Hasso Plattner Institut

Fachgebiet Computergrafische Systeme
Prof. Dr. Jürgen Döllner

Bachelorprojekt 2009/2010

Visualisierung von JAVA-implementierten Softwaresystemen

Hintergrund

Komplexe Softwaresysteme müssen kontinuierlich an sicher verändernde Nutzeranforderungen und Systemumgebungen angepasst werden – sie unterliegen ständigen Änderungen und Weiterentwicklungen. Diese Maintenance-Arbeiten sollten im Idealfall von Softwareentwicklern durchgeführt werden, die ein Gesamtverständnis der Struktur und des Verhaltens des Systems haben. Dies ist im Allgemeinen jedoch aufgrund der Größe und Komplexität der Systemimplementierungen nicht möglich. Die Folge ist: Mit zunehmendem Alter der Softwaresysteme sind Maintenance-Arbeiten immer schwieriger durchzuführen und nehmen mehr Zeit in Anspruch, so dass diese Arbeiten mitunter den größten Teil der Kosten während des gesamten Lebenszyklus hervorrufen. Waren bisher COBOL, C oder C++ Softwaresysteme aufgrund ihres Alters anfällig für die genannten Maintenance-Probleme, so sind zunehmend auch Systeme betroffen, die in der relativ jungen Programmiersprache JAVA geschrieben wurden (Java 1.0 erschien 1996).

In diesem Projekt soll ein **Analyse- und Visualisierungswerkzeug** für komplexe JAVA-implementierte Softwaresysteme konzipiert und implementiert werden. Das Werkzeug soll auf der (objektiven) Grundlage der Systemimplementierung die Struktur und das Verhalten des Systems sichtbar und verstehbar machen. Die Visualisierung des Laufzeitverhaltens spielt insofern eine besondere Rolle beim Systemverständnis, da Softwareentwickler zwar den statischen Quellcode bearbeiten, dieser allerdings nur eine Beschreibung der Möglichkeiten ist, wie sich das System zur Laufzeit verhalten kann. Die dynamischen Vorgänge im System müssen Softwareentwickler aufwendig über „Code-Lesen" mental simulieren und konstruieren. Debugging ist hierfür ein typisches Beispiel: Softwareentwickler führen manuell den Code schrittweise aus, springen dabei von Codestelle zu Codestelle und konstruieren dabei ein meist flüchtiges mentales Modell der Systemausführung.

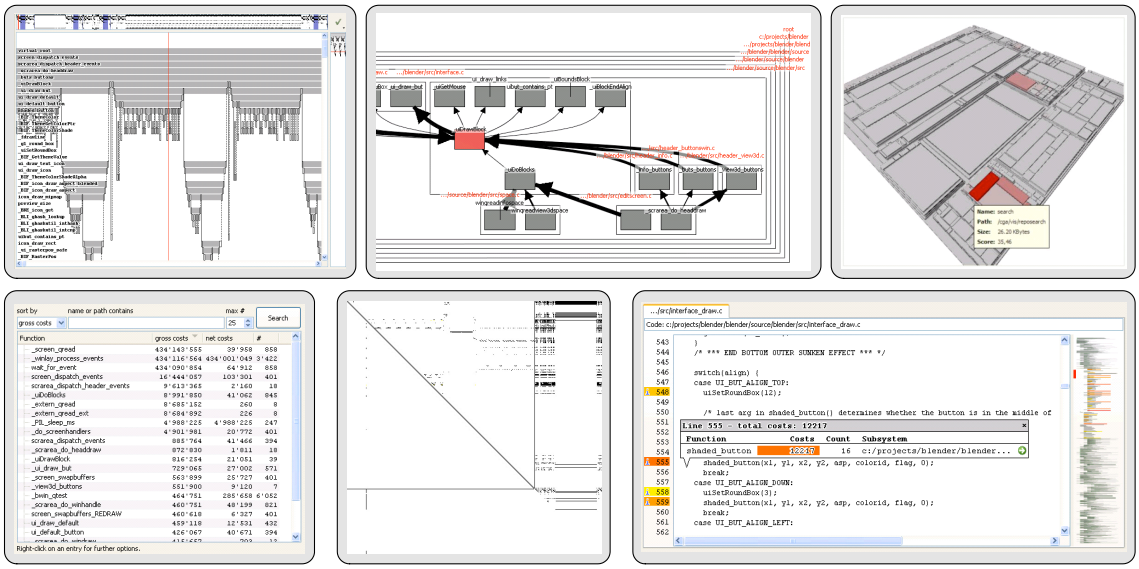
Genau hierfür, der **Konstruktion eines mentalen Modells des Systemverhaltens**, soll das Visualisierungswerkzeug Unterstützung bieten. Die Analyse und Visualisierung des Systems geschieht dabei post-mortem: Während einer gezielten Systemausführung werden der Kontrollfluss und ausgewählte Variablenbelegungen automatisiert protokolliert; diese Daten werden aufbereitet und dann über interaktive Visualisierungen den Nutzern bereitgestellt.

Gegenstand des Projekts

Das Bachelorprojektteam soll ein Werkzeug entwickeln, das das Laufzeitverhalten in JAVA implementierter Softwaresysteme analysiert und visualisiert. Das Werkzeug soll als Plugin in die Softwareentwicklungsumgebung Eclipse integriert werden. Die fachlichen Themen des Bachelorprojekts umfassen:

- Analyse und Auswahl von Techniken zur Extraktion von Java-Laufzeitinformationen
- Implementierung eines Extraktionsmechanismus für Laufzeitinformationen
- Entwurf und Implementierung eines Plugins zur Integration der bislang für C++ ausgelegten Technologieplattform CGA in die Entwicklungsumgebung Eclipse
- Analyse von Debugging-Szenarien und Herausarbeitung typischer Debugging-Workflows
- Entwurf und Visualisierungsunterstützung von Debugging-Workflows
- Erprobung des Werkzeugs mit Hilfe realer Java-Softwaresysteme Dritter
- Optimierung der Laufzeitdatenextraktion
- Optimierung der 2D- und 3D-Visualisierung

Die Realisierung des Werkzeugs erfolgt auf der Grundlage einer Technologieplattform, die von der Software-Diagnostics Technology GmbH, Potsdam, bereitgestellt wird. Insbesondere steht zur Umsetzung der Call-Graph Analyzer (CGA) bereit, der derzeit zur Analyse und Visualisierung von C++-Softwaresystemen eingesetzt wird.



Beispiele des bislang für C++ vorhandenen Analyse- und Visualisierungswerkzeugs CGA.

Organisatorische Umsetzung

Das Projekt ist eingebettet in die laufende Forschungsarbeit des Fachgebiets auf dem Gebiet der Software-Visualisierung. Die Mitarbeit erfordert ein entsprechendes eigenständiges wissenschaftliches Herangehen. Die Vorgehensweise im Projekt orientiert sich am Unified Process (UP) und wird methodisch im Sinne des Extreme Programming (XP) durchgeführt. In der Vorbereitungsphase werden fachliche Aspekte (z. B. Reverse-Engineering, Visualisierung, Laufzeitdatenextraktion, etc.) und organisatorische Aspekte, wie z. B. Projektplanung, Arbeitsmethodik, vermittelt. Dazu werden ausgewählte Aufgabenstellungen exemplarisch im Team durchgearbeitet, die zugleich die Einarbeitung in das Gesamtsystem bewirken. Es wird von den Studierenden nicht erwartet, dass sie alle genannten Grundlagen im Vorfeld beherrschen – detaillierte Kenntnisse werden im Vorbereitungsseminar vermittelt. Wir erwarten, dass jeder Teilnehmer sich in projektrelevante State-of-the-Art Methoden des Software-Engineering einarbeitet.

Umfeld

Das Projekt wird in enger Zusammenarbeit mit Software-Diagnostics Technology GmbH, Potsdam, durchgeführt. Darüber hinaus soll im Projekt mit ausgewählten Softwareentwicklungsfirmen kooperiert werden, die JAVA-Technologie für ihre Software verwenden und deren Systeme als Fallstudien dienen.
Die Studierenden werden im Rahmen dieser Kooperationen bei Veranstaltungen, Workshops, Präsentationen und Messeauftritten einbezogen. Es ist zu erwarten, dass bei erfolgreicher Bearbeitung nach Abschluss des Projekts auch eine studentische Beschäftigung am HPI oder beim Kooperationspartner möglich ist.
Das Projekt bietet aufgrund seiner wissenschaftlichen Ausrichtung eine gute Vorbereitung auf das Masterstudium in IT Systems Engineering und Informatik; eine Reihe von Aufgabenstellungen lässt sich aus dem Themenkomplex für spätere Abschlussarbeiten ableiten.

Gruppenstruktur

Zwischen 4 und 6 Teilnehmer können in diesem Bachelorprojekt mitarbeiten. Aufgaben und Organisation werden bei Projektbeginn mit den Projektmitgliedern erarbeitet.

Technische Umsetzung

Der Kern des Analysewerkzeugs wird basierend auf dem Framework CGA realisiert und ist daher in der Programmiersprache C++ implementiert. Für die Laufzeitdatenextraktion sowie für die Integration von CGA in Eclipse als Plugin wird zusätzlich die Programmiersprache JAVA verwendet. Eine Einführung in C++ sowie in das eingesetzte Framework CGA und JAVA wird in der Projektvorbereitung gegeben.

Information

Weiterführende Informationen zu diesem Bachelorprojekt sind bei Prof. Döllner und Johannes Bohnet erhältlich.