

# Monitoring unternehmenskritischer Anwendungen unter Verwendung modellbasierter Performance Constraints

Andreas Textor  
textor@informatik.fh-wiesbaden.de  
Fachhochschule Wiesbaden  
Fachbereich DCSM – Informatik

Unternehmenskritische Anwendungen werden zunehmend komplexer. Für die Entwicklung und den Betrieb derartiger Systeme sind strukturierte Methoden notwendig. Anwendungsmodelle werden aus Anforderungen konstruiert und dienen als Grundlage der Entwicklung der Software. Für die Modellierung hat sich insbesondere die Unified Modeling Language (UML, [Obj]) etabliert. Arbeitsabläufe (Workflows) von Business-Prozessen werden ebenfalls durch Modelle konkretisiert, und Teile von Workflows, die in Form einer Anwendung umgesetzt werden, können durch UML-Aktivitätsdiagramme beschrieben werden. Zur Sicherstellung der Performance einer kritischen Anwendung muss sie laufend überwacht werden. Für dieses Monitoring müssen Daten aus den verschiedenen Komponenten der Anwendung gesammelt und aufbereitet werden, um mögliche Leistungspässe zu finden und darauf reagieren zu können.

Dieser Beitrag stellt einen Ansatz vor, in dem im Workflow-Modell einer Anwendung Regeln für die Auswertung von Monitoring-Daten der Anwendung formal definiert werden können. Die Regeln, sogenannte Constraints, erlauben es, Aussagen über die Soll-Ausführungsreihenfolge und -dauer von Abschnitten des Workflow-Modells der zu überwachenden Anwendung machen zu können. Constraints werden auf der Ebene von Workflow-Modellen mittels der Object Constraint Language (OCL, [Obj06]) festgelegt, einer deklarativen Sprache zur Formulierung von Ausdrücken und Bedingungen in objektorientierten Modellen, die Teil der UML ist. Die Anwendung liefert zur Laufzeit Monitoring-Daten mittels Log4J oder ARM-Messungen (Application Response Measurement, [Ope03]), die über spezielle Mechanismen bestimmten Entitäten im zugehörigen Workflow-Modell eindeutig zugeordnet werden. Dies erlaubt es, die definierten Constraints anhand des Modells und der zugeordneten Monitoring-Daten zur Laufzeit automatisch zu überprüfen. Details über die Zuordnung und die Konstruktion solcher Constraints können in [Tex08] gefunden werden.

Performance-Daten können sowohl online ausgewertet werden, also während die zu überwachende Anwendung läuft, oder offline, indem zuvor aufgezeichnete Log-Files und Messwerte als Eingabestrom verwendet werden. Für die Offline-Auswertung kann mittels eines Reporting-Moduls ein Bericht erstellt werden, der aufzeigt, welche Constraints wie oft verletzt wurden; Leistungspässe in bestimmten Anwendungskomponenten können so auch im Nachhinein gefunden und zusammengefasst werden. Bei der Online-Auswertung können beim Auftreten von Constraint-Verletzungen SNMP-Traps an eine übliche Management-Konsole verschickt werden, die Informationen über den untersuchten Workflow und das verletzte Constraint enthalten. Diese Informationen können als Grundlage für ein Selfmanagement der überwachten Anwendung verwendet werden.

Der vorgestellte Ansatz entstand im Rahmen des Forschungsprojektes „PerManEntA“<sup>1</sup> des Labors für Verteilte Systeme der Fachhochschule Wiesbaden zusammen mit dem Projektpartner Lufthansa Systems Passenger Services (LHS), wo die entwickelten Methoden auf die LHS-Check-In-Anwendung „PICTURES“ angewendet wurden.

<sup>1</sup> Performance Management of Enterprise Critical Applications (gefördert durch die BMBF unter Förderkennzeichen 1706X07), [Fac08]

## Literatur

- [Fac08] Fachhochschule Wiesbaden - Labor für verteilte Systeme. *Performance Management of Enterprise Critical Applications (PerManEntA)*, 2008. <http://wwwws.informatik.fh-wiesbaden.de/projekte/permanenta.html>.
- [Obj] Object Management Group. *Unified Modeling Language (UML) Specification, Version 2.1.2*. [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm#UML](http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML).
- [Obj06] Object Management Group. *Object Constraint Language Specification, Version 2.0*, 2006. <http://www.omg.org/technology/documents/formal/ocl.htm>.
- [Ope03] The Open Group. *ARM 4.0 Java Language Binding Technical Standard 4.0*, October 2003. <http://www.opengroup.org/arm/uploads/40/3945/C037.pdf>.
- [Tex08] Andreas Textor. *Monitoring unternehmenskritischer Anwendungen unter Verwendung modellbasierter Performance Constraints*, September 2008. FH Wiesbaden, Fachbereich DSCM - Informatik, Bachelorarbeit.