

# **Titel: Monitoring von Software-Laufzeiteigenschaften binärer Programme**

**Typ: Projektarbeit (BSc), Abschlussarbeit (BSc)**

**Zeitraum: Beginn: sofort; Ende: Beginn + 3 Monate**

## **Motivation:**

Konfigurierbarkeit und Laufzeit-Anpassbarkeit von Programmen gewinnen zunehmend an Bedeutung. Dies wird beispielsweise ausgenutzt um die Dienst-Qualität von Programmen abhängig vom Nutzungsverhalten anzupassen, z.B. bezüglich Antwortgeschwindigkeit, Speicher- oder Energieverbrauch. Eine besondere Herausforderung ist es, die notwendigen Anpassungen autonom aufgrund von Messwerten (Monitoring) zu bestimmen. Als Grundlage hierzu hat die Arbeitsgruppe SSE ein Framework namens SPASS-meter entwickelt, das den Ressourcen-Verbrauch von Java-Komponenten zur Laufzeit ermitteln kann [1, 2].

Ziel dieser Arbeit ist es das Monitoring auf binäre Programme auszudehnen. Für viele Programme, z.B. sogar die in Windows enthaltenen, lassen sich die notwendigen Basisinformationen mit Hilfe des PIN-Frameworks ([www.pintool.org](http://www.pintool.org)) ermitteln. Diese Informationen sollen dann in das SPASS-meter Framework eingespeist werden.

Diese Arbeit bietet Ihnen einen praxisnahen Einstieg in das spannende Gebiet der Laufzeit-Programmanalyse.

## **Erläuterung der Arbeit:**

Zunächst arbeiten Sie sich in die Grundlagen von PIN [3] und SPASS-meter [1,2] ein und erstellen mit PIN ein einfaches Analyseprogramm, das die notwendigen Basisinformationen ermitteln kann. Im nächsten Schritt verbinden Sie das Analyseprogramm mit SPASS-meter auf zwei verschiedene Arten, 1) indem Sie die ermittelten Informationen via Netzwerk an SPASS-meter senden und 2) indem Sie aus dem relevanten Teil von SPASS-meter mit einem Java-Nativ-Compiler, z.B. JET [4], in eine native Bibliothek überführen, die Sie direkt in Ihrem Analyseprogramm verwenden können. Abschließend sollte Ihr Analyseprogramm wie auch die Integration mit SPASS-meter ausführlich getestet und analysiert werden.

*Der Umfang der Projektarbeit würde nur das Analyseprogramm und dessen Tests umfassen. Bei Interesse könnte die Arbeiten auch auf das Monitoring anderer Programmiersprachen als Java wie z.B. einen Vertreter der.NET-Sprachfamilie wie C# angepasst werden.*

## **Aufteilung der Arbeit:**

Theorie: 20%                      Implementierung: 50%                      Literatur: 30%

## **Literatur:**

[1] H. EICHELBERGER, K. SCHMID, *Erhebung von Produkt-Laufzeit-Metriken: Ein Vergleich mit dem SPASS-meter Werkzeug*, *Metrikon '12*, 171-180

[2] H. EICHELBERGER, K. SCHMID, *Flexible Resource Monitoring of Java Programs*, *Journal of Software and Systems*, 2014 (in print)

[3] M. BACH, M. CHARNEY, R. COHN, ET AL., *Analyzing Parallel Programs With PIN*, *IEEE Computer*, 43, p. 34-41, 2010

[4] Excelsior JET, <http://www.excelsior-usa.com/jet.html>

## **Kontakt:**

**Dr. Holger Eichelberger, [eichelberger@sse.uni-hildesheim.de](mailto:eichelberger@sse.uni-hildesheim.de)**