



Stichting NIOC

Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2025, gehouden op donderdag 27 maart 2025 jl. en georganiseerd door Hogeschool Windesheim). Bij elkaar zo'n 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op 18 maart 2027 in Arnhem en wordt georganiseerd door HAN University of Applied Sciences.

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

De clientkant van webapplicaties in het universitaire onderwijs

Auteurs

ir. S. Stuurman
Open Universiteit
Email: Sylvia.Stuurman@ou.nl

ir. H. Koppelman
Open Universiteit
Email: Herman.Koppelman@ou.nl

Samenvatting

We laten aan de hand van de geschiedenis van webapplicaties zien dat het verklaarbaar is dat er over het algemeen weinig aandacht wordt besteed aan de clientkant van webapplicaties in opleidingen in de informatica, en we geven argumenten waarom de ontwikkelingen in webapplicaties meer aandacht voor de clientkant rechtvaardigen. Tenslotte schetsen we de manier waarop we bij de Open Universiteit een cursus rond de clientkant van webapplicaties hebben ingevuld.

Trefwoorden

Clientkant van webapplicaties, onderwijs, programmeertalen, web development

De clientkant van webapplicaties in het universitaire onderwijs

1 **Introductie**

Bij een webapplicatie zijn meerdere systemen betrokken. Wanneer een gebruiker in een webbrowser een URL in de adresbalk typt of op een link klikt, wordt er een request verstuurd die (via één of meer nameservers) bij een webserver aankomt. De webserver kan dan een (statisch) HTML-bestand, met bijbehorende bestanden (stijl, script, en afbeeldingen bijvoorbeeld) naar de browser sturen. Een andere mogelijkheid is dat de webserver een script of programma start (eventueel via een applicatieserver), dat gegevens kan raadplegen (via een database of via webservices bijvoorbeeld), en uiteindelijk HTML genereert, die dan wordt verstuurd naar de browser.

De bestanden die bij de browser aankomen worden daar geïnterpreteerd. Zo wordt er een webpagina opgebouwd, en er kan een bij de webpagina behorend script actief zijn, dat reageert op een muisklik van de gebruiker of op andere events. Zo'n script kan de pagina en de elementen daarop als object benaderen en wijzigen, en kan zelf communiceren met de webserver. Het script kan gebruik maken van de op die manier verkregen gegevens om de pagina te wijzigen zonder dat de browser een nieuwe pagina hoeft binnen te halen. Het mechanisme waarbij een script met de webserver communiceert en de binnengekomen gegevens verwerkt in de huidige pagina (in plaats van het mechanisme dat er een nieuwe pagina wordt opgehaald) wordt aangeduid met de term Ajax.

De clientkant van webapplicaties bestaat uit alles dat in de browser wordt geïnterpreteerd, zoals HTML, CSS en JavaScript. Voor de clientkant doet de vraag of die bestanden statisch op de webserver aanwezig waren of dynamisch zijn gegenereerd niet ter zake.

In dit artikel zullen we laten zien hoe de geringe aandacht voor de clientkant van webapplicaties historisch verklaarbaar is, zullen we argumenteren waarom er in het universitaire curriculum wel aandacht voor de clientkant behoort te zijn, en zullen we laten zien hoe we die aandacht bij de faculteit Informatica van de Open Universiteit hebben ingevuld.

2 **Webapplicaties in de loop der tijd**

Het web is statisch begonnen, in de vorm van webserver (de server-kant) die HTML-pagina's naar webbrowsers (de client-kant) konden sturen. Dynamiek aan de serverkant (in de vorm van bijvoorbeeld scripting talen) en aan de clientkant (in de vorm van JavaScript) werd ongeveer in dezelfde periode mogelijk. Op het gebied van browsers was er een grote concurrentiestrijd aan de gang, waarbij elke browser een eigen JavaScript-variant ontwikkelde, en een eigen manier om een webpagina als object aan te bieden aan dergelijke client-side scripts. Aan de serverkant was er minder concurrentie, en bovendien vormt een verschil tussen webserver wat server-side oplossingen betreft geen probleem voor de ontwikkelaar van een webapplicatie: een ontwikkelaar bouwt een webapplicatie voor één specifieke webserver, namelijk, voor de server waarop de webapplicatie zal gaan draaien. Verschillen in technieken aan de clientkant vormen wel een probleem: een ontwikkelaar heeft over het algemeen niet in de hand in welke browsers een webpagina getoond zal worden. Een webapplicatie die op één server draait, zal dus moeten fungeren binnen liefst elke mogelijke webbrowser. De nadruk lag bij ontwikkelaars daarom lange tijd op de serverkant.

JavaScript werd vooral gebruikt door hobbyisten, die het gebruikten voor pop-ups of bewegende afbeeldingen. Scripts werden vaak via knippen en plakken geconstrueerd, met veel fouten.

Logica aan de clientkant is de laatste jaren belangrijker geworden, sinds de opkomst van Ajax. Steeds meer webapplicaties maken gebruik van Ajax om het de gebruiker gemakkelijker te maken: auto-completion bij zoekopdrachten, validatie van formulieren voordat de gegevens worden opgestuurd naar de server, een adres laten verschijnen als de gebruiker een postcode heeft ingevoerd, dat soort toepassingen wordt steeds algemener. Daarnaast worden er steeds meer complete applicaties aan de clientkant gebouwd. Er bestaan office suites die geheel in de browser draaien, aan de clientkant. De browser fungeert daarmee als besturingssysteem voor de webapplicatie. Een ander voorbeeld van dit soort toepassingen is webgebaseerde e-mail, waarbij het weergeven van mappen, de opmaak van een e-mail bericht, het opzoeken van adressen, zich aan de clientkant afspeelt, waarbij via Ajax met de server wordt gecommuniceerd. Het geeft de gebruiker dezelfde beleving als een desktop e-mail applicatie. Ook applicaties voor mobiele telefoons kunnen gebruik maken van de browser als besturingssysteem. De komende versie van HTML biedt allerlei extra's voor applicatie-ontwikkelaars, zoals een clientkant database.

De geschiedenis van de ontwikkeling van de technieken voor de clientkant, met een gebrekkige ondersteuning van standaarden, een wildgroei aan verschillen in browsers, en daardoor de onmogelijkheid om serieuze programma's te schrijven voor de clientkant, heeft er voor gezorgd dat de programmeertaal voor de clientkant, JavaScript, lange tijd is beschouwd als een speelgoedtaaltje. JavaScript werd op één hoop gegooid met HTML en CSS. De slechte kwaliteit van de code die geproduceerd werd door mensen zonder enige opleiding in de informatica was daar mede debet aan, en zo werd het een zichzelf versterkend proces.

De clientkant van webapplicaties is daardoor buiten het gezichtsveld van opleidingen in de informatica gevallen, terwijl het daar wel in thuishoort.

3 De clientkant in het curriculum

Met die ontwikkeling van serieuze applicaties aan de clientkant zijn de eisen aan de clientkant veranderd. De basistechnieken daarvoor zijn HTML, CSS, JavaScript, de DOM en Ajax. De HTML moet netjes en foutloos zijn, waar HTML historisch gezien vaak veel fouten bevat. De scripts aan de clientkant worden steeds groter, en moeten foutloos zijn, en onderhoudbaar. Er zijn ontwikkelaars nodig die op een gedegen manier software voor de clientkant kunnen ontwikkelen, met kennis van alle van toepassing zijnde technieken. Een eerste argument om aandacht te besteden aan de clientkant van webapplicaties in het universitaire curriculum is dus de behoefte die er is ontstaan aan goed opgeleide ontwikkelaars met kennis op dat gebied. Er zijn meer argumenten:

Er is een specialisatie aan het ontstaan binnen de software engineering: web engineering. De reden daarvoor is dat web applicaties inherent lastig zijn: ze bestaan uit een veelvoud aan technieken en het gaat om gedistribueerde applicaties, met code die op de server draait, code die op de client draait, een database, en onderlinge afhankelijkheden. Web engineering zal belangrijker worden omdat goed onderhoudbare webapplicaties op een juiste manier moeten zijn opgezet. Een web engineer moet kennis hebben van alle aspecten van webapplicaties, en dus ook van de clientkant. Een web engineer moet ook toekomstige ontwikkelingen in technieken kunnen duiden, en moet mensen kunnen aansturen in het juiste gebruik van standaarden.

De clientkant van webapplicaties biedt een terloopse manier om studenten kennis te laten maken met een aantal aspecten die later binnen het curriculum in detail bestudeerd kunnen worden. Webapplicaties kennen bijvoorbeeld een inherent asynchroon communicatie-aspect. Webapplicaties zijn voorbeelden van een gedistribueerd systeem.

JavaScript is een bijzondere programmeertaal, die op veel punten verschilt van bijvoorbeeld Java of C#. Via JavaScript maakt de student al in een vroeg stadium in de studie kennis met andere programmeerparadigma's: eventgebaseerd programmeren, functioneel programmeren, prototypegebaseerd programmeren. Later in het curriculum kunnen deze concepten worden uitgediept.

JavaScript is een dynamisch getypeerde taal is, en laat daardoor programmeerfouten toe die een compiler van een statisch getypeerde taal zou afstraffen. Dat betekent aan de ene kant dat JavaScript niet geschikt is als eerste programmeertaal, maar aan de andere kant dat het de gelegenheid biedt om studenten te leren zelf bewust dat soort fouten te vermijden.

4 De cursus Webapplicaties: de clientkant bij de Open Universiteit

De cursus 'Webapplicaties: de clientkant' is een cursus van 4.3 ects, in de propedeuse van de bachelor Informatica. Als voorkennis is kennis over Java vereist: studenten moeten weten wat variabelen zijn, moeten klassen en methoden kunnen schrijven, en kunnen werken met objecten.

De leerdoelen van de cursus zijn als volgt.

We verwachten dat een student na het bestuderen van de cursus:

- de basiselementen van de standaarden en technieken (X)HTML, CSS, JavaScript, DOM en Ajax kan noemen
- deze kan toepassen voor het ontwikkelen van webapplicaties aan de clientkant
- de technieken met elkaar kan vergelijken en kan uitleggen wat de voordelen, nadelen, mogelijkheden en beperkingen van de genoemde technieken zijn
- relevante software-engineering principes kan toepassen bij het creëren van clientkant webapplicaties
- de voordelen kan uitleggen om daarbij te werk te gaan volgens webstandaarden
- de consequenties kan uitleggen van het feit dat webapplicaties op een sterk uiteenlopende verzameling (en niet van te voren geheel gekende) platforms gebruikt zullen worden
- enkele principes voor het maken van gebruikersvriendelijke user interfaces kan noemen en toepassen.

Lange tijd was het onmogelijk om webpagina's netjes te schrijven volgens de standaarden van het World Wide Web Consortium: de gebrekkige ondersteuning van browsers liet dat niet toe. Daardoor is er een traditie ontstaan van webpagina's waarbij structuur en presentatie niet netjes gescheiden zijn, waarbij bijvoorbeeld HTML-tabellen, bedoeld voor gestructureerd tonen van gegevens, gebruikt worden voor de layout. Veel tekstboeken maken zich aan dat soort praktijken schuldig.

HTML en CSS vormen niet het hoofdbestanddeel van een vak over de clientkant van webapplicaties, maar omdat de user-interface van zo'n webapplicatie geschreven wordt in HTML en CSS, en pagina-elementen die daarmee worden gebouwd, met behulp van JavaScript kunnen worden gemanipuleerd, is het wel nodig dat de student ook kort kennismaakt met HTML en CSS. Dat moet uiteraard op een manier gebeuren waarin recht gedaan wordt aan de bestaande standaarden, en waarin er, zoals dat hoort, een scheiding wordt aangebracht tussen HTML voor structuur, en CSS voor presentatie en layout.

Eenzelfde ontwikkeling heeft zich voorgedaan in de manier waarop JavaScriptcode wordt gebruikt: lange tijd was het de gewoonte om JavaScriptcode binnen HTML te gebruiken, maar sinds browsers de standaarden ondersteunen is het mogelijk om alle JavaScriptcode in een apart bestand te houden, zodat structuur (HTML), presentatie (CSS) en gedrag (JavaScript) netjes gescheiden blijven.

We hebben een Engelstalig tekstboek gevonden dat zich op de juiste manier aan de standaarden houdt: *Web programming, Step by Step*, door Marty Stepp, Jessica Miller en Victoria Kirst. Daarbij hebben we een werkboek geschreven dat extra uitleg biedt, dieper ingaat op JavaScript en webstandaarden, ingaat op HTML5, de nadruk legt op software engineering principes bij het programmeren, en waarin extra opdrachten worden gegeven.

Het aanbieden van de stof gaat gepaard met veel voorbeelden en kleine opdrachten. Er wordt veel aandacht besteed aan het juiste gebruik van webstandaarden, en we leggen uit hoe de student software engineering principes kan gebruiken bij het bouwen van clientkant webapplicaties, zoals het scheiden van verantwoordelijkheden en het principe van modulariteit.

Dat is ook de manier waarop we het hoofd willen bieden aan de snelle veranderingen op dit gebied. Met de nadruk op standaarden, op software engineering principes en op taalaspecten, leggen we de nadruk op aspecten van de clientkant van webapplicaties die het minst aan verandering onderhevig zijn. Door zelf een aantal ontwikkelingen te schetsen en te laten zien hoe die te plaatsen zijn in de geschiedenis van webapplicaties, geven we studenten de mogelijkheid zelf toekomstige ontwikkelingen te duiden.

5 **Cursus in vijf blokken**

Het eerste blok geeft de context aan: internet en het World Wide Web.

Het tweede blok heet “De voorkant van het web: het statische gedeelte”. Hierin komen HTML en CSS aan bod. Daarbij leiden we de student niet op tot CSS-expert: dat gedeelte van het bouwen van webapplicaties zal over het algemeen aan specialisten worden overgelaten. We laten de student wel zien hoe CSS werkt: een student zal kunnen begrijpen wat het effect is van CSS-regels.

Het derde blok is getiteld “De voorkant van het web: programmeren in de browser”. Het blok geeft een inleiding in JavaScript, waarbij ook kort aan bod komt dat er functioneel geprogrammeerd kan worden met JavaScript, en waarin de student ook iets te zien krijgt over het feit dat JavaScript prototype-gebaseerd is. In dit blok maakt de student ook kennis met het Document Object Model: de webpagina en alle elementen daarvan, in de vorm van een geheel van objecten. Tenslotte komen events aan bod, en wordt Prototype geïntroduceerd als voorbeeld van een library.

Het vierde Blok heet “Clientkant webapplicaties”. De kennis over JavaScript wordt uitgebreid met Ajax, en studenten zijn vanaf dat moment in staat om volwaardige clientkant webapplicaties te schrijven. Er wordt daarnaast aandacht besteed aan het gebruiken van externe bronnen in webapplicaties, en de geschiedenis van de webstandaarden passeert de revue.

Het laatste blok geeft aandacht aan usability aspecten van webapplicaties, en geeft informatie over waar de term Web 2.0 voor staat.

Overal is geprobeerd de student op zo'n manier inzicht te geven, dat het gemakkelijker wordt om toekomstige ontwikkelingen te duiden: de voorkant van webapplicaties is een terrein waarin de ontwikkelingen snel gaan. Met de keuze voor webstandaarden, en voor het aanduiden van algemene principes, hebben we getracht daarin de meest vaste lijn te kiezen.

Voor meer informatie kunt u terecht op:

<http://portal.ou.nl/web/webapplicaties-de-clientkant/cursusinformatie>