

Einleitung

Datum: 5. Mai 2008
Prüfung: **Diplomprüfung Praktische Informatik**

- Web Engineering
- Designing Interactive Systems 1
- Introduction to Database Systems

Prüfer: **Professor Schroeder, Professor Borchers**
Dauer: ca. 45 Minuten
Note: 1,7

Gedächtnisprotokoll

Web Engineering (Professor Schroeder)

Schröder: Wir haben uns ja Server-Side und Client-Side Technologien angesehen und das HTTP Protokoll, was ja Stateless ist. Was kann man denn da machen um das zu umgehen?

Ich: Man kann Werte als Hidden Fields oder als Parameter übertragen, was aber nicht so praktisch ist wie Session Management.

Nicht so praktisch, was gibt es denn da für verschiedene Methoden?

Man kann die Daten auf dem Client speichern, und auf dem Server. Wenn man sie auf dem Server speichert gibt man dem Client typischerweise eine Session-ID um die Session dem Client wieder zuzuordnen zu können.

Wie werden denn Sessions über HTTP übertragen?

Über Hidden Fields oder Parameter, oder über Cookies.

Genau. Cookies. Wie funktioniert das?

Der Server speichert z.B. die Session-ID in einem Cookie und sendet diesen an den Client, und der überträgt dann mit jedem HTTP-Request automatisch den Cookie mit.

Wie funktioniert denn URL-Rewriting?

Der Server fügt die Session-ID an alle Links im Dokument an. Als Parameter.

Dann hatten wir ja auch über HTML und CSS gesprochen. Wofür ist CSS denn da?

Also HTML ist ja nur strukturierter Content, und das CSS ist dazu da das Aussehen der HTML-Elemente und das Layout zu verändern.

CSS heisst ja Cascading Style Sheets, was bedeutet denn „Cascading“?

Man kann CSS ja an unterschiedlichen Stellen definieren, und der Interpreter entscheidet dann anhand der Cascading Regeln welche der Definitionen gilt, wenn es mehrere Definitionen für das gleiche Element an unterschiedlichen Stellen gibt.

Wie kann man denn CSS in HTML einbinden?

Man kann es direkt in den Tag schreiben, das hat dann die höchste Priorität und überschreibt alle anderen möglichen Definitionen, dann kann man noch in den HEAD des HTML-Dokuments mit `<style type="text/css">` CSS-Definitionen schreiben, und auch in externe Dateien, die man dann im HEAD mit dem link-Tag einbindet.

Wie sieht so eine Definition im Tag aus?

Da benutzt man das style-Attribut des HTML-Tags.

Wie funktioniert denn Vererbung bei CSS?

Vererbung? Ähm...

In einem HTML-Dokument hat man je Elemente.

Ach so... Wenn ein Element einen bestimmten Style hat, dann erben alle Kindelemente den Style des Elternelements automatisch, also z.B. die Schriftgröße. Es sei denn man redefiniert es in einem Kindelement wieder.

Gut. Dann haben wir ja Server-Side J2EE angesehen, mit Servlets und JSP. Wie funktionieren denn Servlets?

Servlets sind Java Programme, die auf einem Webserver laufen. Dort sind sie als Bytecode gespeichert...

... Machen Sie doch mal eine Skizze wie ein Servlet auf einem Webserver funktioniert.

Skizze mit Client, HTTP-Request, Servlet Container, Request und Response Objekt gezeichnet. Servlet Container regelt die Kommunikation mit dem Client, und erstellt vom Servlet einen Thread. service() Methode bestimmt Request-Typ und ruft dann doGet() oder doPost() auf und dort wird das Response Objekt bearbeitet und wieder an den Container zurückgegeben.

Wer erstellt denn das Servlet-Objekt?

Naja... ich schätze mal der Servlet Container.

Ja und wie funktioniert das dann.

Ich denke bei der ersten Anfrage an das Servlet wird er das erzeugen, wenn es noch nicht vorhanden ist. Sonst wird dann immer ein Thread davon erzeugt.

Wenn wir eine Datenbank an das Servlet anbinden, wie geht denn das? Was gibt es da für Zusammenhänge.

Die Datenbank wird in der doGet() Methode angesprochen. Dann gibt es noch den Deployment Descriptor, und den Servlet Context. Da kann man dann Concurrency Probleme bekommen.

Was ist das denn?

Da verschiedene Threads von verschiedenen Clients den gleichen Context verwenden könnte es Probleme geben, wenn ein Client Werte im Context ändert, da sie dann auch für die anderen Clients geändert sind.

Und wie kann man das verhindern?

Da gab es eine Methode mit der man das Thread-Safe machen kann. Bin ich mir nicht sicher.

Man kann Thread-Safe auf den Context zugreifen.

Genau, so ähnlich wie bei CVS, indem man den Zugriff für die anderen Threads sperrt.

Dann kommen wir zu XML-Technologien. Dazu habe ich eine DTD vorbereitet. Schreiben Sie doch mal eine valide XML-Datei dazu auf, und erklären Sie dabei was „valide“ bedeutet.

Gemacht, valide bedeutet dass es die DTD oder die XSD erfüllt, well-formed bedeutet dass es keine Syntaxfehler enthält.

Sie haben ja schon gesagt dass man bei der DTD nicht viel einschränken kann, wie könnte man denn hier mit einer Schemadefinition etwas einschränken.

Ich habe ein Attribut „minAge“ mit `<xs:attribute> <xs:simpleType base="xs:integer"> <xs:minInclusive ...> <xs:maxExclusive>...` auf Integer zwischen irgendwelchen Zahlen eingeschränkt.

Genau, und dann nimmt man da Werte die Sinn machen. Wenn man aus der XML-Datei jetzt eine Tabelle machen möchte, wie macht man das?

Dazu kann man eine XSL-Transformation benutzen. XSLT transformiert XML in ein beliebiges Zielformat. Wenn das eine HTML-Tabelle werden soll, dann spezifiziert man die HTML-Tags und mit einer `xsl:for-each` Schleife geht man die Autoren (aus dem XML) durch. Mit XPath Ausdrücken kann man dann auf die einzelnen Werte und Attribute zugreifen.

Wie kann man denn das XSL noch strukturieren?

Ähm...

Wenn man das nicht alles vom Root-Element aus machen will.

Man kann das in Templates unterteilen, z.B. ein Template für jede Zeile in der Tabelle und dann mit `xsl:apply-template` oder so aufrufen.

[Danach kamen noch ein paar Fragen zu einem Zusatzthema das nicht zur Vorlesung gehört, um auf 4SWS zu kommen. Da sie nicht zur eigentlichen Vorlesung gehören habe ich es weggelassen.]

Designing Interactive Systems 1 (Professor Borchers)

Borchers: Wir haben ja in der Vorlesungen Modelle kennengelernt wie ein Mensch mit der Umwelt interagiert. Die kann man unterteilen in qualitative und quantitative (Am Ende kommen Zahlen raus) Modelle. Machen wir doch mal eine Liste. Was gibt es da so?

Ich: CMN Model - habe ich fälschlicherweise zu den Qualitativen gepackt, Fitt's Law - Quantitativ, GOMS auch quantitativ.

Und Qualitative?

Die Modelle aus dem Norman-Buch. Affordances, Mappings, Constraints, Slips.

Fällt Ihnen da noch ein Qualitatives Modell von Norman ein?

Die 7 stages of action mit den Gulfs.

Genau. Schreiben Sie mit doch mal Fitt's Law in der Originalfassung auf.

Fitt's Law beschreibt die Positionierungszeit von eindimensionalen Bewegungen. Formel hingeschrieben.

Und als Grafik?

Hier hat es dann ziemlich gehakt. Mir fielen nicht mehr ein welche Achse was darstellt, nur die Grafik an sich. Mit einigem Hin und Her wurde dann langsam etwas draus.

Wie sieht denn eine Grafik aus, die die Positionierung mit D und W darstellt?

Hingemalt, aber D nicht bis zur Mitte des Targets gezogen, sondern nur an den Anfang.

Ich bin nicht ganz glücklich.

Nicht? Weil D nicht bis zur Mitte des Targets geht?

Genau. Das sollte bis zu Mitte gehen. Wenn ich jetzt ein Handy habe, und es in alle Richtungen doppelt so groß mache, wie ändern sich denn dann die Positionierungszeiten nach Fitt's Law?

Wenn die Knöpfe weiter voneinander entfernt sind, dann werden auch die Zeiten höher.

Ich mache ja alles doppelt so groß.

Ach so... dann ändern sich die Zeiten nicht.

Genau. Sieht man ja auch an der Formel, wenn ich D und W verdopple kürzt sich das ja wieder weg. Jetzt nehmen wir an Sie arbeiten bei einem Handyhersteller, und wegen Fitt's Law kommt er auf die Idee das Handy extrem klein zu machen. Was sagen Sie dazu als UI Designer?

Naja, irgendwann wird es zu klein um es zu bedienen, weil man mit den Fingern mit mehr hinkommt. Man kann es nicht beliebig klein machen.

Wie muss man denn dann die Grafik anpassen um das auszudrücken?

Das darf nicht gerade bis zum Nullpunkt gehen, sondern sollte bei nicht unter einen bestimmten Wert fallen hin zur y-Achse.

Dann haben Sie konstante Positionierungszeiten für das Mini-Handy. Ist das richtig?

Hmm... dann sollte es wahrscheinlich eher zur y-Achse hin ins Unendliche gehen.

Genau. Wenn Sie jetzt so ein doppelt so großes Handy bauen wollen, was müssen Sie da machen?

Einen Prototypen. In diesem Fall einen Hardware-Prototypen.

Dann haben wir uns Evaluationstechniken angesehen. Wie können Sie diesen Prototyp dann testen?

Erst habe ich es mit GOMS versucht, aber gemerkt dass es Quatsch ist. Zuerst würde ich ein „Literature Review“ machen, und gucken ob schonmal jemand versucht hat ein Handy doppelt so groß zu machen. Dann muss man es Benutzern in die Hand geben und sie ausprobieren lassen. [Er wollte wohl auf „Controlled Experiment“ raus, aber das habe ich nicht so genannt.]

Wie würden sie den Test mit den Benutzern denn dann machen?

Erstmal im Lab, nicht im Field. Und dann würde ich ihnen das Handy geben und die Tests aufzeichnen.

Würden Sie denn allen Usern alle Prototypen geben?

Kommt drauf an wie viele User ich zur Verfügung habe. Wenn es wenige sind müssen alle alles machen. Allerdings kann es dann zu Problemen mit Lerneffekten kommen.

Wie können Sie denn bei wenigen Usern trotzdem Lerneffekte verhindern?

Hmm...

Wenn ihr Chef sagt „Wir haben nicht so viel Geld!“?

Keine Ahnung.

Sie können Gruppen bilden und sie jeweils mit anderen Prototypen anfangen lassen. Was wäre denn dann eine Hypothese die sie da testen?

Dass sich das Handy genauso schnell bedienen lässt wie das normale?

Ich hätte es gerne etwas schärfer.

Dass es sich schneller bedienen lässt?

Noch schärfer.

Hmm...

Also quantifiziert, z.B. dass es 50ms schneller ist. O.K. das wars dann.

Introduction to Database Systems (Professor Schroeder)

Schroeder: Sie haben sich ja vor allem mit relationalen Datenbanken befasst. Wie entwirft man denn eine Datenbank?

Ich: Wir haben das ER-Modell kennengelernt, mit Entities, Relationships, Entities haben Attribute und man kann Aggregationen definieren.

Malen Sie doch mal so ein beliebiges ER-Diagramm hin.

[Ich habe ein Diagramm mit Professor – prüft – Student hingemalt, und beide haben sich amüsiert, dass immer das gleiche Diagramm kommt... passt halt zur aktuellen Situation.]

Ich habe die Matrikelnummer als „Key“ bezeichnet.

Sie sagten ja „Key“. Was ist denn ein Key?

Eine minimale Menge von Attributen, die ein Tupel eindeutig identifiziert.

Den Relationships kann man noch Zahlen zuordnen. Was würden Sie da in dem Fall machen, dass wie hier auch mehrere Professoren eine Prüfung machen können?

Dass wäre eine m:n Beziehung, da Sie ja auch mehrere Studenten prüfen.

Wenn Sie das ER-Modell jetzt in ein Relationenmodell übersetzen wollen, wie machen Sie das?

Beim ER-Modell gibt es Entities und Relationships, beim RD-Modell nur Relations, also muss man das übersetzen. Entities werden zu Relations/Tabellen in der Datenbank, und die Relationships je nachdem was es ist... [Einzelfälle erklärt]

Bei schlechten Schemata kann es zu Anomalien führen, welche gibt es da?

Insert, Update und Deletion Anomaly jeweils erklärt.

Was kann man dagegen tun? Wie müssen Schemata aussehen damit sie gut sind?

Da haben wir Normalformen kennengelernt. 1NF, 2NF 3NF erklärt. BCNF kann ich mir nie merken.

Warum können Sie sich denn BCNF nie merken?

In der Praxis strebt man eher 3NF an, weil die FDs in den einzelnen Relationen nicht verlorengehen.

Dann haben Sie auch Semi-strukturierte Modelle kennengelernt. Was gibt es da?

XML-Dokumente als Datenbasis und XQuery mit den FLWOR-Ausdrücken um auf die Daten zuzugreifen. Ausserdem haben wir noch RDF besprochen.

Was ist das?

Beschreibt Ressourcen in einer Form, Subjekt-Prädikat-Objekt, damit auch Computer die Semantik verstehen können. Wichtig für das Semantic Web z.B.

Dankeschön. Warten Sie bitte kurz draußen.

Literatur

Die Folien zu Designing Interactive Systems 1 und Web Engineering sind sehr gut strukturiert, enthalten Zusammenfassungen, Literaturhinweise, Grafiken, alles was man sich wünschen kann. So macht das Lernen Spaß, und die Lehrenden leisten ihren Beitrag zur Vermittlung der Materie.

Die Folien zu Introduction to Database Systems sind leider mangelhaft, viel Text, wenige Grafiken, einfache Zusammenhänge möglichst kompliziert dargestellt, und dann noch lückenhaft... hier hilft nur ein ordentliches Buch (Kemper, Eickler).

Neben den Folien, Prüfungsprotokollen, Wikipedia, Google, Links aus den Folien und anderen Internetseiten habe ich folgende Bücher benutzt:

The Design Of Everyday Things; Donald Norman

Ohne Kommentar, wer DIS1 hört weiss Bescheid... muss man haben.

Kompendium der Web-Programmierung; T. Walter

Ein Buch voller Beispiele, Screenshots, Code, zu den meisten Themen aus der Vorlesung Web Engineering. Auf die Theorie und Hintergründe der einzelnen Themen wird hier nicht oder nur sehr kurz eingegangen, ausserdem kommt der Teil Servlets/JSP hier für die Vorlesung zu kurz. Die vielen Beispiele haben mir aber durchaus geholfen.

Datenmodelle, Datenbanksprachen und DBMS; G. Vossen

Hier findet man die ganzen „Leerfolien“ aus der Vorlesung abgedruckt. Leider ziemlich trocken und theoretisch, aber wers mag... Ist anscheinend auch die Basis der Vorlesung, und es sind auch durchaus hilfreiche Beispiele drin. Die Teile XML und OODB sind hier nicht enthalten, ist halt schon was älter.

Datenbanksysteme; A. Kemper, A. Eickler

Weniger trocken als Vossen, mehr Beispiele, praktischer und moderner geschrieben und vor allem verständlicher. Hier sind auch die Teile XML, OODB enthalten. Ein paar theoretische Aspekte sind hier weggelassen, die meiner Meinung nach für eine praktische Diplomprüfung aber auch nicht relevant sind. Uneingeschränkt empfehlenswert!

Wie immer weitere Bücher, die zwar bei den Prüfungsthemen nicht weiterhelfen, dafür aber beim Lernen generell sehr hilfreich sind:

Effektives Lernen; H. Dahmer, J. Dahmer

Thema des Buchs ist zu lernen wie man lernt. Welche Formen des Lernens es gibt, wie gut man mit den verschiedenen Formen Gelerntes behält, wie man sich motiviert, und viel psychologisches Grundwissen zum Thema Lernen. Die Autoren stellen das Zettelkastenprinzip vor, was besonders gut ist zum Vokabellernen, also nicht unbedingt für diese Prüfung, aber auch bei den Prüfungsthemen gibt es „Vokabeln“, die man so sehr gut lernen kann.

Keine Angst vor Prüfungsangst; H. Knigge-Illner

Thema des Buchs sind Strategien zur optimalen Prüfungsvorbereitung. Diese Strategien helfen auch Menschen ohne Prüfungsangst, also nicht vom Titel abschrecken lassen. Besonders empfehlenswert sind die Teile Arbeitsplanung, Zeitmanagement und Strukturierendes Lernen. Ausserdem enthält das Buch ein komplettes Kapitel über Training für mündliche Prüfungen.