



Thematische Einordnung:

Sicherheitskritische, verteilte Echtzeitsysteme werden heute zunehmend auch im Automobil oder auch der Aeronautik eingesetzt. Um die Zuverlässigkeit dieser Systeme zu erhöhen, werden vermehrt fundierte Überwachungskonzepte verwendet.

Monitoring-based Runtime Reflection bezeichnet einen am Lehrstuhl für Software & Systems Engineering entwickelten ganzheitlichen Ansatz zur Überwachung, Diagnose und Steuerung von Software-Systemen. Er ist prädestiniert für verteilte Anwendungen, wie zum Beispiel Web-Server, eignet sich aber auch zur Überwachung der Kommunikation von Steuerelementen z.B. im Automobil. Runtime reflection besteht aus den Komponenten Runtime verification, Diagnose und Recovery und erweitert somit das insbesondere bei der NASA entwickelte Konzept der Laufzeitüberwachung (engl. Runtime verification) erheblich.

Zielsetzung:

Im Rahmen dieses Praktikums soll das Monitoring-Based-Runtime-Reflection-Framework mit besonderem Fokus auf die Entwicklung von Java-Programmen praxistauglich gemacht werden. Dazu sind unterstützende Tools (verschiedene Übersetzer), Beschreibungen der Tools sowie deren Einsatzes in Form von Online-Tutorials mit entsprechenden Web-Portalen zu entwickeln.

Konkret sollen folgende Arbeitspakete behandelt werden:

- Erarbeitung eines Interface-Konzeptes zu Log4j von Apache
- Erweiterung eines bestehenden Übersetzers von (nicht zeitkritischen) Überwachungseigenschaften in Monitore
- Erarbeitung eines Übersetzers von realzeitkritischen Überwachungseigenschaften in Monitore.
- Erarbeitung eines Deployment-Schemas, welches die Platzierung von Monitoren in einem verteilten System erlaubt
- Erarbeitung eines entsprechenden Deployment-Tools, welches Monitore erzeugt, verteilt startet, eine vorhandene Diagnose-Engine startet und diese miteinander verbindet.

Durchführung:

Grundlage ist eine vom Lehrstuhl gestellte Spezifikation, sowie die existierenden Teilergebnisse (geschrieben in Java/C++), die bislang im Projekt Runtime Reflection erarbeitet wurden. Die Aufgabe der Teilnehmer umfasst das Design, die Implementierung, das Testen und die Dokumentation der Erweiterungen, sowie das Projektmanagement. Eine intensive Zusammenarbeit mit den Betreuern wird vorausgesetzt.

Inhaltlich werden folgende Themen adressiert:

- Einarbeitung in das Monitoring-Based-Runtime-Reflection-Framework
- Einarbeitung in die Algorithmen, die dem Runtime-Reflection-Framework zugrunde liegen
- Einarbeitung in Projektkonfigurations-Tools, wie autoconf, make, gcc, etc.
- Einarbeitung in Logging-Frameworks wie Log4j von Apache
- Arbeiten mit einer aktuellen Entwicklungsumgebung mit Einsatz von Modellierungswerkzeugen
- Einsatz von Versionssystemen
- Debugging von Code

Das angebotene STP stellt in der Aufgabenstellung und Komplexität ein realistisches Entwicklungsprojekt dar, wie es auch in der Industrie anzutreffen ist. Die Entwicklung umfasst die Anforderungsanalyse, den Systementwurf und die Implementierung inkl. Test der zu entwickelnden Software. Da die Ergebnisse des STPs direkt in weitere Projekte einfließen werden, erfolgt die Entwicklung nach klaren Qualitätssicherungsvorgaben.

Organisatorisches:

Das Praktikum bietet die Möglichkeit, realitätsnahe Erfahrung in der Projekt- und Teamarbeit zu gewinnen. Die Ergebnisse sollen als Open-Source in die zukünftige praktische Arbeit von Programmierern einfließen und somit die Art der Entwicklung von Software in der Zukunft beeinflussen. Das Praktikum erfordert, wie von der Prüfungsordnung angestrebt, einen Zeitaufwand von ca. 1/3 Semesterwochenstunden.

- Teilnehmerzahl: 4-6
- Aufwand: 12 - 20 Std. / Woche
- Scheine: Praktikum aus dem Schwerpunkt Software-Engineering (Bachelor), Masterpraktikum (Master), Wahlpflichtpraktikum im Bereich praktische Informatik (Diplom)

Anmeldung:

Eine Vorbesprechung mit Anmeldung findet am Dienstag, den 16.01.07, im Raum 01.11.018 (Zuse) um 16.00 Uhr statt. Wer verhindert ist, möge sich bitter per Email an leucker@in.tum.de anmelden.

Voraussetzungen:

- Diplom-, Master-, oder Bachelorstudiengang
- gute Kenntnisse in den Sprachen Java und C++
- hohe Motivation und Lernbereitschaft
- Teamfähigkeit
- Allgemeine Kenntnisse über Softwaretechnik und Software-Engineering

Aufgabensteller:

Professor Dr. Dr. h.c. Manfred Broy

Betreuer:

Dr. Martin Leucker
Andreas Bauer

leucker@in.tum.de
baueran@in.tum.de

