Distributed Modeling of Use Case Diagrams with a Method Based on Think-Pair-Square: Results from Two Controlled Experiments

Giuseppe Scanniello\textsuperscript{a}, Ugo Erra\textsuperscript{a}

\textsuperscript{a}Dipartimento di Matematica, Informatica e Economia, University of Basilicata, Viale dell’Ateneo, I-85100, Italy
giuseppe.scanniello@unibas.it; ugoerra@unibas.it

Abstract

Objective. In this paper, we present the results of two controlled experiments conducted to assess a new method based on think-pair-square in the distributed modeling of use case diagrams.

Methods. This new method has been implemented within an integrated environment, which allows distributed synchronous modeling and communication among team members. To study the effect of the participants’ familiarity with the method and the integrated environment, the second experiment is a replication conducted with the same participants as the original experiment. The results show a significant difference in favor of face-to-face (i.e., the chosen baseline) for the time to complete modeling tasks, with no significant impact on the quality of the produced models.

Results. The results on participants’ familiarity indicate a significant effect on the task completion time (i.e., more familiar participants spent less time), with no significant impact on quality.

Practice. One of the most interesting practical implications of our study is - in case the time difference is not an issue, but moving people might be a problem, the new method and environment could represent a viable alternative to face-to-face. Another significant result is that also people not perfectly trained on our method and environment may benefit from their use: the training phase could be shortened or skipped. In addition, face-to-face is less prone to consolidate participants’ working style and to develop a shared working habit of participants.

Implications. This work is in the direction of the media-effect theories applied to requirements engineering. The results indicate that the participants...
in the experiments significantly spent less time when modeling use case diagrams using face-to-face. Conversely, no significant difference was observed on the quality of the artifacts produced by the participants in these tasks.

**Keywords:** Experiments, Distributed Software Development, Requirements Modeling

1. Introduction

Splitting the development of a software product (e.g., system or service) among globally distributed sites is increasingly becoming a common practice in the software industry [1, 2]. Academy and industry communities refer to this phenomenon as global, distributed, or multi-site software development [3, 4]. In the recent years, an increasing interest has been manifested on the global software development because it offers several benefits for software organizations, such as working cost reduction, enhanced availability of skilled development staff, proximity to the market, and flexibility and efficiency for in-house staff usage to adapt quickly to volatile business needs [3]. One of the drawbacks of this software development practice is that very often it creates challenges due to the impact of temporal, geographical, and socio-cultural differences of stakeholders [1, 5].

Both in traditional and in global software development, requirements engineering aims at discovering and documenting features that a subject system must implement (i.e., functional requirements) and constraints that it must satisfy to be accepted by the customer (i.e., non-functional requirements). One of the most challenging phases of the requirements engineering process is requirements elicitation in which stakeholders (e.g., requirements analysts, customers, and end-users) have to cooperate and communicate to discover and to document requirements (e.g., models abstracting functional requirements) [6, 7]. Depending on the software project, the number, the kinds, and the roles of stakeholders might variate. For example, the user and the requirements analysis roles could coincide for in-house software projects, where the system is developed inside the software company for its own use [8].

In global software engineering, requirements engineering is greatly affected by the geographical distribution of different stakeholders [9]. Despite some advances in collaboration methods and technologies, the greater part of available distributed methods and approaches adapts practices based on face-to-face communication that are well established in traditional software
engineering field (e.g., [10, 11, 12, 13]). On the basis on recent findings [13, 14], it seems promising to rethink interaction methods among stakeholders taking advantage the number of possibilities new communication medium offer. Independently from the kind of method/approach, to determine whether it is effective and successful, we should look at whether it solves the problem for which that method/approach has been conceived [15]. Clear evidences of the advantages of a new method/approach should be experimentally demonstrated, so speeding up its acceptance in industry [16]. In other words, promising experimental results could speed up the technology-transfer process from research laboratories to the software industry [15].

Although there are a number of empirical studies on software engineering (e.g., [17, 18]), few evaluations on global software engineering have been conducted as the results of the Systematic Literature Review (SLR) by Smite et al. [2] show. This lack is even more evident in the requirements engineering process: only 10% of the empirical investigations included in that SLR are concerned with requirements engineering.

In this paper, we empirically assess a new method [19] in the context of global software development to document functional requirements with use case diagrams [20]. This method has been implemented in an integrated environment, which allows geographically distributed stakeholders (e.g., requirements analysts) to synchronously communicate and model requirements with use case diagrams. We have conducted a controlled experiment and a replication with the same participants as in the original experiment to study the effect of participants familiarity with that new method and its integrated supporting environment. In these experiments, we have compared that new method with face-to-face interaction that is considered the baseline for experiments similar to ours (e.g., [9, 10, 11, 12]). This comparison is based on the time to complete modeling tasks and on the quality of the models produced in these tasks. In the replication, we have introduced some variations (e.g., new experiment objects) to verify whether better trained participants benefit more in case they are more familiar with method and integrated environment.

The work presented in this paper is built on that presented in [19]. The main new contributions here can be summarized as follows:

- The method is described in more detail.
- The integrated environment is presented here for the first time.
• The method to quantitatively assess the quality of use case diagrams is fully described, providing also an application example.

• The data analysis is extended and the discussion of the achieved results has been improved.

• The implications of our study from the practitioner and the researcher perspectives are presented and discussed here for the first time.

• A deeper and more comprehensive discussion of related work is given.

Paper Structure. The remainder of the paper is organized as follows. In Section 2, we summarize our method and its supporting synchronous computer-mediated communication and modeling environment. The study definition and details on the adopted experiment design are presented in Section 3. The results of the data analysis are presented in Section 4, while we discuss these results and their practical implications in Section 5. In this section, we also discuss possible threats that may affect the validity of our study. In Section 6, we discuss related work. Final remarks and future direction for our research conclude the paper.

2. The Proposed Method

Psychologists in learning settings are successfully applying computer-mediated communication (see the LEAD 1 european project, for example) because its use better takes into account social, cognitive, and developmental processes of learners. On the basis of these considerations and taking into account that requirements engineering is about cooperative learning of a problem domain and it is about the discovering and the documentation of requirements, it could make sense to investigate the possibility of using new collaborative learning methods based on computer-mediated communication. This is somewhat supported from results of recent empirical studies (e.g., [9, 14]) in the field of distributed requirements engineering. Among the methods for collaborative learning, think-pair-square [21, 22] seems to be appropriate for the problem at hand because conceived for promoting active learning and discussions and for solving problems in cooperative fashion. We preliminarily applied this method in the requirements engineering field [14].

1http://lead2learning.org/
(see Section 6) and until now it has not been applied in fields different from cooperative learning and computer-mediated communication.

In the following subsections, we describe both the method and the supporting synchronous computer-mediated communication and modeling environment. In the following, we will also refer to this environment as communication/modeling environment or simply environment or integrated environment.

2.1. Background

2.1.1. Use Case Modeling

Requirements engineering aims at discovering and documenting the features that a subject system must implement (i.e., functional requirements) and the constraints that it must satisfy to be accepted by the customer (i.e., non-functional requirements). The most challenging and demanding phase is requirements elicitation [7]. It is in charge of collecting and representing requirements from the perspective of users and/or customers [23]. The main goal consists in documenting requirements, so that they can be understood by both customers and users. Requirement elicitation is a non-trivial task because you can never be sure you get all requirements from the user and customer by just asking them what the system should do. Communication, cooperation, functional and non-functional requirements modeling, and problem-solving are the most demanding activities of this phase.

Functional requirements describe the interactions among the system, the environment, and the users. Their specification is independent from the implementation. Use case modeling is a valuable means to specify functional requirements [24]. Use case modeling is widely used in the software industry [25] and includes the following tasks: identifying and specifying use cases and actors, recognizing relationships between model elements, and refining use cases [7]. These tasks become even more difficult in case stakeholders are geographically distributed.

As far as the specification of a use case, several templates can be exploited. Some of these template are very simple and suggest to specify uses cases as plain text, while others are more formal [26]. Whatever is the template, use cases might be classified in: summary (representing the goal of the system), user-goal (representing the system functionalities from the user perspective) and sub-function (used to avoid describing the same use case twice) [27]. A summary use case might include user-goal use cases, which in turns might include sub-function use cases. Each of these kinds of use case highlights
different aspects with respect to functional requirements. For example, user-goal use cases are better focussed on the functionality from the end-user perspective. Therefore, if a use case diagram includes only user-goal use cases, it can be used to describe all the functionalities of a system from the end-user perspective. Use case diagrams are also valuable because they allow stakeholders to easily represent and understand relationships among use cases and between use cases and actors. Use case diagrams that include only user-goal use cases describe high level functionality of a system in a horizontal way from the end-user perspective rather than represent the goal of the system, sub-functions, or details of an individual functionality.

2.1.2. Think-Pair-Square

Think-pair-square has been conceived for promoting active learning and discussions and for solving problems in cooperative fashion among individuals. In particular, it gives individuals the opportunity to discuss ideas and possible solutions for a given problem and provides these individuals with means to observe and refine strategies of the other individuals involved. Individuals are grouped in homogeneous or heterogeneous way and are asked to solve a given problem through the following sequential three steps/phases:

**think** - It is individually accomplished to approach a solution for a given problem. The prompt should be limited so that each individual can really focus on the start point. This allows for wait time and helps each individual to control the urge to impulsively shout out the first answer that comes to mind.

**pair** - It is performed in pair. Individuals possibly use the solutions individually identified in the think step to get a common solution for a given problem. In this step, individuals may wish to revise or alter their original ideas.

**square** - It is performed by a team composed of all the individuals, who work on the steps think and pair. Individuals work on the basis of the solutions identified in the pair step, so forming a square and discussing on these solutions. This allows individuals to converge towards the same and more comprehensive solution. Consensus can be the goal, but engaging together should be the objective of this step.

Each individual is equally important in the think-pair-square method. Individuals perform the pair and square phases in the same physical setting.
through face-to-face interaction. The rationale for performing these steps relies on the fact that if one pair of individuals is unable to solve the problem, the other pairs can often explain their answer and strategy. Finally, if a suitable solution for a problem is not identified, pairs combine their results.

With respect to other active learning methods, the think-pair-square method is considered relatively low-risk and ideally suited for individuals who have some experiences in collaborative groups [21]. This method is particularly suitable to let individuals use higher level thinking and justify their reasoning and solutions. These are some of the main motivations to have focused our research on that method. There are also some slightly variations of that method [21]. For example, the think-pair-share method replaces the square step with the share step. Individuals are called upon to share with the rest of the group possible solutions of the problem obtained in the pair step. Differently from the think-pair-square method, these solutions are only show to all the individuals, namely no discussion among individuals is executed to identify a common solution to a given problem. That is, all the individuals will have knowledge about all the possible solutions emerged in the first two steps of the method. Recently, some technological variations have been also proposed, so showing the high level of flexibility of the methods think-pair-square and think-pair-share. An example is the think-tweet-share method. Individuals think of response and then generate a 140 character representation of a Tweet. In the think-ink-pair-square method, students are actively engaged through writing (e.g., notecards), but without limits in the characters to be used.

2.2. Distributed Think-Pair-Square and Supporting Tool

Based on our experience [14] and on the fact that requirements modeling concerns active discussions to cooperatively discover and document requirements, it seemed reasonable to reformulate think-pair-square for synchronous computer-mediated communication and modeling in distributed requirements engineering. Similar to think-pair-square, the new method is composed of three subsequent steps:

**think** - It is individually accomplished. Each team member works to produce one or more use case diagrams.

**pair** - It is performed in pair. Team members work in pairs on the models that they have individually identified.
square - All the team members together exploit the model of each pair to produce a unique common and more comprehensive model. Teams can be composed of any even number greater or equal to four.

With respect to the original definition of the think-pair-square method, we have replaced face-to-face interaction with synchronous computer-mediated communication: a text-based chat and a synchronous distributed modeling environment. Even if we took into consideration here an editor for use case diagrams, the new method could easily modified using any kind of UML distributed modeling tool. Whatever is the modeling tool, it allows the creation of software models, but also enables implicit communication among team members while performing modeling tasks. The considered new method can be used by different stakeholders with an adequate modeling experience. The end-user or the customer could also participate in the square step to clarify possible concerns.

A prototype of a supporting system has been implemented using the CoFFEE framework [28]. Although this framework is mainly aimed at implementing environments to improve cooperative learning in the context of computer support work-groups, it can be also employed to develop tools for supporting synchronous computer-mediated tasks. CoFFEE provides three main means: (i) Session Editor, (ii) Controller, and (iii) Discusser. In the following subsections, we provide details on how we used these means to implement the studied method.

2.2.1. The Session Editor Tool

The Session Editor tool is used to define a communication/discussion environment by means of a session that is in turn composed of steps. The activities that can be accomplished within each step are defined combining one or more basic tools. We have used here two tools: the Chat Tool (CT) and the Shared Drawing Tool (SDT). The Chat Tool is a traditional synchronous text-based chat, while SDT is a synchronous environment for the modeling of UML diagrams [20]. The environment used here is that for drawing use case diagrams.

As shown in the tab Session Details of Figure [1], the session underlying the method is composed of three steps (one for each phase/step of the think-

---

2It is a knowledge transfer process based on communication through a shared mental or abstract model.
pair-square method). In all the steps, the tool SDT is used to draw use case diagrams and to show them to the next step (i.e., pair and square). Differently, the CT tool is included in the second and third steps to enable the communication among team members. Each team member employs CT to communicate and discuss issues with other members of the team. Our integrated environment can be easily customized to meet the team and the project needs. For example, the number of the steps could be reduced (the think step could be skipped, for example).

Figure 1 shows (see the tab Step Details) the tools used in the Pair step. The CoFFEE Session Editor also provides a layout preview to highlight how the tools will be shown to users (see on the bottom right hand side of Figure 1). At each step the model created by using SDT is made available to the subsequent one.

2.2.2. The Controller Tool

The Controller tool allows executing sessions previously defined with the Session Editor tool. A coordinator has first to compose a team and then
he/she can start the session. The coordinator can also constantly monitor the interactions among team members. The session behind the new method and how it is shown within the Controller tool is depicted in Figure 2. On the top left hand side of this figure, it is shown the tab Control Panel that lets the coordinator associate each member (e.g., RUSSO) to a team (e.g., 1) within the Think step (see on the middle of the tab SessionControl). In this case, each team is assigned to a different group since the Think step has to be individually accomplished. The team composition has to be performed also for the subsequent steps of the method. For the Pair step, the teams will be composed of two members. To run a session the button on the top of the tab SessionControl has to be clicked.

2.2.3. The Discusser Tool

The Discusser tool allows team members to exploit the environment created by the Session Editor tool and executed by the Controller tool. In particular, the communication follows the session underlying the communication environment and is supported by the tools that compose this session. Figure 3 and Figure 4 show our communication/modeling environment in the Think and Square steps of our method, respectively. Regarding the Think step, the discusser (see Figure 3) only provides features to draw a use case diagram (i.e., the SDT tool). Differently, our environment provides both the
tools CT and SDT in the Pair and Square steps (see Figure 4). The diagram modeled in the previous accomplished step is made available (see on the left hand side of Figure 4) in the SDT tool. In the Square step, team members can browse modeled use case diagrams and chat messages of the Pair step (see the tabs on the top of Figure 4).

It is worth mentioning that from the practical perspective the access to chat messages of other stakeholders could lead to self-censoring in the case in which there are personal or goal conflicts among stakeholders. In fact, stakeholders may refrain from expressing their true opinions on conflictive matters via the chat if there is a risk that later on other stakeholders can read their conversation. In the current implementation of the integrated environment, discussions cannot be kept private. To deal with self-censoring issues, we plan to extend our environment to specify if a discussion has to be public or private.

3. The Controlled Experiments

In this section, we present the original experiment (Experiment I in the rest of the paper) and its replication (i.e., Experiment II). This replication is...
a differentiated replication [29] because we introduced a variation in the conditions of the original experiment, namely the used experiment objects. The experiments were carried out by following the recommendations provided by Juristo and Moreno [30], Kitchenham et al. [31], and Wholin et al. [32]. The experiments were reported according to the guidelines suggested by Wholin et al. [32]. For replication purposes, the experimental package and the raw data of both the experiments are available for downloading on the web.

3.1. Definition

The main goal of the experiments is to investigate whether the new method and its implementation in our integrated communication/modeling environment are effective as traditional face-to-face in the requirements elicitation phase when modeling high level functionality of a software system based on use case diagrams that include only user-goal use cases (see Section 2.1.1) or simply use case diagrams from here on. In the following, we will refer to the new approach and our integrated environment for synchronous computer-mediated communication and modeling as TPS. On the
other hand, F2F is used to indicate face-to-face interaction.

The goal of our experimentation can be defined by using the GQM (Goal Question Metric) [33] template as follows: “Analyze distributed requirements engineering for the purpose of comparing the use of TPS and F2F with respect to the time to specify high level functionality and the quality of the produced models from the point of view of the researcher in the context of global software engineering and students in Computer Science, and from the point of view of the requirements analyst in the context of young software engineers who are geographically distributed”.

3.2. Context

The context of the experiments was constituted of Bachelor and Master students in Computer Science at the University of Basilicata. 27 were students of a Software Engineering (SE) course of the Bachelor program, while 9 were students of a Computer Graphic (CG) course of the Master program. Before conducting the experiments, the students were asked to fill in a pre-questionnaire to get some information on their industrial working experience and their GPA [4] (Grade Point Average). The results of this questionnaire were used to equally distribute high and low ability participants among the teams (see Section 3.4). Students with a GPA less than 25 are considered to be low ability participants, otherwise high [34, 35].

One of the main topics of SE course is modeling of object-oriented systems using the UML. As mandatory laboratory activity of that course, the students attended lectures and designed and developed an object oriented software system adopting an incremental process. Moreover, they were grouped in teams of seven/eight members each. Each team performed a complete requirements elicitation and analysis and high level design of a software system using the UML. Each subsystem was in turn incrementally developed in Java. In the requirements elicitation phase, the students were asked to model functional requirements based on summary level, user-goal, and sub-function use cases. The students of the SE course were asked to accomplish the controlled experiments as a part of a series of optional laboratory exercises of that course. Before these experiments, the students had already taken exams for the following courses: Web Technology (WT), Object Oriented Program-

---

4In Italy, the exam grades are expressed as integers and assume values between 18 and 30. The lowest grade is 18, while the highest is 30.
ming (OOP), and Data Base (DB). Some of these courses already introduced students to modeling and the UML (OOP and DB).

The Master students have previously passed the SE course in their Bachelor program. The lecturer of that course was the same as the SE course mentioned above. Therefore, the students designed and developed a software system as the Bachelor students did. The participants had also passed exams for the following courses: WT, OOP, Advanced Object Oriented Programming (AOOP), and DB. As laboratory activity of the CG course, the students were asked to develop either video games (e.g., doom-like game) or three dimensional environments for software and scientific visualization. The students worked individually or in most cases in small teams composed of two or three members. The controlled experiments presented in this paper represented an optional laboratory exercise for the students of the CG course. It is worth mentioning that the Master students had some working experience because they joined a company for an industrial internship at the end of their Bachelor degree program. This internship lasted in between 3 to 6 months.

All the students (i.e., Bachelor and Master students) were volunteers and were aware of the practical and pedagogical purposes of the experiments, but they did not know the experimental hypotheses. These students were asked to perform the assigned tasks in a professional way. In addition, they were not graded on the results they obtained in the experiments and they have never used TPS before Experiment I.

3.3. Hypotheses Formulation

The classification proposed by Robert [36] considers the communication media in this investigation both synchronous (i.e., communication happens in real time), but with different space dimension. Accordingly, we can assert that the participants in the experiments are virtually and physically collocated when interact to accomplish a given task, when using TPS and F2F, respectively. Then, from the communication media theory perspective, we are interested in investigating whether virtually collocated participants may obtain similar performances (i.e., the time and quality of the software artifacts produced within modeling tasks) than physically collocated ones. Therefore, we first study the effect of TPS and F2F on the time needed to model use case diagrams starting from a specification document that describes the features a system should support and the environment in which it should be deployed. Second, we verify whether the use of TPS and F2F
influences the quality of the produced diagrams. In the experiments, we test the following two null hypotheses:

**Hn1**  The time to model a use case diagram is not significantly greater when using TPS (with respect to F2F).

**Hn2**  There is not a significant difference in the quality of a use case diagram when using F2F or TPS.

The fist null hypothesis is one sided because we can postulate that participants should spend less time to accomplish tasks using F2F. This is why a chat tool and a shared editor needs more time to reconcile the different perspectives of participants compared to face-to-face. The second null hypothesis is two sided because we cannot do any postulation on the quality of the use case diagrams produced. The alternative hypothesis for Hn1 is Ha1, while the alternative hypothesis for Hn2 is Ha2.

We are also interested in assessing whether the familiarity of the participants with the new method and the integrated supporting environment might affect the time and the quality of the use case diagrams produced in the modeling tasks. In particular, we investigate whether:

(i) better trained participants spend less time to accomplish a task and (ii) the quality of use case diagrams does not depend on the familiarity with method and environment. To this end, the following two further null hypotheses are tested:

**Hn3**  A higher level of familiarity with TPS does not significantly reduce the time to model use case diagrams.

**Hn4**  There is not a significant difference of the use case diagram quality if participants are more familiar with TPS.

Hn3 is one sided because we expect that better trained participants spend less time to accomplish a task. Hn4 is two sided because we cannot do any postulation on the quality of the use case diagrams produced. Indeed, the quality of a use case diagram should be independent from the familiarity with TPS. In case the defined null hypotheses can be rejected, it is possible to accept the alternative ones (i.e., Ha3 and Ha4 for Hn3 and Hn4, respectively).
3.4. Design

The experiments were performed under controlled conditions in a laboratory at the University of Basilicata according to the design reported in Table 1. This design ensured that each team worked on two different experiment objects (i.e., Object1 and Object2) in two laboratory runs (i.e., Run1 and Run2). The teams were grouped into the groups A and B. For example, the teams in the group A accomplished first the task with TPS on Object1 and then on Object2 with F2F. We opted for this design to mitigate as much as possible learning/fatigue effects and the effect of the experiment objects on the results.

For Experiment I, Object1 is Library (a software system to manage books and users of a library), while Object2 is FilmCollection (a software system for the selling and the rental of films within a shop). The experiment objects were different in Experiment II. In particular, Object1 and Object2 were Rent (a car rental software to manage available cars, customers, and reservations) and ECP (an E-Commerce Platform to order CDs and books via the Internet from an online catalogue), respectively. The software systems of Experiment I and Experiment II refer to application domains on which the participants were or should be familiar with, so reducing as much as possible guessing effect and mitigating external validity threats. Due to the familiarity of the participants with the application domain and to their experience in modeling, we can consider Bachelor students as novice participants and Master students as less novice participants.

The specification documents of the four experiment objects can be considered as developed in small-sized development projects of the following kinds: in-house software (the system is developed inside the software company for its own use), product (a system is developed and marketed by a company, and sold to other companies), or contract (a supplier company develops and delivers a system to a customer company) [8]. In all these kinds of projects, the client role is not present or coincide with that of the software engineer (e.g., developer or requirements analyst). The used specification documents are available in Italian and English in the experimental package. Note also that the size of the text within these document was nearly the same for all the systems (about 300 words and 2100 chars). The specification document of ECP also included some screen mockups [7] (i.e., 6). This difference was deliberately introduced and properly analyzed (see Section 5.1).

According to the results of the pre-questionnaire, we equally distributed high and low ability participants in 9 teams (the interested reader can find
demographic information of the participants in the experimental package). Each team contained 3 Bachelor students and 1 Master student. This design choice allowed us to have teams heterogeneously composed in terms of modeling experience with the UML. Such a kind of difference is also presented in industry. The composed teams were successively randomly assigned to the groups A and B. In particular, 5 and 4 teams were assigned to the groups A and B, respectively. The rationale for choosing 4 as size of each team relies on the fact that this size allowed us to get the highest number of observations (4 represents the smallest size as possible to instantiate our method). The analysis of the effect of the team size could represent a possible future direction for our research.

The groups and the teams in Experiment II were the same as in Experiment I. This design choice allowed us to assume that the participants when passing from Experiment I to Experiment II increased their familiarity with TPS. Furthermore, each participant consolidated his/her working style and developed/improved a shared working habit with the other participants in his/her team [37].

<table>
<thead>
<tr>
<th>Groups</th>
<th>Run1</th>
<th>Run2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Object1, TPS</td>
<td>Object2, F2F</td>
</tr>
<tr>
<td>B</td>
<td>Object1, F2F</td>
<td>Object2, TPS</td>
</tr>
</tbody>
</table>

3.5. Selected Variables

The main factor of the study is Method (the variable we can manipulate). It is a nominal variable with two possible values: F2F, TPS. Two additional variables (also named factors or co-factors) have been considered and controlled in each experiment: Run (∈ {Run1, Run2}) and Object (∈ {Object1, Object2}). To analyze the effect of the participants’ familiarity with TPS, we also analyzed the variable Experiment (∈ {ExperimentI, ExperimentII}).

According to the defined null hypotheses, we selected and used the following dependent variables:

**Time** indicates the number of minutes that all the participants within a given team spent;

**Quality** indicates the quality of the use case diagram produced.
The number of minutes to complete the tasks with TPS was directly recorded by our integrated environment. On the other hand, we asked the participants to note down the start and stop time of the task, when F2F was used. We monitored this process to reduce possible biases.

We used an approach based on the information retrieval theory [38] to get a quantitative assessment of the quality of use case diagrams modeled with both TPS and F2F. Use case diagrams are compared with an oracle (i.e., the correct use case diagram) in terms of actors/use cases (a/u) and dependencies (d) between them:

\[
\text{precision}_{a/u} = \frac{|A_{a/u} \cap O_{a/u}|}{|A_{a/u}|}, \quad \text{recall}_{a/u} = \frac{|A_{a/u} \cap O_{a/u}|}{|O_{a/u}|},
\]

\[
\text{precision}_{d} = \frac{|A_{d} \cap O_{d}|}{|A_{d}|}, \quad \text{recall}_{d} = \frac{|A_{d} \cap O_{d}|}{|O_{d}|}.
\]

\(O_{a/u}\) indicates the known correct set of expected actors/use cases that can be easily derived by an oracle. Similarly, \(O_{d}\) indicates the correct set of dependences among actors and use cases. Instead, \(A_{a/u}\) represents the set of actors/use cases in the use case diagram modeled by a participant or a given group of participants, while \(A_{d}\) is the set of dependencies between actors and use cases. Accordingly, \(\text{precision}_{a/c}\) measured the correctness of both actors/use cases belonging to a given use case diagram. The correctness of the dependencies between actors and use cases was measured by using \(\text{precision}_{d}\). On the other hand, \(\text{recall}_{a/u}\) measured the completeness of a use case diagram with respect to its actors/use cases, while \(\text{recall}_{d}\) was used to get a quantitative assessment of the completeness of a use case diagram with respect to the dependencies among actors and use cases.

Precision and recall quantitatively summarize two different concepts. Therefore, we used their harmonic mean [38] to get a balance between correctness and completeness of actors/use cases within a use case diagram:

\[
F - \text{Measure}_{a/u} = \frac{2 \times \text{precision}_{a/u} \times \text{recall}_{a/u}}{\text{precision}_{a/u} + \text{recall}_{a/u}}.
\]

Similarly, we employed the harmonic mean of precision and recall to get a balance between correctness and completeness of the dependencies among actors and use cases:

\[
F - \text{Measure}_{d} = \frac{2 \times \text{precision}_{d} \times \text{recall}_{d}}{\text{precision}_{d} + \text{recall}_{d}}.
\]
\( F - Measure_{a/u} \) quantitatively summarizes the accuracy of a use case diagram with respect to its actors/use cases and compared with an oracle. Similarly, \( F - Measure_d \) summarizes the accuracy of the dependencies among actors and use cases with respect to an oracle. The quality dependent variable is computed as the arithmetic mean of \( F - Measure_{a/u} \) and \( F - Measure_d \).

All the measures above assume values in between 0 and 1. Whatever is the measure, 0 is the worst value and 1 is the best. As for quality, values close to 1 mean that students modeled use case diagrams very similar to the oracle. Conversely, values close to 0 indicate that modeled diagrams are far from the oracle.

The quality variable has been defined to give the same relevance to correctness and completeness of use case diagrams with respect to both actors/use cases and dependencies among them. In order to better comprehend how the defined measure works, we provide an example of its application. In particular, let the use case diagrams shown in Figure 3 and Figure 5 be respectively the diagram modeled by a group of participants and an oracle, the quality measure assumes 0.825 as the value. Table 2 summarizes how this value has been obtained with respect to the other measures on which quality is based on.

![Figure 5: Oracle](image-url)

The authors developed the oracles (one for each experiment object) before the experiments took place. An independent expert successively reviewed the oracles to detect possible defects. The reader may object to the fact that there could be more ways for modeling use case diagrams. This is true, but our method is fair enough so conditioning the quantitative assessment of the
quality of use case diagrams in favor of neither F2F nor TPS. In addition, we asked the participants to not specify relationships (e.g., inclusions or extensions) among use cases and to only report user-goal use cases. Please also note that the use of different measures (e.g., based on a checklist) does not reduce the concern that there are more ways for modeling use cases. The use of quantitative assessments conducted by experts could introduce different threats and could affect the repeatability of our empirical investigation. Summarizing, the defined and used measure is trustworthy for the problem at hand. The application of that measure in different context could be subject of future work.

The values for the quality variable may be also affected by ambiguities in associations between use case names of the oracle and use case names of the diagram modeled by the participants. To effectively carry out this association use cases should be deeply analyzed and comprehended. This was not possible since the participants were not asked to produce use case narratives. The rationale for this design choice was inspired by the need of achieving a trade-off between complexity of modeling tasks and the fatigue to accomplish them. To control as much as possible threats related to the association between use case names of an oracle and use case names of the diagram modeled by the participants, the authors and an independent expert conducted a meeting. The goal of this meeting was to discuss and solve possible issues and ambiguities.

3.6. Pilot, Preparation, and Execution

Before Experiment I, we conducted a pilot experiment with four students of the Bachelor program in Computer Science at the University of Basilicata. They were volunteers and were not successively involved in the controlled
experiments. The goals of the pilot experiment were: (i) getting some indications on task completion time and (ii) testing our computer-mediated communication and modeling environment. The results indicated that the experiment objects were well suited and that one hour and half was sufficient to accomplish the experiment tasks.

The participants attended a training session before Experiment I. We posed great care to explain how they had to document use case diagrams. We explained that we were interested in use case diagrams containing only user-goal use cases and that relationships among use cases were not required (only relationships between actors and use cases had to be present). We also gave instructions on the experimental procedure to be followed and details on both the method and the integrated environment. We also asked the participants to freely use our integrated environment. At the end of the training session, the participants filled in and returned to the experiment supervisors the pre-questionnaire mentioned in Section 3.2.

Each experiment was organized in the following steps:

- No interaction among participants was allowed and no time limit was imposed to accomplish the tasks. We provided the participants with a pencil, some paper sheets, and the printed copy of the following material: (i) the user’s manual of our modeling and communication environment; (ii) a set of slides describing the experimental procedure to follow; and (iii) the specification documents of the software systems to be used in the tasks. We did not give the participants details on how dealing with their modeling tasks. In case participants had issues with specification documents, we gave them some clarifications. Regarding Run2, specification documents were given to the participants only when they accomplished the modeling task of Run1.

- At the end of each experiment, the participants were asked to fill in a post experiment survey questionnaire (see Table 3). This survey mainly aimed at getting information about the perceived: (i) quality of the provided material; (ii) effort to perform the tasks and their clarity; (iii) satisfaction about the methods used in the experiments. All the statements expected closed answers according to the following five-point Likert scale [39]: 1 (strongly agree); 2 (agree); 3 (neither agree nor disagree); 4 (disagree); and 5 (strongly disagree). To avoid that participants answered without paying attention to these statements, we
formulated them in either positive or negative form (see for example Q2 and Q3).

Table 3: Post experiment questionnaire

<table>
<thead>
<tr>
<th>ID</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I had enough time to accomplish the tasks.</td>
</tr>
<tr>
<td>2</td>
<td>The objectives of the experiment were perfectly clear.</td>
</tr>
<tr>
<td>3</td>
<td>Face-to-face made problematic the interaction with team members.</td>
</tr>
<tr>
<td>4</td>
<td>Modeling a use case diagram was problematic when performing the task within a face-to-face session.</td>
</tr>
<tr>
<td>5</td>
<td>I found useful the synchronous computer-mediated communication and modeling environment.</td>
</tr>
<tr>
<td>6</td>
<td>The synchronous computer-mediated communication and modeling environment made problematic the interaction with team member.</td>
</tr>
<tr>
<td>7</td>
<td>Modeling a use case diagram was problematic when performing the task with the synchronous computer-mediated communication/modeling environment.</td>
</tr>
<tr>
<td>8</td>
<td>It was difficult to apply the method and the communication/modeling environment to accomplish the task assigned to me.</td>
</tr>
<tr>
<td>9</td>
<td>The use of the method and the communication/modeling environment was too much expensive to be applied.</td>
</tr>
<tr>
<td>10</td>
<td>The method and the synchronous computer-mediated communication and modeling environment easily supported the communication, while accomplishing the task assigned to me.</td>
</tr>
<tr>
<td>11</td>
<td>The experiment was useful from the practical and pedagogical point of views.</td>
</tr>
</tbody>
</table>

Regarding the PCs used in the experiments and in the training session, we installed the CoFFEE Controller on 5 desktops (i.e., the server nodes), while we installed on the participants’ PCs the CoFFEE Discusser. All the PCs were equipped by a 3.0 GHz Intel Pentium 4 with 1 GB of RAM and Windows XP Professional SP 3 as operating system. The participants’ PCs were connected to the corresponding servers by a LAN. The size of each monitor was 15 inches with a resolution of 1024x768. The Discusser tool was run in full-screen mode and the iconizable button was disabled. This forced the participants to use only our tool during the laboratory runs.
3.7. Analysis Procedure

We show the distribution of the data (e.g., values of time and quality and answers to the post-experiment survey questionnaire) using boxplots. They provide a quick visual representation to summarize the data using five numbers: the median, upper and lower quartiles, minimum and maximum values, and outliers. We also summarize the gathered data using descriptive statistics (i.e., minimum, maximum, median, mean, and standard deviation).

To test the defined null hypotheses, we planned to use parametric statistical tests. To apply parametric tests, the normality of data is verified by means of the Shapiro-Wilk test [40]. It tests the null hypothesis that the sample is drawn from a normally distributed population. A p-value smaller than the $\alpha$ threshold allows us to reject the null hypothesis and conclude that the distribution is not normal. We opted for this normality test because of its good power properties as compared to a wide range of alternative tests. In case the p-value returned by the Shapiro-Wilk test is larger than the $\alpha$ threshold, we used an unpaired t-test. We exploited unpaired tests due to the experiment design (participants experimented TPS and F2F on two different experiment objects).

Other than testing null hypotheses, it is of practical interest to estimate the magnitude of performance differences (time and quality) of the participants when using TPS and F2F. The effect size can be computed in several ways. In case of parametric analyses, we used Cohen’s $d$ [41] to measure the difference between two groups. The effect size can be considered negligible for $|d| < 0.2$, small for $0.2 \leq |d| < 0.5$, medium for $0.5 \leq |d| < 0.8$, and large for $|d| \geq 0.8$. These thresholds are those suggested in [42].

For each test performed, we also analyzed its statistical power (i.e., post hoc or retrospective power analysis). To compute the power, we also took into consideration the type of hypothesis tested (one-tailed vs. two-tailed). Whatever is the kind of test, statistical power represents the probability that a test will reject a null hypothesis when it is actually false. Then, it is worthy to be analyzed only in case a null hypothesis is rejected. The highest value is 1, while 0 is the lowest. The higher statistical power, the higher the probability to reject a null hypothesis when it is actually false. The value 0.80 is considered as a standard for adequacy [43].

We used interaction plots to study the presence of possible interactions between the main factor and the Run/Object co-factors. They are line graphs in which the means of the dependent variables for each level of one factor are plotted over all the levels of the second factor. If the lines are almost
parallel no interaction is present, an interaction is present otherwise. Cross 
lines indicate a clear evidence of an interaction between factors. If data are 
normal distributed (tested with the Shapiro-Wilk test) and their variance 
is constant (by the Levene test), we used a two-way Analysis of Variance 
(ANOVA) as alternative to the interaction plots. It is worth mentioning 
that the analyses for Run and Object coincide in both the experiments (see 
Section 3.4) because of the experiment design we adopted.

In all our statistical tests, we decided (as usual) to accept a probability 
$\alpha = 5\%$ of committing Type-I-Error, i.e., of rejecting a null hypothesis 
when it is actually false, namely its alternative hypothesis is actually true.

4. Results

In the following subsections, we present the results for Experiment I first 
and then those for Experiment II. Next, we present the results on the effect 
of participants’ familiarity with TPS and highlight the outcomes of the post 
experiment survey questionnaires.

4.1. Experiment I

As for Experiment I, descriptive statistics on time and quality (grouped 
by Object and Method) are shown in Table 4 and Table 5, respectively. These 
statistics show that the average time to accomplish the tasks using TPS was 
much more than the mean time to accomplish the same tasks with F2F. For 
quality, we can observe that the participants achieved on average slightly 
better scores (the difference is 0.07) on Run1 using F2F and slightly better 
scores (the difference is 0.02) on Run2 using TPS.

The values for time and quality grouped by experiment are also shown by 
means of boxplots (see on the left hand side of Figure 6). Boxplots show that 
the time to accomplish a task using TPS was larger than that to accomplish 
a task with F2F. Regarding quality, the boxplots in Figure 6 show that: the 
distribution is more skewed for F2F and three outliers are present (two for 
TPS and one for F2F). Overall, there is not a substantial difference between 
the distributions of TPS and F2F.

4.1.1. Effect of Method

For both the dependent variables, the results of the Shapiro-Wilk test 
showed that parametric statistical analyses could be applied. Regarding
Table 4: Descriptive statistic on time (in minute) for Experiment I

<table>
<thead>
<tr>
<th>Method</th>
<th>Object1 (Library), Run1</th>
<th>Object2 (FilmCollection), Run2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS</td>
<td>239</td>
<td>348</td>
</tr>
<tr>
<td>F2F</td>
<td>240</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 5: Descriptive statistic on quality for Experiment I

<table>
<thead>
<tr>
<th>Method</th>
<th>Object1 (Library), Run1</th>
<th>Object2 (FilmCollection), Run2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS</td>
<td>0.535</td>
<td>0.925</td>
</tr>
<tr>
<td>F2F</td>
<td>0.76</td>
<td>0.865</td>
</tr>
</tbody>
</table>

time, the p-values are: 0.237 for TPS on and 0.217 for F2F, while the p-values are 0.317 and 0.085 for TPS and F2F on quality, respectively. Table 6 summarizes the results of the unpaired t-test for Hn1 and Hn2 and reports the values of effect size and statistical power. This table also shows the number of teams that spent more time to accomplish the task with F2F (F2F > TPS) and the number of teams that spent less time to accomplish the task with TPS (F2F < TPS). Similarly, the results on quality are reported.

Table 6: Results of the analyses on Method for Experiment I

<table>
<thead>
<tr>
<th>Hypotheses</th>
<th>p-value</th>
<th>Effect size</th>
<th>Stat. Power</th>
<th>F2F &gt; TPS</th>
<th>F2F &lt; TPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hn1 (time)</td>
<td><strong>YES (&lt;0.001)</strong></td>
<td>Large (1.304)</td>
<td>0.972</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Hn2 (quality)</td>
<td>NO (0.371)</td>
<td>Small (-0.316)</td>
<td>0.131</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

The t-test revealed that Hn1 (i.e., the null hypothesis used to assess the influence of Method on time) can be rejected since the p-value is less than 0.001. This value is highlighted in bold. The effect size was large (i.e., 1.39) and the statistical power was 0.972. Additionally, 8 out of 9 teams spent less time to accomplish modeling tasks when using F2F.

No significant difference on quality was observed (p-value = 0.371) in case participants used F2F and TPS (i.e., Hn2 cannot be rejected). The effect size is small as the Cohen d value is -0.34 (i.e., a small practical difference was present in favor of F2F on quality). The $\beta$-value is high (i.e., 0.869): the

---

It represents the probability of committing a Type-II-Error. That is, the probability of accepting the null hypothesis when it is actually false. Therefore, it makes sense to discuss about $\beta$-values only in case null hypotheses are not rejected. The $\beta$-value
probability that the null hypothesis $H_{n2}$ is false and that the unpaired $t$-test erroneously fails to reject that hypothesis is low. Finally, 5 teams modeled better quality use case diagrams using TPS (5 out of 9).

4.1.2. Effect of Object

As far as the co-factors Method and Object are concerned, the p-values returned by the Shapiro-Wilk test were: 0.717 (TPS and Library), 0.307 (TPS and FilmCollection), 0.227 (F2F and Library), and 0.352 (F2F and and FilmCollection). The variance was constant as the results of the Levene test suggested (the p-values are 0.113 for Method and 0.119 for Object). Therefore, we could apply a two-way ANOVA on time. The results for time is computed as 1 less the value of the statistical power.
Table 7: ANOVA results for Experiment I

<table>
<thead>
<tr>
<th>Factor</th>
<th>Time</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>YES (&lt; 0.001)</td>
<td>NO (0.703)</td>
</tr>
<tr>
<td>Object</td>
<td>YES (&lt; 0.001)</td>
<td>NO (0.69)</td>
</tr>
<tr>
<td>Method vs. Object</td>
<td>YES (0.002)</td>
<td>NO (0.414)</td>
</tr>
</tbody>
</table>

are summarized in Table 7. We can observe a positive effect of: Method (confirming with stronger evidence the effect of Method on time) and Object (the time to accomplish the task on Library was significantly less than that to accomplish FilmCollection). The effect of Object could be due to the different levels of participants’ familiarity with the problem domains of the systems used in Experiment I. An interaction between Method and Object is present as well. This result suggests that on some software systems the use of a method (i.e., F2F and TPS) could make the difference in the time spent to accomplish a task (i.e., modeling a use case diagram).

We also applied a two-way ANOVA on quality (see Table 7). The p-values returned by the Shapiro-Wilk test were: 0.051 (TPS and Library), 0.117 (TPS and FilmCollection), 0.406 (F2F and Library), and 0.426 (F2F and and FilmCollection). The variance is constant: the p-values are 0.613 for Method and 0.821 for Object. The two-way ANOVA test showed: the absence of a statistical significant effect of Method and Object and no interaction between these two factors. The participants’ familiarity with the problem domain of the two systems affected the time to accomplish the tasks, but not the quality of the models produced.

4.2. Experiment II

The descriptive statistics on time and quality for Experiment II are reported in Tables 8 and 9, respectively. These statistics are grouped by Object and Method. Similarly to Experiment I, the mean value for time with TPS was much more than that with F2F. As far as quality is concerned, the results for TPS and F2F are similar on Rent. With the exception of standard deviation, the descriptive statistics for TPS on ECP are slightly better that those for F2F on the same experiment object.

The boxplots for time and quality are shown on the right hand side of Figure 6. The time to accomplish tasks using TPS was on average larger than that to accomplish tasks with F2F. Regarding quality, the boxplots suggest that the quality of use case diagrams was slightly better when using TPS.
Table 8: Descriptive statistic on time for Experiment II

<table>
<thead>
<tr>
<th>Method</th>
<th>Object1 (Rent), Run1</th>
<th>Object2 (ECP), Run2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS</td>
<td>132</td>
<td>188</td>
</tr>
<tr>
<td>F2F</td>
<td>60</td>
<td>160</td>
</tr>
</tbody>
</table>

Table 9: Descriptive statistic on quality for Experiment II

<table>
<thead>
<tr>
<th>Method</th>
<th>Object1 (Rent), Run1</th>
<th>Object2 (ECP), Run2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS</td>
<td>0.575</td>
<td>0.94</td>
</tr>
<tr>
<td>F2F</td>
<td>0.65</td>
<td>0.845</td>
</tr>
</tbody>
</table>

4.2.1. Effect of Method

The results of the Shapiro-Wilk test indicated that the unpaired t-test could be applied. For time, the returned p-values were: 0.494 for TPS and 0.246 for F2F. The p-values were 0.579 and 0.272 for TPS and F2F on quality, respectively. Table 10 summarizes the results of the parametric analyses we performed. The null hypothesis Hn1 can be rejected since the p-value returned by the t-test was less than 0.001. The effect size was large (the Cohen’s d value was 1.624). The statistical power was high (0.997) and 8 out of 9 teams spent less time to accomplish a task when using F2F with respect to TPS. Similar to Experiment I, we could not reject the null hypothesis Hn1 (p-value = 0.175). The effect size is small as the Cohen’s d value showed (0.496). The $\beta$ value is 0.741. 6 out of 9 teams modeled better use case diagrams when using TPS.

4.2.2. Effect of Object

The Shapiro-Wilk test returned the following p-values: 0.237 (TPS and Rent), 0.246 (TPS and ECP), 0.079 (F2F and Rent), and 0.776 (F2F and ECP). The p-values obtained by applying the Levene test were: 0.63 and 0.523 for Method and Object, respectively. Accordingly, we could apply the two-way ANOVA, which showed a statistically significant effect of Method on time (confirming the results of the unpaired t-test): the p-value was less

Table 10: Results of the analyses on Method for Experiment II

<table>
<thead>
<tr>
<th>Hypotheses</th>
<th>p-value</th>
<th>Effect size</th>
<th>Stat. Power</th>
<th>F2F&gt;TPS</th>
<th>F2F&lt;TPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hn1 (time)</td>
<td>YES (&lt;0.001)</td>
<td>Large (1.624)</td>
<td>0.997</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Hn2 (quality)</td>
<td>NO (0.175)</td>
<td>Small (0.496)</td>
<td>0.259</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>
than 0.001. The results of this test also showed that the effect of Object was not statistically significant (p-value = 0.108). No interaction between Method and Object was also shown (p-value = 0.352).

As far as quality is concerned, a two-way ANOVA could not be applied: the p-value returned by the Shapiro-Wilk test was less than 0.05 for F2F on the object Rent. Therefore, we used the interaction plots to study the interaction between Method and Object on quality. The interaction plots (see Figure 7) showed a slightly interaction. The interaction plots also indicated that the quality of the use case diagrams for Rent is nearly the same for F2F and TPS. For ECP, these plots suggested that the participants in Experiment II modeled with TPS use case diagrams that are better than those modeled with F2F in terms of quality.

4.3. Participants’ Familiarity with TPS

We could apply parametric analyses also to test Hn3 and Hn4 (see the Shapiro-Wilk test results shown above). The results of these analyses are summarized in Table 11. The results of the unpaired t-test allowed rejecting Hn3 (p-value < 0.001). The effect size was large (2.079) and the statistical power was high (0.999). Moreover, 9 teams out of 9 spent less time in Experiment II with respect to Experiment I, when using TPS. This finding is
Table 11: Results of the analyses on the participants’ familiarity with TPS

<table>
<thead>
<tr>
<th>Hypotheses</th>
<th>p-value</th>
<th>Effect size</th>
<th>Stat. Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hn3 (time)</td>
<td>YES (&lt;0.001)</td>
<td>Large (2.079)</td>
<td>0.999</td>
</tr>
<tr>
<td>Hn4 (quality)</td>
<td>0.603</td>
<td>Negligible (0.09)</td>
<td>0.043</td>
</tr>
</tbody>
</table>

even clearer in Figure 6 (white boxes of Experiment I and Experiment II for time). The red boxes (dark grey in case of a black and white copy of the paper) in Figure 6 also suggest that passing from Experiment I to Experiment II a reduction in term of time was also present for F2F. The results of the t-test showed that this difference was statistically significant (p-value = 0.012). The effect size was large (0.931) as well as the statistical power (0.816). The reduction of the task completion time for TPS was much larger than that for F2F (e.g., the values of the effect size are 2.079 vs. 0.931). This result could suggest that TPS more than F2F allowed each participant to consolidate his/her working style and develop a shared working habit with other participants.

As for quality, we did not reject Hn4: the participants familiarity with TPS did not affect the quality of the modeled use case diagrams. The effect size is 0.09 (negligible), while the \( \beta \)-value is 0.957. As shown in Figure 6, the white boxes are proportioned (even if two outliers are present in Experiment I) and the medians of the distributions for quality are almost the same (see also Table 5 and Table 9).

4.4. Further Analysis

We also computed the magnitude of the quality differences among the subsequent steps of the method in both the experiments. In particular, we computed the Cohen’s \( d \) to measure the standardized difference between the quality of the modeled use case diagrams when passing from the step think to the step pair and when passing from the step pair to the step square. As far as Experiment I, the effect size was small \( (0.411) \) in the first case, while it was large \( (0.932) \) in the second. The difference between the step pair and the step square was statistically significant. The p-value returned by the unpaired t-test was 0.025, while the statistical power was 0.748. We could apply this test because the results of the Shapiro-Wilk test (the p-values were 0.44 and 0.317, respectively). For Experiment II, the effect size was medium \( (0.614) \) between the step think and the step pair, while it was small \( (0.454) \) between the step pair and the step square. In the first case, the unpaired t-test revealed a statistically significant difference since the p-value
was 0.038 (the Shapiro-Wilk test returned as the p-values 0.154 and 0.449, respectively). The statistical power was 0.768. To complete this further analysis, we also compared the quality of the use case diagrams produced in the two experiments for each step of the method. The results showed that the effect size is negligible (0.033 and 0.09) in favor of Experiment II for the steps think and square, respectively. That difference was medium (0.339) for the step pair.

4.5. Post-Experiment Survey Questionnaires

The data collected from the post-experiment survey questionnaires are visually summarized in Figure 8. For each statement, the distributions of the answers in both the experiments are nearly identical with the only exception of some outliers in Experiment I. Accordingly, we describe the results of the post-experiment survey questionnaire for Experiment I and Experiment II together. Differences are highlighted if present.

The results of an analysis on the answers to the statement Q1 suggest that the time to accomplish the tasks was considered adequate: any time constraint was perceived by the participants. The boxplots of Q2 show that the participants found the objectives of the experiment clear. As for Q3, the participants did not have any problem to conduct and manage face-to-face interaction. Similarly, the participants declared that face-to-face interaction did not present any concerns to execute the tasks (see the boxplots of Q4).

The boxplots of Q5 show that the participants expressed a positive judgment on the usefulness of the proposed environment. The distribution of the answers of Q6 for Experiment I is more skewed than that of Experiment II for the same statement. However, both the boxplots show that the greater part of the participants did not find problematic the interaction when using our environment. Further on, the participants stated that they did not have any problems to model use case diagrams (see the boxplots of Q7). The boxplots of Q8 show that the participants of both the experiments did not have any problem to accomplish the modeling tasks with our environment. The analysis of the answers to the statement Q9 suggests that our method and environment were considered neither cheap nor expensive by the participants in the experiments. The boxplots of Q10 show that the participants found easy the communication with the other participants when using the method and its supporting environment. Finally, the greater part of the participants in the experiments found very useful the experiments from the practical and pedagogical point of views (see the boxplots of Q11).
Summarizing, the participants generally manifested a larger satisfaction level when using face-to-face. In particular, the interaction and the requirements specification were considered simpler with respect to the use of our integrated environment. However, a slight difference between the median values of Q3 and Q6 is observable. Similarly, the participants found simpler the modeling of use case diagrams when using face-to-face (see the boxes of Q4 and Q7).

5. Discussion

In the following, we first discuss the achieved results and then their possible practical implications. A discussion on the threats that could affect the validity of our results concludes this section.
5.1. Discussion

The results of the data analysis conducted on Experiment I and Experiment II indicate a significant difference between face-to-face interaction and the new method on the time needed to model use case diagrams, with no significant impact on quality. In both the experiments, the teams spent significantly less time to accomplish the tasks using face-to-face, while the quality of the produced models is almost the same.

The data analysis on Experiment I alone indicated that the members within a team needed less time to accomplish the task in Run2, when using F2F (see descriptive statistics in Table 4 and consider the interaction between Method and Run). The new method might be used before a face-to-face session to reduce the time for documenting functional requirements. Differently, this result did not hold in case the participants used face-to-face before. A plausible justification for this result is that face-to-face is less prone to improve cohesion among team members with respect to our method. Additionally, we also found that the new method more than face-to-face allowed each participant to consolidate his/her working style and to develop a shared working habit with other participants. This finding is on line with those present by Calefato et al. [9] and Elsan et al. [45]. However, caution is needed because in the literature related to computer-mediated communication, richer communication media are considered more prone to improve cohesion among team members.

As for Experiment II, we observed that the presence of screen mock-ups in the specification document of ECP did not affect the overall quality of the modeled use case diagrams (a slight interaction is present between Method and Object). Furthermore, the descriptive statistics of Table 9 suggested that the screen mockups did not properly supported participants in the accomplishment of a task when using face-to-face interaction. This result could somewhat seem to be in contrast with that obtained by Ricca et al. [46], where the presence of screen mockups in requirements specification documents improve the comprehension of functional requirements. Indeed, modeling is different from comprehension especially when we are dealing with synchronous and cooperative modeling of functional requirements. We also noted that the participants who first used our environment on Rent (its specification document was not added with screen mockups) produced on average worse quality use case diagrams when they successively worked on ECP with F2F (0.794 and 0.565, respectively). A slightly difference in term of the quality of use case diagrams was observed in case the participants accomplished
Table 12: Descriptive Statistic on the TPS steps

| Variable | Step | Experiment I | | | Experiment II | | |
|----------|------|--------------|------|------|--------------|------|
|          | Think| 30   | 29     | 3.1      | 15.5 | 15.5   | 4.2      |      |
|          | Pair | 20   | 19.3   | 2.5      | 13   | 12.7   | 3.5      |      |
|          | Square| 28 | 26     | 7.8      | 14   | 14.6   | 4.1      |      |
| Quality  | Think| 0.6  | 0.578  | 0.162    | 0.584| 0.584  | 0.19     |      |
|          | Pair | 0.673| 0.641  | 0.14     | 0.698| 0.698  | 0.173    |      |
|          | Square| 0.75| 0.759  | 0.1      | 0.77 | 0.77   | 0.127    |      |

The obtained results also indicated that the effect of participants’ familiarity with the method and the supporting integrated environment does not significantly affect the quality of the modeled use case diagrams. Although we cannot conclude anything definitively (the unpaired t-test did not reject Hn4), the results of a further analysis indicated a slight positive effect of the participants’ familiarity on the quality of the diagrams modeled in the tasks.

A further analysis was conducted on both the experiments to investigate how and whether the overall quality of the use case diagrams changes across the subsequent steps of the method. Table 12 reports some descriptive statistics on time and quality. These statistics are grouped by step (i.e., think, pair, and square) and experiment. The descriptive statistics show that given a step of the method the participants spent on average less time in Experiment II with respect to Experiment I (e.g., 15.5 minutes vs. 29 minutes for the step think). This result could be due to the increased familiarity of each participant with method and integrated environment. The descriptive statistics also show that on average the time to accomplish each step is almost the same in Experiment II (15.5, 12.7, and 14.6, respectively). Although the small difference, the most expensive steps seemed to be think and square in both the controlled experiments.

The descriptive statistics suggest that the quality of a use case diagram
increased when passing from a step to the next one. This hold for both the experiments. The boxplots in Figure 9 graphically show the distributions of the quality values obtained in all the steps of the new method in both the experiments. The boxplots confirm with a stronger evidence that the quality of use case diagrams increased throughout the steps of the method. It can be also noted that these boxplots are less skewed when passing from the think step to the pair step and when passing from the pair step to the square step.

Regarding the differences between the quality of the modeled use case diagrams when passing from the step think to the step pair and when passing from the step pair to the step square, we observed that: (i) the quality of the use case diagrams increased throughout the steps and (ii) a higher familiarity level of participants with method has an effect on the quality of use case diagrams in each step of the method, but this effect is not significant.

5.2. Implications

To judge the implications of our experiments, we adopted a perspective-based approach [47]. We based our discussion on the practitioner/consultant (simply practitioner in the following) and researcher perspectives [48]. The main implications of our study can be summarized as follows:

• The results of the data analysis indicated that the participants grouped in teams spent significantly less time to accomplish the tasks using face-to-face interaction, while the quality of the produced use case diagrams is almost the same. This finding is relevant for the practitioner: the new method and the integrated environment could represent a viable solution to face-to-face sessions when time difference among distributed team members is not an issue, while moving them might be a problem. In other words, this method and its supporting environment can be used in industry to reduce the cost and the time to move geographically distributed employees without affecting the quality of the modeled use case diagrams. This finding could ease the transferring of the experimented technology to practitioners. To improve our awareness on this outcome, it would be the case to study videoconferencing communication tools in the context of our empirical investigation (i.e., distributed modeling of use case diagrams).

• We observed that the new method might be used before a face-to-face session to reduce the time to model use case diagrams. Both the practi-
tioner and the researcher could be interested in this result. In particular, this finding increases the awareness of the practitioner that the new method and the integrated environment can be effectively employed in his/her own company. On the other hand, the researcher could be interested in further: studying the relation between our method and face-to-face interaction and analyzing whether cohesion among team members actually improves using that method and its supporting environment.

- The results of the data analysis suggested that the new method more than face-to-face allowed each participant to consolidate his/her work-
ing style and to develop a shared working habit with the other participants. The researcher could be interested in further investigating this finding because in the literature related to computer-mediated communication richer communication media are considered better than other media. Our findings and the results of previous studies [9, 45] pose the basis in that research direction.

- The presence of screen mockups in the specification documents slightly affected the quality of the use case diagrams when using our method and environment. This result is relevant for the practitioner because the introduction of screen mockups in specification documents has a cost [27]. The researcher could be interested in evaluating that cost in relation with the quality of the modeled use case diagrams.

- The quality of use case diagrams seemed to be not related with the familiarity of the participants with method and environment. This result has a practical implication because a training phase on that technology could be shortened without affecting the overall quality of modeling. This is also practically relevant because analysts with different profiles and skills (also customers and clients, for example) might effectively use the developed method/environment in their own company. These two findings could promote the technology transfer of the developed method and environment to the industry. This is clearly relevant also for the researcher.

- The most expensive steps seemed to be think and square. As for the think step, it could be due to the specification documents to be read and understood. The effort to communicate and to get an agreement among team members could justify the result observed for the square step. Analyzing the effect of replacing the integrated environment with a face-to-face interaction in the square step could be of interest for the researcher. If confirmed, we will increase our awareness that the method is flexible and that it can be tailored to meet company needs.

- When passing from the think step to the pair step and when passing from the pair step to the square step the quality of the modeled use case diagrams improved. This result is relevant from the researcher, who could be interested in assessing whether there is room for improvement tailoring the steps of the method. The researcher could be
also interested in adding a new step either to further refine use case diagrams or to make evident the rationale behind the choices taken.

- The use of the think-pair-square method in global software engineering has been proposed for the first time in [14], where it has been applied and assessed in the modeling of use case narratives. We have assessed here that method in a different experimental context (e.g., use case diagram modeling and participants with two levels of experience with that method). The achieved results and those presented in [14] give with stronger evidence that this method can be successfully applied in global software engineering. This is relevant from both the practitioner and the researcher perspectives.

- The study is focussed on desktop and web applications. For the practitioner, it is relevant since on these kinds of applications the new method and the developed integrated environment seem promising to be used in actual software projects. The effect of that technology on different type of software applications represents a possible future direction. This is why this point is also of interest for the researcher.

- The specification documents could be considered as developed in the following kinds of projects: in-house software, product, or contract [8]. This result is of interest for the practitioner and suggests that the new method and the integrated environment could be applied also in other kinds of software development projects.

- We considered here an integrated environment developed within a research laboratory. Using different implementations of the method could lead to different results in global software engineering. This aspect is relevant for the practitioner, interested in understanding the best way for using our method, and the researcher, interested in investigating why and how a different implementation of that method should affect synchronous communication and modeling.

- The experiment objects were realistic enough for small-sized projects. However, we are not sure that the achieved results scale to real projects. This point is of interest for the researcher.

- We assessed the method and the integrated environment using teams composed of four people. The researcher could be interested in studying
the relation between the size of a team and the quality of the use case diagrams modeled. It would be also worth analyzing the effect of the size of teams when their members are virtually or physically collocated \cite{36}. In this context, we would be of interest to investigate the effect of group dynamic issues \cite{49} (e.g., lack of focus). Again, our findings pose the basis for future studies.

5.3. Threats to Validity

The Internal Validity threat is relevant in studies that try to establish a causal relationship. This threat was mitigated by the experiment design because each group of participants in Experiment I and Experiment II worked over two tasks, on two different objects. Even if learning effect should be mitigated by this design, we found in Experiment I a significant effect of Object/Run on time (p-value < 0.001). Moreover, we observed a significant interaction between Method and Object/Run on time (p-value = 0.002). Specifically, in the second task we observed a reduction of the mean time to accomplish the tasks. As for quality, the effect of Object/Run and the interaction between Method and Object/Run are both not significant. To mitigate fatigue effect, a break between the two laboratory runs of each experiment was allowed. Another issue concerns the possible information exchanged among the participants while performing the tasks. We prevented this in several ways. Finally, to reduce internal validity threats, the participants did not know the hypotheses of the experiments and were not evaluated on their performances (task completion time and quality of modeled use case diagrams).

External Validity refers to the generalization of the results within different contexts. This threat may present when experiments are conducted with students, since they could not be representative as software professionals \cite{50}. The Masters students are not far from junior software engineers and almost all of them have working experience related to their industrial internship. The greater part of the Bachelor and Master students will work or is working in industry, so they are not far from professional software engineering. To increase our awareness in the achieved results, case studies within industrial contexts are needed. This is actually the most demanding and challenging part of our research \cite{2}. Threats to external validity can be also related to the modeling tasks. We can argue that the tasks used in the experiments are realistic for small-sized projects and they are not trivial. Further investigations with larger and more demanding modeling tasks are needed to confirm
or contradict our findings. Another possible threat concerns the fact that we did not involve participants with cultural and background diversities. In fact, space and time are not the only issues in the global software development context. Work that takes place over long distances means that communication will often involve different cultures. Regarding external validity, it is worth mentioning that all the participants successfully accomplished the tasks in both the experiments.

**Construct Validity** threats concern the possibility that the relationship between cause and effect is causal. This validity was mitigated by the experiment design used. Selection and measurements of the dependent variables could threaten construct validity. Quality was measured using an information retrieval based approach to avoid as much as possible any subjective evaluation (see Section 3.5). However, oracles could affect the measurement of quality in case they are incomplete (e.g., an actor is not present). The value of quality for a use case diagram that correctly included parts not in the oracle would be incorrectly computed. We prevented this issue in two ways: (i) carefully defining the oracles before the experiments took place and (ii) performing a post analysis on the use case diagrams modeled in the experiments. Although the results of that analysis did not reveal any issue (i.e., no teams correctly included in their diagrams actors and/or use cases that should have been present in the oracles), construct validity could be still affected by the used oracles. As mentioned in Section 3.5, different assessments (quantitative and qualitative) of quality could lead to different biases (e.g., repeatability), which could be even worse than those we could have by applying our approach. Regarding time, we asked the participants to note down the start and stop time when they accomplished modeling tasks with F2F. This information was validated by supervisors. Another possible threat concerned the fact that we did not involve stakeholders with different roles. Modeling was mostly based on the specification documents of the software systems used in the experiments. The specification documents were developed to make as much as possible clear the user perspective about the systems to be modeled. Although this could appear as an oversimplification, this scenario is quite common for small-sized development projects.

The experiment design also mitigated **Conclusion Validity** threats, which concern the relationship between the treatment and the obtained findings. In our experiments, the selection of the population may affect this experimental validity. To reject the null hypotheses, we used statistical tests and power analyses. In case differences were present but not significant, this
was explicitly mentioned, analyzed, and discussed. The conclusion validity could be also affected by the sample size. In each experiment, we involved 36 students grouped in 9 teams. Since each team accomplished two tasks in each experiment, we had 36 observations in total. A larger number of teams in studies similar to the one we have presented here is not common in the literature (e.g., [13]). Finally, the used post experiment survey questionnaire was designed using standard ways and scales [39].

6. Related Work

Similarly to traditional software engineering, global software engineering requires that stakeholders spend a large part of their time to communicate both indirectly, by means of software artifacts [51, 52, 53], and directly, through meetings and informal conversations [10, 54]. In the following subsections, we first present tools and environments for supporting distributed and cooperative modeling and then empirical studies on the effect of communication media in the requirements engineering process.

6.1. Distributed and Cooperative Modeling

The computer supported cooperative work research is focused on supporting cooperation in many applications including distributed software engineering and software modeling [5, 55]. This resulted in the development of a number of distributed synchronous and asynchronous modeling environments that enable geographically dispersed developers to edit and discuss about the same model. In the literature, a large number of these tools has been proposed and analyzed [56, 57, 58, 59, 60, 61]. Accordingly, we cannot be exhaustive and therefore we present and discuss only widely known modeling environments in this section. Dewan and Riedl [56] describe Flecse, an integrated environment to support concurrent software engineering. Flecse integrates a number of tools to support software engineers in the development process (e.g., inspection and debug of source code artifacts). The authors also present the results of a study on concurrent software engineering applied to different software development tasks. The study shows that this kind of development increases the effectiveness of teams working together. Differently from the method we took into consideration in this paper, Flecse does not allow synchronous, concurrent, and cooperative modeling based on the Unified Modeling Language (UML) [20]. From a different perspective, Grundy et al. [62] present an integrated environment to provide multiple
textual and graphical views for constructing programs in an object-oriented language. The tool supports synchronous and asynchronous modeling and the versioning of the created models. The modeling is not based on the UML. An extension of this environment is presented in [57]. That extension supports the synchronous and asynchronous object-oriented development in cooperative fashion. Differently from our integrated environment, it is not based on a method conceived for distributed modeling.

Many UML tools provide some functionality for cooperative modeling. For instance, Boulila [63] focuses on the specific problem of distributed brainstorming and the construction of UML models of software through successive distributed teamwork meetings. In particular, the authors investigate the issues in designing a unified framework based on CSCW concepts and software engineering to support concurrent object-oriented software analysis and design phases. De Lucia et al. [59] present a synchronous concurrent environment for cooperative modeling of UML diagrams. Change and configuration management functionalities for both the diagrams and graphical objects are provided. De Lucia et al. [59] also propose COMOVER [60], a tool for asynchronous modeling based on UML diagrams. This tool is also integrated in an artifact management system to enable the reuse of model elements. The main difference between our integrated environment and the tools presented in [59, 60, 63] is that these tools are founded on well-established practices in the traditional software engineering. Furthermore, no empirical studies in terms of controlled experiments are performed to assess the validity of the proposed tools.

Approaches and tools for the synchronously modeling of UML diagrams based on E-whiteboard are proposed [58, 64]. These have a pen-based sketching interface to support cooperative design, including free-hand annotations and handwritten text. These modeling tools are conceived to support early design-stage and the communication among software engineers is only supported by a large electronic whiteboard and by using an interaction style similar to that of traditional whiteboards. These points represent the most relevant difference with respect to our environment that also supports synchronous communication among team members. In addition, we implement in this environment a method conceived for distributed modeling.

In large and complex organizations, collaboration across different locations is a key success factor [65, 66]. Salger et al. [65] present the results of a two-years research initiative carried out at Capgemini sd&m. In particular, the authors present an assessment framework addressing challenges
in the global software development and how it can be tailored to different development models supporting offshore development as well as outsourcing of work to third party suppliers. The most important lesson learned of that two-years initiative is the insight that global software engineering necessitates a deep change in a software engineers work habits. This requires the development of special conceived tools as we have done in the research presented in this paper. Fassbinder and Henz [66] present approaches to analyze and improve collaboration across locations and across functions. These approaches are applied in different Siemens business units. The results indicate that potential improvements could be related to project setup, process, knowledge management, communication, and training. Differently from the study presented in this paper, the benefits deriving from the collaboration among team members are not investigated through controlled experiments.

6.2. The Effect of Communication Media on Requirements Engineering Tasks

A typical and meaningful example in which communication plays an important role (both in the traditional and distributed software engineering) is the requirements engineering process [6, 10]. In the past [2], the effect of different communication media in the requirements engineering process has been marginally investigated. These studies are generally related to media theories, where a critical claim regards the assertion that the stakeholders’ performances improve when a medium with an appropriate richness is used. These conceptual contributions do not deal directly with work groups or with how technology affects them. For example, Damian et al. [10] show an empirical study to compare five physical group configurations: one face-to-face and four distributed. In case of distributed settings different locations of the stakeholders were considered. The stakeholders’ communication was based on computer-conference communication. This study indicates that the highest group performance occurred when clients are separated from each other and collocated with the system analyst. Our work is different because we focussed on a well defined phase of the requirements engineering process (i.e., distributed modeling of use case diagrams) and because we proposed a new method based on think-pair-square.

Calefato et al. [11] present the results of a study on the effect of using a synchronous unstructured text-based communication in distributed requirements workshops. The authors compare the communication media in terms of satisfaction with performance and comfort with communication mode. Results indicate that the satisfaction with performance is better when
text-based communication is used. Participants also found more comfortable face-to-face interaction. Several are the differences with respect to our investigation. The main difference concerns the kind of task the participants are asked to accomplish in the requirements engineering process and the fact that interaction among distributed participants is based on a novel method for synchronous communication and modeling.

Erra and Scanniello [12] present the results of an empirical study in the context of requirements negotiation. They compare a traditional face-to-face meeting, an enhanced chat, and a three dimensional virtual environment (implemented in Second Life). In this comparison, they consider the time needed to negotiate software requirements and the quality of the negotiation. The latter two media reproduce/simulate face-to-face communication through synchronous computer-mediated communication tools. The results suggest that there is a difference in favor of face-to-face meeting regarding the time (less time is needed to negotiate requirements), while the quality of negotiation is not significantly influenced by the used media. There are several differences between the research reported in [12] and the study presented in this paper. The main difference is that we assess here a new approach to allow the synchronous communication among stakeholders and to cooperatively and synchronously modeling functional requirements with use case diagrams. More recently, Erra et al. [14] preliminarily assess an approach for the specification of use case narratives in the context of global software development. The results of this study show a significant difference in favor of the face-to-face interaction with respect to the time for creating use case narratives with no significant impact on their quality. The quality of these use case narratives is measured using an approach based on checklists. Several are the differences with the investigation presented in this paper. The most relevant difference concerns the kind of modeling task participants accomplished: use case narratives vs. use case diagram based specifications. As direct consequence, the supporting integrated environment is different and a distinct new measure to quantitatively assess the quality of the produced models is used. Further on, we also investigate here the effect of the participants' familiarity with method and integrated environment. Although the number of differences among the research presented in this paper and that presented in [12, 14], there is a similar pattern in the results: there is a difference in favor of face-to-face meeting regarding the time, while the quality of the produced artifacts is not significantly influenced by the media.

An investigation on the effect of using mixed media (i.e., rich and lean) in
distributed requirements negotiations is proposed by Damian et al. [13]. The study is conducted with graduate, undergraduate, master, and doctorate students. Differently from us, the students used an asynchronous text-based tool and a synchronous videoconferencing based communication tool. The results of that study show that requirements negotiation is more effective when an asynchronous structured discussion is conducted before a synchronous negotiation meeting. This finding is consistent with the results of the study presented in this paper. The authors also observe that asynchronous communication is useful to resolve issues related to uncertainty, thus enabling the videoconferencing negotiations to only concentrate on removing ambiguities. It is worth remarking that the authors do not compare computer-mediated communication tools with face-to-face communication as we have done in our research. However, we have obtained results consistent with those shown in [13].

Calefato et al. in [9] present an empirical study to analyze the effectiveness of synchronous computer-mediated communication in requirements elicitations and negotiations. The most relevant results can be summarized as follows: (i) face-to-face communication is not always the most preferred medium for requirements tasks and in a number of conditions text-based communication is preferred; (ii) team performances are often not affected by the communication medium used in the requirements tasks; and (iii) distributed requirements elicitation is the task where computer-mediated communication tools have most opportunity for successful application. Differently from us, Calefato et al. apply well established practices of traditional software engineering to global software development.

In distributed requirements engineering, empirical investigations different from controlled experiments have been also conducted. For example, Boehm and Egyed in [67] capture and analyze requirements negotiation behavior for groups of 90 undergraduate students. The participants use an instrumented version of the WinWin groupware system on 15 different projects. The results of this study show several real problems, such as fuzzy requirements, conflicts with resources and personnel, and so on.

6.3. Remarks

The obtained results indicate that there is a significant difference between face-to-face and the proposed method with respect to the quality of the produced models. Indeed, the quality of the models produced using our supporting integrated environment is on average better than the quality of the models produced in a face-to-face session. Furthermore, the familiarity
of the participants with that method and integrated environment does not significantly affect the quality of the models, while better trained participants significantly spend less time to accomplish the tasks. In some sense, our empirical investigation can be considered as based on the results of the study by Calefato et al. [9]: (i) face-to-face communication is not always the most preferred medium and in a number of conditions text-based communication is preferred and (ii) team performances are often not affected by communication medium. Since our results are coherent with those presented in that paper and those shown in [11, 14], we can then assert that our empirical investigation increases the body of knowledge in the context of synchronous computer-mediated communication in distributed requirements engineering. That is, the quality of the models is not affected by the used communication, so synchronous text-based communication could represent a viable alternative to face-to-face when time difference is not an issue, but moving people might be a problem. In addition, the findings achieved in our investigation and those presented in literature (e.g., [9, 10, 11, 12]) suggest that it is now the case of using a baseline different from that used in the past (i.e., face-to-face). In fact, in the context of distributed requirements engineering, it seems that young software engineers obtain similar results when using both face-to-face and computer-mediated methods with respect to both quantifiable and non-quantifiable benefits. For example, for a given software engineering task a baseline could be a method well established in traditional software engineering using text-based communication.

7. Conclusions and Future Work

This work is in the direction of the media-effect theories applied to requirements engineering. Most of these theories indicate that face-to-face is the richest medium with respect to all the other communication media (including computer conferencing) [68]. These theories assert that the participants’ performances decrease when leaner media are used [69]. As recently shown in the global software engineering literature [9, 11, 12, 14], these claims could not hold for distributed requirements engineering tasks. A plausible justification for that finding might be related to the familiarity of the participants with social networks (e.g., Facebook, Google+, and Twitter) and instant messaging tools (e.g., Skype). The use of these tools is changing (or has changed) the way in which people communicate and deal with software engineering tasks (e.g., [70]).
In this paper, we have assessed a new method, which has been implemented in a synchronous computer-mediated communication/modeling environment. The new method is based on think-pair-square [21, 22]. To assess the method and its supporting environment, we have conducted two controlled experiments. In particular, these experiments aimed at investigating whether the method and the environment (participants are virtually collocated) are effective as face-to-face interaction (participants are physically collocated). The results indicate that the participants in the experiments significantly spent less time when modeling use case diagrams using face-to-face. Conversely, no significant difference was observed on the quality of the artifacts produced by the participants in these tasks.

We exploited controlled experiments because they allow controlling a number of confounding and uncontrollable factors that could be present in case studies conducted both in research laboratory and in industrial settings [32, 71]. For example, in case studies it is quite impossible to control factors such as learning/fatigue effects and to select specific tasks. Furthermore, controlled experiments should be conducted in the early steps of an empirical investigation plan and before conducting empirical investigations in more realistic settings, see for example [72].

The conducted experiments were designed considering quantifiable benefits. Concerns and advantages deriving from the work in groups [31] have been ignored as well as cultural and background diversities [37]. Despite this design choice, we were able to conduct a very preliminary qualitative analysis on how charisma and the authoritativeness of participants affect team performances. This was possible due to our personal knowledge of the participants in the experiments. The results suggested that: charisma and authoritativeness affect more the modeling results in case of face-to-face interaction. On the basis of this result, we are going to conduct a qualitative study to get a better understanding on charisma, authority, and how participants’ ability affects team performances.

Future work will be also devoted to improve our communication/modeling environment to better support the proposed method. For example, in the future implementation of our environment, we could allow the members of a team to directly manage the transitions from a step of our method to another one. It would be also worth replacing the text-based chat of our communication/modeling environment with different synchronous communication methods (e.g., audio and video conferencing). Such a new environment could be subject of future empirical studies aimed at investigating whether the use of
different synchronous communication methods in the pair and square steps confirms or contradicts the results presented in this paper. On the base of the results of previously conducted empirical studies (e.g., [9, 10, 12]), we can postulate that the use of different synchronous communication methods could affect the time to produce models, while their quality not. Another future possible direction for our research could consist in comparing our integrated environment with the traditional think-pair-square method. This kind of investigation could be useful to better understand the penalty distribution brings. Finally, a possible future direction for our research could consist in turn our method from synchronous into asynchronous. In fact, each step can be scheduled on different time periods (provided that the think, pair, and square steps are sequentially accomplished). Empirical investigations could be conducted to study the effectiveness of a such a new solution.

8. Acknowledgments

The authors would like to thank the developers of the CoFFEE framework for the support provided. Special thanks are due to the Ana Portnova and the students that participated in the experiments.


[64] C. Damm, K. Hansen, M. Thomsen, Tool support for object-oriented cooperative design: Gesture-based modeling on an electronic whiteboard,


