A Systematic Review of Pair Programming Research – Initial Results

Norsaremah Salleh
Department of Computer Science
University of Auckland
nsal017@ec.auckland.ac.nz

ABSTRACT
This paper presents the initial results of our systematic review of pair programming studies. As an alternative to performing a literature review, we conducted a systematic review in order to reveal the answers to our research questions pertaining to the issues of pair programming in an educational context. Our focus is to identify factors affecting the effectiveness of students who pair programmed, particularly related to the psychosocial factors such as compatibility, personality and gender issues. Our systematic review follows the phases described in the procedures for performing systematic reviews outlined in [18]. We have included 66 studies for the synthesis of evidence, of which 10 studies were found relevant to answer the first research question. The findings showed that personality type is the most common factor investigated. However, the results of those studies were somewhat inconclusive.

Categories and Subject Descriptors
k.3.2 [Computer and Information Science Education]: Computer Science Education

General Terms: Management, Human Factors

Keywords: pair programming, systematic review, empirical studies

1. INTRODUCTION
The term “pair programming” was first used in 1999 as one of the core practices in the Extreme Programming (XP) software development methodology in industry. As defined by Williams et al. [37] pair programming refers to a practice in which two programmers sitting side by side using only one computer to work collaboratively on the design, algorithm, code or test. One of them is the “driver”, who is responsible for typing the code and has control over the resources such as computer, mouse and keyboard, whereas the partner known as “navigator” or “observer” has the responsibility of observing how the driver works. In every single task, the navigator is expected to detect errors made by the driver and offer ideas in solving the problem.

The practice of pair programming has been widely implemented in the industry as well as in educational settings (Domino et al. [8], Chong et al. [6]). A vast amount of research on pair programming has been conducted to observe the benefits of the technique and to understand how the practice can improve students’ learning outcomes. The stated benefits of pair programming were identified as follows:

- Improvement in students’ academic performance such as in final and mid term exams, quizzes, programming assignments and overall course grades [22, 23, 24, 28, 25, 26].
- Improvement in programming productivity in terms of the time spent on coding and quality of the software produced [29, 37, 11, 13, 21].
- Increase in students’ retention rate and course completion rate [23, 24, 28].
- Increased students’ confidence level and enjoyment in learning programming [36, 23, 24, 25, 26].
- Reduced staff workload [38, 7].
- Increased efficiency in helping female students to work in programming tasks [2, 35, 14].

In this paper, we describe the technique that we have adopted in conducting the literature review of pair programming research in an academic setting. We have employed the systematic review approach to systematically and comprehensively assess the relevant literature or studies in pair programming following the guidelines in [18]. The result of this review can be used to understand the current “state of the art” of research in pair programming when applied to an educational context.

The remaining of this paper is organized as follows: Section 2 describes the background of pair programming followed by an explanation of the systematic review’s approach in Section 3. Section 4 reports our initial results and finally we conclude and suggest our future work in Section 5.

2. BACKGROUND ON PAIR PROGRAMMING
The success stories of pair programming in the industry have been reported by Haungs [12], where the practice has been successfully implemented in the C3 projects payroll computations for hundred of thousands of employees. As a result of its successful implementation, pair programming was reported to efficiently increase the improvement of programming productivity and also increase the knowledge transfer among software developers.

A study by Nosek [29], which involved 15 programmers, showed that all groups (paired teams) outperformed individual programmers. In his study, pair programmers produced a more readable and functional code compared to solo programmers. They also found that pair programmers enjoyed more the collaborative problem solving and were more confident in their solutions. The benefits of pair programming in industry motivated many academics and researchers to apply the technique in higher educational institutions particularly in the field of Software Engineering (SE) and Computer Science (CS).
Anecdotal and empirical evidence shows that the pair programming technique has brought significant contributions to the performance of learning introductory programming courses in CS curriculum [7]. In 1999, Williams et al. [36] conducted a study among senior Software Engineering students to validate the effectiveness of pair programming. In terms of the time spent on programming, initially the pair programmers took 60 more hours to complete the assignment, but it reduced by 15% after the adjustment period (i.e. “pair jelling”). They found that, compared to solo students, paired students completed their assignments 40% to 50% faster. In terms of students’ satisfaction, many studies reported that the majority of students enjoyed the experience of using pair programming and showed positive attitudes towards the practice [36, 9, 23, 34, 25, 26].

Despite the importance of studying the effectiveness or impact of pair programming practice towards educational benefits, investigating factors that contribute to the success of pairing work has never been neglected. In [5], Chaparro et al. found that matching skill levels and programming tasks have a strong influence on pairs’ collaboration. In another study, Katira et al. [15] suggested that compatibility of pair programmers has a significant impact on work productivity. Her team has identified personality type, actual skill level, perceived technical competence and self esteem as potential factors that determine the compatibility of pair programmers. Their studies on 564 undergraduate and graduate students showed that pair programmers were highly compatible and successful if paired randomly, without necessarily considering personality type, skill level or self-esteem. The most significant finding from their study was that students preferred to pair with someone they perceived to be of similar technical competence, regardless of their personality type [15, 40].

Research results, however, are found to be contradictory as highlighted by Gallis et al. [10]. For example, a study by Freeman et al. [9], found that there was no significant difference between pair and solo students in terms of student’s academic performances (quizzes, final exams, and course grades) and the time spent on the project. However, the findings in McDowell et al. [22], showed that the performance of paired students was significantly higher than the solo students both in the final exams and course grades. Canfora et al. [3] also reported a decrease in time spent on programming for pair programmers instead of solo programmers, thus refuting the claim made by Freeman.

Considering the importance of assessing the pair programming research, we performed the systematic review in order to comprehensively review all related studies. Our major focus is to review all empirical studies within the domain of CS/SE teaching and learning for the purpose of gaining understanding on the social aspects of pair programming such as gender issues, personality type and compatibility and other relevant factors.

3. SYSTEMATIC REVIEW METHOD

3.1 Introduction

A systematic review refers to a process of identifying, assessing and interpreting all available research evidence with the purpose to provide answers to a particular research question [18]. It provides means or ways to conduct a literature review using extensive and comprehensive strategies based on several defined stages (see Figure 1). Our systematic review followed the steps outline below, some of which with iteration:

- Identification of the need of the review and develop the review protocol
- Formulation of the research questions
- Identification of relevant literature by conducting a comprehensive and exhaustive search
- Selection of primary studies based on inclusive/exclusive criteria
- Data extraction together with the quality assessment
- Synthesis of evidence
- Interpretation of results and report writing

The development of a review protocol is important as a framework for the systematic review. It describes the strategies to be executed when the real conduct of review is carried out. Most importantly, the protocol defines the research questions to be answered towards the end of the review, describes how the literature will be searched and stored and the methods to synthesize the evidence in order to answer the research questions [18].

3.2 Research Questions

Within the perspective of our systematic review, our focus is to understand the factors that influence the effectiveness of pair programming practice within an educational setting, particularly in relation to the behavioral aspects. As suggested by Petticrew & Robert [30], the formulation of research questions which aim to answer a research question about effectiveness, should comprise of 5 elements known as PICOC: Population, intervention, comparison, outcomes and context.
The population comprises specific target groups of people that the study will cover. In our systematic review, our population of interest consists of both undergraduate and graduate Computer Science and Software Engineering students in higher educational institutions. Thus, we excluded those studies focusing on industrial contexts. Table 1 shows the PICOC structure of our questions.

<table>
<thead>
<tr>
<th>Population</th>
<th>CS/SE students</th>
</tr>
</thead>
</table>
| Intervention & Comparison | Pair programming, compatibility factor (intervention)  
|                     | Solo programming (comparison) |
| Outcomes            | Measurement metrics of pair programming effectiveness and measurement metrics of quality |
| Study Design        | Review(s) of all empirical studies |
| Context             | Within the domain of CS/SE education |

Our systematic review intends to answer the following research questions:

**Question 1:** What evidence is there about compatibility factors that affect the effectiveness of pair programming?

**Question 2:** What are the metrics that have been used to measure the effectiveness of pair programming?

**Question 3:** How is quality measured in the studies that used quality as a measure of effectiveness?

**Question 4:** Which pairing configurations are considered as most effective looking at the compatibility factors obtained from Question 1?

### 3.3 Identifying Relevant Literature

The first task in identifying the literature is to develop the search strings that will be used to search for the relevant articles. The strategy we used to construct the search terms is as follows:

- a) Derive major terms used in the review questions (i.e. based on the population, intervention, comparison, outcome and context)
- b) List down keywords mentioned in the articles.
- c) Search for the synonyms or alternative words. We have also consulted our subject librarian to seek further advice in the proper use of the terms.
- d) Use the Boolean OR to incorporate alternative spellings and synonyms
- e) Use the Boolean AND to link the major terms from population, intervention and outcome.

The complete search string we initially used for the searching of online literature is as follows:

(student* OR undergraduate) AND (pair programming OR pair-programming) AND (compatibility OR Personality type OR Ethnic OR Self-esteem OR Confidence OR Gender OR skill*) AND (Experiment* OR Measurement OR evaluation OR assessment) AND (Effectiveness OR efficient OR successful) AND (Quality OR excellence)

The identification of primary sources from online databases, journals, conferences and grey literature is important to ensure the wide coverage of potential sources. The searching of literature covered the study published within the period of ten years between 1997 and 2007.

The initial phase of our search process involved the use of the following online databases to search for candidate primary sources:

- ACM Digital library
- Current Contents
- EBSCOhost
- IEEEExplore
- ISI Web of Science
- INSPEC
- ISI Proceedings
- ProQuest
- Sage Full Text Collections
- ScienceDirect
- SpringerLink
- Scopus.

In our preliminary search, we retrieved a very small number of articles when using the complete search string defined in the protocol. For instance, when we used IEEEExplore, Inspec, and ProQuest, each retrieved only one article. On the other hand, we retrieved in total 320 relevant articles of pair programming when the search string was more general. In particular, we ran the search using the keyword “pair programming” OR “pair-programming” and our systematic review relied on the results of these general search strings. In relation to this, we also sought the opinion from our subject librarian regarding the appropriate use of the search string and we received the advice that more results would appear if the strings defined were a bit looser rather than strictly adhered to search term defined in the protocol. Our experience supports the suggestion made by Kitchenham et al. [18], which states that “the simpler search string might have been just as effective”.

The results of our search are stored using the EndNote software. After screening the titles and abstracts, and excluding all irrelevant and duplicate studies, finally we selected 66 studies to be included in the review. We discuss our selection criteria of the studies in the following section.

### 3.4 Selection of Primary Studies

As inclusion criteria, we only considered empirical studies of pair programming that involved students as the population of interest. Therefore, other types of studies that involve scientific or engineering methods, such as simulation studies or generations of economics and mathematics models were excluded. We excluded the following studies from our selection:

- Papers presenting the opinion of the researcher(s) without evidence to support the claim.
- Papers about Agile/XP describing development practices other than pair programming such as test-first programming, refactoring etc.
- Papers that described tools (software or hardware) to support the pair programming practice.
- Papers not written in English.
3.5 Quality Assessment and Data Extraction
In assessing the quality of studies, we developed a checklist outlining the major important quality criteria expected from the primary study. We built the list based on the quality criteria adapted from Leedy et al. [20], Kitchenhem et al. [18], and Spencer [33].

Based on the studies included in the review, data extraction was carried out on all papers that passed the screening process. The data from each of the studies were recorded using a data extraction form and saved in an individual word document file. For validation purposes, 15 of the 66 primary studies were randomly selected and had their data extracted by both authors. The extracted data was later compared in a review meeting attended by both authors and any disagreements were resolved during the meeting.

4 INITIAL RESULTS
In this Section we present the initial outcomes and analysis of the systematic review. Our intention is to synthesize the evidence based on the data extracted described in the previous Section. The synthesis aims to answer the research questions stated in Section 3.2.

The process of identifying relevant literature produces 320 empirical studies, of which only 140 studies were found to be potentially relevant and useful (See Figure 2). However, after the screening of titles and abstracts, and removing all duplicate studies, only 66 studies1 were included for the synthesis of evidence. This figure represents 49% of all primary studies included based on screening of titles and abstracts. There were 64 empirical studies conducted in an academic setting and 2 studies were done in a mixed setting (academic and industry).

![Figure 2: Identifying Relevant Literature](http://www.cs.auckland.ac.nz/~emilia/esem08/)

Our analysis on the type of research approach used in these studies is shown in Figure 3. The classification of research approach was done based on classification used by Wohlin et al. [41]. Formal experiments were found to be the most popular research approach used, when compared to other approaches such as Surveys, Case Studies, Mixed-methods, and pure qualitative studies.

In this paper, we will only report the results of the first research question. The synthesis of the other research questions (i.e. Question 2, 3 and 4) is still in progress.

Question 1: “What evidence is there about compatibility factors that affect the effectiveness of pair programming?”

The compatibility factors are factors that are believed to influence the compatibility of students when programming in pairs. These include psychosocial factors, communication, and education background of the students who pair programmed. Table 2 shows the summary of evidence found from the review of studies investigating pair compatibility.

From our analysis, we found only three studies [15, 16, 40] that specifically addressed the compatibility of pair programmers as part of their research questions. The results of compatibility are found to vary among different treatment groups in the study conducted by Williams et al. [40]. For instance, their experiment results of CS1 and SE courses were contradictory when students with different personality types, similar actual skill level, and similar self-esteem were paired together. The samples used in those studies were from undergraduate and graduate students; thus experience and academic background might be the reasons why these results varied.

In addition to the abovementioned studies, another three studies [13,19,32] looked at personality type, but not as part of their research question(s). We found that Myers-Briggs personality test (MBTI) was the most commonly used instrument to measure personality type [15,16,19,32,40]. Other than MBTI, NEO-PI had been used in [13] to measure the personality type. They discovered that the personality of a single programmer does not affect the results of using pair programming in general, but it might not be the case for the combination of personalities in a single pair. The only study that used the Keirsey Temperament Sorter test (KST) was [32]. The KST was used to measure the temperament of developers.

![Figure 3: Studies by Research approach](http://www.cs.auckland.ac.nz/~emilia/esem08/)
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Type of study</th>
<th>Compatibility factor</th>
<th>Outcomes on pair compatibility</th>
<th>Course(s)/students involved</th>
<th>Sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chao et al. (2006)</td>
<td>Survey &amp; Formal Exp.</td>
<td>Personality traits</td>
<td>Pair programming success (code quality and pair compatibility) is not influenced by differences in personality traits.</td>
<td>CS1 (UG)</td>
<td>58</td>
</tr>
<tr>
<td>Chaparro et al. (2005)</td>
<td>Field Study (Exploratory)</td>
<td>Skill level, type of role (driver or navigator), type of tasks</td>
<td>Skill level and the type of tasks are the most influential factors affecting the perceived effectiveness of pair programming. Skill level gap between the partners should not be too broad.</td>
<td>OOP (PG)</td>
<td>58</td>
</tr>
<tr>
<td>Heiberg et al. (2003)</td>
<td>Formal Exp. &amp; Exploratory</td>
<td>Personality type</td>
<td>The individual personality traits do not have significant consequences to pair programming performance.</td>
<td>OO (UG)</td>
<td>82</td>
</tr>
<tr>
<td>Katira et al. (2004)</td>
<td>Formal Exp.</td>
<td>Personality type, actual skill level, perceived technical competence, and self-esteem</td>
<td>Students were compatible if being paired randomly. Students have a preference to pair with a student they perceived to be of similar technical competence. They also prefer to work with partners with different MBTI (Myers Briggs Type Indicator) skills. Graduate students prefer to collaborate with partners of similar actual skill level. Pair compatibility does not affected by the students' self-esteem.</td>
<td>CS1 (UG) SE (UG) OO (G)</td>
<td>589 (2 academic Semester)</td>
</tr>
<tr>
<td>Katira et al. (2005)</td>
<td>Formal Exp.</td>
<td>Personality types, actual and perceived skills, self-esteem, gender and ethnicity</td>
<td>Compatibility was significantly influenced by the perceived skill and actual skill. Personality types and self-esteem are not critical for pair compatibility. Compatibility can be achieved if the pair consists of: - similar perceived skills level - similar actual skills level - female students - minority students</td>
<td>SE (UG) OO (G)</td>
<td>361 (3 academic Semester)</td>
</tr>
<tr>
<td>Layman (2006)</td>
<td>Survey</td>
<td>Personality type, learning style, work ethic, time management ability</td>
<td>Students who dislike pairing were those who experienced having incompatible partner. Students who possess sensing-intuitive learning style showed higher preference to work in pairs similar to those extraverts of MBTI skill. Students in the group who disliked collaborating were reflective learners, introverts and strong coders. Students' preference whether to pair or not was highly affected by the compatibility of the pair. Personality and learning style had little influence towards perception of collaboration.</td>
<td>SE (UG)</td>
<td>119 (2 academic semester)</td>
</tr>
<tr>
<td>Muller &amp; Padberg (2004)</td>
<td>Formal Exp.</td>
<td>“Feelgood” factor</td>
<td>Pair performance is correlated with the “feelgood” factor of the pair. However, the study cannot determine whether the performance is originally due to the “feelgood” factor or because the developers have the impression that they are performing well.</td>
<td>XP (UG)</td>
<td>38 (2 academic semester)</td>
</tr>
<tr>
<td>Sfetsos et al (2006)</td>
<td>Formal Exp.</td>
<td>Personality type and temperament type</td>
<td>Pairs of mixed-personalities and temperaments showed better performance and collaboration-viability. They achieved better points on assignments and shorter time to complete the tasks.</td>
<td>SE (UG)</td>
<td>84</td>
</tr>
<tr>
<td>Thomas et al. (2003)</td>
<td>Formal Exp.</td>
<td>Confidence level</td>
<td>Students who identified themselves as “warrior” preferred to work alone and enjoyed pair programming less. Paired students of similar confidence level can cause greater performance.</td>
<td>CS1 (UG)</td>
<td>60 approx.</td>
</tr>
<tr>
<td>Williams et al (2006)</td>
<td>Formal Exp.</td>
<td>Personality types, learning styles, skill levels, self-esteem, work ethic and time management skills</td>
<td>Compatibility can be achieved if students were paired randomly. The study identified the three factors contribute to compatibility: perceived skill, work ethic, and learning style. Students were compatible with partner they perceived of similar or higher skill level, similar work ethic, and different learning styles. Other variables that were studied such as personality types, actual skill level, self-esteem and time management skills are not significant contributors.</td>
<td>CS1 (UG) SE (UG) OO (G)</td>
<td>1350 (two-phased study 2002 – 2005)</td>
</tr>
</tbody>
</table>
Based on the review, we could see that personality type was the most common factor investigated. However, only [15] and [32] showed that there is a significant difference between pairs of mixed developer personalities and temperaments, and pairs of the same personalities and temperaments. Their results showed that the former communicate and collaborate better than the latter. There were no significant findings in the other studies in terms of influence of personality type on pair compatibility.

Our systematic review also discovered that students should be paired with a partner of a similar or higher perceived skill in order to achieve greater pair performance. The experiments demonstrated that pairs work well when both students have similar ability and motivation to succeed in the course [15,16,40]. In terms of gender issues, we found only one study [16] that investigated gender compatibility as one of the factors likely to determine effective pairing organization. Other gender related studies [2,14,23] were more concerned with assessing the usefulness of pair programming in increasing participation and retention of female students in CS/SE courses. Similarly, [16] is the only study to investigate ethnicity by classifying students as either belonging to majority or minority group. Their results showed that minority students were in favor of pairing. However, they did not investigate the effects on pairing compatibility when students within the same ethnicity were paired.

From the synthesis of evidence in answering the first research question, we foresee that further research related to achieving pair compatibility particularly related to gender differences and ethnicities is important to investigate success in pair programming practice. Investigating the personality types is also essential in order to confirm the earlier research findings.

6. CONCLUSIONS AND FUTURE WORK
This paper presented the initial results of a systematic review of pair programming empirical studies aimed at investigating the factors affecting the pair compatibility and how and when the effectiveness in pair programming were measured. From the 66 studies included in the review, only 10 studies were found relevant to answer the first research question.

In our initial findings, there were only three studies investigating the compatibility issues of pair programmers [15,16,40]. However, the findings of those studies were somewhat inconclusive. Overall, 12 compatibility factors investigated were personality type, skill level, self-esteem, gender, ethnicity, learning style, work ethic, time management ability, “feel good” factor, confidence level, type of role and the type of tasks. Even though personality type was investigated in many studies, only two studies [15,32] yielded positive results suggesting different or mixed personality types will lead to pair compatibility, while the remaining studies did not produce any significant results.

Our major finding from the review shows that pairing students of similar or higher perceived skills will lead to pair compatibility, thus achieving greater pair performance. The initial results from the systematic review also suggest that there is a lack of research concerning the social factors such as gender and ethnicity in understanding their influence on the pair programming practice. Our future work involves performing the analysis of Question 2 until Question 4 investigating issues on measuring pair programming effectiveness and its associated quality measurement. The results would help researchers in identifying gaps in the current pair programming research in improving the implementation of the practice in higher education institutions.

7. ACKNOWLEDGMENTS
The author would like to thank the University of Auckland for providing the funding to present this work at NZCSRSC08. We would also like to thank Emilia Mendes for her valuable comments and editing of this paper, and finally the ELSAC for proofreading the paper.

8. REFERENCES


[40] L. Williams, L. Layman, J. Osborne, and N. Katira, Examining the compatibility of Student Pair Programmers, the Agile 2006 International Conference, 2006.