

Empirische Untersuchungen zwei- und dreidimensionaler Darstellungen von zur Laufzeit erzeugten Objektstrukturen

Michael Striewe
Lehrstuhl Software-Technologie, Universität Dortmund
michael.striewe@uni-dortmund.de

Abstract: Es werden die Methoden und Ergebnisse einer empirischen Studie vorgestellt, mit der der Nutzen zweidimensionaler und dreidimensionaler Visualisierungen von zur Laufzeit erzeugten Objektstrukturen in objektorientierten Systemen verglichen wurde. Es wird gezeigt, wie der effektive Nutzen von Visualisierungen dadurch gemessen und belegt werden kann, dass Testpersonen anhand von vorgelegten Visualisierungen Fragen zu Programmstrukturen in JAVA beantworten.

1 Einführung

Die grafische Darstellung und insbesondere die Nutzung dreidimensionaler Visualisierungen sind zur Analyse von Laufzeitdaten in objektorientierten Systemen wenig etablierte Techniken. Während sich für einige Informationen, wie beispielsweise Variablenwerte bei primitiven Datentypen oder aktive Threads in verteilten Systemen, nicht sofort intuitive Darstellungsansätze finden lassen, bietet sich das Beziehungsgeflecht aus Objekten und Referenzen für eine Darstellung durch einen Graphen im dreidimensionalen Raum an. Aus der Planungsphase sind zweidimensionale Darstellungen insbesondere durch die UML bekannt und wurden auch für dreidimensionale Erweiterungen vorgeschlagen [RG00]. Welchen Nutzen eine solche Visualisierung konkret hat, wurde dagegen bisher kaum empirisch untersucht. Daher mussten vor der Durchführung der Studie zunächst auch Methoden entwickelt werden, die den Nutzen von Strukturvisualisierungen messen und belegen können.

2 Vorgehensweise

Es wurden drei Teilbereiche des Untersuchungsthemas identifiziert, zu denen je zwei Testfälle ausgearbeitet wurden. Zu jedem Testfall wurden Laufzeitinformationen aus JAVA-Programmcode erzeugt und durch zwei Werkzeuge zweidimensional [Zün07] und dreidimensional [DFSB05] visualisiert. Durch eine heuristische Evaluation der Visualisierungen anhand zuvor aufgestellter Kriterien wurden Hypothesen über die Nutzbarkeit der Darstellungen gewonnen. Die Überprüfung der Hypothesen erfolgte durch die Befragung von Testpersonen, die Visualisierungen vorgelegt bekamen und dazu Fragen zum zugehörigen Softwaresystem beantworten sollten.

2.1 Konstruktion der Testfälle

Fall 1 befasste sich mit der Verknüpfung zwischen bekanntem Programmcode und Visualisierungen. Für jede der beiden Verknüpfungsrichtungen wurden zwei- und dreidimensionale Visualisierungen in mehreren Komplexitätsgraden erzeugt. Zu jedem Programmcode wurden Variationen erzeugt, um je zwei abweichende Visualisierungen bzw. zwei abweichende Codeversionen zu erhalten. Insgesamt sollten die Testpersonen damit sechsmal zu einem gegebenen Programmcode aus drei Visualisierungen die zutreffende wählen (Abbildung 1) und sechsmal zu einer Visualisierung aus drei Programmcodes den zutreffenden bestimmen. Pro Frage erhielten sie eine Bearbeitungszeit von 3 Minuten, indem die Tests wie auch im folgenden Fall 2 unter Aufsicht durchgeführt wurden.

Fall 2 befasste sich mit der Gewinnung von Aussagen über unbekanntem Programmcode. Dazu wurden Laufzeitinformationen zu verschiedenen Zeitpunkten aus zwei Systemen, deren Programmcode den Testpersonen unbekannt war, zwei- und dreidimensional visualisiert. Insbesondere erzeugte der Programmcode Objekte verschiedener, teilweise innerer Klassen, deren Eigenschaften und Beziehungen untereinander im Programmcode rasch zu erkennen wären. Anhand von je fünf Fragen zu den Systemen, für die je System eine Bearbeitungszeit von insgesamt 8 Minuten zur Verfügung stand, sollte festgestellt werden, welche Aussagen über die Struktur durch die Betrachtung der Visualisierungen getroffen werden können und ob zwischen immer gültigen und situationsabhängigen Eigenschaften der Struktur unterschieden werden kann.

Fall 3 befasste sich mit der Erkennung erwarteter Strukturen in der Visualisierung. Dazu wurde den Testpersonen das Muster einer einfachen Objektstruktur vorgegeben. Zu je neun zwei- und dreidimensionalen Visualisierungen sollten sie entscheiden, ob und ggf. wie oft das Muster in der vorgegebenen oder einer veränderten Form auftritt, wie viele Objekte in der jeweiligen vorgelegten Struktur beteiligt sind oder wo es bei mehrfachem Auftreten die größte Struktur erzeugt. Die Bearbeitung dieses Falles und aller weiteren erfolgt durch die Testpersonen einzeln und ohne Aufsicht an einer Web-Schnittstelle, die die Antwortzeiten pro Frage erfasste.

Fall 4 befasste sich mit der Erkennung wiederkehrender Strukturen ohne vorherige Vorgabe eines Musters. In insgesamt 16 Visualisierungen sollten die Testpersonen bestimmen, ob auffällige Muster besonders häufig auftreten, wie viele Objekte an ihnen beteiligt sind oder welche Bereiche der Visualisierung dadurch besonders strukturiert erscheinen.

Fall 5 befasste sich mit der detaillierten Analyse von Baumstrukturen zum Zweck des Debuggings. Dazu wurden neun unterschiedlich große und komplexe Baumstrukturen implementiert und zwei- und dreidimensional visualisiert. Einige der Strukturen wurden mit Fehlern versehen. Durch die Betrachtung der Visualisierung sollte von den Testpersonen festgestellt werden, ob die Struktur fehlerfrei ist oder nicht (Abbildung 2).

Fall 6 befasste sich mit der Analyse eines Verbundes aus einem Array und zwei Listen ebenfalls zum Zweck des Debuggings. Dazu wurde eine Datenstruktur zur Speicherung von Spielpositionen auf einem Schachbrett einmal korrekt und dreimal fehlerhaft gefüllt. Erneut sollten die Testpersonen durch die Betrachtung der Visualisierung Aussagen über das Auftreten eines Fehlers und seine Art gewinnen.

Aufgabe 1A-2

Betrachten Sie folgenden Programmcode.

Frage: Welche der drei Abbildungen auf der folgenden Seite stelle die Objektstruktur dar, die von diesem Code erzeugt wird?

Antwort: _____

Hinweis: Gehen Sie davon aus, dass alle im Code erwähnten Klassen keine weiteren Objekte instanzieren.

```

public class TemplateParser {
    Tag tags[] = new Tag[5];
    DBConnect db;

    ContentElement header;
    ContentElement body;
    ContentElement footer;

    public void init() {
        tags[0] = new Tag();
        tags[1] = new Tag();
        tags[2] = new Tag();
        tags[3] = new Tag();
        tags[4] = new Tag();
        this.db = new DBConnect();
        // ...
    }

    public void parse(String id) {
        header = new ContentElement(0);
        header.connectToTp(this);
        body = new ContentElement(0);
        body.connectToTp(this);
        footer = new ContentElement(0);
        footer.connectToTp(this);
        // ...
    }

    public static void main(String[] args) {
        TemplateParser parser = new TemplateParser();
        parser.init();
        parser.parse(args[0]);
    }
}

public class ContentElement {
    TemplateParser tp;

    public void connectToTp(TemplateParser tp) {
        this.tp = tp;
    }

    public ContentElement(int data) {
        // ...
    }
}
    
```

Abbildung 1:

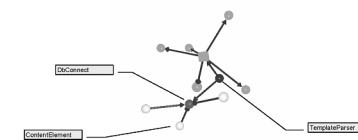


Abbildung 2:

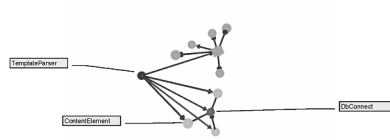


Abbildung 3:

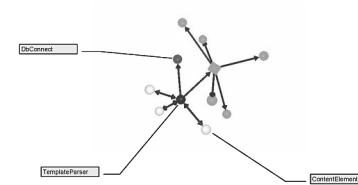
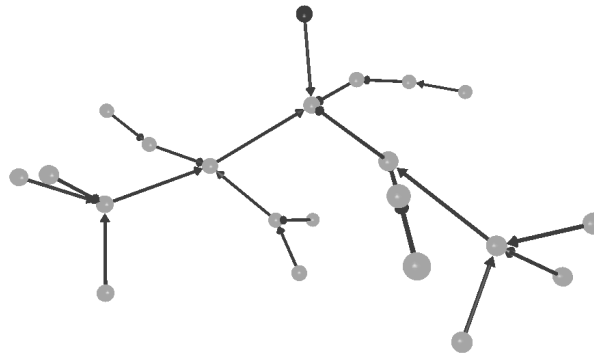


Abbildung 1: Frage 2 aus Fallstudie 1, bei der zu einem gegebenen Programmcodeausschnitt die zugehörige Visualisierung bestimmt werden sollte. Die Frage wurde in Papierform auf zwei DIN A4 Seiten gestellt.

Frage 14



Die gezeigte Datenstruktur ist als Baum geplant, in dem jeder Knoten maximal drei Kinder hat. Referenzen zeigen in dieser Visualisierung vom Kind zum Elternknoten. Ist die Struktur fehlerfrei implementiert?

- Die Struktur ist fehlerfrei
- Die Struktur ist fehlerhaft, weil ein Knoten zu viele Kinder hat
- Die Struktur ist fehlerhaft, weil ein Knoten zu wenig Kinder hat
- Die Struktur ist fehlerhaft, weil ein Knoten zwei Eltern hat
- Die Struktur ist fehlerhaft, weil mehrere der genannten Fehler auftreten

Abbildung 2: Frage 14 aus Fallstudie 5, bei der genau eine der Multiple-Choice-Antworten die Frage zum Debugging der gezeigten Struktur beantwortet. Die Frage erschien in Bildschirmgröße in der Web-Schnittstelle.

2.2 Heuristische Evaluation

Die Methode der heuristischen Evaluation wird insbesondere bei der Beurteilung von Nutzerschnittstellen angewandt [NM90] und wurde im Vergleich zu anderen Evaluationstechniken positiv bewertet [KS05]. In einer vereinfachten Form, in der die Evaluation von nur einem Experten anhand einer zuvor aufgestellten Liste von Kriterien durchgeführt wurde, diente sie der Herleitung von Hypothesen für die Studie. Die Kriterien ergaben sich aus der Gestalttheorie [CDT02] als weitgehend allgemeingültige Anforderungen an eine gut erfassbare Darstellung sowie als spezielle Anforderungen aus den Zielen, die der Nutzer bei seiner Arbeit verfolgt. Zu den allgemeinen Kriterien gehörten ergonomische Aspekte, beispielsweise die deutliche Semantik der Farb- oder Formgebung der verwendeten Symbole oder die Verteilung von Objekten im Darstellungsraum sowie technische Aspekte wie die Zahl der Kreuzungspunkte von Kanten oder die Stabilität gegenüber Perspektivwechseln bei dreidimensionalen Darstellungen. Spezielle Anforderungen waren je nach Fall beispielsweise die identische Darstellung identischer Teilstrukturen oder die intuitiv erkennbare Darstellung von Datenstrukturen wie Bäumen oder Listen.

2.3 Testpersonengruppe

In jedem Testfall der Studie wurden 9 bis 12 Personen aus einer größeren Gruppe herangezogen, die sich aus männlichen und weiblichen Informatik-Studierenden im Hauptstudium zusammensetzte. Diese verfügten aus Lehrveranstaltungen über Erfahrung in objektorientierter Programmierung mit JAVA, im Lesen von Objektdiagrammen in UML-Notation und dem Umgang mit Debugging-Werkzeugen. Aufgrund von unterschiedlichen Studienschwerpunkten der Testpersonen waren Erfahrung und Routine im Umgang mit UML und grafischen Darstellungen jedoch unterschiedlich stark ausgeprägt. Zur Feststellung der Stärke derartiger Co-Faktoren wurden den Testpersonen vier Fragen zur Selbsteinschätzung bezüglich objektorientierter Programmierung und Erfahrung im Umgang mit allgemeinen zwei- und dreidimensionalen Darstellungen gestellt. Als Beispiele für allgemeine Erfahrungen mit zweidimensionalen Darstellungen wurden dabei der Umgang mit Landkarten, Buslinienplänen, Schaltplänen und ähnlichem vorgeschlagen. Für die allgemeine Erfahrung mit dreidimensionalen Darstellungen wurden das visuelle Vorstellungsvermögen und der Umgang mit 3D-Shootern als Hilfestellung zur Selbsteinschätzung angegeben.

2.4 Experimentklassen und Fragentypen

Die Studie verwendete zwei unterschiedliche Experimentklassen. In vier Fällen wurde mit Vorgaben gearbeitet, die die Testpersonen auf die Visualisierungen beziehen mussten. In Fall 1 war dies vorgegebener Programmcode (Abbildung 1), in Fall 3 eine beispielhafte Visualisierung des gesuchten Strukturmusters und in den Fällen 5 und 6 die Bezeichnung der geplanten Datenstruktur (Abbildung 2). In den Fällen 2 und 4 wurde ohne Vorgaben

gearbeitet, da hier speziell der Nutzen reiner Visualisierungen ohne zusätzliche Vorgaben untersucht werden sollte. Während dies bei Fall 2 zu positiven und gut verwertbaren Resultaten führte, waren in Fall 4 die Testpersonen teilweise zu stark auf die Mustervorgabe aus Fall 3 fixiert. Eine Durchführung der Fälle 3 und 4 in umgekehrter Reihenfolge oder mit deutlich größerem zeitlichen Abstand hätte dieses Problem beheben können.

Ferner verwendete die Studie zwei unterschiedliche Fragentypen. Die Mehrheit der Fragen gehörte zum Typ der Multiple-Choice-Fragen mit zwei bis sechs Antwortmöglichkeiten und erwies sich als sehr gut geeignet. Lediglich in Fällen, in denen es zu einer auffälligen Häufung oder Streuung von falschen Antworten kam, die Missverständnisse in der Fragestellung vermuten lassen, verhinderte das Fehlen frei formulierter Aussagen eine genaue Ursachenforschung. In Fall 2 sowie einigen Fragen der Fälle 3 und 4 kamen als zweiter Fragentyp freie Fragen vor, in denen die Testpersonen Antworten frei formulieren bzw. Visualisierungsbereiche durch Anklicken in der Web-Schnittstelle frei bestimmen konnten. Dadurch entfielen die Nachteile missverständlich formulierter Antwortmöglichkeiten. Trotzdem konnte auch bei diesen Fragen das häufige Auftreten falscher Antworten beobachtet werden, wenn die zugehörigen Fragen insbesondere in den Fällen 3 und 4 den Testpersonen zu viel Interpretationsspielraum ließen.

2.5 Auswertungsmethoden

Die Auswertung der Studien erfolgte auf der Basis einfacher statistischer Methoden. Hypothesen über die effektive Nutzbarkeit einer bestimmten Darstellungstechnik wurden so aufgestellt, dass sie als Alternativhypothese angesehen wurden zu der Nullhypothese, dass ein Programmierer eine Visualisierung nicht nutzen konnte und zur Beantwortung von Testfragen raten musste. Über ihre Annahme oder Ablehnung wurde durch einen Signifikanztest mit einem Signifikanzniveau von 1% entschieden. Hypothesen über die Bearbeitungszeit wurden so formuliert, dass sie zwischen zwei Techniken verglichen. Zwei gemessene Mittelwerte galten genau dann als signifikant abweichend, wenn ihre Differenz größer war als das Maximum ihrer beiden Standardabweichungen.

In Fällen, in denen einzelne Bearbeitungszeiten deutlich um ein Vielfaches von den sonstigen Werte sowohl bei der zugehörigen Frage als auch bei anderen Antworten derselben Testperson abwichen, wurden diese vor der Berechnung des zur Auswertung herangezogenen Mittelwerts und der Standardabweichung aus der Statistik entfernt.

3 Ergebnisse der Untersuchungen

Im Bereich der allgemeinen Visualisierung von Programmstrukturen (Fälle 1 und 2) wurden von insgesamt 264 Fragen 183 korrekt beantwortet, was den Nutzen von Visualisierungen eindeutig belegt. Dabei erwiesen sich dreidimensionale Techniken als mindestens genauso mächtig wie die eher verbreiteten zweidimensionalen Techniken. Die Testpersonen konnten sogar tendenziell mehr Fragen im dreidimensionalen Fall korrekt beantworten

als im zweidimensionalen Fall, wobei zufällige Einflüsse der verwendeten Werkzeuge und gewählten Beispiele nicht auszuschließen sind und die Unterschiede nicht signifikant ausfielen. Bei der Erkennung wiederkehrender Strukturen (Fälle 3 und 4) ergab sich, dass in dreidimensionalen Darstellungen die Erkennung vorgegebener Strukturmuster nach im Schnitt 15,6 Sekunden deutlich schneller gelingt als in zweidimensionalen Darstellungen nach im Schnitt 26,8 Sekunden. Ohne Hilfestellung eines speziellen Layouts, das typische Strukturmuster hervorhebt, gelingt auch die Erkennung unbekannter Strukturmuster in dreidimensionalen Darstellungen mit 16,8 gegen 18,3 Sekunden tendenziell schneller. Im zweidimensionalen Fall konnte zudem gezeigt werden, dass spezielle Layouts zu bis zu dreimal besseren Fehlerraten führen. Ein ähnlicher Einfluss des Layouts auf die Bearbeitungszeit konnte dagegen nicht nachgewiesen werden. Beim Debugging von Datenstrukturen (Fall 5 und 6) konnte dies ebenfalls beobachtet und zudem die Größe der Strukturen berücksichtigt werden. Hier sinkt in komplexen Fällen die Quote richtiger Antworten in zweidimensionalen Visualisierungen um über 53%, während sie im dreidimensionalen Fall nur um ca. 36% zurück geht.

4 Erfahrungen und Empfehlungen

Der Ablauf der Studien mit der Aufstellung der Kriterien, der Erzeugung von Testfällen, der heuristischen Evaluation zur Hypothesengewinnung und der Befragung der Testpersonen hat sich bewährt. Die in den Fällen 3 bis 6 verwendete Web-Schnittstelle vereinfachte die Durchführung der Studien, erleichterte die Auswertung und trug durch die Möglichkeit zur Zeitmessung erheblich zur Gewinnung aussagekräftiger Ergebnisse bei. Nachteilig wirkte sich aus, dass die Testpersonen durch die Befragung ohne Aufsicht anders als in den Fällen 1 und 2 keine direkten klärenden Rückfragen zu den Aufgaben stellen konnten und somit unklare Formulierungen die Testergebnisse potenziell verfälschen konnten. Zu einer leichten Abschwächung dieses Effektes trug bei, dass in jeder Studie ab Fall 3 die ersten Fragen der Testreihe als Einführungsfragen nicht gewertet wurden. Ein solches Vorgehen scheint für derartigen Studien sehr empfehlenswert zu sein.

Die Dauer der Testreihen in den Fällen 3 bis 6 erreichte mit je 7 bis 14 Minuten ein Maß, das die Testpersonen nicht überforderte. Die Vorgabe von 8 Minuten Bearbeitungszeit in Fall 2 passt in diesen Rahmen, während die Vorgabe von 12 mal 3 Minuten in Fall 1 als zu großzügig zu betrachten ist. Im Konzentrations- und Motivationsverlauf der Teilnehmer konnte festgestellt werden, dass die Durchführung von mehr als zwei Testreihen unmittelbar hintereinander nicht sinnvoll ist. Die Anzahl der pro Fall eingesetzten Testpersonen erwies sich als gerade ausreichend, um sinnvolle Ergebnisse zu erzielen, wenngleich eine Beteiligung von 12 bis 15 Personen pro Fall wünschenswert erschien.

In weiteren Studien sind jedem Fall mangelnde Eigenschaften der verwendeten Werkzeuge zu berücksichtigen. Bieten die verwendeten Werkzeuge für identische Probleme unterschiedliche Lösungen an, so könnten die sich daraus ergebenden positiven oder negativen Effekte auf die Nutzbarkeit der Darstellung das Ergebnis verfälschen. Im Rahmen der Untersuchung wurden derartige Effekte, soweit es möglich war, in der heuristischen Evaluation nicht angerechnet. Trotzdem konnte nicht grundsätzlich ausgeschlossen werden,

dass Darstellungen von den Testpersonen immer als Verbund aus Dimensionalität, Layout, Farbgebung, Semantik der Symbole und textuellen Informationen gesehen werden.

5 Ausblick

Im Rahmen der Studien konnten längst nicht alle Fragestellungen zu Visualisierungstechniken empirisch untersucht werden. Um beispielsweise festzustellen, ob dreidimensionale Visualisierungen tatsächlich in großen und komplexen Strukturen überlegen sind, müssten in gezielten Testreihen alle mutmaßlich entscheidenden Faktoren, wie die Zahl der Objekte, die Zahl der Beziehungen, die Regelmäßigkeit und Tiefe der Strukturen sowie die Möglichkeiten zur optischen Unterscheidung von Objekten und Objektarten berücksichtigt werden. Bereits während der Durchführung der Studien zeigte sich ferner, dass sowohl im Verlauf einzelner Testreihen als auch in der Abfolge der Testreihen insgesamt innerhalb der Testpersonengruppe eine Gewöhnung an die bislang wenig verbreitete dreidimensionale Darstellung von Laufzeitinformationen stattfand. Es könnte sich daher anbieten, im Rahmen von Langzeitstudien einer Testpersonengruppe eine Einarbeitung in geeignete Werkzeuge anhand von praktischen Aufgabenstellungen zu ermöglichen und sowohl durch die Beobachtung der Arbeit als auch durch gezielte Testreihen festzustellen, wie groß der Einfluss von Erfahrung tatsächlich ist und in welchen Aufgabenbereichen dreidimensionale Darstellungen von Testpersonen aufgrund ihrer Vorteile bevorzugt eingesetzt werden.

Literatur

- [CDT02] Dempsey Chang, Laurence Dooley und Juhani E. Tuovinen. Gestalt theory in visual screen design: a new look at an old subject. In *CRPIT '02: Proceedings of the Seventh world conference on computers in education conference on Computers in education: Australian topics*, Seiten 5–12, Darlinghurst, Australia, Australia, 2002. Australian Computer Society, Inc.
- [DFSB05] Ernst-Erich Doberkat, Alexander Fronk, Jens Schröder und Sören Blom. Abschlussbericht der Projektgruppe 3Debug (PG 458). Bericht 156, August 2005. Internes Memorandum des Lehrstuhls für Software-Technologie.
- [KS05] Rex Bryan Kline und Ahmed Seffah. Evaluation of integrated software development environments: Challenges and results from three empirical studies. *International Journal of Human-Computer Studies*, 63(6):607–627, 2005.
- [NM90] Jakob Nielsen und Rolf Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 249–256, New York, NY, USA, 1990. ACM Press.
- [RG00] Oliver Radfelder und Martin Gogolla. On Better Understanding UML Diagrams through Interactive Three-Dimensional Visualization and Animation. In *Proceedings of the working conference on Advanced visual interfaces*, Seiten 292–295. ACM Press, 2000.
- [Zün07] Albert Zündorf. eDOBS - Java Heap as UML Object Diagram. URL, zuletzt gesichtet: 5.2.2007. www.se.eecs.uni-kassel.de/se/index.php?edobs.