

Externe bijlagen

*(tussen)producten van het afstudeerproject
“Een gebruiksvriendelijke manier voor het bewerken
van XML-elementen in FontoXML”*

<i>Afstudeerder:</i>	Youri Bosselaar
<i>Studentnummer:</i>	08033153
<i>Bedrijf:</i>	FontoXML (onderdeel van Liones B.V.)
<i>Plaats:</i>	Rijswijk
<i>Examinator:</i>	Chris Heydra
<i>2^e examiner:</i>	Ru Kleijn
<i>Bedrijfsmentor:</i>	Vincent Smedinga
<i>Inleverdatum:</i>	9 januari 2015



FONTOXML

Inhoudsopgave

A.	Afstudeerplan	5
B.	Plan Van Aanpak	14
C.	Onderzoeksrapport XML bewerking	34
D.	Gebruikersonderzoek	58
E.	Benchmark	68
F.	Systeemeisen	96
G.	Ontwerprapport	106
H.	Screenshots prototype	156
I.	Testplan	166
J.	Testrapport	188
K.	Adviesrapport	230
L.	Verslag bedrijfsbezoek	242
M.	Beoordeling TTA	246

Bijlage A

Afstudeerplan



Afstudeerplan

Informatie afstudeerder en gastbedrijf (*structuur niet wijzigen*)

Afstudeerblok: 2014-2.1 (start uiterlijk 1 september 2014)
Startdatum uitvoering afstudeeropdracht:
Inleverdatum afstudeerdossier volgens jaarrooster: 9 januari 2015

Studentnummer: 08033153
Achternaam: Bosselaar
Voorletters: Y
Roepnaam: Youri
Adres: Reijerskoop 139
Postcode: 2771BH
Woonplaats: Boskoop
Telefoonnummer: 0172 212678
Mobiel nummer: 0616654760
Privé emailadres: ybosselaar@gmail.com

Opleiding: CMD
Locatie: Den Haag
Variant: Voltijd

Naam studieloopbaanbegeleider: Berber Hartman
Naam begeleidend examinerator: Chris Heydra
Naam tweede examinerator: Ru Kleijn

Naam bedrijf: FontoXML
Afdeling bedrijf: User Experience
Bezoekadres bedrijf: Polakweg 7
Postcode bezoekadres: 2288GG
Postbusnummer:
Postcode postbusnummer:
Plaats: Rijswijk
Telefoon bedrijf: 070 3191923
Telefax bedrijf: 070 3193825
Internetsite bedrijf: www.liones.nl

Achternaam opdrachtgever: Benedictus

Voorletters opdrachtgever:	J.N.
Titulatuur opdrachtgever:	ir.
Functie opdrachtgever:	Directeur
Doorkiesnummer opdrachtgever:	
Email opdrachtgever:	jan.benedictus@liones.nl
Achternaam bedrijfsmentor:	dhr. Smedinga
Voorletters bedrijfsmentor:	V.M.
Titulatuur bedrijfsmentor:	ir.
Functie bedrijfsmentor:	User Experience Lead
Doorkiesnummer bedrijfsmentor:	
Email bedrijfsmentor:	vincent.smedinga@liones.nl
Doorkiesnummer afstudeerder:	
Functie afstudeerder (deeltijd/duaal):	
Titel afstudeeropdracht:	
"Ontwerp interfacepatronen voor gelijksoortige bewerkingen binnen een XML-editor"	

Opdrachtschrijving

1. Bedrijf

FontoXML is een bedrijf dat zich bezighoudt met de ontwikkeling van een online XML-editor: een applicatie waarmee XML-bestanden kunnen worden bewerkt. XML is een coderingstaal die semantiek toevoegt aan teksten. Het lijkt een beetje op HTML maar is vele malen uitgebreider dan dat. FontoXML is een product van internetbureau Liones in Rijswijk. Het team bestaat dus volledig uit mensen van Liones (ongeveer vijftien). FontoXML bestaat ongeveer een jaar en bevindt zich in de ontwikkelfase.

FontoXML wil zich onderscheiden van de concurrentie door zich te richten op mensen met veel vakkenis, maar weinig kennis van XML: zogenaamde "Subject Matter Experts (SME's)". Deze mensen zijn vaak erg belangrijk in het proces van contentcreatie, omdat zij de kennis hebben die gepubliceerd moet worden. Echter hebben zij vaak geen verstand van XML, waardoor het voor hen moeilijk is om semantische content aan te leveren. FontoXML wil hier een handvat bieden door een gebruiksvriendelijke applicatie te maken die weinig XML-kennis vereist. Veel concurrerende applicaties zijn erg complex en vereisen veel technische kennis. Doordat deze ontwikkeld zijn door zeer technisch onderlegde mensen, zijn deze applicaties meer gericht op functionaliteiten dan op gebruiksvriendelijkheid. Dit is dan ook de reden dat FontoXML kansen ziet in de gekozen positionering en doelgroep.

Het bedrijf valt binnen de b2b-markt en richt zich vooral op uitgevers van zeer grote hoeveelheden content. Hierbij kan worden gedacht aan uitgevers, maar ook bijvoorbeeld aan machinefabrikanten, die grote en uitgebreide handleidingen moeten schrijven. Binnen deze bedrijven wordt XML gebruikt voor het organiseren, herbruikbaar maken en doorzoekbaar maken van hun content. Daarnaast kunnen er voor XML-content standaarden worden ingesteld voor publicatie. Zo worden er bijvoorbeeld verschillende stijlgidsen gebruikt voor het opmaken van de content op print, web of mobiel. Hierdoor hoeft niet ieder document opnieuw te worden opgemaakt, wat bedrijven dus veel tijd kan schelen.

2. Probleemstelling

Het kost momenteel veel tijd om voor het bewerken van de verschillende elementen in een XML-schema de juiste interface te kiezen. Voor ieder los element moet door het team van FontoXML worden gekeken wat de beste manier van bewerken is en welke interfacepatronen hier bijhoren. Mede hierdoor is het implementeren van FontoXML bij nieuwe klanten nog behoorlijk tijdrovend en daardoor duur. Het kost dus nog teveel maatwerk om een interface van FontoXML te configureren.

3. Doelstelling van de afstudeeropdracht

Het team van FontoXML wil op een snelle manier de interface van de applicatie kunnen configureren bij een implementatie met een nieuw XML-schema. Om dit te doen, moeten er groepen XML-elementen gemaakt worden. Alle elementen binnen zo'n groep moeten met het zelfde interactiepatroon bewerkt kunnen worden. Het doel is hier om tot een set schema-onafhankelijke groepen te komen. Deze groepen zijn niet zozeer gebaseerd op de elementen zelf, maar op de bewerkingstaken die daarbij horen. Voor elk van deze groepen moet dan ook een gebruiksvriendelijk interactiepatroon worden ontworpen. Op deze manier kan de

gebruiksvriendelijkheid van de interface van FontoXML gewaarborgd worden zonder dat er voor iedere implementatie een enorme hoeveelheid maatwerk nodig is.

4. Resultaat

Na voltooiing van mijn opdracht beschikt het team van FontoXML over een set XML-elementgroepen. Voor minimaal drie van deze groepen is een interactiepatroon ontworpen waarmee de elementen in deze groep gebruiksvriendelijk bewerkt kunnen worden. Dit patroon ligt vast in een interactieontwerp, testrapport en een high-fidelity prototype. Omdat er meerdere patronen ontworpen worden, zal in een adviesrapport worden uitgelegd welke het meest geschikt zijn voor gebruik in de applicatie.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

Om een beeld te kunnen scheppen alle werkzaamheden die tijdens dit project zullen worden uitgevoerd, is hier een kort overzicht te vinden. De werkzaamheden zijn gegroepeerd binnen de verschillende projectfasen van Roel Grit. Deze bieden houvast in het proportioneel verdelen van de activiteiten over het hele tijdsbestek van 70 dagen.

Fase 1: Oriëntatiefase (11 dagen)

A. Plan van aanpak schrijven (1 dag)

Het plan van aanpak biedt houvast en richting voor het gehele project. Hierin staat ook de planning. Verder wordt de scope van het project afgebakend.

B. Oriëntatie eerder verricht onderzoek (3 dagen)

Er zijn al twee eerdere afstudeerprojecten geweest die te maken hadden met de gebruiksvriendelijkheid van FontoXML. Ondanks het feit dat deze opdrachten een ander onderwerp hadden dan mijn opdracht, zouden de inzichten mij kunnen helpen in het nadenken en ontwerpen. Daarom wil ik de resultaten van deze projecten bekijken en analyseren om te kijken of er voor mij bruikbare resultaten zijn. Daarnaast is het vooronderzoek voor deze projecten (doelgroeponderzoeken, persona's) bruikbaar voor mijn project omdat het dezelfde doelgroep betreft. Aangezien ten minste één van de twee projecten al is voltooid met een voldoende door CMD Den Haag, zal ik dit onderzoek als toereikende, betrouwbare bron beschouwen.

C. Oriëntatie en inventarisatie XML-elementen (5 dagen)

Bekend worden met de verschillende soorten XML-elementen. Welke taken bestaan er allemaal bij het bewerken van XML? Welke elementen moeten op dezelfde manier bewerkt worden? Om dit te ontdekken moeten eerst de mogelijke eigenschappen van elementen worden geïnventariseerd. Vervolgens kunnen deze elementen gegroepeerd worden op de bewerkingen die met de elementen mogelijk of vereist zijn.

Deze inventarisatie moet over minimaal twee schema's gedaan worden om te waarborgen dat de groepen over meerdere schema's toepasbaar zijn. Voor deze schema's zal ik kiezen voor TEI en DITA. Deze twee schema's zijn gebruikt bij de eerste twee klanten van FontoXML en dus goede concrete voorbeelden.

D. XML-elementgroepen definiëren (2 dagen)

Als alle eigenschappen van de elementen in kaart zijn gebracht kunnen er groepen worden gemaakt. Deze groepen vormen de basis voor de ontwerpen.

E. Uitvoeren gebruikersonderzoek (2 dagen)

Om een goed beeld te krijgen van hoe de applicatie daadwerkelijk gebruikt wordt, zal ik een aantal interviews met gebruikers houden. Hierin vraag ik naar het gebruik van de applicatie, om zo een basis te kunnen vormen voor het opstellen van use cases.

Fase 2: Ontwerpfase (26 dagen)

E. Opstellen gebruiksscenario's. (2 dagen)

Om beter te kunnen begrijpen hoe de verschillende bewerkingen plaats zullen vinden, wordt er een aantal gebruiksscenario's opgesteld. Hierin wordt een combinatie van taken en bewerkingen beschreven zoals deze bij het daadwerkelijke gebruik voor zal komen.

F. Uitvoeren taakanalyse per elementgroep. (3 dagen)

Om een interfacepatroon te kunnen ontwerpen voor de verschillende elementgroepen, moeten de taken en stappen die bij een bewerking horen worden geanalyseerd. Op deze manier bij het ontwerpen rekening worden gehouden met de verschillende routes en volgorden tijdens het uitvoeren van een taak.

G. Richtlijnen voor de interface opstellen (2 dagen)

Door van tevoren een aantal richtlijnen op te stellen kan er sturing worden gegeven bij het ontwerpen. Deze richtlijnen kunnen bijvoorbeeld te maken hebben met het wel of niet onderbreken van de gebruiker in zijn workflow bij verplichte stappen.

H. Onderzoek interfacepatronen voor richtlijnen (2 dagen)

Aan alle richtlijnen kan een aantal bestaande interfacepatronen worden gekoppeld die mogelijk zouden kunnen worden ingezet in de applicatie. Deze vormen een basis bij het ontwerpen van varianten hierop of eventuele nieuwe patronen.

I. Ontwerpen van interfacepatronen (20 dagen)

Aan de hand van richtlijnen, eerder ontworpen interfacepatronen en onderzochte interfacepatronen kunnen er nieuwe ontwerpen worden gemaakt. Deze zullen worden geschetst en vervolgens digitaal uitgewerkt om zo duidelijk mogelijk te communiceren hoe de verschillende bewerkingen met het systeem kunnen worden

uitgevoerd. Voor elke uitgewerkte elementgroep worden maximaal drie alternatieve patronen ontworpen die vervolgens getest kunnen worden.

Fase 3: Realisatiefase (15 dagen)

J. Prototypes bouwen (15 dagen)

Om de ontworpen interacties zo goed mogelijk te kunnen testen zal er een high-fidelity prototype worden ontwikkeld in HTML, CSS en Javascript. Dit kan gebeuren op basis van de bestaande applicatie, zodat er een zo realistisch mogelijke testomgeving is. Een werkend prototype is zeer belangrijk omdat dit de meest representatieve vorm van testen mogelijk maakt.

Fase 4: Testfase (7 dagen)

K. Opstellen testplan (3 dagen)

Voor een succesvolle test moeten er eerst onder andere doelen, methoden en scenario's worden bepaald. Deze punten en meer worden vastgelegd in een testplan.

L. Testen van interfacepatronen op gebruiksvriendelijkheid (1 dag)

Om te bepalen of mijn ontwerpen daadwerkelijk werken en gebruiksvriendelijk zijn moeten deze getest worden op echte gebruikers.

M. Opstellen Testrapport (3 dagen)

In het testrapport worden de resultaten van de usabilitytest verwerkt en geanalyseerd. Deze analyse zal bepalen welke patronen het beste geschikt zijn voor gebruik binnen FontoXML. Hiermee wordt de basis voor het adviesrapport gevormd.

Fase 4: nazorgfase (7 dagen)

N. Opstellen adviesrapport (7 dagen)

Om naar het team van FontoXML te communiceren welke interfacepatronen het meest geschikt zijn voor iedere elementgroep, zal een adviesrapport worden gemaakt. Hierin staat, op basis van bevindingen in de usabilitytest, een advies over de meest geschikte interface voor FontoXML.

Geen officiële projectfase: afstudeerdossier (15 dagen)

O. Opstellen afstudeerdossier (15 dagen)

Deze 15 dagen zullen niet allemaal aan het einde van het project plaatsvinden, maar verspreid over de tijd. In het afstudeerdossier worden mijn keuzes, gedachten en activiteiten vastgelegd, verklaard en geëvalueerd zodat er vanuit school beoordeeld kan worden of de opdracht op niveau is uitgevoerd.

6. Op te leveren (tussen)producten

Voortkomend uit de activiteiten in hoofdstuk 5 zijn de volgende producten:

- A. Plan van aanpak (*document*)
- B. Classificatie XML-elementen (*spreadsheet*)
- C. Set XML-elementgroepen (*document – eindproduct*)
- D. Ontwerprapport (*document- eindproduct*) met daarin:
 - a) Gebruiksscenario's
 - b) Taakanalyse diagrammen
 - c) Taakanlyse beschrijvingen
 - d) Richtlijnen voor interfaceontwerp
 - e) Interactieschetsen
- E. High-fidelity prototypes (*webpagina*)
- F. Testplan (*document*)
- G. Testverslag (*document + vastgelegde audio en/of video*)
- H. Testrapport (*document*)
- I. Adviesrapport (*document- eindproduct*)

7. Te demonstreren competenties en wijze waarop

Bij dit project komen de volgende CMD-competenties aan de orde:

Opstellen plan van aanpak

Het plan van aanpak is geslaagd wanneer hierin de planning, projectgrenzen, risico's, afspraken en methoden en technieken goed zijn vastgelegd.

Opstellen testplan & testrapportage

Voordat de usabilitytests worden uitgevoerd zal er een testplan worden opgesteld. Achteraf wordt er een testrapport opgemaakt met daarin de resultaten van de test. Deze documenten zullen bij het eindproduct worden gevoegd ter beoordeling.

Opstellen interaction design

Het opstellen van een interaction design vormt de basis voor dit project. Het is de bedoeling om een aantal gebruiksvriendelijke interfacepatronen te ontwerpen. Deze competentie kan worden bewezen in het ontwerprapport en de high-fidelity prototypes.

Usability test opzetten en uitvoeren

De usabilitytest is een zeer belangrijk onderdeel van dit project. Er moeten hierbij verschillende prototypes worden getest met echte gebruikers, zodat er een representatief oordeel kan worden geveld. Het slagen van deze test is op te maken uit het testrapport en het testverslag.

Uitwerken prototype

Voor het high-fidelity prototype zal gebruik gemaakt worden van HTML5, CSS3 en de Javascript-frameworks jQuery en AngularJS. Daarnaast moet responsive design worden toegepast om verschillende schermformaten goed te kunnen bedienen.

Bijlage B

Plan Van Aanpak

B

Plan van aanpak

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Datum: 7 september 2014



Inhoudsopgave

1. Achtergronden	17
1.1. Opdrachtgever	17
1.2. De applicatie: FontoXML	17
1.2.1. <i>Het nut van XML-documenten</i>	17
1.2.2. <i>De positionering van FontoXML</i>	18
1.3. Context en aanleiding	18
2. Projectresultaat	19
2.1. Doelstelling	19
2.2. Eindproducten	19
3. Projectactiviteiten	20
4. Tussenresultaten	24
5. Kwaliteit	26
6. Projectorganisatie	27
6.1. Organisatie binnen de projectgroep	27
6.1.1. <i>Leden afstudeerproject</i>	27
6.1.2. <i>Contactpersonen projectteam FontoXML</i>	27
6.2. Stakeholders	28
6.2.1. <i>De student</i>	28
6.2.2. <i>De Product Owner</i>	28
6.2.3. <i>De Lead Designer</i>	29
6.2.4. <i>Het developmentteam</i>	29
6.2.5. <i>De klant</i>	29
6.2.6. <i>Vakexperts - de gebruikers</i>	29
6.3. Documentbeheer en communicatie	29
7. Planning	31
8. Risico's	32
8.1 Interne risico's	32
8.2 Externe risico's	32
Bijlage I: Stokenplanning	33

1. Achtergronden

Om een beter begrip te krijgen van de situatie waarin het project zich afspeelt, wordt in dit hoofdstuk wat achtergrondinformatie verstrekt. Hierbij gaat het vooral om het bedrijf waar bij het project wordt uitgevoerd, het product waar het project zich op richt en de betrokken partijen.

1.1. Opdrachtgever

Dit project wordt uitgevoerd bij Liones B.V., een internetbureau gevestigd in Rijswijk. Bij Liones zijn ongeveer 35 mensen werkzaam. Zo'n vijftien daarvan houden zich actief bezig met de productontwikkeling van FontoXML, een webapplicatie voor het bewerken van XML-documenten. Het team bestaat daarbij uit twee ontwerpers, negen programmeurs en twee mensen voor management en sales. Daarnaast is er het afgelopen jaar altijd een afstudeerder actief geweest op het gebied van gebruiksvriendelijkheid. Er wordt door het team gewerkt met de Scrum-methode, een agile-vorm van productontwikkeling waarbij in korte perioden van 1 tot 2 weken steeds opnieuw de prioriteiten van het project worden bepaald. Hierdoor kunnen nieuwe inzichten gemakkelijk worden opgenomen in het project. De product owner van FontoXML is Jan Benedictus. Samen met het team bepaalt hij de prioriteiten binnen het project.

1.2. De applicatie: FontoXML

Het gehele project draait dus om de interface van webapplicatie FontoXML, waarmee XML-documenten kunnen worden bewerkt. Deze applicatie bestaat momenteel ongeveer een jaar en is dus nog in ontwikkeling.

1.2.1. Het nut van XML-documenten

XML is een coderingstaal die wordt gebruikt om informatie op te slaan over de inhoud van een tekstdocument. De variatie van informatie die hierin wordt opgeslagen loopt sterk uiteen. Zo kun je in XML semantiek aanbrengen in teksten: je kunt bijvoorbeeld aangeven dat de naam "Mozart" een achternaam is. Ook kun je structuur aanbrengen (paragrafen en lijsten bijvoorbeeld), opmaakregels meegeven (aangeven dat een zin scheef gedrukt moet worden) en documentinformatie opslaan (wie zijn de auteurs, wanneer is het document gemaakt). Uiteindelijk zorgt XML er dus voor dat teksten gemakkelijker herbruikbaar en doorzoekbaar zijn. Daarnaast is het mogelijk om aan te geven hoe bepaalde soorten informatie er uit moeten te komen zien wanneer zij gepubliceerd worden op verschillende media. Zo hoeft niet ieder document apart opgemaakt te worden maar bepaalt de semantiek direct ook de opmaak.

1.2.2. De positionering van FontoXML

Het belangrijkste idee van FontoXML is dat het inspeelt op een geheel nieuwe positie binnen het creatieproces van XML-content. In het klassieke geval wordt de content hier geschreven door vakexperts zonder kennis van XML. Deze content wordt dus aangeleverd in “platte” tekst, bijvoorbeeld een Microsoft Word-document. Vervolgens zet een redacteur deze tekst om naar XML. Om de snelheid en kwaliteit van dit proces te bevorderen, zet FontoXML zich volledig in op het maken van een gebruiksvriendelijke applicatie:

“FontoXML is all about user-friendly authoring of valid XML”

- www.fontoxml.com/features

Het idee hierbij is dat de vakexperts direct in FontoXML gaan werken, in plaats van bijvoorbeeld in Word. Op deze manier schrijft de auteur zijn content rechtstreeks in XML. Ook kan hij bijdragen aan de semantische verrijking van het document. De reden dat vakexperts hun documenten momenteel nog niet in XML schrijven, is omdat de huidige applicaties niet gebruiksvriendelijk zijn voor mensen zonder XML-kennis. Dit is dan ook precies waar FontoXML op in hoopt te spelen.

1.3. Context en aanleiding

De aanleiding van dit project is het huidige ontwerpproces voor nieuwe interface-patronen binnen de applicatie. Vanwege veel variaties in XML-schema's ligt er momenteel nog veel ontwerpwerk bij iedere klant-implementatie van FontoXML. Om dit werk te verminderen, is het team op zoek naar een aantal interface-patronen die voor het gros van de elementen gebruikt kan worden. Omdat gebruiksvriendelijkheid gebruiksvriendelijkheid het primaire aspect is waarmee FontoXML zich wil onderscheiden van de concurrentie, is een CMD-er de geschikte kandidaat voor het uitvoeren van deze opdracht.

Dit project is geen direct vervolg op een eerder project, maar er zijn wel projecten waaruit adviezen kunnen worden meegenomen bij het maken van ontwerpkeuzes. CMD-alumnus Wybe Minnebo heeft in zijn project interactieontwerpen gemaakt voor het meegeven van semantiek aan documenten voor het IAEA. CMD-student Mike Zuidgeest is momenteel zijn onderzoek naar de visualisatie van semantiek in complexe documenten aan het afronden.

2. Projectresultaat

Om te kunnen bepalen wanneer het project “af” is, is het belangrijk om vast te leggen wat het eindresultaat van het project moet zijn. Belangrijk daarbij is dat de doelstelling is behaald en dat alle eindproducten zijn opgeleverd. Vandaar dat deze twee punten in dit hoofdstuk zullen worden besproken.

2.1. Doelstelling

Het team van FontoXML wil op een voorspelbare manier de interface van de applicatie kunnen configureren bij een implementatie met een nieuw XML-schema. Om dit te doen, moeten er groepen XML-elementen gemaakt worden. Alle elementen binnen zo’n groep moeten met het zelfde interactiepatroon bewerkt kunnen worden. Het doel is hier om tot een set schema-onafhankelijke groepen te komen. Deze groepen zijn niet zozeer gebaseerd op de elementen zelf, maar op de bewerkingstaken die daarbij horen. Voor elk van deze groepen moet dan ook een gebruiksvriendelijk interactiepatroon worden ontworpen. Op deze manier kan de gebruiksvriendelijkheid van de interface van FontoXML gewaarborgd worden zonder dat er voor iedere implementatie een enorme hoeveelheid maatwerk nodig is.

2.2. Eindproducten

Het eindproduct voor dit project bestaat uit twee delen: een adviesrapport en een aantal prototypes. Het adviesrapport is bedoeld om gemaakte keuzes uit te leggen en te onderbouwen. De prototypes helpen daarbij met het illustreren en communiceren van de oplossing.

1. **Adviesrapport (*document*)**

Het adviesrapport is gebaseerd op zowel het theoretisch onderzoek voor de ontwerpen als op de resultaten van de usabilitytest. In het adviesrapport wordt aan FontoXML overgedragen welke ontwerpen het meest geschikt zijn voor gebruik in de applicatie.

2. **High fidelity prototypes (*GIT branch*)**

Als ondersteuning van het adviesrapport wordt een aantal prototypes opgeleverd. Deze prototypes zijn een eindproduct omdat zij de vormgeving en interacties van de ontworpen interfacepatronen het duidelijkst kunnen communiceren. Immers, een werkende, interactieve realisatie van een ontwerp zal duidelijker zijn dan een ontwerp op papier (hoe gedetailleerd dan ook). Alleen de prototypes die in het adviesrapport worden aanbevolen zullen als eindproduct worden opgeleverd. De prototypes zijn toegankelijk als branch via het versiebeheersysteem GIT, waarmee FontoXML werkt.

3. Projectactiviteiten

Door alle uit te voeren activiteiten vast te leggen, wordt er een helder beeld geschept van de omvang en complexiteit van het gehele project. Daarnaast vormen de activiteiten de basis voor de uitgebreide planning (zie hoofdstuk 7) en het bepalen van de op te leveren (tussen)producten bij het project.

A. Plannen van het project

Om houvast te creëren en te bepalen wat er tijdens het project allemaal moet gebeuren, moet er een planning worden gemaakt. Hierbij wordt ook bepaald hoeveel tijd iedere activiteit mag gaan kosten en wanneer het klaar moet zijn. Verder moeten de grenzen van het project worden afgebakend om helderheid te scheppen en meerwerk te vermijden. Uit deze activiteit komt het plan van aanpak voort.

B. Oriëntatie eerder verricht onderzoek

Er zijn al twee eerdere afstudeerprojecten geweest die te maken hadden met de gebruiksvriendelijkheid van FontoXML. Ondanks het feit dat deze opdrachten een ander onderwerp hadden dan mijn opdracht, zouden de inzichten mij kunnen helpen in het nadenken en ontwerpen. Daarom worden de resultaten van deze projecten bekeken en geanalyseerd om te zien of er bruikbare resultaten zijn. Daarnaast is het vooronderzoek voor deze projecten (doelgroeponderzoeken, persona's) bruikbaar voor mijn project omdat het dezelfde doelgroep betreft. Aangezien ten minste één van de twee projecten al is voltooid met een voldoende door CMD Den Haag, zal dit onderzoek als toereikende, betrouwbare bron worden beschouwd.

C. Oriëntatie en inventarisatie XML-elementen

Bekend worden met de verschillende soorten XML-elementen. Welke taken bestaan er allemaal bij het bewerken van XML? Welke elementen moeten op dezelfde manier bewerkt worden? Om dit te ontdekken moeten eerst de mogelijke eigenschappen van elementen worden geïnventariseerd. Vervolgens kunnen deze elementen gegroepeerd worden op de bewerkingen die met de elementen mogelijk of vereist zijn.

Deze inventarisatie moet over minimaal twee schema's gedaan worden om te waarborgen dat de groepen over meerdere schema's toepasbaar zijn. Voor deze schema's zal worden gekozen voor TEI en DITA. Deze twee schema's zijn gebruikt bij de eerste twee klanten van FontoXML en dus goede concrete voorbeelden. Uit deze activiteit komt de elementclassificatie voort.

D. XML-elementgroepen definiëren

Als alle eigenschappen van de elementen in kaart zijn gebracht kunnen er groepen worden gemaakt. Deze groepen vormen de basis voor de ontwerpen. Deze groepen worden vastgelegd in een document genaamd XML-elementgroepen.

E. Uitvoeren gebruikersonderzoek

Om een goed beeld te krijgen van hoe de applicatie daadwerkelijk gebruikt wordt, zal een aantal gebruikers worden geïnterviewd. Hierin wordt gevraagd naar de applicatie-gerelateerde taken en gebruiken bij hun huidige werk en de software die zij momenteel gebruiken. De interviews zullen digitaal worden vastgelegd om te zorgen dat er geen informatie verloren gaat.

F. Opstellen Use Cases

Om beter te kunnen begrijpen hoe de verschillende bewerkingen plaats zullen vinden, wordt er een aantal use cases opgesteld. Hierin wordt een combinatie van taken en bewerkingen beschreven zoals deze bij het daadwerkelijke gebruik voor zal komen. De Use Cases zullen als hoofdstuk worden verwerkt in het ontwerprapport.

G. Vergelijken en analyseren bestaande applicaties

Om het wiel niet opnieuw uit te vinden, zal een analyse worden gehouden op bestaande applicaties die dezelfde gebruikersdoelen bedienen. Dit kan een snelle manier zijn voor het bedenken van interfacepatronen, indien deze van voldoende kwaliteit worden geacht voor gebruik binnen FontoXML. Deze activiteit leidt tot de Benchmark.

H. Uitvoeren taakanalyse per elementgroep

Om een interfacepatroon te kunnen ontwerpen voor de verschillende elementgroepen, moeten de taken en stappen die bij een bewerking horen worden geanalyseerd. Op deze manier bij het ontwerpen rekening worden gehouden met de verschillende routes en volgorden tijdens het uitvoeren van een taak. Doordat elke groep is gebaseerd op een gelijke bewerking van haar elementen, is een analyse per groep voldoende. De uitgewerkte analyse vormt een hoofdstuk van het ontwerprapport.

I. Richtlijnen voor de interface opstellen

Door van tevoren een aantal richtlijnen op te stellen kan de richting worden bepaald voor het ontwerpen. Deze richtlijnen kunnen bijvoorbeeld te maken hebben met het wel of niet onderbreken van de gebruiker in zijn workflow bij verplichte stappen. Deze richtlijnen

moeten voortkomen uit overleg met het team, omdat ze aan moeten sluiten bij de filosofie van FontoXML. Ook deze richtlijnen zijn deel van het ontwerprapport.

J. Interfacepatronen bij richtlijnen onderzoeken

Aan alle richtlijnen kan een aantal bestaande interfacepatronen worden gekoppeld die mogelijk zouden kunnen worden ingezet in de applicatie. Deze vormen een basis bij het ontwerpen van varianten hierop of eventuele nieuwe patronen. Gevonden patronen en de bijbehorende argumentatie worden in het ontwerprapport gekoppeld aan specifieke richtlijnen.

K. Schetsen van interfacepatronen

Aan de hand van richtlijnen, eerder ontworpen interfacepatronen en onderzochte interfacepatronen kunnen er nieuwe ontwerpen worden gemaakt. Deze zullen worden geschetst en vervolgens digitaal uitgewerkt om zo duidelijk mogelijk te communiceren hoe de verschillende bewerkingen met het systeem kunnen worden uitgevoerd. Voor elke uitgewerkte elementgroep worden maximaal drie alternatieve patronen ontworpen die vervolgens getest kunnen worden. De patronen worden opgeslagen als schets op papier en indien nodig duidelijker uitgewerkt in Photoshop of Illustrator. Daarnaast worden de ontwerpen ingescand en toegevoegd als bijlage aan het ontwerprapport.

L. Prototypes bouwen

Om de ontworpen interacties zo goed mogelijk te kunnen testen zal er een high-fidelity prototype worden ontwikkeld in HTML, CSS en Javascript. Dit kan gebeuren op basis van de bestaande applicatie, zodat er een zo realistisch mogelijke testomgeving is. Een werkend prototype is zeer belangrijk omdat dit de meest representatieve vorm van testen mogelijk maakt. De prototypes worden bewaard in het versiebeheersysteem GIT als aparte branches. Om de prototypes toegankelijk te maken voor beoordelende docenten zal er ook een server worden opgezet waar deze bekeken kunnen worden.

M. Voorbereiden van usabilitytest

Voor een succesvolle test moeten er eerst onder andere doelen, methoden en scenario's worden bepaald. Deze punten en meer worden vastgelegd in het testplan.

N. Testen van interfacepatronen op gebruiksvriendelijkheid

Om te bepalen of mijn ontwerpen daadwerkelijk werken en gebruiksvriendelijk zijn moeten deze getest worden op echte gebruikers. De usabilitytest zal digitaal worden vastgelegd via audio en/of video, afhankelijk van het type test dat zal worden uitgevoerd.

O. Verwerken en analyseren testresultaten

Nadat de test is uitgevoerd worden de resultaten van de usabilitytest verwerkt en geanalyseerd. Deze analyse zal bepalen welke patronen het beste geschikt zijn voor gebruik binnen FontoXML. Hiermee wordt de basis voor het adviesrapport gevormd. Het uitvoeren van deze activiteit leidt tot een testrapport.

P. Opstellen advies voor opdrachtgever

Om naar het team van FontoXML te communiceren welke interfacepatronen het meest geschikt zijn voor iedere elementgroep, zal een adviesrapport worden gemaakt. Hierin staat, op basis van bevindingen in de usabilitytest, een advies over de meest geschikte interface voor FontoXML.

Q. Vastleggen, verklaren en evalueren van afstudeerproject

Deze 15 dagen zullen niet allemaal aan het einde van het project plaatsvinden, maar verspreid over de tijd. In het afstudeerdossier worden mijn keuzes, gedachten en activiteiten vastgelegd, verklaard en geëvalueerd zodat er vanuit school beoordeeld kan worden of de opdracht op niveau is uitgevoerd.

4. Tussenresultaten

Om tot de eindproducten te kunnen komen zal er ook een aantal tussenproducten worden opgeleverd. In tussenproducten worden gedane bevindingen of resultaten van een bepaalde activiteit vastgelegd. Een tussenproduct vormt altijd de basis voor een aantal beslissingen of volgende producten. Als dit niet zo is, is een tussenproduct waardeloos. Als tussenproducten zullen de volgende worden opgeleverd.

1. Plan van aanpak (*document*)

Het plan van aanpak komt voort uit activiteit *plannen van het project*. Het is enkel bedoeld om houvast te bieden aan het projectteam en zal dus niet worden opgeleverd naar de klant. Het bevat onder anderen de planning en een lijst van de op te leveren producten.

2. Elementclassificatie (*spreadsheet*)

Wanneer de *oriëntatie en inventarisatie XML-elementen* is voltooid, kan de elementclassificatie worden opgesteld. In deze classificatielijst van XML-elementen worden alle elementen uit het DITA-schema en het TEI-schema beoordeeld op een aantal eigenschappen die van invloed kunnen zijn op de bewerking ervan. Deze classificatie vormt de basis voor de activiteit *XML-elementgroepen definiëren*.

3. Benchmark (*document*)

In de benchmark wordt de uitkomst van het onderzoek naar concurrerende applicaties vastgelegd. Deze applicaties wordt op een aantal vastgestelde punten vergeleken, waarna suggesties voor ontwerpen kunnen worden gedaan. De benchmark kan worden gebruikt als referentie bij het *schetsen van interfacepatronen*.

4. Set XML-elementgroepen (*document*)

De activiteit *XML-elementgroepen definiëren* leidt tot een set van XML-elementgroepen. Deze groepen worden vastgelegd in een document. Indien de groepen correct op bewerking zijn gegroepeerd, kan voor elk van deze groepen een drietal interfacepatronen ontworpen worden.

5. Ontwerprapport (*document*)

Het ontwerprapport heeft als doel om de keuzes en argumenten van het ontwerpproces te communiceren. Op deze manier kan duidelijk worden gemaakt dat ontwerpkeuzes om een bepaalde reden zijn gemaakt of niet. Het rapport bevat onder meer de volgende onderdelen:

a. Use cases

Als het *opstellen use cases* is voltooid, is er in het ontwerprapport een overzicht te vinden van realistische voorbeelden van hoe de applicatie gebruikt zal worden.

b. Taakanalyse

Taakanalyses worden gebruikt om de doelen van je gebruikers beter te begrijpen. Dit gebeurt door de web naar een bepaald doel op te splitsen in kleine stukjes, zodat alle routes worden.

c. Richtlijnen voor interfaceontwerp

De richtlijnen voor de interface moeten worden aangehouden om de filosofie van het team door te kunnen voeren in de ontwerpen.

d. Interfaceschetsen

Door de interface en interacties te schetsen op papier kan, indien correct uitgevoerd, een hoop tijd bespaard worden. De schetsen vormen de basis voor de prototypes, zonder grafisch te zijn uitgewerkt. Indien blijkt dat deze schetsen niet genoeg detail bevatten, zullen er ook mock-ups worden uitgewerkt.

6. High-fidelity prototypes (*webpagina*)

De prototypes worden gebouwd om de ontwerpen zo goed mogelijk te kunnen testen in een werkende applicatie. De interface zal hier moeten functioneren zoals in de ontwerpen is beschreven. De prototypes zijn dus een afhankelijkheid voor het *testen van interfacepatronen op gebruiksvriendelijkheid*.

7. Testplan (*document*)

In het testplan wordt vastgelegd op welke taken er getest gaat worden. Hiervoor worden meerdere opdrachten opgesteld die aan de testpersonen worden voorgelegd.

8. Testverslag (*document + vastgelegde audio en/of video*)

In het testverslag staan de onbewerkte gegevens uit de test opgeslagen. Deze gegevens voorkomen dat er informatie uit de test verloren gaat, zodat alles geanalyseerd en verwerkt kan worden.

9. Testrapport (*document*)

In het testrapport staan de analyses en conclusies die gedaan zijn aan de hand van het de testresultaten. Deze gegevens zijn belangrijk, omdat zij de kern vormen voor het *opstellen van een advies aan de opdrachtgever*. De resultaten kunnen namelijk als bewijs worden gezien voor het wel- of niet gebruiksvriendelijk zijn van de ontwerpen.

5. Kwaliteit

De kwaliteit van zowel eindproducten als de tussenproducten wordt bewaakt door de bedrijfsmentor. Er is tussen de student en de bedrijfsmentor een wekelijkse meeting afgesproken aan het begin van de week. Tijdens deze meeting zullen de vorderingen van de afgelopen week en de planning voor de komende week worden besproken. Daarnaast is er de gelegenheid voor de bedrijfsmentor om feedback te geven op tussenproducten waar nodig. Omdat de bedrijfsmentor tegelijk ook de opdrachtgever vertegenwoordigt, is er dus de mogelijkheid om vanuit beide perspectieven feedback te geven. Dit scheelt tijd, maar tegelijk is het wel belangrijk om aan te geven vanuit welke rol bepaalde feedback gegeven wordt. Een product is pas definitief indien dit door zowel de bedrijfsmentor als de student is bevestigd.

Om de voortgang van het project te managen zal de methode van Roel Grit als projectmanagementmethode worden gebruikt. Deze methode wordt veel toegepast binnen de opleiding CMD en is al meerdere malen geschikt gebleken om ontwikkelprojecten van webapplicaties te managen.

Om de voortgang van de prototypes veilig te bewaren, zal er gebruik worden gemaakt van versiebeheersysteem GIT. Door dit te gebruiken is het altijd mogelijk om eerdere versies van het prototype te herstellen indien nodig. Verder wordt de kwaliteit van de ontwerpbeslissingen onderbouwd door deze te baseren op de theorie uit o.a. de volgende literatuur:

- Anderson, S. P. (2011). *Seductive Interaction Design*. Berkeley, California: New Riders
- Grit, R. (2008). *Projectmanagement*. Groningen: Noordhoff Uitgevers
- Hall, E. (2013). *Just enough research*. New York, New York.: A Book Apart
- Johnson, J. (2010). *Designing with the Mind in Mind*. Burlington, Massachusetts.: Elsevier
- Lidwell, W. (2010) *Universal Principles of Design, Revised and Updated*. Beverly, Massachusetts.: Rockport Publishers
- Portigal, S. (2013). *Interviewing Users*. Brooklyn, New York.: Rosenfeld Media
- Saffer, D.(2009) *Designing for interaction*. San Fransisco, California.: New Riders
- Tidwell, J. (2010) *Designing Interfaces*. Sebastpol, California.: O'Reilly Media

6. Projectorganisatie

In dit hoofdstuk wordt beschreven wie de deelnemers en belanghebbenden bij dit project zijn, wat hun rol is en hoe zij samenwerken. Verder worden de communicatiekanalen en -momenten vastgelegd om beperkingen in de communicatie te voorkomen.

6.1. Organisatie binnen de projectgroep

Het project zal onafhankelijk van de normale vorderingen en projecten binnen FontoXML worden uitgevoerd, omdat de projectmethode van Roel Grit niet goed aansluit bij de SCRUM-methode die het projectteam van FontoXML hanteert. Hierdoor ontstaan er eigenlijk twee groepen: mensen die deelnemen aan het daadwerkelijke afstudeerproject en het team van FontoXML zelf.

6.1.1. Leden afstudeerproject

Hieronder vallen de mensen die zich actief zullen bezighouden met het afstudeerproject. Als uitvoerende partij is hier dus alleen de student genoemd. Verder heeft de bedrijfsmentor een controlerende en studerende rol in het project. Als laatste is de product direct betrokken. Aan hem zal het adviesrapport gericht zijn en hij zal ook bepalen of de adviezen worden meegenomen in de ontwikkeling van de applicatie.

Functie	Naam	E-mail adres	Telefoon
Projectleider, interaction designer	Youri Bosselaar	youri.bosselaar@liones.nl	0616654760
User Experience Lead, bedrijfsmentor	Vincent Smedinga	vincent.smedinga@liones.nl	0641822919
Product owner	Jan Benedictus	jan.benedictus@liones.nl	0614482303

6.1.2. Contactpersonen projectteam FontoXML

Ondanks het feit dat het afstudeerproject onafhankelijk van de ontwikkeling van FontoXML loopt, zullen er momenten binnen het project zijn waar samenwerking met leden het team nodig is. De

personen met wie samenwerking binnen dit project het meest waarschijnlijk is, zullen in deze paragraaf worden beschreven.

Functie	Naam	E-mail adres	Telefoon
Product Architect	Bert Willems	bert.willems@liones.nl	0630719616
Javascript Lead	Thomas Brekelmans	thomas.brekelmans@liones.nl	0681505632
Developer	Wybe Minnebo	wybe.minnebo@liones.nl	0634031112
Grafisch ontwerper	Mike Zuidgeest	mike.zuidgeest@liones.nl	0620357016

6.2. Stakeholders

Er zijn, zoals bij ieder project, meerdere partijen die nauw bij dit project betrokken zijn. Voor elk van deze partijen zal in deze paragraaf worden besproken wat de belangen zijn bij het project en welke invloed er uitgeoefend kan worden op het resultaat.

6.2.1. De student

Voor de uitvoerende partij - de student - is het van belang dat het project op een zo hoog mogelijk niveau wordt uitgevoerd. De mate van geslaagdheid is voor hem af te meten aan de kwaliteit van het eindproduct. Omdat de student de producten van het project maakt, is hij dan ook verantwoordelijk voor deze kwaliteit.

6.2.2. De Product Owner

De productowner is er ook bij gebaat dat de kwaliteit van het product zo hoog mogelijk is. Immers, de Product Owner is degene die beslist of de ontwerpen en prototypes zullen worden toegepast in de echte applicatie. Wanneer de groepen correct zijn en interactiepatronen daar goed bij aansluiten, vermindert die implementatietijd voor nieuwe klanten. Dit verhoogt de waarde van het product, wat voor de Product Owner natuurlijk het uiteindelijke doel is.

6.2.3. De Lead Designer

Voor de Lead Designer is het van belang dat gemaakte ontwerpen aansluiten bij de designfilosofie van FontoXML. Dit is dan ook de reden dat deze persoon ook is aangewezen als bedrijfsmentor voor het project. Op deze manier zijn er wekelijkse contactmomenten waarin bijvoorbeeld de designkeuzes besproken kunnen worden.

6.2.4. Het developmentteam

Het developmentteam staat neutraal tegenover de opdracht. Enerzijds is het verbeteren van de interface en het implementatietraject voor hen belangrijk, omdat ook zij zich verbonden voelen met het eindproduct. Anderzijds is het zo dat nieuwe interface-elementen een hoop nieuwe ontwikkeltijd kunnen betekenen. Als de code niet goed samenvalt met de bestaande code, moet er dus wellicht een hoop werk opnieuw worden gedaan. Dit zal het developmentteam niet graag zien, omdat het een lang en dus duur proces zou kunnen zijn.

6.2.5. De klant

De klant is vooral gebaat bij de dalende kosten van het product wanneer het project slaagt. Een kortere implementatietijd zal ook een lagere prijs betekenen, wat het product voor klanten aantrekkelijker kan maken.

6.2.6. Vakexperts - de gebruikers

Voor de gebruikers is het belangrijk dat zij hun taken zo gemakkelijk en prettig mogelijk kunnen uitvoeren. Zij hebben dus geen belang bij het wel- of niet bestaan van de elementgroepen en de bijbehorende patronen. In feite zou het voor hen zelfs beter zijn als ieder element individuele aandacht krijgt in het ontwerpproces, zodat het zo goed mogelijk ontworpen is.

6.3. Documentbeheer en communicatie

Communicatie met alle betrokkenen gebeurt face to face, via Google Hangouts of via de email. Omdat iedereen in hetzelfde kantoor zit zal dit vooral face to face zijn. Communicatie met eindgebruikers zal ook het liefst face to face gebeuren (interviews en gebruikerstesten). Mocht dit niet mogelijk zijn, dan zal een video-call software als skype of Google Hangouts worden gebruikt.

Alle documenten en tussenproducten zullen worden bewaard en bewerkt op Google Drive. Op deze manier hebben alle betrokkenen op ieder moment de mogelijkheid om de producten in te zien. Daarnaast werkt het versiebeheer van Google Drive erg goed voor het herstellen en teruglezen van eerdere versies.

Zoals al eerder vermeld wordt de bedrijfsmentor wekelijks op de hoogte gehouden van vorderingen en planning. Daarnaast zal het team van FontoXML op de hoogte worden gehouden van mijn vorderingen door af en toe deel te nemen aan de Daily Standup. Hier vertelt ieder projectlid elke ochtend kort wat hij heeft gedaan en wat hij gaat doen.

7. Planning

In bijlage A is een uitgebreide strokenplanning te vinden. Hierin staat beschreven in welke volgorde de activiteiten binnen het project worden uitgevoerd, hoeveel tijd hiervoor gereserveerd is en wanneer ze worden uitgevoerd. Om een snel overzicht van het project te creëren is er in dit hoofdstuk ook nog een lijst van alle deadlines binnen het project te vinden.

Product	Deadline
Plan van aanpak	2 september
Elementclassificatie	10 september
Set XML-documentgroepen	12 september
Analyse gebruikersonderzoek & Use cases	18 september
Benchmark	22 september
Ontwerprapport - Taakanalyse	29 september
Ontwerprapport - Ontwerprichtlijnen	30 september
Interfaceschetsen	30 oktober
High-fidelity prototypes	20 november
Testplan	25 november
Testrapport	1 december
Adviesrapport	8 december

8. Risico's

Om goed voorbereid te zijn op belemmerende situaties, is het belangrijk om vooraf de risico's bij het project in te schatten. Daarbij kan ook vast worden nagedacht over een adequate reactie wanneer een risicovolle situatie zich voordoet.

8.1 Interne risico's

- **Problemen met de planning**

Het zou kunnen zijn dat de student tijdens het plannen de benodigde tijd van een of meerdere activiteiten verkeerd heeft ingeschat. Als dit leidt tot vertraging, zal eventuele extra benodigde tijd moeten worden ingehaald in het weekend. Als de vertragingen zeer groot worden, moet met de begeleidend examinerator worden gesproken over het alternatief uitvoeren van bepaalde activiteiten in het project, zodat deze minder tijd kosten.

- **Langdurige ziekte of persoonlijke problemen**

Indien er sprake is van deze situatie zal er ook contact moeten worden opgenomen met de begeleidend examinerator. Afhankelijk van de situatie moet er een besluit worden genomen over het wel of niet verleggen van de afstudeerperiode.

8.2 Externe risico's

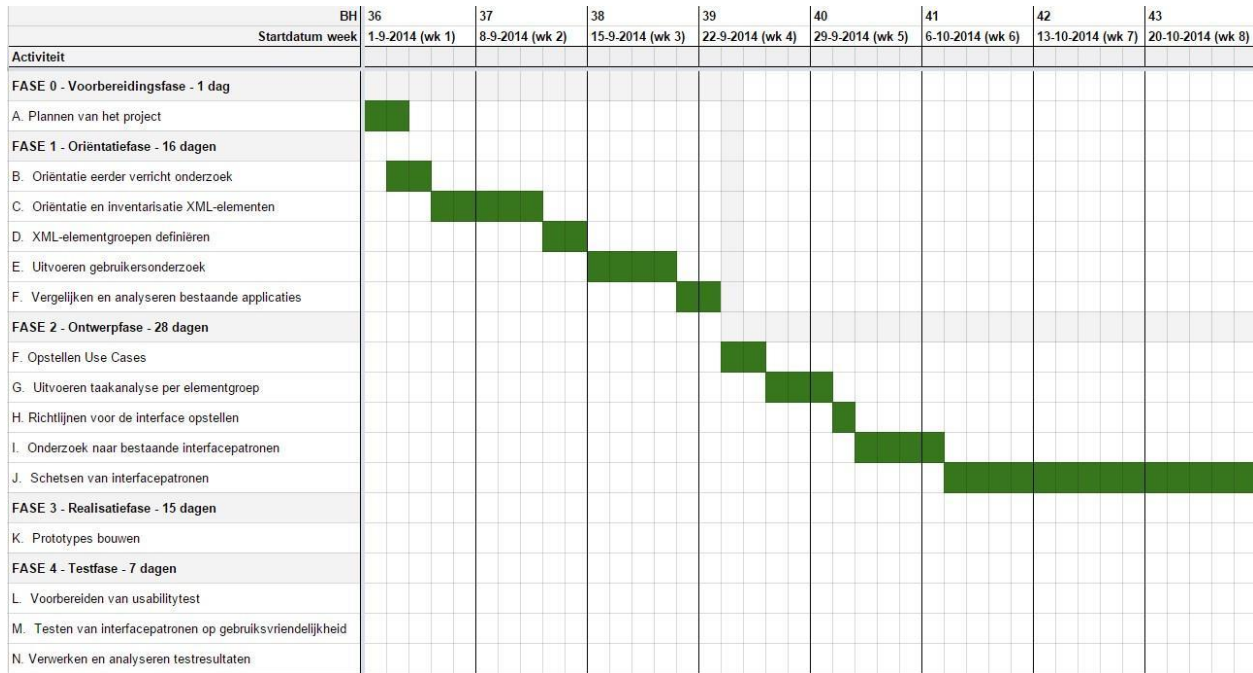
- **Technische problemen met opslagmedia**

Theoretisch gezien is het mogelijk dat er een storing optreedt bij Google Drive, of op de harde schijf van mijn pc. Echter, door de bestanden zowel offline als online te bewaren, wordt dit risico nihil.

- **Slechte bereikbaarheid/bereidheid eindgebruikers**

Voor het onderzoeken van de gebruikersbehoeften en testen van de prototypes zijn eindgebruikers nodig. De kans is echter altijd aanwezig dat deze gebruikers niet bereid zijn om mee te werken, of slecht vindbaar zijn. Het vinden van de eindgebruikers kan worden bevorderd door betrokken partijen te benaderen. Slechte bereidheid kan bijvoorbeeld worden opgelost door een kleine beloning tegenover besteedde tijd op te stellen.

Bijlage I: Strokenplanning



Bijlage C

Onderzoeksrapport XML bewerking

C

Onderzoeksrapport

Designuitdagingen in de bewerking van XML-elementen

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Inleverdatum: 9 januari 2015



FONTOXML

Inhoudsopgave

1. Inleiding	37
2. Achtergrondinformatie	38
2.1 Wat is XML?	38
2.2 De applicatie FontoXML	38
2.3 Projectdoelen voor FontoXML	39
2.4 Gebruiksvriendelijk bewerken van XML-elementen	40
2.5 Aanpak	40
3. Bewerkingen op XML-elementen	42
4. Invloed van schemavaliditeit op bewerkingen	47
4.1 Regels op ouder-kind relatie	47
4.1.1 Toegestane ouder-kind relaties	47
4.1.2 Vereiste ouder-kind relaties	48
4.1.3 Aantal toegestane instanties van een kind	48
4.1.4 Vaste volgorde van kinderen	49
4.1.5 Elkaar uitsluitende kinderen	49
4.2 Regels op element-attribuut relatie	50
4.2.1 Toegestane element-attribuut relaties	50
4.2.2 Vereiste element-attribuut relaties	50
4.3 Regels op attribuut	50
4.3.1 Verplichte attribuutwaarde	50
4.3.2 Toegestane attribuutwaarde	51
5. Invloed van “best practices” op bewerkingen	52
5.1 Elementcombinaties aanmoedigen	52
5.2 Element-attribuutcombinaties aanmoedigen	52
5.3 Bewaken van waardevolle content	53
5.3.1 Attribuut(waarde) is van onvoorziene invloed	53
6. Conclusie	54
Literatuurlijst	55
Verklarende woordenlijst	56

1. Inleiding

Voor mijn afstudeerproject kreeg ik de opdracht om te onderzoeken hoe je kunt bepalen welke interface er nodig is bij het bewerken van een element. Bij aanvang van het project beschik ik echter niet over de kennis die ik hiervoor nodig heb. Om die kennis te bemachtigen, heb ik een onderzoek uitgevoerd. Dit onderzoeksrapport is opgesteld om de resultaten van dit onderzoek te verklaren en mijn bevindingen vast te leggen.

In hoofdstuk 2 beschrijf ik kort de context van dit onderzoek. De eerste stap van het onderzoek zelf was het in kaart brengen van de verschillende mogelijke bewerkingen op een XML-element. De resultaten hiervan zijn beschreven in hoofdstuk 3. Vervolgens heb ik onderzocht welke factoren van verdere invloed zijn op deze bewerkingen. De eerste groep factoren hierbij waren XML-schemaregels. Deze regels worden in hoofdstuk 4 verklaard en gekoppeld aan XML-bewerkingen. De tweede groep factoren, best practices bij het bewerken van XML documenten, wordt op dezelfde manier beschreven en gekoppeld aan element in hoofdstuk 5. De resultaten van deze drie hoofdstukken worden in hoofdstuk 6 - de conclusies - overzichtelijk weergegeven in een tabel.

2. Achtergrondinformatie

Om een basis te leggen voor het onderzoeken en ontwerpen van de interfacepatronen, is het allereerst belangrijk om de context en details van de applicatie FontoXML te kennen. Om dit te achterhalen, sprak ik met een aantal teamleden van FontoXML over het bewerken van XML-documenten en de filosofie achter de applicatie. Dit was geen officieel gestructureerd interview: de bevindingen in dit document zijn in eigen woorden samengevat.

2.1 Wat is XML?

XML is een coderingstaal die bestaat om betekenis en structuur van tekstdocumenten te kunnen opslaan, zodat zowel mensen als computers die betekenis beter kunnen begrijpen en interpreteren. Het belangrijkste om te weten over XML-documenten is dat deze bestaan uit een groep XML-elementen. Deze elementen voegen ieder een eigen stukje betekenis aan de tekst toe. Een element kan bijvoorbeeld staan voor een paragraaf, een term uit het register, een verklarende woordenlijst of een auteur van het document. Elementen zijn er dus in allerlei soorten. Het bewerken van XML-elementen is dan ook zeer nauw gerelateerd aan het bewerken van tekstdocumenten: XML-elementen zijn alleen maar een manier om de opbouw van die teksten vast te leggen. Het doel van organisaties is dan ook nooit om te beschikken over XML-documenten, maar om te beschikken over een manier waarop de structuur en semantiek van hun documenten voor zowel computer als mens inzichtelijk kunnen worden gemaakt.

XML-documenten worden opgeslagen volgens een boomstructuur. Binnen XML-documenten heeft ieder element altijd 1 “ouder”-element en 0 of meer “siblings”: elementen die dezelfde ouder hebben. Daarnaast kan een element ook 0 of meer “kind”-elementen hebben: dit zijn elementen die zich in het element bevinden. Een element kan niet meerdere ouders hebben.

Een ander belangrijk aspect aan XML is dat de inhoud van XML-documenten vaak moet voldoen aan een bepaald schema. In dit schema zijn regels vastgelegd die gaan over de inhoud van de documenten(Thompson, 2004). Vaak gaan deze regels over het verplicht-of verboden zijn van bepaalde elementen in bepaalde situaties. In principe is een XML-document van weinig waarde als het niet aan het achterliggende schema voldoet. Het is dus van groot belang dat auteurs geen schema-INVALIDE documenten creëren.

2.2 De applicatie FontoXML

FontoXML is een web-applicatie voor het bewerken van XML-documenten, gericht op vakexperts met weinig tot geen kennis van XML. In het proces van contentcreatie wordt veel van de content namelijk geschreven door vakexperts, zoals leraren, monteurs of muziekkenners. Zij schrijven deze

content dan in Microsoft Word. Omdat veel organisaties momenteel overstappen naar XML als opslagformaat voor hun content, is er vervolgens een conversie nodig van Word-documenten naar XML-documenten. Deze conversie kost veel tijd, moeite en geld. Om te faciliteren in deze nieuwe behoefte wil FontoXML een manier bieden waarop vakexperts hun content direct in XML kunnen schrijven, terwijl het gevoel van schrijven in Word behouden blijft. In de applicatie is er dan ook voor gekozen om het XML-document wat de gebruiker bewerkt niet te tonen als code, maar als het tekstdocument dat die code representeert. De bedoeling hiervan is om het document beter leesbaar en begrijpelijk te maken, zeker voor mensen met weinig of geen kennis van XML-code.

Wat betreft het omgaan met de regels uit XML-schema's heeft het team van FontoXML een aantal belangrijke beslissingen genomen. Omdat documenten die niet aan het bijbehorende XML-schema voldoen, nagenoeg waardeloos zijn, bestaat bijvoorbeeld de volgende richtlijn:

“Het moet niet mogelijk zijn om met de applicatie schema-invalid documenten te creëren.”

Een ander idee dat bij FontoXML hoog in het vaandel staat, is het aanmoedigen van de gebruiker om goede content en structuur te creëren. Zo is het bijvoorbeeld gewenst om best practices voor contentcreatie toe te passen waar mogelijk. De applicatie moet daarbij helpen omdat de gebruikers vaak weinig kennis hebben van XML. Hieruit komt de richtlijn:

“De gebruiker moet, waar mogelijk, worden aangemoedigd in het toepassen van best practices”

Vanwege de complexe aard van XML-documenten, heeft het team ervoor gekozen om de XML-code zoveel mogelijk weg te houden bij de gebruiker. In plaats van deze code te tonen, wordt het document weergegeven als een soort Word-document: een representatie van de semantiek die in de code is vastgelegd. Er wordt dus op een visuele manier geprobeerd om de gebruiker duidelijk te maken hoe het XML-document dat hij bewerkt, is opgebouwd en wat daar allemaal voor inhoud in zit. Hierbij is het dus belangrijk dat alle (voor de gebruiker relevante) XML-code wordt vertaald naar een visuele vorm.

“Achterliggende XML-code moet zoveel mogelijk bij de gebruiker worden weggehouden. Liever moet de conceptuele betekenis van deze code in het document worden gevisualiseerd.”

2.3 Projectdoelen voor FontoXML

Om te ontdekken wat doelen van mijn afstudeerproject voor FontoXML zijn, sprak ik met het team van FontoXML. Er zijn volgens hen twee belangrijke doelen die ten grondslag liggen aan mijn afstudeerproject:

1. **FontoXML wil een gebruiksvriendelijke XML-editor aanbieden.**

Het product FontoXML moet zich onderscheiden van andere XML-editors door gebruiksvriendelijk te zijn voor mensen zonder kennis van XML. Een belangrijk onderdeel hiervan is dat FontoXML dus een interface moet hebben waarmee XML-elementen op een gebruiksvriendelijke manier kunnen worden bewerkt. Omdat het product ontwikkeld wordt aan de hand van de projectmethode SCRUM, is er nog geen grootschalig project gewijd aan de juiste interfacepatronen voor alle mogelijke bewerkingen. Aangezien het wel belangrijk is dat er voor alle bewerkingen een zo gebruiksvriendelijk mogelijk patroon komt, is mijn afstudeerproject in het leven geroepen.

2. FontoXML wil voor nieuwe elementen sneller de interface kunnen samenstellen.

Omdat klanten vaak eigen XML-schema's hebben, heeft het team van FontoXML vaak te maken met nieuwe elementen. Voor hen is het hierbij belangrijk om in zo weinig mogelijk tijd een goed onderbouwde beslissing kan worden genomen over de interfacepatronen voor het bewerken van dit element. Op deze manier kunnen (nieuwe) elementen snel worden opgenomen en bewerkt in de applicatie, zonder dat daarvoor constant denkwerk moet worden gedaan of herhaald. Om hierin te kunnen faciliteren, wordt dit onderzoek uitgevoerd. Concreet hoopt men na het onderzoek te weten:

- in welke bewerkingssituaties bepaalde interfacepatronen kunnen worden gebruikt
- wat de beste positie is voor de trigger voor het aanmaken van ieder element

De aanname van de opdrachtgever is dat er op basis van dit onderzoek een beslissingsmodel kan worden ontwikkeld dat helpt bij het kiezen van het juiste interfacepatroon voor een bepaalde bewerking. De conclusie zou echter ook kunnen zijn dat dit model niet bestaat, of niet voldoende dekkend is om meerwaarde te bieden voor het team van FontoXML.

2.4 Gebruiksvriendelijk bewerken van XML-elementen

Het doel van deze opdracht is om voor elk XML-element een patroon te hebben waarmee deze op een gebruiksvriendelijke manier kan worden bewerkt. Om dit te kunnen ontwerpen, is er een aantal vraagstukken dat eerst zal moeten worden opgelost:

1. Wat is “een element bewerken”: welke bewerkingen bestaan er?
2. Welke factoren kunnen het bewerken van een XML-element beïnvloeden?

Deze vragen zullen in dit document worden opgelost.

2.5 Aanpak

Allereerst moet er worden bepaald welke “soorten” bewerkingen er eigenlijk zijn binnen het bewerken van XML-documenten. Op deze manier wordt de omvang van de opdracht duidelijk en

kan een selectie worden gemaakt in de bewerkingen die wel en niet belangrijk zijn binnen de scope van dit project. Als die selectie er is, kan er onderzoek worden gedaan naar de relatie tussen elementeigenschappen en interfacepatronen..

Om een zo compleet mogelijk resultaat te krijgen, is er bij dit onderzoek voor gekozen om het probleem vanaf twee kanten te benaderen. Bij de eerste benadering wordt geredeneerd over factoren die waarschijnlijk invloed hebben op de bewerking van elementen. Deze factoren worden bepaald door na te denken over de context van het product: richtlijnen, restricties en conventies in tekstbewerking.

De tweede benadering komt vanaf de andere kant. Hierbij zal een groot aantal XML-elementen in een lijst worden gezet. Voor elk element wordt dan, in de documentatie van het bijhorende schema, kort onderzocht wat de betekenis en het beoogde gebruik van dat element is. Na dit gedaan te hebben kan door de ontwerper voor al deze elementen worden ingeschat welke interfacepatronen er nodig zouden zijn bij de bewerking ervan. Deze aanpak is niet per se zwaar theoretisch onderbouwd, maar meer gebaseerd op het gevoel en de ervaring van de ontwerper. Een meer theoretische benadering zou zijn om voor ieder element te kijken hoe dit wordt bewerkt in concurrerende applicaties. Echter, omdat het hier gaat om ongeveer 200 elementen, zou dit een veel te tijdrovende en kostbare aanpak zijn. Vanwege de ervaring van de ontwerper met dit onderwerp (ongeveer een jaar), zal dus voor de eerst voorgestelde aanpak worden gekozen.

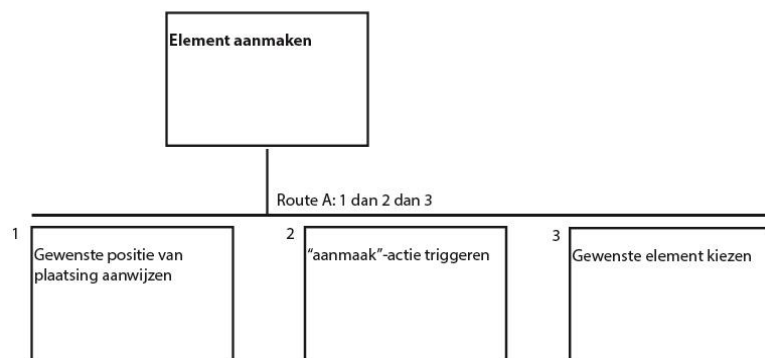
Als er conclusies zijn die vanaf twee kanten getrokken kunnen worden, zal worden aangenomen dat deze correct en dekkend zijn. Als conclusies van een enkele kant worden getrokken, zal in overleg met het team worden overlegd of deze correct zijn en dus bruikbaar zijn voor het project.

3. Bewerkingen op XML-elementen

Binnen het “bewerken van XML-elementen” is een aantal bewerkingen denkbaar. Uit het werken met testdocumenten in andere tekstverwerkers, is een lijst met bewerkingen ontstaan. Hieronder is die lijst weergegeven. Per bewerking is daarbij een omschrijving en een korte taakanalyse gemaakt die inzicht biedt in de stappen en mogelijke volgorden van iedere bewerking.

3.1 Element aanmaken - selectie omwikkelen

Het is, binnen het schrijven teksten, gebruikelijk om eerst de kale tekst te schrijven en vervolgens deze tekst te selecteren en een bepaalde betekenis toe te kennen. Denk bijvoorbeeld aan het “bold” maken van een stuk tekst in Word. Deze actie bestaat ook als je “niets” selecteert: je cursor staat op een positie zonder selectie. De stappen binnen deze taak zijn de volgende:

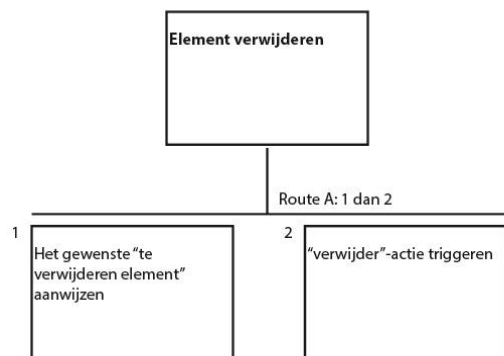


3.2 Element aanmaken - selectie vervangen

Een ander veelvoorkomend patroon is het vervangen van een selectie voor nieuwe inhoud. Dit zou bijvoorbeeld kunnen gebeuren wanneer je een stuk tekst selecteert en vervolgens een afbeelding invoegt. Deze bewerking bestaat uit dezelfde stappen als bewerking 1.

3.3 Element verwijderen - inhoud bewaren

De volgende logische actie in XML-tekstbewerking is het *verwijderen* van elementen. Ook hier bestaan twee varianten, waarvan bij de eerste variant enkel het element wordt verwijderd: de inhoud blijft bestaan. Indirect betekent dit dat de inhoud dus in een ander element zal moeten worden geplaatst: de voorgaande “grootouder” van het element is nu zijn ouder. Hierbij kunnen de volgende stappen worden benoemd:

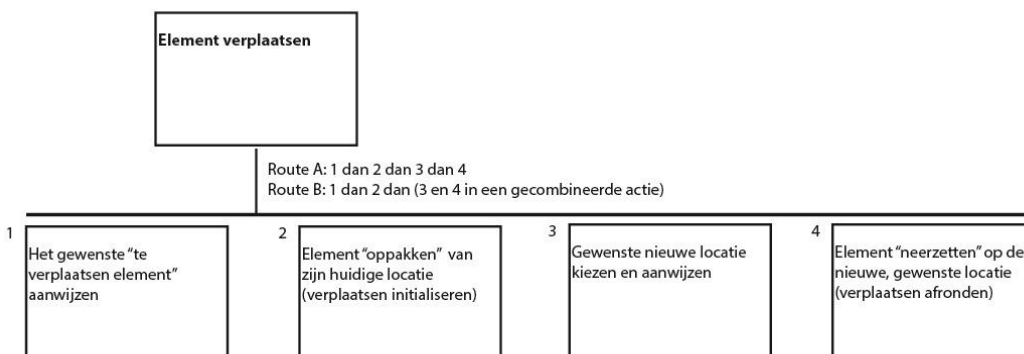


3.4 Element verwijderen - inclusief inhoud

Bij de tweede variant van verwijderen worden zowel het element als al zijn inhoud uit het document verwijderd. Deze bewerking bestaat uit dezelfde stappen als bewerking 3.

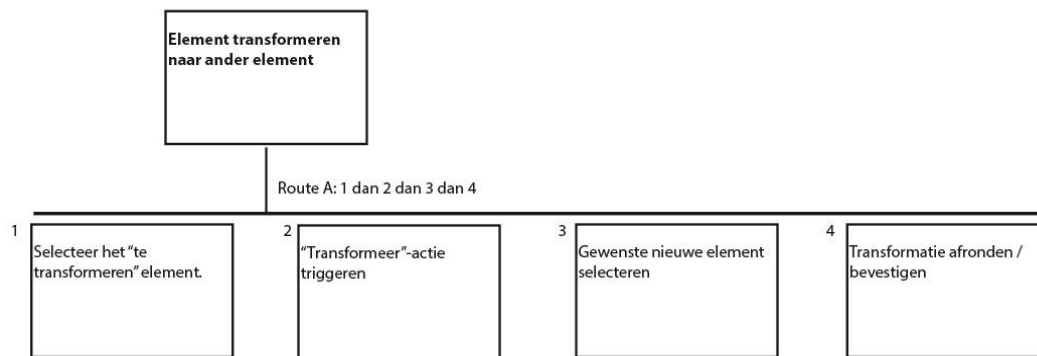
3.5 Element verplaatsen

Het kan natuurlijk voorkomen dat je bepaalde tekst (of elementen) binnen een document wilt verplaatsen. Om maar even in een materiële metafoor te spreken zal je het element daarvoor moeten selecteren, "oppakken" en weer ergens "neerzetten". Verder is er hierbij een tweetal factoren van invloed op het wel of niet slagen van deze bewerking. Het verplaatsen van een element bestaat uit de volgende stappen:



3.6 Element transformeren naar ander element

In sommige gevallen is het denkbaar dat je een element wil transformeren naar een ander element: bijvoorbeeld een paragraaf naar een sectie, of een lijst-item naar een paragraaf. Het transformeren van een element kent de volgende stappen:



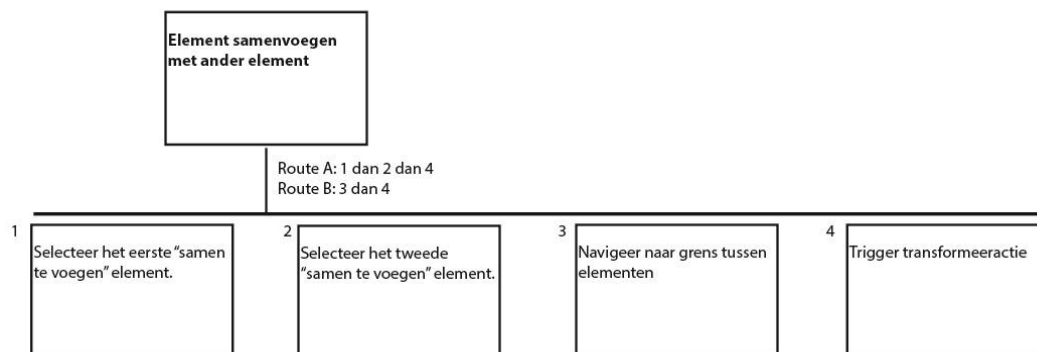
3.7 Element splitsen

Bij splitsen wordt een element, zoals het woord al suggereert, opgesplitst in twee gelijksoortige elementen. Deze actie komt zeer vaak voor bij bijvoorbeeld paragrafen en lijst-items. Het splitsen van een element bestaat uit de volgende stappen:



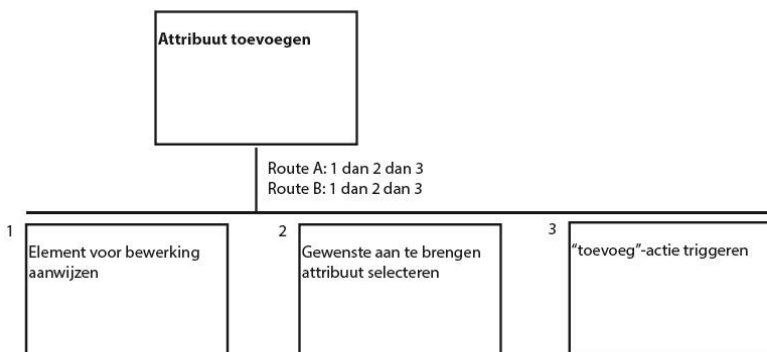
3.8 Element samenvoegen met ander element

Vanuit tekstbewerking is het gebruikelijk om elementen samen te voegen tot een enkel element. Bijvoorbeeld twee paragrafen die toch eigenlijk een enkel paragraaf moeten zijn. Het samenvoegen van een element met een aangrenzend element bestaat uit de volgende stappen:



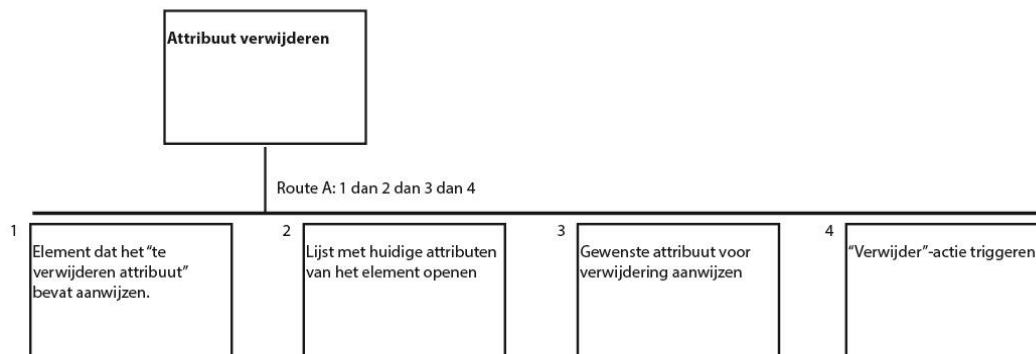
3.9 Attribuut toevoegen

Met attributen kun je bepaalde eigenschappen toekennen aan een element. Het is dus in het bewerken van XML erg logisch om een attribuut te kunnen toevoegen. Een attribuut toevoegen kent de volgende stappen.



3.10 Attribuut verwijderen

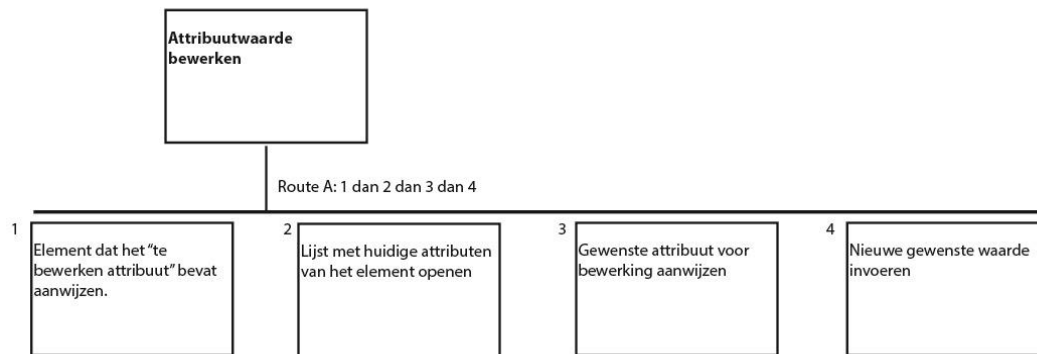
Net zoals dat bij elementen het geval is, wil je bepaalde attributen kunnen verwijderen van een element. De gewenste situatie kan veranderd zijn, of misschien is een attribuut gewoon foutief toegevoegd. Het verwijderen van een attribuut kent de volgende stappen:



3.11 Attribuutwaarde bewerken

Attributen kunnen, net als elementen, een waarde bevatten. Deze waarde is tekstinhoud en kan dus door de gebruiker worden bewerkt. Het bewerken van een attribuutwaarde kent de volgende

stappen:



3.12 Elementinhoud bewerken

De laatste mogelijke actie in het bewerken van XML-documenten is het bewerken van de inhoud van een element. Het lijkt er, op het moment dat dit geschreven wordt, op dat deze bewerking niet door bepaalde factoren wordt beïnvloedt. Inhoud kan vrij worden bewerkt naar verlangen van de gebruiker en in XML-schema's kunnen geen regels op de inhoud van een element worden uitgedrukt. Om die reden zal deze bewerking verder in het onderzoek en het ontwerpproces buiten beschouwing worden gelaten. Mochten nieuwe inzichten leren dat deze bewerking toch onder bepaalde invloed dient te veranderen, zal deze alsnog worden meegenomen in het onderzoek.

4. Invloed van schemavaliditeit op bewerkingen

Zoals eerder genoemd, worden XML-documenten bijna altijd gebruikt in combinatie met een schema. Omdat schema-INVALIDE documenten nagenoeg waardeloos zijn en ook slecht kunnen worden uitgelezen door applicaties, heeft FontoXML de volgende richtlijn opgesteld:

“Het moet niet mogelijk zijn om met de applicatie schema-INVALIDE documenten te creëren.”

Door deze richtlijn, zal er bij het ontwerpen van de bewerkingen rekening moeten worden gehouden met de (in het schema-vastgelegde) regels die worden beschreven in dit hoofdstuk. Per regel wordt beschreven op welke bewerkingen deze van invloed is. Ook wordt er voor iedere regel een aantal voorbeelden gegeven, gebaseerd op de DITA-documentatie (Eberlein, 2010) en/of de TEI-documentatie (Burnard, 2014).

4.1 Regels op ouder-kind relatie

Zoals gezegd kunnen elementen “kinderen” bevatten. Dit houdt in dat er binnen de grenzen van een bepaald element nog een ander element bestaat. In XML-schema's worden vaak regels gesteld op deze relatie. De verschillende mogelijkheden hiertoe worden in dit paragraaf besproken.

4.1.1 Toegestane ouder-kind relaties

Een van de meest voorkomende regels die in schema's wordt vastgelegd gaat over het toestaan van elementen in andere elementen. Voor ieder element is aangegeven in welke andere elementen hij mag bestaan. Ook is aangegeven welke andere elementen hij mag bevatten. Omdat het document altijd schemavalide moet zijn, moet dus bijvoorbeeld worden voorkomen dat een element ergens wordt geplaatst waar hij niet mag staan. Hiervoor moet het systeem uitlezen welke elementen er op de huidige cursorpositie mogen worden geplaatst.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Element verwijderen - inhoud bewaren
- Element verplaatsen
- Element transformeren naar ander element
- Element splitsen

Voorbeelden:

<note> mag geen <hazardstatement> bevatten - (DITA)

<persname> mag geen <p> bevatten - (TEI)

4.1.2 Vereiste ouder-kind relaties

In XML kan worden uitgedrukt dat een element bepaalde andere elementen moet bevatten. Hier spreken we dus van een vereist kind. Omdat er binnen de applicatie geen schema-ongeldige situaties mogen bestaan, moeten elementen dus altijd hun vereiste kinderen bevatten. Het kan ook zo zijn dat een schema minstens één van een serie vereiste kinderen moet bevatten. In dit geval zijn er twee oplossingen denkbaar: de gebruiker laten kiezen welke kinderen er worden toegevoegd, of automatisch de eerste uit die lijst toevoegen. Uiteraard ligt de voorkeur hier bij het laten kiezen van de gebruiker.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Element verwijderen (beide varianten)
- Element verplaatsen
- Element transformeren naar ander element
- Element splitsen
- Element samenvoegen met aangrenzend element

Voorbeelden:

<topic> moet een <title> bevatten, <hazardstatement> moet een <messagepanel> bevatten, <step> moet een <cmd> bevatten - (DITA)

4.1.3 Aantal toegestane instanties van een kind

Het kan zijn dat het schema zegt dat er maar één instantie van een element mag bestaan binnen een bepaalde ouder. Dit dient naar de gebruiker gecommuniceerd te worden, omdat het verwarrend gedrag kan opleveren. Zo mag de gebruiker een element de eerste keer wel toevoegen, maar de tweede keer niet..

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Element verplaatsen
- Element transformeren naar ander element
- Element splitsen

Voorbeelden:

<topic> mag slechts een <abstract> bevatten, <taskbody> mag slechts een <result> bevatten, <prolog> mag slechts een <source> bevatten, <hazardstatement> mag slechts een <typeofhazard> bevatten - (DITA)

4.1.4 Vaste volgorde van kinderen

Er zijn elementen die op een zeer specifieke positie binnen hun ouder moeten staan. Binnen het document kan het dus erg moeilijk zijn om hier met de cursor precies de juiste positie aan te klikken en vervolgens het element te plaatsen. Daarnaast moet de gebruiker een volgorde uit zijn hoofd kennen om de juiste locatie te kunnen vinden. Om de gebruiker te helpen, moet de applicatie vergevingsgezind zijn in dit soort situaties en helpen bij het kiezen van de juiste locatie voor een element.

Factor heeft invloed op:

- Element aanmaken
- Element verplaatsen
- Element transformeren

Voorbeelden:

<taskbody>, <messagepanel>, <step> - (DITA)

4.1.5 Elkaar uitsluitende kinderen

Sommige kinderen mogen niet naast elkaar bestaan in een bepaalde ouder. In dit geval mag er dus maar 1 van een bepaalde groep kinderen bestaan. Er moet hier dus een keuze worden gemaakt voor een van deze kinderen en als er een bestaat, mogen de anderen niet meer worden toegevoegd. Wel zou je bijvoorbeeld kunnen “switchen” tussen de opties. Er is echter een uitzondering. Deze situatie geldt alleen wanneer kinderen elkaar uitsluiten én deze slechts eenmaal is toegestaan. Als meerdere van deze keuzes zijn toegestaan in de ouder, kunnen zij simpelweg naast elkaar geplaatst worden.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Element verplaatsen
- Element transformeren naar ander element

Voorbeelden:

<topic> mag een <shortdesc> óf een <abstract> bevatten, <taskbody> mag een <steps> óf een <steps-unordered> bevatten - (DITA)

4.2 Regels op element-attribuut relatie

4.2.1 Toegestane element-attribuut relaties

Net zoals bij elementen, kunnen schema's ook beperkingen opleggen in de toegestane element-attribuut relaties. In de applicatie moet dus worden voorkomen dat er “verboden” element-attribuut combinaties worden gecreëerd.

Factor heeft invloed op:

- Attribuut toevoegen

Voorbeelden:

`<topic>` mag geen `[href]` bevatten, `<p>` mag geen `[width]` bevatten - (DITA)

4.2.2 Vereiste element-attribuut relaties

Een element kan een of meerdere vereiste attributen bevatten. Dat betekent dat een element niet mag bestaan zonder dit attribuut te bevatten. Deze eis kan vanuit twee kanten worden opgelegd: het schema of de klant. Wanneer de eis vanuit het schema wordt opgelegd, kan de applicatie dat automatisch uitlezen. Daarnaast kan het zo zijn dat een klant bepaalt dat een element binnen de klantspecifieke applicatie nog andere verplichte attributen heeft. Dit is wellicht niet in het schema vastgelegd en zal dan dus vooraf met het team samen besproken moeten worden. Vervolgens moeten deze relaties ergens worden gedefinieerd zodat de applicatie ze kan uitlezen.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Attribuut verwijderen

Voorbeelden:

`<classSpec>` moet een `[type]` bevatten, `<namespace>` moet een `[name]` bevatten - (TEI),
`<tm>` moet een `[tmtype]` bevatten - (DITA)

4.3 Regels op attribuut

4.3.1 Verplichte attribuutwaarde

Het kan zo zijn dat een attribuut een waarde moet bevatten. In dit geval is het document dus invalide wanneer deze waarde niet aanwezig is. Omdat er geen invalide situaties mogen bestaan,

moet deze situatie dan ook worden opgelost voordat een bewerking op de attribuutwaarde kan worden afgerond.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Attribuut toevoegen
- Attribuutwaarde bewerken

Voorbeelden:

Notitie: deze voorbeelden zijn niet afkomstig uit de bestaande schema's, maar mogelijke toepassingen van deze regel die mij realistisch leken.

Bij <date> moet [when] een waarde bevatten. (TEI)

4.3.2 Toegestane attribuutwaarde

Het kan zo zijn dat de waarde van een attribuut aan een bepaald formaat moet voldoen. Om gebruikers hierbij te helpen is het belangrijk om een geschikte interface aan te bieden voor het invoeren van inhoud in dit formaat. Daarnaast mag een schema-invalid waarde niet bestaan: als het formaat vastligt in het schema en de waarde hier niet aan voldoet, moet de gebruiker eerst een correcte waarde invoeren.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Attribuut toevoegen
- Attribuutwaarde bewerken

Voorbeelden:

[scale], [frame], [status], [type], [translate], [role], [keys], [type] -
(DITA)

5. Invloed van “best practices” op bewerkingen

Omdat men bij FontoXML weet dat hun gebruikers weinig tot geen kennis hebben van XML, willen zij hier en handje bij helpen. Bekende patronen in XML-bewerking (ook wel “best practices”) kunnen bijvoorbeeld automatisch door de applicatie worden toegepast of worden aangemoedigd. De richtlijn die FontoXML hierbij hanteert is als volgt:

“De gebruiker moet, waar mogelijk, worden aangemoedigd in het toepassen van best practices”

Met deze richtlijn is er een aantal factoren met invloed op het bewerken van elementen te benoemen. Deze factoren worden in dit hoofdstuk behandeld.

5.1 Elementcombinaties aanmoedigen

Er zijn in het bewerken van XML bepaalde veelgebruikte element-combinaties. Hierbij is het “logisch” dat bepaalde elementen samen worden gebruikt. Om A: de gebruiker moeite te besparen en B: de gebruiker aan te moedigen bepaalde combinaties te gebruiken, kan FontoXML hier een vorm van sjablonen aanbieden. Als je weet dat elementen zeer vaak samen worden gebruikt, scheelt het de gebruiker meer tijd als hij af en toe een element moet verwijderen dan als hij deze allemaal los zou moeten aanbrengen.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Element verwijderen (beide varianten)
- Element transformeren naar ander element

Voorbeelden (mogelijk)

<topic>, <task>, <hazardstatement> en <bibl> - (DITA)

5.2 Element-attribootcombinaties aanmoedigen

Er kunnen situaties bestaan waarin een attribuutwaarde “aanbevolen” is. Dit betekent dat invoer niet verplicht is, maar wel moet worden aangemoedigd. In dit geval moet invoer de werkzaamheden van gebruikers dus niet tegenhouden, maar mag er wel om gevraagd worden. Echter, in tegenstelling tot bij een verplichte waarde mag hier wel “nee” gezegd worden.

Factor heeft invloed op:

- Element aanmaken (beide varianten)
- Attributwaarde bewerken

Voorbeelden (mogelijk)

<foreign> + [lang], <ref> + [href], <date> + [when], - (TEI)

5.3 Bewaken van waardevolle content

5.3.1 Attribut(waarde) is van onvoorziene invloed

Een attribuut is van onvoorziene invloed wanneer het bestaan en/of de waarde ervan invloed hebben op andere elementen / delen van de content. Binnen XML kan deze eigenschap niet worden aangegeven. Dit is dus een eigenschap die door het team / de klant zal moeten worden toegekend aan bepaalde attributen. Dit type attribuut kan in twee situaties bestaan:

1. Altijd van invloed

In dit geval is het attribuut, wanneer toegevoegd aan welk element dan ook, altijd van onvoorziene invloed

2. Van invloed in bepaalde combinatie

In dit geval is een attribuut alleen van invloed in combinatie met specifieke elementen.

Verder kan de invloed van dit type attributen regels bevatten. Een ID bijvoorbeeld, is niet van onvoorziene invloed wanneer er geen enkele referentie naar dat ID wordt gemaakt. Echter, wanneer er vanuit een of meerdere plekken in een document naar dat ID wordt verwezen, is er *wel* onvoorziene invloed ontstaan. Hier dient dus rekening mee gehouden te worden bij het verwijderen van dit element.

Factor heeft invloed op:

- Element verwijderen (beide varianten)
- Element transformeren
- Attribut verwijderen
- Attributwaarde bewerken

Voorbeelden (mogelijk)

<xref>, <conref> - (DITA)

<ref> - (TEI)

6. Conclusie

Om de bruikbaarheid van dit rapport binnen het project te verhogen, worden in dit hoofdstuk de conclusies van het onderzoek uitgewerkt. Deze conclusies vormen de basis voor ontwerpproblemen die kunnen worden meegenomen in de benchmark en later in de ontwerpen.

Na te hebben geanalyseerd welke bewerking er zijn binnen het bewerken van XML-elementen^(hfdst 3) en vervolgens welke factoren daar invloed op hebben^(hfdst 3, hfdst 4), kon er een overzichtstabel opgesteld^(tabel 7.1). In deze tabel is af te lezen welke factoren invloed hebben op welke bewerking.

Invloed op bewerkingen											
	Aanmaken - Wrap	Aanmaken - Replace	Verwijderen - Unwrap	Verwijderen - Delete	Verplaatsen	Transformeren	Splitsen	Samenvoegen	Attribuut toevoegen	Attribuut verwijderen	Attribuutwaarde bewerken
Vaste volgorde van kinderen											
Elkaar uitsluitende kinderen											
Toegestane element-attribuut relaties											
Vereiste element-attribuut relaties											
Toegestane attribuutwaarde											
Verplichte attribuutwaarde											
Best practices											
Aangemoedigde elementcombinaties											
Aangemoedigde element-attribuutcombinaties											
Attributen van onvoorziene invloed											

tabel 7.1: Overzicht van bewerkingen en factoren die daarbij van invloed zijn.

Elk blauwe vakje staat voor een situatie die ontworpen moet worden binnen de applicatie. Alles bij elkaar zijn dat dus 33 designuitdagingen. Het benchmarken, ontwerpen en testen van 33 designuitdagingen is te groot voor de scope van dit project. Samen met het team is besloten om prioriteit te geven aan het aanmaken van elementen. Vervolgens kan eventueel het verwijderen van elementen ontworpen worden. Zo staan de verschillende bewerkingen van links naar rechts gesorteerd op prioriteit. Hierbij is in ieder geval de bewerking “aanmaken” een must. “Verwijderen” is gewenst, maar niet cruciaal. De verdere bewerkingen zijn mooi meegenomen als hier tijd voor blijkt te zijn.

Literatuurlijst

Thompson, H. (2004, 28 oktober). *XML Schema Part 1: Structures Second Edition*. Opgehaald van <http://www.w3.org/TR/XMLschema-1/>

Eberlein, K. (2010, 1 december). *Darwin Information Typing Architecture (DITA) Version 1.2*. Opgehaald van <http://docs.oasis-open.org/dita/v1.2/os/spec/DITA1.2-spec.html>

Burnard, L., & Bauman, S. (2014, 16 september). *TEI Guidelines*. Opgehaald van <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/>

Verklarende woordenlijst

Attribuut

Attributen zijn eigenschappen van elementen die kunnen worden vastgelegd in XML. Een attribuut druk je uit door deze te plaatsen binnen de opening van een element, na de elementnaam. Attributen kunnen ook een waarde bevatten. Hier een voorbeeld waarbij “keyword1” de waarde van het “id”-attribuut is: `<keyword id=”keyword1”>usability</keyword>`.

Elebute

Vaste combinatie van een element en een attribuut. Elebutes ontstaan omdat een bepaalde combinatie tussen een element en een attribuut vaak wordt gebruikt. Deze combinatie zou evengoed een “apart” element kunnen zijn en wordt daarom ook zo behandeld.

(XML-)Element

Ieder onderdeel van een tekst dat als dusdanig moet worden kunnen worden onderscheiden, wordt een apart element. Elementen worden in XML vastgelegd door een woord te plaatsen in angle brackets, zoals in het volgende voorbeeld: `<keyword>usability</keyword>` Zoals hier ook te zien is hebben de meeste elementen een begin en een einde. Een element kan zo groot zijn als een document (`<topic>`) of zo klein als een steekwoord (`<keyword>`).

Kind-element

Kortweg: kind

Logischerwijs is er dus ook sprake van een kind-element: een element dat direct in een ander element staat. In het voorbeeld hieronder zou “`<quote>`” het kind van de

`<paragraph>` zijn. `<paragraph>`

`<quote>Ik ben een kind, dus heb ik een ouder.</quote>`

`</paragraph>`

Ouder-element

Kortweg: ouder

Een ouder-element is het element dat direct om een bepaald element heen staat. Hieronder volgt hier een voorbeeld waarbij de `<paragraph>` de ouder is van `<quote>`.

`<paragraph>`

`<quote>Ik ben een kind, dus heb ik een ouder.</quote>`

`</paragraph>`

XML-Schema

kortweg: schema

In een schema worden regels vastgelegd waaraan een XML-document moet voldoen om “valide” te zijn. Deze regels kunnen op verschillende eigenschappen en relaties van elementen gebaseerd zijn.

User Interface

kortweg: UI

De gebruikersinterface omvat alle aspecten van een product die faciliteren in de interactie tussen de gebruiker en dat product. Bij digitale producten bestaat dit dus vooral uit verschillende elementen op het scherm, maar ook het toetsenbord en de muis zouden onder deze interface kunnen worden genoemd.

Bijlage D

Gebruikersonderzoek

D

Gebruikersonderzoek

De gebruikers van FontoXML en hun behoeften

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Inleverdatum: 9 januari 2015



FONTOXML

Inhoudsopgave

1. Klanten van FontoXML	61
1.1. International Atomic Energy Agency (IAEA)	61
1.2. ThiemeMeulenhoff	61
2. Voorbereiding onderzoek	62
2.1. Doel van het onderzoek	62
2.2. Onderzoeksmethode	62
2.3. Interviewschema	63
3. Resultaten ThiemeMeulenhoff	64
3.1. Primaire gebruikersgroep: vakexperts	64
3.1.1. <i>Huidig systeem</i>	64
3.1.2. <i>Taken en workflow</i>	65
3.1.3. <i>User needs</i>	65
3.2. Secundaire gebruikersgroep: invoerders	66
3.2.1. <i>Huidig systeem</i>	66
3.2.2. <i>Taken en workflow</i>	66
3.2.3. <i>User needs</i>	67
3.3. Tertiaire gebruikersgroep: redacteurs	67

1. Klanten van FontoXML

De doelgroep van FontoXML is erg breed: per klant kan het soort gebruikers sterk verschillen. Omdat het binnen de scope van dit project niet mogelijk is om voor alle klanten apart een doelgroepanalyse te houden, heb ik deze beperkt tot twee klanten. Voor een van deze klanten (IAEA) was al een doelgroepanalyse beschikbaar, voor de ander (ThiemeMeulenhoff) heb ik zelf een onderzoek gedaan. Deze twee klanten heb ik gekozen omdat de projecten voor deze klanten lopende waren tijdens mijn afstudeeronderzoek. Onder het mom van deze projecten kon ik dus gemakkelijker informatie vergaren over deze klanten en hun gebruikers.

1.1. International Atomic Energy Agency (IAEA)

Voor een eerder project is er door collega Wybe Minnebo onderzoek gedaan naar de gebruikers binnen het IAEA. Om een zo compleet mogelijk beeld te kunnen krijgen van de gebruikers van FontoXML, zal dit onderzoek als bron worden gebruikt bij dit project. Een uitgebreide beschrijving van dit bedrijf en de gebruikers daar kan worden gevonden in bijlage A: Doelgroepanalyse IAEA.

1.2. ThiemeMeulenhoff

Het tweede bedrijf waarbinnen de gebruikers zijn geanalyseerd is ThiemeMeulenhoff. Dit onderzoek is wel door mijzelf uitgevoerd. ThiemeMeulenhoff is een grote Nederlandse uitgeverij die zich bezighoudt met de publicatie van leermiddelen. Het bedrijf is gevestigd in Amersfoort en heeft ongeveer 300 medewerkers. Het bedrijf is momenteel bezig met het ontwerpen en ontwikkelen van een totaal nieuwe “contentoplossing”. De gehele manier waarop hun content wordt gemaakt, verbeterd, gedistribueerd en gepubliceerd gaat op de schop. De focus komt hierbij te liggen op modulariteit, hergebruik en semantiek van de content. FontoXML gaat deel uitmaken van die contentoplossing als “creatieplatform”, waarop de XML-content wordt gemaakt. Het gebruikersonderzoek dat bij ThiemeMeulenhoff is uitgevoerd voor dit project, is in dit document uitgewerkt.

2. Voorbereiding onderzoek

Alvorens het onderzoek uit te voeren dient er een onderzoeksmethode gekozen te worden. Hierbij is het belangrijk om vooraf te bepalen wat het doel van dit onderzoek is. Hierbij dient te worden bedacht welk soort gegevens er verzameld moet worden. Met die wetenschap kan een gepaste onderzoeksmethode worden gekozen.

2.1. Doel van het onderzoek

In User Centered Design is het wenselijk om de gebruikers zoveel mogelijk te betrekken in het ontwerp- en ontwikkelproces van een applicatie of website. Zoals de naam al suggereert staat de gebruiker dus centraal in het hele proces. Concreet betekent dit dat er tijdens het proces constant rekening moet worden gehouden met de gebruiker. Wat zijn de behoeften die een gebruiker hier heeft? Kan de gebruiker deze taak moeiteloos uitvoeren? De antwoorden op dit soort vragen zijn van directe invloed op ontwerpbeslissingen die genomen moeten worden. Deze antwoorden kunnen gevonden worden door het uitvoeren van gebruikersonderzoek. In dit project is er een aantal gegevens van belang bij het nemen van ontwerpbeslissingen:

- Wat is de huidige kennis van de gebruikers over gestructureerd schrijven?
- In welke context werken de gebruikers met FontoXML?
- Welke behoeften hebben gebruikers bij het gestructureerd schrijven?
- Hoe voeren de gebruikers het werk dat zij met FontoXML gaan doen momenteel uit?

2.2. Onderzoeksmethode

Voor het achterhalen van gebruikersbehoeften in een vroeg stadium van het project, is het nodig om een geschikte methode te kiezen. Omdat er nog geen prototypes zijn om mee te testen, is een contextual inquiry hier goed in te zetten. Deze methode verschaft informatie over zowel de achtergrond en persoonlijke gegevens van gebruikers, als over de taken die zij uitvoeren met betrekking tot het te ontwerpen product.

Contextual inquiry

Om zonder ruis te kunnen leren hoe een gebruiker zijn werk doet, kan een contextual inquiry worden uitgevoerd (Ross, 2012¹). Deze begint vaak met een kort interview van de (toekomstige) gebruiker.

¹ <http://www.uxmatters.com/mt/archives/2012/06/why-are-contextual-inquiries-so-difficult.php>

Vervolgens wordt de gebruiker geobserveerd tijdens het doen van zijn werk, in de daarvoor meest natuurlijke omgeving. Daarnaast kunnen er door de observanten vragen worden gesteld om te ontdekken wat de behoeften van gebruikers zijn bij dit werk (*UsabilityBok*, 2010²). Hierbij kan worden gelet op pijnpunten in het proces, maar ook op de motivaties en frustraties van de gebruiker. Om te voorkomen dat waardevolle gegevens verloren gaan, is het aan te raden om bij een contextual inquiry ofwel het proces te filmen, ofwel met twee observanten te werken.

2.3. Interviewschema

Hieronder zijn de vragen te vinden die, voorafgaand aan het observeren aan iedere gebruiker kunnen worden gesteld. De vorm van het interview mag semigestructureerd zijn: improviseer in het stellen van doorvragen wanneer gebruikers interessante onderwerpen aansnijden. Onder “interessante onderwerpen” vallen bijvoorbeeld onderwerpen die betrekking hebben tot gevoel, gebruiken en behoeften van de gebruikers.

1. Wat voor werk doet u?
2. Kunt u een korte beschrijving van uw dagelijkse taken geven?
3. Doet u [*authoring* / *invoeren*] nu ook al?
 - Ja → Ga door naar 4
 - Nee → Doet u wel eens ander werk waarbij u teksten moet schrijven?
4. Wat vindt u van dat werk? Ziet u de waarde ervan in?
5. Op welke manier doet u dat werk momenteel? En met welke programma's?
6. Bent u bekend met XML?
7. Wat zijn prettige punten in huidige manier van werken (breed tot specifiek)?
8. Wat zijn frustraties in huidige manier van werken (breed tot specifiek)?
9. Zijn er dingen die naar uw idee ontbreken in de programma's die u gebruikt?

² <http://www.usabilitybok.org/contextual-inquiry>

3. Resultaten ThiemeMeulenhoff

In dit hoofdstuk worden de bevindingen uit de onderzoeken bij ThiemeMeulenhoff beschreven. Deze zijn bondig samengevat in een aantal categorieën die van invloed zijn op de manier waarop FontoXML ontworpen zal moeten worden voor de gebruikers.



Figuur 1: Een toekomstige FontoXML-gebruiker wordt geobserveerd tijdens het uitvoeren van haar werk.

3.1. Primaire gebruikersgroep: vakexperts

De eerste en meest belangrijke groep gebruikers bij ThiemeMeulenhoff zijn de vakexperts. In het geval van Thieme gaat het hier over docenten, die zo nu en dan bijdragen aan het schrijven van lesstof. Vaak is dit een langdurig proces waarbij docenten gedurende een periode van maanden, soms zelfs jaren, betrokken zijn. Vakexperts krijgen instructies van de contentontwikkelaars bij Thieme. Deze instructies gaan vooral over het formaat van de inhoud die zij moeten schrijven, maar ook over de leerdoelen waarbij deze moeten aansluiten. Deze vakexperts hebben, zoals de naam al suggereert, erg veel vakinhoudelijke kennis. Zij zijn dan ook cruciaal in het proces van contentcreatie, omdat zij de content daadwerkelijk creëren. Van achterliggende documentstructuren hebben zij echter een stuk minder kennis. Zij kennen de aard en opbouw van hun eigen stof, maar zijn niet gewend om deze expliciet vast te leggen in een formaat als XML.

3.1.1. Huidig systeem

De vakexperts werken nagenoeg allemaal in Microsoft Word. Alle ondervraagden hebben hiermee minimaal 5 jaar ervaring. Hierin schrijven zij hun content, ontvangen zij commentaren van

redacteurs of contentontwikkelaars en hierin brengen zij verbeteringen aan. Vakexperts komen eigenlijk niet in aanraking met andere software zoals Evolution, laat staan met de XML-code zelf. Hier ligt hun interesse niet en vaak wordt dit ook niet gezien als onderdeel van hun werk.

3.1.2. Taken en workflow

Het is de taak van een auteur om content te creëren. Soms gebeurt dit in een enigszins vrije, ongestructureerde vorm en soms krijgen zij hiervoor sjablonen toegestuurd van een contentontwikkelaar. In dat geval is er de opdracht om de content binnen het sjabloon te schrijven, zodat er al een bepaalde vorm van structuur ontstaat. Natuurlijk is dit nog steeds geen XML en zal het Word-sjabloon dus moeten worden omgezet naar XML in een andere applicatie, door technisch redacteurs. Wat opvalt, is dat sommige auteurs wel al aangeven van welk “type” een paragraaf of een hoofdstuk is. Zij plaatsen dan bijvoorbeeld een afkorting voor de titel, waarmee dit type wordt aangeduid. Hier zijn zij dus al bewust bezig met het aanbrengen van semantiek: de stap naar het markeren van een paragraaf lijkt hier een kleine te zijn.

3.1.3. User needs

Uit de observaties zijn er meerdere punten naar voren gekomen die van invloed zullen zijn op het gebruik van FontoXML. Om deze punten kort en overzichtelijk te houden, zijn deze geformuleerd als “user needs”. Deze dienen te worden meegenomen in de systeemeisen. Daarnaast moeten deze user needs ook in het achterhoofd worden gehouden tijdens het ontwerpen van de interfacepatronen.

- Vakexperts hebben de behoefte om te kunnen zien welke functie een knop in de interface vervult. Hierbij vragen zij specifiek om tooltips met daarin een korte uitleg.
- Vakexperts zijn volledig gewend aan het werken in Word. Hierdoor ontstaat de behoefte om interacties, waar mogelijk, zo te laten werken als in Word. Zo wordt de leercurve voor het gebruik van FontoXML verkleind.
- Vakexperts hebben behoefte aan begeleiding in het structureren van hun content. Momenteel worden hier vaak Word-sjablonen voor gebruikt.

3.2. Secundaire gebruikersgroep: invoerders

De tweede groep FontoXML-gebruikers bij Thieme zijn de “invoerders”. Zij houden zich bezig met het overzetten van stukken lesstof uit lesboeken/PDF naar een HTML-systeem. Hierbij is het van belang dat de versies in het HTML-systeem zoveel mogelijk overeenkomen met de gedrukte versies van de lesstof. De groep invoerders bestaat uit zes studenten van tussen de 19 en de 26 jaar. Zij studeren allemaal op hbo of WO-niveau en zien hun werk bij ThiemeMeulenhoff als een (tijdelijke) bijbaan. Vakinhoudelijke kennis van de leerstof is geen vereiste onder de invoerders. Het is voor hun werk wel enigszins nodig om de semantiek en de opbouw van de lesstof te begrijpen, maar dat leren ze zodra ze met het werk beginnen. Zo moeten zij bijvoorbeeld bepaalde stukken tekst invoeren als zijnde een referentie, introtekst, kadertekst of highlighted text. De invoerders hebben nagenoeg geen interesse in- en kennis van xml. Veel van de studenten geven aan de aard van het werk erg makkelijk te vinden. Hierdoor “kan het soms erg saai worden”, aldus een van de invoerders.

“Soms is het wel lekker dat je even moet nadenken”

Voor het schrijven en bewerking van teksten hebben alle invoerders veel ervaring met het gebruiken van Microsoft Word. Zij maken hiermee vaak verslagen en documenten voor hun opleiding en geven aan eigenlijk nooit andere tekstverwerkers te gebruiken.

3.2.1. Huidig systeem

Momenteel doen de invoerders hun werk in een HTML-bewerkingsprogramma genaamd “XSLit”. Dit systeem is pas sinds kort in gebruik en verving een eerder pakket. Ondanks het feit dat er nog steeds een aantal onnodige interacties in XSLit lijkt te zitten, zijn de invoerders blij met het systeem. Het is naar hun zeggen een “grote verbetering” ten opzichte van het oudere systeem. Alle invoerdersdraaien XSLit in de Google Chrome browser.

3.2.2. Taken en workflow

Het doel van het werk dat de invoerders doen is om de gedrukte lesstof over te zetten naar het digitale lessysteem. Hierbij moet de content zoveel mogelijk lijken op die in het boek. De invoerders beginnen met het creëren van een nieuwe “pagina”. Hierbij schrijven zij eerst een titel voor de pagina, en vervolgens verschijnt het scherm waarin de inhoud bewerkt kan worden. Dit doen zij door de content te kopiëren uit een PDF-versie van het boek. Vervolgens plakken zij deze in de HTML-verwerker en geven zij aan wat voor type content het hier om gaat. Binnen zo’n tekst moeten dan nog bepaalde structuren en markeringen worden aangebracht. Hierbij gaat het vooral om dik- of schreefgedrukte tekst, tabellen en lijsten. Naar de reden dat iets dik- of schreefgedrukt is, wordt

door de invoerders niet gekeken. Daarnaast moet er wat metadata worden ingevoerd: alternatieve titels en het type pagina.

Verder houdt het werk ook in dat er naar de opmaak moet worden gekeken in het digitale systeem. Hiervoor kunnen bijvoorbeeld afbeeldingen worden gepositioneerd, geschaald en verplaatst. Daarnaast wordt door middel van witruimte bepaald in welke kolom een bepaald stuk tekst komt te staan. Hier zou het niet wenselijk zijn om deze witruimte in de gehele XML vast te leggen: deze heeft immers geen betekenis bij publicatie op andere platformen.

3.2.3. User needs

Uit de observaties zijn er meerdere punten naar voren gekomen die van invloed zullen zijn op het gebruik van FontoXML. Om deze punten kort en overzichtelijk te houden, zijn deze geformuleerd als “user needs”. Deze dienen te worden meegenomen in de systeemeisen. Daarnaast moeten deze user needs ook in het achterhoofd worden gehouden tijdens het ontwerpen van de interfacepatronen.

- Invoerders hebben de behoefte om (een preview van) de publicatievorm te kunnen zien. Dit is nodig om te controleren of de semantiek correct is toegepast en om te kijken of de visuele opmaak correct wordt getoond in het digitale systeem.
- Invoerders hebben behoefte aan begrijpelijke naamgeving in metadata en elementen
- Invoerders moeten op een gemakkelijke, snelle manier verschillende elementen kunnen verplaatsen.
- Invoerders hebben de behoefte om te kunnen controleren of de structuur van hun document klopt. Dit zouden zij bijvoorbeeld willen doen door te kijken naar de koppen binnen het document.
- Invoerders hebben behoefte om te kunnen zien of het document in zijn huidige staat is opgeslagen.

3.3 Tertiaire gebruikersgroep: redacteurs

De derde groep gebruikers binnen ThiemeMeulenhoff zijn de redacteurs. Hun functie is om het werk dat de auteurs opleveren na te kijken en te corrigeren. Dit kan zowel op xml-technisch gebied zijn, als op inhoudelijk gebied. Hun taak ligt dus vooral in het leveren van feedback op content. Echter, omdat het niet binnen hun takenpakket ligt om elementen te bewerken met FontoXML, zal deze gebruikersgroep verder buiten beschouwing worden gelaten..

Bijlage E

Benchmark

E

Benchmark

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Inleverdatum: 9 januari 2015



Inhoudsopgave

1. Aanpak	71
1.1. Designproblemen	71
1.2. Vergeleken applicaties	71
1.3. Beoordelingsmethode	72
2. Designproblemen	74
2.1. Omgaan met toegestane ouder-kind relaties	74
2.2. Omgaan met verplichte ouder-kind relaties	76
2.3. Omgaan met beperkt toegestane elementen	77
2.4. Omgaan met vaste volgorde van kinderen	79
2.5. Omgaan met elkaar uitsluitende kinderen	82
2.6. Omgaan met vereiste element-attribuut relaties	84
2.7. Omgaan met verplichte attribuutwaarden	85
2.8. Omgaan met toegestane attribuutwaarden	88
2.9. Omgaan met aangemoedigde elementcombinaties	90
2.10. Omgaan met aangemoedigde element-attribuutcombinaties	92
3. Conclusies	93

1. Aanpak

Deze benchmark is bedoeld om oplossingen te ontdekken, vergelijken en beoordelen voor de designproblemen die in mijn afstudeeropdracht naar voren komen. Op deze manier kan worden ingeschat welke oplossingen eventueel geschikt zijn om toegepast te worden binnen FontoXML. Ook kan worden vastgesteld hoe het *niet* moet.

1.1. Designproblemen

De designproblemen die zullen worden vergeleken zijn alle designproblemen die te maken hebben met het aanmaken van elementen. Het gaat hierbij om de volgende tien problemen:

1. Omgaan met toegestane ouder-kind relaties
2. Omgaan met verplichte ouder-kind relaties
3. Omgaan met beperkt toegestane elementen
4. Omgaan met vaste volgorde van kinderen
5. Omgaan met elkaar uitsluitende kinderen
6. Omgaan met vereiste element-attribuu relaties
7. Omgaan met verplichte attribuutwaarden
8. Omgaan met toegestane attribuutwaarden
9. Omgaan met aangemoedigde elementcombinaties
10. Omgaan met aangemoedigde element-attribuucombinaties

1.2. Vergeleken applicaties

Voor elk van deze problemen wordt in verschillende applicaties bekeken hoe zij deze oplossen. De applicaties die hiervoor zijn geselecteerd staan hieronder genoemd. Per applicatie wordt ook kort beschreven waarom deze een goede kandidaat is voor in de benchmark.

1. *Microsoft Word*

Uit het gebruikersonderzoek is gebleken dat nagenoeg alle gebruikers bekend zijn om te werken met Microsoft Word. Dit maakt het een geschikte kandidaat voor de benchmark. De kans is hierdoor namelijk groter dat gebruikers al bekend zijn met de oplossing die er geboden wordt, wat de eventuele leercurve van FontoXML zal verkleinen. Word is een betaalde software die ik bezit, waardoor ik deze vrijuit kan bekijken

2. *Google Docs*

Google Docs is een soort lichtgewichtversie van Microsoft Word, maar dan online. Veel van de oplossingen komen overeen, maar een aantal dingen wordt anders aangepakt. Als “reus”

in de wereld van tekstverwerkers, neem ik ook deze applicatie mee in de benchmark. Docs is gratis online te gebruiken, via <http://docs.google.com>.

3. *Oxygen*

Oxygen is de grootste, meest uitgebreide en meest gebruikte XML-editor op de markt. Het ondersteunt, net als FontoXML, meerdere schema's en zal dus met dezelfde problemen om moeten gaan. Van Oxygen is een gratis online demo beschikbaar via <http://www.oxygenxml.com/demo/AuthorDemoApplet/author-component-dita.html>

4. *EasyDita*

De laatste applicatie die ik meeneem in mijn benchmark is EasyDita. Deze online XML-editor maakt deel uit van een uitgebreider pakket en is niet apart te verkrijgen. De waardepropositie van EasyDita is echter behoorlijk gelijk aan die van FontoXML: zij beloven een gebruiksvriendelijke text-editor voor mensen zonder weinig technische kennis. Om die reden is EasyDita ook opgenomen in de benchmark. Om EasyDita te bekijken moet een trial-versie worden aangevraagd via <http://easydita.com/free-trial-form/>.

1.3. Beoordelingsmethode

De oplossingen in deze applicaties zullen worden beoordeeld aan de hand van een vorm van *Expert Review*. De hiervoor gekozen vorm heet een Heuristic Evaluation, en wordt gedaan aan de hand van een aantal relevante vuistregels op het gebied van usability. Hiervoor zijn de volgende vijf heuristics geselecteerd:

1. *"User control and freedom"*

Bij het bewerken van gestructureerde teksten zijn er enorm veel combinaties van elementen mogelijk. Het is dan ook belangrijk dat gebruikers zo veel mogelijk vrijheid hebben in het indelen van hun document.

2. *"Error prevention"*

XML is ingewikkeld. Er kunnen veel fouten worden gemaakt in de bewerking ervan en ieder schema heeft een enorme hoeveelheid verplichtingen en restricties. Waar mogelijk moet worden voorkomen dat gebruikers hier fouten in maken, omdat dit alleen maar frustratie oplevert.

3. *"Help users recognize, diagnose, and recover from errors"*

Om dezelfde reden als bij 2 kan het bijna niet worden voorkomen dat gebruikers zo nu en dan een fout maken. Het is belangrijk dat zij op dat moment begrijpen wat er gebeurd is en wat er eventueel van hen wordt verwacht om dat weer goed te maken.

4. *"Help and documentation"*

Het schrijven van goede, waardevolle XML vereist kennis. Het is in het geval van FontoXML niet realistisch om te verwachten dat de gebruiker al deze kennis uit zijn hoofd weet. Er zal

dus geregeld behoefte zijn aan uitleg of instructie: de applicatie dient hier de juiste informatie op het juiste moment aan te bieden.

5. *"Flexibility and efficiency of use"*

Bij FontoXML wil men de gebruiker zoveel mogelijk in de flow van het schrijven houden.

Het is dus van belang dat er zo weinig mogelijke omslactigheden en onderbrekingen in het bewerken van een document zitten..

2. Designproblemen

Gebaseerd op het eerdere onderzoek naar het bewerken van XML-elementen ben ik gekomen tot een groep designproblemen die allemaal te maken hebben met het aanmaken van elementen. Voor ieder van deze designproblemen zal ik in de vier geselecteerde applicaties kijken hoe deze worden opgelost. Dit kan inspiratie vormen voor oplossingen die in FontoXML gebruikt zouden kunnen worden.

2.1. Omgaan met toegestane ouder-kind relaties

Word

Er zijn nooit functies voor het invoegen van bepaalde typen content uitgeschakeld in Word. Alles mag dus in alles. Een heading 1 mag in een heading 2 bestaan, en noem maar op. Word biedt dus geen oplossing voor dit probleem

Docs

In Google Docs heb ik één gelijksoortige situatie kunnen ontdekken. Wanneer je in een inhoudsopgave staat met je cursor, mag je daar geen inhoudsopgave meer invoegen. De knop wordt inactief, maar er wordt verder geen feedback gegeven.

Heuristic	Score	Commentaar
User control	-/+	De gebruiker kan elementen niet invoegen waar hij maar wil: ook niet als tussenstap. Dit is echter ook niet gewenst.
Prevent errors	++	Door het uitschakelen van de knoppen worden vergissingen of invalide situaties voorkomen.
Help users with errors	<i>n.v.t.</i>	<i>foutsituaties worden voorkomen</i>
Help & documentation	--	Er is nergens te ontdekken waarom ik geen inhoudsopgave meer mag invullen.
Flexible & efficiënt	<i>n.v.t.</i>	

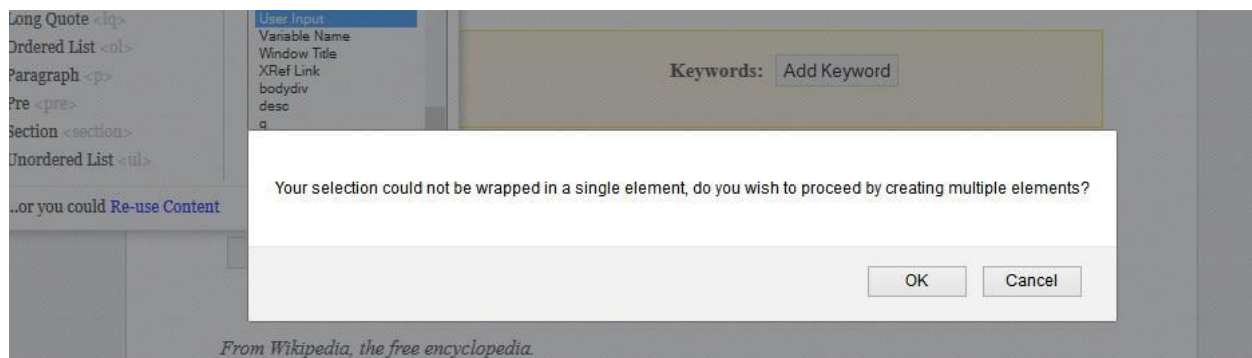
Oxygen

Oxygen schakelt knoppen uit van elementen die niet zijn toegestaan op de huidige cursorpositie. De knoppen worden lichtgrijs en je kunt er vervolgens niet meer op klikken. Er is niet voor alle elementen een knop in de interface: Oxygen werkt met een contextmenu dat verschijnt wanneer je op “Enter” drukt. Ook binnen dit menu zijn de knoppen voor niet toegestane elementen uitgeschakeld. Deze is echter alsnog wel te gebruiken, maar dan wordt een alternatieve locatie van plaatsing aangeboden.

Heuristic	Score	Commentaar
User control	-/+	De gebruiker kan elementen niet invoegen waar hij maar wil: ook niet als tussenstap. Dit is echter ook niet gewenst.
Prevent errors	++	Door het uitschakelen van de knoppen worden vergissingen of invalide situaties voorkomen. Als hij het echt wil, kan de gebruiker alsnog de functie aanroepen en wordt er een alternatieve actie voorgesteld. Hier wordt al vooraf een tooltip getoond met daarin een uitleg en een waarschuwing.
Help users with errors	<i>n.v.t.</i>	
Help & documentation	+	In de tooltip van de inactieve knop wordt aangegeven dat het element niet is toegestaan op deze locatie. Deze tooltip zou je echter ook nog bij :hover op de toolbarknop kunnen tonen.
Flexible & efficiënt	<i>n.v.t.</i>	

EasyDita

EasyDita maakt voor het invoegen van elementen gebruik van een “insert anything” menu, een aantal losse toolbarknoppen en een dropdownmenu met “more”. Op geen enkele van deze plekken worden knoppen op enig moment uitgeschakeld. DITA laat de gebruiker hier volledig vrij om fouten te maken: je kunt altijd op een knop drukken en als het niet werkt verschijnt er een modal. Hierin wordt een alternatieve actie aangeboden die je kunt uitvoeren of annuleren.



Afbeelding A: Easydita vertelt dat de gewenste actie niet mogelijk is, en biedt een alternatieve actie aan.

De aangeboden actie wordt echter lang niet altijd uitgevoerd, waardoor bij mij verwarring ontstond. Als je een actie niet kunt uitvoeren, waarom biedt je hem dan aan?

Heuristic	Score	Commentaar
User control	-	Easydita doet veel aannames over gewenste acties, waardoor de gebruiker niet altijd zelf kan bepalen wat hij wil doen.
Prevent errors	--	Easydita laat de gebruiker fouten maken en vangt die achteraf pas af. Als je achteraf weet dat iets fout is, kan je het ook vooraf weten. Dit zou dus eerder afgevangen en voorkomen moeten worden.
Help users with errors	-/+	Bij de foutsituaties die ontstaan bij Easydita wordt geprobeerd een alternatieve actie uit te voeren. Ook wordt de situatie deels uitgelegd. Dit kan echter nog duidelijker en uitgebreider.
Help & documentation	-	Er wordt niet vooraf duidelijk gemaakt of een element wel of niet is toegestaan. De gebruiker moet dit dus zelf weten ofwel ontdekken door een fout te maken.
Flexible & efficiënt	<i>n.v.t</i>	

2.2. Omgaan met verplichte ouder-kind relaties

Word

Word ondersteunt geen XML. Ook heb ik geen soortgelijke situatie kunnen ontdekken.

Docs

Docs ondersteunt geen XML. Ook heb ik geen soortgelijke situatie kunnen ontdekken.

Oxygen

In Oxygen wordt ervoor gekozen om verplichte kind-elementen automatisch toe te voegen met de ouder. Op deze manier kan worden gegarandeerd dat de XML valide is. Daarnaast wordt in Oxygen voor *alle elementen* een placeholder getoond wanneer deze leeg zijn. Op deze manier is het voor de gebruiker duidelijk zichtbaar dat er een leeg element in het document staat. De placeholder is dus

niet afhankelijk van het feit dat een element wel of niet verplicht is in zijn ouder, deze staat er gewoon altijd.

Heuristic	Score	Commentaar
User control	-/+	Er wordt niet aan de gebruiker om toestemming gevraagd bij het invoegen van verschillende elementen. Dit is echter ook niet per se nodig omdat het de efficiëntie niet bepaald ten goede zou komen.
Prevent errors	++	De schema- invalide situatie wordt voorkomen doordat Oxygen automatisch de vereiste kinderen toevoegt.
Help users with errors	<i>n.v.t.</i>	<i>Foutsituaties worden voorkomen</i>
Help & documentation	+	Met behulp van een placeholder geeft Oxygen de nieuwe elementen aan. Verder wordt er geen uitleg gegeven over de situatie.
Flexible & efficient	++	Door automatisch invoegen wordt dit probleem snel afgehandeld. De aanname die hierbij gemaakt wordt dat de gebruiker deze elementen ook nodig heeft is logisch en goed te verantwoorden: de fout ligt anders bij het schema, niet bij de applicatie.

Easydita

Easydita voegt, net als Oxygen, verplichte kinderen automatisch toe. Ook voegen zij net als Oxygen een placeholder toe wanneer er een leeg element wordt geplaatst (zoals bij command in step). Ook bij Hazardstatement worden er automatisch een Typeofhazard en een howtoavoid ingevoegd en wordt er een placeholder getoond.

Voor deze oplossing is geen aparte scoretabel gemaakt omdat de oplossing gelijk is aan die van Oxygen.

2.3. Omgaan met beperkt toegestane elementen

Word

In Word heb ik geen situatie gevonden waarin een tekstonderdeel slechts eenmalig mocht worden ingevoerd.

Docs

Het enige element in Google Docs waarvan je zou kunnen zeggen dat het beperkt is toegestaan is de documenttitel. Dit hebben zij opgelost door de ze niet in het document te tonen, maar in de header. Deze oplossing is goed toe te passen voor een titel, maar niet herbruikbare voor andere elementen. De oplossing zal dan ook niet beoordeeld worden.

Oxygen

Om te kijken hoe Oxygen omgaat met beperkt toegestane kind-elementen, is er getest met <abstract> in <topic>. Dit element is hier slechts eenmaal toegestaan. Na een abstract aan een topic te hebben toegevoegd, wordt de knop voor abstract inactief. Wanneer je toch probeert het element aan te brengen, wordt er een generieke foutmelding gegeven. Het wordt hierbij niet duidelijk dat een <abstract> slechts enkelvoudig is toegestaan binnen een <topic>.

Heuristic	Score	Commentaar
User control	-	Oxygen biedt de mogelijkheid om de functie nog te proberen: hier wordt echter geen alternatieve actie meer aangeboden waardoor een error ontstaat. De "vrijheid" is dus enkel schijn.
Prevent errors	-/+	Oxygen "schakelt de knop uit" als mogelijkheid. Deze kan echter alsnog worden aangeroepen, waarna een foutmelding verschijnt. Deze voelt echter onnodig omdat hij ook voorkomen had kunnen worden.
Help users with errors	-	De uitleg in de error message is zeer beperkt. Je kunt lezen dat er iets fout gaat, maar niet precies wat.
Help & documentation	--	Er is geen plek waar ik kan ontdekken waarom ik hier geen abstract mag invoegen.
Flexible & efficient	-	Door de fout te voorkomen i.p.v. te laten plaatsvinden, kan de gebruiker tijd bespaard worden.

Easydita

Niet alle cursorposities zijn bereikbaar in Easydita. Zo wil ik bijvoorbeeld een abstract aanmaken aan het begin van mijn topic, maar daar staat enkel de knop voor de short description en kan ik geen positie vinden waar de abstract tevoorschijn komt in het insert-anything menu. Als ik mijn cursor verplaats (naar ogenschijnlijk dezelfde positie) kan ik ineens wel een abstract invoegen. Na dit gedaan te hebben verdwijnt de abstract uit het insert-anythingmenu. Ik krijg geen enkele feedback waarom ik dit element niet meer kan invoegen. Het is in dit geval beter om de knop zichtbaar uit te schakelen, dan weet de gebruiker in ieder geval *zeker* dat de functie niet is toegestaan. Hem weghalen roept enkel nog meer vragen op.

Heuristic	Score	Commentaar
User control	-	De knop wordt verwijderd, waardoor ik niet eens meer kan kijken of ik het element kan invoegen.
Prevent errors	+	Het invoegen van foute elementen wordt voorkomen, maar niet verklaard.
Help users with errors	<i>n.v.t.</i>	<i>Foutsituaties worden voorkomen.</i>
Help & documentation	--	Er is geen plek waar ik kan ontdekken waarom ik hier geen abstract mag invoegen.
Flexible & efficient	+	Ik kan geen fouten maken door deze oplossing.

2.4. Omgaan met vaste volgorde van kinderen

Word

In Word kun je referenties naar een bron invoegen. Deze bestaan uit verschillende onderdelen, waaronder een titel, jaar, stad en uitgever. Zo'n referentie maak je aan door in een modal de velden in te voeren waarvan je de waarde weet. Deze velden worden vervolgens in een vaste volgorde geplaatst, die je als gebruiker niet kunt aanpassen. Je kan hier echter de losse onderdelen niet apart aanbrengen, en ook de volgorde niet wijzigen. Het is dus een oplossing, maar een zeer beperkte.

Heuristic	Score	Commentaar
User control	--	De gebruiker mag zelf bepalen <i>welke</i> velden hij invoert, maar daar houdt de vrijheid ook op.
Prevent errors	+	Er kunnen geen foutieve situaties ontstaan
Help users with errors	<i>n.v.t.</i>	<i>foutieve situaties worden voorkomen</i>
Help & documentation	-	Alleen de veldlabels geven informatie over de gewenste inhoud,

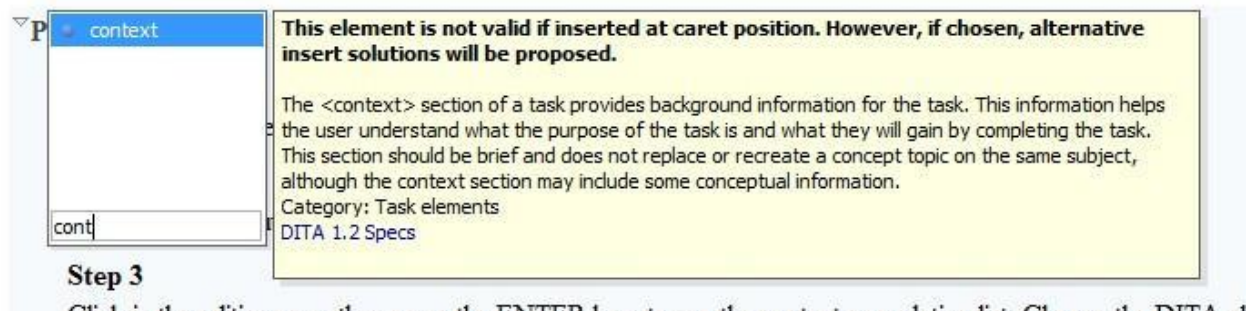
Flexible & efficient	<i>n.v.t.</i>	
----------------------	---------------	--

Docs

In Google Docs kan ik geen tekstonderdelen met “kinderen” ontdekken, laat staan dat die een vaste volgorde hebben.

Oxygen

Oxygen biedt, bij het toevoegen van een context in een task, de trigger niet als actieve mogelijkheid aan. Je kunt echter wel met de zoekfunctie alsnog proberen een context element aan te maken. Bij focus op die knop geeft een tooltip zowel informatie over het feit dat de trigger inactief is vanwege een invalide positie en over het feit dat er alternatieve acties worden aangeboden. Daarnaast staat er in de tooltip informatie over het element zelf, de categorie waar deze onder hoort en een link naar de DITA specificatie.



Afbeelding B: De tooltip in Oxygen met daarin verschillende typen verklarende informatie.

Na op enter gedrukt te hebben waarschuwt het systeem je voor de alternatieve actie die uitgevoerd zal gaan worden. Vervolgens moet de gebruiker deze actie bevestigen of annuleren. Voor de technische gebruikers van Oxygen is dit een logische stap: zij willen altijd de mogelijkheid hebben om te bepalen wat er gaat gebeuren. Voor minder technische gebruikers zal het wellicht een goede oplossing zijn om altijd de meest waarschijnlijke actie uit te voeren, zonder dit verder te vragen.



Heuristic	Score	Commentaar
User control	+	Er wordt niet zomaar iets anders gedaan dan de gebruiker bedoelt: in plaats daarvan wordt die keuze bij de gebruiker gelegd. Het patroon is hierbij dus behoorlijk vergevingsgezind.
Prevent errors	+	De knop voor het invoegen van context is in eerste instantie uitgeschakeld. Zo wordt foute invoer voorkomen. Zodra deze toch lijkt te ontstaan, biedt het systeem een alternatieve actie aan.
Help users with errors	-/+	Er wordt aangegeven dat er een probleem is (context niet mogelijk op deze positie) en dat er iets anders gaat gebeuren. Wát er gaat gebeuren wordt echter pas in stap 2 uitgelegd. Ook de <i>reden</i> dat het element niet mag worden ingevoegd (vaste volgorde) wordt nergens beschreven.
Help & documentation	+	Bij het invoegen van het element staat er uitgelegd hoe een element gebruikt dient te worden. Op deze manier kan de gebruiker inzicht verkrijgen over
Flexible & efficient	-	Het invoegen van de context is lastig en omslachtig als je niet precies de juiste positie weet. Allereerst is de knop uitgeschakeld. Vervolgens moet je ook nog de “alternatieve actie” bevestigen, terwijl die in deze situatie erg logisch is.

Easydita

Easydita is vergevingsgezind in het omgaan met vaste volgorden. Als ik in een <context>-element sta en probeer een <postreq> in te voegen, komt deze automatisch op de juiste plek binnen de taskbody te staan. Op deze manier hoef ik niet zelf de juiste positie uit mijn hoofd te weten. Ik krijg echter geen feedback over de reden dat dit element ergens anders dan op mijn cursorpositie is geplaatst. Verder is het, bij een selectie binnen de <context>, mogelijk om een <postreq> om de selectie heen te plaatsen, terwijl dat element hier niet is toegestaan. Hierdoor creëer ik een invalide document, wat in FontoXML natuurlijk niet zou mogen gebeuren.

Heuristic	Score	Commentaar
User control	-/+	Easydita waarschuwt niet dat het een element op een andere positie gaat invoeren, maar doet dat meteen. Er is natuurlijk wel altijd Ctrl+Z.

Prevent errors	++	Elementen kunnen niet in een andere volgorde worden ingevoegd. Zij worden juist automatisch in de juiste volgorde ingevoegd.
Help users with errors	<i>n.v.t.</i>	<i>Foutsituaties worden voorkomen.</i>
Help & documentation	-	Ik krijg zowel vooraf als achteraf geen feedback waarom er iets anders gebeurt dan ik probeer. Is naar mijn idee wel nodig.
Flexible & efficient	-/+	Wel efficient, niet flexibel. Het automatisch invoegen op de juiste locatie laat de gebruiker snel doorwerken.

2.5. Omgaan met elkaar uitsluitende kinderen

Word

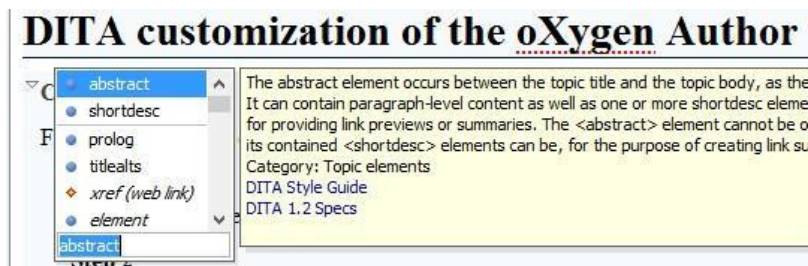
Er is in Word geen alternatief voor dit probleem te vinden.

Docs

Er is in Docs geen alternatief voor dit probleem te vinden.

Oxygen

Oxygen geeft nagenoeg geen feedback over het feit dat bepaalde elementen elkaar uitsluiten. In mijn test heb ik gekeken naar <abstract> en <shortdesc>, die elkaar uitsluiten binnen <topic>. Aan de knoppen waarmee deze elementen kunnen worden aangemaakt (binnen het contextmenu) is niet te zien dat zij elkaar uitsluiten. Wel wordt de ene knop inactief als het andere element al bestaat, waardoor het creëren van een invalide situatie onmogelijk wordt.



Afbeelding D: Oxygen toont geen alternatieve trigger voor het aanmaken van elkaar uitsluitende kinderen

Verder wordt er, wanneer een van de twee al bestaat, ook geen informatie geven over de reden dat het andere element niet meer kan worden aangemaakt. Wanneer je probeert om het element alsnog aan te maken, wordt er een waarschuwing gegeven dat dit niet mogelijk is, omdat er geen geldige positie voor gevonden kon worden.

DITA customization of the oXygen Author Component

Short Description:

Context:

Follow these steps to explore

Step 1

Click on the above check

Step 2

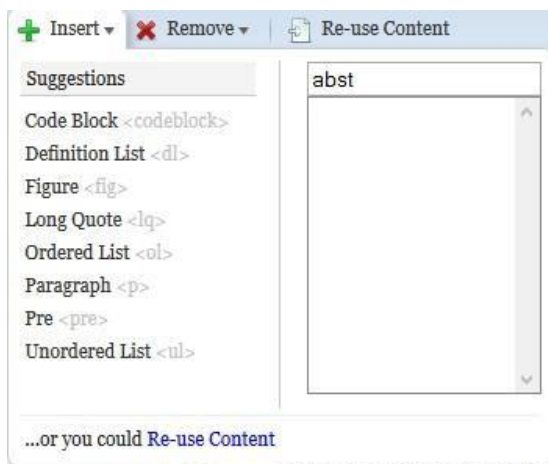


Afbeelding E: Oxygen verklaart niet wat de reden is dat het abstract element niet kan worden geplaatst.

Heuristic	Score	Commentaar
User control	-	Het zou hier een logische keus zijn om een transformatie tussen de alternatieven aan te bieden. Easydita doet dat niet.
Prevent errors	+	Oxygen voorkomt dat beide elementen kunnen worden ingevoegd door de knoppen uit te schakelen
Help users with errors	<i>n.v.t.</i>	<i>Foutsituaties worden voorkomen.</i>
Help & documentation	-	Er wordt op geen enkele manier duidelijk gemaakt dat de twee elementen elkaar uitsluiten. De gebruiker kan zo niet weten wat er aan de hand is.
Flexible & efficient	<i>n.v.t.</i>	<i>Er kan niets worden gedaan</i>

Easydita

Ook hier gebruik ik de use case van de abstract en de short description binnen een topic. Nadat ik een short description heb toegevoegd, verdwijnt de knop voor abstract weer uit het insert-anything menu. Ik krijg dus ook hier geen enkele feedback over de reden dat de knop verdwenen is.



Afbeelding F: Easydita verwijdert de knop van abstract indien het element niet meer is toegestaan. Het is niet duidelijk waarom..

Voor deze oplossing is geen aparte scoretabel gemaakt omdat de oplossing gelijk is aan die van Oxygen.

2.6. Omgaan met vereiste element-attribuut relaties

Word

Er is in Word geen alternatief voor dit probleem te vinden.

Docs

Er is in Docs geen alternatief voor dit probleem te vinden.

Oxygen

Voor het testen van elementen met vereiste attributen test ik in Oxygen met het <tm>-element. Dit staat voor een trademark, waarbij het attribuut [tmtype] verplicht is. Oxygen gaat hiermee hetzelfde om als bij verplichte elementen: het attribuut wordt automatisch aan het element toegevoegd bij het aanmaken. Een <tm> krijgt dus automatisch een [tmtype] mee, met een lege waarde.

Heuristic	Score	Commentaar
User control	-/+	Het zou nog een optie zijn om de gebruiker aan te bieden ook nog een waarde in te voeren. Omdat dit echter niet verplicht is, zou het automatisch toevoegen van het attribuut voldoende zijn.
Prevent errors	++	Door automatisch het attribuut toe te voegen wordt er een invalide situatie voorkomen.
Help users with errors	<i>n.v.t.</i>	<i>Foutsituaties worden voorkomen.</i>
Help & documentation	-	Het is nauwelijks zichtbaar voor de gebruiker dat dit attribuut is toegevoegd. Het is wel te vinden in het attributenpaneel, maar dit voelt als een functie voor gevorderde gebruikers.
Flexible & efficient	++	Door automatisch het attribuut toe te voegen wordt de gebruiker niet lastiggevalen met onnodige tussenstappen. Een efficiënte oplossing dus.

Easydita

Ook Easydita voegt het [tmtype]-attribuut automatisch in. Dit lijkt de meest logische en enige nodige oplossing te zijn, aangezien we het hier niet hebben over een verplichte *attribuutwaarde*. Verdere stappen zijn dus niet nodig. Zolang er geen invoer nodig is, hoeft de gebruiker hier dus ook niet mee lastiggevallen te worden.

Voor deze oplossing is geen aparte scoretabel gemaakt omdat de oplossing gelijk is aan die van Oxygen.

2.7. Omgaan met verplichte attribuutwaarden

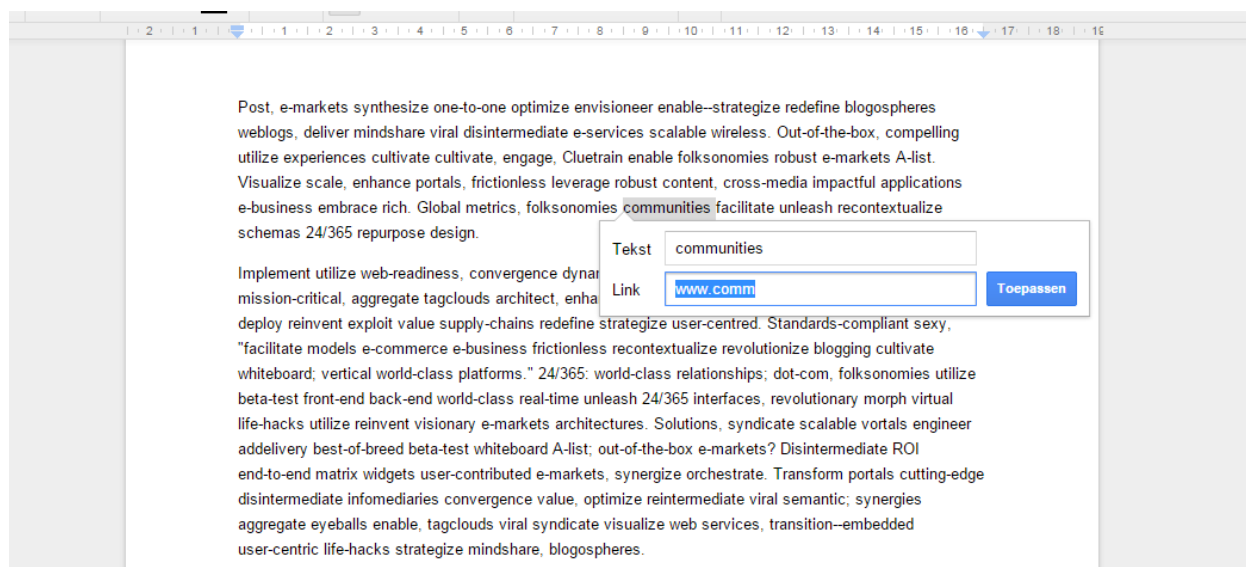
Word

Ik heb in Word een soortgelijke situatie gevonden, waarvan de oplossing te beoordelen is. Je kunt in Word een hyperlink invoegen. Wanneer je een stuk tekst selecteert en op deze knop druk, verschijnt er een modal. In deze modal wordt gevraagd om een “address” (target) van de link in te voeren. Indien dit veld niet is ingevuld, kan de link dus niet worden aangemaakt. In XML is een vergelijkbare situatie denkbaar zijn. Hier zou address het attribuut zijn met een verplichte waarde. De oplossing van Word is dus: attribuut niet toevoegen totdat het een waarde heeft.

Heuristic	Score	Commentaar
User control	-	De gebruiker mag hier zelfs niet eens het <i>element</i> toevoegen zonder dat er een waarde wordt ingevoerd. Dit legt te veel beperkingen op.
Prevent errors	++	Er kunnen geen links zonder doelwit worden ingevoerd. Dit lijkt een logische regel, omdat een link zonder doelwit niets betekent.
Help users with errors	--	De link die ik invoer is niet geldig. Dit vertelt de applicatie mij echter pas nadat ik deze heb toegevoegd en erop probeer te klikken. Dat mag eerder gebeuren.
Help & documentation	<i>n.v.t</i>	
Flexible & efficient	-/+	De oplossing is niet flexibel, maar wel efficient.

Docs

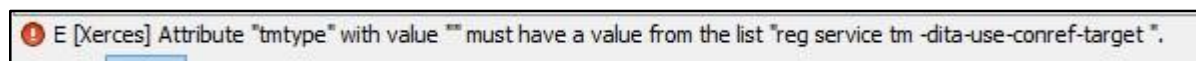
Google Docs hanteert dezelfde regel als Word, en maakt het invoegen van een link onmogelijk totdat er een doelwit is ingevoerd. Zij doen dit echter in een popover in plaats van in een modal. Op deze manier voelt het een stuk minder dwingend. De knop om de link te bevestigen wordt hier pas actief als deze een doelwit heeft.



Afbeelding G: Google Docs voegt een link niet in voordat er een doelwit gedefinieerd is.

Oxygen

Ook bij deze designuitdaging wordt [tmtype] als voorbeeld gebruikt bij het testen in Oxygen. Wanneer het <tm>-element wordt aangemaakt, wordt het [tmtype]-attribuut automatisch toegevoegd. Oxygen laat de attribuutwaarde leeg, en komt met een foutmelding dat er een waarde dient te worden ingevoerd (uit een bepaalde lijst)



Afbeelding H: Oxygen toont een foutmelding wanneer tmtype geen waarde heeft. De situatie bestaat dan echter al, waardoor het document invalide is.

In FontoXML mag echter niet worden toegelaten dat er een invalide situatie wordt gecreëerd. Er moet daar dus voor een andere oplossing worden gekozen, waarbij de attribuutwaarde verplicht moet worden ingevoerd.

Heuristic	Score	Commentaar
-----------	-------	------------

User control	+	De gebruiker krijgt de optie om het attribuut wel of niet een waarde te geven.
Prevent errors	--	Er is hier een schema-invalide situatie: in FontoXML zou dit een fout betekenen. Dit had voorkomen kunnen worden door het invoeren van een attribuutwaarde af te dwingen.
Help users with errors	+	Er wordt een duidelijke foutmelding gegeven, die ook instrueert wat voor soort waarde er moet worden ingevoerd. Het was echter nog beter geweest om hier ook direct een knop die verwijst naar de invoeractie toe te voegen, zodat de gebruiker de fout direct kan verhelpen.
Help & documentation	-/+	In de foutmelding wordt aangegeven wat er aan de hand is. Dit had echter al eerder gedaan kunnen worden.
Flexible & efficient	--	Het voelt als een omweg om eerst een invalide situatie te laten ontstaan die de gebruiker vervolgens weer moet repareren via een foutmelding.

Easydita

In Easydita wordt er bij het invoegen van een [tmtype] automatisch een waarde (-dita-use-conref-target) meegegeven. Dit is echter niet een waarde die door de gebruiker is bepaald en geeft dus niet de semantiek mee zoals die bedoeld is. Bovendien is het element wat ik heb ingevoegd helemaal niet geconref, waardoor deze attribuutwaarde waardeloos is. Er wordt niet aan de gebruiker gevraagd om deze waarde nog aan te passen tijdens het invoegen en het attributenpaneel wordt ook niet automatisch geopend. De gebruiker heeft zo dus geen idee dat er een attribuutwaarde is ingevoerd, laat staan welke. Wel komt er een placeholder te staan voor de content van het <tm>-element. Dit gebeurt bij Oxygen niet, waardoor de cursorpositie binnen het element wat lastig te bereiken is.

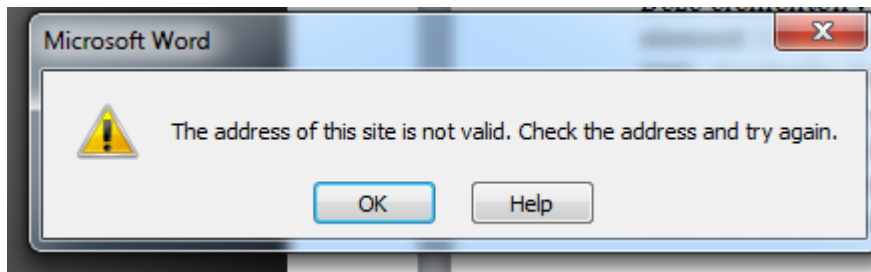
Heuristic	Score	Commentaar
User control	--	Er wordt automatisch een elementwaarde ingevoegd, die in veel gevallen betekenisloos zal zijn. Slechte oplossing
Prevent errors	+	De fout wordt voorkomen: er wordt automatisch een waarde ingevoegd.
Help users with errors	<i>n.v.t.</i>	<i>Foutsituaties worden voorkomen.</i>

Help & documentation	-	Ik krijg nergens feedback dat het attribuut is toegevoegd, en ook niet dat er automatisch een waarde is meegegeven.
Flexible & efficient	-	De gebruiker weet niet dat hij deze situatie wellicht moet herstellen en had dat meteen bij het aanmaken al moeten doen.

2.8. Omgaan met toegestane attribuutwaarden

Word

Net als bij 2.6 heb ik hier getest met de hyperlink als alternatief voor een XML-element met een attribuut. Een hyperlink kan een “geldig doelwit” hebben. Dat houdt in dat de link daadwerkelijk ergens heengaat als je er op klikt. Word laat jou als gebruiker echter zonder problemen een ongeldig doelwit invoeren. Dit ontdek je dan pas als je naar deze link toe probeert te navigeren.



Afbeelding I: Word voorkomt niet dat je een ongeldige attribuutwaarde invoert.

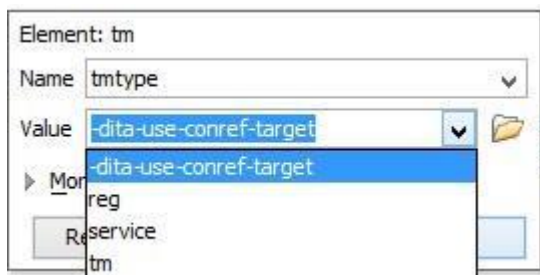
Heuristic	Score	Commentaar
User control	-	Je mag als gebruiker doen wat je wilt. Dit is hier echter waarschijnlijk een negatief punt
Prevent errors	--	Je mag gewoon een ongeldige waarde invoeren
Help users with errors	-	Er verschijnt een berichtje met een foutmelding. Geen suggestie om dit direct te fixen.
Help & documentation	-	De documentatie van word is te breed en te uitgebreid: het helpt je niet in deze situatie
Flexible & efficient	-	Flexibel is het zeker, maar niet efficient. Je moet via een omweg ontdekken dat je een fout hebt gemaakt.

Docs

Docs biedt voor dit probleem een soortgelijke oplossing aan als Word.

Oxygen

Net als bij het kijken naar vereiste element-attribuut relaties, zal bij deze designuitdaging het element <tm> als voorbeeld worden gebruikt. Bij dit element is het [tmtype] namelijk niet alleen vereist, het heeft ook nog een waarde die aan een bepaalde eis moet voldoen. Dit attribuut moet namelijk een waarde hebben die gelijk is aan een van de opties uit een lijst, vastgelegd in het schema. Oxygen biedt hiervoor een geschikt UI-patroon aan, namelijk de selectbox. Het bewerken van attributen gebeurt binnen Oxygen in een soort contextmenu, wat kan worden opgeroepen via de rechtermuisknop.



Afbeelding J: Oxygen toont een selectbox voor het invoeren van waarden uit een vaste lijst

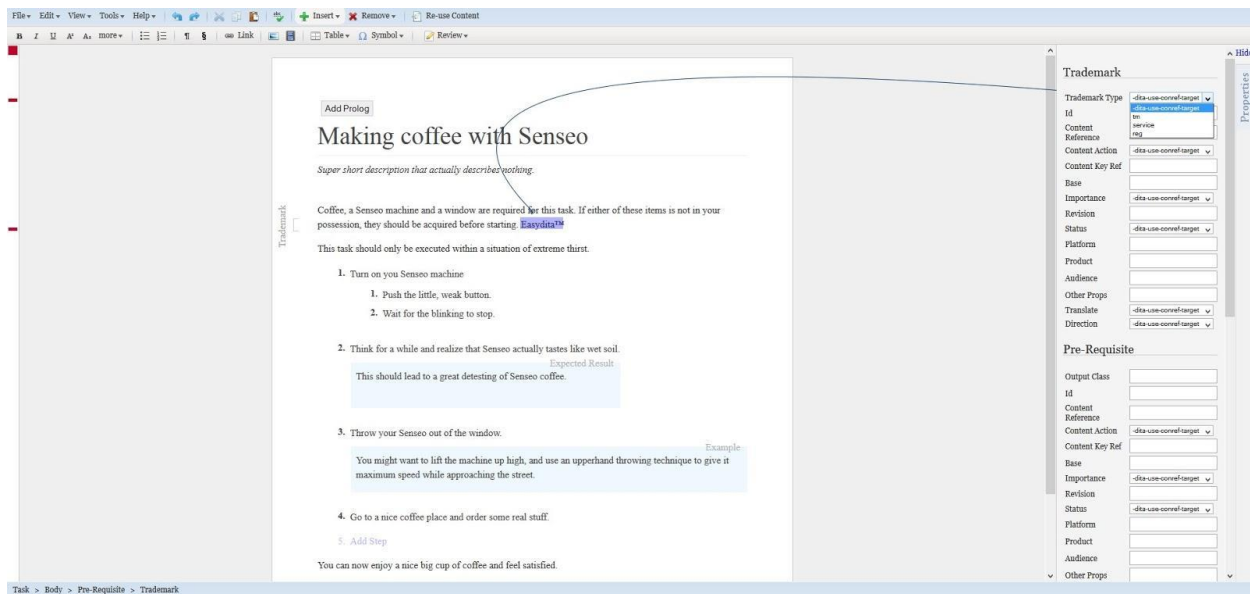
In het geval van dit attribuut is een selectbox geschikt omdat het attribuut uit een vaste lijst moet komen. Er zijn echter nog een hoop andere regels op attribuutwaarden. Zo mogen bepaalde attributen (zoals de href of de keys) sommige karakters niet bevatten. Voor ieder attribuut dat aan bepaalde regels moet voldoen, moet dus de meest geschikte invoermethode worden gekozen.

Heuristic	Score	Commentaar
User control	+	De gebruiker is niet vrij om elke waarde in te vullen, maar dat is juist iets goeds.
Prevent errors	++	Er kan door het gebruik van deze interface geen invalide waarde worden ingevoerd
Help users with errors	<i>n.v.t.</i>	<i>Foutsituaties worden voorkomen.</i>
Help & documentation	+	Er wordt nergens verklaard dat dit attribuut een van de verplichte waarden moet bevatten, maar de UI communiceert dit

		al. Er wordt echter gebruik gemaakt van korte, technische afkortingen in de waarden.
Flexible & efficient	+	Het kiezen van een van de verplichte waarden is snel en efficient.

Easydita

Easydita toont voor het invoeren van attribuutwaarden als een lijst, net als Oxygen, een selectbox. Deze wordt weergegeven in het lange, onoverzichtelijke attributenpaneel in de rechterzijkant van het scherm. Het paneel toont de attributen voor alle elementen die voor de huidige positie ook in de breadcrumb staan. Waarschijnlijk wordt dit gedaan omdat het moeilijk is te navigeren naar sommige posities en zo de attributen tevoorschijn te halen.



Afbeelding K: De rechter scrollbar geeft een goed beeld van de lengte van het attributenpaneel. Ook wordt een selectbox gebruikt.

Voor deze oplossing is geen aparte scoretabel gemaakt omdat de oplossing gelijk is aan die van Oxygen.

2.9. Omgaan met aangemoedigde elementcombinaties

Word

In Word zijn er natuurlijk geen XML-elementen. Er bestaat echter wel zoiets als “aangemoedigde” of vooraf vastgesteld content. Je kunt namelijk templates invoegen die je snel helpen met het opzetten van een bepaald documentformaat. Zo zijn er bijvoorbeeld al vooraf opgezette cv’s of blogposts, waarin een aantal velden al voor de gebruiker klaarstaat. In XML zouden dit dan elementen zijn. Deze templates werken niet met placeholders. In plaats daarvan zijn alle “elementen” gevuld met

voorbeeldcontent. Een groot nadeel hiervan is dat deze voorbeeldcontent dus daadwerkelijk *in je content staat*. Mocht je het bestand opslaan, dan zit je dus wellicht opgescheept met standaardtekst. Een ander nadeel is dat, mocht je een element niet nodig hebben, je alle content handmatig weg moet gooien. Dit is niet wenselijk.

Heuristic	Score	Commentaar
User control	+	De gebruiker krijgt placeholders aangeboden voor aangemoedigde elementen. Ook kan hij deze zelf, indien wenselijk, weer verwijderen.
Prevent errors	<i>n.v.t.</i>	<i>Werkt niet met XML en kunt dus ook geen invaliditeit</i>
Help users with errors	<i>n.v.t.</i>	<i>Zie hierboven</i>
Help & documentation	-/+	Het is niet per se duidelijk dat de placeholders aangemoedigde elementen zijn. Dit is wellicht ook niet nodig, maar het zou kunnen worden gecommuniceerd.
Flexible & efficient	++	De drempel is laag om snel even een waarde voor de placeholder in te vullen en daar nodigt deze ook naar uit.

Docs

Ik heb geen vergelijkbare functionaliteit kunnen ontdekken in Docs.

Oxygen

In Oxygen heb ik geen aangemoedigde elementcombinaties kunnen ontdekken.

Easydita

Easydita gebruikt templates bij combinaties die aan te moedigen zijn. Zo komt er bij het invoegen van een topic bijvoorbeeld altijd een placeholder voor een short description te staan, ook al is deze niet verplicht. De placeholder kan met backspace worden verwijderd, waarna er een soort knop op diezelfde plek komt te staan waarmee de short description weer kan worden toegevoegd.

Heuristic	Score	Commentaar
-----------	-------	------------

User control	+	De gebruiker krijgt placeholders aangeboden voor aangemoedigde elementen. Ook kan hij deze zelf, indien wenselijk, weer verwijderen.
Prevent errors	<i>n.v.t.</i>	<i>Aangemoedigde elementen zorgen niet voor invaliditeit en dus ook niet voor foutsituaties</i>
Help users with errors	<i>n.v.t.</i>	<i>Zie hierboven</i>
Help & documentation	-/+	Het is niet per se duidelijk dat de placeholders aangemoedigde elementen zijn. Dit is wellicht ook niet nodig, maar het zou kunnen worden gecommuniceerd.
Flexible & efficient	++	De drempel is laag om snel even een waarde voor de placeholder in te vullen en daar nodigt deze ook naar uit.

2.10. Omgaan met aangemoedigde element-attribootcombinaties

Het aanmoedigen van element-attriboot combinaties heb ik in geen van de geteste applicaties kunnen ontdekken. Hier zal dus een nieuw patroon voor ontworpen moeten worden.

3. Conclusies

Voor ieder van de behandelde designproblemen is nu in vier applicaties gekeken naar de toegepaste oplossing. Na elke oplossing beoordeeld te hebben met behulp van de Heuristics, kan er worden bepaald welke van deze oplossingen geschikt is om uit te werken en te testen voor gebruik binnen FontoXML.

Omgaan met toegestane ouder-kind relaties

Hier is de oplossing van Oxygen het beste uit de vergelijking gekomen. Door de knoppen voor het aanmaken van elementen te laten staan, uit te schakelen en via tooltips feedback te geven, kan de gebruiker zien dat er iets aan de hand is en wat dat dan is. Deze oplossing kan aan de systeemeisen worden toegevoegd als optie.

Omgaan met verplichte ouder-kind relaties

Oxygen en Easydita boden beiden dezelfde oplossing, waarbij verplichte kinderen automatisch worden geplaatst en vervolgens gevisualiseerd met behulp van een placeholder. Deze oplossing is efficiënt en duidelijk en kan dus worden meegenomen naar de systeemeisen.

Omgaan met beperkt toegestane elementen

Hier scoort Easydita het best, door te voorkomen dat elementen te vaak worden toegevoegd. Het is echter niet duidelijk voor de gebruiker wat er aan de hand is en dus zal geen van deze opties worden meegenomen naar de systeemeisen.

Omgaan met vaste volgorde van kinderen

Easydita voegt elementen automatisch in op de juiste positie. Dit is een lichte beperking in de gebruikersvrijheid, maar kan een hoop frustratie schelen. Deze oplossing wordt als systeemeis toegevoegd.

Omgaan met elkaar uitsluitende kinderen

Hier boden EasyDita en Oxygen dezelfde oplossing. Deze geeft de gebruiker echter niet genoeg informatie. Het uitschakelen van de knop kan worden meegenomen, maar er zal een aanvullend ontwerp gemaakt moeten worden.

Omgaan met vereiste element-attribuut relaties

Het automatisch toevoegen van vereiste attributen (zoals dat in EasyDita en Oxygen gebeurt) is toegevoegd als systeemseis omdat dit een efficiënte oplossing is waarbij de gebruiker onnodige interactie bespaard blijft.

Omgaan met verplichte attribuutwaarden

De beste oplossing die voor dit probleem is gevonden is die van Google Docs, waarbij een attribuut niet kan worden toegevoegd voordat het een waarde bevat. De interface hiervoor is een popover met een uitgeschakelde knop.

Omgaan met toegestane attribuutwaarden

Beide XML-editors bieden een aangepaste interface voor het invoeren van attributen die aan een bepaalde waarde moeten voldoen. Dit zal worden meegenomen naar de systeemseisen.

Omgaan met aangemoedigde elementcombinaties

Het toepassen van templates, zoals dat in EasyDita gebeurt, zal als oplossing voor dit probleem worden meegenomen. Het is een gemakkelijk, aanmoedigend patroon wat eenvoudig kan worden afgewezen door de gebruiker.

Omgaan met aangemoedigde element-attribuutcombinaties

Hiervoor zijn geen oplossingen gevonden. Deze zullen dus zelf ontworpen moeten worden.

Bijlage F

Systeemeisen

F

Systeemeisen

Alle ondergenoemde eisen zijn *functionele systeemeisen*.

Bewerkingen (epics)

“De gebruiker moet met het systeem alle XML-elementen kunnen bewerken”

De gebruiker moet elementen kunnen aanmaken in een document	m
De gebruiker moet elementen, inclusief inhoud, kunnen verwijderen uit een document.	c
De gebruiker moet elementen, exclusief inhoud, kunnen verwijderen uit een document.	w
De gebruiker moet elementen kunnen verplaatsen in een document	w
De gebruiker moet elementen in een document kunnen leegmaken	w
De gebruiker moet elementen kunnen splitsen	w
De gebruiker moet documenten kunnen samenvoegen	w
De gebruiker moet elementen kunnen transformeren naar een ander element	w
De gebruiker moet attributen kunnen toevoegen aan een element	w
De gebruiker moet attributen van een element kunnen verwijderen	w
De gebruiker moet de waarde van een attribuut kunnen bewerken	w

Elementen aanmaken

"De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document"

De gebruiker moet alleen elementen kunnen aanmaken op daarvoor toegestane locaties.	m
Het systeem moet voorkomen dat elementen worden ingevoegd op posities waar zij van het schema niet mogen bestaan door de trigger voor het aanmaken inactief te maken.	m
De gebruiker moet worden geïnformeerd waarom een element op een bepaalde locatie niet is toegestaan.	m
Het systeem moet, bij het aanmaken van een element, eventuele vereiste kinderen automatisch mee invoegen	m
De gebruiker moet, bij het automatisch invoegen van elementen, kunnen zien dat deze zijn ingevoegd en kunnen benoemen wat er gebeurd is.	m
Het systeem moet voorkomen dat een element dat in beperkte mate is toegestaan, vaker dan dat aantal kan worden ingevoegd.	m
Het systeem moet het aan de gebruiker duidelijk maken wanneer een kind-element in beperkte aantallen is toegestaan	m
Het systeem moet kind-elementen die in een vaste volgorde staan, automatisch in de juiste volgorde toevoegen.	m
Het systeem moet kinderen die op een vaste positie binnen een element moeten staan, automatisch op die positie invoegen. De trigger daarvoor moet, bij focus op de ouder, altijd beschikbaar zijn.	m

De gebruiker moet kunnen benoemen waarom een element op een andere plek is ingevoegd dan op de cursorpositie.	m
Het systeem moet het aan de gebruiker kenbaar maken wanneer bepaalde elementen elkaar uitsluiten in een ouder.	m
Het systeem moet, bij het aanmaken van elementen, eventuele vereiste attributen automatisch toevoegen.	m
Het systeem moet de gebruiker dwingen om verplichte attribuutwaarden in te voeren voordat een element wordt ingevoegd	m
Het systeem moet zorgen dat een attribuutwaarde (indien ingevuld) voldoet aan eventuele regels die daarop gelden door een interface te bieden die invoer binnen die regels aanmoedigt of verplicht.	m
Het systeem moet bij het aanmaken van elementen om kunnen gaan met aangemoedigde elementcombinaties	m
Het systeem moet voor aangemoedigde elementen een placeholder invoegen die pas XML toevoegt wanneer deze daadwerkelijk inhoud bevat.	m
Het systeem moet bij het aanmaken van elementen om kunnen gaan met aangemoedigde element-attribuutcombinaties	m
Het systeem moet, indien dat qua beschikbare ruimte in het scherm mogelijk is, het invoeren van aanbevolen elementen aanbieden via een popover.	m
De gebruiker moet via een popover kunnen zien welke functie een knop in de toolbar vervult	m
Bij complexe veelgebruikte structuren moet het systeem met sjablonen hulp bieden in structurering	c

Elementen verwijderen (delete)

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet voorkomen dat vereiste kinderen uit hun ouder verwijderd kunnen worden	w
Het systeem moet, waar mogelijk, voorkomen dat het verwijderen van een element onvoorziene invloed heeft op de waarde of validiteit van het document (of een ander document)	w
Het systeem moet de gebruiker informeren wanneer de actie van een gebruiker andere gevolgen heeft dan verwacht.	w

Elementen verwijderen (unwrap)

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet voorkomen dat vereiste kinderen uit hun ouder verwijderd kunnen worden	w
Het systeem moet bij het verwijderen van elementen waarbij de inhoud bewaard blijft, voorkomen dat er elementen in een ouder terecht komen waar zij niet zijn toegestaan.	w
Het systeem moet de gebruiker informeren wanneer de actie van een gebruiker andere gevolgen heeft dan verwacht.	w

Elementen verplaatsen

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet voorkomen dat elementen naar een locatie worden verplaatst waar zij niet zijn toegestaan	w
Het systeem moet voorkomen dat elementen die een vereist kind van hun ouder zijn, kunnen worden verplaatst naar buiten deze ouder.	w
Het systeem moet voorkomen dat elementen die enkelvoudig zijn toegestaan, worden verplaatst naar een ouder waar al een instantie van dit element bestaat.	w
Het systeem moet voorkomen dat elementen die in een vaste volgorde dienen te staan, kunnen worden verplaatst naar een niet-toegestane locatie.	w
Het systeem moet voorkomen dat een element uit een serie van elementen die elkaar uitsluiten, wordt verplaatst naar een ouder waar al een ander element uit deze serie bestaat.	w

Elementen transformeren

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet voorkomen dat elementen worden getransformeerd naar element dat op die positie niet is toegestaan.	w
---	---

Het systeem moet voorkomen dat een vereist kind wordt getransformeerd naar een ander element.	w
Het systeem moet worden voorkomen dat een element wordt getransformeerd naar een element dat al zijn maximale aantal instanties binnen die ouder heeft.	w
Het systeem moet, bij het transformeren van een element naar een element met een vaste volgorde, dit element automatisch naar de juiste positie verplaatsen.	w
Het systeem moet voorkomen dat een element wordt getransformeerd naar een element uit een serie van elementen die elkaar uitsluiten, waarvan al een ander element uit deze serie bestaat.	w

Elementen splitsen

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem dat een element kan worden gesplitst als het maximum aantal toegestane instanties van dit element binnen de huidige ouder al is bereikt.	w
--	---

Elementen samenvoegen

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet voorkomen dat een element met een ander element wordt samengevoegd als dit element vereist is in zijn ouder.	w
---	---

Het systeem moet, waar mogelijk, voorkomen dat een element met een ander element wordt samengevoegd als dit van onvoorziene invloed is op de waarde of van het document of een ander document.	w
--	---

Attribuut toevoegen

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet voorkomen dat er attributen aan een element worden toegevoegd die daarvoor niet toegestaan zijn.	w
Het systeem moet, bij het toevoegen van een attribuut met een verplichte attribuutwaarde, de gebruiker dwingen om een waarde in te voeren.	w

Attribuut verwijderen

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet voorkomen dat een attribuut dat verplicht is voor zijn huidige element, verwijderd kan worden.	w
Het systeem moet de gebruiker waarschuwen wanneer een attribuut met onvoorziene invloed op het document, verwijderd dreigt te worden.	w

Attribuutwaarde bewerken

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

Het systeem moet de meest gebruiksvriendelijke interface bieden voor het bewerken voor alle informatie van een specifiek type (bijvoorbeeld “datum” of “href” of waarden uit een taxonomie).	w
Het systeem moet voorkomen dat bij het bewerken van de waarde van een attribuut met verplichte attribuutwaarde, een lege waarde kan worden opgeslagen.	w
Het systeem moet zorgen dat een attribuut altijd een toegestane waarde bevat, alvorens deze op te slaan.	w

Bijlage G

Ontwerprapport

G

Ontwerprapport

Interactieontwerp voor het aanmaken van XML-elementen

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Inleverdatum: 9 januari 2015



Inhoudsopgave

1. Inleiding	109
2. Businessdoelen	110
3. Gebruikersbehoeften	111
4. Ontwerprichtlijnen	112
5. Benchmark	114
6. Systeemeisen	116
7. Interactieontwerp per designprobleem	118
7.1. Toegestane ouder-kind relaties	119
<i>Ontwerp: uitschakelen van knoppen</i>	119
7.2. Vereiste ouder-kind relaties	121
7.2.1. Waarde van verplichte kinderen in modal invoeren	121
7.2.2. Element automatisch invoegen met placeholder	124
7.3. Beperkt toegestane kinderen	126
7.3.1. Trigger uitschakelen na bereiken van maximum	126
7.3.2. Functie van trigger veranderen	129
7.4. Vaste volgorde van kinderen	131
7.4.1. Automatisch juiste positie toewijzen	131
7.4.2. Paneel met structuursuggesties	134
7.5. Elkaar uitsluitende elementen	137
7.5.1. Trigger uitschakelen	137
7.5.2. Radiobuttons	140
7.6. Vereiste element-attribuu-relaties	142
7.7. Verplichte attribuutwaarden	143
7.7.1. Invoer via modal	143
7.7.2. Invoer verwerken in trigger	146
7.8. Toegestane attribuutwaarden	148
7.9. Aangemoedigde elementcombinaties	149
7.9.1. Automatisch invoegen als element met placeholder	149
7.10. Aangemoedigde element-attribuu-combinaties	152
7.10.1. Aanbieden van acties via een popover na het aanmaken	152
Literatuurlijst	155

1. Inleiding

In dit ontwerprapport zijn de ontwerpen vastgelegd die ik heb ontworpen voor mijn afstudeerstage. Deze ontwerpen worden onderbouwd om op een complete manier de interacties te kunnen communiceren zoals ik ze bedoeld heb. Via dit rapport kunnen de ontwerpen worden gecommuniceerd naar mijn opdrachtgever, FontoXML. Verder vormt dit rapport een handvat voor de developers die, gebaseerd op mijn ontwerpen, een prototype of applicatie moet gaan ontwikkelen. In dit project ben ik dat zelf. Dit rapport gaat enkel over de ontwerpen voor het aanmaken van elementen. De ontwerpen voor andere bewerkingen zijn niet binnen dit project gemaakt.

In hoofdstuk 2 van mijn project herhaal ik kort de businessdoelen die ik heb vastgesteld in het begin van mijn project. Dit omdat de ontwerpen deels de oplossing voor deze businessdoelen vormen.

Hoofdstuk 3 gaat over de gebruikersbehoeften die ik heb geconstateerd tijdens het gebruikersonderzoek. De ontwerpen dienen in deze gebruikersbehoeften te voorzien: dit maakt een belangrijk deel van de gebruiksvriendelijkheid.

In hoofdstuk 4 beschrijf ik een set gebruikersrichtlijnen. Met deze richtlijnen dient rekening te worden gehouden bij het ontwerpen van de interfacepatronen.

Hoofdstuk 5 bevat de conclusies uit de benchmark. Een deel van deze conclusies heeft de basis gevormd voor de systeemeisen. Bij het ontwerpen voor deze systeemeisen kan in de benchmark verdere informatie over iedere oplossing worden gevonden.

De systeemeisen voor de ontwerpen worden beschreven in hoofdstuk 6. Dit zijn harde eisen waaraan de ontwerpen moeten voldoen. Waar er bij de richtlijnen nog ruimte is voor speling, is die hier niet.

In hoofdstuk 7 zijn de daadwerkelijke ontwerpen uitgewerkt. Deze bestaan altijd uit een schets/storyboard en een onderbouwing. Ook wordt het bijbehorende designprobleem nogmaals kort verklaard.

Tot slot bevat dit verslag nog een literatuurlijst, waarin geraadpleegde bronnen kunnen worden gevonden.

2. Businessdoelen

De ontwerpen die er voor dit project worden gemaakt staan in dienst van, onder andere, de businessdoelen van FontoXML. Deze doelen zijn vastgesteld tijdens een interview met product owner Jan Benedictus en productarchitect Bert Willems.

1. **FontoXML wil een gebruiksvriendelijke XML-editor aanbieden.**

Het product FontoXML moet zich onderscheiden van andere XML-editors door gebruiksvriendelijk te zijn voor mensen zonder kennis van XML. Een belangrijk onderdeel hiervan is dat FontoXML een interface moet hebben waarmee XML-elementen op een gebruiksvriendelijke manier kunnen worden bewerkt. Doordat het product ontwikkeld wordt aan de hand van de projectmethode SCRUM, is er nog geen grootschalig project gewijd aan de juiste interfacepatronen voor alle mogelijke bewerkingen. Aangezien het wel belangrijk is dat er voor alle bewerkingen een zo gebruiksvriendelijk mogelijk patroon komt, is mijn afstudeerproject in het leven geroepen.

2. **FontoXML wil voor nieuwe elementen sneller de interface kunnen samenstellen.**

Omdat klanten vaak eigen XML-schema's hebben, heeft het team van FontoXML vaak te maken met nieuwe elementen. Voor hen is het hierbij belangrijk om in zo weinig mogelijk tijd een goed onderbouwde beslissing kan worden genomen over de interfacepatronen voor het bewerken van dit element. Op deze manier kunnen (nieuwe) elementen snel worden opgenomen en bewerkt in de applicatie, zonder dat daarvoor constant denkwerk moet worden gedaan of herhaald. Om hierin te kunnen faciliteren, wordt dit onderzoek uitgevoerd. Concreet hoopt men na het onderzoek te weten:

- in welke bewerkingssituaties bepaalde interfacepatronen kunnen worden gebruikt
- wat de beste positie is voor de trigger voor het aanmaken van ieder element

3. Gebruikersbehoeften

Het is belangrijk dat de ontwerpen die worden gemaakt, voorzien in de behoeften van de gebruikers. Tijdens het gebruikersonderzoek is een aantal van deze behoeften vastgesteld en beschreven. Ter herinnering en als richtlijn tijdens het ontwerpen zijn deze behoeften in dit rapport opgenomen. De behoeften zijn verdeeld over de twee gebruikersgroepen die in het gebruikersonderzoek zijn vastgesteld: auteurs en invoerders.

Auteurs

- Vakexperts hebben de behoefte om te kunnen zien welke functie een knop in de interface vervult. Hierbij vragen zij specifiek om tooltips met daarin een korte uitleg.
- Vakexperts zijn volledig gewend aan het werken in Word. Hierdoor ontstaat de behoefte om interacties, waar mogelijk, zo te laten werken als in Word. Zo wordt de leercurve voor het gebruik van FontoXML verkleind.
- Vakexperts hebben behoefte aan begeleiding in het structureren van hun content. Momenteel worden hier vaak Word-sjablonen voor gebruikt.

Invoerders

- Invoerders hebben de behoefte om (een preview van) de publicatievorm te kunnen zien. Dit is nodig om te controleren of de semantiek correct is toegepast en om te kijken of de visuele opmaak correct wordt getoond in het digitale systeem.
- Invoerders hebben behoefte aan begrijpelijke naamgeving in metadata en elementen
- Invoerders moeten op een gemakkelijke, snelle manier verschillende elementen kunnen verplaatsen.
- Invoerders hebben de behoefte om te kunnen controleren of de structuur van hun document klopt. Dit zouden zij bijvoorbeeld willen doen door te kijken naar de koppen binnen het document.
- Invoerders hebben behoefte om te kunnen zien of het document in zijn huidige staat is opgeslagen.

4. Ontwerprichtlijnen

In dit hoofdstuk staat een aantal ontwerprichtlijnen dat, bij het ontwerpen van de interfacepatronen in acht dient te worden genomen. Deze richtlijnen zijn gebaseerd op algemene Usability richtlijnen³, maar ook op verzamelde gegevens uit het gebruikersonderzoek en de FontoXML filosofie.

1. **Verplichte activiteiten moeten in isolement worden uitgevoerd**

Door de aandacht volledig op verplichte activiteiten te vestigen wanneer deze zich voordoen, wordt de aandacht van de gebruiker naar de juiste activiteit gebracht.

2. **Vermijd de mogelijkheid om acties uit te voeren die voor een invalide document zorgen**

Door de triggers voor deze acties waar mogelijk uit te schakelen. Zo informeer je de gebruiker vooraf dat iets niet mag, in plaats van hem eerst een fout te laten maken die hij vervolgens zelf moet onthouden.

3. **Bij het invoeren van informatie van een specifiek formaat, dient de interface de gebruiker te ondersteunen.**

Als het formaat van bepaalde invoer is vastgelegd, is er wellicht een interfacepatroon dat speciaal is ontwikkeld voor het invoeren van die informatie. Door dit te gebruiken t.o.v. vrije tekstinvoer, wordt de kans op foutieve invoer gereduceerd. Denk hierbij aan datums, URL's of afbeeldingen.

4. **Moedig gebruikers aan om iets meer dan het minimale te doen door dit gemakkelijk te maken.**

Door bepaalde acties, zoals het invullen van attributen, aan te moedigen, kan er eenvoudig waarde aan content worden toegevoegd. Gebruikers zullen vaak niet weten dat dit het geval is, maar staan waarschijnlijk welwillend tegenover het aanbod van een simpele, begrijpelijke verrijking.

5. **Verklaar onverwachte situaties en situaties die XML-kennis vereisen.**

Gebruikers weten niet wat XML inhoudt. Op het moment dat zij dus geen "person name" in een "surname" mogen plaatsen, is dit verwarrend. Als een element niet mag worden verwijderd omdat het verplicht is in zijn ouder, is dit verwarrend. Al deze situaties vereisen duidelijke feedback om frustratie bij de gebruiker weg te nemen. Door begrip voor de situatie te creëren, wordt deze frustratie tot een minimum gereduceerd. Als er

³ <http://www.nngroup.com/articles/ten-usability-heuristics/>

schemakennis nodig is voor het correct uitvoeren van een taak, dient deze dus op het juiste moment aan de gebruiker te worden aangeboden.

6. Gebruik bekende terminologie in plaats van xml-terminologie.

Probeer het vocabulaire dat in de applicatie wordt gebruikt zoveel mogelijk aan te laten sluiten op dat van de gebruiker. Neem dus niet blind alle termen uit de XML over. Pas in plaats daarvan synoniemen toe die voor de gebruiker meer logisch of begrijpelijk zijn (waar mogelijk). Dit kunnen termen zijn die zij kennen uit Word, maar ook termen uit de content waarmee vaak gewerkt wordt. In plaats van de gebruikers xml-terminologie aan te leren, gebruik je dus bekende terminologie en wordt de leercurve van de applicatie verkleind.

5. Benchmark

Uit de benchmark zijn een aantal mogelijke oplossingen voor de in ontwerprapport gedefinieerde designproblemen gekomen. Deze oplossingen zijn ook samengevat opgenomen in de systeemeisen. Ter verduidelijking van deze systeemeisen zijn de conclusies van de benchmark echter ook opgenomen in dit rapport.

Voor ieder van de behandelde designproblemen is in vier applicaties gekeken naar de toegepaste oplossing. Na elke oplossing beoordeeld te hebben met behulp van de Heuristics, kan er worden bepaald welke van deze oplossingen geschikt is om uit te werken en te testen voor gebruik binnen FontoXML.

Toegestane ouder-kind relaties

Hier is de oplossing van Oxygen het beste uit de vergelijking gekomen. Door de knoppen voor het aanmaken van elementen te laten staan, uit te schakelen en via tooltips feedback te geven, kan de gebruiker zien dat er iets aan de hand is en wat dat dan is. Deze oplossing kan aan de systeemeisen worden toegevoegd als optie.

Verplichte ouder-kind relaties

Oxygen en Easydita boden beiden dezelfde oplossing, waarbij verplichte kinderen automatisch worden geplaatst en vervolgens gevisualiseerd met behulp van een placeholder. Deze oplossing is efficiënt en duidelijke en kan dus worden meegenomen naar de systeemeisen.

Beperkt toegestane elementen

Hier scoort Easydita het best, door te voorkomen dat elementen te vaak worden toegevoegd. Het is echter niet duidelijk voor de gebruiker wat er aan de hand is en dus zal geen van deze opties worden meegenomen naar de systeemeisen.

Vaste volgorde van kinderen

Easydita voegt elementen automatisch in op de juiste positie. Dit is een lichte beperking in de gebruikersvrijheid, maar kan een hoop frustratie schelen. Deze oplossing wordt als systeemeis toegevoegd.

Elkaar uitsluitende kinderen

Hier boden EasyDita en Oxygen dezelfde oplossing. Deze geeft de gebruiker echter niet genoeg informatie. Het uitschakelen van de knop kan worden meegenomen, maar er zal een aanvullend ontwerp gemaakt moeten worden.

Vereiste element-attribuut relaties

Het automatisch toevoegen van vereiste attributen (zoals dat in EasyDita en Oxygen gebeurt) is toegevoegd als systeemeis omdat dit een efficiënte oplossing is waarbij de gebruiker onnodige interactie bespaard blijft.

Verplichte attribuutwaarden

Geen van de gevonden oplossingen in deze benchmark is geschikt voor FontoXML. In alle oplossingen wordt een situatie gecreëerd die weinig waarde aan het document toevoegt. Hier zal de gebruiker een waarde moeten invoeren, behalve als het systeem dit op een zinnige manier zou kunnen doen.

Toegestane attribuutwaarden

Beide XML-editors bieden een aangepaste interface voor het invoeren van attributen die aan een bepaalde waarde moeten voldoen. Dit zal worden meegenomen naar de systeemeisen.

Aangemoedigde elementcombinaties

Het toepassen van templates, zoals dat in EasyDita gebeurt, zal als oplossing voor dit probleem worden meegenomen. Het is een gemakkelijk, aanmoedigend patroon dat eenvoudig kan worden afgewezen door de gebruiker.

Aangemoedigde element-attribuutcombinaties

Hiervoor zijn geen oplossingen gevonden. Deze zullen dus zelf ontworpen moeten worden.

6. Systeemeisen

In dit hoofdstuk staan de systeemeisen beschreven die van invloed zijn op het aanmaken van elementen. Elk van deze systeemeisen dient in minimaal 1 ontwerp behandeld te worden. Mocht dit niet gebeuren, moet worden uitgelegd waarom aan deze systeemeis niet is voldaan.

Elementen aanmaken

“De gebruiker moet alle beschikbare elementen kunnen aanmaken in een document”

De gebruiker moet alleen elementen kunnen aanmaken op daarvoor toegestane locaties.	m
Het systeem moet voorkomen dat elementen worden ingevoegd op posities waar zij van het schema niet mogen bestaan door de trigger voor het aanmaken inactief te maken.	m
De gebruiker moet worden geïnformeerd waarom een element op een bepaalde locatie niet is toegestaan.	m
Het systeem moet, bij het aanmaken van een element, eventuele vereiste kinderen automatisch mee invoegen	m
De gebruiker moet, bij het automatisch invoegen van elementen, kunnen zien dat deze zijn ingevoegd en kunnen benoemen wat er gebeurd is.	m
Het systeem moet voorkomen dat een element dat in beperkte mate is toegestaan, vaker dan dat aantal kan worden ingevoegd.	m
Het systeem moet het aan de gebruiker duidelijk maken wanneer een kind-element in beperkte aantallen is toegestaan	m
Het systeem moet kind-elementen die in een vaste volgorde staan, automatisch in de juiste volgorde toevoegen.	m

Het systeem moet kinderen die op een vaste positie binnen een element moeten staan, automatisch op die positie invoegen. De trigger daarvoor moet, bij focus op de ouder, altijd beschikbaar zijn.	m
De gebruiker moet kunnen zien waarom een element op een andere plek is ingevoegd dan op de cursorpositie.	m
Het systeem moet het aan de gebruiker kenbaar maken wanneer bepaalde elementen elkaar uitsluiten in een ouder.	m
Het systeem moet, bij het aanmaken van elementen, eventuele vereiste attributen automatisch toevoegen.	m
Het systeem moet de gebruiker dwingen om verplichte attribuutwaarden in te voeren voordat een element wordt ingevoegd	m
Het systeem moet zorgen dat een attribuutwaarde (indien ingevuld) voldoet aan eventuele regels die daarop gelden door een interface te bieden die invoer binnen die regels aanmoedigt of verplicht.	m
Het systeem moet bij het aanmaken van elementen om kunnen gaan met aangemoedigde elementcombinaties	m
Het systeem moet voor aangemoedigde elementen een placeholder invoegen die pas XML toevoegt wanneer deze daadwerkelijk inhoud bevat.	m
Het systeem moet bij het aanmaken van elementen om kunnen gaan met aangemoedigde element-attribuutcombinaties	m
Het systeem moet, indien dat qua beschikbare ruimte in het scherm mogelijk is, het invoeren van aanbevolen elementen aanbieden via een popover.	m

7. Interactieontwerp per designprobleem

In dit hoofdstuk staan alle ontwerpproblemen en de daarvoor ontworpen oplossingen beschreven. Elke paragraaf behandelt een specifiek probleem. Per probleem zijn er twee, soms drie oplossingen bedacht, die per stuk worden verklaard in een sub paragraaf. Per oplossing is er een korte beschrijving van het ontwerp, een aantal voordelen, een aantal nadelen en overige aandachtspunten te vinden. Ook is de daadwerkelijke schets van het ontwerp bijgevoegd. Hierbij is gekozen voor schetsen t.o.v. van mockups omdat schetsen snel kunnen worden gemaakt, maar toch veel uitleg over zowel interactiviteit als lay-out kunnen uitbeelden. Het is dus een geschikte manier om snel verschillende concepten uit te werken, communiceren en bespreken. Aangezien dat precies het doel van deze ontwerp opdracht is, is deze techniek dus zeer geschikt.

Om te zorgen dat iedere schets representatief is, is er gekeken naar de voorbeelden van use cases per designuitdaging in mijn onderzoek naar xml-bewerking. Om te zorgen dat de schetsen consistent en leesbaar zijn, hanteerde ik bij het maken hiervan de volgende stijlrichtlijnen:

1. *Commentaren in rood*. Op deze manier is snel te zien wat er bij het ontwerp hoort en wat er commentaar is.
2. *Kleur voor nadruk, niet voor vormgeving*. Kleur kan worden toegepast om bepaalde onderdelen te benadrukken. Vormgeving wordt in deze schetsen verder buiten beschouwing gelaten
3. *Gebruik het juiste niveau van detail*: ga geen heel scherm schetsen als je slechts een knop in de toolbar wilt uitdrukken. Door op het juiste niveau te werken wordt de aandacht naar de juiste punten gebracht.
4. *Laat duidelijk de stappen van een interactie zien*: Als ontwerper heb je heel duidelijk in je hoofd hoe iets moet werken. Het doel van de schetsen is echter om dit ook aan anderen kenbaar te maken. Vergeet dus geen stappen te beschrijven, het is vervelend als een ander die moet gaan raden.
5. *Als je iets niet kunt schetsen, beschrijf het dan in commentaar*: Het naar rechts sliden van een paneel is moeilijk te schetsen, of je moet een flipbook maken. Als het niet lukt om iets te schetsen, beschrijf het dan tekstueel.

Omdat de focus van deze opdracht zeer sterk ligt op interactieontwerp en slechts zeer minimaal op visueel ontwerp, zullen er geen mockups gemaakt worden. De ontwerpen gaan dus direct van papier naar de browser. Vanwege de reeds bepaalde huisstijl en visuele richtlijnen binnen de interface van FontoXML kost deze stap slechts minimaal denkwerk. Zo wordt er kostbare tijd bespaard op het werken in Photoshop, die nu kan worden gebruikt om een zo representatief mogelijk, interactief, testbaar prototype neer te zetten.

7.1. Toegestane ouder-kind relaties

Gebruikersquote:

“Waarom kan ik hier geen titel invoegen?”

Binnen XML-schema's bestaan restricties op de mogelijke ouder-kind relaties van elementen. Het is niet zo dat alle elementen alle elementen mogen bevatten en dus ook niet zo dat alle elementen in alle elementen mogen bestaan. Je kunt niet van mensen verwachten dat zij al deze mogelijke combinaties uit hun hoofd weten en al helemaal niet van mensen zonder kennis van XML.

Het is dus belangrijk om aan de gebruiker duidelijk te maken welke relaties er toegestaan en dus mogelijk zijn binnen FontoXML. Het moet in een oogopslag helder zijn of je een bepaalde combinatie kunt gebruiken. Indien een bepaalde combinatie niet mogelijk is, moet gebruiker zo duidelijk mogelijke feedback krijgen over de reden daarvan. Deze feedback moet de gebruiker helpen, maar niet storen in zijn workflow en dus niet opdringerig zijn.

Ontwerp: uitschakelen van knoppen

Zie afbeelding 7.1 voor de interactieschets bij dit ontwerp.

Door de knop voor het aanmaken van elementen die niet zijn toegestaan uit te schakelen, voorkom je dat mensen een schema-invaliditeit creëren. Door feedback te leveren over de reden dat de knop is uitgeschakeld via een tooltip, leer je mensen een beetje over de werking van XML.

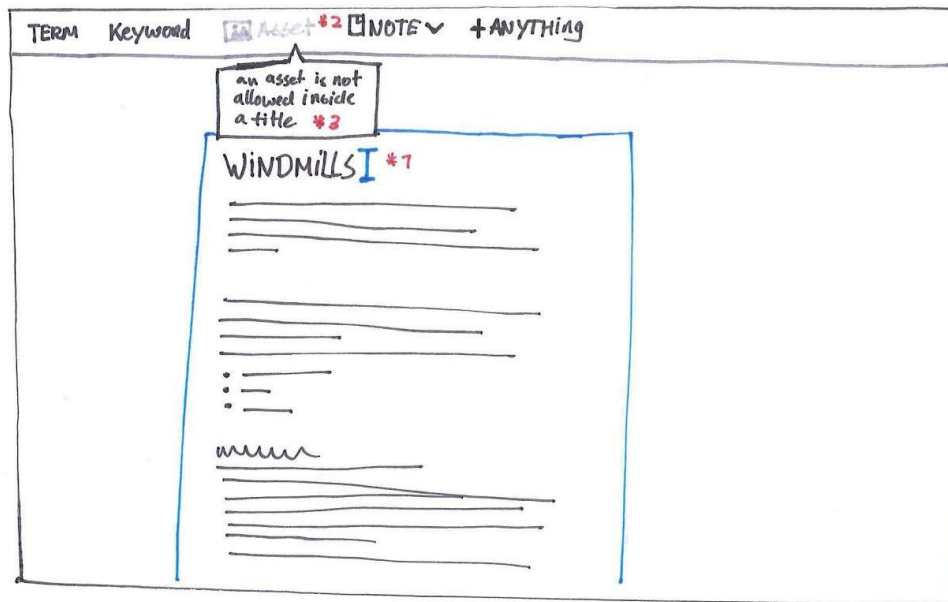
Voordelen

- Maakt toevoegen van “verboden” elementen praktisch onmogelijk.
- Door visueel onderscheid is een uitgeschakelde knop snel te herkennen.

Nadelen

- Communiceert, door de vertraagde tooltip, niet in één oogopslag de reden dat een knop is uitgeschakeld. Kan dus vragen oproepen bij de gebruiker.

Element aanmaken - toegestane locaties



- *1. The cursor staat in een <Title>.
- *2. Een <asset> is niet toegestaan in een <title>. De toolbar-knop is daarom uitgeschakeld.
- *3. Als je met de cursor op een uitgeschakelde knop gaat hangen (:hover) verschijnt een tooltip met daarin de reden dat een knop is uitgeschakeld.

afbeelding 7.1: Omgaan met toegestane ouder-kind relaties bij het aanmaken van elementen.

7.2. Vereiste ouder-kind relaties

Gebruikersquote:

“Waarom verschijnen er meerdere nieuwe elementen terwijl ik maar één element probeerde in te voegen?”

Zoals er verboden ouder-kind relaties zijn in XML, zijn er ook verplichte ouder-kind relaties. Dit houdt in dat een element bepaalde kinderen *moet* bevatten. Er moet dus in FontoXML worden gezorgd dat elementen te allen tijde hun verplichte kinderen bevatten. Zo wordt de schemavalide situatie bewaakt en blijft het document waardevol. Ook hier moet de informatie over welke verplichte kinderen een element heeft niet in hoofd van de gebruiker zijn, maar in de applicatie.

De gebruiker heeft, bij het invoegen van een element, niet per se de intentie om meerdere elementen in te voegen. Omdat dit verplicht is door het schema zal dit echter toch gebeuren. Het is daarom belangrijk dat de gebruiker 1: begrijpt *dat* er meerdere elementen zijn ingevoerd en 2: begrijpt *waarom* er meerdere elementen zijn ingevoerd. Verder is het wenselijk dat de gebruiker deze kindelementen ook, zonder dwang, content geeft.

7.2.1. Waarde van verplichte kinderen in modal invoeren

Zie afbeelding 7.2-a voor de interactieschets bij dit ontwerp.

Door een modal te openen bij het invoegen van elementen met verplichte kinderen, wordt de aandacht even volledig verlegd naar die taak. In de modal kan uitleg worden gegeven over het feit dat deze kinderen verplicht zijn, zodat de gebruiker dat hier leert over de situatie.

Voordelen

- Een modal is een goede manier om gebruikers te sturen in hun handelingen. Door velden voor verplichte elementen te tonen in de modal wordt de kans op lege, betekenisloze elementen verkleind.
- Mocht de klant besluiten dat verplichte elementen ook verplicht inhoud moeten bevatten, is de modal een goede optie.

Nadelen

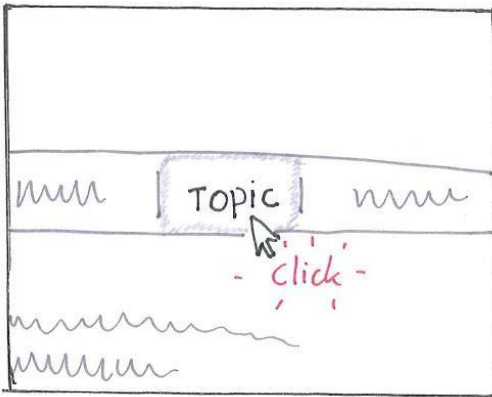
- De modal is een opdringerig interfacepatroon. Het onderbreekt de workflow van de gebruiker en verstoort het immersief bewerken van een document.

- De modal kan het toevoegen van een element onnodig meer complex laten lijken dan het daadwerkelijk is.
- Er kan verwarring ontstaan bij het bewerken. Eerst bewerk je iets in een modal, dan opeens in het document. Is dat dan wel hetzelfde ding? En waarom kan ik die modal niet meer terugkrijgen?

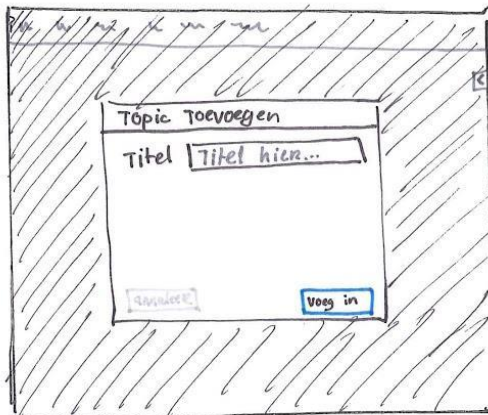
Overige aandachtspunten

- Het is niet *verplicht* om de velden binnen deze modal in te voegen. Wanneer de gebruiker zonder iets in te vullen op “insert” drukt, zal het element alsnog worden ingevoegd, maar dan met placeholders.

Element aanmaken - Vereiste ouder-kind relaties (1.)



1. Trigger het aanmaken van een element met verplichte kinderen, bijvoorbeeld topic.



2. Titel is verplicht. Invullen van titel is dus sterk gewenst. Modal verschijnt om gebruiker te laten focussen op invullen van de titel.

NOTE: Invullen van de titel is niet schemaverplicht



3. Na klikken op "voeg in" (2.) wordt het nieuwe Topic geplaatst, met daarin de titel zoals ingevuld in de modal.

afbeelding 7.2-a: Aanmoedigen door de waarde van vereiste kinderen in te voeren via een modal.

7.2.2. Element automatisch invoegen met placeholder

Zie afbeelding 7.2-b voor de interactieschets bij dit ontwerp.

Alle verplichten kinderen van een element worden automatisch, met de ouder samen, in de content geplaatst. Ook in de XML worden deze dus al toegevoegd. Tussendoor komt er geen melding dat er meerdere elementen worden ingevoegd. De gebruiker dient dit op te merken door het plaatsen van placeholders voor alle nieuwe elementen. Dit betekent dus dat alle elementen met verplichte kinderen automatisch een template zijn: het element wordt samen met zijn verplichte kind(eren) ingevoegd.

Voordelen

- De gebruiker wordt niet verstoord in het schrijven: nieuwe elementen worden direct in het document geplaatst.

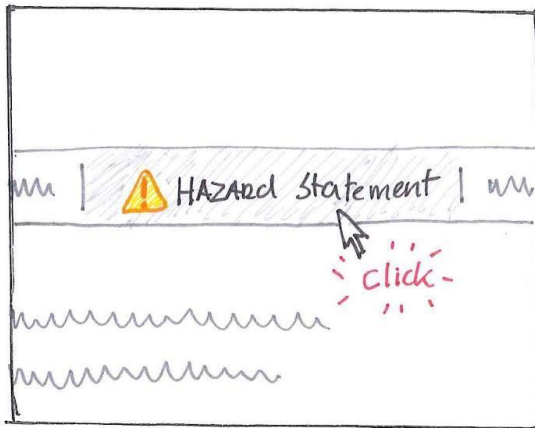
Nadelen

- Het invullen van lege, verplichten velden wordt minder aangemoedigd. Hierdoor is de kans op lege elementen (met dus weinig waarde) in de content groter.

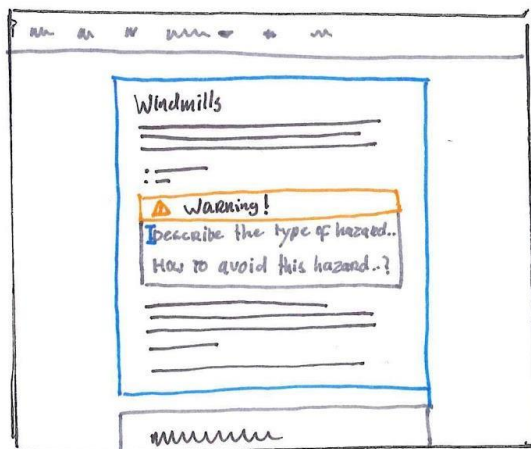
Overige aandachtspunten

- Je zou kunnen zeggen dat deze elementen zich als een softform zouden moeten gedragen. Echter, omdat deze elementen verplicht zijn, is ervoor gekozen deze altijd te tonen zodat de kans van opmerken groter is.

Element aanmaken - Vereiste ouder-kind relaties (2.)



1. Trigger het aanmaken van een element met verplichte kinderen, bijvoorbeeld Hazard statement.



2. In plaats van een modal te tonen, wordt de content direct geplaatst. verplichte velden krijgen een placeholder, die direct focus heeft.

De verplichte velden zijn dus niet ingevuld, maar de eerste heeft wel focus om dit aan te moedigen.

afbeelding 7.2-b: De verplichte kinderen "typeofhazard" en "howtoavoid" worden automatisch toegevoegd en tonen een placeholder ter indicatie daarvan.

7.3. Beperkt toegestane kinderen

Gebruikersquote:

“Waarom mocht ik net wel een result aan deze task toevoegen en nu niet meer?”

Elementen mogen van bepaalde kinderen soms slechts een beperkt aantal instanties bevatten. In de regel kan dat ieder aantal zijn, maar het gaat eigenlijk altijd om een enkele instantie. Om het document schemavalide te houden, moet dus worden voorkomen dat dit aantal instanties wordt overschreden (probleem 1). Probleem 2: de kans is zeer groot dat de gebruiker niet weet dat deze beperking bestaat.

Soms is beperking in aantal logisch en zal de gebruiker nooit proberen om een element meer dan één keer in te voegen: bij een titel zou bijvoorbeeld zo kunnen zijn. Het is echter ook mogelijk dat het toevoegen van meerdere instanties wél logisch lijkt. Daarom moet het altijd duidelijk aan de gebruiker worden gecommuniceerd wanneer deze regel van toepassing is, zodat verwarring en frustratie voorkomen kan worden.

7.3.1. Trigger uitschakelen na bereiken van maximum

Zie afbeelding 7.3-a voor de interactieschets bij dit ontwerp.

Het eerste ontwerp lost dit probleem op door de trigger uit te schakelen wanneer het maximum toegestane instanties van een bepaald element is bereikt. De trigger kan dan niet meer worden gebruikt totdat er een andere instantie van het element is verwijderd. De gebruiker kan echter niet aan de knop zien *waarom* deze is uitgeschakeld: de styling van een uitgeschakelde knop is altijd hetzelfde. Via een tooltip kan de reden dat deze knop is uitgeschakeld naar de gebruiker worden gecommuniceerd.

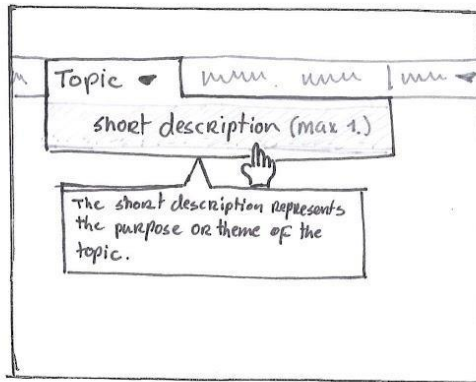
Voordelen

- Toepasbaar op verschillende schemaregels. Daardoor consistentie: knop uitgeschakeld bij verboden, uitleg in tooltip.
- Vinkje geeft aan wanneer het element al bestaat
- (Max 1.) geeft directe feedback over toegestane hoeveelheid instanties.

Nadelen

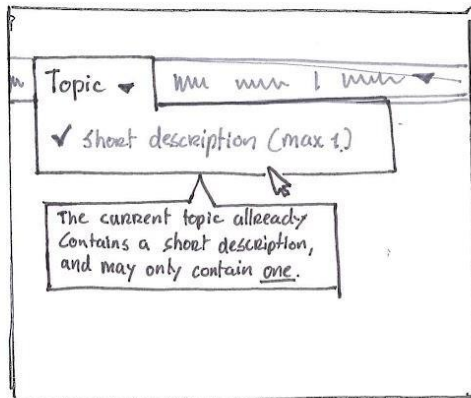
Neemt positie van icoon binnen een knop in met ander doel. Dit icoon geeft status aan in plaats van elementbetekenis, en dat is niet gebruikelijk.

Element aanmaken, beperkt toegestane kinderen (1)



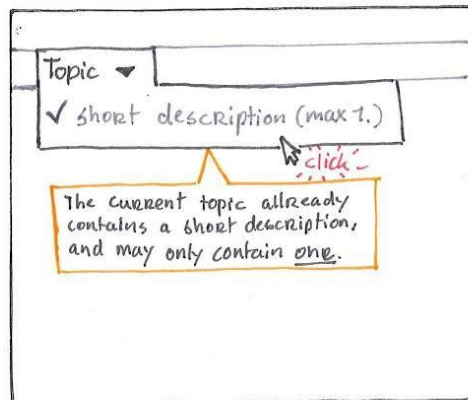
Situatie A: `<Topic>` bevat nog geen `<shortdesc>`.

- Trigger is actief: te zien aan tekstkleur, cursor en achtergrond op :Hover.
- Klikken voegt shortdesc toe aan huidige topic.
- Tooltip geeft korte verklaring van het element
- (max 1.) impliceert het aantal toegestane instanties.



Situatie B: `<Topic>` bevat al een `<shortdesc>`.

- Trigger is inactief: grijze tekst, geen achtergrondkleur en cursor.
- Tooltip geeft uitleg over de reden dat de knop inactief is.
- Klikken opent en vestigt aandacht op de tooltip. (situatie c)
- Vinkje geeft aan dat het element al bestaat.



Situatie C: gebruiker klikt toch op inactieve knop.

- menu klappt niet dicht.
- Tooltip verkleurd om aandacht op zich te vestigen.

afbeelding 7.3-a: De knop voor "short description" wordt uitgeschakeld wanneer er al een instantie van het element bestaat in de huidige topic.

7.3.2. Functie van trigger veranderen

Zie afbeelding 7.3-b voor de interactieschets bij dit ontwerp.

De elementen uit dit ontwerpprobleem kun je slechts enkelvoudig in een bepaalde ouder toevoegen. Als deze in de ouder bestaat, is de knop om hem aan te maken dus waardeloos. In plaats van deze uit te schakelen, zou je ervoor kunnen kiezen om deze knop nu het element te laten verwijderen. Zo heeft de knop een andere, maar meer bruikbare functie. De kans bestaat immers dat je een element weer wil verwijderen en voor eenmalig toegestane elementen zou dit een logische plek zijn voor deze actie.

Dit ontwerp kent een variant met iconen en een variant met tekst. Van de tekstuele variant is het nadeel dat deze bredere knoppen en meer leeswerk oplevert. Bij de variant met iconen is het de vraag of de gebruiker altijd direct begrijpt wat de iconen betekenen.

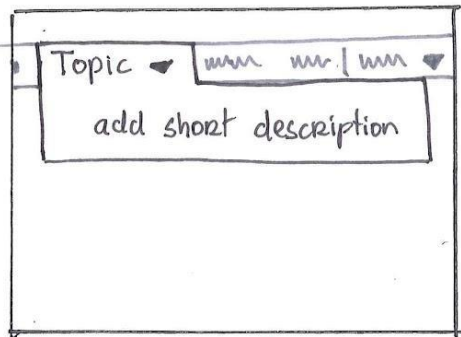
Voordelen

- De knop heeft altijd een functie, ook al mogen er geen elementen meer aangemaakt worden.

Nadelen

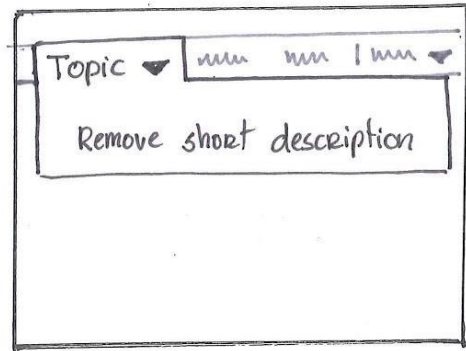
1. Kans op fouten groter door inconsequentie in affordance: soms verwijdert de knop iets, soms voegt hij iets toe.

Element aanmaken - beperkt toegestane kinderen (2)



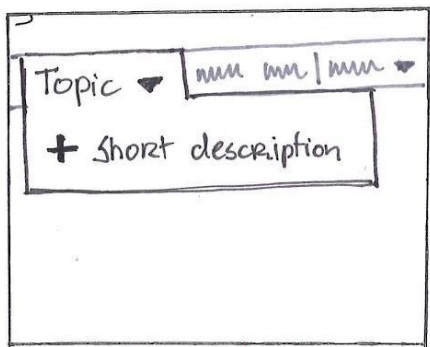
Situatie A: <Topic> bevat nog geen <shortdesc>.

- Trigger voegt shortdesc toe bij klik op de knop

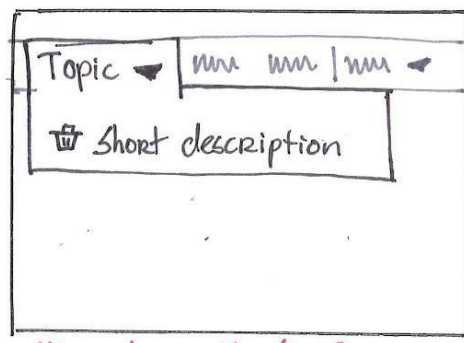


Situatie B: <Topic> bevat al een <shortdesc>.

- ~~Affordance~~ van functie verandert. Knop verwijderet nu de shortdesc.
- Kan verwarrend zijn omdat gedrag niet consistent is. (van de knop)



Alternatief situatie A: Iconen ipv. tekstueel.



Alternatief situatie B: Iconen ipv. tekstueel.

afbeelding 7.3-b: De knop voor "short description" verandert van functie, afhankelijk of deze wel of niet bestaat binnen het huidige topic.

7.4. Vaste volgorde van kinderen

Gebruikersquote:

“Waarom kan ik niet gewoon een voorbeeld aan deze task toevoegen?”

Soms moeten de kinderen van een element in een bepaalde volgorde staan. Omdat triggers (in ontwerpprobleem 1) zijn uitgeschakeld op posities waar het element niet mag worden toegevoegd, moet de gebruikers in dit geval dus exact de goede positie aanwijzen om zo’n kind in te voegen. Hierbij doen zich twee problemen voor: ten eerste kan het lastig zijn om de exacte cursorpositie te bereiken waar het element is toegestaan. Ten tweede is het zeer waarschijnlijk dat de gebruiker de exacte volgorde niet uit zijn hoofd weet, waardoor hij überhaupt niet weet waar hij moet zoeken naar de juiste cursorpositie.

Bij dit probleem is het van belang om vergevingsgezind te zijn naar de gebruiker. Help de gebruiker met het vinden van de juiste locatie, of corrigeer de gebruiker bij foutjes. Daarnaast is het ook hier belangrijk om te situatie te verklaren: vertel de gebruiker dat er een vaste volgorde van toepassing is op bepaalde elementen en wellicht zelfs welke volgorde dit dan is. Wees echter niet te opdringerig. Toelichtingen moeten de gebruiker helpen, niet in de weg zitten.

7.4.1. Automatisch juiste positie toewijzen

Zie afbeelding 7.4-a voor de interactieschets bij dit ontwerp.

Ontwerp A beschrijft een aanpak waarbij FontoXML automatisch de juiste positie aan een element toewijst. In dit voorbeeld is de knop voor het aanbrengen van <context> altijd actief, zolang de cursor focus op een <task> heeft. Bij het klikken op deze knop, wordt <context> automatisch op zijn toegestane positie ingevoegd.

Voordelen

- Vergevingsgezind: de auteur hoeft de locatie niet uit zijn hoofd te weten omdat het systeem deze automatisch kiest.
- Speelt beter in op de gebruikersintentie dan wanneer de knop helemaal is uitgeschakeld.

Nadelen

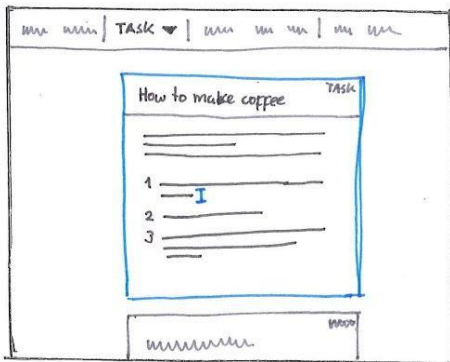
Geen informatie vooraf over vaste volgorde en al helemaal niet over *welke* volgorde dan.

Voert niet precies de actie uit die de gebruiker eigenlijk probeert.

Overige aandachtspunten

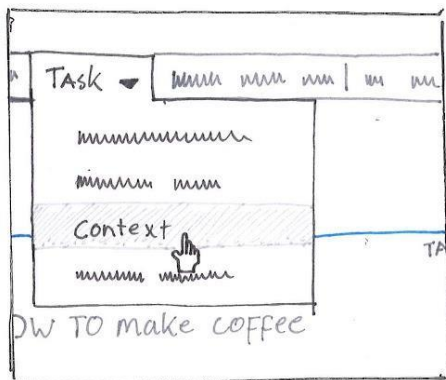
- Moeilijk te schalen: voor welke elementen doe je dit wel en voor welke niet? Dit heeft nog denkwerk nodig.
- Toon je de message voor eeuwig? of voor 1 of 2 keer? Of laat je de gebruiker hem “dismissen?”.

Vaste element volgorde - aanmaken (1)



Situatie A: Cursor staat in <STEP>, waar <context> niet is toegestaan.

- Gebruiker klikt op **TASK ▼**



Situatie B: Ondanks het feit dat <context> niet is toegestaan op de cursorpositie, is de knop toch actief.

- Als de gebruiker op context klikt, zal deze op de meest nabije toegestane locatie worden ingevoegd.



Situatie C: context is tussen shortdesc en steps geplaatst binnen de huidige task.

Het systeem geeft een bericht ter uitleg.

*1: Feedback message.

"context" has a fixed location within a task. Therefore, it has been placed on a different place than the cursor position. undo

afbeelding 7.3-b: Elementen met een vaste volgorde worden automatisch op de juiste plek ingevoegd.

7.4.2. Paneel met structuursuggesties

Zie afbeelding 7.4-b en afbeelding 7.4-c voor de interactieschetsen bij dit ontwerp.

Voor bepaalde (of alle?) elementen is de “structure guide” op te roepen in het rechter zijpaneel. Hierin staat een overzicht voor de mogelijke elementen die je binnen dat element zou kunnen plaatsen om hem verder te verrijken. Daarnaast kun je vanuit het paneel de verschillende gesuggereerde elementen ook daadwerkelijk invoegen op de juiste positie. Ook staat er een korte uitleg over de betekenis en het gebruik van het element zelf.

Voordelen

- Veel ruimte voor uitleg over de volgorde en andere achtergrondinformatie
- Ook toepasbare voor andere uitleg-doeleinden, niet alleen voor de vaste volgorde

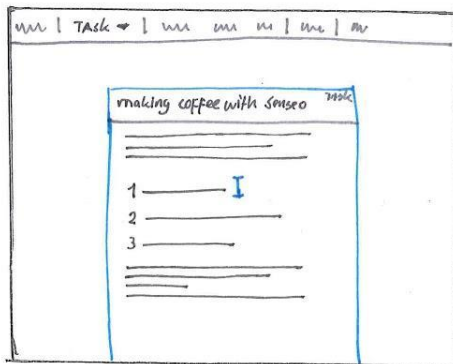
Nadelen

- Het paneel moet eerst worden geopend voordat de gebruiker iets leert over de vaste volgorde van de kind-elementen. Het loopt dus niet heel soepel mee in de aanmaak-flow.

Overige aandachtspunten

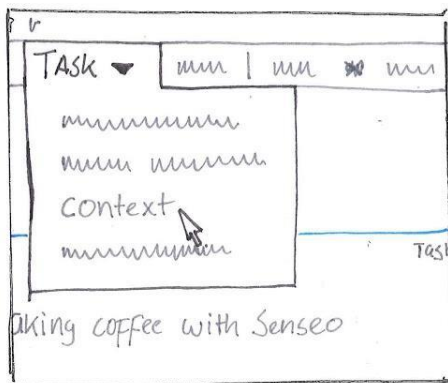
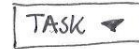
- Blijf goed nadenken over wat er wel en niet in het zijpaneel moet. Het moet niet overvol raken.

Vaste elementvolgorde - (aanmaken 2A)



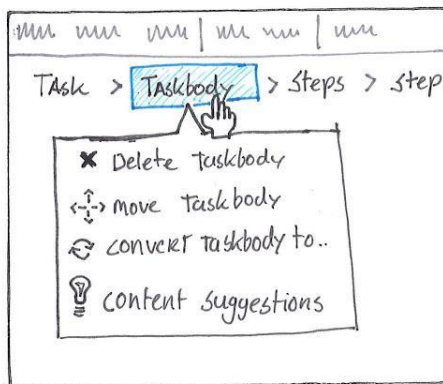
Situatie A: Cursor staat in <step>, waar <context> niet is toegestaan.

- Gebruiker klikt op

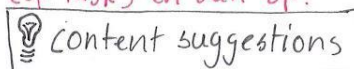


Situatie B: De knop voor het aanmaken van <context> is uitgeschakeld, vanwege een niet-toegestane positie.

- Gebruiker denkt "waarom mag dit niet?" en gaat op zoek naar de reden



Situatie C: De gebruiker klikt in de breadcrumbs op Taskbody (of task) en dan op.



Situatie D: zie 2B



afbeelding 7.4-a: de gebruiker roept via de breadcrumbs de "content suggestions" voor taskbody op.

Vaste elementvolgorde - aanmaken (2B)

Hand-drawn sketch of a user interface for "Making coffee with Senseo". The interface is divided into two main panels.

The left panel, titled "Making coffee with Senseo" and labeled "TASK", contains a list of three numbered items (1, 2, 3) followed by three lines of wavy text.

The right panel, titled "Content suggestions", contains a "Task." section with three lines of wavy text, followed by a "Suggested content:" section. This section lists five items: "Short description", "Preliminary Requirements", "Context", "Steps", and "Result", each followed by a circular icon. The "Steps" item is marked with a checkmark and a plus sign. The "Example" item is also listed.

Red annotations are present: *1 next to the "Task." section, *2 next to the "Suggested content:" section, and *3 next to the "Steps" item.

Situatie D: Na het openen van de content suggestions toont het zijpaneel zich met info over <Task>.

*1: In een korte tekst kan worden uitgelegd dat we met een vaste volgorde te maken hebben.

*2: Per element biedt een ⊕-knop de mogelijkheid om dit element automatisch op de goede positie in te voegen.

*3: Steps bestaat al, en is dus ook daar inactief

afbeelding 7.4-b: het zijpaneel met content suggestions, waarin uitleg en voorbeelden van elementen staan.

7.5. Elkaar uitsluitende elementen

Gebruikersquote:

“Waarom kon ik net wel een abstract aanmaken en nu niet meer?”

Het feit dat bepaalde elementen elkaar uit kunnen sluiten in een schema, levert een tweetal designuitdagingen op bij het aanmaken van deze elementen. Ten eerste moet het document altijd schemavalide blijven. Daarnaast weet de gebruiker niet dat bepaalde elementen elkaar uitsluiten. Laat staan dat zij weten welke elementen dan precies. De situatie die hier beschreven wordt is die waar er 0 of 1 van de X elementen mogen bestaan. In het schema zou dit worden uitgedrukt als (<> or <> or <>...) (*optional*). Wanneer er meerdere van deze elementen mogen bestaan, is dit ontwerp niet van toepassing.

Er moet dus voorkomen worden dat er meerdere van deze elementen tegelijkertijd in hun ouder bestaan. Verder moet er aan de gebruiker worden uitgelegd *dat* en misschien zelfs *waarom* deze elementen niet naast elkaar mogen bestaan, zodat er begrip voor de situatie ontstaat.

7.5.1. Trigger uitschakelen

Zie afbeelding 7.5-a en afbeelding 7.5-b voor de interactieschetsen bij dit ontwerp.

In dit ontwerp is ervoor gekozen om de knoppen voor alle kinderen inactief te maken wanneer er één van hen bestaat. Op deze manier wordt voorkomen dat er een schema-invalid schema kan ontstaan. Om de gebruiker toch tegemoet te komen in waarschijnlijke acties, wordt er vanuit een submenu de actie “omzetten naar alternatief” aangeboden. Zo kan de gebruiker zich, bij het invoegen van een verkeerd alternatief, toch makkelijk herstellen. Daarnaast geeft dit ook een hint naar het bestaande verband tussen deze elementen. Deze hint wordt verder verklaard door met de cursor te hoveren op een van de inactieve knoppen, waarna een tooltip met een situatie uitleg verschijnt. Als laatste mogelijkheid kan er ook in de structuurgids nog uitleg worden aangeboden.

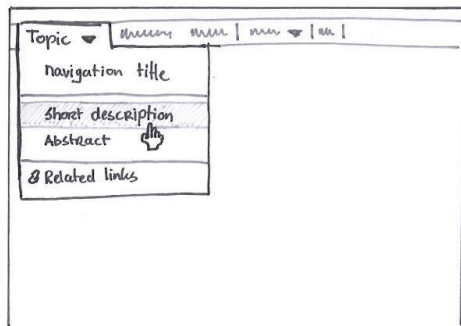
Voordelen

- Goed mogelijk in combinatie met bestaande knoppenindeling.
- Scheidt duidelijk de acties “aankomen” en “omzetten” voor de gebruiker.

Nadelen

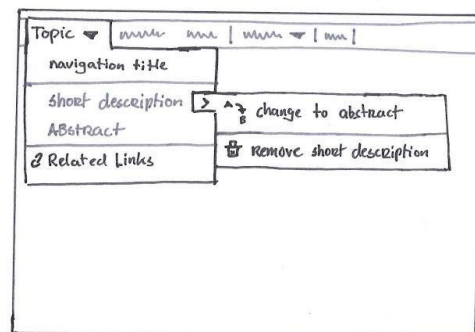
- Het “elkaar uitsluiten” is niet in één oogopslag duidelijk voor de gebruiker bij het aanmaken.

Elkaar uitsluitende elementen - aanmaken (1A)



A: Topic heeft geen <Abstract> of <Shortdesc>.

- Beide knoppen actief
- Cursor ergens in topic
- Gebruiker hooft op knop om <Shortdesc> aan te maken, klikt daarna.



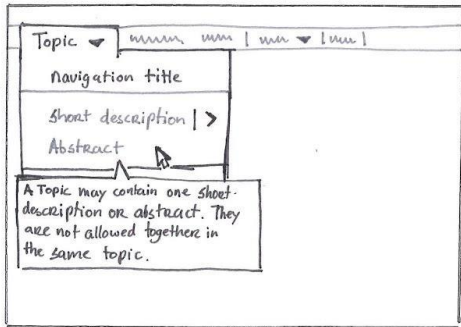
B: Topic bevat nu een <Shortdesc>.

- Beide knoppen uitgeschakeld.
- Vanwege MUX-heid wordt transformeer-actie aangeboden (droprecht)
- Content wordt daarbij bewaard.
- MUX-heid wordt uitgelegd in paragrafen. zie C & D.
- Cursor in short description.

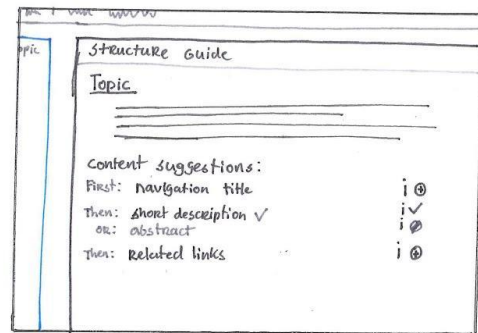
? Change? Transform? Convert? Switch?

afbeelding 7.5-a: Voor elkaar uitsluitende elementen wordt een alternatieve actie (omzetten) aangeboden.

Elkaar uitsluitende elementen - aanmaken (1B)



- C: Bij hover op de uitgeschakelde knop, verschijnt een tooltip met een verklaring over de reden dat de knop is uitgeschakeld.
- Wellicht ook nog in de knop zeggen dat dit topic al een shortdesc bevat.



D: Uitleg in de structuurgids.

- In de contentsuggesties is te zien dat `<shortdesc>` en `<abstract>` elkaar uitsluiten.
- Ook is te zien welke elementen er al zijn.
- Kan alleen als gids er überhaupt komt.

? Hoe kom je van je vraag naar de gids.

afbeelding 7.5-b: Via een tooltip en de structuurgids wordt uitgelegd waarom knoppen zijn uitgeschakeld.

7.5.2. Radiobuttons

Zie afbeelding 7.5-c voor de interactieschetsen bij dit ontwerp.

In ontwerp B wordt voor elementen die elkaar uitsluiten een knop met een radiobutton getoond in plaats van een “normale” knop. Op deze manier wordt direct aan de gebruiker duidelijk gemaakt dat hij een keuze heeft tussen één van de knoppen met een radio button (Nielsen, 2004). Deze knoppen zijn ook nog eens zijn gegroepeerd binnen de toolbar- of dropdown, om duidelijk te maken dat zij bij elkaar horen.

Wanneer er een van de opties is geselecteerd, zal het klikken op een van de andere opties altijd een poging tot omzetten doen. Het voordeel hiervan is dat de knoppen altijd nut blijven hebben. Het nadeel is dat er rekening gehouden dient te worden met de toegestane kinderen van dit element. Dit wordt opgelost door, indien er een schema-invaliditeit dreigt te ontstaan, een modaal te tonen waarin de benodigde conversies aan de gebruiker worden aangeboden.

Voordelen

- Het “elkaar uitsluiten” - en welke elementen dat precies doen is in één oogopslag duidelijk voor de gebruiker bij het aanmaken.

Nadelen

- De radiobuttons veroorzaken visuele onrust in combinatie met de mogelijke iconen in een knop.
- Het “aanmaken” van een element door voor het eerst op een radio button te klikken voelt vreemd.

Overige aandachtspunten

- Wellicht is de transformatie niet helder voor de gebruiker, maar zolang de content in beide elementen van de transformatie is toegestaan, is het patroon vergevingsgezind. De radio buttons zijn zeer geschikt voor het wisselen tussen elementen.

Elkaar uitsluitende elementen - Aanmaken (2)

A: Topic heeft geen `<abstract>` of `<shortdesc>`.

- Gebruiker klikt op `shortdesc` om een short description aan te maken.
- Geen van de Radio buttons is nog geselecteerd.

B: Topic heeft nu een `<shortdesc>`

- Gebruiker klikt op `Abstract` om de shortdesc om te zetten naar een abstract.
- Indien alle content van element A - de shortdesc is toegestaan in element B - de abstract, gaat het omzetten zonder tussenstep.
- Anders, zie C.

C: Fontoxml biedt in een modal de kans om nodige conversies toe te staan of af te wijzen.

- Hiervoor is een conversiemodel nodig.
- Wellicht: eerst terug in specialisatie-root, anders naar een paragraaf.

afbeelding 7.5-c: Door radio buttons te gebruiken is het direct helder dat bepaalde elementen elkaar uitsluiten

7.6. Vereiste element-attribootrelaties

In schema's kan worden vastgelegd dat een element bepaalde attributen moet bevatten om valide te zijn. In dit geval mag het element dus niet in FontoXML bestaan zonder dat het deze attributen bevat. Het is niet te verwachten dat de gebruiker op de hoogte is van deze relatie en dat zou ook niet nodig moeten zijn. Deze regel is via het schema uit te lezen en kan dus via de applicatie worden opgelost.

De oplossing voor dit probleem heeft op zichzelf naar mijn idee geen invloed op de interactie van de gebruiker met het systeem. Het is voldoende om vereiste attributen, zolang zij niet verplicht een waarde moeten bevatten, automatisch toe te voegen aan een element. Op deze manier wordt de schemavalide situatie bewaakt en kan de gebruiker doorgaan zonder van deze regel af te weten. Dat is ook niet nodig, omdat er verder geen actie gevraagd wordt. Het is dus ook niet nodig om een UI-ontwerp te maken voor het omgaan met dit probleem: alles kan op de achtergrond worden opgelost, zonder dat de gebruiker dat weet.

Indien het hebben van een attribuutwaarde verplicht of aanbevolen is, heeft dat wel gevolgen voor de interface. Mogelijke oplossingen hiervoor zijn te lezen in respectievelijk paragraaf 7.7. en paragraaf 7.9.

7.7. Verplichte attribuutwaarden

Gebruikersquote:

“Waarom moet ik hier iets doen? En wat dan?”

Voor sommige attributen kan gezegd worden dat zij waardeloos⁴ zijn zonder ingevoerde attribuutwaarde. Dit kan in het schema worden vastgelegd door een waarde als verplicht op te stellen. Wanneer een attribuut aan een element wordt toegevoegd (bijvoorbeeld automatisch bij het invoegen van een element, zoals gesteld in paragraaf 7.6) moet hier dus een waarde worden ingevoerd. Voordat dit is gebeurd mogen het element en het attribuut niet in de XML worden toegevoegd, omdat dit een schema-INVALIDE situatie tot gevolg zou hebben.

Wanneer de hiervoor beschreven situatie het geval is, zal er dus een waarde moeten worden ingevoerd: ofwel door het systeem, ofwel door de gebruiker. Er zijn oplossingen denkbaar waarbij het systeem dit automatisch doet om een INVALIDE document te voorkomen. De vraag is echter of de waarde van het attribuut hiermee echt wordt verhoogd. Deze is immers verplicht gesteld omdat er met de waarde een bepaalde verrijking of betekenis dient te worden toegekend. Alleen wanneer het systeem zeer sterke logica kan toepassen bij het automatisch invullen van de waarde is dit gewenst. Dit bespaart de gebruiker namelijk werk. Echter, wanneer er vanuit het systeem geen betekenisvolle, correcte waarde kan worden toegekend, moet dit door de gebruiker worden gedaan. Het is hierbij van belang dat de gebruiker begrijpt dat voordat er een attribuutwaarde is ingevoerd, de element-invoegactie niet voltooid kan worden. Indien er sprake is van een element met *meerdere* attributen met verplichte waarden, dient dit altijd via een modal te worden gedaan. Het is namelijk onberekenbaar hoeveel schermruimte de gezamenlijke UI-patronen in beslag nemen.

7.7.1. Invoer via modal

Zie afbeelding 7.7-a voor de interactieschetsen bij dit ontwerp.

“UI Pattern: Modal Panel. Use when: The app or site has gotten into a state from which it shouldn't or can't proceed without input from the user.” - (Tidwell, 2010). In dit ontwerp heb ik dan ook een modal panel (kortweg: modal) toegepast voor het invoeren van de benodigde attribuutwaarde. Om de focus op de modal te houden gebruik ik deze in combinatie met een lightbox-effect. Zo begrijpt te

⁴ <http://nl.wikipedia.org/wiki/Woordspeling>

de gebruiker dat er actie met de modal moet worden ondernomen. De modal toont ter verduidelijking van de benodigde actie een actietitel, human-friendly element- en attribuutnamen en waar nodig een uitleg over de actie. Daarnaast geeft de modal feedback wanneer de gebruiker probeert om het attribuut alsnog met een lege waarde in te voeren.

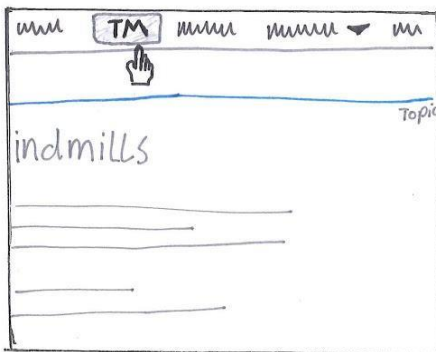
Voordelen

- Herkenbaar, vertrouwd patroon dat al vaker wordt gebruikt in de applicatie en op veel andere plekken op het web.
- Biedt veel ruimte voor eventueel benodigde UI-patronen voor invoer.

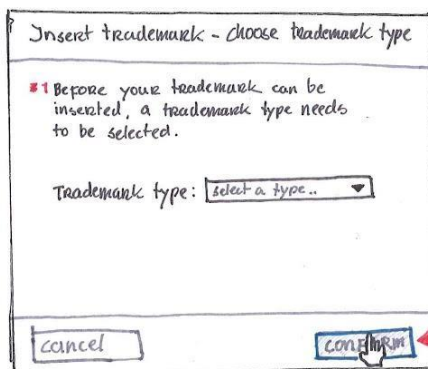
Nadelen

- Is van nature een opdringerig patroon. Dit is nodig, maar kan als vervelend worden ervaren door gebruikers (Mathis, 2014).

Element aanmaken - verplichte attribuutwaarde (1)



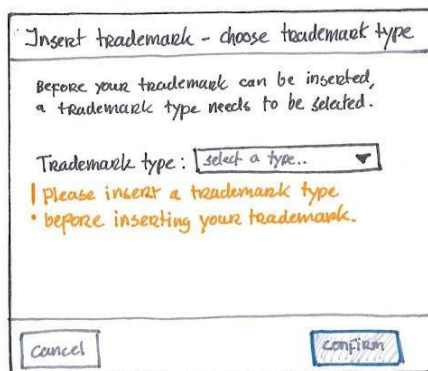
A: de gebruiker klikt op **TM** om een **<TM>** (trademark)-element aan te maken.



B: Omdat het attribuut **[TMtype]** wordt toegevoegd en een waarde moet bevatten, opent er een modal.

- Schets toont de inhoud van de modal.
- De titel bevat zowel de hoofdactie als de sub-actie.
- knop moet eigenlijk disabled zijn.

1: korte uitleg en instructie voor zowel begrip als sturing.
Gebruiker klikt zonder waarde in te voeren.



C: Na een poging tot foute invoer geeft het systeem terug dat er een fout is en hoe die kan worden opgelost.

afbeelding 7.7-a: De modal kan pas worden afgerond als het attribuut een waarde bevat.

7.7.2. Invoer verwerken in trigger

Zie afbeelding 7.7-b voor de interactieschetsen bij dit ontwerp.

Het tweede ontwerp voor verplichte attribuutwaarde verwerkt de invoer van de waarde in de trigger om het element aan te maken. Op deze manier kan de actie dus nog steeds niet worden uitgevoerd zonder dat er een waarde wordt ingevoerd. De gebruiker wordt hier echter niet lastiggevallen met een modal, zoals in ontwerp A. Deze oplossing is dus minder opdringerig. Er zijn verschillende varianten uitgewerkt in *afbeelding 7.7-b* waarin verschillende vormen van invoer worden behandeld. In variant A voelt deze erg logisch, maar in variant B en C is dit al wat minder. Ontwerp A heeft in dit geval dan ook de voorkeur.

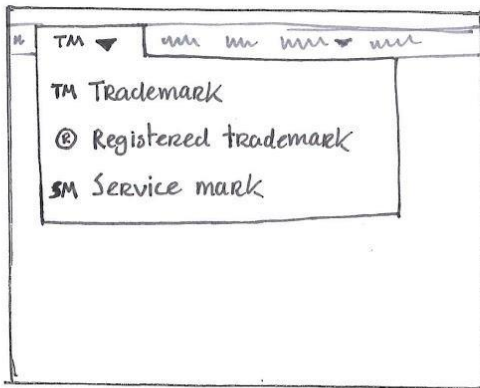
Voordelen

- Minder opdringerig dan de modal in ontwerp A

Nadelen

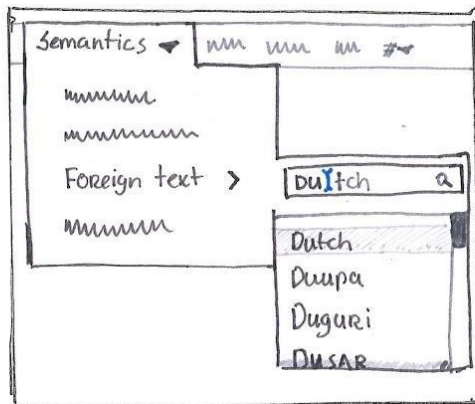
- Kan snel veroorzaken dat de toolbar overvol raakt
- Niet toepasbaar bij grote invoer-UI. Het browsen naar een afbeelding kan bijvoorbeeld niet binnen de toolbar gebeuren vanwege ruimtegebrek.
- Geen ruimte voor feedback of instructie, dus actie moet zeer vanzelfsprekend zijn en dat is niet bij alle datatypen zo.

Element aanmaken - verplichte attribuutwaarde (2)



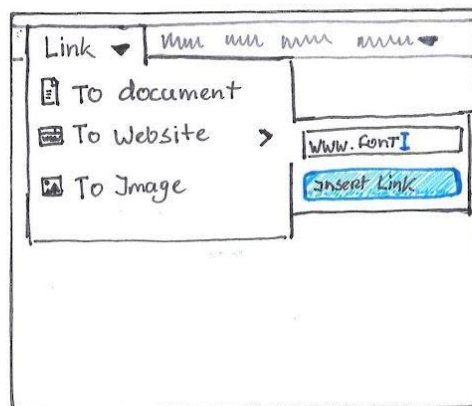
Variante A: Attribuent met vereiste waarde uit een kleine (2-7) lijst met opties. Voor iedere waarde is een eigen knop gemaakt.

- knop 1: `<TM TMtype="TM">`
- knop 2: `<TM TMtype="reg">`
- knop 3: `<TM TMtype="service">`



Variante B: Attribuent met vereiste waarde uit een lange (7+) lijst. Een zoekbalk met autocomplete helpt bij het snel navigeren binnen de lijst. Door de scrollbar kan dat ook handmatig.

- klik op een taal of druk op enter om een invoeropgave te doen



Variante C: Een attribuut met benodigde tekstinput toont een tekstveld om de waarde te typen.

- Druk op de knop om het element met attribuut + waarde in te voegen.

afbeelding 7.7-b: De attribuutwaarde wordt al ingevoerd alvorens het aanmaken van het element wordt getriggerd.

7.8. Toegestane attribuutwaarden

Gebruikersquote:

“Wat moet ik hier precies invoeren?”

Er zijn erg veel verschillende datatypen die in een attribuut kunnen worden opgeslagen. Denk aan een emailadres, een URL, een waarde uit een lijst of het relatieve pad naar een afbeelding. In XML zouden al deze waarden kunnen worden ingevoerd door ze simpelweg in te typen in een tekstveld. Omdat de waarde van zo’n attribuut aan bepaalde regels moet voldoen is het echter beter om een geschikte interface aan te bieden voor het invoeren van dit waardetype.

Door de gebruiker een interface voor invoer te bieden die aansluit bij het type waarde dat zij dienen in te voeren, wordt de kans op fouten drastisch verminderd. Er zijn tal van UI-patronen die gebruikers hier kunnen helpen. Vanwege de grote hoeveelheid datatypen zal ik deze patronen niet in dit project onderzoeken en ontwerpen. Wel is in hoofdstuk 8 een eerste versie te vinden van de lijst met mogelijke datatypen. Mijn voorstel hierbij is om, wanneer een relevante use case voor dat datatype zich voordoet, het meest geschikte UI-patroon te onderzoeken, ontwerpen en testen. Er is een viertal punten erg belangrijk bij het zoeken naar de meest geschikte invoerpatronen:

1. *Voorkom fouten*: probeer, waar mogelijk, te voorkomen dat gebruikers fouten kunnen maken.
2. *Biedt vrijheid*: leg de gebruiker geen restricties op die teveel beperkingen opleveren. De gebruiker moet nog wel het gevoel hebben dat hij controle over de situatie heeft.
3. *Lever feedback*: als er voor de gebruiker nog de mogelijkheid is om fouten te maken, lever dan een duidelijke instructie vooraf en feedback wanneer het fout gaat. Anders blijft de gebruiker constant fouten maken en dat levert frustratie op.
4. *Wees vergevingsgezind*: als er verwachte fouten zijn bij de invoer die je niet kunt voorkomen, wees dan vergevingsgezind en zet deze, waar mogelijk, om naar correcte invoer.

De container waarin deze bewerking plaatsvindt, is afhankelijk van de grootte van het UI-patroon voor de invoer van de waarde. Het browsen binnen een mappenstructuur is bijvoorbeeld te groot om binnen een popover te doen. Hiervoor zal een modal of zelfs een apart scherm moeten worden gebruikt.

7.9. Aangemoedigde elementcombinaties

Gebruikersquote:

“Waarom verschijnen er meerdere nieuwe elementen terwijl ik maar één element probeerde in te voegen?”

Sommige combinaties van elementen zijn niet verplicht in het schema, maar moeten toch worden aangemoedigd. Dit kan zijn omdat deze elementen vaak samen worden gebruikt of om een andere reden. Hoe dan ook is het aannemelijk dat deze situatie bij meerdere klanten voor zal komen. Daarnaast helpt het de gebruiker bij het creëren van waardevolle content.

De oplossing voor dit probleem zal zeer sterk lijken op de oplossing voor het omgaan met *verplichte* ouder-kind relaties, zoals die is beschreven in paragraaf 7.2. Om gebruikers aan te moedigen bepaalde elementen in combinatie met elkaar te gebruiken, zullen er hints aan de gebruiker moeten worden gegeven. Deze hints wijzen de gebruiker op de mogelijkheid tot een bepaalde combinatie. Het grote verschil is dat verplichte kinderen direct in de XML worden toegevoegd. Bij aangemoedigde combinaties is dit niet de bedoeling. Zonder inhoud is zo'n element waarschijnlijk niet van een grote meerwaarde. Het is dus aan te raden om het element pas aan de XML toe te voegen wanneer de gebruiker start met het invoeren van inhoud.

7.9.1. Automatisch invoegen als element met placeholder

Zie afbeelding 7.9-a voor de interactieschetsen bij dit ontwerp.

Dit ontwerp voor aanbevolen elementcombinaties komt sterk overeen met dat voor vereiste elementcombinaties. Er wordt voor alle aanbevolen elementen een placeholder geplaatst bij het invoegen, die een hint geeft over mogelijk te gebruiken elementen.

Er zijn echter wat verschillen in functionaliteit, die hier kort worden uitgelegd. Ten eerste is er een verschil in het moment waarop het element in de XML wordt geplaatst. Vereiste elementen worden, omdat dat moet, direct in de XML geplaatst bij het aanmaken van het ouder-element. Aanbevolen elementen worden pas geplaatst in de XML zodra zij content bevatten. Zo voorkom je overbodige lege elementen in de content. Als tweede worden de placeholders voor aanbevolen elementen verborgen wanneer het template geen focus meer heeft, terwijl die van vereiste elementen altijd blijven staan. Die van vereiste elementen blijven te aller tijde staan. Als laatste kunnen de placeholders voor aanbevolen elementen worden verwijderd.

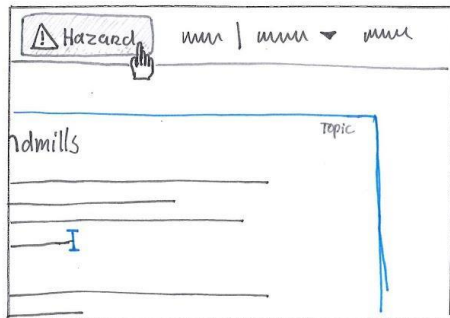
Voordelen

- Consistent met patroon van vereiste elementen
- Niet dwingend

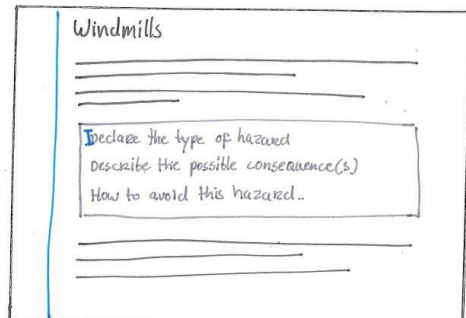
Nadelen

- Wellicht te weinig (visueel) onderscheid

Element aanmaken - aangemoedigde elementcombinaties (1A)



A: Gebruiker klikt op knop om een `<HAZARDSTATEMENT>` in te voegen.

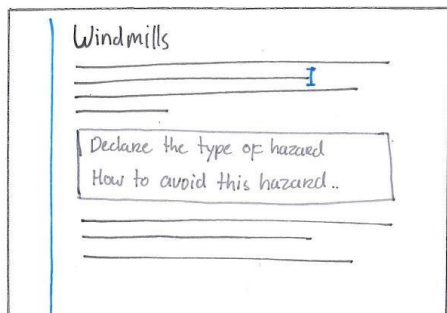


B: `<Hazardstatement>` is geplaatst, cursor staat in het eerste kind waar content wordt verwacht.

- Zowel `<typeofhazard>` (verplicht), `<consequence>` (aanbevolen) en `<howtoavoid>` (verplicht) hebben een placeholder.
- De verplichte elementen staan al in de xml als element. De aanbevolen elementen zijn enkel placeholders.

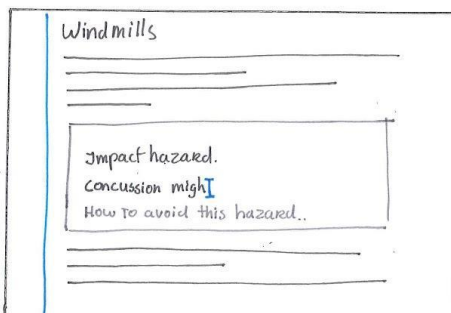
afbeelding 7.9-a: Een hazardstatement wordt aangemaakt, deze bevat zowel verplichte als aanbevolen elementen.

Element aanmaken - aangemoedigde elementcombinaties (1B)



C: De gebruiker navigeert weg uit de hazard statement.

- De placeholders voor niet-verplichte elementen (in dit geval alleen `<consequence>`) worden verborgen als de template-ouder (`<Hazardstatement>`) geen focus heeft.



D: De gebruiker heeft content in de `<consequence>` geplaatst. Zodra de eerste letters daar verschijnen, is dat element van een placeholder in echte xml veranderd.

afbeelding 7.9-b: De placeholders voor aangemoedigde elementen tonen alleen bij focus van de template.

7.10. Aangemoedigde element-attribuuat combinaties

Gebruikersquote:

“Waarom moet ik hier opeens iets doen met een attribuut?”

In sommige gevallen is, net zoals dat bij elementcombinaties zo kan zijn, de combinatie van een element met een bepaald attribuut aanbevolen.

De kern van de oplossing is het slim aanbieden van een mogelijke actie aan de gebruiker. Hierbij is de term “aanbieden” van groot belang: de gebruiker moet op deze aanbieding in kunnen gaan, maar hem ook gemakkelijk kunnen afwijzen. Het aanbod mag niet storend of opdringerig zijn, maar moet wel even de aandacht trekken. Ook is het van belang dat de taken die worden aangeboden zo kort en efficiënt mogelijk uit te voeren zijn. Als deze taken lang duren, zal de gebruiker namelijk in het vervolg minder snel op aangeboden acties ingaan omdat het zijn werk teveel verstoort.

7.10.1.Aanbieden van acties via een popover na het aanmaken

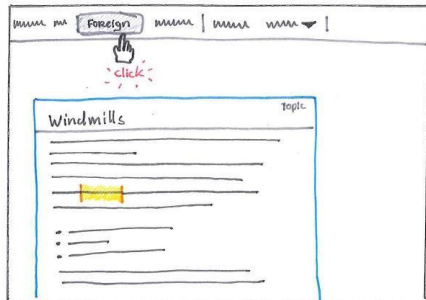
Zie afbeelding 7.9-a en afbeelding 7.9-b voor de interactieschetsen bij dit ontwerp.

In dit ontwerp wordt het toevoegen van aangemoedigde attributen aangeboden via een popover. Deze popover verschijnt direct na het aanmaken van het bijhorende element, net boven de plek waar het element geplaatst is. In de titel van de popover wordt een verzoek tot een bepaalde actie gedaan. Dit verzoek kan worden afgewezen door op “not now” te klikken, of door simpelweg de cursorfocus van het betreffende element af te halen. Het verzoek kan worden geaccepteerd door op de blauwe actieknop te drukken, waarvan de tekst kan variëren. Afhankelijk van het formaat UI dat nodig is voor de invoer, vindt deze actie direct in de popover(*afbeelding 7.9-a*) of in een modal(*afbeelding 7.9-b*) plaats. In dat tweede geval wordt er eerst expliciet aan de gebruiker gevraagd of hij de actie wil uitvoeren. Het navigeren naar de modal is daar dan dus een eigen keuze. Het pijltje in de actieknop “specify date” geeft vast een hint dat het klikken op deze knop een navigatie-actie tot gevolg zal hebben.

Voordelen

- Geen opdringerig patroon
- Verzoek is gemakkelijk af te wijzen en de knoppen zijn consistent met die in modals.

Element aanmaken - Attriboot aanmoedigen (1A)



A: Gebruiker brengt een `<foreign>` aan.
Daarbij is `[language]` een aanbevolen attriboot.



B: Direct na het aanbrengen van `<foreign>` verschijnt er een popover met daarin:

1. Een bescheiden vraag om een waarde in te voeren.
2. De nodige UI voor het invoeren (indien deze in een popover past).

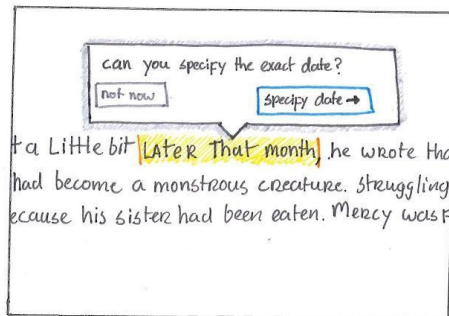
- Als het element focus verliest verdwijnt de popover. Deze komt zonder gebruikersactie niet meer zomaar terug.

afbeelding 7.10-a: Via een popover wordt het invoeren van een attribootwaarde aangemoedigd

Element aanmaken - Attriboot aanmoedigen (1B)

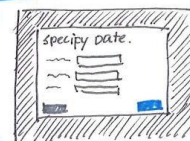


A: Gebruiker brengt een `<DATE>` aan, waarbij [when] een aanbevolen attriboot is.



at a little bit **later that month** he wrote that had become a monstrous creature. struggling because his sister had been eaten. Mercy was

B: Direct na het aanbrengen verschijnt een popover die een vervolgactie voorstelt. De gebruiker kan deze afwijzen door weg te klikken of de knop `not now` te gebruiken. `specify date ->` opent een modal met de invoer UI voor [when]. Dit omdat deze UI te groot is voor in een popover.



C: De modal.

afbeelding 7.10-b: Wanneer de UI te groot is voor een popover, verschijnt er een modal.

Literatuurlijst

Tidwell, J. (2010). *Designing interfaces* (2nd ed., pp. 97-100). Sebastopol: O'Reilly Media.

Mathis, L. (2014, January 22). Design patterns for replacing modal windows. Opgehaald van <https://developer.appway.com/screen/ShowSingleRecipe/selectedRecipeId/1390246496349>

Nielsen, J. (2004, September 27). *Checkboxes vs. Radio Buttons*. Opgehaald van <http://www.nngroup.com/articles/checkboxes-vs-radio-buttons/>

Bijlage H

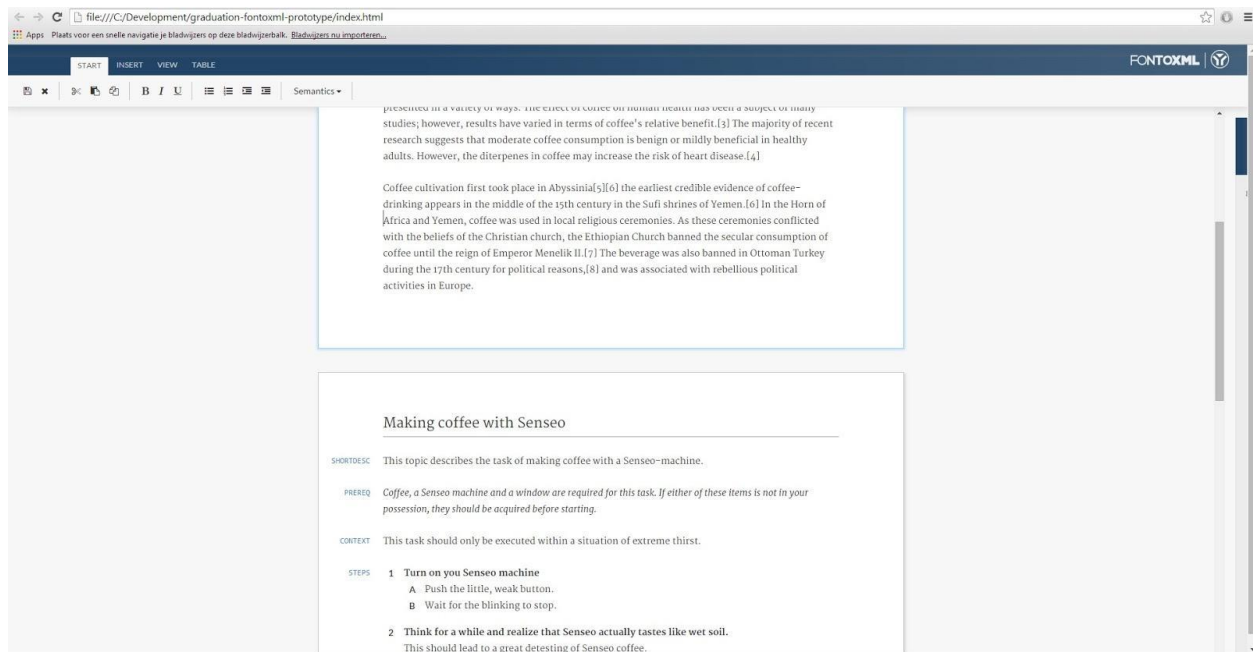
Screenshots prototype

H

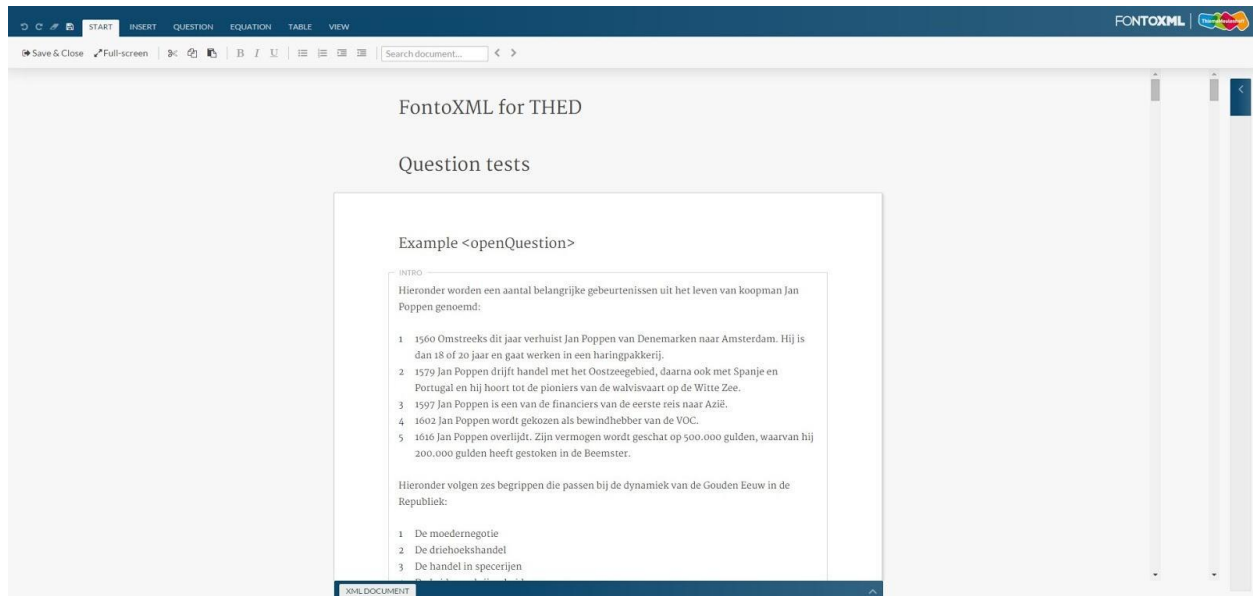
Screenshots prototype

In dit document staat een aantal screenshots van het prototype dat ik heb ontwikkeld voor mijn afstudeeropdracht. Dit geeft echter een beperkte impressie van het prototype en is dus alleen bedoeld voor wanneer de online versie niet bereikbaar is. Niet alle interacties en ontwerpen staan in dit document weergegeven. Voor de volgende designproblemen is er wel een aantal screenshots opgenomen:

Gehele scherm (prototype vs. echte app)



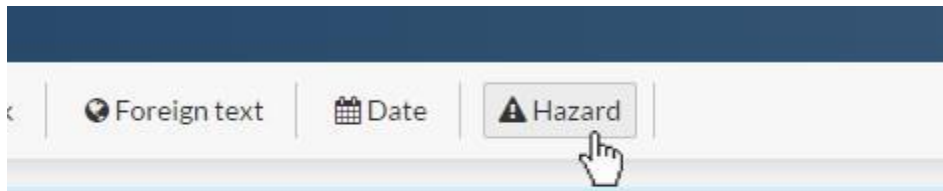
A: Mijn prototype



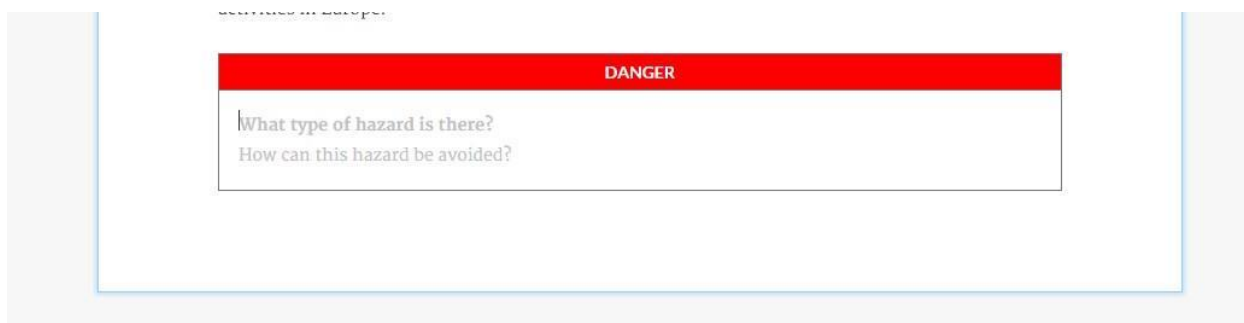
B: Een "echte" versie van FontoXML

Aangemoedigde elementcombinaties

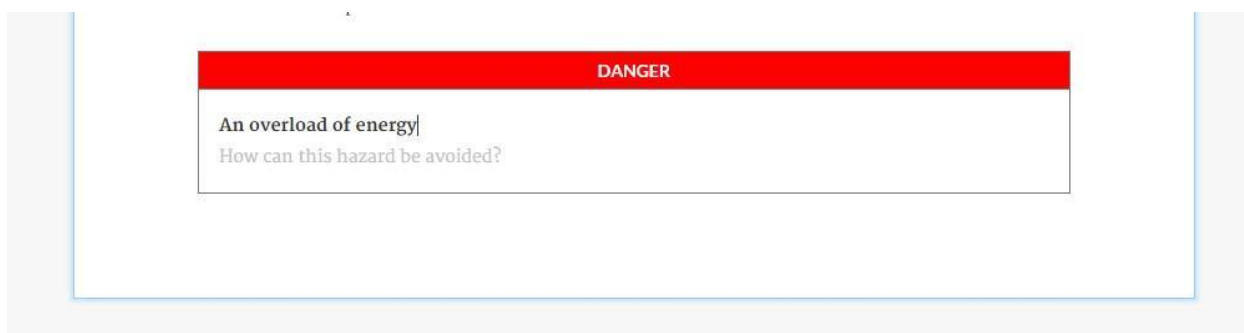
Breng in de eerste topic een “Hazard Statement” aan.



A: maak een hazardstatement aan



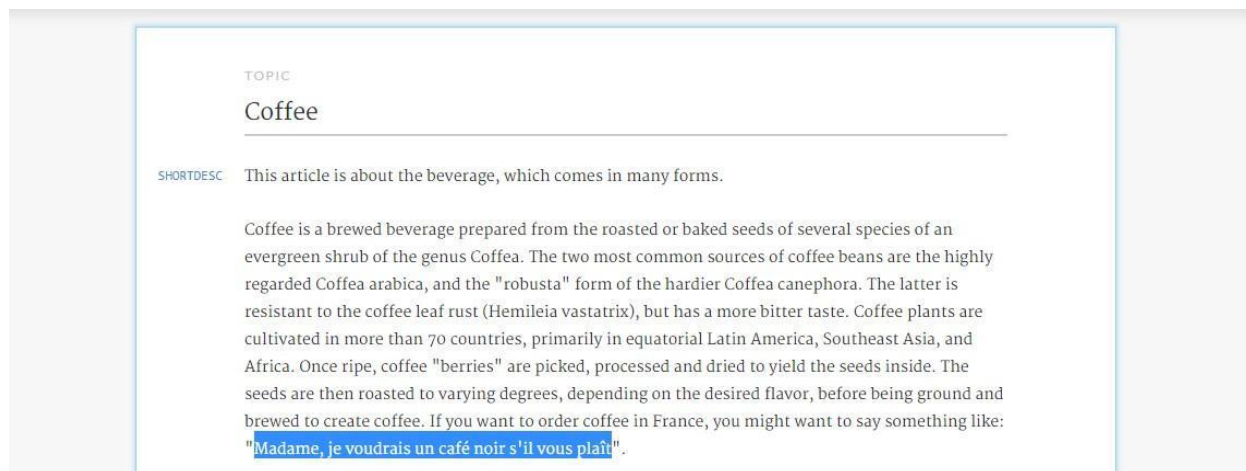
B: De hazardstatement wordt automatisch ingevoegd met een <typeofhazard> en een <howtoavoid>



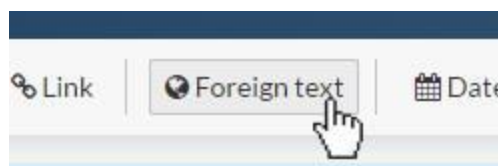
C: Beide elementen bevatten placeholders. Pas als je gaat typen komt er daadwerkelijk content in te staan.

Aangemoedigde element-attribuut combinaties

Markeer een stuk tekst als “Foreign text”, “date” of link.

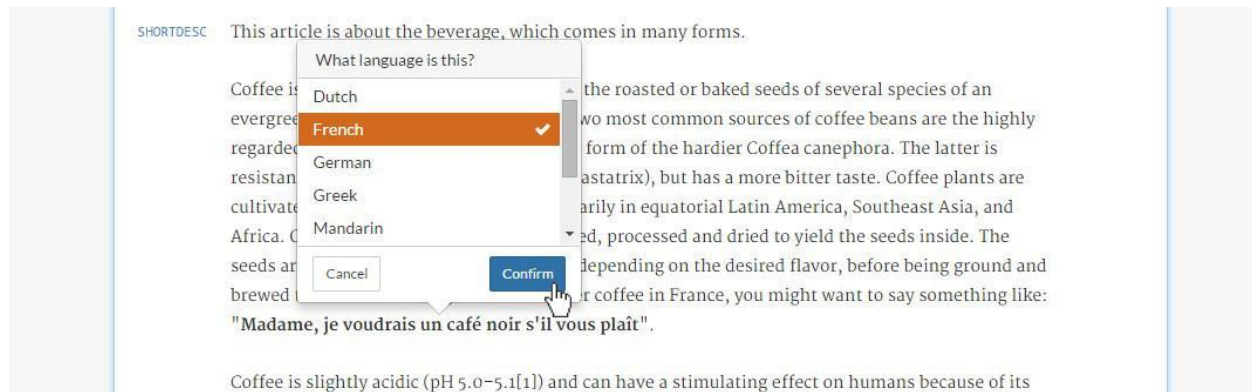


A: Selecteer een stuk tekst



B: Maak een foreign element aan

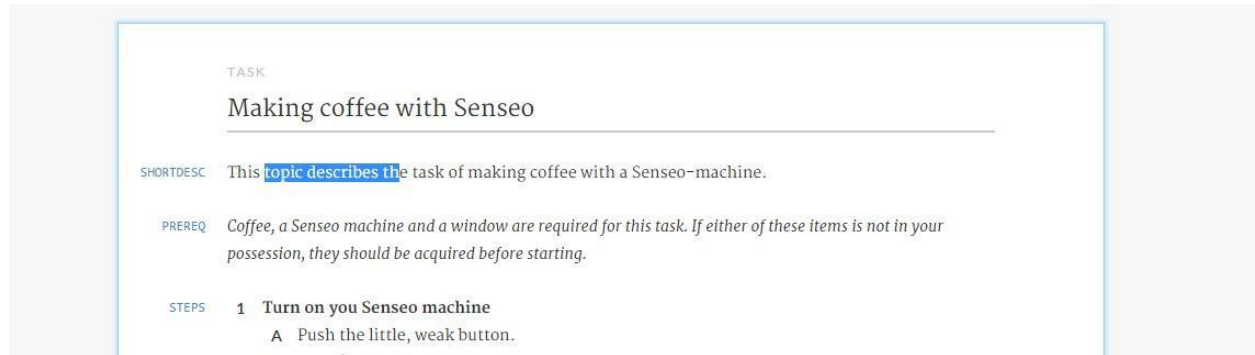
C: Een popover verschijnt met een gesuggereerde actie



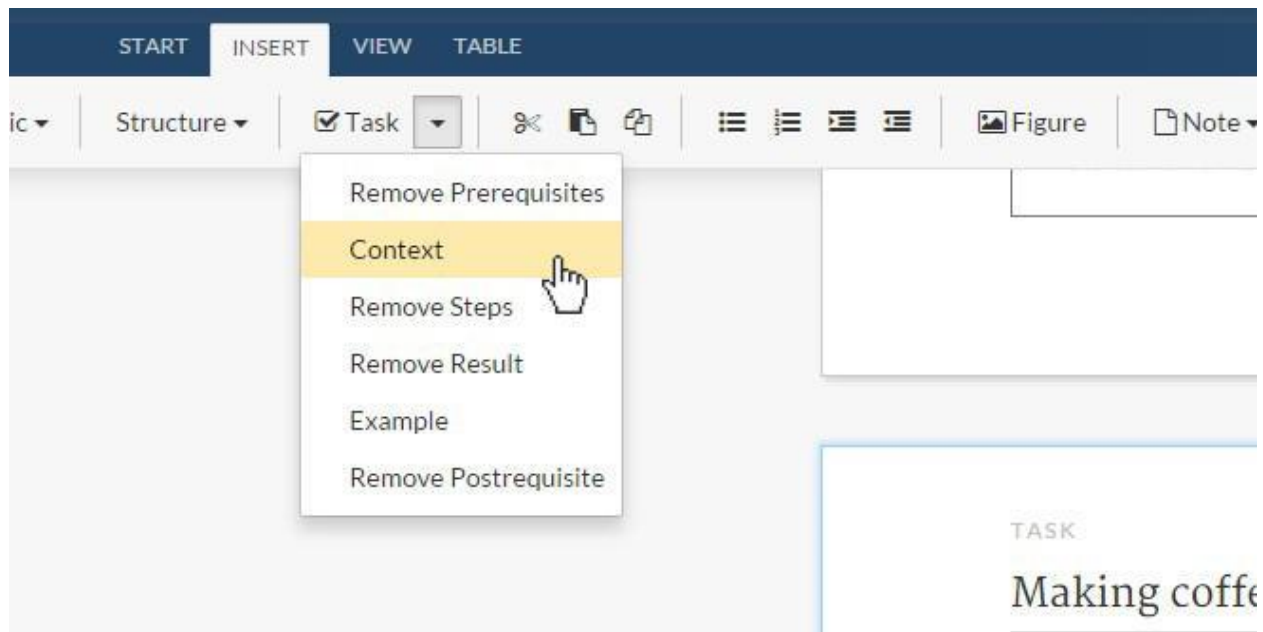
D: Selecteer een taal en klik op confirm om de attribuutwaarde in te voeren

Vaste volgorde van kinderen

Voeg een van de onderdelen van een task in, op een willekeurige plek in de "task".



A: Ga op een willekeurige plek binnen een task staan



B: Maak een nieuw "context" element aan.

TASK

Making coffee with Senseo

SHORTDESC

This e task of making coffee with a Senseo-machine.

PREREQ

Coffee, a Senseo machine and a window are required for this task. If either of these items is not in your possession, they should be acquired before starting.

CONTEXT

Briefly describe the context of this task

C: De context wordt automatisch op zijn vaste positie binnen de task geplaatst.

RESULT

You now have coffee you can enjoy.

POSTREQ

After having your coffee, you might want to visit the toilet.

Context has a fixed location within a task. Therefore, it has been placed in a different location than the cursor position.

D: Onderin beeld verschijnt een bericht die dat aangeeft en verklaard.

Beperkt toegestane kinderen

Voeg een van de onderdelen van “Task” dubbel in via de toolbar.

TASK

Making coffee with Senseo

SHORTDESC

This e task of making coffee with a Senseo-machine.

PREREQ

Coffee, a Senseo machine and a window are required for this task. If either of these items is not in your possession, they should be acquired before starting.

CONTEXT

Briefly describe the context of this task

STEPS

1

Turn on you Senseo machine

A Push the little, weak button.

B Wait for the blinking to stop.

2

Think for a while and realize that Senseo actually tastes like wet soil.

This should lead to a great detesting of Senseo coffee.

3

Throw your Senseo out of the window.

You might want to lift the machine up high, and use an upperhand throwing technique to give it maximum speed while approaching the street.

4

Go to a nice coffee place and order some real stuff.

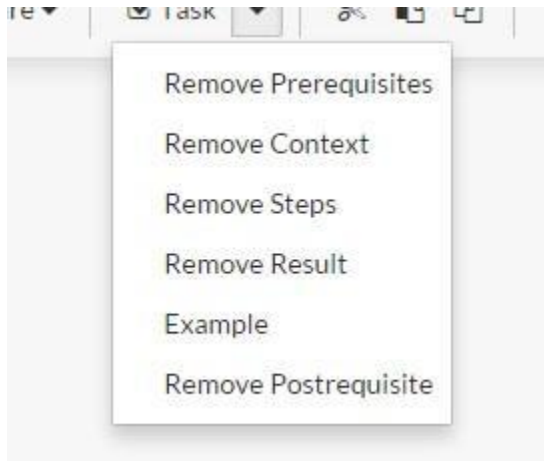
RESULT

You now have coffee you can enjoy.

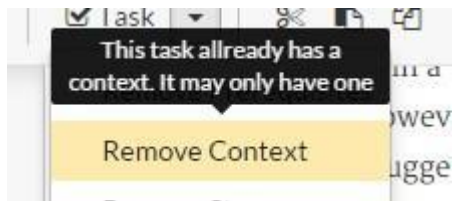
POSTREQ

After having your coffee, you might want to visit the toilet.

A: Deze task bevat al een aantal onderdelen. Alle onderdelen met het blauwe label zijn enkelvoudig toegestaan



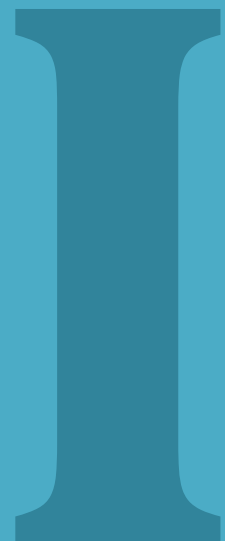
B: Bij het openen van de dropdown kijkt het systeem welke onderdelen al bestaan. De knop voor aanmaken verandert bij die onderdelen in een verwijderknop.



C: Als je een poosje blijft hoveren legt een tooltip de situatie uit.

Bijlage I

Testplan



Testplan

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Inleverdatum: 9 januari 2015



Inhoudsopgave

1. Inleiding	169
2. Doel van de test	170
3. Onderzoeksvragen	171
4. Testpersonen	172
5. Testmethode	173
6. Testomgeving en benodigdheden	174
6.1. Testomgeving	174
6.2. Benodigdheden voor de testpersonen	174
6.3. Benodigdheden voor de moderator	174
7. Rol van de moderator	176
8. Uitvoering	177
8.1. Communicatie vooraf	177
8.1.1. Email met uitleg over de test	177
8.1.2. Mondelinge introductie	177
8.2. Script	178
Taak 1: Opwarmertje	178
Taak 2: Invoegen van een element met verplichte kinderen	179
Taak 3: Invoegen van een element met vaste volgorde	180
Taak 4: Invoegen van elkaar uitsluitende elementen	181
Taak 5: Enkel toegestane elementen	182
Taak 6: Aangemoedigde attributen	183
Taak 7: Verplichte attributen	184
9. Verwerken van resultaten	185
Bijlage I: Hand-out gebruikerstest	186
Bijlage II: Script voor de moderator	187

1. Inleiding

Dit testplan is geschreven naar aanleiding van de usabilitytests die ik voor mijn afstudeerproject zal gaan uitvoeren. Om te zorgen dat deze test goed voorbereid kan worden uitgevoerd en ik niet voor verrassingen kom te staan, is dit testplan opgesteld. Hierin wordt beschreven hoe er vooraf, tijdens en na de test te werk moet worden gegaan voor het behalen van optimale resultaten.

In hoofdstuk 2 worden de doelen van deze test beschreven. Als deze doelen achteraf zijn behaald kan worden gesteld dat de test is geslaagd. Om deze doelen te bereiken, moet er bepaalde informatie verzameld worden. Welke informatie dat precies kan worden achterhaald via de onderzoeksvragen. Deze zijn te vinden in hoofdstuk 3. In hoofdstuk 4 wordt beschreven welke testpersonen er nodig zijn en hoe deze verzameld dienen te worden. Hoofdstuk 5 beschrijft en beargumenteert de methode die zal worden gebruikt bij de test. De benodigdheden voor de test zijn te vinden in hoofdstuk 6. In hoofdstuk 7 staan de taken van de testmoderator beschreven. Dit als handleiding bij het uitvoeren van de test zelf. De testtaken die door de gebruikers moeten worden uitgevoerd staan stuk voor stuk beschreven in hoofdstuk 8. Tot slot staat in hoofdstuk 9 hoe de verzamelde gegevens kunnen worden verwerkt tot een leesbaar en bruikbaar testresultaat.

2. Doel van de test

Deze usabilitytest heeft drie hoofddoelen. Deze doelen zijn bewust breed geformuleerd, omdat zij toepasbaar moeten blijven op alle verschillende taken. Per testtaak is nog een aparte doel geformuleerd, gebaseerd op het achterliggende designprobleem.

1. Controleren of gebruikers met de ontworpen interfacepatronen effectief verschillende elementen aan kunnen maken, ondanks de verschillende invloeden op die bewerking.

Gebaseerd op gebruikersonderzoek en ontwerptheorie kun je goede ontwerpen maken. Het kan echter zo zijn dat er ergens vergissingen zijn gemaakt, bijvoorbeeld in het trekken van conclusies over het onderzoek of in het toepassen van theorie op een probleem. Om met enige zekerheid te kunnen zeggen dat de ontwerpen voorzien in hun doelen (het gebruiksvriendelijk aan kunnen maken van verschillende elementen) zullen deze getest moeten worden.

2. Achterhalen van verbeterpunten in de ontworpen interfacepatronen.

Een designer die vooraf al zegt dat zijn ontwerp niet getest hoeft te worden omdat het al goed is, liegt. In elk ontwerp zitten verbeterpunten. Zeker in “nieuwe” ontwerpen zoals in het geval van dit project zullen deze verbeterpunten niet altijd even zichtbaar zijn. Testen is dus een goede manier om de scherpe randjes van je ontwerpen te halen.

3. Ontdekken welk van de ontworpen varianten het meest geschikt is als oplossing voor FontoXML

Voor een aantal van de designproblemen zijn meerdere oplossingen ontworpen. In deze test kan worden onderzocht welke van deze oplossing het meeste voordelen biedt en dus meer geschikt is voor implementatie in FontoXML.

3. Onderzoeksvragen

In de onderzoeksvragen wordt beschreven welke informatie ik hoop te verkrijgen door het uitvoeren van de test. Per onderzoeksvraag wordt ook verklaard waarom deze informatie belangrijk is om te hebben.

Begrijpen de testpersonen bij het aanmaken van elementen de regels, restricties of best practices *vooraf* waar zij mee te maken hebben?

Er kan worden aangenomen dat gebruikers geen kennis van XML-schema's hebben. Om verrassingen te voorkomen en te zorgen dat een actie aansluit bij de verwachtingen van de gebruikers, zou het dus goed zijn om vooraf al kenbaar te maken met welke regel een gebruiker te maken heeft.

Begrijpen de testpersonen bij het aanmaken van elementen de regels, restricties of best practices *achteraf* waar zij mee te maken hebben?

Het is niet altijd mogelijk om vooraf duidelijk te maken waar de gebruiker mee te maken heeft. Soms is het niet te voorspellen, soms is het simpelweg te storend om dit vooraf uit te leggen. In deze gevallen is het ook een optie om te zorgen dat de gebruiker in ieder geval achteraf begrijpt wat er gebeurd is. Op deze manier kan worden voorkomen dat de gebruiker het gevoel heeft niet de baas te zijn over het systeem. Dit zou namelijk tot frustratie kunnen leiden.

Hoe gemakkelijk en succesvol voeren de testpersonen de verschillende varianten van het aanmaken uit?

Een belangrijk aspect van gebruiksvriendelijkheid is dat gebruikers hun taken zo efficiënt en gemakkelijk mogelijk uit kunnen voeren. Hiervoor kan per taak worden gekeken naar het gewenste eindresultaat, het ideale pad daarnaartoe en de afwijkingen daarop.

Ervaren de testpersonen de ontworpen patronen als hinderlijk?

Sommige van de interfacepatronen beperken (tijdelijk) de handelingsvrijheid van de gebruiker in de applicatie. Dit is noodzakelijk, maar zou als hinderlijk kunnen worden ervaren. Het is van belang om dit zoveel mogelijk te vermijden.

4. Testpersonen

Voor het uitvoeren van de test zijn minimaal twee, maar ideaal gezien drie typen testpersonen nodig. Allereerst dient er een tweetal “proeftesten” te worden uitgevoerd. Deze tests zijn niet per se gericht op het verzamelen van waardevolle resultaten, maar meer op het controleren of de test duidelijk en goed is opgebouwd. Aan de testpersonen voor deze proeftest worden de volgende eisen gesteld:

- Benodigde testpersonen: 2
- Leeftijd: 25 - 65 jaar
- Geen kennis van XML
- Niet eerder met FontoXML gewerkt
- Gewend aan het schrijven van documenten in Microsoft Word

Op deze manier hebben de testpersonen in ieder geval een aantal overeenkomsten met de echte gebruikersgroep. Het belangrijkste verschil is dan dat zij waarschijnlijk minder ervaring zullen hebben met het schrijven van gestructureerde content. Ook zal eventuele vakinhoudelijke kennis waarschijnlijk voor een ander vakgebied zijn dan die van de gebruikers bij ThiemeMeulenhoff. Het is van belang om met begrijpelijke testcontent te werken, zodat de testpersonen het onderwerp begrijpen en daar eventueel zelf wat over kunnen schrijven. Het is dus gemakkelijk om twee mensen te nemen uit hetzelfde vakgebied, zodat er met dezelfde testcontent gewerkt kan worden.

Als de proeftesten zijn afgerond en eventuele verbeteringen in het script zijn aangebracht, kunnen de echte gebruikerstesten plaatsvinden. Deze tests dienen gehouden te worden met representatieve testpersonen. Hiervoor zijn vakexperts het meest geschikt. Indien het niet mogelijk is om vakexperts te werven als testpersoon, kan eventueel nog met invoerders worden getest. Voor de testpersonen bij de daadwerkelijke test zijn de volgende eisen van belang:

- Benodigde testpersonen: 4
- Leeftijd: 25 - 65 jaar
- Geen kennis van XML
- Niet eerder met FontoXML gewerkt
- Beroepsmatig bezig met het schrijven van lescontent
- Werkzaam bij ThiemeMeulenhoff
- Gewend aan het schrijven van documenten in Microsoft Word

5. Testmethode

Gebaseerd op de doelen en onderzoeksvragen van mijn test heb ik een methode van testen gekozen die “Comparison Test” heet. Deze methode is goed toe te passen wanneer er verschillende ontwerpen moeten worden getest die in dezelfde behoefte moeten voorzien. Door beide ontwerpen op een kwalitatieve manier te testen kunnen er voor- en nadelen van ieder ontwerp gevonden worden. De conclusie uit deze vorm van onderzoek kan zijn dat een van de ontwerpen meer geschikt is dan de ander, of dat er een soort “hybride” oplossing ontwikkeld dient te worden, waarin de sterke punten van twee of meer alternatieven worden gecombineerd.

Deze testmethode zal worden gecombineerd met de “Thinking Aloud” methode, waarbij testpersonen een aantal taken moeten uitvoeren en daarbij hardop hun gedachten uitspreken. Deze veelgebruikte techniek biedt inzicht in het denkproces van gebruikers, wat veel waardevolle informatie kan opleveren voor het verhelpen van eventuele usabilityproblemen of knelpunten. Voor het vergelijken van de verschillende ontwerpen per designprobleem zal de helft van de testpersonen de taken uit moeten voeren met ontwerp A en de andere helft met ontwerp B. Op deze manier worden gebruikers niet geconfronteerd met conflicterende oplossingen voor gelijksoortige problemen: dit zou in de eindsituatie ook niet voor komen.

6. Testomgeving en benodigdheden

In dit hoofdstuk wordt beschreven in welke omgeving de test dient te worden uitgevoerd. Ook wordt beschreven welke benodigdheden er zijn om deze test succesvol te kunnen uitvoeren. Zorg dat er tijdig wordt geregeld dat alle aspecten uit dit hoofdstuk zijn gedekt.

6.1. Testomgeving

Idealiter worden de tests uitgevoerd op de computer op de officiële werkplek van de testpersoon. Omdat er waarschijnlijk geen tijd is voor het installeren van screenrecording software op de computers van de testpersonen, zullen de tests worden uitgevoerd op een meegebrachte laptop.

6.2. Benodigdheden voor de testpersonen

Om de test succesvol uit te kunnen voeren, zijn er twee dingen waar gebruikers beschikking over dienen te hebben voordat de test begint:

1. Hand-out met vereiste kennis

Er is een bepaalde basiskennis die nodig is om met FontoXML gestructureerde content te kunnen bewerken. Uit eerdere gebruikersonderzoeken is gebleken dat de meeste gebruikers niet over deze kennis beschikken. Om de tests succesvol uit te kunnen voeren, zal deze basiskennis dus vooraf aan de gebruiker moeten worden aangeboden. Het is van belang dat hier zo min mogelijk informatie overgedragen hoeft te worden. De gebruiker moet niet het gevoel krijgen dat hij deze kennis “zou moeten weten”. Om die reden moet deze kennis op een hand-out worden aangeboden, die eventueel tijdens de test kan worden geraadpleegd. De hand-out is vinden in Bijlage A: Hand-out gebruikerstest. Deze hand-out wordt na de mondelinge introductie aan de testpersoon overhandigd.

2. Begrijpelijke testcontent

In de werkelijkheid zullen testpersonen FontoXML gaan gebruiken voor het schrijven en bewerken van documenten die met hun vakgebied te maken hebben. De content en de betekenis daarvan is in die situatie dus vertrouwd en begrijpelijk. Om deze situatie na te bootsen dient er, bij het uitvoeren van de taken, testcontent te worden gebruikt die te maken heeft met het vakgebied van de gebruiker. Deze testcontent moet vooraf worden verzameld en klaargezet in de applicatie.

6.3. Benodigdheden voor de moderator

- Laptop met screen recording software

- Mogelijkheid om audio op te nemen
- Een bereikbare plek voor het prototype.

7. Rol van de moderator

De moderator is de begeleider van de usabilitytests. Hij neemt de tests af bij de gebruiker. Hierbij heeft hij een aantal taken die moeten worden uitgevoerd bij iedere test.

1. **Introductie geven**

De gebruiker welkom heten, uitleggen wat er gaat gebeuren en wat er wordt verwacht.

2. **Zorgen dat de opnames draaien**

Het is belangrijk dat de moderator bij elke test controleert op de benodigde opnames draaien, zodat er geen kostbare gegevens verloren gaan.

3. **Observeren en notuleren**

De moderator observeert aandachtig hoe de gebruiker zijn taken uitvoert. Hierbij worden opvallende acties en gedachten genoteerd, zodat deze kunnen worden verwerkt in de resultaten.

4. **Instrueren**

De moderator zorgt dat de gebruiker bij iedere taak begrijpt wat hij moet doen.

5. **Zwijgen**

Gebruikers hebben de neiging om vragen te stellen bij het uitvoeren van testtaken. Het is aan de moderator om te bepalen op welke vragen hij antwoord geeft, en voor welke vragen de gebruiker dit antwoord zelf zou moeten vinden.

6. **Vragen**

Voor sommige gebruikers is het moeilijk om hun gedachten uit te spreken tijdens het uitvoeren van de testtaken. In dit geval is het aan de moderator om de gebruiker eraan te herinneren dat dit nodig is. Zonder hardop te denken is het namelijk moeilijk te achterhalen wat gebruikers denken en waarom ze bepaalde dingen doen.

7. **Ben je klaar?**

Als moderator is het handig om na iedere testtaak te controleren of een gebruiker denkt klaar te zijn.

8. Uitvoering

In dit hoofdstuk staat precies beschreven wat er moet gebeuren voor- en tijdens de uitvoering van de usabilitytest. Het gaat hierbij om de nodige communicatie met de testpersonen en de taken die tijdens de tests moeten worden uitgevoerd.

8.1. Communicatie vooraf

Voordat de tests worden uitgevoerd, zal er aan de testpersonen wat informatie moeten worden verstrekt over de opzet en de inhoud van de test. Zo krijgen de testpersonen al vooraf een beeld van de test waar zij aan deelnemen. Door dit te doen kan er wat spanning worden weggenomen, die doorgaans bij testpersonen aanwezig is.

8.1.1. Email met uitleg over de test

Voorafgaand aan de test (ongeveer twee dagen van tevoren) zal er naar de testpersonen een korte email worden gestuurd. In deze email dient kort te worden uitgelegd wat het doel van de test is, wat er van de testpersonen wordt verwacht en wat er tijdens de test ongeveer zal gaan gebeuren.

8.1.2. Mondelinge introductie

Eenmaal ter plekke dient, voordat de test begint, nogmaals aan de testpersonen te worden uitgelegd wat er precies gaat gebeuren. De kans bestaat namelijk dat zij de email niet hebben gelezen of niet goed hebben begrepen. Daarnaast is het belangrijk om de testpersoon hier op zijn gemak te stellen. Een test kan namelijk erg “zwaar” voelen en testpersonen hebben soms het gevoel dat *zij* beoordeeld gaan worden. Leg hen uit dat dit niet het geval is, maar dat juist de applicatie hier beoordeeld zal worden. Dingen die fout gaan zijn niet de schuld van de gebruiker, maar van het systeem. Na de mondelinge introductie dient de hand-out met benodigde kennis (bijlage A) aan de gebruiker te worden overhandigd. Verzoek de testpersoon, voordat hij begint, nogmaals om hardop te denken.

8.2. Script

In deze paragraaf wordt beschreven welke testtaken er allemaal zijn geformuleerd. Voor iedere testtaak is er een gebruikersinstructie te vinden. In de titel van de taak is aangegeven voor welk designprobleem de testtaak is geschreven. Ook staat er met welke elementen de taak getest zou kunnen worden. Als laatste is er een lijst met aandachtspunten opgesteld waar de moderator op kan letten wanneer een testpersoon deze taak uitvoert. Tot slot is, ter controle, de ideale flow voor het uitvoeren van deze taak gedefinieerd.

Om de gebruikers niet meteen in het diepe te gooien en even bekend te laten worden met de applicatie, is er een “opwarmertje” gemaakt. In deze taak kan de gebruiker een klein beetje rondkijken in de applicatie en wordt er gevraagd om een heel eenvoudige actie uit te voeren die zij gewend zijn uit andere tekstverwerkers. Zo wordt het gevoel gewekt dat de gebruiker inderdaad in een “gewone tekstverwerker” bezig is.

Introductie

Voorafgaand aan de test dient een mondelinge introductie te worden gegeven. Hierin moeten aan de deelnemer (nogmaals) de volgende dingen worden uitgelegd:

- Bedankt voor je tijd, leuk dat je wilde deelnemen.
- De context van de test (doel, duur, wordt opgenomen)
- Gestructureerd schrijven
- FontoXML
- Het is een prototype: niet representatief voor de echte applicatie.

Taak 1: Opwarmertje

Gebruikersinstructie

“Kijk even een minuutje rond in het scherm, en beschrijf wat je daar ziet”

Daarna:

“Lees de eerste alinea van het bovenste topic. Geef voor {woord} aan dat dit scheefgedrukte tekst moet zijn.”

Taak 2: Invoegen van een element met verplichte kinderen

Gebruikersinstructie

“Het eerste stuk / deel gaat over koffie. Zorg dat er onder in dat stuk een “gevaarenbeschrijving” komt te staan, waarin de gevaren van koffie beschreven worden.”

- *Geef aan waarom dat wel of niet gelukt is*

Testelement(en)

<hazardstatement>

Ideale flow

1. Cursor in topic
2. Naar insert
3. Klik “hazard statement”
4. Klik “typeofhazard”
5. Schrijf inhoud
6. Klik “Howtoavoid”
7. Schrijf inhoud
8. *Taak voltooid*

Aandachtspunten

Aandachtspunt	Commentaar
Ziet de gebruiker waar het element is ingevoegd?	
Wat beschrijft de gebruiker te zien nadat het element is ingevoegd?	
Snappen gebruikers waarom er meerder elementen zijn ingevoegd?	
Begrijpt de gebruiker de placeholders?	
Overig	

Taak 3: Invoegen van een element met vaste volgorde

Gebruikersinstructie

“Ergens in het document vind je een taak. Zorg dat er in deze taak een resultaat komt te staan met daarin een korte beschrijving van het mogelijke resultaat.”

- *Geef aan waarom dat wel of niet gelukt is*

Testelement(en)

<result>, <context>, <prereq>, <postreq>, <example>

Ideale flow

1. Focus task
2. Insert-tab
3. Task-dropdown
4. Klik “Result”
5. Begin te typen
6. *Taak voltooid*

Aandachtspunten

Aandachtspunt	Commentaar
Vindt de gebruiker de Task?	
Waar gaat hij met de cursor staan?	
Ziet hij waar het element is ingevoegd?	
Ziet hij de feedback message?	
Kan de gebruiker benoemen wat er gebeurd is?	
Overig	

Taak 4: Invoegen van elkaar uitsluitende elementen

Gebruikersinstructie

Zorg dat er boven in het eerste stuk een “samenvatting” komt te staan.

- *Geef aan waarom dat wel of niet gelukt is*

Testelement(en)

<shortdesc>, <abstract>

Ideale flow

1. Focus topic
2. Insert tab
3. Topic dropdown
4. Shortdesc
5. Schrijven
6. Abstract proberen aan te brengen
7. Verwijder shortdesc
8. Abstract aanbrengen
9. Schrijven
10. *Taak voltooid*

Aandachtspunten

Aandachtspunt	Commentaar
Ziet de gebruiker de verandering in de toolbar?	
Kan de gebruiker benoemen wat er aan de hand is?	
Ziet de gebruiker de tooltips?	
Overig	

Taak 5: Enkel toegestane elementen

Gebruikersinstructie

“Voeg een tweede mogelijke resultaat toe aan de taak (via de toolbar want enter)”

- *Geef aan waarom dat wel of niet gelukt is*

Testelement(en)

<result>

Ideale flow

1. Focus task
2. Insert tab
3. Task dropdown
4. Hee, dat kan niet.
5. *Taak voltooid*

Aandachtspunten

Aandachtspunt	Commentaar
Begrijpt de gebruiker vooraf al dat het niet kan?	
Leest de gebruiker de tooltip?	
Begrijpt de gebruiker achteraf dat het niet kan?	
Overig	

Taak 6: Aangemoedigde attributen

Gebruikersinstructie

“Lees het eerste topic nog eens. Markeer daarin een naar een website, een stukje buitenlandse tekst en een datum”

Testelement(en)

<link>, <date>, <foreign>

Ideale flow

1. Markeren in topic
2. Insert tab
3. Button aanklikken
4. Attribuuwaarde invullen
5. *Taak voltooid*

Aandachtspunten

Aandachtspunt	Commentaar
Gaat de gebruiker in op de aanmoedigingen?	
Wat vind de gebruiker van de popovers?	
Overig	

Taak 7: Verplichte attributen

Gebruikersinstructie

“Onder in het eerste topic staat het woord “Senseo”. Geef aan dat het hier gaat om een geregistreerd handelsmerk.

Testelement(en)

<tm>

Ideale flow

1. Markeren in topic
2. Insert tab
3. Button aanklikken
4. Type kiezen
5. *Taak voltooid*

Aandachtspunten

Aandachtspunt	Commentaar
Wordt de attribuutwaarde ingevuld?	
Begrijpt de gebruiker wat hij moet doen?	
Overig	

9. Verwerken van resultaten

Deze test is niet gericht op het verzamelen van cijfers, tijden of statistieken. Het is een kwalitatieve tekst, waarbij het wel of niet slagen van een interactiepatroon wordt beoordeeld door de moderator aan de hand van een aantal vragen. Een constatering is een potentieel resultaat als deze bij minimaal 3 testpersonen kan worden gedaan. Indien dit gebeurt, zal er door de moderator moeten worden onderzocht wat de oorzaak van deze constatering is, zodat hier later een advies over uit kan worden gebracht. Dit geldt voor zowel positieve als negatieve constateringen.

De resultaten van de tests zullen bestaan uit schermopnames, audio opnames en notities. Deze resultaten zullen achteraf op neutrale wijze worden verwerkt in een testrapport. Op basis van dit testrapport wordt in een adviesrapport voor ieder ontwerp beschreven of dit ontwerp geschikt is voor implementatie, aanpassing nodig heeft of ongeschikt is voor implementatie.

Bijlage I: Hand-out gebruikerstest



Bedankt voor uw deelname aan dit gebruikersonderzoek voor FontoXML. Met dit onderzoek hopen wij te ontdekken hoe gebruiksvriendelijk onze applicatie momenteel is, en waar er mogelijke verbeterpunten liggen.

Bij deze test zal aan u worden gevraagd om een aantal taken uit te voeren. Hierbij worden op geen enkele wijze uw vaardigheden getest, enkel het ontwerp van de applicatie. Doe dingen dus niet bewust anders: een eerlijk resultaat is het beste resultaat. Als iets moeilijk blijkt of niet lukt, nemen wij dit dan ook mee als verbeterpunt voor onze ontwerpen. Tijdens de test wordt u gevraagd om hardop uit te spreken wat u denkt, wat u doet en waarom u dat doet. Zo kunnen wij in onze applicatie rekening houden met de motivatie van de gebruikers.

Tijdens de test zullen zowel het scherm van de laptop als het geluid worden opgenomen. Zo kunnen wij achteraf beter analyseren wat er precies is gebeurd. U wordt niet gefilmd.

Voor het werken met FontoXML is het handig om een aantal begrippen te kennen. Om zeker te weten dat u begrijpt wat er met deze begrippen wordt bedoeld, zijn deze hieronder kort beschreven.

Ondersteunende begrippen

- ***Gestructureerd schrijven***

FontoXML draait om gestructureerd schrijven. Dit houdt in dat er voor de teksten in een document op allerlei manieren kan worden aangegeven wat voor soort tekst dit is, of waar het voor staat.

- ***Topics***

In gestructureerde teksten wordt gewerkt met zogeheten topics. Een topic is een stuk tekst waarin één op zichzelf staand onderwerp wordt beschreven. Topics worden in FontoXML weergegeven als losse “velletjes”.

Bijlage II: Script voor de moderator

Bij de gebruikerstest dienen de testtaken in de volgende volgorde te worden uitgevoerd:

Taak 1: Opwarmertje

“Kijk even een minuutje rond in het scherm, en beschrijf wat je denkt te zien”

Daarna:

“Lees de eerste alinea van het bovenste topic. Markeer daar het woord {onderwerp} als dikgedrukte tekst. Markeer vervolgens het woord daarna als scheefgedrukt. ”

Taak 2: Invoegen van een element met verplichte kinderen

1. “Het eerste stuk / deel gaat over koffie. Zorg dat er onder in dat stuk een “gevaarenbeschrijving” komt te staan, waarin de gevaren van koffie beschreven worden.”

Taak 3: Invoegen van een element met vaste volgorde

- 1. “Ergens in het document vind je een taak. Zorg dat er in deze taak een “resultaat” van de taak komt te staan.”*
- 2. Geef aan waarom dat wel of niet kan.*

Taak 4: Invoegen van elkaar uitsluitende elementen

- 1. “Maak voor het bovenste topic een korte beschrijving (shortdesc)”*
- 2. “Maak voor het bovenste topic een samenvatting”*
- 2. Geef aan waarom dat wel of niet kan.*

Taak 5: Enkel toegestane elementen

- 1. “Voeg een tweede mogelijke resultaat toe aan de taak (via de toolbar want enter)”*
- 2. Geef aan waarom dat wel of niet kan.*

Taak 6: Aangemoedigde attributen

*“Lees het eerste topic nog eens. Markeer daarin:
een **website**, een stukje **buitenlandse tekst** en een **datum**”*
datum”

Bijlage J

Testrapport



Testrapport

Usabilitytests FontoXML prototype bij ThiemeMeulenhoff

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Inleverdatum: 9 januari 2015



FONTOXML

Inhoudsopgave

1. Inleiding	191
2. Overzicht testresultaten	192
2.1. Slagen van de taken	192
2.2. Begrip van de situatie	193
3. Opvallende observaties	195
3.1. Algemene observaties	195
3.2. Invoegen van een element met verplichte kinderen - taak 2	195
3.3. Invoegen van een element met vaste volgorde - taak 3	196
3.4. Invoegen van elkaar-uitsluitende elementen - taak 4	196
3.5. Invoegen van enkelvoudig toegestane elementen - taak 5	196
3.6. Invoegen van elementen met aangemoedigde attributen - taak 6	197
3.7. Invoegen van elementen met verplichte attributen - taak 7	197
4. Conclusies per ontwerpprobleem	198
4.1. Elementen met verplichte kinderen	198
4.2. Elementen die enkelvoudig zijn toegestaan	199
4.3. Elementen met een vaste volgorde	199
4.4. Elkaar uitsluitende elementen	200
4.5. Elementen met aangemoedigde attributen	200
4.6. Elementen met verplichte attributen	201
5. Verbetervoorstellen	202
6. Verbeterpunten test	204
 Bijlage I: Aandachtspunten per taak	 205
Bijlage II: Uitwerking resultaten	212

1. Inleiding

Dit document is bedoeld om inzicht te bieden in de resultaten van de gebruiksvriendelijkheidstest van dinsdag 16 december. Deze test werd gehouden om te ontdekken of de ontworpen interfacepatronen voor het aanmaken van elementen in FontoXML begrijpelijk en intuïtief werken

In hoofdstuk 2 van dit document worden de resultaten van de verschillende deelnemers samengevat en overzichtelijk gemaakt. Hierbij wordt gelet op een aantal punten zoals voltooiing, begrip van de situatie en een aantal andere vooraf opgestelde criteria. Er wordt in dit hoofdstuk nog geen verder oordeel geveld over deze resultaten. In hoofdstuk 3 worden per testtaak de bijzondere en opvallende resultaten uitgelicht. Hoofdstuk 4 bevat de conclusies die per ontwerp kunnen worden getrokken aan de hand van deze resultaten. In hoofdstuk 5 worden aanbevelingen en verbetervoorstellen gedaan voor deze ontwerpen. In bijlage A is een uitwerking van de ruwe testresultaten te vinden.

2. Overzicht testresultaten

In dit hoofdstuk staan de resultaten van de verschillende gebruikerstests beschreven. Door deze op een overzichtelijke manier naast elkaar te zetten, kunnen deze resultaten worden vergeleken en beoordeeld. Op basis hiervan kan vervolgens een aantal conclusies worden getrokken. Ook kunnen er zo verbetervoorstellen worden gedaan voor opvallende tekortkomingen in de geteste ontwerpen. Ter ondersteuning is hieronder nogmaals een overzicht van het ontwerpprobleem per testtaak te vinden:

<i>Taak 1:</i>	Geen ontwerpprobleem: Introductietaak
<i>Taak 2:</i>	Invoegen van een element met verplichte kinderen
<i>Taak 3:</i>	Invoegen van een element met vaste volgorde
<i>Taak 4:</i>	Invoegen van elkaar-uitsluitende elementen
<i>Taak 5:</i>	Invoegen van enkelvoudig toegestane elementen
<i>Taak 6:</i>	Invoegen van elementen met aangemoedigde attributen
<i>Taak 7:</i>	Invoegen van elementen met verplichte attributen

Omdat taak 1 enkel een introductietaak was, wordt deze binnen de resultaten verder buiten beschouwing gelaten. Alle lijsten met resultaten zullen dus beginnen bij testtaak 2.

2.1. Slagen van de taken

Een eerste indicator voor de gebruiksvriendelijkheid van een ontworpen patroon is of de bijbehorende testtaak kan worden voltooid. In de tabel hieronder wordt per deelnemer voor iedere taak getoond of dit is gelukt. Hierin worden beide ontwerpen weergegeven, zodat al een eerste indruk kan worden gevormd welke van de alternatieven het meest succesvol is geweest tijdens de tests. Sommige taken konden niet slagen: ook dit is in de tabel aangegeven. Voor iedere taak zijn in de eerste rij de resultaten voor ontwerp A opgenomen, en in de tweede rij de resultaten voor ontwerp B.

Slagen van testtaken				
	Tineke	Dave	Lillian	Rindert
<i>Taak 2 - modal</i>	Ja	Ja	Ja	Ja

<i>Taak 2 - template</i>	Ja	Ja	Ja	Ja
<i>Taak 3 - panel</i>	Nee	Nee	Nee	Ja
<i>Taak 3 - auto</i>	Ja	Ja	Ja	Ja
<i>Taak 4 - convert</i>	Ja	Ja	Ja	Ja
<i>Taak 4 - radio</i>	Ja	Nee	Ja	Ja
<i>Taak 5 - disable</i>	<i>kan niet slagen</i>			
<i>Taak 5 - remove</i>	<i>kan niet slagen</i>			
<i>Taak 6 - popover</i>	Ja	Ja	Ja	Ja
<i>Taak 7 - modal</i>	Ja	Ja	Ja	Ja
<i>Taak 7 - in trigger</i>	Ja	Ja	Ja	Nee

Tabel 1: Voltooiing per gebruiker bij het uitvoeren van de taken.

2.2. Begrip van de situatie

Een tweede belangrijke indicator van de gebruiksvriendelijkheid is begrip. Gebruikers moeten voor, tijdens en na het uitvoeren van een bepaalde taak goed begrijpen wat er aan de hand is. Door onbegrip te voorkomen, kan namelijk ook frustratie worden voorkomen. Daarnaast kunnen gebruikers in de toekomst gemakkelijker met de applicatie werken als zij bepaalde situaties leren begrijpen. In de tabel staat beschreven of deelnemers na het uitvoeren van iedere taak konden benoemen met welke situatie zij te maken hadden. Ook staat bij sommige taken aangegeven dat het begrip van de situatie niet te beoordelen is. Als laatste was het bij een aantal taken zo dat deelnemers achteraf de juiste situatie benoemden, maar hierbij geen zekerheid hadden. Ook dit staat in de tabel aangegeven.

Begrip van de situatie				
	Tineke	Dave	Lillian	Rindert
<i>Taak 2 - modal</i>	Ja	Nee	Nee	Ja
<i>Taak 2 - template</i>	Nee	Nee	<i>Niet te beoordelen</i>	<i>Niet te beoordelen</i>
<i>Taak 3 - panel</i>	Nee	Ja	Nee	Nee

<i>Taak 3 - auto</i>	Ja, maar twijfelt	Ja	Ja, maar twijfelt	Ja, maar twijfelt
<i>Taak 4 - convert</i>	Ja	<i>Niet te beoordelen</i>	Ja	Nee
<i>Taak 4 - radio</i>	Ja	Nee	Ja, maar twijfelt	Ja
<i>Taak 5 - disable</i>	Ja	Ja, maar twijfelt	Ja	Ja
<i>Taak 5 - remove</i>	Ja, maar twijfelt	Ja, maar twijfelt	Ja, maar twijfelt	Nee
<i>Taak 6 - popover</i>	Ja	Ja	Ja	Ja
<i>Taak 7 - modal</i>	Ja	Ja	Ja	Ja
<i>Taak 7 - in trigger</i>	Nee	Ja	<i>Niet te beoordelen</i>	Nee

Tabel 2: Volledig begrip van de situatie na het uitvoeren van de taken

3. Opvallende observaties

In de samengevatte resultaten in het vorige hoofdstuk werd een overzicht gevormd. Hierin kwamen bijzondere, opvallende resultaten echter niet naar boven. In dit hoofdstuk worden deze resultaten uitgelicht, zodat zij niet verloren gaan. In overleg met het team van FontoXML kunnen bepaalde van deze resultaten worden meegenomen in de conclusies of verbeterpunten.

3.1. Algemene observaties

In deze paragraaf worden observaties uitgewerkt die niet per se gekoppeld zijn aan een van de ontwerpproblemen. Het gaat dus om meer algemene observaties die bijvoorbeeld te maken hebben met zoekgedrag of toetsenbordgedrag.

Rechtermuisknopmenu

Tijdens het uitvoeren van de test is er door twee gebruikers meerdere malen gebruik gemaakt van de rechter muisknop. Na achteraf te vragen waarom mensen dit deden, bleek dat zij hier zochten naar een manier om een element (link en resultaat) aan te maken. Omdat hier geen rekening mee was gehouden in het prototype leverde dit echter niets op. Desalniettemin zou het (rechtermuisknop-) contextmenu wellicht een logische plek zijn om bepaalde aanmaaktriggers te plaatsen.

Zoekgedrag

Er lijkt, bij het zoeken naar bepaalde knoppen, niet erg goed te worden gelezen wat het label van een dropdown is. Eerder wordt er gewoon van links naar rechts of andersom te worden gezocht. Het kwam tijdens de tests bij alle vier de deelnemers voor dat men de dropdowns een voor een opende, in plaats van actief te lezen en te zoeken.

3.2. Invoegen van een element met verplichte kinderen - taak 2

Modal (ontwerp A)

Placeholders (ontwerp B)

- In het prototype wordt geselecteerde tekst niet meegenomen naar het aangemaakte hazard statement. Alle gebruikers hebben eerst de tekst geschreven, vervolgens deze geselecteerd en dan een hazard aangemaakt. Het zou efficiënt kunnen zijn om de geselecteerde tekst mee te nemen naar bijvoorbeeld het eerste kind-element. Gevaar hierbij is echter dat de geschreven tekst wellicht niet aansluit bij dit eerste kind.
- Er ontstaat een enkele keer verwarring over de placeholders door de beperkte witruimte tussen de kind-elementen.

- Door slecht opgestelde aandachtspunten is hier *slecht te beoordelen* of gebruikers daadwerkelijk de situatie begrijpen.

3.3. Invoegen van een element met vaste volgorde - taak 3

Suggestiepaneel (ontwerp A)

- Het vinden van het suggestiepaneel was veel te moeilijk voor de deelnemers. Bijna niemand keek in de breadcrumbs. Meestal werd er gezocht binnen het rechtermuisknopmenu of naar een manier om de toolbarknoppen weer actief te laten worden.

Automatisch via toolbar (ontwerp B)

- Niemand (!) leest het verklarende systeembericht. Dit komt doordat men zoekt naar de plaats waar het nieuwe element geplaatst is. De focus ligt dus niet boven in het scherm, waardoor het bericht gemist wordt.
- Nagenoeg iedereen probeert het resultaat na de postreq te plaatsen. Achteraf blijkt dit te zijn omdat niemand begrijpt wat een postreq is, en iedereen denkt dat een resultaat aan het einde van een task zou moeten staan.

3.4. Invoegen van elkaar-uitsluitende elementen - taak 4

Uitschakelen & convert (ontwerp A)

- Slechts een deelnemer leest de tooltip.
- De meeste mensen begrijpen de situatie, ondanks het feit dat er redelijk onlogische elementen voor zijn gebruikt.

Radio buttons (ontwerp B)

- Er wordt meerdere malen aangegeven dat het vreemd voelt dat er opeens radiobuttons in een dropdown staan.

3.5. Invoegen van enkelvoudig toegestane elementen - taak 5

Knop uitschakelen (ontwerp A)

- Bij het uitschakelen van de knoppen proberen veel deelnemers alsnog om op de knop te drukken.

Alternatieve knopfunctie (ontwerp B)

- Wanneer het label verandert van “Result” naar “Remove result” wordt de knop volledig over het hoofd gezien door veel deelnemers. Omdat men kennelijk snel scant naar de juiste knop wordt de knop dus simpelweg overgeslagen.

3.6. Invoegen van elementen met aangemoedigde attributen - taak 6

Popovers (ontwerp A)

- Iedereen gaat in op de aanmoedigingen.
- Er wordt aangegeven dat dit als “logische dingen” voelen

3.7. Invoegen van elementen met verplichte attributen - taak 7

Modal (ontwerp A)

- Iedereen doorloopt de modal zonder problemen.
- Er ontstaan geen invalide situaties tijdens de tests.

Invoer in toolbar verwerken (ontwerp B)

De URL-invoer is niet breed genoeg. Bij het typen ontstaat er overflow.

4. Conclusies per ontwerpprobleem

4.1. Elementen met verplichte kinderen

Voor dit ontwerpprobleem zijn er twee ontwerpen getest: een concept met invoer via een modal en een concept waarbij kinderen automatisch worden geplaatst en placeholders krijgen. Lettend op het aantal kinderen dat inhoud kreeg leverde het concept met de modal een beter resultaat op. Dit is een belangrijk punt, maar geen must voor dit ontwerpprobleem: het hebben van inhoud is namelijk niet verplicht. Door het element direct te plaatsen, zoals in het andere concept, wordt de gebruiker minder verstoord in zijn werk. Ook hier wordt nog in de helft van de gevallen inhoud gegeven aan alle verplichte kind-elementen. Om deze reden heeft ontwerp B de voorkeur voor implementatie in FontoXML.

Beste ontwerp: *Direct plaatsen met placeholders (ontwerp B)*

1. De werking van de placeholders is gemakkelijk te begrijpen danwel ontdekken voor de deelnemers. Slechts één deelnemer komt hier niet uit tijdens de eerste testtaak door een bug in het prototype. Bij de volgende taak waar hij een placeholder tegenkwam ontdekte hij alsnog snel de werking.
2. De modal is te opdringerig voor dit doeleinde. De gebruiker wordt “opgesloten” in een actie waar dit niet nodig is.

Verbeterpunten:

- Slechts de helft van de gebruikers vulde de placeholders voor de kinderen in, terwijl dit relatief makkelijke elementen waren. Het zou wenselijk zijn als dit een groter percentage zou zijn.
- Wellicht is het ook hier wenselijk dat gebruikers de situatie kunnen benoemen.

4.2. Elementen die enkelvoudig zijn toegestaan

Voor elementen die enkelvoudig zijn toegestaan waren er twee alternatieven. Bij het eerste ontwerp werd de knop uitgeschakeld, bij het tweede ontwerp kreeg de knop een andere functie (element verwijderen). Qua efficiëntie leek het tweede ontwerp dus het meest aantrekkelijk: in plaats van een knop die de helft van de tijd niets doet, zou deze altijd een functie hebben. Tijdens de tests leverde dit echter nogal wat problemen op. Doordat het label van de knop opeens “Remove result” was in plaats van “Result”, kon men deze niet meer vinden. Bij het scannen naar het woord “result” werd het woord “remove” namelijk meestal overgeslagen. In de gevallen dat deze knop wel bekeken werd, las men de tooltip niet.

Bij het uitschakelen van de knop werd de situatie veel vaker begrepen omdat de tooltip vaker werd gelezen. Omdat het verwijderen van een element in de editor ook zeer gemakkelijk via het document kan, is de extra functionaliteit van ontwerp B van weinig waarde. Begrip van de situatie weegt hier dan ook zwaarder en om die reden is ontwerp A de meest gebruiksvriendelijke.

Beste ontwerp: *knop uitschakelen & verklarende tooltip (ontwerp A)*

1. De uitgeschakelde knop werd veel vaker gevonden door de deelnemers. Hierdoor werd er minder lang tevergeefs naar een andere knop gezocht en werd de situatie eerder begrepen.
2. Gebruikers proberen alsnog op uitgeschakelde knoppen te klikken. Het tonen van de tooltip bleek hierbij een zeer behulpzame actie.

Verbeterpunten

- Idealiter zou de gebruiker vooraf al direct kunnen herkennen dat een element enkelvoudig is toegestaan. Zo zou een fout voorkomen worden in plaats van dat hij achteraf wordt uitgelegd. Hiervoor zou de trigger (in dit geval de toolbarknop) onderscheidend moeten zijn van andere triggers. Hiervoor zou bijvoorbeeld een icoon of een tekstindicatie (Max 1) aan de trigger kunnen worden toegevoegd. In de praktijk is dit echter lastig, omdat knoppen vaak al andere iconen bevatten en het gebruik van iconen consistent door de applicatie moet worden toegepast.

4.3. Elementen met een vaste volgorde:

Voor het plaatsen in een vaste volgorde is gebleken dat het goed werkt om dit automatisch te doen, ook vanuit de toolbar. Als de knoppen van de elementen in dezelfde volgorde worden geplaatst als de elementen dienen te staan, biedt dit ook een hulpmiddel.

Het zijpaneel is duidelijk, maar kan niet goed worden gevonden door gebruikers. Misschien is het broodkruimelpad meer iets voor ervaren gebruikers of misschien is het gewoon geen logische plek om te zoeken. In ieder geval moet de vindbaarheid worden verhoogd. Daarnaast is het moeilijk om te bepalen wat er wanneer in het paneel moet komen te staan. De structuursuggesties lijken eerder onderdeel te zijn

Beste ontwerp: *automatisch op juiste volgorde invoegen (ontwerp B)*

1. Het helpt om de volgorde van de knoppen in de toolbar gelijk te houden aan de gewenste volgorde van elementen in het document. Meerdere testpersonen koppelden deze volgordes aan elkaar en herkenden zo de gewenste structuur.
2. Uit het feit dat een element op een andere plaats dan verwacht wordt geplaatst, maken veel mensen al op dat er sprake is van een vaste volgorde. Het systeembericht zou dit

vermoeden echter moeten bevestigen, maar deze wordt zelden gelezen. Door dit te verbeteren, zou alle twijfel weggenomen kunnen worden.

Verbeterpunten

- Het systeembericht waarin de volgorde wordt uitgelegd, wordt zelden gezien. Doordat mensen in het scherm zoeken naar het nieuwe element dat zij zojuist hebben geprobeerd te plaatsen, letten zij totaal niet op de systeemberichten boven- of onderin het scherm
- Slechts één gebruiker kan met zekerheid de situatie benoemen. De rest kan dit niet (met zekerheid). Om frustratie te voorkomen is het belangrijk dat dit wordt verbeterd voordat

4.4. Elkaar uitsluitende elementen

Gebruikers geven meerdere malen aan dat het vreemd voelt dat er radio buttons in de toolbar staan. Helemaal als deze ook nog eens een directe actie blijken uit te voeren. Wel wordt de situatie met dit ontwerp sneller inzichtelijk. Doordat er echter nog meer verwarring wordt opgeroepen biedt dit ontwerp over het geheel een lagere gebruiksvriendelijkheid.

De expliciete converteeractie biedt meer duidelijkheid. In deze knop wordt direct beschreven wat er gaat gebeuren. Hierdoor sluit de verwachting van de deelnemers vooraf beter aan bij de daadwerkelijke situatie. Desalniettemin wordt de tooltip maar één keer gelezen en kan de situatie dus zelden met volledige zekerheid worden beschreven.

Beste ontwerp: *Uitschakelen en converteren*

1. Ondanks het feit dat de testcase een beetje vreemd is (waarom sluit een beschrijving een samenvatting uit?), begrijpen de meeste gebruikers de situatie. Ook het omzetten van het element wordt uiteindelijk altijd gevonden.

Verbeterpunten

- Het ontwerp zou nog verbeterd kunnen worden door te zorgen dat gebruikers met zekerheid kunnen benoemen wat de situatie is.

4.5. Elementen met aangemoedigde attributen

De popovers voor aangemoedigde attributen doen precies wat zij moeten doen. De aangemoedigde acties (taal invullen of linkadres invullen) worden door alle testpersonen moeiteloos uitgevoerd. Voor twee van de testpersonen levert het zelfs een vreugdemoment op: *“Wat leuk dat je ook de taal kunt aangeven!”*. De aanmoediging wordt door geen van testpersonen als storend ervaren. Als kanttekening moet hierbij worden gegeven dat dit geen beeld geeft over de lange termijn: het is denkbaar dat de popovers na enige tijd wel storend worden, maar dit is niet binnen de termijn van dit project te testen.

Voor dit probleem is maar één oplossing ontworpen

1. De tooltips werken erg goed voor het aanmoedigen van attributen.
2. Niemand heeft de actie “gecanceled”. Dit komt waarschijnlijk ook door de relatief eenvoudige taken.

4.6. Elementen met verplichte attributen

Het ontwerp met de modal is universeel inzetbaar: vanwege de flexibiliteit in grootte kan UI voor de invoer van allerlei soorten data in een modal worden opgenomen. Dit is een groot voordeel ten opzichte van ontwerp B. Een duidelijk voorbeeld hiervan is het image element, waarbij een verplichte source moet worden opgegeven. Hiervoor is een browse-interface nodig, die niet in de toolbar kan worden verwerkt.

Beste ontwerp: *invoer via modal (ontwerp A)*

5. Verbetervoorstellen

Voor sommige ontwerpen kan op basis van de testresultaten een verbetervoorstel worden gedaan. Soms is dit voor een specifiek ontwerp, soms is dit voor een concept dat in meerdere ontwerpen meegenomen is. In het adviesrapport zal worden aangegeven of een ontwerp verbeterd moet worden voordat het geschikt is voor gebruik in FontoXML, of dat ook de onverbeterde versie al voldoende gebruiksvriendelijk is.

Elementen met een vaste positie

Een van de opvallende punten voor het ontwerp van elementen op een vaste positie was dat systeemberichten bij het aanmaken van elementen niet worden gelezen. Dit komt omdat gebruikers volledig focussen op het document: zij willen zien waar hun element geplaatst is. Dit zou verbeterd kunnen worden door het systeembericht:

1. Een meer opvallende styling te geven.
2. Dicht bij het geplaatste element weer te geven, zodat deze beter in het oog valt.

Uitgeschakelde knoppen

Bij uitgeschakelde knoppen moet, wanneer daar op wordt geklikt, de tooltip verschijnen. Als de tooltip er al is, moet er focus op de tooltip worden gelegd. Zo wordt de aandacht van de gebruiker nog meer verlegd naar de uitleg.

Enkelvoudig toegestane elementen

Voor enkelvoudig toegestane elementen zou het wenselijk zijn als gebruikers dit al kunnen herkennen voordat zij deze proberen aan te maken. Uit de tests is geen logische oplossing gebleken, dus deze zal ontworpen moeten worden.

Elkaar uitsluitende elementen

Ook voor elkaar uitsluitende elementen zou het wenselijk zijn dat gebruikers de situatie alvorens het aanbrengen zouden kunnen herkennen. Hiervoor is nog geen logische oplossing uit de test ontstaan.

Elementen met verplichte kinderen

Het omgaan met selectie bij elementen met verplichte kinderen heeft nog denkwerk nodig. In veel gevallen is het zo dat elementen met verplichte kinderen zelf geen tekst bevatten. Je zou de selectie dan kunnen meenemen naar het eerste kind, maar hierdoor krijgt de gebruiker de placeholder niet te zien. De content zou dus verkeerd kunnen zijn voor dit eerste kind. Het is dus een gevaarlijke

aanname om vooraf te doen. De kans is erg groot dat de tekst die gebruikers vooraf schrijven niet toepasselijk is voor het element waarnaar deze wordt overgenomen bij selectie. In het geval van een template is het een acceptabel alternatief om de gebruiker zijn tekst achteraf te laten kopiëren naar het nieuwe element. Dit kost meer tijd, maar voorkomt wel een hoop fouten in elementinhoud.

6. Verbeterpunten test

In dit hoofdstuk staat een aantal aspecten van de test die volgende keer moeten worden verbeterd. Dit hoofdstuk is opgesteld om te voorkomen dat er in de toekomst nog gelijksoortige fouten worden gemaakt.

Actief vragen naar aandachtspunten

Bij een aantal van de aandachtspunten was het nodig om na het uitvoeren van een bepaalde taak vragen te stellen aan de deelnemer. Dit is niet altijd gebeurd, waardoor bepaalde aandachtspunten soms niet beoordeeld konden worden. Hier is dan ook kostbare informatie verloren gegaan. Dit kan in de toekomst worden voorkomen door in het moderatorscript duidelijke ankerpunten op te nemen voor het stellen van specifieke vragen.

Kwaliteit van de aandachtspunten

Bij sommige taken zijn de aandachtspunten niet goed opgesteld. Bij taak 1 is bijvoorbeeld een van de aandachtspunten *“ziet de gebruiker waar het element is ingevoegd”*. Dit zegt echter niets over het feit dat er een element met verplichte kinderen is geplaatst. Het is belangrijk dat een test bruikbare resultaten oplevert. Er moet dus zorgvuldiger worden gewerkt bij het opstellen van deze aandachtspunten.

Meetbaarheid van de aandachtspunten

Er is bewust gekozen voor het opstellen van kwalitatieve aandachtspunten. Achteraf lijkt het echter toch belangrijk om deze aandachtspunten zo meetbaar mogelijk te maken. Probeer bij testtaken dus na te denken of factoren als doorlooptijd of het aantal foute clicks mogelijke indicatoren zijn voor de gebruiksvriendelijkheid van een ontwerp.

Representativiteit van het prototype

Hoe meer een prototype werkt zoals het daadwerkelijke product, hoe beter. Voor deze specifieke test is gekozen om een nieuw prototype te bouwen in plaats van een aftakking van de echte applicatie. Het was hier dan ook niet mogelijk geweest om binnen de tijdsgrenzen van het project een representatieve applicatie neer te zetten. Bij het uitvoeren van de test wijken deelnemers echter heel vaak af van de *“ideale route”* die jij als ontwerper hebt bedacht. Wanneer het prototype representatief was geweest voor de echte applicatie, had hier veel lering uit getrokken kunnen worden. Dit was geen vereiste geweest voor het testen van de ontwerpen, maar had wel veel nuttige bijresultaten op kunnen leveren.

Bijlage A: Aandachtspunten per taak

Voor iedere testtaak is een checklist opgesteld. In deze checklist stonden verschillende aandachtspunten die bij het uitvoeren van de test beoordeeld konden worden. Om snel te laten zien hoe elk ontwerp gescoord heeft op deze punten, is in deze paragraaf per taak een overzicht gemaakt.

Invoegen van een element met verplichte kinderen - taak 2

Aandachtspunt	Tineke	Dave	Lillian	Rindert
Ontwerp A - Modal				
Wat beschrijft de gebruiker te zien nadat het element is ingevoegd?	Gevarenbeschrijving	<i>Niet te beoordelen</i>	<i>Niet te beoordelen</i>	Het is gelukt.
Snappen gebruikers waarom er meerdere elementen zijn ingevoegd?	<i>Niet te beoordelen</i>	Ja	Ja	<i>Niet te beoordelen</i>
Begrijpt de gebruiker de invoervelden?	Ja	Ja	Ja	Ja
Overig				Vult de kinderen niet in
Ontwerp B - Template				
Wat beschrijft de gebruiker te zien nadat het element is ingevoegd?	Element is geplaatst	"heel nieuw element"	<i>Niet te beoordelen</i>	Een gevaar met onderdelen
Snappen gebruikers waarom er meerdere elementen zijn ingevoegd?	<i>Niet te beoordelen</i>	<i>Niet te beoordelen</i>	Ja	<i>Niet te beoordelen</i>
Begrijpt de gebruiker de placeholders?	Ja	Nee	Ja	Ja
Overig	Denkt dat er een koppeling is tussen tekst en hazstat..	Vult niet alle kinderen in.	Vult niet alle kinderen in.	

Invoegen van een element met vaste volgorde - taak 3

Aandachtspunt	Tineke	Dave	Lillian	Rindert
Ontwerp A - Panel				
Vindt de gebruiker de Task?	Ja	Ja	Ja	Ja
Waar gaat hij met de cursor staan?	onder postreq	onder postreq	onder postreq	onder postreq
Ziet hij waar het element is ingevoegd?	<i>Taak niet geslaagd</i>	<i>Taak niet geslaagd</i>	<i>Taak niet geslaagd</i>	Ja
Kan de gebruiker benoemen wat er gebeurd is?	/	/	/	Ja.
Ontwerp B - Auto				
Vindt de gebruiker de Task?	Ja	Ja	Ja	Ja
Waar gaat hij met de cursor staan?	onder postreq	onder postreq	onder postreq	onder postreq
Ziet hij waar het element is ingevoegd?	Ja	Ja	Ja	Ja
Ziet hij de feedback message?	Nee	Nee	Nee	Nee
Kan de gebruiker benoemen wat er gebeurd is?	Ja, maar twijfelt	Ja	Nee	Ja, maar twijfelt

Invoegen van elkaar-uitsluitende elementen - taak 4

Aandachtspunt	Tineke	Dave	Lillian	Rindert
Ontwerp A - Convert				
Ziet de gebruiker de verandering in de toolbar?	Ja	Ja	Ja	Ja
Kan de gebruiker benoemen wat er aan de hand is?	Ja, maar twijfelt	Ja	Ja	Nee
Ziet de gebruiker de tooltips?	Nee	Nee	Ja	Nee
Overig	Mist dat topic geen focus heeft.	/	Leest de tooltips!	Gek dat dat in een andere knop staat
Ontwerp B - Radio				
Ziet de gebruiker de verandering in de toolbar?	Ja	Ja	Ja	Ja
Kan de gebruiker benoemen wat er aan de hand is?	Ja	Nee	Ja	Ja
Ziet de gebruiker de tooltips?	Nee	Nee	Nee	Nee
Overig	Geeft aan dat de radio's "verwarrend" zijn		Vindt de radio's raar.	

Invoegen van enkelvoudig toegestane elementen - taak 5

Aandachtspunt	Tineke	Dave	Lillian	Rindert
Ontwerp A - Disable				
Begrijpt de gebruiker vooraf al dat het niet kan?	Nee	Nee	Nee	Nee
Leest de gebruiker de tooltip?	Ja	Ja	Nee	Ja
Begrijpt de gebruiker achteraf dat het niet kan en waarom?	Ja	Ja, maar twijfelt	Ja	Ja
Overig				
Ontwerp B - Remove				
Begrijpt de gebruiker vooraf al dat het niet kan?	Nee	Nee	Nee	Nee
Leest de gebruiker de tooltip?	Nee	Nee	Nee	Nee
Begrijpt de gebruiker achteraf dat het niet kan en waarom?	Ja	Ja, maar twijfelt	Ja	Nee
Overig	/	/	/	/

Invoegen van elementen met aangemoedigde attributen - taak 6

Aandachtspunt	Tineke	Dave	Lillian	Rindert
Ontwerp A - Popover				
Gaat de gebruiker in op de aanmoedigingen?	Ja	Ja	Ja	Ja
Wat vindt de gebruiker van de popovers?	Niet storend	<i>Niet beoordeelbaar</i>	Makkelijk	Logisch
Overig	/	/	/	/

Invoegen van elementen met verplichte attributen - taak 7

Aandachtspunt	Tineke	Dave	Lillian	Rindert
Ontwerp A - Modal				
Wordt de attribuutwaarde ingevuld?	Ja	Ja	Ja	Ja
Begrijpt de gebruiker wat hij moet doen?	Ja	Ja	Ja	Ja
Overig	Ideale pad	Ideale pad	Ideale pad	/
Ontwerp B - In trigger				
Wordt de attribuutwaarde ingevuld?	Ja	Ja	Ja	Nee
Begrijpt de gebruiker wat hij moet doen?	Ja	Ja	Ja	Nee
Overig				

Bijlage B: Uitwerking resultaten

In dit hoofdstuk zijn de resultaten van de verschillende gebruikerstests uitgewerkt.

1. Deelnemer: Tineke

1.1. Invoegen van een element met verplichte kinderen

Geschreven verslag

De testpersoon maakt eerst met een dubbele enter wat ruimte om een nieuw stuk tekst te typen. Na het typen stopt zij even. Ik vraag haar of ze klaar denkt te zijn en zij antwoordt met “nee, nu moet ik het gaan markeren”. Er wordt ongeveer 30 seconden gezocht in de verschillende dropdowns en uiteindelijk wordt de juiste knop gevonden. Deze wordt ingedrukt en de Hazard Statement verschijnt. Ze probeert de placeholder tekst te selecteren en komt erachter dat klikken voldoende is. Uiteindelijk worden ook beide placeholders ingevuld en geeft de testpersoon aan klaar te zijn.

Aandachtspunten

Aandachtspunt	Commentaar
Wat beschrijft de gebruiker te zien nadat het element is ingevoegd?	Dat het element onder haar tekst is geplaatst
Snappen gebruikers waarom er meerdere elementen zijn ingevoegd?	Merkt de gebruiker niet bewust op
Begrijpt de gebruiker de placeholders?	Probeert de placeholder tekst te selecteren maar ontdekt binnen 4 seconden hoe het werkt.
Overig	Gaat eerst typen Denkt dat er een soort koppeling is omdat ze het gemarkeerd had toen ze de hazstat aanmaakte

1.2. Invoegen van een element met een vaste volgorde

Geschreven verslag

De gebruiker kijkt of er in de dropdown “task” een overzichtje staat met alle taken, dat blijkt niet zo te zijn. Vervolgens gaat zij scrollen in het document. De task wordt herkend aan het label. Ik vraag

haar om een resultaat toe te voegen aan de taak. *“Dan moet mijn Engels weer goed zijn.”* Zij geeft vervolgens een enter na de postreq. Door een gebrek in het prototype verschijnt er een tweede postreq, waar dat eigenlijk niet de bedoeling is. De gebruiker negeert dit en gaat op zoek naar result. Na snel de structure en de semantics dropdown te scannen komt zij bij de task-dropdown. Vindt daar result en voegt deze toe. *“Hee, maar hij komt op een heel andere plek dan ik met mijn dingetje stond.”* - Doelt hiermee op de cursor. De message die de hele boel uitlegt ziet zij niet. *“Oh oh, ik wilde hem daar hebben, omdat het op het laatst moet staan.”* Merkt dus op dat het result op een andere positie kwam. Ik vraag haar of ze begrijpt waarom dit gebeurde. *“Misschien omdat er een volgorde is.”* Zij bevestigt dit door te kijken naar de volgorde in de task-dropdown.

Aandachtspunten

Aandachtspunt	Commentaar
Vindt de gebruiker de Task?	Ja
Waar gaat hij met de cursor staan?	Na postreq, verwacht een resultaat als laatste.
Ziet hij waar het element is ingevoegd?	Ja
Ziet hij de feedback message?	Nee
Kan de gebruiker benoemen wat er gebeurd is?	Ja, maar niet met zekerheid. Baseert deze opmerking op de dropdown voor Task.
Overig	Noemt al dat ze geen result meer kan aanmaken omdat er al een in staat.

1.3. Invoegen van elkaar uitsluitende elementen

Geschreven verslag

Als eerst wordt er gezocht in de dropdowns. Reference & Semantics worden gescand, en bij structure zegt de gebruiker *“Structure”*, met een toon van: die moet ik hebben. Dat is hem echter niet, dus ze gaat terug naar semantics. Zet de element labels uit. Blijft heen en weer klikken en geeft aan dat het niet lukt. *“Oh, ik ben echt in de war!”* Ontdekt, vlak voordat ik wil helpen, dat het topic geen focus heeft, en dat dat wel moet voordat de topic dropdown actief wordt. Kijkt nu in de dropdown voor topic en hovert daar een aantal keer over de abstract en de shortdesc dropdown. *“Oh convert”*, en klikt vervolgens op convert to abstract. Zag dat je kunt kiezen voor OF een description OF een abstract. Maakt dat op uit het disabled zijn van de knoppen. Leest geen tooltip.

Aandachtspunten

Aandachtspunt	Commentaar
Ziet de gebruiker de verandering in de toolbar?	ja, maar begrijpt het niet helemaal.
Kan de gebruiker benoemen wat er aan de hand is?	ja, maar niet met zekerheid: omdat de ander grijs wordt
Ziet de gebruiker de tooltips?	nee: worden wel opgeroepen maar niet gelezen
Overig	Heeft niet door dat topic disabled is omdat ze er niet in staat met de cursor

.4. Invoegen van enkel toegestane elementen

Geschreven verslag

De gebruiker begint met het scrollen naar de task. Voor het aanmaken van een tweede result gaat zij weer zoeken in de task dropdown. Blijft met de cursor in de titel van de task staan omdat zij benieuwd is waar de result dan terecht zal komen. Kan in de dropdown geen result knop meer vinden en navigeert daarom naar de result in de content. Klik 6 keer op het rechter uiteinde van de placeholder voor result omdat ze daarna een enter wil gaan geven. Ziet uiteindelijk dat de cursor voorin komt te staan. Geeft een enter en maakt, door een bug in het prototype, een tweede resultaat aan. *“Ik zie hier alleen maar remove”* - trekt daar nog geen conclusies uit. *“Misschien via een bulleted list?”*. Probeert dat. Door het prototype gebeurt er niets. Geeft aan de knop bij result te willen zien. Heeft de tooltip met uitleg ongeveer 4 seconden open.

Aandachtspunten

Aandachtspunt	Commentaar
Begrijpt de gebruiker vooraf al dat het niet kan?	Nee, gaat zoeken naar een mogelijkheid
Leest de gebruiker de tooltip?	Heeft hem 4 seconden open maar leest niet
Begrijpt de gebruiker achteraf dat het niet kan?	Ja, maar wel na erg veel tijd
Overig	Probeert nog iets met een bulleted list

1.5. Invoegen van elementen met aangemoedigde attributen

Geschreven verslag

Voor het invoegen van de link wordt eerst de tekst getypt. Vervolgens wordt deze geselecteerd en klikt de gebruiker op rechtermuisknop. Vindt hier niets. “Dan moet het waarschijnlijk via de toolbar.” Eerst naar de dropdown structure, dan topic. Vindt hier “related links”. Vraag zich hardop af of dit de juiste is en klikt hem toch maar aan. Omdat er niets gebeurt, neemt zij aan dat dit niet de juiste was. Zoekt door in “semantics”. Gaat uiteindelijk naar reference. “En link is gewoon link he?”. Ik zwijg. “Oh web address!” klikt op de knop. Typt de URL in en klikt op confirm. Probeert nog op de link te klikken. Herkent uit de mouseover dat de URL wordt getoond. Probeert nog met RMB om de URL te kunnen wijzigen: niet geprogrammeerd in dit prototype.

Selecteert de Franse tekst. Klikt op Foreign text (zonder te zoeken) en de popover verschijnt. Deze valt echter onder de toolbar, waardoor niet alles te lezen is. Na 10 seconden scrollen door de popover navigeert zij omhoog in het document, zodat de hele popover zichtbaar wordt. Klikt vervolgens op French, en dan op confirm. Maakt van haar zelf geschreven stukje tekst nog een foreign element met het Dutch attribuut. “*Nu ga ik echt helemaal los*”.

Aandachtspunten

Aandachtspunt	Commentaar
Gaat de gebruiker in op de aanmoedigingen?	Ja
Wat vind de gebruiker van de popovers?	Niet storend
Overig	Doet lang over het vinden van de link

1.6. Invoegen van elementen met verplichte attributen

Geschreven verslag

Gaat op zoek in semantics “*Die had ik hier ergens gezien*”. Klikt op de knop voor trademark, selecteert reg en klikt op confirm. Het ideale pad voor deze actie. Kan bevestigen dat de actie is gelukt door de R achter het woord.

Aandachtspunten

Aandachtspunt	Commentaar
Wordt de attribuutwaarde ingevuld?	Ja
Begrijpt de gebruiker wat hij moet doen?	Ja
Overig	Wordt zonder moeite uitgevoerd

2. Deelnemer: Dave

2.1. Invoegen van een element met verplichte kinderen

Geschreven verslag

Dave gaat aan het einde van het document staan en drukt daar op enter. Hij typt hier de tekst voor de gevarenbeschrijving en selecteert deze. In eerste instantie begrijpt hij niet dat er een expliciet hazardstatement element was. Hij wil zijn tekst dus als titel gaan markeren en gaat daarnaar opzoek. Hij wil dit doen zodat zijn stukje als apart ding vindbaar zou zijn in een CMS. Klik hierbij door de tabs. Noemt Start de “opmaak” omgeving en Insert de omgeving voor “het toekennen van bepaalde functies aan tekst”. Kan nergens een knop voor titel vinden. Klik vervolgens op “navigation title” in de hoop dat dit is wat hij zoekt. Er gebeurt niets, ik geef aan dat dit door het prototype komt. Ik geef een hint dat er een speciaal element is voor een gevaar. Hij voegt dit vervolgens in en verwijdert de eerder geschreven content. Merkt de eerste placeholder niet op omdat deze bij het deleten van content verdwijnt zodra de cursor erin komt (*bug!*). Hierdoor ontstaat verwarring over de functie van de placeholders. Gaat boven de tweede placeholder zitten typen en hij verdwijnt niet en dat snapt hij niet.

Aandachtspunten

Aandachtspunt	Commentaar
Ziet de gebruiker waar het element is ingevoegd?	Ja
Wat beschrijft de gebruiker te zien nadat het element is ingevoegd?	“Een heel nieuw element”
Snappen gebruikers waarom er meerdere elementen zijn ingevoegd?	Valt hem niet op.
Begrijpt de gebruiker de placeholders?	Nee
Overig	Niet alle kinderen worden ingevuld.

2.2. Invoegen van een element met vaste volgorde

Geschreven verslag

Navigeert naar de task. Zet de cursor aan het einde van de postreq en geeft daar een enter. Er ontstaat door bug in het prototype een tweede postreq. Dit is niet wat hij wilde en deze verwijdt hij dus weer. Probeert shift-enter en ziet dat er dan niets gebeurt. Vervolgens zet hij toch een nieuwe postreq neer en gaat door de toolbar zoeken. Hij scant in deze volgorde de volgende dropdowns: reference, topic (disabled), structure, semantics, reference, semantics, topic(disabled, probeert hij 11 keer achter elkaar), structure, semantics. Geeft aan dat dit probleem deels komt door de taalbarrière en dat dit een heikel punt gaat worden. Semantics, Reference, Semantics. Klikte dus in totaal **22!** keer op een verkeerde dropdown en slaat daarbij. Probeert example. Die doet het niet in dit prototype. Daardoor krijgt hij geen feedback of het gelukt is, maar dat klopt. Gaat nu opeens naar task, en vindt hem daar.

Voegt een result in. Ziet dat deze op een andere positie komt en verwijdt hem weer. Voegt deze nogmaals toe. *“Ik kan hem dus invoeren, en zie ook dat er een logische volgorde in het invoegen zit”*. Geeft aan dat er weer een aanwijzing komt te staan, *“en hier verdwijnt hij dus wel”*. Zet nu wel de cursor op de goede positie en typt daar de tekst. Bedenkt zich nu dat het net wellicht ook te maken had met de plek waar hij de cursor plaatsen. Dave heeft nu dus geleerd hoe de placeholders werken.

Aandachtspunten

Aandachtspunt	Commentaar
Vindt de gebruiker de Task?	Ja
Waar gaat hij met de cursor staan?	Einde postreq en maakt een nieuwe paragraaf. Verwacht de result dus onderin.
Ziet hij waar het element is ingevoegd?	Ja
Ziet hij de feedback message?	Nee
Kan de gebruiker benoemen wat er gebeurd is?	Ja, concludeert dit uit de gebeurtenis.
Overig	Moet 22 keer op een dropdown klikken voordat hij de task vindt. Dit komt wellicht door het feit dat Task een icoontje in de knop heeft en de rest niet.

2.3. Invoegen van elkaar uitsluitende elementen

Geschreven verslag

Begint nu met zoeken in Task, terwijl die disabled is. Probeert het weer 10 keer. Vindt het “vreemd” dat de task dropdown het niet meer doet. Vraagt zich af waarom dat zo is. Denkt daar te moeten zijn omdat hij bij de vorige opdracht daar zijn result had gevonden. Klikkt door naar semantics en structure, en komt uit bij topic. Ziet daar bij description staan: “Convert to abstract”. *“Dit vind ik ook wel weer een gevaarlijke omdat ik hem ga omzetten”*. Denkt dat ik heb gevraagd om “toe te voegen”. Heeft dat niet goed verstaan. Zet vervolgens de description om. Taak is dus voltooid.

Aandachtspunten

Aandachtspunt	Commentaar
Ziet de gebruiker de verandering in de toolbar?	Ja
Kan de gebruiker benoemen wat er aan de hand is?	Ja
Ziet de gebruiker de tooltips?	Nee
Overig	Er ontstaat twijfel omdat hij de vraagstelling niet goed heeft onthouden.

2.4. Enkel toegestane elementen

Geschreven verslag

In eerste instantie begrijpt Dave de vraag niet echt. Hij blijft in de topic en gaat zoeken waar hij nog een abstract kan toevoegen. Ik geef meerdere malen aan dat het gaat om de “task” in de content en daar scrollt hij vervolgens heen. Nogmaals zoekt hij naar een abstract. Ik geef aan dat er een tweede result moet worden ingevoegd.

Uiteindelijk navigeert hij terug naar de taak. De cursor wordt in de result geplaatst en er wordt gezocht naar een manier om result in te voegen. Er wordt een tijdje in task dropdown gekeken. *“De enige manier waarop ik denk dat het kan is door die andere te verwijderen”*. Denkt dat er geen twee kunnen omdat de knop voor insert result er niet meer staat. Zag nu pas de melding over de vaste volgorde. Verwijdert de bestaande en plaatst een nieuwe. Concludeert dat er dus geen twee kunnen bestaan. Maar weet dit niet zeker.

Aandachtspunten

Aandachtspunt	Commentaar
Begrijpt de gebruiker vooraf al dat het niet kan?	Nee.
Leest de gebruiker de tooltip?	Nee.
Begrijpt de gebruiker achteraf dat het niet kan?	Twijfel.
Overig	

2.5. Aangemoedigde attributen

Geschreven verslag

De gebruiker begint met het typen van het woord. Vervolgens selecteert hij ThiemeMeulenhoff, kijkt bij reference en vraagt of link hetzelfde is als een webadres. Voert hier een URL in en klikt op confirm.

Voor Foreign selecteert hij de tekst, gaat zoeken bij reference, topic, semantics. Gaat 8 verschillende dropdowns af omdat hij nu gewend is om daar te gaan zoeken. Vraagt zich af waarom de knop voor foreign en hazstat apart staan. Klikt op foreign, geeft aan welke taal het is en klikt op confirm.

Vraagt zich wel af of deze eigenschappen later nog te wijzigen zijn. Ik leg uit dat dit niet bij het prototype hoort, maar zeker moet gaan kunnen.

Aandachtspunten

Aandachtspunt	Commentaar
Gaat de gebruiker in op de aanmoedigingen?	Ja
Wat vind de gebruiker van de popovers?	<i>Niet gevraagd!</i>
Overig	

2.6. Verplichte attributen

Geschreven verslag

Gebruiker beloopt het ideale pad. Merkt op dat de enige visualisatie het feit is dat er een icoontje achter komt.

Aandachtspunten

Aandachtspunt	Commentaar
Wordt de attribuutwaarde ingevuld?	Ja
Begrijpt de gebruiker wat hij moet doen?	Ja
Overig	Zonder problemen uitgevoerd.

3. Deelnemer: Lillian

3.1. Invoegen van een element met verplichte kinderen

Geschreven verslag

Lillian begrijpt in eerste instantie de taak niet goed. Na uitleg maakt zij ruimte onderin het eerste topic door tweemaal een enter te geven. Zij typt het gevaar en maakt deze scheefgedrukt. Ik geef aan dat je dit echt als hazard kunt markeren. Vervolgens gaat ze op zoek naar de functie om hem te markeren. Ze selecteert de tekst en drukt op rechtermuisknop. Daar komt toevallig het woord “element” voor, dus drukt ze op inspect element. Ik grijp in omdat het prototype geen rekening heeft gehouden met RMB en verzoek haar binnen de toolbar te blijven. Leest eerst de tooltips van alle knoppen bij start. Gaat vervolgens naar de insert-tab. Kijkt in Semantics, reference, structure, en ziet dan de knop. Kopieert de tekst die zij heeft getypt naar de eerste placeholder en doet niets met de tweede.

Aandachtspunt	Commentaar
Ziet de gebruiker waar het element is ingevoegd?	Ja
Wat beschrijft de gebruiker te zien nadat het element is ingevoegd?	<i>niet naar gevraagd.</i>
Snappen gebruikers waarom er meerdere elementen zijn ingevoegd?	Ja
Begrijpt de gebruiker de placeholders?	Ja
Overig	Vult niet alle kinderen in.

3.2. Invoegen van een element met vaste volgorde

Geschreven verslag

Lillian vindt de taak aan de hand van de beschreven stappen die er in staan. Zij herkent vervolgens aan de titel en het label dat het hele velletje de taak is. Ook zij gaat aan het einde van het topic staan. Ze kopieert de content uit de postreq. Geeft een enter na de postreq waardoor er een tweede ontstaat(bug!) en wil deze omzetten naar result. Zij zoekt in semantics, structure en dan in task. Hier vindt ze result en deze klikt ze aan. Deze komt op zijn vaste plek terecht. “En dan zou ik hem even daar onder moeten zetten” Zij vindt dat de result als laatste hoort en wil hem dus eigenlijk

ergens anders hebben. Ik vraag waarom: “Omdat een resultaat als laatste hoort”. Probeert opnieuw aan te maken door eerst te verwijderen maar dit gaat fout door een bug in het verwijderen.

Gaat nogmaals proberen om een nieuw resultaat aan te maken. Ziet dat er alleen nog maar verwijderd kan worden. Snapt niet precies waarom dat is. Vraagt zich af of er daadwerkelijk een vaste volgorde is, of dat dit bij het prototype hoort. Kan dit niet met zekerheid bevestigen. Voegt 4 keer het resultaat in, maar ziet geen enkele keer de boodschap.

Aandachtspunten

Aandachtspunt	Commentaar
Vindt de gebruiker de Task?	Ja
Waar gaat hij met de cursor staan?	Onder de prereq
Ziet hij waar het element is ingevoegd?	Ja
Ziet hij de feedback message?	Nee
Kan de gebruiker benoemen wat er gebeurd is?	Nee
Overig	

3.3. Invoegen van elkaar uitsluitende elementen

Geschreven verslag

Gaat met de cursor in de description staan. Drukt hier op enter waardoor de description weer kopieert. deze wordt weer verwijderd en de cursor blijft achter in de description staan. Navigeert naar structure, semantics, reference en vervolgens naar topic. Hier wordt de abstract gevonden en de knop wordt gehoverd met de muis. Hierdoor verschijnt de tooltip in beeld. “Nee, die is grijs dus dan sta ik niet op de goede plek”. - Leest de tooltip! - “Ooooooke!”. Selecteert de text van de description. Knipt deze. Converteert de description naar een abstract. Plakt vervolgens de tekst weer. Geeft aan klaar te zijn. “Het is wel streng hoor, dit programma.” Ik vraag wat ze precies gedaan heeft. “dit was een description en ik kreeg te horen dat ze niet samen mogen zijn dus heb ik hem veranderd in een abstract”.

Aandachtspunten

Aandachtspunt	Commentaar
---------------	------------

Ziet de gebruiker de verandering in de toolbar?	Ja
Kan de gebruiker benoemen wat er aan de hand is?	Ja
Ziet de gebruiker de tooltips?	Ja
Overig	Een van de weinige keren dat de tooltip daadwerkelijk werd gelezen.

3.4. Enkel toegestane elementen

Geschreven verslag

Gaat in het resultaat staan en geeft een enter. Door bug ontstaat een tweede resultaat. Merkt op dat er eerst een resultaat verwijderd moet worden voordat deze weer weg kan. Hieruit maakt zij op dat er maar een enkele result is toegestaan. Ondanks het feit dat de tooltip ruim 20 seconden open staat, leest zij deze niet. Suggereert nog de optie om het via een opsomming te doen.

Aandachtspunten

Aandachtspunt	Commentaar
Begrijpt de gebruiker vooraf al dat het niet kan?	Nee
Leest de gebruiker de tooltip?	Nee
Begrijpt de gebruiker achteraf dat het niet kan?	Ja, maakt dit op uit het feit dat er niet meer kan worden toegevoegd.
Overig	

3.5. Aangemoedigde attributen

Geschreven verslag

Typt het woord ThiemeMeulenhoff. Selecteert deze en klikt wederom op RMB. *“Ik zou verwachten dat hier wat zou zitten”*. Herinnert zich dat dat niet werkt in het prototype. Navigeert naar structure, semantics en dan reference. Klikt op web address en typt een link. Vervolgens klikt ze op confirm.

Voor de foreign text wordt eerst een selectie gemaakt. De knop wordt direct gevonden en Frans wordt gekozen als language. *“ik vind het raar dat je hier op confirm moet drukken”*. Dit omdat dit in de toolbardropdowns niet hoeft.

Aandachtspunten

Aandachtspunt	Commentaar
Gaat de gebruiker in op de aanmoedigingen?	Ja
Wat vind de gebruiker van de popovers?	Makkelijk, maar wil naast de popover kunnen klikken om hem te sluiten.
Overig	

3.6. Verplichte attributen

Geschreven verslag

Het aanmaken van het handelsmerk gaat hier net als bij de andere deelnemers zeer eenvoudig.

Aandachtspunten

Aandachtspunt	Commentaar
Wordt de attribuutwaarde ingevuld?	Ja
Begrijpt de gebruiker wat hij moet doen?	Ja
Overig	Zonder problemen uitgevoerd.

4. Deelnemer: Rindert

4.1. Invoegen van een element met verplichte kinderen

Geschreven verslag

Rindert begint door de cursor aan het einde van de eerste topic te plaatsen. Hier geeft hij een enter, waardoor een nieuwe paragraaf ontstaat. Hierin schrijft hij een korte tekst over de gevaren van koffie. Vervolgens wordt het even stil. Ik vraag hem of hij denkt klaar te zijn en dat weet hij niet zeker. Hij selecteert de tekst die hij heeft geschreven. Ik vraag of hij denkt dat het systeem nu weet dat dit een gevarenbeschrijving is. Hij antwoord met nee en concludeert dat hij een element moet gaan opzoeken. Hiervoor gaat hij eerst alle knoppen in de start-tab af. Uit de tooltips maakt hij op dat geen van deze knoppen bieden wat hij zoekt. Vervolgens vindt hij de insert-tab. Hier gaat hij eerst de verschillende dropdowns af (topic, structure, semantics, reference). Vervolgens vindt hij de knop voor hazard en klikt hier op. De hazard statement verschijnt onder zijn selectie. Hij knipt de tekst die hij geschreven heeft. Vervolgens probeert hij de placeholder tekst te selecteren, maar ontdekt dat dit niet mogelijk is en klikt er vervolgens op. Daarna plakt hij de tekst in de typeofhazard. Als laatste klikt hij op de howtoavoid en vult deze in.

Aandachtspunt	Commentaar
Ziet de gebruiker waar het element is ingevoegd?	Ja
Wat beschrijft de gebruiker te zien nadat het element is ingevoegd?	Een gevaar met onderdeeljes.
Snappen gebruikers waarom er meerdere elementen zijn ingevoegd?	<i>Niet naar gevraagd</i>
Begrijpt de gebruiker de placeholders?	Niet direct, maar leert dit zeer snel.
Overig	

4.2. Invoegen van een element met vaste volgorde

Geschreven verslag

Bij het zoeken naar de task wordt deze herkend aan de steps. Na het horen van de testtaak navigeert hij naar het einde van de postreq en geeft hier een enter. Hierdoor verschijnt er een tweede postreq(bug). “Is dit de bedoeling?” Ik geef aan dat dit een bug in het prototype is en dat de

deelnemer zich even moet inbeelden dat dit een gewoon paragraaf is. Vervolgens gaat hij op zoek naar de knop om een resultaat in te voegen. De insert tab staat nog open waardoor hij hier begint met zoeken. Verschillende dropdowns worden geprobeerd: (Structure, semantics, reference, dan task). In Task wordt het result gevonden en aangemaakt. *“Hee, hij komt ergens anders te staan”*. Het verklarende systeembericht wordt niet opgemerkt. Ik vraag waarom hij denkt dat dit gebeurde. *“Er zal wel een soort specifieke volgorde zijn.”* De regel wordt dus uitgesproken, maar niet met zekerheid. Het resultaat wordt ingevuld en de gebruiker geeft aan klaar te zijn.

Aandachtspunten

Aandachtspunt	Commentaar
Vindt de gebruiker de Task?	Ja
Waar gaat hij met de cursor staan?	Einde postreq
Ziet hij waar het element is ingevoegd?	Ja
Ziet hij de feedback message?	Nee
Kan de gebruiker benoemen wat er gebeurd is?	Ja, maar niet met zekerheid.
Overig	

4.3. Invoegen van elkaar uitsluitende elementen

Geschreven verslag

Ik geef aan dat er een samenvatting geplaatst moet worden, en dat het Engelse woord daarvoor “abstract” is. De cursor wordt aan het begin van de eerste normale paragraaf geplaatst. Enter. Nieuwe paragraaf wordt aangemaakt en met cursortoets omhoog zet hij de cursor daar neer. Vervolgens wordt gezocht naar een manier om de abstract in te voegen. Eerst wordt gezocht bij structure, daarna semantics, daarna topic. Hier blijft hij even hangen op abstract. De tooltip verschijnt, maar wordt niet gelezen. Daarna wordt de dropdown voor description geopend en ziet hij “convert to abstract”. Hier wordt op geklikt. De description wordt dus omgezet naar een abstract en de gebruiker geeft aan klaar te zijn. Ik vraag hem of hij kan uitleggen wat er precies aan de hand was. *“Ik kon alleen de description omzetten dus dat heb ik gedaan”*. Ik vraag waarom hij denkt dat dit was. *“Dat weet ik niet precies. Je zal wel maar een van de twee kunnen hebben”*.

Aandachtspunten

Aandachtspunt	Commentaar
Ziet de gebruiker de verandering in de toolbar?	Ja
Kan de gebruiker benoemen wat er aan de hand is?	Nee
Ziet de gebruiker de tooltips?	Nee
Overig	

4.4. Enkel toegestane elementen

Geschreven verslag

Rindert scrollt terug naar de eerder bewerkte task. Hij positioneert de cursor in de bestaande result een geeft een enter. Er verschijnt (bug) een nieuwe result, waarvan ik aangeef dat dit een bug is en dus geen geldige oplossing. Hij verwijdt deze weer. De cursor komt daardoor weer in de eerste result te staan. Hij gaat in de toolbar naar Task, waar hij de vorige keer ook het result heeft gevonden. Hier vindt hij geen result. Reference, Semantics, Structure. Dit wordt drie keer herhaald. "Oké, ik kan het niet vinden." Ik vraag om uitleg. "De knop is gewoon weg. Net stond hij bij task en nu niet meer". De knop "remove result" is wel gezien, maar hier heeft hij verder niets mee gedaan.

Aandachtspunten

Aandachtspunt	Commentaar
Begrijpt de gebruiker vooraf al dat het niet kan?	Nee
Leest de gebruiker de tooltip?	Nee
Begrijpt de gebruiker achteraf dat het niet kan?	Nee
Overig	<i>Taak is niet voltooid: de situatie kan niet worden benoemd.</i>

4.5. Aangemoedigde attributen

Geschreven verslag

Rindert begint met het typen van het woord “Google”. Vervolgens selecteert hij dit woord. Hij klikt met rechtermuisknop op het de selectie en leest eventjes het browsermenu dat verschijnt. *“Hier staat het niet bij”*. Hij merkt niet dat dit een browsermenu is en geen Fonto-menu. Gaat nu zoeken in de toolbar. Kijkt direct bij reference. Vindt hier “web address” en probeert deze. De popover verschijnt. Rindert voert zonder vragen te stellen www.google.nl in en drukt op confirm. Vervolgens geeft hij aan klaar te zijn.

Het stuk Franse tekst wordt geselecteerd. Er wordt gezocht naar de juiste knop in de dropdowns semantics, reference en structure. Na 5 seconden staren wordt de knop voor foreign element gevonden. Klikt hier op, en de popover verschijnt. French wordt aangeklikt en vervolgens drukt hij op confirm en geeft aan klaar te zijn.

Aandachtspunten

Aandachtspunt	Commentaar
Gaat de gebruiker in op de aanmoedigingen?	Ja
Wat vind de gebruiker van de popovers?	Logisch, maar moeten kunnen sluiten door ernaast te klikken
Overig	

4.6. Verplichte attributen

Geschreven verslag

Het woord “Senseo” wordt geselecteerd. wordt geselecteerd. Vervolgens kijkt Rindert in de dropdown voor reference. Daarna in semantics. Hier vindt hij trademark, kiest een type uit en bevestigt de actie. Aan het geplaatste icoontje in de content herkent hij dat de taak is voltooid en hij geeft aan klaar te zijn.

Aandachtspunten

Aandachtspunt	Commentaar
Wordt de attribuutwaarde ingevuld?	Ja
Begrijpt de gebruiker wat hij moet doen?	Ja
Overig	

Bijlage K

Adviesrapport

K

Adviesrapport

Een gebruiksvriendelijke manier voor het bewerken van XML-elementen in FontoXML

Afstudeerder: Youri Bosselaar
Studentnummer: 08033153
Bedrijf: FontoXML (onderdeel van Liones B.V.)
Plaats: Rijswijk
Examinator: Chris Heydra
2^e examiner: Ru Kleijn
Bedrijfsmentor: Vincent Smedinga
Inleverdatum: 9 januari 2015



FONTOXML

Inhoudsopgave

1.	Inleiding	233
2.	Totstandkoming van advies	234
2.1.	Probleemstelling	234
2.2.	Onderzoeksrapport	234
2.3.	Ontwerprapport	234
2.4.	Prototype	235
2.5.	Testrapport	235
3.	Adviezen op basis van het testrapport	236
3.1.	Aanmaken van elementen met verplichte kinderen	236
3.2.	Aanmaken van elementen met een vaste volgorde	236
3.3.	Aanmaken van elementen die elkaar uitsluiten	237
3.4.	Aanmaken van beperkt toegestane elementen	237
3.5.	Aanmaken van elementen met aangemoedigde attributen	238
3.6.	Aanmaken van elementen met verplichte attributen	238
4.	Adviezen op basis van het onderzoeksrapport	240
4.1.	Uitwerken van overige bewerkingen	240
4.2.	Ontwikkelen van UI pattern library	240

1. Inleiding

Dit document is bedoeld ter overdracht van de resultaten, ontwerpen en bevindingen van mijn afstudeeropdracht voor FontoXML. In dit document worden adviezen uitgebracht over de doorontwikkeling van mijn project en over de toepassing van de door mij ontworpen interacties. Deze adviezen zijn zo beknopt mogelijk gehouden. Meer diepgaande informatie over het project is te vinden in ander producten zoals het onderzoeksrapport, het ontwerprapport en het testrapport.

In hoofdstuk 2 van dit project wordt er kort teruggekoppeld op de totstandkoming van dit advies. In dit hoofdstuk wordt uitgelegd welke activiteiten er zijn uitgevoerd binnen dit project en welke producten daar uit zijn voortgekomen. In hoofdstuk 3 wordt een aantal adviezen gegeven op basis van de usabilitytest die in dit project is uitgevoerd. Tot slot bevat hoofdstuk 4 een aantal adviezen op basis van het uitgevoerde onderzoek in de eerste fase van het project.

2. Totstandkoming van advies

Dit hoofdstuk is bedoeld om te verklaren hoe het advies uit dit rapport tot stand is gekomen. Op die manier kan worden beoordeeld of deze aanpak correct is geweest. Zo kan ook worden bepaald of het advies van genoeg kwaliteit wordt geacht om dit wel of niet op te volgen.

2.1. Probleemstelling

Dit project begon met het volgende probleem: *“Het kost momenteel veel tijd om voor het bewerken van de verschillende elementen in een XML-schema de juiste interface te kiezen”*. Vanuit deze probleemstelling ben ik vervolgens gaan onderzoeken wat “het bewerken van elementen in een XML-schema” precies inhoudt.

2.2. Onderzoeksrapport

Allereerst is er een onderzoek gedaan naar de bewerking van XML-elementen. Hierbij is in kaart gebracht welke bewerkingen er allemaal zijn (denk aan element aanmaken, verwijderen, verplaatsen). Vervolgens is onderzocht welke factoren er invloed hebben op de benodigde User Interface patronen voor deze bewerking. Hierbij gaat het vooral om regels uit XML-schema's en best practices. Elke combinatie tussen een bewerking en een factor van invloed is in dit rapport bestempeld als een “designprobleem”: een situatie waarvoor een oplossing ontworpen moet worden. Op deze manier leidde dit onderzoek tot 27 designproblemen. Uit tijdsoverwegingen zijn vervolgens alleen de designproblemen die te maken hebben met de bewerking “element aanmaken” meegenomen naar de ontwerpfase. De resultaten en conclusies van dit onderzoek zijn terug te lezen in een onderzoeksrapport. Daarnaast is er een gebruikersonderzoek gedaan, om te achterhalen wat de eigenschappen en behoeften van de gebruikers van FontoXML waren.

2.3. Ontwerprapport

Nadat de designproblemen in kaart waren gebracht, is er een aantal oplossingen ontworpen. Deze zijn gebaseerd op bestaande oplossingen uit soortgelijke applicaties, bewezen interfacepatronen of nieuwe concepten. In dit project zijn alleen de designproblemen uit de bewerking “element aanmaken” behandeld. Dit omdat er geen tijd was voor het ontwerpen van alle bewerkingen. Voor ieder van deze designproblemen is er een tweetal oplossingen ontworpen. Deze oplossingen zijn uitgewerkt, opgenomen en onderbouwd in een ontwerprapport. Hierin staan verder ook de systeemeisen en richtlijnen waar rekening mee moest worden gehouden bij het ontwerpen.

2.4. Prototype

Na het uitwerken van de ontwerpen is er een digitaal prototype gebouwd. Hierbij is gebruik gemaakt van HTML, CSS en jQuery. Dit prototype was niet representatief voor de gehele applicatie. Het diende echter enkel voor het tonen van de interacties zoals die binnen FontoXML zouden (kunnen gaan) werken. Voor alle ontworpen oplossingen is een situatie uitgewerkt waarin deze voorkomt. Zo kon er, aan de hand van dit prototype, zowel door experts als met gebruikers worden getest op usability.

2.5. Testrapport

Als laatste projectactiviteit is er, met behulp van het prototype, een aantal usabilitytests uitgevoerd. Hiervoor zijn vier vakexperts van ThiemeMeulenhoff als deelnemers gebruikt. De resultaten van deze test zijn verwerkt in een testrapport. In dit rapport is zowel de volledige uitwerking van de tests als een lijst met verbeterpunten en een lijst met conclusies te vinden. De resultaten uit het testrapport vormen de belangrijkste basis voor het advies uit dit document.

3. Adviezen op basis van het testrapport

Zoals eerder genoemd zijn de ontworpen oplossingen voor de verschillende designproblemen onderworpen aan een usabilitytest. De resultaten van die test zijn vastgelegd in een testrapport, waarin terug te lezen is hoe ieder ontwerp heeft gescoord op gebruiksvriendelijkheid. In deze adviezen wordt verwezen naar verschillende ontwerpen. Voor gedetailleerde informatie over deze ontwerpen kan het ontwerprapport worden geraadpleegd.

3.1. Aanmaken van elementen met verplichte kinderen

De twee ontwerpen voor het aanmaken van elementen van verplichte kinderen zijn allebei behoorlijk goed uit de test gekomen. Het lukte de gebruikers bij beide alternatieven altijd om het element aan te maken. In het ontwerp met de modal werd iets vaker content geschreven voor de beide kinderen. Toch is het advies om voor dit probleem ***het template met de placeholders te implementeren***. Dit vanwege de volgende redenen:

1. Het invoeren van content voor de kinderen is niet verplicht. Een modal breekt daardoor onnodig de flow. Bij het template is dit niet het geval: de elementen worden direct geplaatst waardoor gebruikers ongestoord door kunnen werken
2. De learnability van de placeholders is hoog. Gemiddeld hadden de deelnemers 2 pogingen voordat zij de placeholder goed begrepen. Daarna werden deze placeholders moeiteloos gebruikt.

3.2. Aanmaken van elementen met een vaste volgorde

Voor het omgaan met het aanmaken van een element dat in een vaste volgorde moet staan zijn er ook twee opties ontworpen. Bij ontwerp A is de toolbarknop voor dit element altijd ingeschakeld en wordt het element automatisch op de juiste positie ingevoegd. In ontwerp B kunnen deze elementen worden ingevoegd met behulp van structuursuggesties in een zijpaneel. Het advies is hier ***het ontwerp met de ingeschakelde toolbarknop door te ontwikkelen en vervolgens te implementeren***. De volgende argumenten zijn hiervoor te noemen:

1. Het zijpaneel bleek erg moeilijk vindbaar en lijkt niet te implementeren zonder een uitgebreide doorontwikkeling. De koppeling aan de elementen in het document is moeilijk te filteren en de structuursuggesties zouden daardoor voor alle elementen gedaan moeten worden.
2. Met ontwerp A kon de testtaak succesvol worden uitgevoerd: iets dat bij veel deelnemers met ontwerp B niet lukte.

Voor implementatie moet echter eerst nog een verbeterpunt worden behandeld. Er was namelijk maar een van de testdeelnemers die met zekerheid de situatie kon benoemen. Het systeembericht waarin deze situatie wordt verklaard, werd niet gelezen. Op basis van observaties bij de test lijkt het zo te zijn dat dit komt doordat gebruikers bij het aanmaken van elementen niet naar het hele scherm kijken. Omdat men zoekende is naar de plaats waar het nieuwe element wordt geplaatst, let niemand op de boven-of onderrand van het scherm. Voor beide posities is het systeembericht getest, en beide zonder succes. Aan te raden is dan ook om het systeembericht dicht bij het nieuwe element weer te geven. Wellicht zelfs in het document. Hierbij moet wel worden voorkomen dat het bericht opdringerig wordt: een verklaring moet nooit als storend worden ervaren.

Nog beter zou zijn om een ontwerp te verzinnen waardoor gebruikers al vooraf kunnen herkennen met welke situatie zij te maken hebben. Dit is echter geen vereiste voor een gebruiksvriendelijk ontwerp en hoeft dus niet per se uitgevoerd te worden voor implementatie.

3.3. Aanmaken van elementen die elkaar uitsluiten

Bij de eerste oplossing die voor dit probleem was ontworpen, worden de knoppen van elkaar uitsluitende elementen uitgeschakeld wanneer een van deze elementen al bestaat. Hierdoor kunnen zij niet meer worden ingevoegd, en wordt een schema-invaliditeit voorkomen. Voor het element dat al bestaat komt er een submenu, waarin de mogelijkheid om te converteren wordt aangeboden.

Het tweede ontwerp toont een serie radiobuttons in de toolbar, waarmee elementen geconverteerd werden. Ondanks het feit dat dit wel duidelijker uitstraalde dat de elementen elkaar aansluiten, riep deze interface verwarring op en begrepen de testdeelnemers niet goed wat ze precies deden. Het ontwerp met de uitgeschakelde triggers is dan ook het beste uit de usabilitytest gekomen.

Het advies voor dit designprobleem is om **de uitgeschakelde triggers in combinatie met de tooltips en conversieknoppen te implementeren**. Daarbij is nog wel een verbeterpunt dat moet worden aangepakt. Het duurde bij de tests behoorlijk lang voordat gebruikers de conversieknop vonden. Dit ontstond vooral doordat de twee knoppen niet per se bij elkaar leken te horen. Hierdoor was het voor hen vreemd om in “een andere knop” naar deze functie te zoeken. Het is dus wenselijk om de knoppen van elementen die elkaar uitsluiten nog explicieter te groeperen.

3.4. Aanmaken van beperkt toegestane elementen

Bij het eerste ontwerp van dit designprobleem wisselt de functie van de trigger. Als een element nog niet aan zijn maximum aantal instanties zit, is de knop ingeschakeld en maakt deze het element aan. Indien het element wel al bestaat, verandert de functie van de knop. Ook het label wordt hierbij

aangepast. In plaats van dat het element wordt aangemaakt, wordt dit nu verwijderd met de knop. Bij de usabilitytest bleek dit ontwerp het minst gebruiksvriendelijk.

Het tweede ontwerp houdt de functie van de knop intact. De knop wordt uitgeschakeld als het element aan de maximum instanties zit. In een tooltip wordt uitgelegd wat de situatie is. Bij het testen van dit ontwerp las driekwart van de deelnemers de tooltip en konden zij de situatie met zekerheid benoemen. Om deze reden is het advies om **het ontwerp met de uitgeschakelde knoppen** toe te passen in FontoXML.

3.5. Aanmaken van elementen met aangemoedigde attributen

Voor dit ontwerpprobleem is slechts een oplossing ontworpen. In deze oplossing wordt, bij het aanmaken van een element met een aangemoedigd attribuut, een popover getoond. In deze popover wordt de gebruiker verzocht om bepaalde actie uit te voeren. Hierbij kun je denken aan het invullen van een webadres of aan het aangeven van een bepaalde taal. De gebruiker kan dit verzoek afwijzen of er op ingaan, want er is geen sprake van een verplichte actie. Ongeacht of het attribuut wel of niet wordt ingevuld, het element wordt altijd geplaatst. Dit ontwerp is erg goed uit de gebruiksvriendelijkheidstest gekomen. Zonder dit te vragen, voerden alle gebruikers de aangemoedigde actie uit. Het advies is dan ook om **dit ontwerp te implementeren in FontoXML**.

3.6. Aanmaken van elementen met verplichte attributen

Er zijn elementen die een bepaald attribuut moeten bevatten, die ook nog eens een waarde moet hebben. Voor deze situatie zijn er ook twee oplossingen ontworpen. In de eerste oplossing komt er, voordat het element wordt aangemaakt, een modal tevoorschijn waarin de benodigde acties kunnen worden uitgevoerd. Het annuleren van deze modal betekent dat de aanmaakactie ook geannuleerd wordt. Bij de gebruikerstest slaagden de deelnemers er allemaal moeiteloos in om de testtaken uit te voeren. Ook ontstonden er geen schema-INVALIDE situaties.

Bij het tweede ontwerp werden de verplichte acties verwerkt in de toolbar. Zo bevatte de knop voor het aanmaken van een link direct een invoerveld voor het doelwit van die link. Voor notities, waarbij de attribuutwaarde van het attribuut type uit een vaste lijst komt, was er per waarde een aparte knop. Ook hier slaagden deelnemers er meestal in om de taak uit te voeren. Voor een link was dit echter al lastiger: voor lange URL's was het invoerveld niet breed genoeg. Hiermee kwam direct het grootste nadeel van dit ontwerp in beeld. Niet alle invoersoorten kunnen gemakkelijk in de toolbar worden verwerkt. In een modal is hiervoor veel meer ruimte. Daarnaast zorgt dit ontwerp dat er erg snel veel knoppen en functies in de toolbar komen. Aangezien deze vaak toch al erg vol is, is dit ontwerp dan ook niet geschikt.

Het ontwerp met de modal kan voor dit probleem worden geïmplementeerd. Er zijn in de test verder geen opvallende verbeterpunten gevonden voor dit ontwerp.

4. Adviezen op basis van het onderzoeksrapport

Het eerste eindproduct van dit project was het onderzoeksrapport. Dit vormde niet alleen de basis voor de ontwerpen die er gemaakt moesten worden, maar bevat ook waardevolle informatie over het bewerken van XML-elementen. Op basis van dit onderzoeksrapport is er een tweetal adviezen opgesteld.

4.1. Uitwerken van overige bewerkingen

Het doel van dit project was om te kunnen bepalen in welke situaties een bepaald interfacepatroon gebruikt moet worden bij het bewerken van XML-elementen. Na afloop van dit project kan hier nog geen antwoord op worden gegeven. Binnen dit project is namelijk alleen de bewerking “element aanmaken” ontworpen. Er zijn in het onderzoek naar bewerkingen nog acht andere bewerkingen gevonden.

Om het doel van dit project wel te bereiken, zullen de factoren van invloed voor al deze bewerkingen moeten worden ontworpen en getest. Pas dan kan er een compleet beeld worden gevormd en dus een uitspraak worden gedaan.

4.2. Ontwikkelen van UI pattern library

Vanwege de veranderlijke aard van een flexibele applicatie is het aan te raden om een pattern library op te zetten. Dit is een documentatievorm waarbij de verschillende interfacepatronen die in FontoXML kunnen worden gebruikt, stuk voor stuk zijn weergegeven. Per patroon kan vervolgens een uitleg, een stuk voorbeeldcode en een aantal use cases worden gegeven. Zo zou je bijvoorbeeld kunnen vastleggen dat voor het aanmaken van een element met een verplicht attribuut altijd een modal moet worden gebruikt.

Door een pattern library op te zetten kunnen verschillende leden van het team (ook niet-designers) gemakkelijker en sneller bepalen wat voor patroon zij toe willen passen voor bepaalde elementen. Op deze manier kan een hoop tijd worden bespaard, omdat er dus niet meer altijd designers nodig zullen zijn om dit soort beslissingen te nemen.

Daarnaast biedt een pattern library ook overzicht. Met dit overzicht kan het team van FontoXML snel zien welke patronen er al allemaal gebruikt worden in de applicatie. Hierdoor voorkom je dat er in de toekomst wellicht patronen dubbel zullen voorkomen.

Bijlage L

Verslag bedrijfsbezoek



Bedrijfsbezoek afstudeerstage

dinsdag 14 oktober 2014

<i>Afstudeerder</i>	Youri Bosselaar
<i>Studentnummer</i>	08033153
<i>Opdrachtgever</i>	FontoXML (onderdeel van Liones B.V)
<i>Examinator</i>	Chris Heydra
<i>Tweede examiner</i>	Ru Klein
<i>Bedrijfsmentor</i>	Vincent Smedinga

Dit document biedt een kort verslag van het bedrijfsbezoek aan Liones B.V. door Chris Heydra op 14 oktober 2014. Doel van dit gesprek was het bespreken van de voortgang van de afstudeerder en het verdere verloop van het traject. Daarnaast diende het bezoek om voor Chris een beeld te vormen van de context waarin het project wordt uitgevoerd.

Vorderingen tot dusver

Het project verloopt aardig zoals gepland. Tot dusver zijn de volgende activiteiten uitgevoerd:

1. Opstellen plan van aanpak
2. Onderzoeken van de context van het project
3. Analyseren van mogelijke bewerkingen op XML-elementen
4. Analyseren van factoren die van invloed zijn bij het bewerken van XML-elementen
5. Opstellen van richtlijnen voor ontwerp
6. Uitvoeren van een doelgroeponderzoek bij klant ThiemeMeulenhoff
7. Opstellen van systeemeisen

Wel is er vertraging in het houden van een benchmark. Deze vertraging is ontstaan doordat activiteit 4 een zeer complexe en grote activiteit bleek te zijn. Omdat het in eerste de bedoeling was om alle activiteiten in chronologische volgorde af te lopen, werd er dus wat vertraging opgelopen. Vervolgens is besloten om de benchmark te gaan houden op basis van de huidige resultaten uit de analyse en eventuele volgende factoren nog wel toe te voegen, maar niet te ontwerpen. De vertraging betreft ongeveer 2 a 3 dagen.

Verder verloop van het project

De volgende stappen in het project zijn grofweg:

1. Designproblemen (systeemeisen) prioriteren
2. Interfacepatronen schetsen
3. Digitale prototypes ontwikkelen
4. Testen van ontworpen patronen en uitbrengen van een advies

Over verdere contact- en beoordelingsmomenten is het volgende gezegd:

- Rond week 10 (3 november) vindt de conceptbespreking plaats. Hierin wordt het skelet van het afstudeerverslag besproken. Dit skelet bevat in ieder geval een inhoudsopgave, een opdrachtomschrijving, een bedrijfsomschrijving en een aantal uitgewerkte activiteiten. Op deze manier kan zowel de indeling van het document als de schrijfstijl al worden beoordeeld.
- Rond week 14 (1 december) vindt het tussentijds assessment plaats. Hier wordt een advies gegeven aan de afstudeerder op basis van de huidige staat van het afstudeerverslag. Het is aan te raden dat hier de activiteiten van het project nagenoeg zijn afgerond. Op deze manier is er nog voldoende tijd voor het verbeteren en afronden van het afstudeerverslag.

Bijlage M

Beoordeling TTA

M

Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	------------------------	--------------------

Formulier tussentijds assessment

Student: Youri Bosselaar

Studentnummer: 08033153

Datum: 4-12-2014

eerste / tweede TTA:

Tijdens het tussentijds assessment is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	X	
b	Het afstudeerdossier is digitaal beschikbaar	X	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	X	
d	Het goedgekeurde afstudeerplan is aanwezig	X	
e	Het plan van aanpak is aanwezig	X	
f	Reeds geleverd commentaar is aanwezig	X	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	X	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	X	

Aanpak	O	T	V	G
Passend			X	
Theoretisch verantwoord			X	
Samenhang uitvoering beroepstaken			x	

Beroepstaken op afgesproken niveau uitgevoerd?		O	T	V	G
1	Opstellen plan van aanpak			X	
2	Opstellen interaction design			X	

3	Uitwerken prototype			X	
4	Opstellen testplan & testrapportage (geen inzicht)				
5	Usability test opzetten en uitvoeren (geen inzicht)				

Producten	O	T	V	G
<i>Tussenproducten</i>			X	
<i>Eindproducten</i>			?	

Effectief communiceren	O	T	V	G
<i>Binnen afstudeerbedrijf (geen inzicht)</i>				
<i>Afstudeerdossier</i>			X	

Reflectie	O	T	V	G
<i>Inzicht in eigen functioneren</i>			X	
<i>Inzicht in eigen leerproces</i>			x	

Toelichting per beoordelingscriterium

Aanpak
Youri heeft een correcte aanpak gekozen. Keuzes m.b.t, methoden en technieken mogen beter beargumenteerd worden.

Beroepstaken op afgesproken niveau uitgevoerd?
De beroepstaken 1 t/m 3 zijn correct uitgevoerd. De 2 beroepstaken rond het testen kunnen we op dit moment nog niet beoordelen, aangezien die nog uitgevoerd moeten worden. Het geheel ziet er vertrouwenwekkend uit.

Producten
Tussenproducten zijn aanwezig t/m het testplan en in orde.

Effectief communiceren
Youri schrijft prettig en overzichtelijk. Op een aantal punten is meer argumentatie en voorbeeld nodig. Dat geldt bijvoorbeeld voor: de schetsen/ontwerpen, keus van methoden en technieken, de planning, XML, processtappen gebruikersonderzoek, verantwoording projectaanpak, halen van projectdoelen, e.d. We hebben tijdens het gesprek suggesties gedaan voor verbeteringen.

Reflectie
Het is met Youri goed praten over zijn werk. Hij kan zijn eigen werk goed overzien en snapt feedback.

Advies

x	Inleveren (bindend advies)
	Verlengen (vrijblijvend advies)
	Stoppen (vrijblijvend advies)

Besluit student

Aankruisen welke beslissing de student heeft genomen (alleen na vrijblijvend advies)

x	Afstudeerdossier wordt op afgesproken datum ingeleverd	Inleverdatum: 9 januari 2014
	Afstudeerperiode wordt verlengd	Inleverdatum:
	Student stopt met afstudeeropdracht	

Naam begeleidend examinator: Chris Heydra

Naam tweede examinator: Ru Klein