Student Perceptions on the Benefits and Shortcomings of Distributed Pair Programming Assignments

Stelios Xinogalos, Maya Satratzemi, Alexander Chatzigeorgiou, Despina Tsompanoudi
Department of Applied Informatics, School of Information Sciences, University of Macedonia
Thessaloniki, Greece
{stelios, maya, achat, despinats}@uom.edu.gr

***Abstract*—Pair Programming (PP) has been extensively used for enhancing the learning of programming. Specifically, PP is considered to: make the learning of programming more pleasant, promote collaboration and communication between the members of pairs, encourage the sharing of knowledge and skills, and even improve code quality. More recently, systems have appeared that support Distributed Pair programming (DPP). DPP is considered to maintain all the benefits of PP and in addition to allow for the distributed collaboration of pairs from anywhere and at any time. However, DPP might impose limitations as well, such as the requirement from students to configure their systems and ensure a good Internet connection. In order to draw safer conclusions on the benefits and shortcomings of DPP and maximize its effects on the learning of programming, it is necessary to investigate its impact under real world situations. This research is twofold: the effect of DPP on student performance has to be thoroughly studied; student perceptions on the benefits and shortcomings of DPP have to be investigated in order to apply it in the best possible way. The study presented in this paper focuses on the latter issue. Specifically, student perceptions on DPP assignments carried out in the context of an Object-Oriented Programming (OOP) course based on Java throughout a whole semester are quantitatively and qualitatively analyzed. Based on this analysis some guidelines are presented for carrying out DPP assignments more effectively in the context of an OOP, Java-based course.**

Keywords—Ditributed Pair Programming (DPP); Distributed Pair Programming Systems; Object-Oriented Programming (OOP); group formation; Java assignments

I. INTRODUCTION

Pair Programming (PP) has its origins in software industry, where it was applied as part of Extreme Programming [1]. However, the benefits of PP were considered important for the teaching of programming as well. PP assists students in learning programming through communication and collaboration, easier correction of errors, as well as sharing of knowledge and skills ([2], [3]). In the context of PP the two members of a pair share the same computer for developing software. The members of the pair change frequently the roles of the "driver" that has possession of the keyboard and mouse and is writing the code, and the role of the "navigator" (or "observer") that constantly reviews the code and assists/guides the driver. The evolution of PP has resulted in Distributed Pair Programming (DPP) that actually gives the chance to apply PP remotely, which means that the members of a pair – or simply the partners – can collaborate from different locations as long as they both have an Internet connection.

DPP can be applied in educational settings using specially designed educational DPP systems. One such system is SCEPPSys [4] that consists of an Eclipse plugin used by students for applying DPP, as well as a web-authoring tool used by instructors for scripting DPP. Moreover, the system records a variety of information and reports several statistics both for pairs and students individually. SCEPPSys is being used for three years now for carrying out homework assignments in the context of an undergraduate Object-Oriented Programming (OOP) course based on Java. The research carried out so far in the context of this course has shown that DPP assignments can have a positive effect on student performance [5], while the data recorded by the system can assist instructors in monitoring the fulfillment of the course goals and the programming habits and progress of students [6]. The study presented in this paper aims to investigate student perceptions on the benefits and shortcomings of DPP assignments. It is our belief that having knowledge of student perceptions on DPP will help us detect potential problems and give us the chance to provide guidelines for applying DPP in the most effective way.

The rest of the paper is organized as follows. In Section II SCEPPSys is presented from the point view of instructors and students respectively. In Section III the main research questions and the methodology of the study are presented, while the results are analyzed in Section IV. This is followed by a discussion of the results and guidelines for applying DPP in the classroom based on the experience so far and student perceptions.

II. THE DPP SYSTEM OF SCEPPSYS

SCEPPSys is based on a typical client-server architecture and consists of:

- a *server* for dispatching messages between the clients

- a *database* for storing users' accounts, information about the courses and the groups of students, assignments, shared projects and statistics

- a *web-based authoring tool* used by instructors for scripting DPP and

- an *Eclipse plugin* installed by students.

In the next sections the process that an instructor applies for setting up a course, as well as a typical DPP session carried out by students are briefly described. More information can be found in [4].

A. *Preparing DPP Assignments*

The preparation of the DPP assignments is accomplished using the web-based authoring tool of SCEPPSys and includes:

Defining learning goals (e.g. constructor definition, object construction, inheritance) that will be used for characterizing the various tasks assigned to students.

Defining the collaboration script for each assignment and more specifically specifying:

- *participants* - students enrolled to the course

- groups or *pairs* - pairs can be formed randomly, with comparable skill or contribution levels, or freely.

- the problem solving *tasks* or *steps* that comprise the assignment. Each step is characterized by one of the learning goals (not visible to students) defined when setting up the course and has an accompanying hint that can be optionally consulted by students.

- *task distribution policies* - rotating role switching of driver/observer, balanced knowledge switching aiming at achieving symmetry in skill acquisition (learning goals) or free switching.

- *Scheduling each assignment*: updating the timetable with the start and end date of each assignment.

B. *Carrying out DPP Assignments*

A DPP assignment can be solved in several sessions before the deadline expires.

- A DPP session starts when the pair members meet online and request a pair programming session.

- A shared project is automatically generated inside the workspace of both students and the programming tasks are displayed in a separate area (Fig.1c).

- Students solve the tasks by adopting the *roles* of the driver and navigator (Fig. 1e) and switch roles according to the task distribution policy.

- Hints can be retrieved for each task that provide students help for completing the task.

- A text-based *chat* tool (Fig. 1b) can be used for communication and coordination purposes between the team members.

- Metrics, such as the driving time and *individual* participation rates are displayed throughout the session for helping the students balance their participation.

Students may submit the assignment on session close or continue the DPP session at another time.

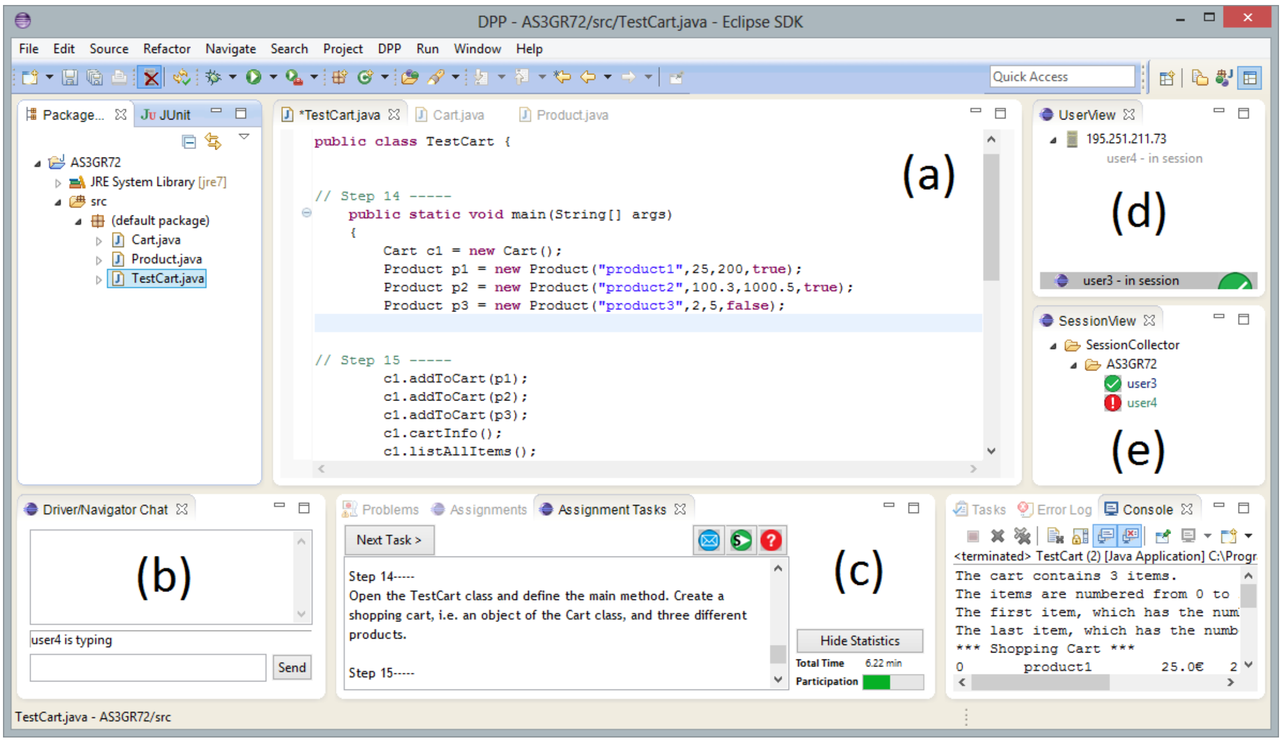

Fig. 1. SCEPPSys: (a) the shared editor, (b) the embedded chat tool, (c), script instructions for each step, (d) awareness indicators of user status, (e) roles.

III. Research Questions and Methodology

A. Context of the Study

The study presented in this paper was carried out in the context of an undergraduate course on Object-Oriented Programming (OOP) during the academic year 2015-16. Information for this course is presented in Table I.

In the context of the course students carried out six DPP assignments using SCEPPSys. The preparation of students and the realization of DPP assignments included the following steps:

- *Group formation*: students were informed for the DPP assignments at the beginning of the course and were asked to form groups freely using a Google form, providing amongst others a password for logging in the system. The instructors used the data from the form for creating students' accounts in the system (using a csv file).
- *Showcasing the system*: students were separated in three groups (with approximately 30 students each) keeping the pairs in the same group and were called in pairs in an extra one-hour lab. In this lab the process of collaboratively solving an assignment using SCEPPSys was presented to students. After this presentation students downloaded from the University Learning Management System (LMS) the SCEPPSys plugin for Eclipse and started solving the first assignment. Students were positioned in adjacent computers in order to be able to see their partner and realize how DPP works. The instructor observed students, answered questions and in case of situations considered by students as system problems showed them the appropriate way of using it.
- *Announcing assignments*: assignments were announced through email sent to the institutional student accounts through the LMS. Moreover, a schedule with information for the DPP assignments (deadline, OOP concepts required in the assignment, number of steps) was available throughout the semester in the LMS.
- *Support during problem solving*: a *hint* can be retrieved for every task through the system. Moreover, the *discussion forum* of the LMS is used for discussing problems that arise.
- *Submitting assignments*: the assignments were submitted directly from Eclipse when the assignment was completed.

Information on the assignments is presented in Table II.

After the end of the course students were asked to complete an online questionnaire regarding their experience on the DPP assignments, as well as potential problems that they encountered. Most of the questions were closed-type, but there was also an open-type question for comments. The questionnaires were named in order to be able to draw conclusions for the pairs of students. Fifty seven out of the 94 students (61%) that participated in the DPP assignments filled in the questionnaire.

B. Research Questions

This study aimed to record students' perceptions on the following issues:

- How do students evaluate the experience on DPP assignments?
- Does free selection of partners by students themselves lead to effective group formation?
- What are students' perceptions on the benefits of DPP assignments?
- What factors hinder student collaboration and experience on DPP assignments?

TABLE I. Course Outline

Department	Applied Informatics
Course	Object-Oriented Programming
Semester	3rd
Programming language	Java
Syllabus	▪ Objects and classes (necessity of using classes) ▪ Class definition (fields, constructors, methods) ▪ Constructing objects and calling methods (main) ▪ Class associations ▪ Groups of objects (array, ArrayList) ▪ Inheritance, polymorphism and overriding ▪ Abstract classes and interfaces ▪ Graphical User Interface (constructing a simple GUI, event handling, interaction with domain classes) ▪ Collection framework of Java ▪ Manipulation of text and binary files
Duration	13 weeks, 3 hours per week
Teaching approach	▪ 3 hour lab session every week ▪ OOP concepts are approached through hands on exercises at lab ▪ BlueJ is used for presenting the structure (simplified UML class diagram) of projects ▪ Eclipse is used for programming exercises at labs and assignments ▪ New OOP concepts are presented in the context of extending projects developed in previous lab sessions

TABLE II. DPP Assignments

Academic year	2015-16
Participants	94 (47 groups)
Prior programming knowledge	1st semester "Procedural Programming" course based on C
Prior experience on DPP	none
Group formation	Free selection of partner
DPP system	SCEPPSys
Assignments	1. Class definition, main 2. Class associations 3. Object collections – ArrayList 4. Inheritance & polymorphism 5. GUI, event handling (+inheritance) 6. Binary files (+inheritance, ArrayList, Comparator)
Deadline for each assignment	Approximately 10 days

IV. Results

In this section the results of analyzing students' replies on the questionnaire are presented using descriptive statistics.

A. Overall Experience

In order to investigate students' perceptions on DPP as an overall experience in general and in the context of DPP assignments more specifically, students were posed with two relevant questions that are analyzed in the following paragraphs both quantitatively and qualitatevely.

***Q1**. How would you evaluate the distributed, collaborative solution of assignments as an overall experience?*

As presented in Fig. 2, the majority of students (83%) evaluated the overall experience in distributed and collaborative solution of assignments as a good (50%) or very good experience (33%). However, 10% of the students (6 replies) evaluated negatively the overall experience with DDP assignments.

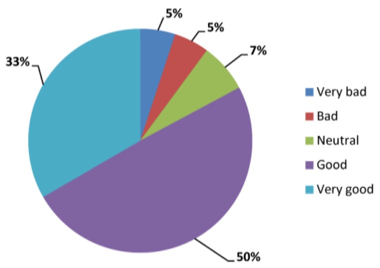

Fig. 2. Overall experience with DPP assignments.

In order to study if students' evaluation of their overall experience with DDP assignments is affected by their performance in the course, a Spearman's rank-order correlation analysis was carried out. In this statistical test the final marks of the 52 students that took part in the final exams (out of the 57 that filled in the questionnaire) were analyzed. The distribution of students' replies in Q1 for every mark in the scale of 1 to 10 is presented in Fig. 3. The results of the Spearman rank-order correlation suggest that there is no association between the final mark in Java and how students evaluated the DPP experience (ρ<.730)

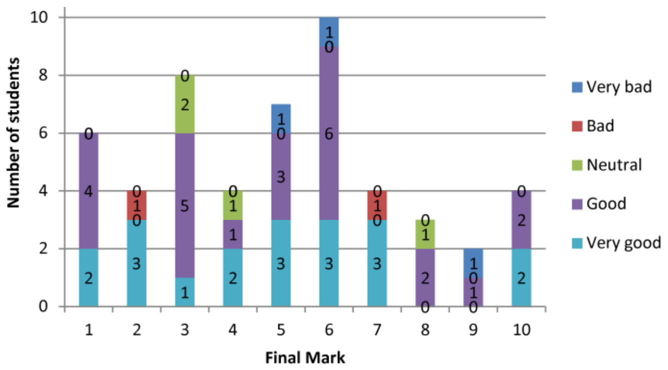

Fig. 3. Students' final marks in relation to their replies in Q1.

***Q2**. Based on your experience in DPP would you prefer to work individually or collaboratively in programming assignments?*

Based on their experience in DPP in the context of the OOP course and working individually in programming assignments in the previous Procedural Programming course, the majority of students (77%) stated that would prefer to work collaboratively, as shown in Fig. 4.

The qualitative analysis of students' replies in Q1 and Q2 in combination gives some interesting results.

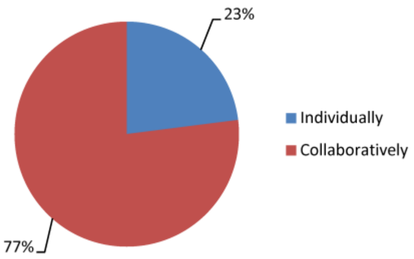

Fig. 4. Preferred mode of carrying out programming assignments.

Out of the 6 students that evaluated their experience with the DPP assignments as bad or very bad:

- 2 students that reported having a bad experience, worked in pairs where both students had failed the introductory programming course or one of them had failed the course and the other had just passed with the minimum passing grade (i.e. five out of ten). One of these students stated that does not believe in the benefits of DPP, while technical problems were mentioned as well. Specific technical problems mentioned by students are presented in the context of Q7 and refer mainly to the responsiveness of the system.

- 1 student reporting a bad experience participated in a pair where both students had good programming skills and mentioned coordination problems with the partner, as well as technical problems.

- 1 student reporting a very bad experience had failed in the prior introductory course, while his/her partner had good prior programming experience. The student mentioned unconformity and coordination problems with the partner.

- 2 students that formed a pair and both had passed the introductory programming course with 7 (out of 10) reported a very bad experience with DPP and as in previous cases one of them reported coordination and technical problems.

Concluding, it seems that students' bad experience was due to *coordination problems* with their partners, as well as *technical problems*. Technical problems are an important issue in DPP and distributed learning in general. Based on the experience of the last three years, the employed distributed tools:

- can be debugged and become more stable through usage and evaluation under real circumstances

- can be improved by identifying bad practices of usage (especially during a controlled usage of the tool in the labs at the beginning of the course) and practical guidelines for avoiding common problems can be given to students
- require appropriate hardware for hosting the tools.

However, there are factors regarding the technical problems that cannot be easily dealt with. These include configuring appropriately students' hardware and software that is usually used for DPP assignments and ensuring a stable Internet connection.

Regarding coordination problems it has become clear to us that several students have difficulty in agreeing on a common time for working on DPP assignments. It is important for instructors to:

- know that more time should be given to students for the collaborative and distributed solution of assignments compared to the individual solution in order to reduce the aforementioned problem
- in the case that students are left free to form groups, they should be consulted to take into account not only friendship relations or/and prior level of programming skills, but also their schedules and academic and extracurricular activities.

It was quite surprising, however, that out of the 6 students that mentioned coordination and technical problems and consequently evaluated negatively the overall experience in the distributed and collaborative solution of the assignments, only 1 student stated that would prefer to work individually in programming assignments in the context of Q1. It seems that although some students faced problems during DPP assignments, they still prefer to work collaboratively. Maybe the students themselves understand that the reported negative experience was due to the aforementioned reasons that could be dealt with.

Out of the 13 students that stated in Q2 that would prefer to work individually, just 1 student evaluated the experience with DPP as bad and mentioned connection problems and 2 students evaluated the experience as neutral. It is clear that for some students (10 students – 18%) working individually in programming assignments is preferable, despite their positive experience on DPP. Maybe giving these students some information on the importance of *agile software development techniques* in the software industry would make them see from a different viewpoint DPP assignments.

B. Group Formation

Based on literature findings group formation is considered to be a very important factor that affects the effectiveness of pair programming (PP), and consequently DPP as well. Pairs can be defined by the instructor or students themselves. In the former case, group formation can be accomplished based on students' programming skill level, their personality, or even randomly. The *personality* of partners has not been proven to have an effect on PP [7]. Toll et al. in [8] concluded that the outcomes of PP are better when the *skills* of the one partner are slightly better or worse than those of the other partner. In cases where there is a big difference in the partners' skills then their matching ceases to be ideal and consequently it is not effective. Katira et al. [9], on the other hand, concluded that pairs are more compatible if students with similar *perceived technical competence* are grouped together. However, this perception cannot be predicted and pairs cannot be formed based on this factor. Regarding the actual skill level and pair compatibility, no correlation was found for the CS1 (freshmen) and SE (advanced undergraduate) students, while a strong positive correlation was found for the graduate OO students. The personality seems to have an effect only on the compatibility of CS1 pair programmers. Katira et al. conclude that "*pairs will be highly compatible and successful if we pair them randomly, without necessarily considering personality type, skill level, or self-esteem*" ([9], p.11). Jacobson and Schaefer [10] based on their experience from a CS1 course indicated that a very high rate of compatible pairings can be accomplished by having students choose their partner. In the CS1 course where the study took place, less than 5% of pairs had compatibility problems, as reported by a member of a pair, observation from teaching assistants or complaints from partners to instructors/teaching assistants. According to the authors' beliefs students seek to find a partner that they perceive to have a skill level at least as high as their own. This is in accordance with Katira et al. [9] who state that students prefer to pair program with a student that they perceive to be of similar technical competence. Williams et al. [11] also concluded that pairs are more compatible if students with similar perceived skill level are grouped together.

In our study students were left free to choose their own partner, without giving them any hints and/or guidelines regarding group formation. In order to study students' selection criteria (Q3) and satisfaction with their partner selection (Q4) students were asked to reply to the following two questions.

Q3. *What was the main selection criterion of your partner?*
Being a friend
Having the same level of programming knowledge with me
Other (please specify):

The main criterion for selecting a partner was *friendship relations* for 87% of the students, while 11% of them selected a partner that was perceived to have the same *level of programming knowledge* (Fig. 5).

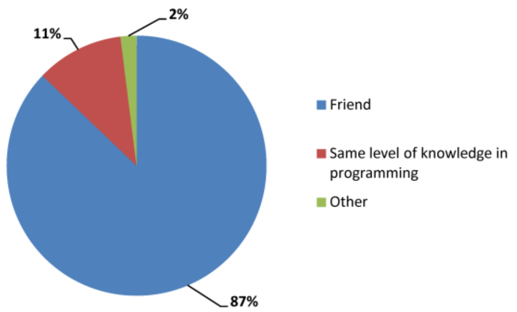

Fig. 5. Selection criteria for partners.

One of the students mentioned that just wanted to have a new experience and found a student that was eager to participate as well. Our results are in contrast with Katira et al. [9] who stated that students prefer to pair program with partners that they perceive to have similar technical competence. In our study, the vast majority of students mentioned friendship relationships as the main selection criterion.

Q4. *Were you satisfied with the selection of your partner?*

Q4 was a closed-type "yes/no" question, but students had the chance to specify collaboration problems with their partner in the context of the closed-type question Q6, as well as an open question for comments regarding any aspect of the DPP assignments. Just 4 (7%) out of the 57 students that filled in the questionnaire stated that were not satisfied with their partner. This result is in accordance with results of Jacobson and Schaefer [10] who found that when students choose their partner, less than 5% of the students have compatibility problems. In all the problematic cases in our study the partner was selected based on a friendship relation. The members of the pairs that these students participated in had both the similar programming skills and more specifically 3 of them had low and 1 high skills in programming. The most serious problem mentioned by three of the students was the lack of knowledge from their partner, in spite of the fact that all partners had similar skills in programming. Two students mentioned that had problems in agreeing when to collaborate, while one partner was supposed to be unreliable.

C. Perceived Benefits of DPP Assignments

The benefits of PP have been heavily studied in the literature, in contrast with DPP. However, it is considered that the benefits of PP apply to DPP as well. One of the most referenced works on the benefits and costs of PP is that of Cockburn & Williams [2]. Based on this study some significant benefits of PP are the following:

- *detection of errors* during coding
- better *program design* and shorter code length
- faster *solution of problems*
- *enhanced learning* both for the system and software development
- acquiring *communication* and *collaboration skills*
- pairs *enjoy* programming more.

da Silva Estácio and Prikladnicki [12] in a recent systematic literature review on DPP published in 2015 recorded – amongst other – the effects of DPP on various variables concerning both DPP practice and DPP for teaching programming. The main results regarding DPP practice can be summarized as follows:

- the effects on *code quality* are mixed, with two studies reporting a negative and two studies reporting a positive effect
- a positive effect on *knowledge* was recorded (1 study)
- DPP does not have an effect (1 study) or has a positive effect (1 study) on *productivity*
- there is a positive effect on *communication* (1 study)
- there is a negative effect on students' *effort* (1 study)

Regarding the use of DPP for teaching programming, mixed results were recorded regarding students' performance, while a positive effect was recorded for *grades*, *productivity*, *motivation*, *confidence* and *learning*.

In our study, we recorded students' perceptions on several of the aforementioned variables in the context of the following closed-type question:

Q5*.At what degree do you agree that you earned the following benefits from DPP?*
(1=totally disagree, 2=disagree, 3=neutral, 4=agree, 5=totally agree)

Student replies in Q5 regarding the various benefits of DPP are summarized in Table III.

TABLE III. THE BENEFITS OF DPP

	Perceived benefit	**Mean**	**St.Dev**
TIII.1	Sharing knowledge and skills with my partner	3.94	0.87
TIII.2	Quicker correction of logic and syntax errors	**4.08**	0.91
TIII.3	Less time for completing an assignment	3.6	0.92
TIII.4	DPP assisted me in learning programming	3.91	0.91
TIII.5	Learning programming was more pleasant	**4.31**	0.82
TIII.6	Most questions were answered through conversation with my partner	3.94	0.95
TIII.7	I was more confident for the correctness of my solutions	3.82	0.92
TIII.8	Feeling of responsibility for my participation in the assignments	**4.15**	0.84
TIII.9	It forced me to solve more assignments than I would if assignments were solved individually	3.24	1.47
TIII.10	DPP helped me improve the quality of my code	3.82	1.01

The three most prominent benefits of DPP based on students lie in the fact that the DPP assignments:

- made the learning of programming more *pleasant* (TIII.5: mean=4.31, st.dev=0.82) and, moreover, assisted them in learning programming (TIII.4: mean=3.91, st.dev=0.91)
- gave students a feeling of *responsibility* for participating in the assignments (TIII.8: mean=4.15, st.dev= 0.84), without having the feeling of being "forced to solve" more assignments than they would if

they were working individually (TIII.9: mean=3.24, st.dev=1.47)

- helped them in *correcting logic and syntax errors* quicker (TIII.2: mean=4.08, st.dev=0.91).

The majority of students also reported the following benefits:

- the *sharing of knowledge and skills* with the partner, as well as the fact that *most questions were answered through conversation* with the partner with the same mean value (TIII.1 & TIII.6: mean=3.94). These results denote the enhancement of collaboration and communication skills that are considered extremely important in agile software development techniques.

- students are more *confident* for the correctness of their solution and in addition they believe that they write better *quality* code (TIII.7 & TIII.10: mean=3.82). Generally, students are more confident for their work when working in pairs.

It is clear that these results confirm the results of related work on PP and DPP briefly presented at the beginning of the subsection.

D. Perceived Shortcomings of DPP Assignments

In order to detect factors that hinder students' experience on DPP assignments, which is an issue not adequately covered in the literature, as far as we know, the following questions were used.

Q6*. What factors hindered the collaboration and the experience on DPP?*
Coordination problems (collaboration time)
Unreliable partner
Lack of partner knowledge
Dominating role of partner
Technical problems
Difficulty in using the plugin
(1=very much, 2=much, 3=averagely, 4=a little, 5=not at all)

TABLE IV. FACTORS THAT HINDER DPP

	Mean	Mean	St.Dev
TIV.1	Coordination problems (collaboration time)	**3.68**	1.16
TIV.2	Unreliable partner	4.61	0.81
TIV.3	Lack of partner knowledge	4.03	1.22
TIV.4	Dominating role of partner	4.63	0.83
TIV.5	Technical problems	**3.03**	1.06
TIV.6	Difficulty in using the plugin	4.05	0.93

The results confirm the results recorded in previous questions. The most prominent factor hindering collaboration and DPP experience is *technical problems* (TIV.5: mean=3.03, st.dev=1.06) that affected negatively one third of students much (Fig.6: 32%) and another one third averagely (Fig.6: 35%). The second most prominent problem was *coordination on a common time to collaborate* (TIV.1: mean=3.68, st.dev=1.16) that affected much (Fig.6: 19%) or averagely (Fig.6:19%) one fifth of students in each case. A detailed analysis of these factors is presented in subsection IV.A where the results of students' overall experience are analyzed.

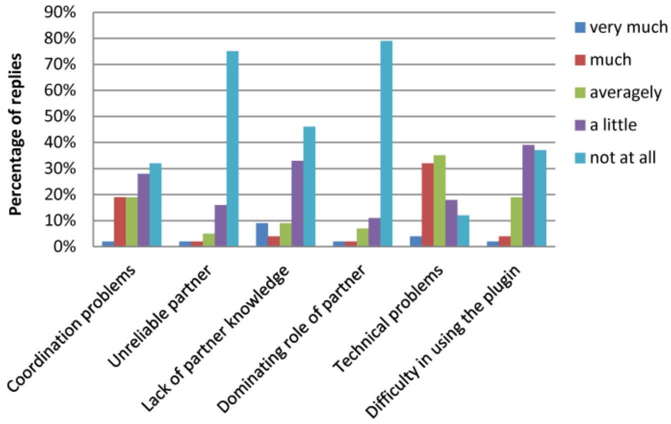

Fig. 6. Factors hindering collaboration and DPP experience.

In order to further investigate potential technical problems students were posed with the following open-type question.

Q7*. What were the main technical problems that you faced during DPP?*

Twenty-five out of the fifty-seven respondents (44%) reported technical problems. The two main problems reported were the following:

- *Responsiveness problems*: eleven students reported problems such as: "*Eclipse did not respond while writing code*", "*auto saving lasted long*", *"session closed without my command".*

- *Tasks were not visible*: fourteen students mentioned that "*at the beginning of the assignments the tasks were loaded for the one of the users (the one that made the invitation)*".

Regarding the technical problems it is clear that they could be caused due to the infrastructure used by students and their Internet connection. However, it came clear during the semester - from discussions in the forum and personal communication with students that faced problems - that the server hosting the system needed to be upgraded as well. This happened and currently (middle of the third semester we are using the system) such problems have not been recorded, or at least have not been reported from students.

Regarding the second problem, it came out that this malfunction was due to the fact that the system demanded both members of a pair to login using the "DPP menu" within Eclipse and select to "Load assignments" for opening a tab were students could see all the assignments submitted and invite their partner for collaboration on the new assignment. Usually, only the partner that sent the invitation for collaboration "Loaded assignments" and this resulted in not showing the tasks to the other partner. This was more usual for pairs that did not attend the showcasing of the system.

V. Discussion and Conclusions

Teaching and learning Object-Oriented Design and Programming is challenging for instructors and students respectively. Some students face difficulties even in comprehending and differentiating between the main OO concepts of “objects” and “classes” [13]. An important aspect for any OOP course is the assignments and/or projects that give students the chance and motivation to practice and comprehend OOP concepts and constructs. In order to further support students through collaboration and sharing of knowledge and skills, we are using the last three years DPP Java assignments. The assignments are carried out with SCEPPSys, an educational DPP system that utilizes collaboration scripts to enhance the benefits of PP. In this paper we focused on analyzing student perceptions on the DPP assignments. Data was collected using an on-line questionnaire completed at the end of the second year (academic year 2015-16) of using DPP assignments.

The results of the study confirmed most of the benefits recorded in the literature for PP and at a much lesser degree for DPP. Of course, in most cases PP is applied during lab sessions, while in our case DPP was applied in the context of assignments and this means that students had to use and configure appropriately their own infrastructure.

The majority of students evaluated positively the overall experience on DPP assignments (83%) and stated that would prefer to carry out assignments collaboratively (77%). The most important benefits according to students are that: learning programming is more pleasant; students feel more responsible for participating in the assignments; and correcting syntax and logic errors is easier. Sharing of knowledge and skills, being more confident for their solution and better code quality were also considered important benefits.

On the other hand, the most important shortcomings are the technical problems that hinder averagely the collaboration and experience on DPP, and the coordination problems of the partners (agreeing on the time to collaborate) that hinder to a small extent the collaboration. Technical problems can be due to the institutions’ and students’ infrastructure, but also due to unexpected use of the system and can be dealt with using the gathered experience of instructors on using DPP assignments under real-world situations.

It is important to mention that the positive attitude of students towards DPP assignments was recorded for pairs of students that were freely formed by students themselves. In most cases the pairs were formed based on friendship relationships (87%) and for a few pairs (11%) on the perceived programming skills level. This might have played an important role in the positive student experience recorded and instructors should consider this way of group formation.

Based on the gathered experience so far and student perceptions, some practical guidelines that could help instructors that consider applying DPP assignments can be proposed:

- *Group formation*: having students form pairs on their own seems to lead to pairs that collaborate effectively. However, it is really important to consult students to take into account each one’s schedules and whether they both have slots for collaboration before forming groups.
- *Showcasing the DPP system*: it is highly recommended to instructors to present students how to carry out a typical DPP session. It is important to have students experiment with the system in a lab session with a test or even the first DPP assignment. The members of each pair can work on adjacent computers in order to be able to see the screen of their partner and realize what the driver and the navigator actions result to. This can help to apply DPP properly and avoid problems that arise from improper usage of the DPP system and are perceived by students as technical problems. Student problems and questions can be resolved in the best possible way by the instructor during such a lab session.
- *Providing support*: besides the common user manual, preparing a video showcasing a typical DPP session that could be accessed at any time is also useful. Using some sort of discussion forum is also useful for reporting and discussing various problems that can arise during problem solving, although most students seem to prefer personal contact with the instructor. Instructors should study frequently the data recorded and the statistics reported through the web-authoring tool. This can help monitor student progress and detect potential problems [6] that sometimes are not, unfortunately, reported by students.
- *Duration of DPP assignments*: although the DPP assignments can be solved at any time, it seems that some pairs have difficulties in finding the appropriate time for collaborating. Instructors should have this in mind and provide more time for DPP assignments than for individual assignments, although the former are considered easier to be completed because of knowledge and skill sharing and collaboration.

References

[1] Laurie Williams, Robert R. Kessler, Ward Cunningham, and Ron Jeffries. 2000. Strengthening the Case for Pair Programming. *IEEE Softw*. 17, 4 (July 2000), 19-25.

[2] Cockburn, A., & Williams, L. (2000). The costs and benefits of pair programming. *Extreme programming examined*, 223-247.

[3] da Silva Estácio, B. J., & Prikladnicki, R. (2015). Distributed pair programming: A systematic literature review. *Information and Software Technology*, 63, 1-10.

[4] Tsompanoudi, D., Satratzemi, M., Xinogalos, S. (2015). Distributed Pair Programming using Collaboration Scripts: An Educational System and initial Results. *Informatics in Education*. Vol. 14, No 2, 291-314.

[5] Tsompanoudi, D., Satratzemi, M., Xinogalos, S. (2016). Evaluating the effects of scripted Distributed Pair Programming on students performance and participation. *IEEE Transactions on Education*. Volume 59, Number 1, 24-31, DOI: 10.1109/TE.2015.2419192.

[6] Xinogalos, S., Satratzemi, M., Tsompanoudi, D. and Chatzigeorgiou, A. (2016). Monitoring an OOP Course Through Assignments in a Distributed Pair Programming System. In: Z. Budimac, Z. Horvath, T. Kozsik (eds.): *Proceedings of the SQAMIA 2016: 5th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications*, Budapest, Hungary, 29.-31.08.2016, Vol. 1677, ISSN: 1613-0073, http://ceur-ws.org/Vol-1677/, 97-104.

[7] Hannay, J. E., Arisholm, E., Engvik, H., & Sjoberg, D. I. (2010). Effects of personality on pair programming. *IEEE Transactions on Software Engineering*,*36*(1), 61-80.

[8] Van Toll, T., Lee, R., & Ahlswede, T. Evaluating the Usefulness of Pair Programming in a Classroom Setting. *In 6th IEEE/ACIS International Conference on Computer and Information Science* (ICIS 2007).

[9] Katira, N., Williams, L., Wiebe, E., Miller, C., Balik, S., & Gehringer, E. (2004, March). On understanding compatibility of student pair programmers. In *ACM SIGCSE Bulletin* (Vol. 36, No. 1, pp. 7-11). ACM.

[10] Jacobson, N., & Schaefer, S. K. (2008). Pair programming in CS1: overcoming objections to its adoption. *ACM SIGCSE Bulletin*, *40*(2), 93-96.

[11] Williams, L., Layman, L., Osborne, J., & Katira, N. (2006, July). Examining the compatibility of student pair programmers. In *AGILE 2006 (AGILE'06)* (pp. 10-pp). IEEE.

[12] Bernardo José da Silva Estácio and Rafael Prikladnicki (2015). Distributed Pair Programming: A Systematic Literature Review. *Information & Software Technology* 63: 1-10.

[13] Xinogalos, S. (2015). Object Oriented Design and Programming: an Investigation of Novices' Conceptions on Objects and Classes. *ACM Transactions on Computing Education*, Vol. 15, Issue 3, Article 13 (September 2015), 21 pages. DOI=10.1145/2700519 http://doi.acm.org/10.1145/2700519