and its ***messageType*** is 'O' or 'R', then its ***basis*** should be one of those values: 0, 10, 20, 29, 30, 31, 40 or 72
9	If the ***basis*** of a ***CancerMessage*** is among {57, 60, 70, 74, 75, 76, 79} and its ***messageType*** is 'H', then its ***surgery*** should neither be 96 nor 99
10	If the ***basis*** of a ***CancerMessage*** is among {33, 34, 35} then the ***messageType*** should be 'K', 'R' or 'H'; If the ***basis*** of the ***CancerMessage*** is 98 then its ***messageType*** should be 'H' or 'O'; If the ***basis*** of the ***CancerMessage*** is 22 or 72, then the ***messageType*** should only be 'K'

Four participants (the real users) participated in the experiment including a manager from CRN with 5-10 years of experience (P1), two medical coders (one with 2-5 years of experience (P2) and one with less than 1 year of experience (P3), and one medical programmer with 2-5 years of experience (P4).

In addition, several researchers were involved throughout the design, execution, and analyses of the results including a Chief Research Scientist (R1), a Senior Research Scientist (R2), Two Post-Doctoral fellows (R3 and R4), and a Research Engineer—the lead developer of iOCL (R5).

3.3 Experiment Materials

For the experiment, the ten medical cancer coding rules (as shown in Table 1) were presented to the participants right before the constraint specification task (Section 3.4) and in total, one hour was provided to the participants for specifying the ten cancer coding rules using iOCL. These constraints were further ordered based on their complexity (Section 2.3).

Furthermore, each participant used his/her own laptop, where he/she accessed the iOCL tool online. Time taken by each participant to write a constraint was automatically recorded on the server hosting iOCL and was used for analysis in this paper. At the beginning of the experiment, a pre-lab questionnaire was distributed to collect necessary background of each participant. At the end of the experiment, a post-lab questionnaire was distributed to collect subjective feedback of each participant about the experiment, their experience of using iOCL, and understandability/ease of its use. Responses to the questions of the pre- and post-questionnaires were solicited on the 5-point Likert scale. The constraints (in English) to be specified and the two questionnaires were on printed papers and handed to the participants.

3.4 Design of the Experiment

The design of the experiment is shown in Table 2, which shows that we had in total five rounds. In the first round, training was given to the participants by the researchers. The training included introducing the domain model and the iOCL tool. The first round lasted for 20 minutes. In the second round, a pre-lab questionnaire was distributed (10 minutes) to solicit views of the participants (P1-P4) about their experience and training of the domain model and iOCL. The third round included specification of the ten cancer coding rules in iOCL. In total, this round lasted for 60 minutes. In the fourth round, a post-lab questionnaire was distributed (10 minutes), with which we solicited the views of the participants about their understanding of the materials used, experiment itself, and understandability/ease to use of iOCL. Finally, an open discussion took place between the researchers and participants of the experiment about the use of iOCL, training of the experiment, and other open discussion related to iOCL. This round lasted for about 20 minutes.

Table 2: Design of the Experiment

Round	Task	Participants	Time (minutes)
1	Training of iOCL	R1-R3, R5, P1-P4	20
2	Pre-Lab Questionnaire	P1-P4	10
3	Specification of constraints (1-10)		60
4	Post-Lab Questionnaire		10
5	Open Discussion	R1-R5, P1-P4	20

3.5 Dependent Variables

The following two dependent variables are defined to analyze results of the experiment: 1) *Time* to specify a constraint (i.e., a cancer coding rule) in iOCL, 2) Complexity Order (*CO*) to indicate the position of a given constraint within a constraint set based on the complexity. Recall that the constraints were ordered based on the four metrics described in Section 2.3. For instance, a *CO* value of 2 for a constraint means the constraint is the second least complex constraint in a given constraint set.

4 Results and Analysis

In this section, we present the results and analyses corresponding to each research question.

4.1 Time Analysis (RQ1)

Table 3 presents the descriptive statistics for the mean time in seconds per constraint for the four participants. As it can be seen in Table 3, the first constraint (i.e., the simplest one) took on average 75 seconds, whereas the last constraint (i.e., the most complex one) took on average 499 seconds. For all the constraints specified by all the participants, mean time is 259.9 seconds.

Table 3: Mean Time (in Seconds) per Constraint for all the Participants

Constraint ID	1	2	3	4	5	6	7	8	9	10
Mean Time	75	154	217	192	260	436	238	224	304	499

On average for the 10 constraints, P1 took 157 seconds, P2 took 273 seconds, P3 took 351 seconds, and P4 took 254 seconds (not shown in the figure). Figure 1 shows the time taken by each participant for the 10 constraints. Notice that there are two data points missing in Figure 1 (i.e., P4 for the constraint 6 and P3 for the constraint 8) since the corresponding participants failed to specify the constraints during the experiment process. As it can be seen from the figure, constraint 1 was easy to specify by all the participants, whereas constraint 10 was the most difficult to specify since all the participants used more time. An exception can be seen from constraint 6, where the average time by the four participants is 436 second (Table 3). We cannot conclude based on the current result why this particular participant performed worse on this particular constraint. More rigorous experiment is definitely required in the future.

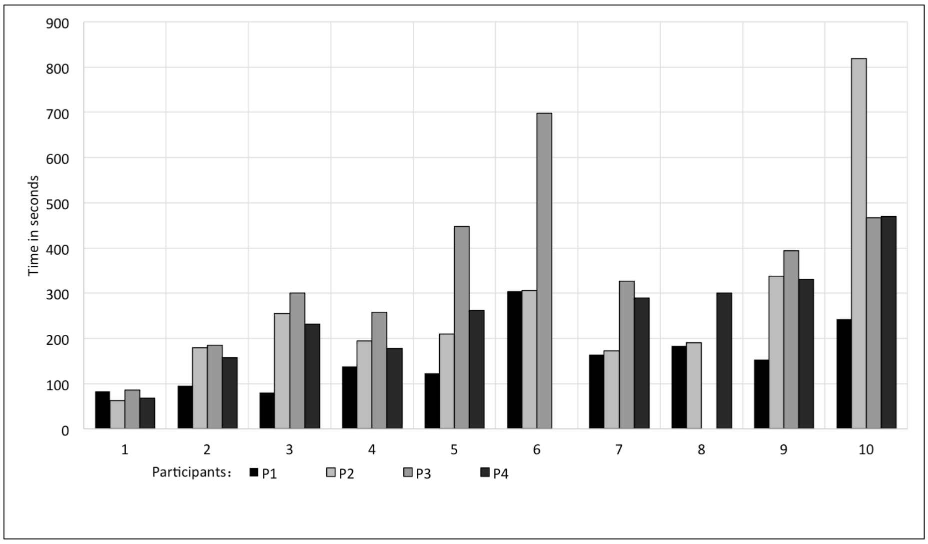

Figure 1. Time taken by each participant for the 10 constraints

In addition, we compare the participants in terms of time taken by them to specify constraints using iOCL. First, we tested the sample for normality using the Shapiro-Wilk W Test [13]. The test revealed a *p*-value of 0.0002, suggesting that the sample departs strongly from normality. Based on this, we chose the Kruskal-Wallis test [14][15], which is a non-parametric test for distributions that are not normal, to compare the four samples of time for the 10 constraints corresponding to each participant. The test revealed a *p*-value of 0.02, meaning that there are significant differences among the four participants in terms of time taken to specify constraints.

Based on this result, we compared each pair of participants with the Wilcoxon Signed Rank test [13][15], a non-parametric test that was chosen since our samples depart strongly from normality. The results are summarized in Table 4. The first two columns show the IDs of the participants being compared. The third column shows the mean difference values. A positive value means the first participant took more time, whereas a negative value means vice-versa, and a value of 0 means no difference. The last column shows the *p*-values. A *p*-value less than 0.05 (chosen confidence level) denotes a significant difference.

As shown in Table 4, we observed significant differences between P3 and P1, P2 and P1, P4 and P1 whereas P1 took significantly less time than P3, P2 and P4 (positive MD values). This could be due to the reason that P1 has much more practical experience than P2, P3 and P4 as discussed in Section 2.3.

Table 4: Comparison of *Time* across the participants*

PID1	PID2	MD (PID1-PID2)	p-value
P3	P1	6.86	**0.008**
P2	P1	5.30	**0.0452**
P4	P1	4.75	**0.0357**
P3	P2	3.48	0.1779
P4	P2	0.53	0.8383
P4	P3	-3.11	0.2164

*MD: Mean Difference, PD: Participant ID

4.2 Complexity and Time Analysis (RQ2)

We studied the correlation of complexity of constraints (measured as *CO*, Section 3.5) with time. We used the Spearman's rank correlation coefficient that is a non-parametric test [13] to study such correlation between the *CO* and *Time* required to specify the corresponding constraints. This test outputs two values, i.e., ρ (correlation coefficient) and a *p*-value. A value of ρ greater than 0 means positive correlation, whereas as a value less than 0 means a negative correlation. For *p*-value, we chose a significance level of 0.05, i.e., a value less than 0.05 means statistically significant (positive/negative) correlation. We obtained ρ of 0.697 indicating a positive correlation and a *p*-value <0.0001 indicating that the complexity of constraints and time are significantly positive correlated.

Moreover, we studied the correlation of *CO* with *Time* for each participant and results are summarized in Table 5. For all the participants (P1-P4), all the correlations are positive since all ρ are greater than 1 and all the correlations are statistically significant since the *p*-values are less than 0.05. This implies that the time taken to specify each constraint is strongly and positively correlated to the complexity of the constraint for all the participants.

Table 5: Correlation Analyses of Complexity and Time

Participant	ρ	p-value
P1	0.752	**0.0001**
P2	0.707	**0.008**
P3	0.713	**0.0001**
P4	0.886	**0.0001**

4.3 Additional Analyses based on Questionnaires

Responses to the pre and post-lab questionnaires were collected on a 5-point Likert scale, where 1 means *Strongly Agree* and 5 means *Strongly Disagree*. As it can be seen in Figure 2 that P1-P3 agreed that sufficient training was provided to them, whereas P4 neither agreed nor disagreed (score of 3).

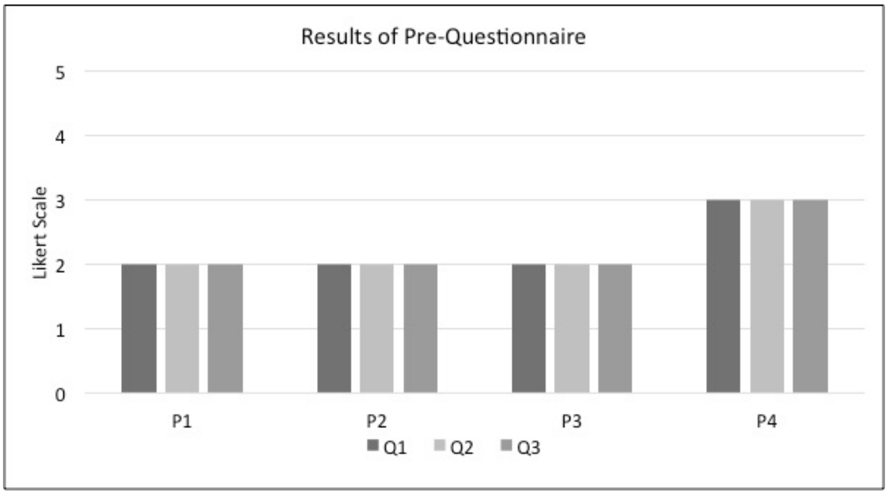

Figure 2. Results for Pre-Lab Questionnaire*

Q1: I have received sufficient training to understand the Cancer Registry domain model (UML Class Diagram) (**Training**).

Q2: I have received sufficient training regarding the tasks (**Training**)

Q3: I have received sufficient training about the iOCL tool (**Training**).

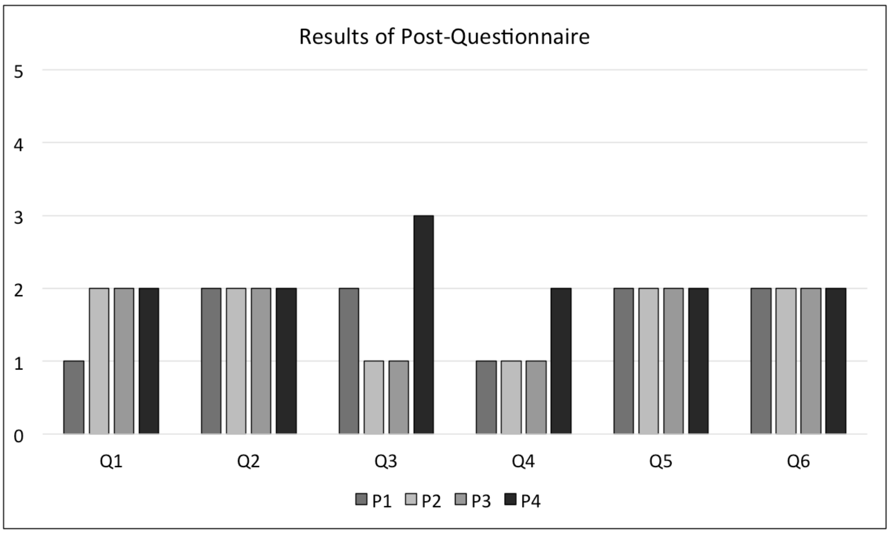

Figure 3. Results for Post-Lab Questionnaire*

Q1: The instruction of the tasks was perfectly clear to me (**Material**).

Q2: I had plenty of time to finish the tasks. (**Experiment**).

Q3: I fully understood the Cancer Registry domain model I was provided (**Material**).

Q4: I fully understood the constraints in English I was provided. (**Experiment**).

Q5: I fully understood how iOCL works **(Understandability)**.

Q6: It was easy for me to use iOCL to specify the constraints **(Ease to Use)**.

As shown in Figure 3, Q1-Q4 were related to experiment (experiment and its material). For all these questions, all the participants *Strongly Agreed* (1)/*Agreed* (2) except for P4, who neither agreed nor disagreed (score of 3). Question 5 was specifically designed to solicit views of the participants related to understanding iOCL. As we can see from the results, all the four participants agreed that they fully understood how iOCL worked. The last question was specifically designed to solicit views about *Ease of Use* of iOCL and all the four participants agreed that it was easy for them to use iOCL (score of 2).

4.4 Summary and Discussions

Based on the results reported in the previous sections, we can conclude that experience (in terms of the number of years of working in CRN) impacted the performance of a participant to specify constraints using iOCL. This was concluded based on the observation that P1 took significantly less time than P2, P3 and P4 since P1 has more practical work experience in CRN. Based on the results reported in Section 4.2, we conclude that irrespective of experience or background, time to specify constraints increased as the complexity of constraints to be specified was increased. Based on Section 4.3, we conclude that all the participants understood iOCL fully and found iOCL easy to use.

Prior to this experiment, we conducted another pilot experiment, where we studied specification of OCL constraints without using iOCL in CRN. We used four OCL constraints of varying complexity with three participants in CRN. On average, each participant took 600 seconds to correctly specify these four constraints with pens and papers. As discussed in Section 4.1, with iOCL, mean time to specify all the constraint was 259.9 seconds. Thus, we compared the results of not using iOCL with the results of using iOCL, by conducting the Wilcoxon Signed Rank test [13][15]. The result of the test revealed a p-value less than 0.0001, meaning that using iOCL to specify constraints can significantly reduce time as compared to not using iOCL. However, more systematic experiments are required to be conducted to confirm these results.

As shown in Table 2, the last round of the experiment is the open discussion session. During that session, we informally asked questions about the overall experience of using iOCL, and their expectations on the adoption of iOCL in their current practice. Two important aspects that were raised by the participants are that iOCL needs to be tailored to better accommodate the domain specific requirements and improve the usability of the tool. The current online version of iOCL has accommodated the issues raised in the open discussion session.

4.5 Threats to Validity

Conclusion validity threats are related to the factors that can affect the conclusions derived from the results of experiments [17]. In our case, the main threat is related to the sample size used to derive conclusions. We had roughly 10 data points per participant (four in total) for time to specify 10 constraints in iOCL and in total 38 data points. Two data points were missing due to the fact that the participants failed to specify the constraints. Indeed, we need more participants to further increase the sample size to strengthen our conclusions. In addition, we used appropriate statistical tests based on the analysis of sample relying on the guidelines proposed in [16][17].

The *internal validity* threats are related to the internal factors [16][17] that can possibly affect the outcomes of results. In our case, each participant worked on his/her own constraint independent of the other participants and thus didn't impact the performance of each participant.

In our pilot, there are some possible *construct validity* threats. For example, in this experiment, we didn't study all the possible constructs of OCL specifications and much larger number of constraints is required to study all the constructs of the OCL specification. However, notice that our constraints are real-world constraints and they reflect the complexity of real medical rules at CRN.

In terms of *external validity* threats, like any other experiment [16][17], we need more case studies to generalize the results. Conducting experiments with additional constraints in CRN is our near future work. In addition, we are also planning to conduct a large-scale controlled experiment with students to assess the applicability of iOCL. This is mainly due to the reason that conducting experiments in a real setting is very expensive.

5 Related Work

There are several existing tools for OCL constraint specification, evaluation and validation, e.g., 1) IBM RSA [24] and Papyrus [23] that provide an integrated modeling environment for modelers to specify constraints in UML models using various languages (e.g., OCL and Java); 2) Eclipse OCL [8] that was designed to specify, validate and evaluate OCL constraints and 3) EsOCL [2] that was developed to automatically generate model instances that comply with the specified OCL constraints using search algorithms. As compared with the existing tools related with OCL, iOCL poses two key differences, which include: 1) the focus is different, i.e., iOCL is mainly designed for easing the process of OCL constraint specification by providing modelers with runtime guidance in an interactive manner; and 2) iOCL integrates several existing OCL tools and supports their functionalities, i.e., OCL constraint evaluation and validation provided by Eclipse OCL [8] and automatic generation of model instances that meet a given set of OCL constraints provided by EsOCL [2] to assess whether the specified constraint can be solved.

Recall that OCL is based on first-order logic and set theory, which has been widely applied as a standard language for specifying various constraints on UML models [5]. The state-of-the-art has shown promising results in terms of applying OCL to solve different industrial problems [4][20]-[22], e.g., model-based test case generation [20]. The existing literature has conducted several controlled experiments with the aim of evaluating the potential benefits of applying OCL [11][18][19] [25][26]. For instance, to collect the evidence of the applicability of OCL, Yue et al. [11] involved 29 trained graduate students and conducted a controlled experiment by comparing OCL and Java in terms of specifying constraints on UML models. Results of the experiment showed that 1) OCL managed to achieve equivalent performance for constraint specification with respect to completeness, conformance, and redundancy; and 2) OCL scaled well even for specifying constraints with high complexity, which was not the case for Java. Briand et al. [18][19] performed a controlled experiment to evaluate the impact of applying OCL in UML-based development from the perspective of model comprehension and maintainability. Their results show that practitioners usually require a learning curve to acquire sufficient knowledge of OCL for gaining benefits when using OCL for constraint specification on UML diagrams. Correa et al. [25] conducted and reported a controlled experiment in terms of evaluating the impact of OCL refactoring with respect to the understandability. Their results show that most of the refactoring operators were able to significantly improve the understandability of OCL specifications.

As compared with the existing work related to the controlled experiments of OCL, the main difference is that we focus on evaluating the applicability of iOCL into practice to ease constraint specification. In particular, we assessed its applicability in the CRN domain with four medical experts from CRN (i.e., healthcare domain) for the experiment.

6 Conclusion and Future Work

In the past, we were frequently challenged by the applicability of the Object Constraint Language (OCL) in practice. Evidence from the literature has shown that OCL is easy to apply in certain contexts and it is also well recognized in the community that tool support is considered as a very important factor to foster the adoption of OCL in practice. In our context of applying OCL in the Cancer Registry of Norway (CRN) for specifying medical cancer coding rules, we also faced the challenges of introducing OCL to their practice, which inspired us to design a user-friendly OCL specification, validation and evaluation tool (coined as iOCL). To ensure the applicability of iOCL in CRN, we conducted a pilot experiment to obtain the initial observations, together with the real future users of the tool, i.e., the experiment participants who play as medical experts in CRN. The participants were also positive about the adoption of iOCL in their practice.

In the future, we plan to conduct more rigorous experiments to evaluate the applicability of iOCL in other practical contexts. We also plan to conduct controlled experiments in an academic setting for the purpose of evaluating the applicability of iOCL in a general context.

Acknowledgement. This research was supported by RFF Hovedstaden funded MBE-CR project. Shaukat Ali and Tao Yue are also funded by EU Horizon 2020 funded project (Testing Cyber-Physical Systems under Uncertainty), RCN funded Zen-Configurator project, RCN funded MBT4CPS project, RCN funded Certus SFI, and IC1404 COST Action. Shuai Wang is also funded by RCN funded Certus SFI and Hong Lu is funded by RCN funded Zen-Configurator project.

References

[1]. Wang, S., Lu, H., Yue, T., Ali, S., Nygård, J.: MBF4CR: A Model-Based Framework for Supporting an Automated Cancer Registry System. The European Conference on Modelling Foundations and Applications, pp. 191-204. Springer, (2016)
[2]. Ali, S., Iqbal, M.Z., Arcuri, A., Briand, L.C.: Generating test data from OCL constraints with search techniques. IEEE Transactions on Software Engineering 39, 1376-1402 (2013)
[3]. Cancer Registry of Norway (CRN), https://www.kreftregisteret.no
[4]. Wang, S., Ali, S., Gotlieb, A., and Liaaen, M.: A systematic test case selection methodology for product lines: results and insights from an industrial case study. Empirical Software Engineering, 21(4), pp. 1586-1622. (2016)
[5]. Object Constraint Language (OCL), http://www.omg.org/spec/OCL/
[6]. Unified Modeling Language (UML), http://www.omg.org/spec/UML/2.5/
[7]. Eclipse Modeling Framework (EMF), https://eclipse.org/modeling/emf/
[8]. Eclipse OCL, http://wiki.eclipse.org/OCL
[9]. Eclipse UML2, http://wiki.eclipse.org/MDT-UML2
[10]. Hammad, M., Yue, T., Ali, S., Wang, S.: iOCL: An Interactive Tool for Specifying, Validating and Evaluating OCL Constraints. ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS) Tool Demonstration Track, pp.24-31. (2016)
[11]. Yue, T. and Ali, S.: Empirically evaluating OCL and Java for specifying constraints on UML models. Software & Systems Modeling journal, vol 15(3), pp. 757-781, (2016).
[12]. MBE-CR Project. Link: https://www.simula.no/research/projects/mbe-cr-innovative-approach-longstanding-development-and-maintenance-automated
[13]. Sheskin D. J.: Handbook of parametric and nonparametric statistical procedures. Chapman \& Hall/CRC, (2007)
[14]. Mcdonald, J.H.: Handbook of Biological Statistics. (2009)
[15]. Arcuri, A. and Briand, L.: A practical guide for using statistical tests to assess randomized algorithms in software engineering. Proceedings of the 33rd International Conference on Software Engineering (ICSE 2011), pp. 1-10. (2011)
[16]. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in software engineering. Springer, New York (2012)
[17]. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering: an introduction. Springer, New York (1999)
[18]. Briand, L., Labiche, Y., Yan, H., DiPenta, M. : Acontrolled experiment on the impact of the object constraint language in UML-based maintenance. In: Software Maintenance, 2004, Proceedings. 20th IEEE International Conference on, pp. 380–389. IEEE (2004)
[19]. Briand, L.C., Labiche, Y., DiPenta, M., Yan-Bondoc, H.: An experimental investigation of formality in UML-based development. IEEE Trans. Softw. Eng. 31, 833–849 (2005)
[20]. Ali, S., Briand, L.C., Hemmati, H.: Modeling robustness behavior using aspect-oriented modeling to support robustness testing of industrial systems. Softw. Syst. Model. 11, 633–670 (2012)
[21]. Aichernig, B.K., Salas, P.A.P.: Test case generation by OCLmutation and constraint solving. In: Proceedings of the Fifth Interna- tional Conference on Quality Software. IEEE Computer Society (2005)
[22]. Distefano, D., Katoen, J.P., Rensink, A.: Towards model checking OCL. ECOOP-Workshop on Defining Precise Semantics for UML (2000)
[23]. Papyrus. http://www.eclipse.org/modeling/mdt/papyrus/ 33.
[24]. IBM Rational Software Architect. http://www-03.ibm.com/ soft-

ware/products/en/ratisoftarch
[25]. Correa, A., Werner, C., Barros, M.: An empirical study of the impact of OCL smells and refactorings on the understandability of OCL specifications. In: Model Driven Engineering Languages and Systems, pp. 76–90 (2007)
[26]. Störrle, H.: Improving the usability of OCL as an ad-hoc model querying language. In: 13th International Workshop on OCL, Model Constraint and Query Languages (OCL 2013), pp. 83–92 (2013)
[27]. Simula Research Laboratory: https://www.simula.no