Ein kontrolliertes Experiment zum Programmverstehen

Jochen Quante*
Arbeitsgruppe Softwaretechnik
Fachbereich 3, Universität Bremen
http://www.informatik.uni-bremen.de/st/
quante@informatik.uni-bremen.de

1 Einführung

Dynamische Objektprozessgraphen (DOPGs, [3]) beschreiben den Kontrollfluss eines Programms aus Sicht eines einzelnen Objekts. Sie enthalten Informationen darüber, wie Objekte benutzt werden und wie diese Benutzungen in Beziehung zueinander stehen. Damit können sie für das Programmverstehen hilfreich sein. Dieser Beitrag berichtet von einem kontrollierten Experiment, das die Frage der Nützlichkeit dieser Technik für das Programmverstehen klären sollte. Eine ausführliche Beschreibung des Experiments findet sich in [2].

2 Hypothesen

Um eine Vermutung in einem kontrollierten Experiment auf ihre Gültigkeit prüfen zu können, bedarf es ihrer Abbildung auf objektiv messbare Kriterien. In unserem Fall wird "Nützlichkeit für das Programmverstehen" abgebildet auf die zwei Faktoren "Zeitaufwand" und "Korrektheit". Die Hypothesen lauten somit wie folgt:

H1 Wenn DOPGs verfügbar sind, werden Fragen zum Programmverstehen – die irgend etwas mit der dem DOPG zugrunde liegenden Komponente zu tun haben – schneller beantwortet als ohne.

H2 Wenn DOPGs verfügbar sind, ist auch die Korrektheit der Antworten größer, d. h. mehr Fragen werden richtig beantwortet.

3 Design des Experiments

In einem kontrollierten Experiment werden möglichst alle Einflussfaktoren konstant gehalten, bis auf diejenigen, die kontrolliert variiert werden. Es werden dann mehrere Gruppen von Probanden (eine pro Wert der kontrollierten Variablen) der gleichen "Behandlung" ausgesetzt, bekommen z. B. die gleichen zu lösenden Aufgaben. Dabei werden die abhängigen Variablen gemessen. Wenn sich die Messergebnisse je nach Wert der kontrollierten Variablen statistisch signifikant voneinander unterscheiden, so kann die Nullhypothese (hier "kein Unterschied") zurückgewiesen werden.

In unserem Fall ist die kontrollierte Variable die Verfügbarkeit von DOPGs. Es werden daher zwei Gruppen von Teilnehmern benötigt, die die gleichen Aufgaben zum Programmverstehen bekommen; Teilnehmer der einen Gruppe benutzen DOPGs, Teilnehmer der anderen Gruppe dagegen nicht. Die abhängigen Variablen, die dann gemessen werden, sind die Antwortzeiten und die Korrektheit der Antworten.

Ziel ist, dass ein beobachteter Effekt eindeutig auf die Variation der kontrollierten Variable zurückführbar ist. Die folgenden Abschnitte beschreiben, wie das Experiment aufgebaut wurde, um möglichst viele Störfaktoren auszuschalten.

3.1 Teilnehmer

Das Experiment wurde mit 25 Informatik-Studenten der Universität Bremen durchgeführt, die sich freiwillig zur Teilnahme gemeldet hatten. Voraussetzung war der erfolgreiche Abschluss des "Software-Projekts", in dessen Rahmen mit mehreren Personen eine etwas größere Java-Anwendung entwickelt wird. Von einer gewissen Programmiererfahrung konnte somit ausgegangen werden. Die individuellen Unterschiede der Teilnehmer wurden durch zufällige Zuordnung zu einer Gruppe so verteilt, dass sie sich einigermaßen ausgleichen sollten. Die Einbindung von Studenten als Probanden erschwert die Verallgemeinerung der Ergebnisse auf Softwareentwickler, allerdings sind professionelle Entwickler für eine solche Studie einfach nicht verfügbar bzw. zu teuer, und die Studenten sind immerhin die Profis von morgen.

Jeder Teilnehmer stellte sich für zwei Stunden zur Verfügung. Bei längerer Experimentdauer wird es zunehmend schwieriger, freiwillige Probanden zu finden. Hier muss daher ein Kompromiss zwischen Dauer und Anzahl Teilnehmer gefunden werden: Bei zu wenigen Teilnehmern leidet die Signifikanz der Ergebnisse. Die zwei Stunden wurden zur Hälfte für Einführung, Schulung und Übung, zur anderen Hälfte für die Bearbeitung der Aufgaben genutzt.

3.2 Aufgaben

Die Wahl des zu untersuchenden Systems sowie der zu lösenden Aufgaben sind von zentraler Bedeutung

*inzwischen bei der Robert Bosch GmbH, Corporate Sector Research and Advance Engineering Software, Stuttgart.

für den Nutzen des Experiments. Wird das Experiment nur an einem "Spielbeispiel" durchgeführt, ist die Verallgemeinerbarkeit auf große Systeme äußerst fraglich. Ist das System dagegen zu groß, brauchen die Teilnehmer zu viel Zeit für die Bearbeitung, die in der Regel nicht zur Verfügung steht. Im konkreten Fall fiel die Wahl auf ArgoUML[1] (160 KSLOC) und GanttProject[2] (43 KSLOC, beides Java).

Bei der Wahl der Aufgaben stellt sich die Frage, wie weit diese für das "Programmverstehen" repräsentativ sind. Im Experiment lag der Schwerpunkt auf Aspekten der Merkmalslokalisierung, d. h. der Frage, wo und wie ein bestimmtes Feature implementiert ist. Die Ergebnisse lassen sich somit auch nur eingeschränkt auf andere Aspekte verallgemeinern.

3.3 Weitere potentielle Störfaktoren

Die folgenden weiteren Faktoren können einen Einfluss auf das Ergebnis des Experiments haben. Dazu sind die jeweils angewandten vorbeugenden Massnahmen beschrieben.

Instrumentierung: Um sicherzustellen, dass die Messung objektiv erfolgt und nicht verfälscht werden kann, wurde diese automatisiert (per Eclipse-Plugin).

Sitzungsunterschiede: Das Experiment bestand aus mehreren Sitzungen. Um einen einheitlichen Ablauf zu gewährleisten, wurde anhand eines "Experimenthandbuchs" immer gleich vorgegangen. Die einführende Schulung wurde anhand der immer gleichen Folien durchgeführt. Allerdings konnte es Unterschiede bzgl. der Fragen der Teilnehmer geben.

Reihenfolgeeffekte: Es waren jeweils zwei Systeme nacheinander zu untersuchen. Um einen möglichen Effekt der Reihenfolge auf das Ergebnis auszugleichen, kamen alle möglichen Kombinationen gleich häufig zur Anwendung.

Wahrnehmung der Teilnehmer: Den Teilnehmern war recht schnell klar, was Gegenstand der Untersuchung ist; dies konnte nicht verheimlicht werden. Zudem konnten sie sich denken, dass DOPGs von der Person, die das Experiment durchführte, entwickelt worden waren. Allerdings war eine Manipulation der Ergebnisse schwierig, da Zeiten und Korrektheit gemessen wurden und keine subjektive Bewertung durch die Teilnehmer erfolgte.

Systemkenntnisse: Es ist ein weniger übliches Szenario, dass ein Wartungsingenieur mit einer für ihn völlig unbekannten Software konfrontiert wird. Der Normalfall wird eher sein, dass er immer die gleiche Software wartet und sie mit der Zeit gut kennt.

Erfahrungen mit DOPGs: Die Teilnehmer hatten nur sehr wenig Zeit, um mit DOPGs vertraut zu werden. Vielen war erst am Ende des Experiments klar, was DOPGs bedeuten und wie man mit ihnen arbeitet. Bei Teilnehmern mit mehr DOPG-Erfahrung sähen die Ergebnisse möglicherweise anders aus.

Wahl der Komponente: Bei der DOPG-Technik muss eine Komponente gewählt werden, auf deren Basis DOPGs extrahiert werden. Im Experiment wurde diese Wahl vorgegeben. In der Praxis stellt sich die Frage, wie ein Entwickler die richtige Komponente ermittelt. Auch die eigentliche Anwendung der Extraktionstechnik war nicht Teil des Experiments.

4 Ergebnisse

Die gemessenen Werte der abhängigen Variablen werden nach Durchführung des Experiments untersucht und eventuelle Unterschiede mittels statistischer Tests auf ihre Signifikanz untersucht. In unserem Fall kamen drei statistische Tests zur Anwendung: Der t-Test, der Mann-Whitney U Test sowie ein Bootstrap-Test.

Die Auswertung der Messwerte ergab widersprüchliche Ergebnisse für die beiden untersuchten Systeme. Im Fall von ArgoUML konnten beide Nullhypothesen deutlich zurückgewiesen werden: Alle drei Tests wiesen eine Wahrscheinlichkeit kleiner 3,5% auf, dass diese Unterschiede zufällig zustande gekommen sind. Bei GanttProject war dagegen kein signifikanter Unterschied zwischen den Messwerten feststellbar. Somit stellt sich die Frage, was die Ursache dieser unterschiedlichen Ergebnisse ist. Als wahrscheinlichste Ursache wurden im Nachhinein die unterschiedliche Anzahl und Größe der DOPGs identifiziert.

5 Fazit

Ein kontrolliertes Experiment ist das Mittel der Wahl für die objektive Verifikation einer Hypothese. Das Design und die Durchführung sind allerdings sehr aufwändig, was dazu führt, dass diese Technik in der Informatik noch sehr selten angewendet wird. In anderen Disziplinen ist sie dagegen Standard, z. B. in Psychologie oder Medizin. Eine gute Einführung in die Thematik findet sich beispielsweise bei Christensen [1].

Wie das beschriebene Experiment gezeigt hat, kann es beim Entwurf eines Experiments leicht passieren, dass wichtige Einflussfaktoren übersehen werden. In diesem Fall war es vermutlich die Anzahl und Größe der DOPGs, die zu widersprüchlichen Ergebnissen geführt hat. Diese Faktoren müssten nun – in weiteren kontrollierten Experimenten – untersucht werden.

Literatur

[1] Larry B. Christensen. *Experimental Methodology.* Allyn & Bacon, Boston, USA, 8th edition, 2001.

[2] Jochen Quante. Do dynamic object process graphs support program understanding? – A controlled experiment. In *Proc. of 16th ICPC*, pages 73–82, 2008.

[3] Jochen Quante and Rainer Koschke. Dynamic object process graphs. *Journal of Systems and Software*, 81(4):481–501, April 2008.

[1] `http://argouml.tigris.org/`
[2] `http://ganttproject.biz/`