On the Difficulty of Replicating Human Subjects Studies in Software Engineering

Jonathan Lung, Jorge Aranda, Steve Easterbrook and Greg Wilson
Department of Computer Science,
University of Toronto
Toronto, Canada, M5S 2E4
{lungj, jaranda, sme, gvwilson}@cs.toronto.edu

ABSTRACT

Replications play an important role in verifying empirical results. In this paper, we discuss our experiences performing a literal replication of a human subjects experiment that examined the relationship between a simple test for consistent use of mental models, and success in an introductory programming course. We encountered many difficulties in achieving comparability with the original experiment, due to a series of apparently minor differences in context. Based on this experience, we discuss the relative merits of replication, and suggest that, for some human subjects studies, literal replication may not be the the most effective strategy for validating the results of previous studies.

Categories and Subject Descriptors: A.m General Literature: MISCELLANEOUS

General Terms: Experimentation

Keywords

experience report, empirical, human subjects, replication

1. INTRODUCTION

Replication of empirical studies is frequently advocated but rarely practiced. For example, Basili *et al.* argue that systematic replication of experiments is crucial for building knowledge [1], while Kitchenham *et al.* identify the lack of incentive for conducting replications as one of the barriers to evidence-based software engineering [9]. In a recent survey of the empirical software engineering literature, Sjøberg *et al.* [14] found only twenty instances of published replications, just nine of which were performed by researchers other than the original team. The problem isn't unique to SE – replications are rare in many fields.

Many have speculated on why replication is rare. Among the reasons cited are the lack of information in published reports, even where materials are available, and that reproducing an experiment requires tacit knowledge that would never be captured in published reports [11]. Also, replications are seen as less interesting than novel research, and there is a perception in the research community that replications are hard to publish [9].

In this paper, we concern ourselves only with replication for experiments involving human subjects. Such experiments are increasingly important for improving our understanding of social and cognitive processes involved in SE. For these experiments, threats to validity are introduced by factors such as variability in human behaviour, difficulty of isolating confounding factors, and researcher bias. Effects observed in a single study might be caused by factors that were not measured or controlled. The aim of replication is to check that the results of an experiment are reliable. In particular, *external replication* (replication by different researchers) can identify flaws in the way that hypotheses are expressed and can help to identify the range of conditions under which a phenomenon occurs [2].

To properly replicate a human subjects experiment, published reports are usually insufficient. Basili *et al.* advocate using lab packages, whereby experimenters provide all their experimental materials along with precise details of their data collection and analysis techniques [1]. Even then, collaboration with the original team is important – possibly even essential.

Unfortunately, there are very few published experience reports of the challenges of replication in SE, beyond those cited above. This leaves many questions about replication unanswered. For example, how much involvement of the original research team is normal or necessary, and how does one achieve a balance between involvement and maintaining independence? How should we balance the goal of attempting a faithful replication against opportunities to improve on the original design? Are there cases where an entirely new study would be more suitable? And, if exact replication is impossible, how close can we can get, and how much do variations matter?

In an attempt to better understand replications, we performed one ourselves. We selected a study that was generating considerable buzz on the Internet in 2006. Dehnadi and Bornat had written a draft paper entitled *The Camel Has Two Humps*, in which they claimed to have developed a test, administered before students were exposed to instructional programming material, that is able to accurately predict which students would succeed in an introductory programming course and which would struggle [5]. The claims were startling enough that, even though the paper was unpublished[1], several groups around the world set out

[1]The paper is still, to date, unpublished.

ICSE'08, May 10–18, 2008, Leipzig, Germany.

to replicate the experiment. We chose to replicate this particular study for a number of reasons: we were interested in the results ourselves; the study design appeared to be sound, but (like all experiments) had a number of potential threats to validity; and the experimental materials were readily available from the original experimenters so that performing the replication seemed straightforward.

In attempting to replicate this experiment, we encountered many unexpected challenges. We discuss how we dealt with each of them and reflect on our experiences. We conclude that conducting a literal replication is hard, even with good access to the original researchers and their materials. For this particular study, we now believe we would have learned more by designing a new experiment rather than replicating the existing one. We draw on our experience performing this replication to explain this conclusion.

2. BACKGROUND

2.1 The Role of Replication in SE

Because of the importance of human activities in software development, empirical methods in SE are typically adapted from disciplines that study human behaviour, both at the individual level (e.g. psychology) and the team and organizational levels (e.g. sociology) [7]. The complexity of human behaviour means that these methods only provide limited, qualified evidence about the phenomena being studied, and all such methods have known weaknesses [10]. To overcome these weaknesses, viable research strategies make use of a series of studies [13]. Each study in the series can be a replication of an earlier study, an improved design over an earlier study, or can apply a different research method.

Experiments involving human subjects can produce highly variable outcomes due to factors such as experimenter bias, attention, prior experience, motivation, and expectations. While good experimental design reduces the impact of these phenomena, it cannot completely eliminate them. Hypotheses supported or generated from the results of such experiments should be subjected to a higher degree of scrutiny, including replication. Ideally, all experiments could and would be replicated. Publishing a paper describing empirical results is a tacit invitation for others to verify the results.

An *exact* replication is impossible [2]. In principle, any change in the experimental conditions could affect the results. Some variations in the subject population (e.g. nationality, culture, education, etc) and contextual factors (e.g. room layout, time of day, or even the weather) are inevitable. Hence, researchers wishing to replicate an experiment must decide which of the many potential variations matter, and then decide for each whether to replicate, control, or merely note the difference, preferably guided by an underlying theory [8]. In a *literal replication*, the goal is to come close enough to the original experiment so that the results can be directly compared. In contrast, a *theoretical replication* [15] seeks to investigate the scope of the underlying theory, for example by redesigning the study for a different target population, or by testing a variant of the original hypothesis.

Replications that obtain results consistent with those previously obtained increase the confidence the results can be trusted, either by showing that the same result holds under the same conditions (literal replication) or that predictably (dis)similar results hold when conditions are systematically altered (theoretical replication). However, *unintended* variations in the experimental conditions reduce the value of a replication, because they reduce the reliability of any comparison of the results[2]. The goal of a literal replication is not to obtain the *same results* as the original study, but to perform identical measurements on similar experimental units, treated as they were in the original experiment. A replication's results may disagree with the original's. In either case, replications serve to increase or decrease confidence in the hypotheses tested and/or to probe the conditions under which the hypotheses hold [2].

Replication of experiments in SE remains a challenge. Empirical studies in SE usually involve testing a tool or observing the software development process. Such studies require access to skilled participants, who may be difficult and/or expensive to attract and retain for a study. Locating suitable subjects can also be problematic because of the wide variety of tools and programming languages: only a small subset of available participants may have the required experience with a particular technology. Many SE tasks involve some degree of creativity, leading to large variations in answers, e.g., quality of source code or a model.

In addition, some experiments are harder to replicate than others. Some experiments are expensive to conduct, because they require large numbers of participants, cooperation of collaborating companies, or lengthy data collection and analysis. Others are hard to replicate because of a lack of information about the original study. For example, [11] assesses the quality of published reports of SE experiments, and discuss how reporting of procedures can be improved to lessen the tacit knowledge problem.

Such difficulties may be overcome if the payoff is great enough [2]. However, it is not clear how to assess the cost-benefit tradeoff for conducting replications. Human subjects studies are expensive and time consuming to conduct and all such studies are limited in some way. The crucial question is how much knowledge is gained by conducting a particular study (or replication), considering the amount of effort invested. For researchers interested in validating results of existing experiments, is it better to attempt a literal or theoretical replication, or to invest that same effort in designing a better study, or to probe the research question in a different way? To explore these questions, we set out to assess the difficulties involved in performing a literal replication.

2.2 The "Camel" Experiment

Our interest in exploring the practicalities of replication arose from several discussions during a graduate course on empirical software engineering methods taught by one of the authors at the University of Toronto in the spring of 2006. During the same course, we had read and discussed a draft paper describing an experiment by Dehnadi and Bornat [5] (hereafter referred to as D&B), in which they explore the question of why some students master programming skills relatively easily, while others seem unable to, regardless of pedagogy. While the draft paper has some stylistic flaws (and indeed, was never published), the study itself appeared to be sound. A careful reading of the report and

[2]One might be tempted to argue that in a replication producing results similar to the original, unintended (i.e. uncontrolled) variations are a good thing, because the effect is shown to hold over more conditions. However, such an argument is fallacious, because of the potential for sampling bias – two data points picked by non-random sampling are no more representative than one.

1. Read the following statements and tick the box next to the correct answer in the next column.

```
int a = 10;
int b = 20;

a = b;
```

Figure 1: Example question [5].

an examination of the experimental materials did not reveal any serious experimental flaws. The experiment met our criteria for a study worth replicating: the reported results were surprising, suggesting an important new theory about programming aptitude; the experiment itself was relatively straightforward; and the authors had made all the experimental materials available on their website.

D&B administered a simple test to subjects before they began an introductory programming course, and again after two weeks of instruction. The results indicated a strong relationship between the types of response given on both tests and the students' final marks in the course, leading D&B to claim the test can predict success in an introductory programming course when administered *prior* to any instruction [5]. After some basic screening questions, the test has 12 multiple-choice programming questions (e.g. see Figure 1). Each question asks for the result of a particular set of Java™ variable initialization and assignment statements, and is designed to determine what mental model the participant has used to ascribe meaning to the various symbols, if any. For the question in Figure 1, participants might use any of the following mental models:

- variables are unaffected by the equal sign;
- the variable to the left of the equal sign takes on the value of the variable to the right and the variable on the right retains its previous value; and
- the value of the variables are exchanged.

From the answers to each question, participants are categorized as: using the same mental model for a large portion of the questions (consistent); answering randomly or using several models haphazardly (inconsistent), or providing insufficient responses to categorize (blank). A participant was categorized as *consistent* if 80% or more of her responses use the same mental model and *blank* if fewer than 8 of the 12 questions are answered. All other respondents are considered *inconsistent*. The results showed that people who were consistent tended to do better in a 12-week introductory programming course than their inconsistent counterparts [5].

If consistent use of mental models correlates with success in introductory programming courses, it is a considerable breakthrough, with major implications for how programming is taught. Previous work on predicting programmer aptitude has been disappointing: it does *not* correlate with grade point averages, mathematics ability, age, nor any previous instrument used to assess academic potential. So exciting was this result that, in a field not known for replications, about half a dozen groups set out to replicate D&B's experiment. At least two replications have now been completed: one by Caspersen *et al.* at the University of Aarhus [4], and ours at the University of Toronto.

Given the choice between designing new studies to explore the theory (*theoretical replication*), or a repeat of the original experiment (*literal replication*), we opted for the latter, expecting that it would be straightforward to do first. If the results bore up, we then planned to continue with further theoretical replications.

3. EXPERIMENTAL REPLICATION

Because we set out to perform a literal replication, it was important that all steps of the experiment be performed as closely to the original as possible. We did not intentionally modify the procedure to improve it. However, as the number of inevitable changes accumulated, had we not justified and accounted for each change, we could not consider our work to be a literal replication. We were careful to distinguish between a literal replication and a new design, so that we could make sound comparisons with D&B's results.

We began by identifying any parts of the procedure that would require contact with the original authors to clarify. While planning may be straightforward, as in our case, certain essential details of the experimental setup may have been left out of publication or were unclear. At this stage we did not feel it necessary to contact the original authors regarding the setup. However, we did eventually exchange over a dozen e-mails with D&B for clarification and to verify our understanding of their analysis. In hindsight, we did miss a step when we planned the experiment: we neglected to record the time used by each participant to complete the test. This was done in the original study, but was not used for analysis. It is possible that timing participants affected the outcome, e.g., by inducing time pressure.

We also needed to make changes to adapt the procedure to local circumstances. To do this, we had to fully understand the procedure, hypothesis, and theory we were testing. We made a number of changes to the experimental phase including deciding which class would be most suitable and how to recruit participants (see Table 1 for a summary).

3.1 Ethics Approval

One source of changes in a replication comes from local constraints imposed by an Institutional ethics Review Board (IRB). In this case, the original study had no ethics review. Our IRB required us to obtain informed consent from the participants and to assure them that the course instructor would have no information on whether they chose to participate. Such ethics requirements impose serious restrictions: the study cannot be endorsed by a course instructor, nor can it be run as part of a required classroom session [12]. This in turn introduces challenges in recruiting subjects.

The ethics protocol we prepared for submission to our IRB was developed in collaboration with one of the other teams planning a replication, but even in this process it became clear that IRBs vary widely in what they will allow. One team told us that their IRB would not allow them to pay participants for participating in the study, and as a result they were unable to recruit many subjects to the study. To improve recruitment, we included a draw for all participants, to win CAD$50 gift certificates to the music retailer HMV.

3.2 Recruiting Subjects

Since it was not feasible for us to run the replication at the same university as D&B, participants were drawn from a different programming course. We recruited participants

Table 1: Summary of differences and changes to experimental phase

Original	U. of Toronto Replication	Primary reason(s) for difference
Participants drawn from courses at Barnet College and Middlesex University [5].	Participants drawn from a course at the University of Toronto.	Not feasible for us to run the experiment in the UK.
Prior mathematics experience not explicitly required.	All participants had grade 12 mathematics or equivalent.	The course requirements at the University of Toronto include this as a pre-requisite.
Some participants were contacted, interviewed, and tested prior to taking their first programming course [5].	Participants were not tested until the second week of classes and were not interviewed at all.	The interviews took place at Barnet College due to local protocol concerning admitting students. Final lists of students in courses at the University of Toronto are not available before classes commence.
The predictive test was administered twice to each participant [5].	The predictive test was administered once to participants.	We could not administer the test before the participants had started programming. Administering the test twice after one week would have likely been minimally informative and discouraging to participants.
The course from which participants were drawn were taught or tutored by an experimenter [5].	None of the participants were taught or tutored by anyone affiliated with the experiment.	Requirement for informed consent preclude the course instructors from endorsing the study. Also, such involvement could be a possible source of error.
Recruitment method unknown.	Participants were given a chance to win $50 DVD gift certificates.	A draw for gift certificates was within our means and an often-used method at the University of Toronto.
Participants were asked what A-Level courses or equivalents were taken [5].	Participants were asked what courses were taken to the highest high school level.	Participants in Toronto would be unfamiliar with the UK GCE system.
Responses by participants were coded using a subjective system [5, 6].	Responses by participants were coded using an automated tool.	Eliminating subjectivity increases reproducability. Since both produced the same result on some test data, results should be comparable. Further, using an automated approach improved reliability, speed, and facilitated additional analysis.

from CSC108H1, an introductory programming course that has tutorials and thirteen weeks of three-hour lectures at the University of Toronto. Based on information available from the websites of Barnet College and Middlesex University, CSC108H1 was similar in both length and content to the courses used in the original experiment. Further, based on descriptions of the exam in D&B and copies of the exam used in CSC108H1, the content and method of evaluation were similar.

As in D&B, this course is geared towards students hoping to obtain computer science degrees who have no prior programming experience. Two other first-year programming courses also exist at the University of Toronto: one for students with some experience with Java™ and one for students with other programming language experience. While students are encouraged to enroll in the course that caters to them, there is no enforcement of this. Meanwhile, the authors of D&B "believe that none [of their participants] had any previous contact with programming" [5]. These characteristics made CSC108H1 a suitable choice for this replication. These differences in instructors and educational system were unavoidable, and serve to increase confidence in the generality of the results.

In D&B, participants were contacted, interviewed, and were tested prior to the first day of classes, which was not possible at the University of Toronto [5]. Instead, potential participants were contacted by announcing the study at the beginning of their first lecture. Hence, participants could not be tested until *after* their first lecture. We invited students to visit a designated room at any time during several multi-hour periods in the second week of classes if they wished to participate. Those who came were given verbal instructions to read (and sign) the consent form, complete the test, and return the materials when they were done. While the timing of the study was not ideal, D&B found the same results in two separate administrations: one before the first lecture and the other after the third week of lectures [5]. Hence, one would expect that, if the relationship for exists, confirmatory results will be found by a test administered some time between the two points originally sampled in D&B. We note that convenience of the room and time turned out to be the major factor in attracting participants (more so than the prize draw!), as a session scheduled immediately after a lecture in a nearby room was the only session that yielded large numbers of participants.

3.3 Population

Ideally, the only difference between the original experiment and a literal replication would be the set of participants involved. We had to identify the population of interest based on the criteria laid out in D&B. We attempted to draw participants randomly from this population while ensuring that no participants had previously partaken in any other instance of the experiment, and that they were not aware of the experiment's exact purpose. We used statistical tests to reveal if the random selection criterion was met.

In this study, the population of interest is the set of all people who have had no prior programming experience and

are enrolled in introductory programming courses. Participants were asked if they had prior programming experience. The only difference of note between the population in D&B and ours was that a prerequisite for CSC108H1 is the completion of a mathematics course at the equivalent of a 12U or OAC course - the highest levels of mathematics offered in the province of Ontario, Canada. No equivalent enrollment restriction applied in D&B. The populations being sampled therefore differ in mathematical experience and it is unknown whether this had a material impact on the results. In fact, the population used for our replication is effectively a subset of the population sampled from in D&B. It is possible that those sampled by D&B would all have met the prerequisite anyway.

Note that theoretically, the study is not limited to those taking an introductory course on Java™, even though D&B's instrument uses Java™ statements. The claim is that D&B's test predicts programming aptitude in general, rather than ability to learn any particular programming language. Thus, students taking a course that uses a language other than Java™ should not be dismissed on the grounds of not sampling from the population of interest. On the other hand, such a change introduces several differences from D&B's original study, as the course content and assessments may vary in important ways, reducing comparability with the original results. An interesting theoretical replication that we considered was to replace D&B's instrument with one that uses a made-up programming language rather than Java™, in order to remove prior experience with Java as a confounding factor. We chose to stick to Java™ for both the test and the course from which we recruited subjects, to satisfy our goal of performing a literal replication.

3.4 Instruments

While several instruments were used, the instrument of interest was Dehnadi's test. Due to the fact that altering the phrasing of a question may have unintended consequences, we were hesitant to make changes. In the event, the only change we made was to the question asking for a list of "A-Level or any equivalent subjects" taken by participants. We localized this question to remove reference to A-Levels.

In D&B, participants were classified as consistent or inconsistent depending on how many of their responses to the 12 questions could be grouped into the same mental model. D&B describes four different categorizations of mental models. The strictest categorization, C_0, had eleven distinct mental models. By grouping some models together on logical similarities, the criteria for the consistent group were relaxed in C_1, C_2, and C_3 [5]; the mental model used for analysis in D&B was C_0 [6]. At the University of Toronto, the set of participants who were consistent at the C_0 level was equal to the set of participants at the C_1 through C_3 level. Thus, in our analysis, these were all interchangeable.

A subjective system for grouping and classification was used in D&B [5, 6]. For our replication, we developed a tool, in consultation with D&B, that produced deterministic results. We wrote a Python™ script that took responses from participants and output a degree-of-consistency (DoC) rating as a percentage, which could then be converted into a classification of consistent/inconsistent. To ensure that the script produced results in agreement with D&B, a few sample responses from our replication were sent to the authors of D&B and the classifications of the program and Dehnadi were compared [6]. Based on the results of this exchange, we found that the DoC produced by the Python™ script is easily converted to the classifications used in D&B.

1. Complete the following method:

```
/** Given an unsorted array of lowercase words
(Strings), return the word (String) that would appear
first in an alphabetically sorted array.  You may
assume there is at least one element in the array.  */
public static String getEarliest(String[] a) {
```

Figure 2: Typical CSC108H1 final exam question [3].

Note that in this study, Dehnadi's test is not the only instrument used, as grading instruments are used to assign a final course mark to each student. While we had no control over the grading instruments, the final exam questions in CSC108H1 did not seem qualitatively different from the sample questions in D&B (see Figure 2). To further test this assumption, we would need to have participants from D&B and our replication write both sets of exams and to check that both provide the same ranking of the students. This also assumes that tests are marked consistently.

3.5 Equipment

No special equipment was used for the experiment. While the actual statistics packages used in D&B were not published, the results were so strong that any stats package would have reproduced the results. Otherwise, it would be normal to use the same stats package as the original experiment, as implementation differences may give different results.

4. ANALYSIS REPLICATION

As has been noted, even the most carefully planned replication may be performed differently from the original. Some differences may serendipitously result in startling or otherwise important findings while other differences are unwanted and require additional analyses to control. We used a combination of experimental and analytical means to ensure that our results could remain comparable to the original results. For all but comparisons to the class as a whole, participants who reported prior programming experience were excluded from calculations - an experimentally identified difference that was controlled during analysis by removing these participants' data points. We did not need to modify our procedure to improve the quality of analysis possible.

4.1 Removing anomalous data

To obtain meaningful data and remove anomalous data points, we performed three data transformations. The first was the exclusion of experimental results of students who did not complete the course or write the final exam. Both cases result in no final mark being available. No inferences about these students are possible because of the variety of possible reasons for not completing the course: students doing poorly may have dropped out; students doing well may have transferred to one of the more advanced classes. Some students defer taking the exam for a variety of reasons. The second transformation was to reclassify two participants who listed only HTML as their programming experience. We re-

classified these as "no experience" since HTML is a markup, not programming, language. The third transformation was to exclude the single participant recruited from a second University of Toronto campus. It turned out that the time slots available for participating in the study on the second campus clashed with two other courses required to enter into the computer science programme which prevented people from showing up to participate. We felt it reasonable to simply exclude that campus from the analysis. All transformations were conducted prior to data analysis.

4.2 Checking response rate

Having "cleaned" our data, we checked our response rate. 96 participants took the initial test, of whom 59 (61.5%) went on to complete CSC108H1 and take the final exam. Our study therefore included a quarter (25.8%, 59/229) of the students who completed CSC108H1.

4.3 Detecting self-selection

We checked the degree of self-selection to see whether participants were different from non-participants. Participants were found to have significantly higher final marks ($\bar{x} = 78.3, \sigma = 11.2, N = 59$) than non-participants ($\bar{x} = 67.2, \sigma = 20.5, N = 170$), ($t(227) = 5.2$, $p < 0.05$, two-tailed). Unfortunately, corresponding data was not available for D&B. This suggests that participants self-selected. Though the programming course was intended for those who had never programmed, just over 60% of our sample declared prior programming experience. A more appropriate test was comparing the average of self-reported beginners to the group of non-participants. In this case, under the assumption that those who have programmed are likely to do no worse in the course than those who have not, one would expect that the average final marks of non-participants (possibly containing people with prior programming experience) to be at least as good as novice participants, though our data does not show a statistically significant difference between experienced and novice programmers ($t(39) = 0.74, p > 0.05$, two-tailed). Surprisingly, the self-reported novice participants fared significantly better ($\bar{x} = 76.9, \sigma = 12.8, N = 23$) than non-participants ($\bar{x} = 67.2, \sigma = 20.5, N = 170$), a portion of which may have contained people with prior programming experience ($t(191) = 3.1, p < 0.05$, two-tailed). Once again, due to a lack of information, it is not possible to compare these results with D&B. Regardless of the cause, the data is suspect (see Section 7 for a discussion of possible implications).

4.4 Comparing data

We checked to see if data that should be invariant across replications had deviated, which would have indicated that we had failed to perform the procedure properly. D&B reported that 8% of responses had an insufficient number of questions answered and were classified as "blank" according to a set of well-defined criteria [6]. Also, about 47% of non-blank participants were classified as "inconsistent". Our values for both these classifications did not differ significantly from D&B, ($\chi^2(1, N = 23) = 0.002, p >> 0.05$). Continuing with this line of testing, the number of "inconsistent" people was comparable to numbers in D&B, ($\chi^2(1, N = 22) = 0.004, p >> 0.05$). Thus, the proportion of people belonging to the blank, consistent, and inconsistent groups was not significantly different in the two instances of the experiment, which is what one would expect if the test were administered and scored in the same way to a random sample from the same population.

One might then think to compare the number of people who passed or failed the course. This type of comparison is not valid and is somewhat akin to comparing responses to a question on a Likert scale with different anchorings; the criteria and guidelines for awarding passing and failing marks may differ between academic institutions, changing mark distributions. E.g., the University of Toronto's Faculty of Arts and Science would consider a mark distribution with more than 20% of the class receiving marks in the E and F range (marks below 60%) to be anomalous. Therefore, one would not expect to find a large class at the University of Toronto where even close to half of the students failed, no matter how difficult the material, especially not on a regular basis as suggested happens in D&B.

Due in no small part to the grading guidelines at the University of Toronto, only 12.9% of the students in the class observed in the experiment failed the course and about 7% received marks in the 'E' range. Meanwhile, only 3% ($N = 2$) of our participants failed. These figures stand in stark contrast to the more than 50% of participants who failed the course in D&B. Since so few students fell into the failing group at the University of Toronto, it is hard to draw any conclusions about those who pass and those who fail. Since we are attempting to compare marks from different institutes, such a comparison is meaningless. Indeed, normally one should find the combining of results from two different institutes as done in D&B to be objectionable. However, in that instance, the same tests used to derive final marks were administered to all participants [5]; assuming that marking was uniform, the combining of marks was justifiable.

4.5 Main analysis

The central hypothesis in D&B is that students who use consistent mental models to interpret the questions on the initial test are more likely to succeed in learning how to program than their counterparts (although this is never stated formally in [5]). As there is no objective means of assessing "success" at learning to program, we use comparative, continuous, measures such as marks in a course instead. On that type of scale, we cannot say whether or not someone has learned to program, but only that some individual has earned a certain grade as assessed by the instructor.

In D&B, "success" was operationalized to mean that a participant had *passed* the course in which he or she was enrolled. This particular choice of operationalization is problematic. As discussed above, though students may have been ranked in the same order at different universities, passing a course at one university might mean something very different from passing at another, precluding the possibility of comparing results. This means that even if we had found a relationship with pass/fail rates in our replication, it should not be considered as evidence in support of D&B; otherwise, given the data, one must hold either the belief that no course at the University of Toronto teaches the same material as the courses in D&B or that first year programming students and/or instructors at the University of Toronto or the University of Aarhus are vastly better than those at Barnet College and Middlesex University [4].

Better operationalizations could use relative measures to account for the shortcomings of pass/fail counts. For

example, we could use a percentile as a dividing point or compare grade averages across the classifications. Using one of these alternative tests requires only that the students are comparable in ability and that course marks can be used to meaningfully rank programming aptitude, which seem to be reasonable assumptions.

Consider an alternate basis of comparison: using the median, the 50^{th} percentile, as a threshold, thus comparing the proportion of those who do better than the median to those who do worse than it. If we assume that our participants were of a similar calibre to those in D&B's study, then the proportion of students who have good programming ability should be the same amongst institutes, all else being equal. This supports comparison because approximately 48% of the participants passed the course in D&B. The result that consistency can predict performance relative to the median (i.e., 50% instead of the original 48%) is almost certainly supported by the data in D&B as it is significant using the existing division ($p = 0.00002$, Fisher's exact test). In our replication, being consistent has no significant correlation with being above or below the median at the end of the course ($p = 1.00$, Fisher's exact test). Equally easy to test is a difference in average marks between the consistent and inconsistent groups. As above, no significant difference was found ($t(20) = 1.0, p > 0.05$). D&B did not perform a similar test, so results cannot be compared.

The operationalization of "inconsistent" is also problematic in D&B. Blank participants were grouped with the inconsistent participants during analysis, but no argument was given as to why. We were unable to think of any convincing argument for it, so we chose not combine the blank group and the inconsistent group. We had only one blank participant, so we did not further analyze this group. However, we note that the final mark for the blank participant was higher than the median and the average inconsistent participant, so adding the result from this blank participant to the inconsistent group would not have helped support the results of D&B, and would have in fact been detrimental.

Dehnadi's threshold for assessing "consistency" can also be questioned. No justification for the use of 0.80 as the DoC cutoff for consistency was given, so this value seems arbitrary. Instead, we considered treating DoC as a points on a scale. Since there is no reason to believe that DoC is an interval or ratio value, DoC was treated as an ordinal value. If we take final marks to reflect programming ability on a continuum, we do not find a significant correlation between DoC and final marks ($\tau = 0.06, p > 0.05$).

Having performed the experiment and initial calculations, interpretation of the results is required. For example, since the relationship being sought was not found, should the theory be immediately dismissed? Perhaps a more liberal or conservative interpretation can be made. The final mark of participants in D&B was based on the average of two tests, while the final mark for participants at the University of Toronto was based on combined results from assignments, labs, and tests. Final marks are thus derived from qualitatively different evaluation methods. One possible reaction to our results is that the theory should be adjusted to limit its scope to programming aptitude on tests rather than programming in general since it is possible to program by "copy-and-paste", trial-and-error, and other methods in coursework assignments. To explore this, we performed the same battery of tests as before, but substituting just the students' exam marks rather than final course marks. These tests also showed no significant correlation, albeit with different p-values.

Note that normally, we would not perform multiple tests on the data, since this would constitute "fishing for results". While, in theory, the same relationship is being sought, it is possible that one test would yield significant results purely by chance, while others do not. The above tests were performed as illustrations of alternative operationalizations of the original hypothesis, because we were unable to apply the original method of analysis. Strictly speaking, for our replication, we should have selected one of them in advance. We could not do this as we did not anticipate the problem with the use of pass/fail threshold. In hindsight, we would have chosen only to do the median test, for best comparability with D&B. While this test was not performed in D&B, we reanalyzed their data so that our results could be compared. In a way, we created and executed a recipe improvement, and "re-ran" the original experiment using this new recipe to achieve comparability.

5. ADDITIONAL ANALYSIS

At this point, our results provided contradictory evidence against D&B. Instead of stopping there, we performed further analysis; some planned, some not.

One planned analysis was to examine how many consistent participants had actually used the correct, Java™-like, model (a "Java™-consistent" model). We suspected that one explanation for D&B's results was that most people who had *consistent* interpretations for the programming symbols were actually using the *correct* ones, because they had more programming experience that they admitted. Even if this was not the case, by being Java™-consistent on the instrument meant the ability to correctly answer similar questions on tests. In other words, responses to D&B's instrument were not independent to the course evaluation instruments; to find at least a small degree of correlation at the end should not be surprising. Most of our consistent participants appeared to used model two from D&B, which is the Java™-consistent model. This Java™-consistent group did not score significantly higher than the inconsistent group ($t(12) = 1.53, p > 0.05$, two-tailed). However, it did score significantly higher than the alternately-consistent group ($t(17) = 2.25, p < 0.05$, two-tailed). The lower average for the alternately-consistent group was not significantly different from the inconsistent group ($t(9) = 0.32, p > 0.05$, two-tailed).

This result suggests a very different interpretation for D&B's results and our failure to confirm them. Quite simply, the consistent group may actually contain two distinct subgroups, one that does much better than the inconsistent group, and one that does much worse. Differences between D&B and our replication can be explained by different proportions of these groups in the two studies. Furthermore, the relevant underlying theory is then one of functional-fixedness. Once a purpose or function has been found for a particular object, in this case the various tokens in Java™ statements, it is difficult to re-purpose them without using special learning/teaching techniques. Those who scored inconsistently have not formed a model and are thus more flexible when it comes to learning.

Table 2: Summary of differences and changes to analysis phase

Original	U. of Toronto Replication	Reason for difference
Tests for self-selection not performed/shown.	Tests for self-selection performed.	Testing for self-selection is a relatively easy way to test data validity and is good practice.
Transformations of data not performed/shown.	Recoding responses of HTML-only experience as having no experience. Exclusion of data of participants who claimed prior programming experience. Removal of data of participants who had no final mark available.	HTML is not a programming languages and those with programming experience are not part of the population of interest. Without final marks, the measure used to determine programming aptitude is missing.
No comparisons of data.	Data compared to D&B's.	D&B was not a replication.
Correlation examined between consistent mental models and passing/failing a course [5].	Correlation examined between consistency and being above/below the median.	The results from using a pass/fail scale cannot be used to compare other results easily.
Blank and inconsistent participants were combined during analysis [5].	Blank participants were not included in analysis.	No rationale given in D&B for combining blank and inconsistent groups. We found this to be flawed. We did not have sufficient blank participants to perform a separate analysis.

6. REDUCING THREATS TO VALIDITY

In performing our replication, we reduced some threats to validity without jeopardizing our ability to compare data with D&B. Some of these could have been avoided in the original experiment. There are two notable examples of this. In D&B, Dehnadi, an author of the paper, was the instructor of one of the courses and a tutor in the other [5]. He was also responsible for interviewing candidates who wished to take the course at Barnet College [5]. Dehnadi may have inadvertently taught the material differently had he not been involved in the experiment or, more likely, both his teaching techniques and the experiment revolve around his theory. Further, students may have reacted differently in the experiment and/or exam because their instructor or tutor was running an experiment in which they were involved. Thus, the observer-expectancy effect was a possible threat to validity. In this case, the course material should not have been taught by anyone involved in the experiment, much less someone with a stake in the outcome [12]. In our replication, the course was taught by parties whose only connection to the experiment was their cooperation in allowing participants to be recruited from the course and providing marks at the conclusion of the course.

Another change we made to eliminate a threat to validity was the introduction of a deterministic scoring algorithm of participants' responses. In the original paper, the subjective determination of whether a participant was consistent or inconsistent was made using a marking sheet on which the participants' standing in the course was already visible [5,6]. In our replication, our automated script ignores participants' marks when classifying participants' consistency, avoiding any potential coding bias.

It is, of course, entirely possible that we introduced new threats to validity that were not present in the original. For example, the data we gathered may have been affected by the fact that students with prior programming experience were encouraged to enroll in courses more suitable to them than CSC108H1. Responses concerning prior programming experience such as, "almost no Java", "I was tutored for two years (privately) but it was a long time ago," and "a very simple program in Turing" indicate that participants may have down-played their experience to avoid (imagined) negative consequences. This may also have simply been a form of image management in case they "performed poorly" on Dehnadi's test. However, it seems unlikely that participants would go so far as to deliberately answer test questions incorrectly on the test instrument to hide their experience. If those who have programmed before did better in the course, the expected effect of hiding experience was that the reported average marks of consistent novice participants is higher than the true average.

All students in the course were invited to partake in the experiment; participants were not required to have no prior programming experience. Had that been the case, a larger number of people may have been inclined to falsely underreport their past programming experience. This effect was not taken into consideration during the planning phase and it is probably fortunate that the invitation did not specify that participants needed to have no experience. None of these factors would have changed the fact that this replication failed to reproduce the results in D&B.

7. OBSERVATIONS

As with performing analysis, observations made should not be restricted to (dis)confirming results of the experiment being replicated. There may be things that were missed by other experimenters or did not happen while they performed the experiment. For example, in our replication, it was found that some participants changed their answers to questions in the first block of questions (a single assignment statement per question) but not in the other sections. This suggests that these people may have had one model to explain the statements and then revised the model upon encountering more complex sets of statements for which their previous models could not account. At the University of Aarhus, it was found that participants who were inconsistent but passed a final exam did not exhibit this behaviour [4].

While performing the replication, we found evidence that

some participants were consciously generating models via comments in the margin such as, "By now, you can see what [I]'m doing, but I haven't the faintest notion of how to approach this other than according to what I believe '=' and ';' to signify," "a is assigned the value of b," and "going logically a=b if b=20 ∴ a=20.b=20." We are unsure if this is significant.

As previously noted, our novice participants performed significantly better than non-participants. With 95% certainty, the novice participants scored 9.7 ± 8.6 percentage points better than non-participants. A 10-percentage-point difference in favour of the novice participants is meaningful as this is a whole letter-grade difference in marks. Such a difference needs to be accounted for if any weight is to be placed on the results. Some possible explanations include

- participants who were able to attend were taking fewer courses; a lighter workload might mean more time to devote to CSC108H1 and do better;
- non-participants registered for the course after the second week of classes, requiring them to catch up;
- non-participants did not hear our announcement because they regularly did not attend class;
- participants had more motivation than non-participants;
- Dehnadi's instrument improved course performance (e.g. by re-enforcing lecture material);
- participants with prior programming experience falsely reported none; or
- participants who were doing poorly in the course were more likely to drop out than non-participants as they knew their final marks would be seen by a third party.

Some of these explanations threaten the validity of our results. Unfortunately, we cannot determine the cause(s) from our data, nor from other reported instances of the experiment [4, 5].

8. DISCUSSION

Although we attempted to duplicate the original procedure, our results could not strictly be compared to those in D&B since the original metric was not suitable for our data. This, however, only became apparent during the analysis of our results. Therefore, as we described above, we had no choice but to iterate upon the original lab package to incorporate a better operationalization of the original hypothesis. Fortunately, D&B included enough data to determine what the outcome would have been had they used our improved design.

Ultimately, our results were the opposite of the findings in D&B. We did not find any strong correlation in our data, which barely deviated from what one would expect if there was no difference between consistent and inconsistent participants. This was the case even with very generous interpretations of the original hypothesis during our analysis. Although the contrast between the original results and ours is stark, the findings of D&B were highly unlikely to have occurred by chance. It is therefore important to identify plausible reasons to explain these differences.

We begin by identifying some relevant differences between our experiment and the original. For instance, due to the student-teacher relationship between Dehnadi and his participants, and to the way responses were scored, the original experiment had many opportunities for the experimenters to bias the results. Furthermore, each study used different exams to evaluate students; the discrepancy may have been caused by variations in emphasis when marking exam questions – for instance, giving different weights to syntax or correctness of code.

Another possible explanation is the difference in teaching techniques, independent of the fact that a researcher was an instructor in D&B. Whether any real differences in technique exist has not been verified. However, there is evidence to support the hypothesis that some form of functional fixedness affected our consistent group. If the teaching style used in D&B includes techniques to counter its effects, it could have the effect of raising the average marks of alternately-consistent participants. A simple follow-up experiment would be to look at the number of people who switch from being alternately-consistent to Java™-consistent at Barnet College, Middlesex University, and the University of Toronto, given the same courses and instructors. If teaching methods are indeed the cause for the discrepancy, one would expect to find a smaller proportion of switchers in our sample at the University of Toronto.

Therefore, even if both sets of results are valid and our study is indeed a literal replication, our apparently contradictory results do not necessarily imply that Dehnadi's theory is wrong. Due to the variability of teaching, we cannot exclude the possibility that Dehnadi's theory holds for certain teaching styles, which would still be interesting as it would suggest better ways to teach those who are able to learn to program. Thus, while our results were negative, we identified possible refinements of Dehnadi's theory.

We must also reflect on whether our effort in performing a literal replication was useful, and to what extent. A literal replication is necessarily focussed on testing the original experimental procedure itself, by applying it to a different sample from the same population. By performing a literal replication, our aim was to obtain results that would be comparable with those from the original study, to gather more evidence on whether D&B's claims were valid. We almost were unsuccessful in that respect, because of an accumulation of unavoidable (but unanticipated) differences between D&B's procedures and our own. In the end, we were only able to validate the results in D&B by coercing the original data into a comparable form by using the underlying theory about consistent mental models to derive better operationalizations.

In fact, our research goal was not to test D&B's procedures *per se*, but to test the underlying theory. In our original discussions of D&B, we had developed several new hypotheses relating to the theory that inconsistent mental models indicate difficulty with programming. For example, a plausible explanation for why some students struggle with programming is because they expect programming languages to be more like human languages, in which meanings of words and phrases are not fixed, but are interpreted by the listener, according to context. Students who use consistent mental models when interpreting unfamiliar programming constructs may have already grasped that programming languages are not like that.

We chose to start with a literal replication because we thought it would be a quick and easy first step to testing

the theory. On reflection, the literal replication was much more complicated than we expected, and told us very little about the underlying theory. On the plus side, we identified a number of flaws in D&B's experimental design, some of which we were able to correct. We also identified a number of further research questions arising from the marked difference in our results from D&B's. However, we invested a great deal of effort in trying to adapt the existing experimental procedures to our own context, and accounting for all the differences, so that we could maintain comparability. In contrast, if we had set out to perform a theoretical replication, we would not have been constrained by the existing experiment. We could have avoided the flaws and sources of error inherent in the original experiment (at the risk, of course, of introducing others). Most importantly, we could have directly tested the hypotheses we consider to be more scientifically interesting.

9. CONCLUSION

We originally set out to perform a literal replication of a seemingly straightforward human subjects experiment. However, throughout the design and execution of our experiment, a number of contextual – and unavoidable – difficulties forced our replication to deviate from the original study. These deviations reduce the comparability of the results with the original, and cast doubt on our ability to conclude that we disconfirmed D&B's results.

Through dealing with issues arising during our replication, we learned some important lessons. First, replicating even straightforward and well reported experiments requires the acquisition of a considerable amount of tacit knowledge. This confirms the observations of Shull *et al.* [11]. Second, even a seemingly simple instrument may be difficult to apply uniformly. Third, attempting to explain differences in results is a fruitful exercise. Fourth, and perhaps more importantly, each replication suffers a different set of contextual issues, and it is important to report experiences with replications for the benefit of the community.

While replications play an integral role in advancing scientific knowledge as they help to bolster or temper our certainty about various claims, refine those claims, discover effects not observed originally, and reduce the likelihood of experimenter bias, we conclude that it is difficult, and in some cases inconvenient, to perform literal replications involving human subjects in our field. Our replication identified some additional threats to validity in the original study that were not otherwise apparent, and we identified a number of questions for further study. However, this knowledge gain seems modest given the effort we invested.

In hindsight, we now believe that it would have been more fruitful to invest our effort in a theoretical replication. We should have devised an alternate experiment that, probing the same phenomenon, would have simultaneously allowed for a better understanding of the problem, further evaluation of the theory, a reduction of the number of confounds, and a less contrived experiment design and execution.

10. ACKNOWLEDGMENTS

We thank Saeed Dehnadi and Richard Bornat for their help in answering our questions about their study, and Jennifer Campbell, Richard Pancer, and Tobi Kral for access to their courses. We also thank the anonymous reviewers for many suggestions on how to improve the paper. This research was funded by NSERC.

11. REFERENCES

[1] V. R. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Trans. Softw. Eng.*, pages 456–473, 1999.

[2] A. Brooks, M. Roper, M. Wood, J. Daly, and J. Miller. Replication's role in software engineering. In F. Shull, J. Singer, and D. I. K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 365–379. Springer, 2007.

[3] J. Campbell. CSC108H1S final examination, 2006. `http://eres.library.utoronto.ca`.

[4] M. E. Caspersen, J. Bennedsen, and K. D. Larsen. Mental models and programming aptitude. *ITiCSE*, 39:206–210, 2007.

[5] S. Dehnadi and R. Bornat. The camel has two humps (working title), 2006. `http://www.cs.mdx.ac.uk/research/PhDArea/saeed`.

[6] S. Dehnadi and R. Bornat. Re: The camel has two humps. Personal correspondence, 2007.

[7] S. M. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting empirical methods for software engineering research. In F. Shull, J. Singer, and D. I. K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer, 2007.

[8] M. Jørgensen and D. I. K. Sjøberg. Generalization and theory building in software engineering research. In *EASE'2004*, pages 29–36. IEE Proceedings, 2004.

[9] B. A. Kitchenham, T. Dybå, and M. Jøorgensen. Evidence-based software engineering. In *ICSE '04: Proc. 26th Int. Conf. on Softw. Eng.*, pages 273–281, Washington, DC, USA, 2004. IEEE Computer Society.

[10] J. E. McGrath. Methodology matters: doing research in the behavioral and social sciences. In R. M. Baecker, J. Grudin, W. Buxton, and S. Greenberg, editors, *Human-Computer interaction: Toward the Year 2000*, pages 152–169. Morgan Kaufmann Publishers, 1995.

[11] F. Shull, V. Basili, J. Carver, J. C. Maldonado, G. H. Travassos, M. Mendonça, and S. Fabbri. Replicating software engineering experiments: Addressing the tacit knowledge problem. In *ISESE '02: Proc. Int. Symp. on Empirical Softw. Eng.*, Washington, DC, USA, 2002. IEEE Computer Society.

[12] J. A. Singer and N. G. Vinson. Ethical issues in empirical studies of software engineering. *IEEE Trans. Softw. Eng.*, 28(12):1171–1180, 2002.

[13] D. I. K. Sjøberg, T. Dybå, and M. Jørgensen. The future of empirical methods in software engineering research. In *FOSE '07: 2007 Future of Software Engineering*, pages 358–378, Washington, DC, USA, 2007. IEEE Computer Society.

[14] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal. A survey of controlled experiments in software engineering. *IEEE Trans. Softw. Eng.*, 31(9):733–753, 2005.

[15] R. Yin. *Case Study Research: Design and Methods (3rd Edition)*. Sage, 2002.