An Experimental Investigation on Understanding the Difficulties and Challenges of Software Modellers When Using Modelling Tools

Parsa Pourali
Department of Electrical and Computer Engineering
University of Waterloo, Canada
ppourali@uwaterloo.ca

Joanne M. Atlee
David R. Cheriton School of Computer Science
University of Waterloo, Canada
jmatlee@uwaterloo.ca

ABSTRACT

Software modelling is a challenging, error-prone and tedious task. Existing Model-Driven Engineering (MDE) tools provide modellers with little aid, partly because tool providers have not investigated users' difficulties through empirical investigations such as field studies. This report presents the results of a two-phase user study to identify the most prominent difficulties that users might face when using UML modelling tools. In the first phase, we identified the preliminary modelling challenges by analysing 30 models that were previously developed by students as a course assignment. The result of the first phase helped us to design the second phase of our user study where we empirically investigated different aspects of using modelling tools: the tools' effectiveness, users' efficiency, users' satisfaction, the gap between users' expectation and experience, and users' cognitive difficulties. Our results suggest that users' greatest difficulties are in (1) remembering contextual information and (2) identifying and fixing errors and inconsistencies.

KEYWORDS

Human-Centric Software Development, Empirical Study, UML, Model-Easing, Modelling Tools.

1 INTRODUCTION

Model-Driven Engineering (MDE) addresses software complexity by raising the level of *abstraction* in software artifacts, and facilitating the *automation* of code generation and software verification [12][20]. However, modellers often find it cognitively difficult to create, edit, and debug models, and they expend a lot of effort on these tasks [9][24][25].

Researchers investigate various tools and methods to reduce the effort of editing and debugging models [8][20][23], but the tools are not adopted because they do not meet the users' needs. The reasons are, in part, that tool designers

(1) have not identified and understood the difficulties and challenges of users (e.g., through empirical observations or field studies);
(2) have not taken into account human-cognition factors that can explain users' difficulties and challenges; and
(3) have conducted few empirical evaluations of the effectiveness of their tools in supporting human users.

We performed a formative user study to learn about modellers' most-severe challenges when using modelling tools and to understand some of the most-critical obstacles to tools adoption. Specifically, we focused on identifying the cognitive challenges that modeller face when designing structural and behavioural models of software systems, as exemplified by the UML Class and State-Machine diagrams. The goal of this work is to help tool researchers and vendors to know where to focus their future tool-building efforts.

We conducted a two-phase user study. In the pre-study phase, we analyzed 30 models (i.e., Class diagrams and State-Machine diagrams) that had previously been developed as solutions to a course assignment. We reviewed the assignments and looked for modelling errors made by the students and looked for evidence of challenges that they faced when doing the assignment. The results obtained from the pre-study phase informed our design of a user study, which investigated modellers' usage of modelling tools, including the tools' effectiveness, users' efficiency, users' satisfaction, the gap between users' expectation and experience, and users' cognitive difficulties. We recruited 18 subjects and ensured that they have sufficient knowledge about the Unified Modelling Language (UML) and have experience of using at least one modelling tool. The subjects were asked to perform seven modelling tasks consisting of developing partial State-Machines of a parking lot system. For each subject, we recorded various User eXperience (UX) metrics such as the subjects' performance and verbal expressions.

The results of the study showed a substantial gap between users' expectations of tools' abilities to alleviate the challenges and the users' actual experience of using the tools. Moreover, the subjects reported a slight level of dissatisfaction with respect to different aspects of the tools. Also, the results revealed modellers' prevalent challenges when using modelling tools, among which 1) remembering contextual information and 2) identifying and fixing errors and inconsistencies are the most-critical and are most in need of consideration from tool vendors.

The rest of this report is organized as follows. In Section 2 we describe the context of our experimental research. Section 3 explains the design of our user study. Section 4 describes the execution procedure and practical considerations of it. We present the results of our study in Section 5, and discuss some important issues in Section 6. In Sections 7 and 8, we discuss the threats to validity of our user study and conclude our work, respectively.

2 EXPERIMENTAL CONTEXT

Models can be developed in various modelling languages. Some companies use their own Domain-Specific Language (DSL) whereas others prefer to use more general-purpose languages such as the Unified Modelling Language (UML) or Systems Modelling Language (SysML). The majority of models that are being developed in industrial practice are based on the UML or UML-like notations as the UML has become the de-facto standard for modelling software systems [14] and is actively taught and used by academics. Accordingly, to target a broader audience, we focus our work on the UML and on easing the editing and debugging of UML models of software systems.

The UML consists of several diagrams that can be partitioned into two types: *static* and *dynamic* [21]. While it would be generally advantageous to investigate the modelling challenges of the entire set of UML diagrams, it would be too time-consuming to cover all diagram types in a single user study, thus, we confined the scope of our study to one important static diagram and one important dynamic diagram [7], in particular Class diagrams and State-Machine diagrams. Hereafter, the term *modelling* refers specifically to Class diagram and State-Machine diagram modelling.

We investigated the following research questions:

- *RQ.1: How effective are tools in communicating with users to improve the experience of performing modelling tasks and the correctness of models?*
- *RQ.2: How efficient are modellers when using modelling tools?*
- *RQ.3: How well do modelling tools meet users' expectations?*
- *RQ.4: Overall, how satisfied are users with modelling tools?*
- *RQ.5: Which challenges are the most severe experienced by modellers employing modelling tools?*

The research questions were investigated by means of a two-phase user study. In the first phase (referred to as the *pre-study*), we conducted a lightweight analysis of a set of existing models developed as part of a course assignment, and we looked for common modelling errors made by the modellers as well as evidence of challenges that the modellers faced. Then in the second phase, we used the results of the pre-study to limit the scope of the user study to model-editing and model-debugging tasks that were most likely to be problematic. For example, we asked nothing in the user study about the structure of a State-Machine or about setting the names of the states because the results of our pre-study showed that most of the subjects could successfully manage such tasks.

2.1 Pre-Study Phase

In the pre-study phase, we examined 30 models that were submitted as solutions to a modelling assignment in the *Software Requirements: Specification and Analysis* course at the University of Waterloo. The course lectures and readings cover the necessary knowledge on the relevant UML modelling, especially Class and State-Machine diagrams. Moreover, students could seek help from the course instructor if they faced any problems understanding the course materials.

We assessed the assignments based on the marking scheme given by the course instructor, and attempted to find common errors made by the students. The marking scheme helped us in evaluating the models from two aspects:

(1) *Information Content* aimed at detecting any inconsistency between the given textual description of the system and the submitted model. We used this as a guideline to estimate how much of the modellers' difficulties actually laid in expressing the domain/system description in the modelling notation.
(2) *Model Quality* pertained to errors related to the wellformness, correctness and consistency of the models. To be more rigorous, we also assessed the models with respect to a taxonomy of error types proposed by Lange et al. [14].

2.1.1 Pre-Study Results. The results of evaluating the information content for the 30 models revealed that only four of the students submitted an incomplete model with respect to the domain description that was given to them. This suggests that most students were able to represent the problem description as a basic UML State-Machine that informally captured all of the described behaviour.

However, the results of evaluating Model Quality suggest that students had difficulty with the details and precision expected in a correct and consistent model. We grouped model quality-related errors into different categories, which are listed below:

- Category 1: Incorrect use of the structure of UML models (e.g., a State-Machine without an initial pseudo-state).
- Category 2: Referring to an undefined variable or entity. This also includes misspelling the name of an existing variable or entity, or writing incorrect paths in navigation expressions.
- Category 3: Wrong or inconsistent use of UML notation and syntax (e.g., using = instead of == in guards, transitions without expression, etc.).
- Category 4: Type mismatch between the left-hand-side (LHS) and the right-hand-side (RHS) of an assignment or a condition.

Table 1 presents the number of subjects that committed errors of each error type, and how many instances of each error type were made in the 30 models. In some cases such as S17 and S25, a large number of mistakes can be seen. This is simply because the student repeated the same mistake for multiple times. For example, using an undefined element over and over without noticing that there is no definition of such element in the Class diagram.

The results of our analysis of student assignments as well as related results in the literature identified the following preliminary list of modelling challenges referred to as *Pre-Study Challenges* (see Table 2). Subsequently, we confined the scope of the tasks in the second (main) phase of our study to focus on the suspected challenges that manifest as quality errors.

3 EXPERIMENTAL DESIGN

In order to analyze and learn about the challenges that modellers face, we conducted a *formative* empirical study rather than a *summative* one, the latter of which is usually performed to test hypotheses. We explain the design of our study in this section.

3.1 Recruitment Procedure

We emailed a recruitment letter[1] to invite interested subjects to fill out a questionnaire using SurveyMonkey. The questionnaire asked the subjects about their demographic, professional and academic backgrounds. We asked the administrative staff at the University

[1] All of the materials regarding the user study presented in this report can be found at the end of this report as appendices.

Subjects	Ctgry 1	Ctgry 2	Ctgry 3	Ctgry 4
S1		6		
S2		4	2	
S3				2
S4	3	3	2	
S5		9		
S6				
S7		7	7	4
S8		6		
S9		7		
S10				
S11		10	3	2
S12		1		
S13	2	8	2	
S14		11	5	
S15		3		7
S16		11		
S17		30		
S18		12		1
S19		11	4	7
S20				
S21	1	20	2	
S22				
S23		4		
S24		5		9
S25	1	30		
S26		4		
S27		18		
S28		15		4
S29		12		
S30				6
No. of Errors	**7**	**248**	**27**	**42**
No. of Subjects per Error Type	**4**	**24**	**8**	**9**

Table 1: Summary of the results of the preliminary study.

Table 2: Description of the Pre-Study Challenges.

ID	Description
Order	Performing a sequence of actions in the right order (omitted or out-of-order modelling actions).
Context	Remembering contextual information (e.g., consulting multiple diagrams to remember element names, association and relations in a related model).
Navigation	Writing navigation expressions (navigating correctly from one model element to a related model element).
Syntax	Remembering the appropriate keywords and syntax of the language.
Type-Matching	Matching the types of the variables that are used for LHS and RHS of an assignment (=) or a condition (==).
Debugging	Locating, understanding, and resolving errors. This includes switching back and forth among multiple diagrams to fix an inconsistency.

Table 3: Number of Subjects by Education, UML Familiarity, and Experience with Modelling Tools.

Category	Sub-Category	Count
Occupation	Graduate Student	13
	Post-Doc Researcher	2
UML Familiarity	Fairly Familiar (Novice)	1
	Familiar	5
	Very Familiar	6
	Strongly Familiar (Experienced)	3
Experience with Tools	One to six months	5
	Seven to 12 months	4
	One year to two years	1
	More than two years	5

of Waterloo to email all the students in the programs of Software Engineering, Computer Science, and Electrical and Computer Engineering. In our opinion, such students were a good fit for our study since they take modelling courses as part of their program. Moreover, the graduate research students were also suitable for our study as some of them might had experienced researching on software engineering; as subjects, they could be comparable to practitioners. Additionally, we distributed flyers around campus to reach possible non-student subjects such as post-docs, alumni, or even subjects from industry.

The Screening Procedure: To help ensure that our subjects are representative of the larger population of UML modellers, we designed a screening questionnaire to collect information about their knowledge of the UML. This questionnaire included multiple-choice questions asking subjects to rank on a Likert scale [18] their familiarity with the UML Class and State-Machine diagrams. In addition, it included 10 UML-specific questions selected from online sample practice tests such as the Sun Certified Java Associate exams.

After receiving a subject's response to the recruitment questionnaire, we reviewed his/her answers to both demographic and UML-related questions to determine his/her eligibility for being a subject of the study. We then contacted eligible subjects to schedule a study session. Ineligible subjects were also contacted and were informed about their ineligibility.

Population: We targeted a total number of 20 subjects (15 were done by the date of this report), based on the advice offered in the three references [13],[26] and [3]. We recruited students who had the required knowledge and experience based on their answers to the recruitment questionnaire. An overview of the subjects' education and experience can be seen in Table 3. As can be seen, all of our subjects were familiar and experienced enough with the UML and its tools.

3.2 The Application Domain

To minimize the effects of domain knowledge on the subjects' performance on tasks, we designed the study around a fairly simple

Table 4: Number of subjects per tool.

StarUML	ArgoUML	Visual Paradigm	MagicDraw	UmLet	Umple	Papyrus	Astah
3	3	3	3	1	2	2	1

application domain: a gated parking-lot system. Moreover, to familiarize the subjects with the application domain, we asked the subjects to study a textual description of the domain as well as the domain model of the system before starting the study's tasks.

3.3 Treatment Allocation

To guard against the threat to validity that poor performance could be due to unfamiliarity with a specific modelling tool, we allowed the subjects to use the modelling tool of their choice: MagicDraw, ArgoUML, Astah, Papyrus, Visual Paradigm, UMLet, and Umple. Table 4 shows a summary of the distribution of the tools amongst subjects. It is notable that most of the tools that were chosen by our subjects are among the list of most-heavily used tools reported in a recent survey by Anger and Lethbridge [1].

3.4 Tasks

Each subject was given a textual description and a partial Class diagram of a parking lot system, and was asked to edit and debug variants of a State-Machine diagram. The names used in all the diagrams (e.g., class attributes' names) were chosen to ease comprehension of the model. Also, the researcher was present during the study to answer the subject's questions about the domain description, or clarify the tasks, if needed.

The experiment comprised seven tasks. The first four tasks were designed to gauge the effort of editing models (e.g., developing State-Machines and editing transition expressions), whereas the last three tasks were designed to understand the challenges of users when debugging models (i.e., finding and fixing errors and inconsistencies in the models). We designed simple tasks mainly for two reasons: 1) to increase the size of the pool of potential eligible subjects, and 2) to ascertain whether challenges exist even for such simple tasks, let alone for complicated tasks.

Model-Editing Tasks: In each of the first four tasks (i.e., *Task1, Task2, Task3, and Task4*), the subjects were given a structured textual description of a transition and were asked to use the modelling tool to set the *triggering event*, *guard*, and *action* of the transition. Below is an example of a model-editing task.

Task1: Please develop the transition that is labelled as T1 in the diagram (based on the following description).

- *Triggering Event: No triggering event is required for this transition.*
- *Guard: If the gate id is B.*
- *Action: The Gate will go to the closed state; that is, the gate position should be set to down.*

Model-Debugging Tasks: For each of the three debugging tasks, we introduced a few inconsistencies in the diagrams and asked the subjects to locate and fix them. They could either examine the diagrams manually or use the tool's diagnostic features. Specifically, *Task5* asks the subjects to rename elements in the Class diagram, and then locate any inconsistencies in the model that were introduced by that action. In *Task6* and *Task7*, the subjects were asked to locate inconsistency errors that were embedded in the model, such as identifying model elements that were used but not defined, and detecting incorrect navigation expressions. Following is a sample model-debugging task (i.e., *Task5*).

Task5: Assume that you are supposed to change the name of the gates from A and D to GA and GD respectively in the Class diagram. Please implement the change in the model and report any inconsistencies you found that are caused by this change.

3.5 Data-Collection Techniques and Design

We evaluated the subjects' performance along three different dimensions, as prescribed by Tullis and Albert [3], namely performance, self-reported, and behavioural metrics. This section precisely defines the variables and metrics measured in the study.

3.5.1 *Performance Metrics.*

We measured three performance metrics:

- **Task Completeness:** We measured the degree to which a subject was effective in completing a task. We defined three levels of success:
  1. *Complete(1.0)*: A subject completed a task without any assistance. Note that, in model-editing tasks, a score of Complete does not mean that the task was error free. A model-editing task is deemed Complete if no errors of omission were performed. Errors of commission are possible. A model-debugging task is deemed Complete only if all errors are found and fixed.
  2. *Partially Complete (0.5)*: A subject asks for help during the task, such as asking about the language syntax, or asking for clarification about a model element.
  3. *Incomplete (0.0)*: A subject was unable to complete a task. For model-editing tasks, a score of Incomplete means that a subject omitted some aspects of a task's requirement, whereas in model-debugging tasks it means a subject could not locate all of the embedded errors in the model.
  4. *Generally Complete*: Refers to the summation of the above three scores for a task (i.e., +1.0 for each subject with a Complete score and +0.5 for each subject with a Partially Complete score).
- **Errors:** We counted the number of errors committed by each subject per task and classified errors as being either *wellformness* and *consistency* errors. [14]. We did not count *completeness* errors as our tasks were not designed to analyze these types of errors. It is important to note that the number of errors is different from the degree of task completeness as a task can be completely done but not be error-free.
- **Efficiency:** We measured efficiency using two metrics:
  1. *Time-on-Task* refers to the time that it takes a user to perform a particular task using a product, and is comparable to Hill's definition of modelling effort [10].

(2) *Lostness* measures how *"lost"* a subject is when performing a task. To assess lostness, the following three factors are measured: 1) R: the minimum number of diagrams or dialogues that must be visited to accomplish the task, 2) S: the total number of diagrams or dialogue-boxes visited while performing the task (counting revisits), and 3) N: the number of different (unique) diagrams or dialogue-boxes the subject visited while performing the task. Lostness, L, is then calculated using the following formula [3][27].

$$L = \sqrt{(\frac{N}{S} - 1)^2 + (\frac{R}{N} - 1)^2} \quad (1)$$

Lostness scores range from *Zero* to *One*. The higher the score, the more trouble the user had finding what they want. Smith [27] found that users with a lostness score of less than 0.4 have no substantial difficulty to fulfil a task, whereas users with a lostness score of greater than 0.5 are definitely lost. One can also estimate subjective lostness by comparing to the optimal score (e.g., to the smallest value of lostness among all the subjects) [3].

3.5.2 Self-Reported Metrics. Perhaps the most traditional means of assessing the usability of a tool is asking users to tell us about their expectation and experience with the tool [2][3]. We designed our experiment to collect self-reported data from users to gauge their satisfaction.

Expectation versus Experience ratings: We asked the subjects to rate two things:

(1) their *Expectation Rating* of how easy or difficult they expected a task to be (based on their understanding of the task and the tool) before starting the task, and
(2) their *Experience Rating* of how easy or difficult each task actually was.

We used the same 7-point rating scale (1=Very easy to 7=Very difficult) for both ratings. For each task, we then calculated an average *Expectation Rating* and an average *Experience Rating* of all the subjects. The difference in the two average ratings indicates a degree to which the tools are effective in satisfying the users' expectations and needs: a difference score of *zero* indicates high effectiveness in satisfying the users' expectations, and a difference score of 6 suggests an imbalance between the users' expectations and what the tools provide.

The expectation and experience ratings were collected in two different stages of the study: 1) before and after performing a task, and 2) at the beginning and at the end of the study.

- *Pre-Session Expectation Rating:* Using the Pre-Session Expectation Rating, we collected data about the subjects' overall expectation of the tools' proficiency with respect to the pre-study challenges before starting the session (Fig. 2).
- *Post-Session Experience Rating:* Once the session was completed, we gave the same set of questions from the Pre-Session Expectation Rating to the subjects, asking their opinions about how well the tool provided features that aided them and how well the tool met their expectations to overcome the pre-study challenges. We can use the averages of the Pre- and Post-Session ratings to get insight into opportunities for improving the tools.
- *Pre-Task Expectation Rating:* Before starting each task, we asked subjects to rate how easy or difficult they thought each task should be based on their expectations of the tool and understanding of the task. The rating was a 7-point Likert-scale from 1=Very easy to 7=Very difficult.
- *Post-Task Experience Rating:* Once each task was finished, we asked users to rate how easy or difficult the task was based on their actual experience of using the tool.

Usability Questionnaire: In addition to the expectation and experience ratings, we gave subjects a usability questionnaire to collect information on the subjects' satisfaction with the usability of their respective tools. Different usability questionnaries could be employed for this assessment, such as System Usability Scale (SUS) [5], Computer System Usability Questionnaire (CSUQ) [17], Questionnaire for User Interface Satisfaction (QUIS) [6], and, Usefulness, Satisfaction, and Ease of Use Questionnaire (USE) [19]. Among these, we decided to use CSUQ [16][17] because the questions listed in the CSUQ were more in line with the goals of our study. It needed almost no adaptation, whereas the other questionnaire types would have needed much more adaptation to their questions. The adaptation that we made in the CSUQ questions were to replace the term *"system"* with the term *"tool"* in the questions. Our CSUQ consisted of 19 statements to which the user rated agreement on a 7-point scale of "Strongly Disagree" to "Strongly Agree", plus N/A.

3.5.3 Behavioural Metrics. The think-aloud protocol [4] allowed us to collect information about usability issues from the subjects' verbal statements such as:

- Verbal expressions of confusion, frustration, dissatisfaction, pleasure, or surprise.
- Verbal expressions of confidence or indecision about a particular action that might be right or wrong.
- Not saying or doing something that they should have done or said.

To get the most out of the subjects, we prompted the subject if he/she did not express his/her thoughts loudly. The prompt messages were based on the situation but some examples are: *What are you thinking now? What are you trying to do? Why did you do that?* To help the accuracy of our data (i.e. time on tasks), we tried to avoid too much prompting and tried to have minimal discussion when the subject was performing a task. After each session, we analysed the audio recording of the subject's session to detect the most valuable verbal expressions, expanding the list of the verbal expressions of the subjects as we processed more subjects. We coded and classified the verbal expressions into different categories. By counting the number of times that the subjects made verbal statements within each category, we could obtain useful results about the prominent challenges that the subjects faced.

4 EXECUTION AND PRACTICAL CONSIDERATIONS

The study was conducted in 18 separate sessions, one for each subject. The subjects performed their tasks using a PC machine in the researcher's office. The duration of each session ranged from one hour to nearly two hours with an average of about 80 minutes. Also, to automate the process of data collection, we developed a

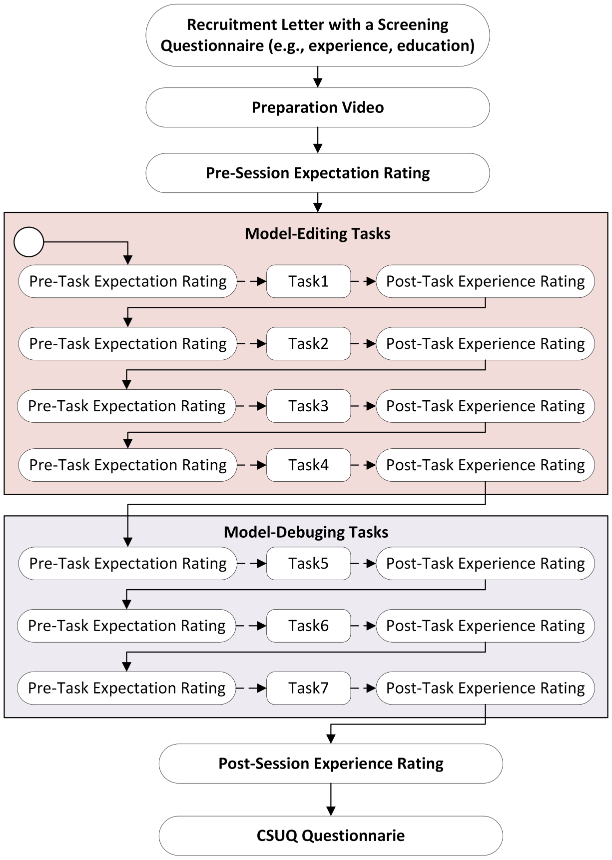
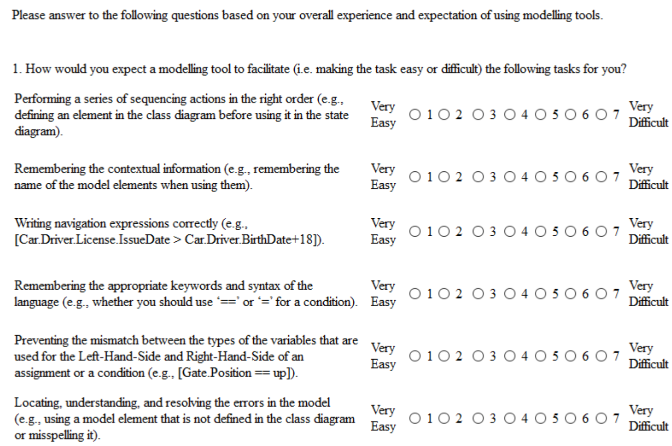


Figure 1: The workflow of a subject's session.

tool that automatically records the time on each task (as a measure of effort). The time on each task starts from the time that the subject begins the task (including reading the description of the task) and ends when the subject acknowledges that he/she is done with the task (i.e., presses the *done* button in the tool). Also, the tool stores other data such as responses to all questionnaires.

The study consisted of several segments: Preparing the Subject, Collecting the Pre-Session and Pre-Task Expectation Ratings, Performing the Tasks, Collecting the Post-Session and Post-Task Experience Ratings, and conducting the CSUQ. Fig. 1 illustrates the structure of a session.

Preparing the Subject: After greetings and signing the consent form, each subject was given an introduction to the study by viewing a preparation video. The subject was asked to watch and listen to the video carefully. The video included general information about the procedure and methods of the study (e.g., think-out-loud method). The advantage of using one video for all the subjects was that they all received the same information with respect to preparing for the study. Moreover, in the video we emphasized that: *in the course of the experiment, the subject will not be evaluated in any way.* We made it clear that it is the tool that is under scrutiny and not them. We needed the subject to understand this because it was important to create a relaxing and informal atmosphere to make the session as effective as possible.


Please answer to the following questions based on your overall experience and expectation of using modelling tools.

1. How would you expect a modelling tool to facilitate (i.e. making the task easy or difficult) the following tasks for you?

Performing a series of sequencing actions in the right order (e.g., defining an element in the class diagram before using it in the state diagram).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Remembering the contextual information (e.g., remembering the name of the model elements when using them).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Writing navigation expressions correctly (e.g., [Car.Driver.License.IssueDate > Car.Driver.BirthDate+18]).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Remembering the appropriate keywords and syntax of the language (e.g., whether you should use '==' or '=' for a condition).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Preventing the mismatch between the types of the variables that are used for the Left-Hand-Side and Right-Hand-Side of an assignment or a condition (e.g., [Gate.Position == up]).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Locating, understanding, and resolving the errors in the model (e.g., using a model element that is not defined in the class diagram or misspelling it).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult

Figure 2: Pre-session expectation rating questionnaire.

To familiarize the subject with the application domain, we asked the subject to study a textual description of the domain, the provided Class diagram and its model elements (e.g., classes, operations, attributes) along with a description of all the elements. Finally, we asked the subject to rank their expectations of the tool's ability to ease the *pre-study challenges* that were found in the pre-study phase (see Fig. 2).

Performing the Tasks: We then asked the subject to perform tasks using the modelling tool of his/her choice. Before each task, we asked the subject to read the description of the task and rank how easy or difficult they thought it would be when using the tool. When performing the tasks, subject was asked to express their thoughts out-loud, so that we could understand the subject's cognitive difficulties. For the purpose of later analysis, we screen-captured the subject's work with the tool as well as audio-recorded their voice. Finally, once the subject finished a task, he/she was asked to provide a Post-Task Experience Rating for the task.

There is a trade-off between spending time on a task to keep the quality of the solution high and making progress on all the study's tasks. Imposing any time pressure on the subjects could reduce the quality of the their solutions. Thus, we allowed the subjects to work at their own pace and to announce when they completed a task. However, this could result in the times on tasks becoming unrealistically long and useless for further statistical analysis. To overcome this challenge, we pursued the following strategies:

- If a subject insisted on solving the task but did not show any signs of progress, then the researcher stepped in and provided some hints to the subject. In this case, the task success was deemed at best Partially Complete (0.5). If after given hints the subject still could not fulfil the task, then we asked him/her to move on to the next task and the task was deemed Incomplete (0).
- We did not offer an hourly rate. Our offer was a fixed honorarium of $20 in return for an estimated 90 minutes (maximum) of work for the study. Hence, the subject knew that he/she would receive the same amount of compensation even if they completed their tasks earlier than expected. This avoided

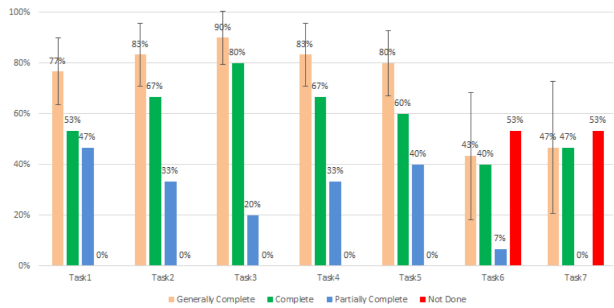


Figure 3: Task completion rate (error bars represent 95% confidence interval).

Figure 4: The average error rate made by the subjects per task.

the threat of subjects playing around with the tool to receive a higher payment.

- To further motivate subjects to finish their tasks in good time, he/she was allowed to immediately leave the experiment once he/she completed the tasks.

Post-Session Activities: At the end of the session, the subject was asked to complete two questionnaires: a Post-Session Experience Ratings and CSUQ[2] [16][17]. The Post-Session Experience Ratings comprised the same set of the questions that were shown in Fig. 2, but this time asked the questions from the perspective of having experience using the tool. The CSUQ collected useful data about the subject's level of satisfaction of using the tools.

In addition, at the end of each session, we asked the subject to provide any additional comments on the tool and their experience of using the tools in an open-ended textual format. This could be positive or negative feedback.

5 RESULTS

This section presents the results of the study with respect to each of the research questions.

5.1 Tools' Effectiveness (RQ.1)

We assessed the effectiveness of the tools in assisting subjects with model-editing and model-debugging tasks by measuring the subjects' success rate and number of errors in their task solutions.

Fig. 3 illustrates the subjects' success rates on the tasks, and shows that a significant number of the subjects were not successful in finishing their tasks unless we provided them with some hints. More importantly, it shows that fewer than 45 percent of the subjects were fully successful in the tasks 6 and 7, which were related to debugging of models. This shows that the tools do not provide enough support to help subjects resolve errors in the models.

We divided the errors made by the subjects during all the tasks into two classes based on the guidelines given by Lange et al. [14]:

- *Consistency errors* are errors that can be temporarily tolerated but that should be fixed before delivering the model. Consistency errors include: an element is used but not defined, misspelled element names, incorrect navigation paths (e.g., g.blockage instead of g.Sensor.blockage), and type-mismatches between the LHS and RHS of an expression.
- *Wellformness rules* are UML conventions that can help maximize the model's understandability. Wellformness errors occurred mostly when a subject used UML syntax incorrectly, or produced an ill-formed expression (e.g., putting extra parentheses or quotations). It is important to note that wellformness errors can be found by performing analysis within a single diagram.

Fig. 4 depicts the subjects' error rates for consistency and wellformness errors per task. For example, in *Task1*, each subject made, in total, 1.33 mistakes. Interestingly, none of the subjects were able to finish all of their tasks without any error. Moreover, more than half of the subjects failed to spot the errors in tasks 6 and 7, where the subjects were asked to debug the models.

This is notable given that the tasks were relatively easy and the model was very small compared to complex industrial models [15]. Based on our observations, the main reason for such high error rates was that the subjects relied too much on the tools, and assumed that the tool would notify them of errors. However, most of the errors were not automatically detected and reported by the tools' consistency checking until the subjects asked for it. It is possible that subjects thought their solutions were correct and did not invoke the tools' consistency checking.

5.2 Efficiency (RQ.2)

Efficiency was investigated by means of two metrics: time on tasks, and lostness. Fig. 5 depicts the results of the time that the subjects took for the tasks. To ensure that our results are meaningful, the times for the Incomplete tasks are not included in our analysis. This is because an unsuccessful subject could take a very long time to give up or to be asked to move on the next task, thus it can dramatically raise the average time on the tasks.

Tullis and Albert [3] suggest that one way to assess time on task is to compare the average time it took all subjects to perform a particular task with the minimum time it took to perform that task. Fig. 6 shows that the subjects' average times on tasks were higher than the best achievable times.

Worse, even the most-efficient subject was not as efficient as he/she could have been as evidenced by the lostness scores. Fig. 7 shows the average lostness scores for subjects' Generally Complete

[2] A sample CSUQ can be found at http://garyperlman.com/quest/quest.cgi

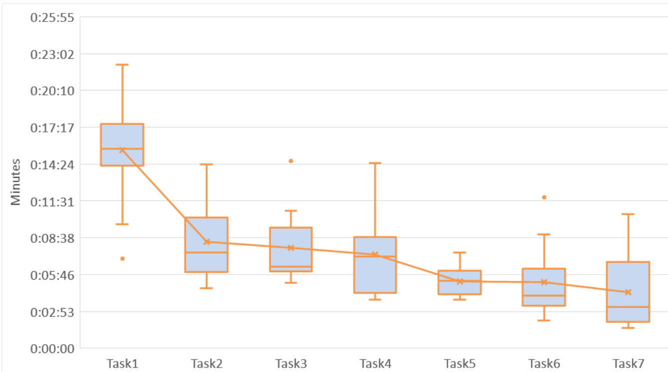


Figure 5: The average time to perform each task.

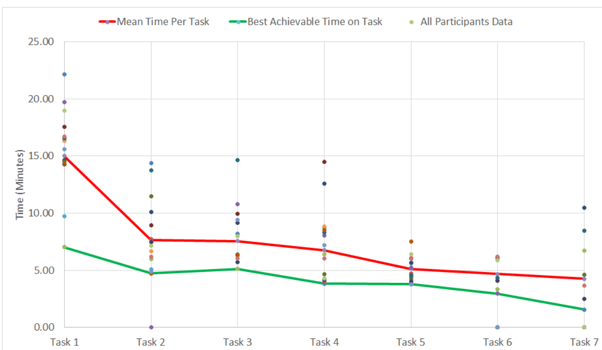


Figure 6: Mean time per task vs. best achieved time per task.

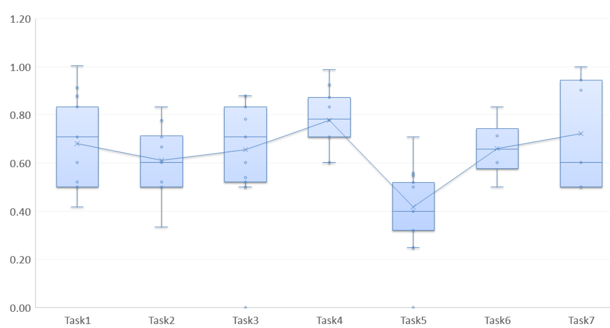


Figure 7: The average lostness score each task.

tasks. The Incomplete tasks are not included in the lostness scores because subjects were not "lost" when they left tasks Incomplete- They presumed they had met the task's requirements.

Using Smith's [27] threshold for lostness scores of 0.4, we see that, on average, the only task that the subjects could perform with a fairly acceptable lostness score was Task 5. In all other tasks, the subjects exceeded the acceptable lostness score which suggests the tools are inefficient.

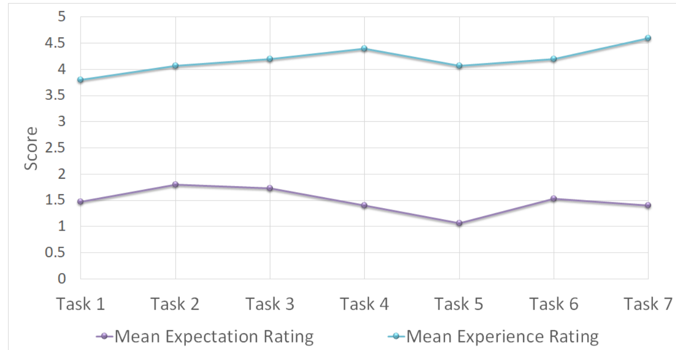


Figure 8: Pre/post-task ratings for each pre-study challenge.

5.3 Satisfiability of Users' Expectations (RQ.3)

We answer RQ.3 by comparing Pre-Task Expectations against Post-Task Experiences. The result indicates that the subjects expected the tasks to be easy (based on their understanding of the task and the tools). However, their Post-Task Experience Ratings show that the tools did not meet their expectations. Fig. 8 shows the gap between the subjects' expectations and experiences in how difficult it was to use the tools to perform the tasks. The subjects, on average, expected the tools to ease modelling challenges (i.e., the mean Likert score was well below the neutral level of 4), but the subjects' experience ratings leaned towards dissatisfaction (i.e., the mean Likert score was slightly above the neutral value). This suggests that tools are not meeting the users' expectations on alleviating modelling tasks. Note that although the subjects could presumably learn from previously performed tasks, the subjects post-experience scores suggest that the tasks became increasingly harder for them.

5.4 Users' Satisfaction (RQ.4)

We used CSUQ to measure the satisfiability and usability of the tools under test where subjects answered questions about: 1) Overall Satisfiability (OVERALL), 2) System Usefulness (SYSUSE), 3) Information Quality (INFOQUAL), and 4) Interface Quality (INTERQUAL). Subjects specified their level of agreement based on the Likert scale that ranged from 1 (Strongly Disagree) to 7 (Strongly Agree) with the neutral value of 4. Fig. 9 shows that subjects were slightly dissatisfied with the tools' usability (SYSUSE) and the interface quality (INTERQUAL) (mean value of ~3), and were more strongly dissatisfied with the tools' ability to provide the relevant or the contextual information during the tasks (mean value of around 2.5 for the information quality (INFOQUAL)).

5.5 The Most-Severe Challenges (RQ.5)

We investigated the users' most-severe challenges by means of three different techniques: 1) Pre- and Post-Session Ratings, 2) Behavioural Metrics, and 3) Analysing Errors. The three analyses produced the same results.

5.5.1 Pre- and Post-Session Ratings. We used Pre- and Post-Session Ratings to identify pre-study challenges that users expected the tools to alleviate, but found that the tools did not. Fig. 10 depicts the mean of the differences between Pre-Session Expectation

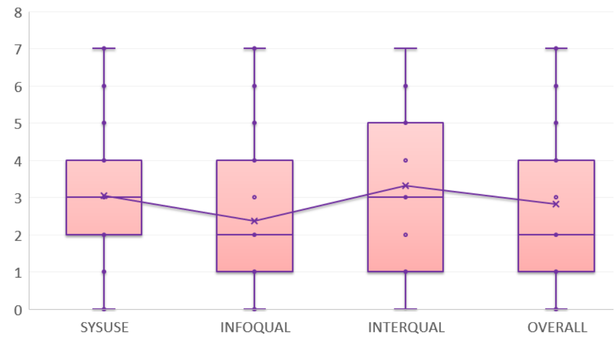


Figure 9: Box-plot based on the result from CSUQ.

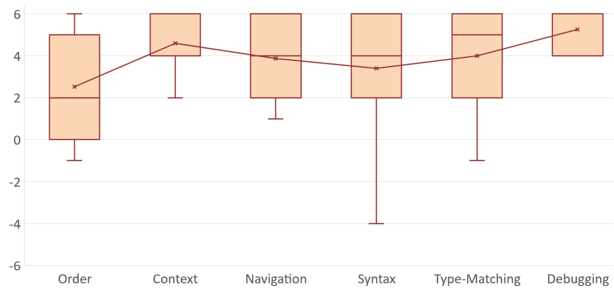


Figure 10: Mean discrepancy between the pre- and post-session ratings for each pre-study challenge.

and Post-Session Experience Ratings with respect to the pre-study challenges. The figure shows that the subjects' expectations and experiences had the greatest disparity with respect to the *Context* and *Debugging* challenges. That is, the subjects expected to face the least difficulty regarding the two challenges, but instead experienced the most difficulty. Moreover, the fact that the box plots for these two challenges are short indicates a high level of agreement among the subjects' views. These results suggest that tool providers should propose tool advances that address these two prominent challenges over other tool advances.

5.5.2 Behavioural Metrics. The think-aloud protocol helped us identify cognitive difficulties and challenges that the subjects faced. We started with a list of the categories of challenges based on the six pre-study challenges. This list was then extended to 10 categories as we learned more from the verbal analysis (see Table 5).

By analysing the subjects' verbal expressions, we were able to correlate each expression to its pertinent category of challenge(s). Sometimes, a statement could fall in two or more categories. In such a case, we correlated the statement to all of the applicable categories. The result of our verbal analysis is shown in Fig. 11. The figure indicates the two major challenges for the users were *Context* and *Debugging*. Some of the statements that the subjects made for *Context* were: "[While trying to remember the name of the elements] The tool should give me some recommendation." ,"what was the name of the class!", and "Oh! I forgot the name again...".

5.5.3 Analysing Errors. Analysis of errors is another method of understanding the modellers' challenges. The errors rates presented

Table 5: Categories of verbal expressions.

Code	Description
Order	Performing a sequence of actions in the right order.
Context	Remembering contextual information.
Navigation	Writing navigation expressions.
Syntax	Remembering keywords and the syntax of the language.
Type-Matching	Matching the types of the LHS expression and RHS in an assignment (=) or a condition (==).
Debugging	Locating, understanding, and resolving errors.
Layout	Issues related to the layout of the diagrams.
Views	Viewing mechanisms to combine different diagrams (e.g., tile and cascade organization of multiple views).
Reuse	Issues related to reusing model elements (copy/paste of elements among different diagrams in textual editors).
Look	Appearance dissatisfaction (e.g., shapes, lines, colors).

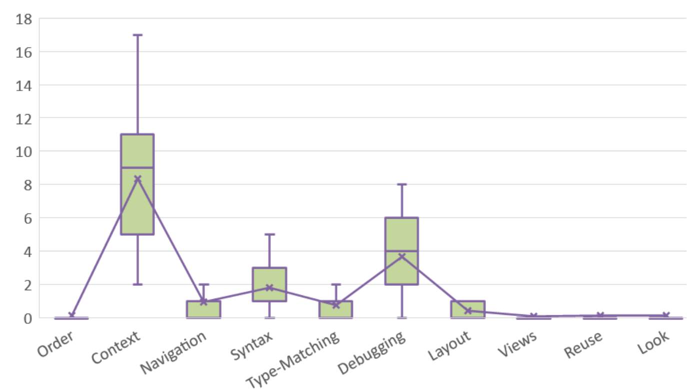


Figure 11: Mean frequency of each challenge expressed verbally by each subject.

in Fig. 4 show that almost all the subjects introduced inconsistencies to the model during their first four tasks, i.e. model-editing tasks. Further analysis showed that the majority of these inconsistencies were related to referring to an incorrect or an undefined element. We believe that this relates to the *Context* challenge and that the subjects had difficulties recalling the intended model elements in the Class diagram. The tasks 5, 6, and 7 were designed to gauge the severity of the *Debugging* challenges, and the results presented in Fig. 3 show a high level of failure in the subjects' average performance for these tasks.

Based on the above results, one can conclude that the *Context* and *Debugging* challenges are the most-severe challenges. Due to the space limitations, we did not present the data collected from other sources that could confirm the above results such as the frequency with which subjects switched to the Class diagram during tasks as a reference metric for the severity of the *Context* challenge.

6 DISCUSSION

Below are additional observations about the subjects' behaviours, and attitudes about the tools, collected during the subjects' sessions with the tools or through an open-ended textual feedback that was answered at the end of the study.

- Sometimes, the subjects performed their tasks and were satisfied with how they fulfilled the tasks and did not realize that they had created an inconsistent or erroneous model. Therefore, it is important that the tools have features that either prevent such inconsistencies and errors in the model or report them during commission. Our subjects expected the tool to provide them with an immediate warning about errors, but most current tools do not provide such feedback.
- Some tools such as MagicDraw [11] or Visual Paradigm [22] allow users to cross-link the operations in the event/effects of a transition to related operations in the Class diagram. However, users avoid creating these cross-links and instead just set the name or label of the transition. It is unknown whether the subjects simply do not care or if they think that it will be complicated for them to cross-link the operations to the related attributes for every transition. Perhaps, the ideal would be if the tools allowed the user to type a transition as a text, and the tool could automatically cross-link.
- Some tools (e.g., UMLet) are very lightweight, very simple, and easy to learn, and are useful for creating simple models but are not suitable for editing complex models because of the lack of features such as syntax and type checking, auto-completion, etc. In contrast, tools that have these features sometimes feel very heavyweight and complicated. As a result, the latter tools have a learning curve that intimidates users because of the many views and features that it offers without proper organization.

7 THREATS TO VALIDITY

In this section, we discuss the most important threats to the validity of the study and provide suggestions for improvements in future studies of this type.

7.1 Construct Validity

As mentioned, we used instrumentation to collect data about the time spent on tasks. The instrumentation improves the level of accuracy of the collected timings; however, because we used the think-aloud protocol, it is probable that the thinking out load and our prompts to the user increased the actual time to solve tasks. Also, our measurements for the time includes the time that the subjects took to read the task description and answer the Pre- and Post-Task Ratings, although this is negligible.

7.2 Internal Validity

One threat to internal validity was the subjects' familiarity with the UML. We tried to mitigate the risks by recruiting subjects who could passed our UML exercises. We also made sure that the subjects had previously passed at least one course that included UML modelling.

A related issue was subjects' level of experience and proficiency with the tools. We asked about the subjects' experiences with the tools and allowed them to choose whichever tool they were comfortable with. However, we were mostly collecting information about their personal perceptions of their experiences. So, it is possible that each subject had a different perception about his/her experience, which may not have been necessarily true. In our case, we did not find that any of our subjects showed signs of such concern.

7.3 External Validity

The main threat to external validity is that the subjects were students rather than experienced modellers who work in industry. To mitigate this threat we kept the scope of the study and the size of the model quite small compared to industrial models of software systems. Moreover, the tasks were relatively small and easy to do in terms of size, complexity and duration. Nevertheless, we cannot rule out the possibility that the observed effects could have been different if the systems and tasks had been larger.

One other threat was the subjects' familiarity with the system being modelled. For example, it is possible that the results may differ from situations where the users have dealt with the model for a long period of time. Therefore, a related threat is whether the short-term results observed in the study are representative of long-term development of models in the real world.

8 CONCLUSION

In this report we presented an empirical (formative) user study to understand the difficulties and challenges of UML modellers when they use modelling tools to edit and debug Class and State-Machine diagrams. We collected information that can be exploited to enhance existing modelling tools and alleviate the challenges of software modelling. We summarize our results with respect to the stated research questions as follows.

- *RQ.1: How effective are tools in communicating with users to improve the experience of performing modelling tasks and the correctness of models?*
  The results showed that almost none of the subjects could finish their tasks without any error. Moreover, the subjects' poor performance on the tasks 5, 6, and 7 indicated that the tools did not effectively address the users' difficulties regarding the model-debugging tasks.
- *RQ.2: How efficient are modellers when using modelling tools?*
  With respect to efficiency, the overall results showed the inefficiency of the users in performing their tasks. The large difference between the best time on the tasks (i.e., 7 minutes) and the worse time on the tasks (i.e., 22 minutes) shows an inefficiency of the subjects with respect to the time on the tasks. Also, according to the the lostness results, the subjects had substantial difficulties ($lostness >= 0.5$) in finding what they needed to fulfil their tasks.
- *RQ.3: How well do modelling tools meet users' expectations?*
  The average discrepancy between the Pre-Task Expectation and Post-Task Experience Ratings showed that there is a notable gap between the users' expectations and the tools' capability to satisfy the expectations.
- *RQ.4: Overall, how satisfied are users with modelling tools?*
  The result of the CSUQ indicated that the subjects' opinion on the tools' usability leaned towards dissatisfaction.
- *RQ.5: Which challenges are the most severe experienced by modellers employing modelling tools?*

> The results of our analysis on the subjects' verbal expressions, Pre- and Post-Session Ratings, and errors in the tasks determined the most-prominent challenges of modellers are: 1) remembering contextual information (*Context*) and 2) locating, understanding and resolving errors in models (*Debugging*).

Next steps are to identify enhancements to tools that address the most-critical challenges. For each challenge, we will identify relevant human-cognition factors that might effectively reduce the challenge, and devise enhancements to the tools that reinforce the identified factors. We will also hold empirical user studies to assess the impact of the tool advances on modellers' effectiveness.

9 ACKNOWLEDGEMENT

We thank the 18 participants of our empirical user study. This work was funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the University of Waterloo.

REFERENCES

[1] Luciane TW Agner and Timothy C. Lethbridge. 2017. A survey of tool use in modeling education. In *20th International Conference on Model Driven Engineering Languages and Systems (MODELS)*. IEEE, 303–311.

[2] William Albert and E. Dixon. 2003. Is this what you expected? The use of expectation measures in usability testing. In *Proceedings of the Usability Professionals Association 2003 Conference, Scottsdale, AZ*.

[3] William Albert and Thomas Tullis. 2013. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes.

[4] Kathy Baxter, Catherine Courage, and Kelly Caine. 2015. *Understanding your users: a practical guide to user research methods*. Morgan Kaufmann.

[5] John Brooke. 1996. SUS: A quick and dirty usability scale. *Usability evaluation in industry* 189 (1996).

[6] J. P. Chin, V. A. Diehl, and L. K. Norman. 1988. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the Special Interest Group on Computer-Human Interaction (SIGCHI) conference on Human factors in computing systems - CHI '88*. ACM Press, New York, New York, USA, 213–218.

[7] John Erickson and Keng Siau. 2007. Can UML be simplified? Practitioner use of UML in separate domains. In *proceedings of Evaluation and Modelling Methods for Systems Analysis and Development (EMMSAD)*. 89–98.

[8] Constance L. Heitmeyer, Ralph D. Jeffords, and Bruce G. Labaw. 1996. Automated consistency checking of requirements specifications. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 5, 3 (1996), 231–261.

[9] Anders Hessellund, Krzysztof Czarnecki, and Andrzej Wąsowski. 2007. Guided development with multiple domain-specific languages. In *International Conference on Model Driven Engineering Languages and Systems (MODELS)*. 46–60.

[10] James H Hill. 2011. Measuring and reducing modeling effort in domain-specific modeling languages with examples. In *18th International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*. IEEE, 120–129.

[11] No Magic Inc. 2013. Magicdraw, UML. (2013).

[12] Stuart Kent. 2002. Model Driven Engineering. In *International Conference on Integrated Formal Methods*. Springer, 286–298.

[13] Andrew J. Ko, Thomas D. Latoza, and Margaret M. Burnett. 2015. A practical guide to controlled experiments of software engineering tools with human participants. *Empirical Software Engineering* 20, 1 (2015), 110–141.

[14] Christian F. J. Lange and Michel R. V. Chaudron. 2004. An empirical assessment of completeness in UML designs. In *Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering (EASE '04)*. IET, 111–121.

[15] Christian F. J. Lange, Michel R. V. Chaudron, and Johan Muskens. 2006. In practice: UML software architecture and design description. *IEEE software* 23, 2 (2006), 40–46.

[16] James R. Lewis. 1992. Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. 36, 16 (1992), 1259–1260.

[17] James R. Lewis. 1995. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction* 7, 1 (1995), 57–78.

[18] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).

[19] Arnold M. Lund. 2001. Measuring Usability with the USE Questionnaire 12 General Background. *Usability interface* 8, 2 (2001), 3–6.

[20] Gunter Mussbacher, Daniel Amyot, Ruth Breu, Jean-Michel Bruel, Betty HC Cheng, Philippe Collet, Benoit Combemale, Robert B. France, Rogardt Heldal, James Hill, et al. 2014. The relevance of model-driven engineering thirty years from now. (2014), 183–200.

[21] Object Management Group. 2015. OMG Unified Modeling Language TM (OMG UML), Superstructure v.2.3. *InformatikSpektrum* 21, March (2015), 758.

[22] Visual Paradigm. 2013. Visual paradigm for uml. *Visual Paradigm for UML-UML tool for software application development* (2013), 72.

[23] Tanumoy Pati, Dennis C. Feiock, and James H. Hill. 2012. Proactive modeling: auto-generating models from their semantics and constraints. In *Proceedings of the 2012 workshop on Domain-specific modeling*. ACM, 7–12.

[24] Tanumoy Pati, Sowmya Kolli, and James H. Hill. 2017. Proactive modeling: a new model intelligence technique. *Software & Systems Modeling* 16, 2 (2017), 499–521.

[25] Jason E. Robbins and David F. Redmiles. 2000. Cognitive support, UML adherence, and XMI interchange in Argo/UML. *Information and Software Technology* 42, 2 (2000), 79–89.

[26] Forrest Shull, Janice Singer, and Dag IK Sjøberg. 2008. *Guide to advanced empirical software engineering*. Vol. 93. Springer.

[27] Pauline A. Smith. 1996. Towards a practical measure of hypertext usability. *Interacting with computers* 8, 4 (1996), 365–381.

APPENDIX A: PRE-STUDY MATERIALS (ASSIGNMENT DESCRIPTION AND MARKING SCHEME)

CS 445 / ECE 451 / CS 645 / SE 463
Software Requirements Specification and Analysis
Assignment 3: State-Machine Modelling, Temporal Logic
Due Date: Tuesday, March 11 at 12:00pm
(Late Date: Thursday, March 13 at 12:00pm)

Submission

Combine your models and answers in a single PDF file and email it to cs445@student.cs.uwaterloo.ca.

1) State Machine Model

You are to create a UML State-Machine model of the proposed system to enforce the parking policies at the Bauer complex. Bauer provides parking for two types of users: customers and employees. There are two levels of parking: surface-level parking and underground parking. Surface-level parking is for customers only. They pay a $2 flat fee for parking. Employees park for free, but only if they park underground. Underground parking is also open to customers in the evenings and on weekends, but of course customers must still pay ($2) to use it.

Gates control access to both levels of parking. Each gate controls traffic flowing in a single direction (e.g., traffic entering the complex), and it opens only for cars that approach the gate from the appropriate direction. Employees' cars have transponders that the gates can sense. Some gates open only for cars that have transponders. In addition, the system uses transponder data (IDs) to keep track of where employees' cars park. There are four gates that are controlled independently:

- Gate **A** controls entry into the Bauer complex. It opens whenever it senses a car approaching from outside the Bauer complex. Cars that pass through gate A enter the surface-level parking lot.
- Gate **B** controls access from surface-level parking to underground parking. Normally, it opens only if an approaching car has a transponder. But during evenings (18:00-6:00) and weekends (Friday 18:00 - Monday 6:00), it remains open to all cars.
- Gate **C** opens whenever it senses a car approaching from the underground parking lot.
- Gate **D** controls exits from the Bauer complex and collects parking fees. The gate opens whenever a $2 fee is paid (no change is provided). In addition, if an approaching car has a transponder, the gate opens if the car had parked underground (i.e., if the transponder triggered gate B since the car entered the complex).

Each of the gates closes 5 seconds after it opens — unless a blockage is detected, in which case it remains open and tries to close again after another 5 seconds. (Don't worry about modelling the delay of the gate opening and closing.)

On the next page is a domain model for the system based on the interface choices already made. The events, conditions, and actions in your state-machine model should be expressed in terms of elements in the domain model. You can declare events and abbreviations based on these elements, to simplify the expressions in your model.

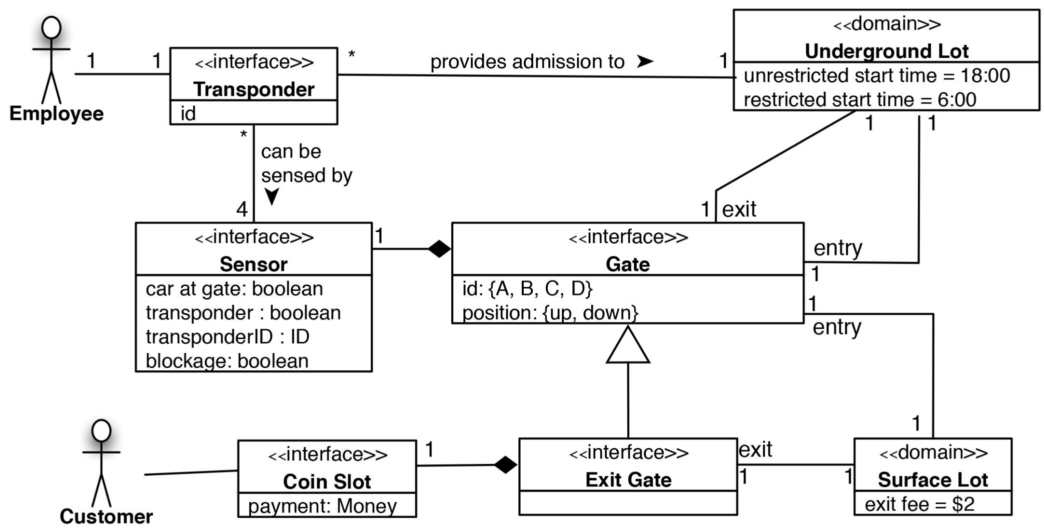
For full credit, your model should make the most effective use of UML state-machine modelling constructs. You must use a software modelling tool or drawing tool to create the state machine model that you include in your submission.

APPENDIX A: PRE-STUDY MATERIALS (ASSIGNMENT DESCRIPTION AND MARKING SCHEME)

CS445 / ECE451 / CS645 / SE643 -- Winter 2014 *Assignment 3*

1) State Machine Model (cont.)

Below is a domain model for the system that enforces the Bauer parking policies.



CS445/ECE451/CS645 - Winter 2014
Marking Scheme for Assignment 3
Out of a total of 70 marks

Question 1 (43 marks) State Machine Model

- (25 marks) Information Content
  - (2) detect cars (events that trigger transitions to open gates)
  - (2) detect transponders (events that trigger transitions to open gates)
  - (2) action to open / close gates
  - (1) detect fee (event that triggers transitions to open gate D)
  - (2) timeout (events that trigger transitions to close gates)
  - (2) gates stay open if blockage is detected
    - (1) blockage is detected at gate B at 6:00am on a weekday morning
  - (4) employees park underground for free
    - (2) including evenings and on weekends
  - (2) gate B remains open in evenings and on weekends
  - (5) model controls all four gates simultaneously
- (18) Model Quality
  - (4 marks) variable, event declarations
  - (4 marks) concurrency, state actions to simplify model
  - (5 marks) UML syntax
  - (5 marks) refers to interface elements in the domain model

APPENDIX B: MAIN STUDY MATERIALS

A Research Study on Understanding and Learning About the Efforts and Challenges of using UML Modelling tools

Recruitment Screener

Hello,

This is a recruitment screener for a study on observing modellers and understanding their difficulties and challenges to improve the quality of modelling tools. It is important to note that you will not be evaluated in any way and it is the tool that is under scrutiny, not you. The activity takes place in the University of Waterloo, room DC2551B. Participation is on a paid basis.

Are you interested in participating?

- **Yes.**
- **No.** *(Will thank the person and end the recruitment.)*

We have a few questions to ask you to see whether you fit the profile of the individuals we need for this study. After you answer the questions, we will take your information for review, and then we will contact you by email to let you know whether your background is a good fit, and to schedule your laboratory session.

A. Background and Demographic Information

Name: *Your name will not be associated with your data.*

Email:

1. Are you currently enrolled as a student at the University of Waterloo?

—Graduate —Undergraduate —No

2. Department/Major? ______________________________

APPENDIX B: MAIN STUDY MATERIALS

3. Have you taken a software engineering course that includes UML modelling?

—Yes —No

4. How would you rate your familiarity with UML?

(0)Unfamiliar

(1)Fairly Familiar (Novice)

(2)Familiar

(3)Very Familiar

(4)Strongly Familiar (Experienced)

5. How would you rate your familiarity with UML Class Diagrams?

(0)Unfamiliar

(1)Fairly Familiar (Novice)

(2)Familiar

(3)Very Familiar

(4)Strongly Familiar (Experienced)

6. How would you rate your familiarity with UML State Diagrams?

(0)Unfamiliar

(1)Fairly Familiar (Novice)

(2)Familiar

(3)Very Familiar

(4)Strongly Familiar (Experienced)

7. Do you have any experience with using modelling tools?

—Yes —No ;

8. (If Yes on Q7, survey monkey will show this question) In total, for how long have you had the experience i.e. how old is your experience (in month) ——

9. (If Yes on Q7, survey monkey will show this question) How frequently did

you use the tools?

- Several times a day
- 3-5 days a week
- 1-2 days a week
- Every few weeks
- Less often

10. (If Yes on Q7, survey monkey will show this question) Please list the name of the tools with which you have experience: ——————

11. (If Yes on Q7, survey monkey will show this question) How much of your experience with the modelling tools was in an industrial setting (in month)? ——

12. In this study, you will be asked to use one of the modelling tools of your choice. What modelling tool do you want us to prepare for you to use (e.g., MagicDraw, VisualParadigm, Papyrus, ArgoUML, etc.)? —————-

B. Confidentiality agreement, and permission to record audio and screen-capture

We also have some questions to make sure you understand and are comfortable with our procedure before you come in:

1. Are you willing to sign a standard consent form, which acknowledges that you agree to participate?

—Yes —No

2. Are you willing to be audio-recorded and your activities with the tool be screen-captured? (The purpose is so we can go back and capture more detailed notes. Videos are seen internally only by members of the research team - mainly the research student - who are interested in what you have to say.)

—Yes —No

APPENDIX B: MAIN STUDY MATERIALS

3. The compensation is 20.00 CAD and the chance to enter into a draw to win a prize of $200 gift card. Are you willing to participate based on the compensation?
—Yes —No

C. Screening Exercise: UML Knowledge Assessment

We would like to ask you a few questions[a] to assess if you are a good match with our study.

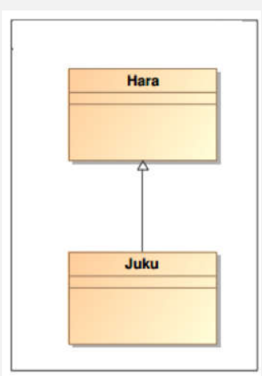
Question 1

What type of relationship is needed to represent the relationship between students and the courses they are enrolled in a university?

1) A one-to-one association.
2) A one-to-one composition.
3) A one-to-many association.
4) A one-to-many composition.
5) A many-to-many association.
6) A many-to-many composition

Question 2

Exhibit:



APPENDIX B: MAIN STUDY MATERIALS

Which of the following is true?

1) Juku is a subclass of Hara.
2) This is NOT a valid UML class diagram.
3) Every Juku has a reference to at least one Hara.
4) Juku is a subclass of Hara and at least one other class

Question 3

You can model the following situations with a state machine diagram:

1) Activities that are executed while the object is in a certain state or while a transition is occurring.
2) Events that trigger transitions.
3) The states that the object of a class can have.
4) Possible transitions from one state to another.
5) All of the above items.

Question 4

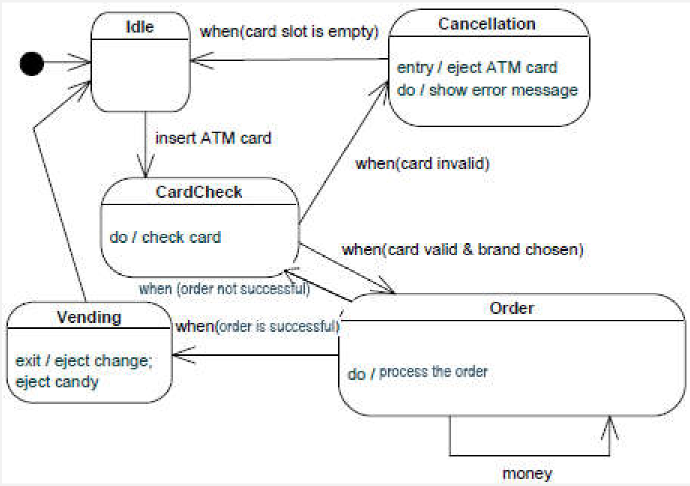
Which of the following statements are true?

1) Every class in a class diagram may or may not have attributes and operations.
2) An attribute can have a type.
3) Operations may have parameters and return values.
4) All of the above items.

Question 5

Exhibit:

APPENDIX B: MAIN STUDY MATERIALS



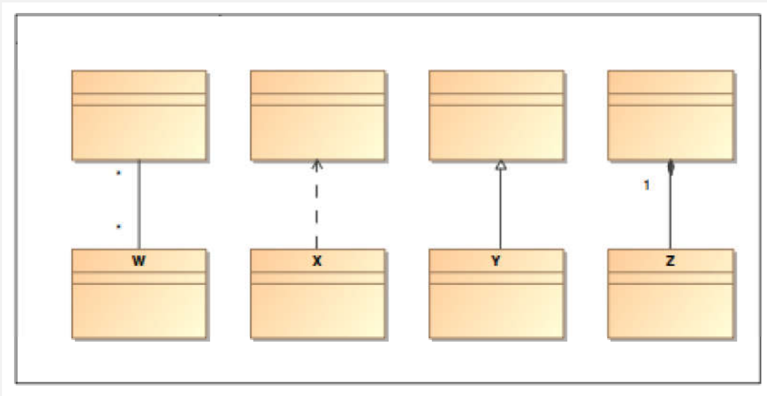
You are given the above state machine diagram. Which of the following statements about allowable state order is true?

1) money is an event.
2) money is a state.
3) When money occurs, the system will go to its Vending state.
4) None of the above.

Question 6

Exhibit:

APPENDIX B: MAIN STUDY MATERIALS



Which class has a superclass relationship?

1) W
2) X
3) Y
4) Z

Question 7

What does the syntax for labelling a transition look like?

1) event [guard] / action
2) [guard] action / event
3) action [guard] / event
4) [action] guard / event
5) [action] event / guard

Question 8

You want to model the following situation: A home delivery service has the two states Wait and Deliver. At the beginning, state Wait is active. As soon as a customer has ordered a product, a transition to state Deliver takes place. During the transition from state Wait to state Deliver, the order is processed. State Deliver stays active until the product has been

APPENDIX B: MAIN STUDY MATERIALS

delivered to the customer, triggering a transition to state Wait. Which transition expression would you select from the list below for the transition from state Wait to state Deliver?

1) order received / process order.
2) / process order
3) order received [process order]
4) [order received] / order is processed
5) [order] / process order
6) [order received] / process order

[a]The questions are originally taken from UML knowledge assessment exercise from the online sample practice tests for the Sun Certified Java Associate exams and the UML quiz website: http://elearning.uml.ac.at.

Domain Description

System Description: You are to create a UML State-Machine model of the proposed system to enforce the parking policies at the Bauer complex. Bauer provides parking for two types of users: customers and employees. There are two levels of parking: surface-level parking and underground parking. Surface-level parking is for customers only. They pay a $2 flat fee for parking. Employees park for free, but only if they park underground. Underground parking is also open to customers in the evenings, but of course customers must still pay ($2) to use it.

Gates control access to both levels of parking. Each gate controls traffic flowing in a single direction (e.g., traffic entering the complex), and it opens only for cars that approach the gate from the appropriate direction. Employees' cars have transponders that the gates can sense. Some gates open only for cars that have transponders. In addition, the system uses transponder data (IDs) to keep track of where employees' cars park. There are four gates that are controlled independently:

1) Gate **A** controls entry into the Bauer complex. It opens whenever it senses a car

APPENDIX B: MAIN STUDY MATERIALS

approaching from outside the Bauer complex. Cars that pass through gate A enter the surface-level parking lot.

2) Gate **B** controls access from surface-level parking to underground parking. Normally, it opens only if an approaching car has a transponder. But during evenings (18:00-6:00), it remains open to all cars.
3) Gate **C** opens whenever it senses a car approaching from the underground parking lot.
4) Gate **D** controls exits from the Bauer complex and collects parking fees. The gate opens whenever a $2 fee is paid (no change is provided). In addition, if an approaching car has a transponder, the gate opens if the car had parked underground (i.e., if the transponder triggered gate B since the car entered the complex).

Each of the gates closes 5 seconds after it opens, unless a blockage is detected, in which case it remains open and tries to close again after another 5 seconds. (Don't worry about modelling the delay of the gate opening and closing.)

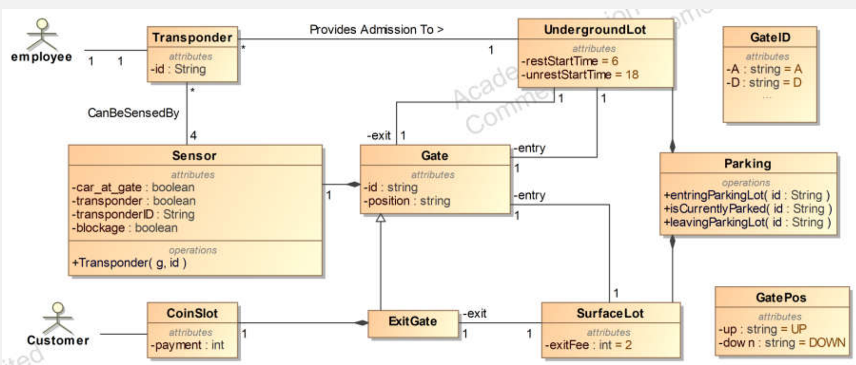


Fig. 11: Class diagram for the parking lot application.

Fig. 11 is the partial domain model for the system based on the interface choices already made. The events, conditions, and actions in your state-machine model should be expressed in terms of elements in the domain model. Moreover, the description of the domain model is given below. It helps you understand the domain, classes, attributes and the messages (operations) that you may need to know during the study. It is suggested that you take a brief look at the description before starting the tasks.

UndergroundLot: Represents the Underground parking of the parking lot.

- *restStartTime:* The Underground Lot has a restricted time for customers starting at 6:00am.
- *unrestStartTime:* The Underground Lot becomes unrestricted for customers starting at 18:00pm.

SurfaceLot: Represents the Surface-Level parking of the parking lot.

- *exitFee:* Represents the $2 parking fee that each customer must pay when exiting the lot.

GateID: Each Gate has an ID of either A, B, C, or D. GateID is an enum variable (class) that represents the id of each gate. Please note that, since the class diagram is not fully complete (intentionally), the id of B is not defined in this class.

GatePos: Gates can be either open or close, that is the gates' position can be either up or down. This enum class represents the allowable values for the gates' position.

Gate: Represents the Gates for both underground and surface-level parking lots.

- *id:* Each Gate has an id of the type GateID, that is a Gate can have an id of A, B, C, or D.
- *position:* Each Gate has a position attribute that is from the type of the class GatePos, and can have a value of up or down.

ExitGate: It shows the exit gate of the surface-level parking where the customer needs to put money in the coin slot.

CoinSlot: When exiting from the exit gate of the surface-level parking, the customer needs to pay the exit fee through a coin slot. The CoinSlot class represents the coin slot entity of the domain.

- *payment:* The amount that the customer has already inserted to the coin slot.

Sensor: Each Gate has a sensor the senses if a car is approaching the gate.

- *car_at_gate:* This is a boolean attribute that returns true if there is a car at the gate, and returns false if there is no car at the gate.
- *transponder:* Returns true if an approaching care has a transponder. Otherwise, returns false.
- *transponderID:* If an approaching care has a transponder, this attribute returns the ID of the transponder.
- *blockage:* Returns true in a case that a car has come through the gate, so that the gate does not close.
- *Transponder (g, id):* Returns true if a transponder (with id=id) is approaching gate **g** from the sensed direction.

Transponder: Each employee has a Transponder installed on their car. that provides admission to the underground parking and can be sensed by the sensors on the gates.

- *id:* Each Transponder has an ID.

Parking: This is a singleton class which records the states of cars with the transponder. Moreover, if a car enters to the parking, its transponder id should be recorded and be added to the collection of the parked cars' transponder ids. Also, when it leaves the parking, the car's transponder id must be removed from the list.

- *enteringParkingLot (id):* When a car is entering to the parking lot, this operation adds the transponder id of the car to the list of the transponder ids of the existing cars in the parking.
- *isCurrentlyParked (id):* This is a boolean operation which checks if a car with a transponder id had parked in the parking.
- *leavingParkingLot (id):* When a car is leaving the parking, the system uses this

APPENDIX B: MAIN STUDY MATERIALS

operation to remove the transponder id of the leaving car from the list of the transponder ids of the parked cars.

☐ I have read the system description carefully.

APPENDIX B: MAIN STUDY MATERIALS

Fig. 12 shows a partial state machine for the system. As can be seen in the state machine, the diagram is complete for the gates A and C, whereas the gates B and D still need to be developed more completely.

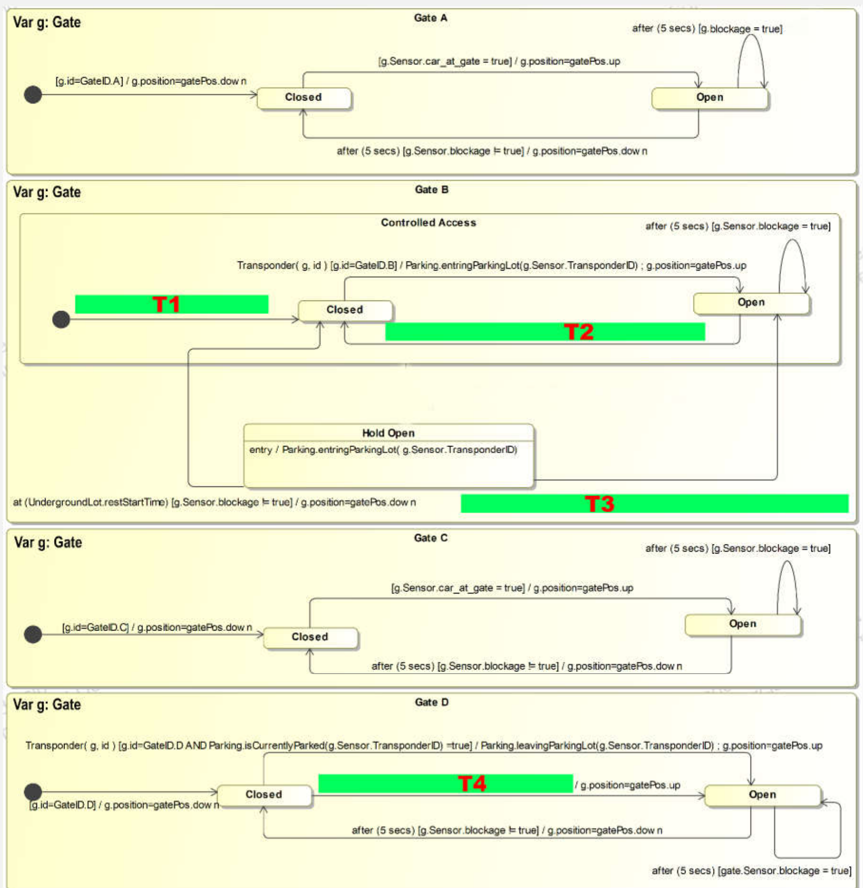


Fig. 12: Partial state-machine diagram for the parking lot application.

APPENDIX B: MAIN STUDY MATERIALS

Task Description

A modeller is responsible for editing several types of models such as Class Diagram, Sequence Diagram, State Diagram, Activity Diagram, and Use-Case Diagram. Assume that as a modeller, you are given a description of a system to enforce the parking policies at the Bauer complex, and your task is to update the State Diagram. Although your task is only to update/complete the State Diagram, you may find it necessary to also update the existing Class Diagram (i.e. adding a few model elements). You need to set the different parts of the transitions, namely, Triggering Event, Guard or Action, based on the description that is given to you for each part. Please remember to say your thoughts (**especially your challenges and difficulties, as well as your expectations from the tool**) loudly so that we can record them clearly.

Before starting your tasks, please answer the following questions.

APPENDIX B: MAIN STUDY MATERIALS

Please answer to the following questions based on your overall experience and expectation of using modelling tools.

1. How would you expect a modelling tool to facilitate (i.e. making the task easy or difficult) the following tasks for you?

Performing a series of sequencing actions in the right order (e.g., defining an element in the class diagram before using it in the state diagram).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Remembering the contextual information (e.g., remembering the name of the model elements when using them).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Consulting multiple diagrams (e.g., switching back and forth among different diagrams to remember the model elements that you may need to use).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Writing navigation expressions correctly (e.g., [Car.Driver.License.IssueDate > Car.Driver.BirthDate+18]).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Remembering the appropriate keywords and syntax of the language (e.g., whether you should use '==' or '=' for a condition).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Preventing the mismatch between the types of the variables that are used for the Left-Hand-Side and Right-Hand-Side of an assignment or a condition (e.g., [Gate.Position == up]).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Locating, understanding, and resolving the errors in the model (e.g., using a model element that is not defined in the class diagram or misspelling it).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult

APPENDIX B: MAIN STUDY MATERIALS

Part 1: In this part, your task is to set the transition expressions for the transitions shown in the gates B and D. For each transition, we provide you with the description of it and you only need to develop the transition using the tool accordingly.

Task1: Please develop the transition that is labelled as T1 in the diagram. Take the following steps one by one.

1) Please first Read the following description of the transition.
   - Triggering Event: No triggering event is required for this transition.
   - Guard: If the gate id is B.
   - Action: The Gate will initially go to the close state, that is the gate position should be set to down.

   ☐ I have read the transition's description carefully.

2) Please answer before starting the task:

		1	2	3	4	5	6	7	
How difficult do you expect the task to be using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

3) Perform the task and develop the Guard and Action parts of the transition expression.

   ☐ I am done with developing the transition.

4) Please answer once you have finished the task:

		1	2	3	4	5	6	7	
How difficult the task was based on your actual experience of using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

5) During the task, did you get any warning or error message indicating that the model is erroneous or problematic?

   ☐Yes ☐No

   If Yes, please answer the following questions:

APPENDIX B: MAIN STUDY MATERIALS

Area			1	2	3	4	5	6	7	
Effectiveness	Where you able to fix it?	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Helpfulness	Overall, the error message(s) and warnings helped me to solve the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Recovery	Overall, the tool provided me with error recovery techniques (or steps) that made it easier to fix the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree

APPENDIX B: MAIN STUDY MATERIALS

Task2: Please develop the transition that is labelled as T2 in the diagram. Take the following steps one by one.

1) Please first Read the following description of the transition (this is a timed transition).
   - Triggering Event: After 5 seconds trigger the transition to check if the gate can be closed.
   - Guard: If the gate is not blocked, that is the gate's sensor does not sense any blockage.
   - Action: Set the gate's position to be closed.

   ☐ I have read the transition's description carefully.

2) Please answer before starting the task:

		1	2	3	4	5	6	7	
How difficult do you expect the task to be using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

3) Perform the task and develop the Event, Guard and Action parts of the transition expression.

   ☐ I am done with developing the transition.

4) Please answer once you have finished the task:

		1	2	3	4	5	6	7	
How difficult the task was based on your actual experience of using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

5) During the task, did you get any warning or error message indicating that the model is erroneous or problematic?

   ☐Yes ☐No

   If Yes, please answer the following questions:

APPENDIX B: MAIN STUDY MATERIALS

Area			1	2	3	4	5	6	7	
Effectiveness	Where you able to fix it?	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Helpfulness	Overall, the error message(s) and warnings helped me to solve the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Recovery	Overall, the tool provided me with error recovery techniques (or steps) that made it easier to fix the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree

APPENDIX B: MAIN STUDY MATERIALS

Task3: Please develop the transition that is labelled as T3 in the diagram. Take the following steps one by one.

1) Please first Read the following description of the transition.
   - Triggering Event: At the start of the underground lot's restricted time.
   - Guard: If the gate's sensor senses blockage.
   - Action: Set the gate's position to be open.

   ☐ I have read the transition's description carefully.

2) Please answer before starting the task:

How difficult do you expect the task to be using the tool?		1	2	3	4	5	6	7	
	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

3) Perform the task and develop the Event, Guard and Action parts of the transition expression.

   ☐ I am done with developing the transition.

4) Please answer once you have finished the task:

How difficult the task was based on your actual experience of using the tool?		1	2	3	4	5	6	7	
	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

5) During the task, did you get any warning or error message indicating that the model is erroneous or problematic?

   ☐Yes ☐No

   If Yes, please answer the following questions:

APPENDIX B: MAIN STUDY MATERIALS

Area			1	2	3	4	5	6	7	
Effectiveness	Where you able to fix it?	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Helpfulness	Overall, the error message(s) and warnings helped me to solve the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Recovery	Overall, the tool provided me with error recovery techniques (or steps) that made it easier to fix the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree

APPENDIX B: MAIN STUDY MATERIALS

Task4: Please develop the transition that is labelled as T4 in the diagram. Take the following steps one by one.

1) Please first Read the following description of the transition.
   - Triggering Event: No triggering event is required for this transition.
   - Guard: The payment received at the gate's Coin Slot is greater than or equal to the gate's Exit Fee.
   - Action: Set the position of the gate to be open.

   ☐ I have read the transition's description carefully.

2) Please answer before starting the task:

		1	2	3	4	5	6	7	
How difficult do you expect the task to be using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

3) Perform the task and develop the Guard part of the transition expression.

   ☐ I am done with developing the transition.

4) Please answer once you have finished the task:

		1	2	3	4	5	6	7	
How difficult the task was based on your actual experience of using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

5) During the task, did you get any warning or error message indicating that the model is erroneous or problematic?

   ☐Yes ☐No

   If Yes, please answer the following questions:

APPENDIX B: MAIN STUDY MATERIALS

Area			1	2	3	4	5	6	7	
Effectiveness	Where you able to fix it?	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Helpfulness	Overall, the error message(s) and warnings helped me to solve the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Recovery	Overall, the tool provided me with error recovery techniques (or steps) that made it easier to fix the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree

APPENDIX B: MAIN STUDY MATERIALS

Part 2: Sometimes it is possible that a modeller does not create well-formed, correct and consistent diagrams. It can be because of either not caring about these errors, or not even knowing about their existence. In this part, your task is to modify or observe the model to find inconsistency or error in the model, if any, and fix them.

Task5: Assume that you are supposed to change the name of the gates from A and D to GA and GD respectively in the Class diagram.

1) Please answer before starting the task:

		1	2	3	4	5	6	7	
How difficult do you expect the task to be using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

2) Realize the change on the model. Please note that, this change may or may not cause inconsistencies in the model. If you believe that there are inconsistencies, please resolve them appropriately.

   ☐ I am done with changing the model.

3) Please answer once you have finished the task:

		1	2	3	4	5	6	7	
How difficult the task was based on your actual experience of using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

4) During the task, did you get any warning or error message indicating that the model is erroneous or problematic?

   ☐Yes ☐No

   If Yes, please answer the following questions:

APPENDIX B: MAIN STUDY MATERIALS

Area			1	2	3	4	5	6	7	
Effectiveness	Where you able to fix it?	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Helpfulness	Overall, the error message(s) and warnings helped me to solve the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Recovery	Overall, the tool provided me with error recovery techniques (or steps) that made it easier to fix the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree

APPENDIX B: MAIN STUDY MATERIALS

Task6: Assume that you are supposed to observe the state diagram and see if you can find any issue within the region of Gate A.

1) Please answer before starting the task:

		1	2	3	4	5	6	7	
How difficult do you expect the task to be using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

2) Please observe the state diagram and report if you find any issue within the region of Gate A in the following comment box.

3) Please answer once you have finished the task:

		1	2	3	4	5	6	7	
How difficult the task was based on your actual experience of using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

4) During the task, did you get any warning or error message from the tool indicating that the model is erroneous or problematic?

☐Yes ☐No

If Yes, please answer the following questions:

Area			1	2	3	4	5	6	7	
Effectiveness	Where you able to fix it?	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Helpfulness	Overall, the error message(s) and warnings helped me to solve the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Recovery	Overall, the tool provided me with error recovery techniques (or steps) that made it easier to fix the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree

APPENDIX B: MAIN STUDY MATERIALS

Task7: Assume that you are supposed to observe the state diagram and see if you can find any issue within the region of Gate C.

1) Please answer before starting the task:

		1	2	3	4	5	6	7	
How difficult do you expect the task to be using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

2) Please observe the state diagram and report if you find any issue within the region of Gate C in the following comment box.

3) Please answer once you have finished the task:

		1	2	3	4	5	6	7	
How difficult the task was based on your actual experience of using the tool?	Very Easy	☐	☐	☐	☐	☐	☐	☐	Very Difficult

4) During the task, did you get any warning or error message from the tool indicating that the model is erroneous or problematic?

☐Yes ☐No

If Yes, please answer the following questions:

Area			1	2	3	4	5	6	7	
Effectiveness	Where you able to fix it?	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Helpfulness	Overall, the error message(s) and warnings helped me to solve the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree
Recovery	Overall, the tool provided me with error recovery techniques (or steps) that made it easier to fix the problem.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree

APPENDIX B: MAIN STUDY MATERIALS

☐ I am done with performing all the tasks.

If you have finished your tasks, please answer the following questions.

Please answer to the following questions based on the experience you just had in the study.

1. How would you rate the difficulty of the following tasks when modelling?

Performing a series of sequencing actions in the right order (e.g., defining an element in the class diagram before using it in the state diagram).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Remembering the contextual information (e.g., remembering the name of the model elements when using them).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Consulting multiple diagrams (e.g., switching back and forth among different diagrams to remember the model elements that you may need to use).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Writing navigation expressions correctly (e.g., [Car.Driver.License.IssueDate > Car.Driver.BirthDate+18]).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Remembering the appropriate keywords and syntax of the language (e.g., whether you should use '==' or '=' for a condition).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Preventing the mismatch between the types of the variables that are used for the Left-Hand-Side and Right-Hand-Side of an assignment or a condition (e.g., [Gate.Position == up]).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult
Locating, understanding, and resolving the errors in the model (e.g., using a model element that is not defined in the class diagram or misspelling it).	Very Easy	○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7	Very Difficult

2. Please write down any other challenges or difficulties that you have experienced when using modelling tools (if not listed above).

APPENDIX B: MAIN STUDY MATERIALS

Please fill the following task satisfaction form.

			1	2	3	4	5	6	7		NA
1	Overall, I am satisfied with how easy it is to use this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
2	It was simple to use this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
3	I can effectively complete my work using this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
4	I am able to complete my work quickly using this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
5	I am able to efficiently complete my work using this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
6	I feel comfortable using this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
7	It was easy to learn to use this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
8	I believe I became productive quickly using this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
9	The system gives error messages that clearly tell me how to fix problems.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
10	Whenever I make a mistake using this tool, I recover easily and quickly.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
11	The information (such as error message, icons, and on-screen messages) provided with this system is clear.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
12	It is easy to find the information I needed.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
13	The information provided for the tool is easy to understand.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
14	The information (provided by the tool) is effective in helping me complete the tasks and scenarios.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
15	The organization of information on the tool screen is clear.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
16	The interface of this tool is pleasant.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
17	I like using the interface of this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
18	This tool has all the functions and capabilities I expect it to have.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐
19	Overall, I am satisfied with this tool.	Strongly Disagree	☐	☐	☐	☐	☐	☐	☐	Strongly Agree	☐

Fig. 13: The CSUQ developed by Lewis.

Thank you for your time and participation in the study.

Study Description and Preparation Script

Hello,

Thank you for accepting our invitation and for your participation in the study. The goal of the study is to observe and analyse modellers when they are performing their modelling

APPENDIX B: MAIN STUDY MATERIALS

tasks, and try to understand the most pressing difficulties and challenges that they might face when using modelling tools.

During the study, you will be given a set of modelling tasks to perform. You will perform the tasks given to you in the modelling tool that is prepared and is open for you to work. In the meantime, we will be capturing the computer screen for further analysis. Also, as you work on the tasks, I will ask you to do what we call 'think out loud'. What that means is that I want you to say out loud what you are thinking as you work. You might speak about the steps in the task as you complete them, as well as your expectations and evaluation statements. Let us show you what we mean by showing you a demo:

https://www.youtube.com/watch?v=g34tOmyKaMM

Do you see what I mean about think out loud? I would like to remind you to please speak loudly since we will be recording your voice. Your voice remains confidential, and the researcher is the only person who will listen to it during later analysis.

The main task of the observer is to jot down what happens. In addition, we might prompt you by asking a few questions to make the session more effective. Such questions may include:

What are you thinking now?

Why did you do that?

please keep talking ...

Now, before we start the experiment, I would like to emphasize a few important notes to you:

1- In the course of the study, you are the expert.

2- Remember that you will not be evaluated in any way. It is the tool that is under scrutiny, not you.

3- You should at all times comment liberally on your actions, intentions and thoughts.

4- You should be at ease and relaxed. Please keep in mind that, the think-aloud method is informal, and the most effective single way to maximise its effectiveness is to create an informal atmosphere. That's why we want you to be relaxed. Again, it is the tool that is under scrutiny, not you.

5- You should try to find your own way as much as possible. You may ask questions at

APPENDIX B: MAIN STUDY MATERIALS

any point in the process, but I may not answer them or only answer those questions that are related to clarifying and understanding the application domain. I will only be able to give you the bare minimum of help. We apologise in advance for this.

6- You can stop a task at any time if it becomes uncomfortable.

7- I will not tell you when you have completed task; you must determine this on your own.

8- If a task is taking longer than what is expected, I might ask you to ignore the current task and move on to the next task.

Now, if you have any remaining questions, please ask them, otherwise, let's begin the tasks.

Thank you.