Quality and comprehension of UML interaction diagrams—an experimental comparison

Chanan Glezer*, Mark Last, Efrat Nachmany, Peretz Shoval

Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

Received 29 September 2004; revised 24 December 2004; accepted 10 January 2005

Abstract

UML (Unified Modeling Language) is a collection of somewhat overlapping modeling techniques, thus creating a difficulty in establishing practical guidelines for selecting the most suitable techniques for modeling OO artifacts. This is true mainly with respect to two types of interaction diagrams: Sequence and collaboration. Attempts have been made to evaluate the comprehensibility of these diagram types for various types of applications, but they did not address the issue of quality of diagrams created by analysts. This article reports the findings from a controlled experiment where both the comprehensibility and quality of the interaction diagrams were investigated in two application domains: management information systems (MIS) and real-time (RT) systems.

Our results indicate that collaboration diagrams are easier to comprehend than sequence diagrams in RT systems, but there is no difference in comprehension of the two diagram types in MIS. Irrespective of the diagram type, it is easier to comprehend interaction diagrams of MIS than of RT systems. With respect to diagram quality, in the case of MIS, analysts create collaboration diagrams of better quality than sequence diagrams, but there is no significant difference in quality of diagrams created in RT systems. Irrespective of the diagram type, more correct diagrams are created in MIS applications than in RT applications.

Keywords: Collaboration diagrams; Experiments; Interaction diagrams; Method/model evaluation; Object-oriented analysis; Sequence diagrams; Software engineering; UML.

1. Introduction

The Unified Modeling Language (UML) accommodates several types of diagrams in an attempt to model the structural and dynamic aspects of systems. The broad acceptance of UML as a language for Object Oriented (OO) analysis and design has fundamentally been based on its usefulness as a means of communication among all parties involved in the application development process [1]. Moreover, the prominent OMG Model Driven Architecture (MDA) initiative [2], which enables interoperability of applications across middleware boundaries, is already adopted by many companies in a plethora of application domains. Nevertheless, the nature of UML, being a collection of somewhat overlapping modeling techniques, and its recent proliferation in heterogeneous application domains, creates a difficulty in establishing practical guidelines for selecting the most suitable UML techniques for modeling OO artifacts.

Methodologies in general (and UML in particular) can be evaluated and compared on various dimensions, i.e. quality of the analysis or design products, comprehensibility, learnability, ease of use, user/developer satisfaction or preference, and more [3]. Criteria such as: degree of chunking and fragmentation of requirements, the number of simultaneous items or semantic processing required for creating a schema, and more are proposed in order to promote the performance and readability of conceptual modeling in terms of performance and readability [4].

Several attempts have been made to evaluate the semantic comprehension of object-oriented diagrams. Diagrammatic reasoning with multiple diagrams is comprised of searching for related information (perceptual reasoning), and developing hypotheses about the target system...
(conceptual reasoning) [5]. The findings from a controlled experiment indicate that perceptual and conceptual integration processes are facilitated by incorporating visual cues and contextual information in the multiple diagrams as representation aids [5]. Otero and Dolado [1,6] studied the use of interaction diagrams in UML for applications of varying complexity, ranging from management information systems, through embedded systems, to real-time reactive systems. Their work focused on evaluating comprehensibility of diagrams, assuming that a fit between the application domain and diagram type will positively impact the level of semantic comprehension of a given system’s specification and, in turn, will indirectly facilitate the use of the proper diagrams in system construction.

No attempt was made, however, to directly investigate the quality of interaction diagrams produced in various scenarios under an objective yardstick, a lacuna which the current study attempts to address. Thus, this article reports the findings from a controlled experiment conducted in a university setting, where both the semantic comprehension and quality of UML interaction diagrams was investigated for various application domains.

The rest of the paper is organized as follows: Section 2 discusses UML and interaction diagrams. Section 3 describes related work on previous empirical comparisons of software modeling frameworks in general, and specifically of UML interaction diagrams. Section 4 describes our experiment, including the dependent and independent variables, the research hypotheses, the participants, the documents included in the experiment, and the experimental design. It also discusses threats to validity and how we treated them. Section 5 presents the results in terms of diagrams comprehensibility, quality of diagram construction, time taken to complete the comprehension and the construction tasks, perceived comprehensibility and perceived ease of construction. Section 6 discusses the results of the experiments in light of the hypotheses and related work, and finally, Section 7 outlines the limitations of this study and suggests topics for future research.

2. UML and interaction diagrams

2.1. The unified modeling language (UML)

As strategic value of software increases for many companies, the industry is eagerly looking for techniques that automate the production of software in an attempt to improve quality, reduce costs and expedite time-to-market. These techniques include, among others: component technology, visual programming, patterns, and modeling frameworks.

![Fig. 1. Sequence Diagram - Library MIS.](image-url)
UML is a highly popular and prominent object modeling framework which helps in specifying, visualizing, and documenting models of software systems. UML attempts to make business functionality more complete and correct in meeting user-needs. In addition, UML-based software designs better support requirements for scalability, robustness, security, extendibility, and other characteristics, before implementation in code renders changes difficult and expensive to make [7].

UML defines twelve types of artifacts in the form of diagrams which are divided into the following three categories: class, object, component, and deployment diagrams represent the application’s static structure; use case, interaction (sequence and collaboration), activity and state charts represent the application’s dynamic behavior; and packages, subsystems, and models represent the application modules organization [7].

The nature of UML, being a collection of somewhat overlapping modeling techniques, and its recent proliferation in heterogeneous application domains, creates a difficulty in establishing practical guidelines for selecting the most suitable UML techniques for modeling OO artifacts. This choice is expected to affect design efficiency, reliability and the semantic comprehension of the design outputs. In reality, various diagrams have been frequently used in systems analysis and design without proper verification of their usability and with a substantial impact on the computational cost of working with them [8].

2.2. Interaction diagrams

The focus of this study is on UML’s interaction diagrams, which depict a pattern of interaction among objects. Interaction diagrams come in two forms emphasizing different aspects of an interaction. The first form, termed Sequence Diagram, depicts an explicit sequence of stimuli messages exchanged among object instances participating in the interaction. Sequence diagrams include lifelines for the instances used to portray the temporal dimension of the modeled pattern. They were introduced into OO modeling by Objectory [9], and since have become popular in OO methods to represent communications between objects [10]. In the telecommunications area, sequence diagrams have been standardized as Message Sequence Charts (MSC) [11] and used in combination with the Specification Description Language to illustrate sequences of messages between a system and its environment.

Two examples of sequence diagrams are presented in Figs. 1 and 2. Fig. 1 is an example taken from a Library MIS (based on [1]). This example models an interaction between a student wishing to borrow a book and a system maintaining information on bibliographic items (journal and book titles). Specifically, the system handles the following tasks: issuing and canceling a title reservation, borrowing an item, returning an item and maintenance (adding/removing items from catalogue, updating borrower’s status, and adding new titles).

![Fig. 2. Sequence Diagram - Security Monitoring System.](image-url)
Fig. 2 is an example from a real-time system. It models the interaction between a security guard and a real-time security monitoring system which monitors a set of detectors capable of alerting a security guard in the case of a fire or a breach. In the event one of the detectors is activated, the siren is turned on. The monitoring system is activated and deactivated using a key.

The second form of an interaction diagram, termed Collaboration Diagram, is a directed graph where nodes represent communicating entities and edges represent communications. The edges are numbered to represent the order of communications. Collaboration diagrams have been introduced by Wirfs-Brock et al. [12] and have, in various guises, become popular in OO methods to represent message exchanges between objects. In UML collaboration diagrams can be extended with annotations that represent data flows between communicating objects and various adornments that represent the way in which communications are implemented [10]. Figs. 3 and 4 depict examples of collaboration diagrams equivalent to the sequence diagrams described in Figs. 1 and 2, respectively.

Collaboration diagrams differ from other communication-oriented diagrams such as Data Flow Diagrams (DFD), Activity Diagrams, Activity Charts, and Block Diagrams, in that their nodes represent objects, not activities. This also sets them apart from Object Communication Diagrams (OCD), because OCD represent object classes, not individual object instances. In addition, just as do sequence diagrams, collaboration diagrams represent the sequence of messages in one particular scenario, whereas all other communication diagrams represent possible communications in all possible scenarios [10].

Sequence and collaboration diagrams express similar information but depict it in a different way. Both diagrams are considered symmetric and it is therefore possible to convert a sequence diagram to a collaboration diagram and vice versa [13]. The main difference between the diagrams is that a sequence diagrams emphasizes the temporal dimension, exploiting the lifelines artifact, and is therefore assumed to be better in depicting an order of events or pause between events, which is an important feature when modeling complex and real time systems [7,13–15]. A sequence diagram is capable of visually modeling a ‘race condition’ which occurs when two events interrupt precisely at the same moment (i.e. a cellular phone receives an incoming call precisely when the user is dialing and preparing an outgoing call) [16,17]. On the other hand, a sequence diagram does not portray the interaction among objects exploited by a system. If a system is complex, it might therefore be difficult to infer the mutual relationship and messages relayed between the objects using a sequence diagram.

A collaboration diagram addresses the loopholes of a sequence diagram by depicting relationships among involved objects. A collaboration diagram is therefore recommended for supplementing both class and use-case (static-view) diagrams because it depicts the interactions among objects (dynamic-view) [16,17]. Moreover, a collaboration diagram is claimed to enable better modeling of complex branching and concurrent activation of multiple objects.
processes [18,19], or control of multiple threads [15]. A collaboration diagram, however, does not capture the temporal dimension and the relative order of messages exchanged between the objects needs to be enumerated explicitly. Thus, collaboration diagrams seem to be more adequate for modeling procedural task such as those supported by management information systems [7,13–15].

3. Related work on the evaluation of software models and interaction diagrams

3.1. Empirical evaluation of software models and UML

Software modeling frameworks have been evaluated extensively in the literature. In their review and synthesis of research on data modeling, Topi and Ramesh [20] surveyed many studies that appeared between 1978 and 2001, which employ mainly laboratory experiments to evaluate the usability of data models/methods. They found that the dependent variables in that research are divided into two main categories: user performance and user attitude. Use performance is divided into three subcategories: model correctness, time used to create the solution, and declarative knowledge (understanding of the notation). In most cases, the correctness of the model has been measured according to the degree to which it corresponds to a predefined solution. User attitude includes mainly preference to use a certain model, and perceived ease of use. In this section we review briefly some of these experimental studies.

Shoval and Even-Chaime [21] compared two methods for designing a database schema: Normalization and NIAM (Nijssen Information-Analysis Method), [22]. Their experiment revealed that the quality of the database schemata designed using Normalization was better than those designed using IA, that Normalization requires less time than IA to perform and is also the preferred method by analysts. Batra and Antony [23] compared Entity-Relationship (ER) with Relational modeling from the point of view of analyst performance. They found that ER diagram reached significantly higher grades compared to relations. Kim and March [24] did not find significant differences in terms of user comprehension between EER (Extended ER) and NIAM from the point of view of user comprehension of diagrams. Shoval and Frumermann [25] compared EER and OO models from the perspective of user comprehension of EER and OO (class) diagrams. They found a significant difference in comprehension of ternary relationships in favor of the EER model, and no significant difference in comprehension of other constructs (e.g. binary relationships). In a follow up study, Shoval and Shiran [26] compared the same EER and the OO models from the point of view of analysts/designers, i.e. quality of schema specification. The authors found that an EER model is better than an OO model in specifying unary and ternary relationships, but there were no significant differences in specifying other constructs. They also found that it takes less time to create EER schemas; and that designers prefer modeling with EER.

Peleg and Dori [27] compared two methodologies: OPM/T, a variant of OPM for real-time systems; and

![Collaboration Diagram - Security Monitoring System.](image-url)
the comprehension of the dynamic models in OO designs depends on the diagram-type and on the complexity of the application (as explained above). With regards to sequence diagram, the results indicated that complexity of designs and application domains do not have an effect on the time needed for understanding them. Sequence diagram designs are therefore more comprehensible as they are not affected by the complexity of the application and their comprehension implies more correct answer in less time, compared to other diagrams. On the other hand, with regards to the collaboration diagram, the results indicate a negative correlation with the complexity of the application: as the complexity of the application domain increases, more time is invested in understanding this diagram and less correct answers are given. When the dynamic behavior is modeled using a collaboration diagram, its design document turns out to be less comprehensible than if a sequence or state diagram had been used. Concretely, this finding suggests that collaboration diagrams should be avoided for real time (especially non reactive) applications, since they are more suitable for modeling management information systems.

In a subsequent study, Otero and Dolado [6] performed an experiment comprising of two parts. The first part was a repetition of their earlier study from 2002. Both of the studies are identical as per the basic design, types of diagrams and design documents employed. On the other hand, the repetition study may be considered more powerful because an independent variable, termed Group, has been added and each subject was assigned at random to one of six groups. The assignment to a Group corresponds to the sequence or order of presentation of the design documents. In this way the presence of the learning effect caused by practice or sequence may be detected and measured.

With regard to the real time non-reactive application (TEL), in the original experiment there was no difference between representing interactions between objects in time (sequence diagram) or in space (collaboration diagram). On the other hand, in the repetition experiment, it was easier to semantically interpret the design when the dynamic diagram used was a sequence diagram, rather than a collaboration or state diagram. In the case of a real-time reactive system (DIC), findings in both experiments indicate that the highest number of correct answers (NRESP) is reached when the dynamic model is a state diagram, and that sequence diagrams are preferred over collaboration diagrams. Finally, in the case of a management information system (LIB), both studies did not find a significant difference in the level of comprehension for state and collaboration diagrams, though the average time needed in the original experiment was significantly higher in the case of a collaboration diagram.

The second part of the 2004 experiment examined which combinations of dynamic diagrams (sequence-collaboration, collaboration state, or sequence-state) improve the comprehension of the dynamic models in OO designs. The authors found that in more cases OPM specifications were more comprehensible than OMT specifications. The authors also found that OPM analysts produced significantly better diagrams than OMT analysts. They explained that OPM users made fewer errors because OPM uses a single model, as opposed to the multiple models in OMT.

Kabeli and Shoval [3] compared OPM and FOOM (Functional and Object-Oriented Methodology) in two controlled experiments from the point of view of users and analysts. Their main results were that FOOM analysis specifications are more comprehensible to users than those based on OPM, whereas in terms of the data model there is no significant difference. From the analysts’ point-of-view of, they compared quality, i.e. correctness of specifications. The main results of these experiments were that analysts create more correct specifications with FOOM methodology than with OPM.

### 3.2. Empirical comparisons of UML interaction diagrams

In the context of UML and specifically interaction diagrams, most of the extant work focused on conceptual analysis and comparison of the features of collaboration and sequence diagrams. The only empirical research that we know of comparing interaction diagrams was performed by [1,6] who performed a set of experiments in an attempt to investigate the comprehensibility of interaction diagrams in UML.

In the first study [1], eighteen students of Informatics analyzed three types of diagrams: sequence-, collaboration- and state-diagrams, within three different applications and application domains: a) a Cellular Telephone (TEL), which is real-time non-reactive (synchronous) system incapable of handling unexpected interrupting events; b) a Library Management Information System (LIB); and c) a Digital Dictaphone (DIC), which is a real time reactive (asynchronous) system, capable of handling unexpected interrupting events. Otero and Dolado ranked the complexity of the three applications by exploiting the OO metrics adapted to interaction diagrams at the operation level [29]. For this purpose, the three applications were ranked according to their average operation size, average number of parameters and number of iterative (looping) and alternative messages. Based on this comparison framework, the applications’ complexity was ranked from high to low as: Dictaphone > Library > Telephone.

Since the sample size at hand was relatively small, Otero and Dolado [1] performed a 3X3 factorial design with repeated measures in an attempt to overcome learning and maturation effects. The two dependent variables were measured on a ratio scale and included: a) elapsed total time for answering multiple-choice questions, and b) the number of correct answers. Their main conclusion was that
regardless of the application domain, a higher semantic comprehension of the application is achieved whenever the dynamic behavior is modeled by using the pair sequence-state diagrams.

The experiments described above suffer from several limitations. First, they compare non-equivalent types of diagrams because State diagrams are analyzed as equivalent to interaction diagrams (sequence and collaboration). Second, the 2004 study evaluated pairs of models without providing sufficient rational and proving that the pairs are in fact interchangeable. Third, they did not address the issue of building diagrams and the quality of the ‘end-product’ with regards to different application domains.

The purpose of our study is to fill the research gaps on UML interaction diagrams. As we have seen, prior research has mainly focused on comprehension of diagrams, but the issue of quality of diagrams constructed using different diagram types has not been addressed yet. By investigating performance in terms of comprehension, quality, time and user/analyst preference in different application settings, the findings from this study are expected to provide a wide-angle user/analyst preference in different application settings, the performance in terms of comprehension, quality, time and diagram types has not been addressed yet. By investigating issue of quality of diagrams constructed using different has mainly focused on comprehension of diagrams, but the UML interaction diagrams. As we have seen, prior research regards to different application domains.

The controlled variables are the tasks and the subjects: in the ‘comprehension’ experiment the task was to answer a questionnaire which measures comprehension of a given diagram; in the ‘quality’ experiment the task was to construct a diagram. The subjects in the two tasks were randomly divided into four groups, as will be explained later on.

The experiment was divided into sessions. In Session 1 we conducted the ‘comprehension’ experiment: subjects were asked to express comprehensibility of a given interaction diagram by answering multiple-choice questions. In Session 2 we conducted the ‘quality’ experiment: subjects were asked to construct an interaction diagram based on a brief narrative specification of the system, a class diagram and a use case diagram. Quality was measured by the correctness of the created diagrams. In addition to performing the tasks, the subjects were required to record the overall time spent to complete each session. At the end, they were also asked to express their subjective opinions on diagrams comprehension and construction.

4. The experiment

4.1. Experimental design and variables

The goal of this study is to evaluate and compare sequence diagrams and collaboration diagrams from two main perspectives: analysts and users. We conducted two controlled experiments: one from a users’ perspective and the other from analysts; perspective. From the users’ perspective, we are interested in the comprehensibility (i.e. understandability) of the diagram, while from analysts’ perspective, we are interested in the quality (i.e. correctness and completeness) of the diagrams. The two types of diagrams are evaluated in two types of applications: a management information system and a real-time reactive system. Fig. 5 depicts the experimental design of the study (it actually describes the two experiments.)

In the ‘comprehensibility’ experiment we compare the comprehensibility of diagrams by users, the time it takes to comprehend the diagrams and the perceived (subjective) eased of comprehension. We measured the following three variables:

- $P_c$ ranking of subject’s perceived comprehensibility of a certain diagram
- $T_c$ total time spent to answer questions on diagrams comprehension
- $C$ total score for questions on diagrams comprehension

In the ‘quality’ experiment we compare the quality of diagrams as created by analysts, the time it takes to create the diagrams and the perceived ease of constructing them. We measured the following three variables:

- $Q$ total score for quality of an interaction diagram constructed
- $T_q$ total time spent on diagram construction
- $P_q$ ranking of subject’s perceived ease of construction of a certain diagram

In both experiments the independent variables are the two types of diagrams: sequence (Seq) and collaboration (Col), and two types of systems: a security system, representing a real-time reactive system, involving time dimension and concurrency (RT); and a library system, representing a management information system (MIS).

Our objective is to address the following questions:

(1) Is there a significant difference between a sequence and a collaboration diagram (in terms of quality as well as comprehensibility) for dynamic modeling of a real-time system?
(2) Is there a significant difference between a sequence and a collaboration diagram (again, in terms of quality and comprehensibility) for dynamic modeling of a management information system?

As we have seen, previous research indicated that sequence diagram has an advantage over collaboration diagram in the representation of the temporal order, which is
particularly important in real-time systems where the processes take pre-defined time slots and may be processed concurrently [1,6,7,13–15] On the other hand, collaboration diagram better depicts static relationships between objects. These relationships are important in management information systems, since they are used to exchange messages between objects.

Since in this study we are comparing between the two types of interaction diagrams using six dependent variables (as detailed above) and the comparison is performed separately for two types of applications (RT vs. MIS), we have a total of $6 \times 2 = 12$ statistical tests. For each test, the null and the alternative hypotheses are formulated in Table 1 below.

Table 1
Statistical tests performed in the experiment

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>MIS</th>
<th>RT system</th>
</tr>
</thead>
<tbody>
<tr>
<td>C: score for diagram comprehensibility</td>
<td>$H_{a0}: \mu_{c}, \text{MIS, col} \neq \mu_{c}, \text{MIS, seq}$</td>
<td>$H_{b0}: \mu_{c}, \text{RT, col} \neq \mu_{c}, \text{RT, seq}$</td>
</tr>
<tr>
<td>$H_{a1}: \mu_{c}, \text{MIS, col} &lt; \mu_{c}, \text{MIS, seq}$</td>
<td>$H_{b1}: \mu_{c}, \text{RT, col} &gt; \mu_{c}, \text{RT, seq}$</td>
<td></td>
</tr>
<tr>
<td>Q: score for quality of diagram constructed</td>
<td>$H_{c0}: \mu_{q}, \text{MIS, col} \neq \mu_{q}, \text{MIS, seq}$</td>
<td>$H_{d0}: \mu_{q}, \text{RT, col} \neq \mu_{q}, \text{RT, seq}$</td>
</tr>
<tr>
<td>$H_{c1}: \mu_{q}, \text{MIS, col} &gt; \mu_{q}, \text{MIS, seq}$</td>
<td>$H_{d1}: \mu_{q}, \text{RT, col} &lt; \mu_{q}, \text{RT, seq}$</td>
<td></td>
</tr>
<tr>
<td>$T_{c}$: time (min.) spent on diagram comprehension</td>
<td>$H_{e0}: \mu_{tc}, \text{MIS, col} \neq \mu_{tc}, \text{MIS, seq}$</td>
<td>$H_{f0}: \mu_{tc}, \text{RT, col} \neq \mu_{tc}, \text{RT, seq}$</td>
</tr>
<tr>
<td>$H_{e1}: \mu_{tc}, \text{MIS, col} &gt; \mu_{tc}, \text{MIS, seq}$</td>
<td>$H_{f1}: \mu_{tc}, \text{RT, col} &lt; \mu_{tc}, \text{RT, seq}$</td>
<td></td>
</tr>
<tr>
<td>$T_{q}$: time (min.) spent on diagram construction</td>
<td>$H_{g0}: \mu_{tq}, \text{MIS, col} \neq \mu_{tq}, \text{MIS, seq}$</td>
<td>$H_{h0}: \mu_{tq}, \text{RT, col} \neq \mu_{tq}, \text{RT, seq}$</td>
</tr>
<tr>
<td>$H_{g1}: \mu_{tq}, \text{MIS, col} &gt; \mu_{tq}, \text{MIS, seq}$</td>
<td>$H_{h1}: \mu_{tq}, \text{RT, col} &lt; \mu_{tq}, \text{RT, seq}$</td>
<td></td>
</tr>
<tr>
<td>$P_{c}$: subjective (perceived) comprehensibility of a certain diagram type</td>
<td>$H_{i0}: \mu_{pc}, \text{MIS, col} \neq \mu_{pc}, \text{MIS, seq}$</td>
<td>$H_{j0}: \mu_{pc}, \text{RT, col} \neq \mu_{pc}, \text{RT, seq}$</td>
</tr>
<tr>
<td>$H_{i1}: \mu_{pc}, \text{MIS, col} &lt; \mu_{pc}, \text{MIS, seq}$</td>
<td>$H_{j1}: \mu_{pc}, \text{RT, seq} &lt; \mu_{pc}, \text{RT, col}$</td>
<td></td>
</tr>
<tr>
<td>$P_{q}$: subjective (perceived) ease of construction of a certain diagram type</td>
<td>$H_{k0}: \mu_{pq}, \text{MIS, col} \neq \mu_{pq}, \text{MIS, seq}$</td>
<td>$H_{l0}: \mu_{pq}, \text{RT, col} \neq \mu_{pq}, \text{RT, seq}$</td>
</tr>
<tr>
<td>$H_{k1}: \mu_{pq}, \text{MIS, col} &gt; \mu_{pq}, \text{MIS, seq}$</td>
<td>$H_{l1}: \mu_{pq}, \text{RT, seq} &lt; \mu_{pq}, \text{RT, col}$</td>
<td></td>
</tr>
</tbody>
</table>
4.3. Subjects

The controlled experiment was carried out as a midterm test in a mandatory course on OO Analysis and Design focusing on UML. Each subject was randomly assigned to one of four groups, differing by diagram-type (collaboration vs. sequence) and application-type (MIS vs. RT system). The theoretical material and the practical examples have been taken from the course textbook [30]. Before participating in the experiment, which was carried out as a midterm test, the students have studied two types of interaction diagrams (sequence diagrams and collaboration diagrams), as well as other UML diagrams. The case studies presented in the class covered a variety of applications including real-time and management information systems. Building interaction diagrams given use cases and class diagrams was part of the course homework assignments.

The seventy-six subjects participating in the experiment constituted about 85% of the course enrollment. No subjects have been lost before both sessions of the experiment were completed.1 As shown below, this sample size enabled us reaching conclusions at the significance level of 0.05 and higher. Generally, we used a standard significance level of $\alpha = 0.05$ throughout this study (significant findings are marked in boldface).

4.4. Documents and questionnaire

In the first session of the experiment, which dealt with comprehension of diagrams, each subject has received a class diagram, a use case diagram, and an interaction diagram (collaboration or sequence) of one of the systems (see Figs 1-4). The questionnaires of the first session included five multiple-choice questions. The questions of for both applications are presented in Appendix A.

In the second session of the experiment, which dealt with quality of constructed diagrams, each subject has received the system requirements in the form of a brief narrative specification of the system (see description in Appendix B), a class diagram, and a use case diagram of one of the systems. This session assignment was to construct a sequence diagram or a collaboration diagram for the corresponding system. The test questionnaire of each group included the following parts

- **Test Overview.** This part described the test sessions, the assignments associated with each session, and the maximum number of scoring points allocated to each session. It also explained that the questionnaire of the second session will be provided only after the student has completed the first session and recorded the time of its completion.

1 Note that our sample has been significantly larger than the sample of 31 subjects used by Otero and Dolado [6].

### Table 2

<table>
<thead>
<tr>
<th>Group</th>
<th>Size</th>
<th>Session 1 (Comprehension)</th>
<th>Session 2 (Quality of Construction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>19</td>
<td>System = RT</td>
<td>System = MIS</td>
</tr>
<tr>
<td>G2</td>
<td>19</td>
<td>System = RT</td>
<td>System = Collaboration</td>
</tr>
<tr>
<td>G3</td>
<td>20</td>
<td>System = MIS</td>
<td>System = Sequence</td>
</tr>
<tr>
<td>G4</td>
<td>18</td>
<td>System = MIS</td>
<td>System = RT</td>
</tr>
</tbody>
</table>

- **Comprehension of Diagrams (Session 1).** As indicated above, the questionnaire of this part included five multiple-choice questions based on a class diagram, a use case diagram and an interaction diagram of the respective system.
- **Construction of Diagrams (Session 2).** Session 2 assignment included a brief description of one system, a class diagram and a use case diagram. The system in Session 2 was different from the one used in Session 1 by the same subject. The subjects were required to build a sequence diagram or a collaboration diagram for a certain use case (an opposite diagram type of the one analyzed by the same subject in Session 1—see Table 2). For the MIS, the use case was ‘Lend Item’. The use case of the RT system included turning the system on by a security person and then activating the burglar alarm sensors by an intruder. The diagram was required to include class names for each participating object, relationships between objects, and directed messages exchanged between objects.
- **Post-Test Questionnaire.** The subjects in each group have received their corresponding version of the post-test questionnaire, where they were asked to score the perceived comprehensibility and the perceived easiness of construction of interaction diagrams they have analyzed / built in their version of the test with respect to the systems they have analyzed in Session 1 and 2. The scores were on a 1–5 Likert ordinal scale.

4.5. Assignment of subjects to treatments

In this experiment we have manipulated two factors (diagram and system), each having two possible levels (types). Consequently, we have used a ‘$2 \times 2$’ factorial design. Each subject was assigned randomly to one of the resulting four groups (‘cells’) and then measured two times, first in the comprehension session and then in the construction (quality) session. Both the type of diagram and the designed system were different from one session to the other. In other words, each subject performed a task of comprehension using a certain type of diagram and a certain type of system; and later on he/she performed a task of constructing using another type of diagram and another type.
of system. The treatment conditions of each group are shown in Table 2 below.

The total score for each session was calculated as follows:

- Each correct answer to a multiple-choice question has received one point. No points were deducted for a wrong answer. Hence, the maximum total score for the first session was five points.
- The maximum total score for the second session was 60 points and it was calculated over all diagram components. The number of points deducted for each mistake in a diagram component was based on the severity of mistake, depending on the different components of the diagram. We identified the following components of interaction diagrams: objects, messages, message sequences, links, and time line. The number of points deducted for each mistake was 2, 4, 6, 8, or 10, depending on the severity of a particular mistake. Totally omitting an object was considered the most severe mistake resulting in a deduction of 10 points. Some mistakes were relevant only to one type of diagram. For example, omitting message number was relevant only to collaboration diagrams, while not showing activity duration was relevant only to sequence diagrams.

4.6. Threats to validity

There are a number of threats to the validity of studies like this (see [6,31]). These threats and their remedies are discussed below.

4.6.1. Internal validity

Internal validity is concerned about the cause-effect relationships induced from the study. Several potential threats are discussed below.

- **History effects.** If the environment changes during the study, the results may be different. This is not a threat to our study since all students have taken the test at the same time.
- **Testing effects.** All test questions were critically evaluated and revised in a pretest experiment, where the test has been solved by four professional system analysts with a strong background in UML. Prior to the test, all students have studied the same material in the same course (Object-Oriented Analysis and Design of Information Systems). Very few students had any kind of industrial experience in system analysis and design and their experience was extremely superficial. Potential cheating was prevented by breaking up the subjects into three random groups assigned to three supervised classrooms. Thus we expect the test results to be affected only by the personal capability of each student.

- **Instrumentation effects.** This threat was handled by giving all groups the same type of forms and evaluating their answers by the same scoring scheme.
- **Mortality effects.** No mortality effects were observed in the first two parts of the experiment, since all students participating in the experiment have filled out the questionnaires of both test sessions (Session 1 and Session 2). This can be easily explained by the students’ motivation to receive the highest possible grade for the test. However a few students have failed to fill out the Summary Questionnaire and the time spent on Session 2.
- **Maturity effects.** There is a threat that subjects become more enthusiastic about the study or, alternatively, become bored as the experiment progresses. Since the students were involved in the study for about two hours only, we do not consider it a major threat. The learning in the course of the experiment was not possible, since students were required to deal with a different type of diagram in each session. Moreover, the form of the first questionnaire (multiple-choice comprehension questions) was radically different from the form of the second questionnaire (a construction question).

4.6.2. External validity

External validity is about the possibility to generalize the results beyond the current experiment. Two threats to external validity are discussed below.

- **Selection of participants.** The students participating in the experiment were senior undergraduate students in Information Systems Engineering at Ben-Gurion University during 2003. We believe that it is possible to generalize our study to senior undergraduate students in Information Systems and related fields, since the training of these students was based on a standard curriculum, which was very similar to IS curricula in many US and European universities. Generalization of our results to IS professionals should be done with care though there are some indications that the differences between students and professionals in terms of judgment may be minor [31]. Using students as subjects in such experiments is very common for practical reasons: professionals are less available and much more expensive than students.

- **Nature of Problems.** The problems presented to the students in both sessions were non-trivial as they required the total time of about two hours to solve them. Both systems were of a similar level of complexity. The number of actors in the sequence diagrams was 1 in the library MIS and 2 in the security system. The number of interactions was 9 in the library MIS and 11 in the security system. Making the problems more time-consuming would stimulate a number of internal threats to the experiment validity such as mortality and maturity effects. We believe that each application (Library/Security system) was representative of typical MIS/RT
systems, respectively. However, we cannot generalize the results to all systems of each type.

5. Results

5.1. Analysis strategy

Figs. 6–7 show the histograms of students’ scores in Sessions 1 and 2, respectively. The actual scores received by every student in Sessions 1 and 2 of the test are given in Appendix C. No outliers were identified in either the scoring or the timing data. An observation would be considered an outlier if found far outside the $\pm 2\sigma$ range.

In order to evaluate the hypotheses stated in Table 1 with respect to the six dependent variables, we have used one-sided $t$-test, which is based on the following assumptions regarding the two samples:

- **Independence.** Based on our experimental settings (Section 4.5), there is no reason to believe that the results of the subjects in the first group (e.g. those doing sequence diagrams) are in any way related to the results of the subjects in the other group (e.g. those doing collaboration diagrams).
- **Normal distribution.** The scores of every group in each session were tested for normality using the chi-square test [32]. The results varied between $p=0.054$...
This means that no statistically significant departure from the normality assumption was found.

• **Equal variance.** We have compared the variances of every group in each session using F-test. Most groups were not found significantly different in terms of their variance except for the significant difference between the scores of Group 4 and Groups 1 and 2 in Session 1 with p-values of 0.009 and 0.003, respectively. As indicated by Minium et al. [32], the violation of the equal variance assumption is not problematic unless the two samples are quite different and one of the samples is small. In our study, we use nearly equal-size and relatively large groups (see Table 2 above).

The results of one-sided t-tests performed in the experiments are summarized in Tables 3–14 below. The two t-tests corresponding to each dependent variable were analyzed independently of each other (and not as a series of t-tests), since we are not interested in the overall differences between diagrams across the diverse application types. The minimum significance level for rejecting a null hypothesis was $\alpha = 0.05$. Detailed explanation and interpretation are provided in sub-sections 5.2-5.8 below.

### Table 3
#### T-Test of diagrams comprehension for MIS

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>4.250</td>
<td>0.966</td>
<td>0.103</td>
<td>0.459</td>
<td>0.156</td>
<td>0.143</td>
</tr>
<tr>
<td>Collaboration</td>
<td>4.222</td>
<td>0.646</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4
#### T-Test of diagrams comprehension for real-time system

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>2.315</td>
<td>1.249</td>
<td>2.345</td>
<td>0.012</td>
<td>1.000</td>
<td>3.227</td>
</tr>
<tr>
<td>Collaboration</td>
<td>3.315</td>
<td>1.376</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 5
#### T-test of diagram quality for MIS

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>46.473</td>
<td>5.337</td>
<td>2.043</td>
<td>0.024</td>
<td>1.000</td>
<td>2.813</td>
</tr>
<tr>
<td>Collaboration</td>
<td>50.842</td>
<td>7.639</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 6
#### T-test of diagram quality for real-time system

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>43.388</td>
<td>7.154</td>
<td>0.909</td>
<td>0.184</td>
<td>0.985</td>
<td>1.258</td>
</tr>
<tr>
<td>Collaboration</td>
<td>41.100</td>
<td>8.245</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 7
#### T-test of comprehension time for MIS

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean (minutes)</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>28.450</td>
<td>6.328</td>
<td>0.962</td>
<td>0.171</td>
<td>1.000</td>
<td>1.313</td>
</tr>
<tr>
<td>Collaboration</td>
<td>26.000</td>
<td>9.235</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 8
#### T-test of comprehension time for real-time system

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean (minutes)</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>22.473</td>
<td>7.890</td>
<td>0.199</td>
<td>0.421</td>
<td>0.447</td>
<td>0.274</td>
</tr>
<tr>
<td>Collaboration</td>
<td>23.052</td>
<td>9.907</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 9
#### T-test of construction time for MIS

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean (minutes)</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>56.812</td>
<td>19.613</td>
<td>0.101</td>
<td>0.518</td>
<td>1.000</td>
<td>1.584</td>
</tr>
<tr>
<td>Collaboration</td>
<td>65.128</td>
<td>26.157</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 10
#### T-test of construction time for real-time system

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean (minutes)</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>42.389</td>
<td>18.728</td>
<td>0.964</td>
<td>0.346</td>
<td>0.743</td>
<td>1.488</td>
</tr>
<tr>
<td>Collaboration</td>
<td>49.136</td>
<td>18.249</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 11
#### T-test of perceived comprehensibility for MIS

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>2.889</td>
<td>1.131</td>
<td>0.398</td>
<td>0.346</td>
<td>0.743</td>
<td>0.591</td>
</tr>
<tr>
<td>Collaboration</td>
<td>2.733</td>
<td>1.099</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 12
#### T-test of perceived comprehensibility for real-time system

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>t</th>
<th>P value</th>
<th>Power</th>
<th>Effect Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>2.778</td>
<td>1.308</td>
<td>0.538</td>
<td>0.296</td>
<td>0.655</td>
<td>0.750</td>
</tr>
<tr>
<td>Collaboration</td>
<td>3.000</td>
<td>1.201</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.5. Time spent on diagram construction

According to the results of one-sided t-tests evaluating the difference in the time (number of minutes) spent on diagrams construction for MIS and RT systems (shown in Tables 9 and 10 respectively), there is no statistically significant difference between the average time spent on each diagram type in either application. This also implies that there is no interaction between the application type and the diagram type with respect to time spent on diagram comprehension. Generally it should be noted that more time is spent on constructing collaboration diagrams in both system types. Irrespective of the diagram type, construction of an interaction diagram took more time in a management information system than in a real-time system.

5.6. Perceived comprehensibility

In the post-test questionnaire, each subject has expressed the perceived comprehensibility of the diagram he/she analyzed. Comprehensibility was expressed using a 1–5 Likert ordinal scale, where the score of 1 indicated that the diagram was very comprehensible, while the score of 5 indicated that the diagram was absolutely incomprehensible. The questionnaires were summarized with respect to the test versions done by the subjects, i.e. by diagram types and application types. According to the results of one-sided t-tests evaluating the difference in comprehensibility scores for MIS and RT systems (shown in Tables 11 and 12 respectively), there is no statistically significant difference between the scores of each diagram type in either application. Consequently, there is no interaction between the application type and the diagram type with respect to perceived comprehensibility. However, it should be noted that the subjects considered collaboration diagrams more comprehensible in the MIS application, while sequence diagrams were more comprehensible in the RT application.

The correlation coefficient between the comprehension score (Session 1) and the perceived comprehensibility score of the same subject is 0.207. This correlation coefficient is not significantly different from zero: p-value of the t test [32] is 0.086. In other words, no relationship was found between subjects’ perception and their actual performance in the test.

5.7. Perceived ease of construction

In the above post-test questionnaire, each subject has also estimated the perceived ease of constructing the diagram he/she has built in the test. The easiness score was given on the 1–5 scale, where the score of 1 indicated that the diagram was very easy to build, while the score of 5 meant that it was extremely difficult to build the diagram. The questionnaires were summarized with respect to the test versions done by the subjects, i.e. by diagram types and application types. According to the results of one-sided
t-tests evaluating the difference in easiness scores for MIS and RT systems (shown in Tables 13 and 14 respectively), there is no statistically significant difference between the scores of each diagram type in either application. This also means that there is no interaction between the application type and the diagram type with respect to perceived ease of diagram construction. However, it should be noted that the subjects considered collaboration diagrams easier to construct in the MIS application, while sequence diagrams were considered easier in the RT application. Interestingly enough, these results match the difference in actual quality of interaction diagrams constructed only in the case of the MIS (see Section 5.3 above). Disregarding the diagram type, the subjects considered construction of an interaction diagram in a real-time system to be easier than in a management information system.

The correlation coefficient between the construction score (Session 2) and the perceived ease of construction of the same subject is 0.054. This correlation coefficient is not significantly different from zero: p-value of the t-test [30] is 0.657. It means that, like in the case of comprehension, there is no relationship between subjects’ perception and their actual performance in the test.

6. Discussion

The means of scores and times for diagram comprehension and quality of construction are shown in Figs. 8 and 9 correspondingly. With respect to comprehension of diagrams, the results of this study indicate that collaboration diagrams are easier to comprehend in the case of RT applications, but there is no difference in comprehension of the two diagram types in the case of MIS. We have also shown that interaction diagrams of MIS are easier to comprehend than those of RT systems, irrespective of the diagram type. Our findings contradict the findings of Otero and Dolado [6], who found that sequence diagrams are easier to comprehend in modeling synchronous real time systems. The fact that comprehension scores obtained in our study for the MIS were significantly higher than those obtained for the RT system can be explained by the fact that our participants lacked adequate skills and training for comprehending real-time applications since students majoring in Information Systems Engineering are trained more in business-oriented applications but not as much in real-time and embedded systems. Since the MIS students subjectively perceived the real-time system to be more complex, and using a collaboration diagrams the real time system appeared more comprehendible than using sequence diagrams, we may conclude the following: while it makes no difference which diagram type to use in the case of a simple system, there is an advantage in using collaboration diagram, compared to sequence diagrams, in the case of a more complex system for the population of Information Systems students. The finding that subjects spend more time on comprehending MIS than RT system (irrespective of the diagram type) can only be explained by that the participants prefer spending time on a type of system which is more familiar to them and easier to comprehend (because they see a better chance to obtain a high test grade, while the amount of time spent on the task did not affect their test grade).

With respect to quality of diagrams constructed by analysts, we found that in the case of MIS, collaboration diagrams are significantly better than sequence diagrams, but there was no significant difference in the quality of diagrams produced for the RT system. As said, this can be explained because of the background of the subjects (who are more familiar with MIS than with RT systems) as well as that the MIS case study was less complex that the other. Although the findings here are not totally consistent with the finding on comprehension (as here there is no difference in
the case of real-time systems while before the indifference was in the case of MIS), still, collaboration diagrams are better than sequence diagrams. This result is indeed reasonable and in accord with the claims about the preference of collaboration diagrams for information systems. However, the opposite claim about preference of sequence diagram for RT system is not supported by our study. But, again, this may be explained due to the subjects who may not be the best candidates to model RT systems.

Our findings were not able to confirm a significant difference between the diagrams with regards to the time needed to construct them. But as we have seen, analysts spend more time on constructing interaction diagrams of an MIS than of a RT system. This result is consistent with the equivalent result on time spent on comprehending diagrams. As explained for the equivalent results, a possible reason for this may be that participants prefer spending time on a type of system which they are more familiar with (because they see a better chance to obtain a high test grade on quality, as the amount of time spent on the task did not affect their test grade).

With respect to the perceived comprehensibility, the results we obtained are congruent with the expectations based on the literature reviewed above [7,13–15]. For the RT application, sequence diagrams yielded a better score than collaboration diagrams, and for an MIS, collaboration diagrams yielded a better score than sequence diagrams. Nevertheless, the differences were not significant and no interaction was found between the application type and diagram type. Surprisingly, the results of perceived ease of construction were not congruent with our finding on comprehensibility; that is, collaboration diagrams yielded a better score for a RT application and sequence diagrams yielded a better score for MIS. Again, the differences were not statistically significant and no interaction was found. Interestingly, the perceived ease of construction for real-time applications in general was better than for MIS applications. In practice, however, the scores obtained by participants in constructing the MIS were better than for the RT system. This contradiction could be explained by the fact that participants were not very familiar with the RT application domain, and therefore tended to underestimate the difficulty of modeling interaction in RT applications.

7. Conclusions

The implications of our study are in further investigating the inherent contingency of UML interaction diagrams in terms of both quality and comprehension for various application settings. As most extant IS development methodologies are designed for situations that fit some ideal specifications, whether stated or unstated. Situations, however, might differ depending on the type of project and its objectives, the organization’s settings and its environment, the users, as well as the developers and their respective skills [33]. In an attempt to meet the challenges posed by such a complexity, a contingency methodology is recommended in order to promote different modeling approaches and artifacts depending on a specific situation (i.e. Multiview2, [34]).

This study adopted the idea of modeling contingency as described above. Consequently, we evaluated the practical feasibility of different interaction diagrams for various application types. The results from the controlled experiments suggests a rule of thumb for employing collaboration and sequence diagrams in modeling RT and MIS applications for both analysis and design stages. In some cases, however, our results contradict earlier studies [1,6].

Our study suffers, however, from several limitations. First, the applications tested in the study (a library system and a security monitoring system) cannot be considered equivalent to full-scale commercial applications. Second, the fact
that only two applications were evaluated limits the external validity of this study. Third, the problems had a very slight difference in complexity; however for a given problem (MIS, RT) this should not have an effect on the findings with regards to the recommended interaction diagram.

A common limitation of experimental research on model/method evaluation refers to the students participating, who in our case played the role of users (in the ‘comprehension of diagrams’ part of the experiment) and of analysts (in the ‘quality of diagram construction’ part). Since the students lack sufficient experience in analysis and design work, especially in the RT domain, it is most likely that our results suffer from a bias in favor of the MIS application. Moreover, the fact the students are not as experienced as IT practitioners limits the external validity of our study and our conclusions cannot be generalized to the industry.

Future research on this matter should first attempt to address some of the limitations mentioned above. Thus, a repeat experiment should use more experienced participants, possibly with professionals from the IT industry working on RT and MIS projects. The bias in favor of the MIS applications could be remedied by using a mixed population of participants, including students of Software Engineering, who are more trained on RT systems, in addition to the Information Systems Engineering students. It is also recommended to diversify and control the complexity of the applications in the experiment, as performed by Otero and Dolado [6].

Regarding the diagram construction session, the quality of the produced diagrams could be further evaluated by developing two versions of the same system: one based on a collaboration diagram and the other based on a sequence diagram. Such an evaluation was not possible in an academic setting. However, if performed, would significantly add to the validity of our findings.

Other important issues for future research are: evaluating the feasibility of interaction diagrams for modeling other types of applications, i.e. workflow and customer relationship management (CRM) applications, which incorporate a substantial amount of interaction with their clients, and were not studied with regards to modeling their dynamic aspect.

Appendix A. Diagram comprehension questionnaires

A.1. Library MIS

The system provides loaning services for items (journal and book titles). Specifically, it handles the following tasks: issuing and canceling a title reservation, loaning an item, returning and item and maintenance (adding/removing items from catalogue, updating borrower’s status, and adding new titles).

Based on the provided use case, class and sequence/collaboration diagrams, answer the following questions:

1. In order to borrow a book or journal item, the following pre-condition must hold:
   a. An order exists in the system
   b. No order exists; however, the title and item are in the library
   c. The title and item are both in the library, whether an order exists or not
   d. None of the above answers is true; the correct answer cannot be determined from the diagrams provided

2. According to the diagrams, what are the possible reasons for an order being deleted?
   a. The 30-day limit has expired and deletion is automatic
   b. The ordered book or journal is on loan
   c. Both A and B are correct
   d. None of the above

3. What is the possible number of orders for a book or journal?
   a. Maximum 10
   b. 1–10
   c. Not limited
   d. None of the above

4. Whenever an order is deleted, the following object is erased:
   a. Title
   b. Item
   c. Both title and item
   d. None of the above

5. Whenever a client, either human or a different library, wishes to borrow a book or journal which is still on loan, the following happens?
   a. The system checks for additional available items of the requested title
   b. The system places an order for the requested book or magazine
   c. An extant order for the book or journal is deleted
   d. None of the above answers is true; the correct answer cannot be determined from the diagrams provided

A.2. Security monitoring system

The system monitors a set of detectors, providing alerts in the event of a fire or a breach.

Based on the provided use case, class diagram and sequence/collaboration diagram, answer the following questions:

1. How is it possible to verify that the system is active?
   a. According to the control box
   b. The siren beeps

2 Use case and class diagrams are not shown due to space limitations and are available upon request.
Appendix B. Diagram construction tasks

B.1. Security monitoring system

The security monitoring system controls a set of detectors capable of alerting a security guard in the case of a fire or a breach. In the event one of the detectors is activated, the siren is turned on. The monitoring system is activated and deactivated using a key.

The system’s hardware comprises of the following components: control box, siren, fire and breach detectors. The security guard activates the system and a beep sound signals that the system has been successfully activated. The control box monitors the fire and breach detectors. If a fire is detected the siren is turned on for 15 min and in case a breach is detected the siren is turned on for 30 min.

Based on the following use case and class diagrams (excluding methods), you are requested to construct a collaboration/sequence diagram for the objects participating in the ‘activate system’ scenario. You should specify a scenario where the security guard activates the system and a burglar activates the detector. The diagram should include:

- class names of the participating objects,
- relationships among objects and
- directed messages exchanges among the objects during the scenario.

B.2. Library MIS

The Library system provides loaning services for bibliographic items (journal and book titles). Specifically, it handles the following tasks: issuing and canceling a title reservation, loaning an item, returning and item and maintenance (adding/removing items from catalogue, updating borrower’s status, and adding new titles)

To issue an order, the identification data of the borrower and the journal or book item need to be provided. When loaning a title, the librarian first locates the reserved title and deletes the reservation. Next, the item’s status is updates and handed to the borrower. Returning an item is performed by locating the item in the catalogue and updating its status.

Based on the following use case and class diagrams (excluding methods), you are requested to construct a collaboration/sequence diagram for the objects participating in the ‘borrow title’ scenario. The diagram should include:

- class names of the participating objects,
- relationships among objects and
- directed messages exchanges among the objects during the scenario.

Appendix C. Selected raw data

Tables C1 and C2.

<table>
<thead>
<tr>
<th>Observation No.</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

Average: 2.316 3.316 4.250 4.222
Std. Dev: 1.250 1.376 0.967 0.647
Table C2
Scores of Session 2 - Diagrams Quality

<table>
<thead>
<tr>
<th>Observation No.</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>54</td>
<td>40</td>
<td>54</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>54</td>
<td>48</td>
<td>46</td>
<td>44</td>
</tr>
<tr>
<td>3</td>
<td>42</td>
<td>54</td>
<td>28</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>56</td>
<td>44</td>
<td>34</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>48</td>
<td>50</td>
<td>44</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>48</td>
<td>54</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>7</td>
<td>34</td>
<td>50</td>
<td>40</td>
<td>46</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>42</td>
<td>28</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>38</td>
<td>38</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>10</td>
<td>42</td>
<td>38</td>
<td>44</td>
<td>42</td>
</tr>
<tr>
<td>11</td>
<td>50</td>
<td>46</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>12</td>
<td>56</td>
<td>40</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>13</td>
<td>60</td>
<td>54</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>14</td>
<td>48</td>
<td>46</td>
<td>40</td>
<td>48</td>
</tr>
<tr>
<td>15</td>
<td>62</td>
<td>46</td>
<td>46</td>
<td>40</td>
</tr>
<tr>
<td>16</td>
<td>52</td>
<td>54</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>17</td>
<td>60</td>
<td>48</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>18</td>
<td>58</td>
<td>46</td>
<td>44</td>
<td>48</td>
</tr>
<tr>
<td>19</td>
<td>52</td>
<td>45</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>46</td>
</tr>
<tr>
<td>Average</td>
<td>50.842</td>
<td>46.474</td>
<td>41.100</td>
<td>43.389</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>7.640</td>
<td>5.337</td>
<td>8.246</td>
<td>7.155</td>
</tr>
</tbody>
</table>

References