

InterAkt

Interaktive Entwicklung Personalisierter Web-Applikationen¹

Projektgruppe des Lehrstuhls V

1. Thema Interaktive Entwicklung und Wartung Personalisierter Web-Applikationen.

2. Zeitraum Sommersemester 2003 und Wintersemester 2003/2004

3. Umfang Jeweils 8 Semesterwochenstunden.

4. Veranstalter

Prof. Dr. Bernhard Steffen, GB IV, Raum 129, Tel. 5801

Dr. Tiziana Margaria, GB IV, Raum 128, Tel. 5804

Dipl.-Inform. Oliver Niese, GB IV, Raum 125, Tel. 5806

5. PG-Aufgaben

5.1 Motivation

Die konzeptuelle Komplexität moderner Web-Applikationen stellt extrem hohe Anforderungen an ihre (integrierten) Entwicklungsumgebungen in Bezug auf

- Aspektorientierte, verteilte Entwicklung,
- Validierung,
- Ausführungsumgebung,
- Tests, und
- das Monitoring in Produktion befindlicher Lösungen.

Um Fehler frühstmöglich zu erkennen, ist es hierbei insbesondere wichtig, dass der jeweilige Nutzer (z.B. Entwickler oder Tester) in **jeder** Entwicklungsphase **interaktiv** mit der Entwicklungsumgebung arbeiten kann, also von der Konzeption bis zur Wartung nach der Inbetriebnahme durchgängig adäquate Ausführungs- und Validierungsmöglichkeiten mit direktem Feedback bekommt.

Moderne internetbasierte Anwendungen sind mehrstufige, verteilte Applikationen die typischerweise auf heterogenen Betriebssystemen ausgeführt werden. Ihre korrekte Ausführung hängt insbesondere von der Komponentenübergreifenden Zusammenarbeit verschiedenster Teilsysteme ab. Abbildung 1 zeigt eine solche Gesamtlösung auf. (Mehrere) Browser stellen hierbei die klassische Bedienoberfläche dar und interagieren mit einem Webserver. Der Webserver wiederum kommuniziert mit einem Applikationsserver, welcher dynamisch die angeforderten Web-Seiten darstellt, wobei die konkreten Daten direkt aus einer Datenbank in Interaktion mit verschiedenen Back-End-Diensten abgefragt werden. Zusätzlich können noch die Verwendung von Firewall-Lösungen, Load-Balancing und Redundante Architekturen den Test dieser komplexen Systeme erschweren. Gute Beispiele für solche Anwendungen sind der

¹Siehe auch <http://ls5-www.cs.uni-dortmund.de/teaching>.

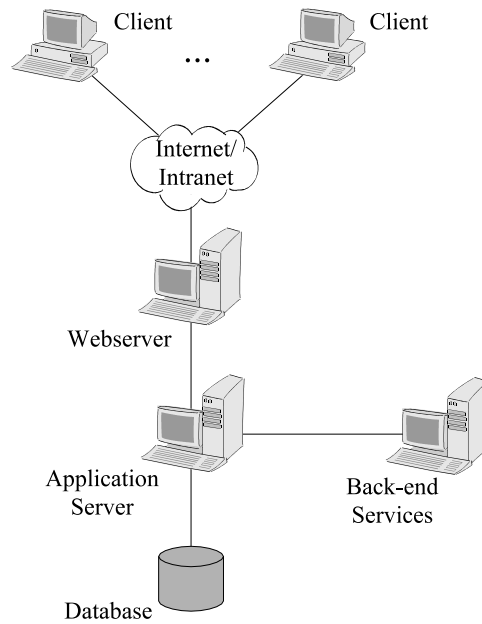


Abbildung 1: Beispielarchitektur für eine internetbasierte Anwendung

zur Zeit im Grundstudium eingesetzte internetbasierte Dienst zur Verwaltung der Grundvorlesung *Datenstrukturen und Algorithmen 1* (<http://templ.us.cs.uni-dortmund.de>) oder personalisierte Web-Portale wie etwa *Yahoo* oder *Lycos*.

Im Rahmen der Projekte “Enhanced Web Information Service” (EWIS) und “*Integrated Test Environment*” (ITE) werden zur Zeit integrierte Lösungen zur Entwicklung und zum Test derartiger Systemlösungen auf Basis des *Agent Building Centers* umgesetzt [6, 3]. Diese drei Umgebungen decken bereits die flexible Entwicklung personalisierter Web-Applikationen sowie die Erstellung von Testfällen und deren Ausführung ab [2]. Eine ganzheitliche interaktive Unterstützung wird aber derzeit nur ansatzweise geleistet [4].

5.2 Ziel

Im Rahmen der Projektgruppe soll eine ganzheitlich ausgerichtete, interaktive Entwicklungs- und Wartungsumgebung für Web-Applikationen erstellt werden, die es erlaubt, Web-Services zu **erstellen**, zu **validieren** und dabei insbesondere zu **animieren**, anschließend zu **testen** bzw. und nach Inbetriebnahme zu **monitoren**. Hierbei steht immer ein abstraktes Modell der Web-Applikation im Hintergrund, welches aspektbezogen transformiert werden muss. Die einzelnen Ziele sind:

Web-Applikations-Programmierung: Hier soll eine (visuelle) Programmierumgebung entstehen, die es erlaubt, personalisierte Web-Applikationen zu erstellen. Hier wird im wesentlichen eine Erweiterung der bereits bestehenden EWIS-Lösung angestrebt.

Erstellung einer Animationsumgebung: Hier soll es einfach möglich sein, eine Web-Applikation innerhalb der Entwicklungsumgebung zu animieren (*Rapid Prototyping*) um be-

stimmte Abläufe schon während der konzeptuellen Applikations-Modellierung animieren zu können. Hierdurch soll der Entwickler sich rein auf funktionale Aspekte, wie sie z.B. in UML durch Use Cases beschrieben werden, konzentrieren können, ohne von einer Laufzeitumgebung, wie z.B. einem (bestimmten)Webserver o.ä., abhängig zu sein.

Automatische Testerstellung: Hier wird im Sinne von [4] eine Konkretisierung des Modells der Web-Applikation angestrebt, die es erlaubt, Testfälle für das ITE zu generieren, um sie dort ausführen zu können. Hier wird ggf. auch eine Integration zusätzlicher Testschnittstellen speziell für den Test von Web-Applikationen durchgeführt werden müssen.

Der Akzent liegt hier auf der ausführbaren Modellierung von End-to-End Service-Verhalten, die aussagekräftig genug sind, um schon im Testlabor Rückschlüsse auf das Verhalten der entstehenden Web-Applikation nach der Inbetriebnahme zu gewinnen.

Monitoring: Um eine Web-Applikation im laufenden Betrieb zu monitoren, d.h. zu überprüfen ob sie sich kontrolliert verhält, wird eine spezielle Transformation des Modells nötig werden. Dieses Monitor-Modell soll dann in das ITE integriert werden, um mithilfe der System-Schnittstellen der Testumgebung das laufende System zu beobachten.

Erstellung einer graphischen Benutzeroberfläche: Hier soll eine graphische Benutzeroberfläche entwickelt werden, um die einzelnen Phasen des Entwicklungsprozesses durch gezielte Animations-/Interaktionsmöglichkeiten zu unterstützen und einem breiten Benutzerkreis zur Verfügung stellen zu können.

Erweiterungsmöglichkeiten: Basierend auf der Projektgruppenarbeit sind Folgeprojekte in Form von Hiwi-Tätigkeiten, Diplomarbeiten und Dissertationen geplant. Beispielt Themenbereiche sind multimediale Animationsumgebungen, die UML-basierte Ausführungssteuerung, sowie die automatische Testfallgenerierung mit automatischer Ergebnisanalyse. Begleitend soll die integrierte Entwicklungsumgebung dabei weiterentwickelt werden, mit dem Ziel, sie einem internationalen Anwenderkreis aus Industrie und Forschung als gemeinsame Plattform für Forschungs- und Entwicklungsprojekte zur Verfügung zu stellen, z.B. im Rahmen der bestehenden CMUDO-Kooperation mit der Carnegie-Mellon University in Pittsburgh, USA.

5.3 Generelle Vorgehensweise: Aspektorientierung

Das Projektgruppenprojekt verlangt aspekt-orientiertes Vorgehen in zwei Dimensionen:

- Die Entwicklungsumgebung selbst soll aspekt-orientiert auf Basis einer spezifisch zu erweiternden Komponentenbibliothek entwickelt und implementiert werden. Auf diese Weise sollen globale Konsistenzeigenschaften der Umgebung wie Versionssicherheit, Reaktivität und Vollständigkeit der Ausnahmebehandlung sichergestellt werden.
- Die Web-Applikations-Entwicklung mithilfe der Entwicklungsumgebung soll aspekt-orientiert unterstützt werden.

Die Aspektorientierung wird in beiden Dimensionen durch den regel-basierten Ansatz und das globale, grobgranulare, graphische Designkonzept des Agent Building Centers unterstützt, da es eine explizit für derartige Aspekte zugeschnittene, Modularisierungsgrenzen überschreitende Modellierungs-, Entwicklungs- und Analyseebene bereitstellt. Diese sogenannte Service-Logik-Ebene eignet sich gleichzeitig für das globale Versions- und Wartungsmanagement.

6 Teilnahmevoraussetzungen

6.1 Erforderlich

- Kenntnisse mindestens einer objektorientierten Programmiersprache (Java, BETA, C++ o.ä.)
- Grundlegende Kenntnisse in einem der Gebiete *Software-Design* oder *Testen* oder *Automatentheorie*, vorzugsweise nachgewiesen durch Teilnahme an einer entsprechenden Vorlesung wie "Softwaretechnologie", "Theorie der Programmierung", "Objektorientierte Modellierung", "Testorganisation komplexer Systeme", o.ä.

6.2 Wünschenswert

- Objektorientierte Modellierung (UML)
- HTML-Kenntnisse

7. Minimalziel

1. Web-Applikations-Entwicklung und Animation auf Basis von EWIS
2. Automatische Erstellung von Animationsskripten und ausführbaren Testfällen auf Basis von ITE.
3. Erstellung einer Bedienoberfläche, die die Interaktionsmöglichkeiten nahtlos in den Entwicklungsprozess integriert.

Literatur

- [1] The Aspect-Oriented Programming Page at Xerox.
<http://www.parc.xerox.com/csl/projects/aop/overview.shtml>.
- [2] T. Margaria, O. Niese, and B. Steffen. *A Practical Approach for the Regression Testing of IP-based Applications*. In IP Applications and Services 2003: A Comprehensive Report, Int. Engineering Consortium (IEC), 2002.
- [3] O. Niese, B. Steffen, T. Margaria, A. Hagerer, G. Brune, and H. Ide. *Library-based design and consistency checks of system-level industrial test cases*. In H. Hußmann, editor, *Proc. of the 4th Int. Conference on Fundamental Approaches to Software Engineering (FASE '01)*, volume 2029 of *Lecture Notes in Computer Science*, pages 233–248. Springer Verlag, 2001.
- [4] H. Raffelt. *Automatisiertes Testen von Internet-Applikationen*. Diplomarbeit am Fachbereich Informatik. Dortmund. 2002.
- [5] Rational, Inc. The Rational Suite description.
<http://www.rational.com/products>.
- [6] B. Steffen, T. Margaria: *METAFrame in Practice: Intelligent Network Service Design*, In *Correct System Design – Issues, Methods and Perspectives*, Springer Verlag, 1999, pp.390-415.
- [7] B. Steffen, T. Margaria: *Coarse-grain Component Based Software Development: The METAFrame Approach*, invited to the 3. Fachkongress "Smalltalk und Java in Industrie und Ausbildung" (STJA'97), 1997, Erfurt.