

Interactiepatronen binnen een applicatie-architectuur

Interactie modellering en simulatie op basis van een eenvoudige applicatie-architectuur

Bert Dingemans

Een applicatie-architectuur op basis van interacties biedt de mogelijkheid om de structuur op te delen in voor gebruikers logische onderdelen. Dit maakt interactief en iteratieve applicatie-ontwikkeling mogelijk. Interactiepatronen beschrijven de basisaspecten van interacties. Er wordt beschreven hoe deze patronen een bijdrage kunnen leveren aan applicatie-ontwikkeling, het opstellen van een applicatie-architectuur en gebruikersinteractie.

Inleiding

Applicatie-architectuur

Bij het ontwikkelen van omvangrijke informatiesystemen wordt steeds vaker gebruik gemaakt van een applicatie-architectuur. Vaak zullen hier redenen als het reduceren van complexiteit of het realiseren van hergebruik (bijvoorbeeld door het toepassen van frameworks) aan ten grondslag liggen. Het opstellen van een dergelijke applicatie-architectuur is voornamelijk van belang in een dynamische organisatie, maar ook binnen een organisatie met een complex applicatie portfolio is een applicatie-architectuur een goed instrument voor het beheerbaar maken van de toegepaste informatiesystemen binnen de organisatie. In de volgende paragrafen wordt kort ingegaan op een eenvoudige applicatie-architectuur InterActory genaamd. Binnen deze architectuur nemen interacties een centrale plaats in. Met interactie wordt in dit artikel verstaan:

De representatie van domeinobjecten binnen informatiesystemen aan gebruikers en eventueel andere informatiesystemen. Hierbij wordt naast de weergave van deze objecten ingegaan op het gedrag van de interactie op het vlak van validaties, volgorde en navigatie.

Interactie onderscheidt zich in deze van een interface dat enkel de representatie van het informatiesysteem aan de gebruiker behandelt. Interactie is dan ook actiever dan de interface omdat gedrag deel uit maakt van een interactie. In het architectuurmodel interActory [Dingemans 2008] wordt een architectuur geïntroduceerd waarbij interacties een centrale plaats innemen. Deze gelaagde architectuur maakt het mogelijk om vanuit werkprocesmodellen de toestand van domeinobjecten te wijzigen via interacties.

Modellering- en werkwijze

De modelleer- en werkwijze dient de totstandkoming van een model van de applicatie-architectuur. Bij het opstellen van dit model wordt in veel situaties een werkwijze gevolgd

waarbij in strikt gescheiden stappen een model wordt opgesteld en dit wordt vervolgens omgezet in een applicatie. Deze werkwijze staat bekend als de waterval methode. Sinds enige jaren zijn een aantal methoden opgekomen waarbij men meer iteratief ontwikkeld. Dit houdt in dat in kleine cycli van ontwerp en ontwikkeling een applicatie ontwikkeld wordt. Dit wordt iteratief ontwikkelen genoemd en bekende methoden van deze werkwijze zijn extreme programming en scrum. Op zich is deze werkwijze voor het ontwikkelen van een applicatie een duidelijke verbetering ten opzichte van de waterval methode. Toch blijft bij deze methoden een aspect onderbelicht en dat is de interactie met de toekomstige gebruikers van het systeem.

In vier stappen wordt binnen InterActory een iteratie cyclus doorlopen. De activiteiten binnen de stappen zijn:

- Interacteer
- Modelleer
- Registreer
- Evalueer prototype

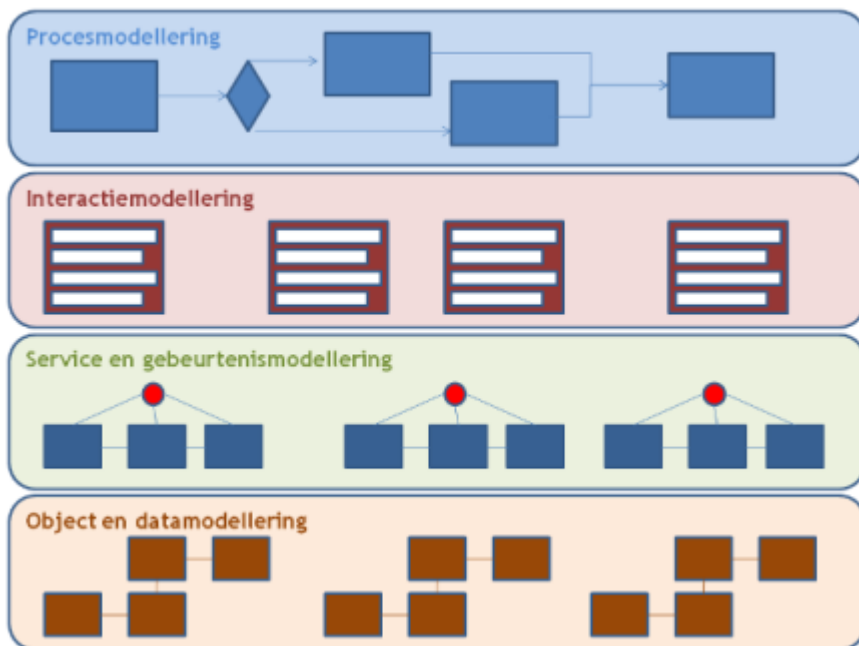
Interacteer

Er wordt met de toekomstige gebruikers gecommuniceerd over het domein dat gemodelleerd dient te worden. In de gesprekken worden de gebruikers uitgenodigd om voorbeelden te noemen van hoe gewerkt wordt. Daarnaast wordt gevraagd om documenten, lijsten en bestaande werkwijzen toe te lichten en wat positieve en negatieve punten van de huidige situatie is. Op basis hiervan kunnen de gewenste interactiepatronen geanalyseerd worden.

Modelleer

Op basis van de interacties met de (toekomstige) gebruikers wordt vervolgens een model opgesteld bestaand uit:

- Procesmodel
 - Processtappen en eventuele condities
 - Actoren
- Interactiemodel
 - Interacties
- Servicemodel
 - Bewerkende services
 - Versteckende services
- Domein- of objectmodel
 - Gebeurtenis typen
 - Object typen



Figuur 1 Gelaagde architectuur en modellering

Zo is het mogelijk bij een eenvoudige applicatie-architectuur een deelmodel op te stellen, bijvoorbeeld alleen een domeinmodel. In complexere situaties kan het betekenen dat in iedere iteraties één van de modellen uitgewerkt wordt. Dat houdt in dat tijdens de interactie ingegaan wordt op een specifiek model. Ten behoeve van ieder hierboven genoemd model zijn binnen InterActory notatiewijzen en entiteitbeschrijvingen beschikbaar.

Registreer

Tijdens de registreerfase wordt het model in detail ingevuld met gegevens relevant voor de modellen en de gewenste entiteiten in het model. Dit registreren biedt enerzijds het voordeel dat men kan verifiëren of het model compleet in kaart is gebracht, anderzijds kan het gebruikt worden om het model te transformeren naar bijvoorbeeld:

- Een prototype
- Ontwerpdocumenten
- Programmatuur

Evalueer prototype

In de laatste fase van een iteratie wordt een prototype opgesteld van het applicatie model. Dit wordt beschikbaar gesteld aan de toekomstige gebruikers van de toepassing. Hierbij wordt vooraf duidelijk gecommuniceerd dat we met prototypes werken en men wordt uitgedaagd om voorstellen te doen wat verbeterd dient te worden om het actuele prototype verder te verbeteren. Hierbij wordt veelal naadloos overgegaan naar de volgende iteratie met de activiteit interacteer.

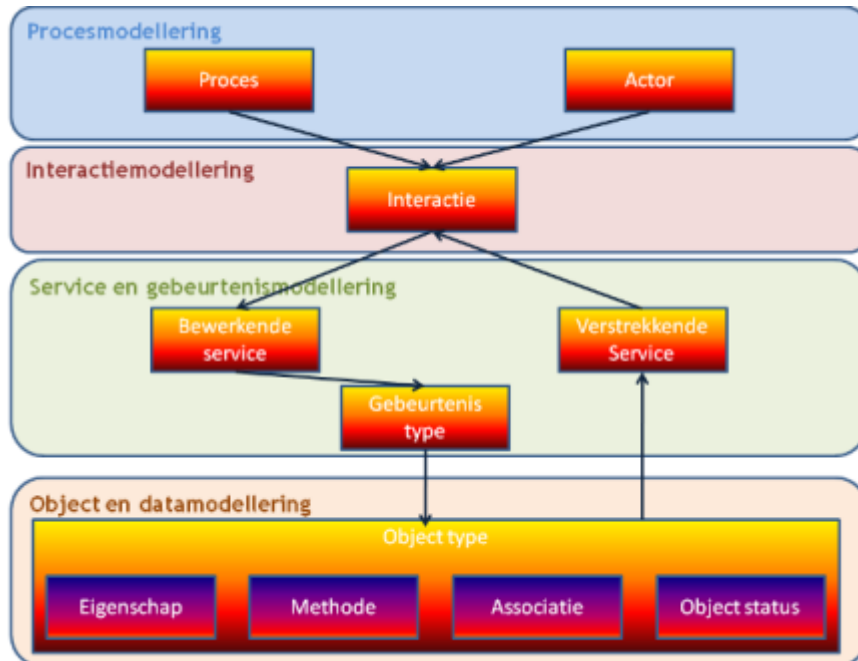
Architectuurentiteiten

Binnen een applicatie-architectuur worden een aantal entiteiten onderkend die met elkaar in relatie staan. In onderstaande afbeelding ziet u een vereenvoudigd beeld van deze entiteiten en hun relaties. Door het registreren van deze entiteiten en het maken van modellen waarin de verbanden tussen deze entiteiten te wordt de applicatie-architectuur "ingevuld".

Binnen een eenvoudige applicatie-architectuur worden veelal de volgende entiteiten onderkend:

- Proces

- Actor
- Interactie
- Bewerkende service
- Verstreckende service
- Gebeurtenis type
- Object type



Figuur 2 Architectuurentiteiten

In de afbeelding ziet u dat interacties de verbinding vormen tussen de werkprocesmodellering en de domeinmodellering. Via gegevensverstreckende services krijgen de interacties gegevens omtrent de toestand van één of meer entiteiten in het domeinmodel. Andersom kan een interactie via bewerkende services de toestand van één of meer entiteiten in het domeinmodel veranderen. Bij het opstellen van een werkproces(model) wordt op basis van condities een activiteit of processtap wel of niet uitgevoerd. Dit uitvoeren kan een handmatige bewerking zijn door de actor die de activiteit uitvoert. Ook is het mogelijk dat er een "interactie" met de applicatie(s) gewenst is, in dat geval zullen één of meerdere interacties binnen de activiteit (van het werkproces) uitgevoerd worden.

Interacties en simulatie

Interacties in de eenvoudigste vorm bestaan uit een scherm (of deel van een scherm) dat gegevens van entiteiten uit het domeinmodel weergeeft. Een eenvoudige interactie toont deze gegevens op een voor de gebruiker prettige wijze. Een prettige wijze betekent:

- De gegevens worden op een prettige wijze gepresenteerd, alleen relevante gegevens, in een logische volgorde met een opmaak die de gebruiker ondersteunt
- Indien er mutaties uitgevoerd worden dan wordt dit gedaan met besturingselementen die de gebruiker op adequate wijze ondersteunen in de werkzaamheden.
- Het uitvoeren van validaties op invoer- en bewerkingen vindt plaats op een logische wijze en op een voor de gebruiker juist moment.
- Navigatie binnen interacties geeft de gebruiker ondersteuning om te bepalen op welke plek de interactie zich bevindt in het werkproces of de applicatie.

Bij deze interacties spelen dus zowel de toestand als het gedrag een centrale rol. Deze combinatie zal voor de gebruiker betekenen dat de interactie het enige zichtbare element is

van de applicatie. Dat betekent dat voor de acceptatie van de oplossing grotendeels afhankelijk is van deze interacties. Reden om hieraan voldoende zorg te besteden. Om dit te bereiken kan gebruik gemaakt worden van simulatie van de interacties. In deze simulaties worden prototypes van de interacties ontwikkeld waarin zowel toestand als gedrag op eenvoudige wijze worden uitgewerkt. Eventueel kan ter voorbereiding van deze simulatie gebruik gemaakt worden van interactiepatronen om een selectie van basisinteracties uit te werken in het prototype en als uitgangspunt te nemen voor de verdere ontwikkeling. Deze simulatie kan plaatsvinden in gezamenlijk ontwerpessies met een aantal vertegenwoordigers van de verschillende gebruikersgroepen, inhoudelijk deskundigen en interactie ontwerpers. Voordeel van deze sessies is dat er interactie ontstaat tussen de verschillende groepen wat de bewustwording over toestand en gedrag van entiteiten bij alle deelnemers vergroot.

Smart use cases

Door Sander Hoogendoorn zijn in dit kader smart uses cases geïntroduceerd [Hoogendoorn, 2009] waarbij use cases gebruikt worden voor het modelleren van applicatie gedrag. Echter deze use cases zijn uitgebreid met stereotype waarmee een aantal standaard use cases gedefinieerd worden. Hiermee wordt het analyseren van de requirements versoepeld en gestandaardiseerd. Het toepassen van smart use cases is een eerste stap in deze standaardisatie. Echter het is mogelijk om dit verder door te voeren door het toepassen van interactiepatronen. Wat interactiepatronen zijn en hoe deze toegepast kunnen worden in een iteratief en interactief architectuur model wordt in de volgende hoofdstukken uitgewerkt.

Interactiepatronen

Informatiewerkers binnen en buiten organisaties komen dagelijks in aanraking met een grote hoeveelheid informatie uit diverse bronnen en in een wisselende vormgeving gepresenteerd. Als deze op een voor de gebruiker prettige manier gepresenteerd worden zal dit de gebruiker van deze informatie ten goede komen. Alan Cooper heeft drie basisregels van informatiewerkers beschreven [Cooper, 2003]:

- Hij/zij wil een redelijke hoeveelheid werk verzetten
- Hij/zij wil niet dom lijken
- Hij/zij wil plezierig werken.

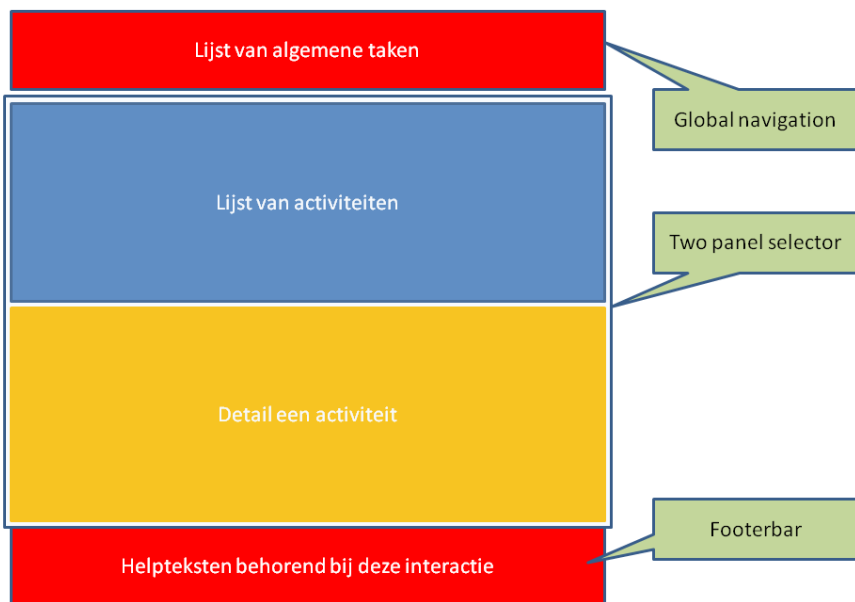
Er zijn de afgelopen jaren veel ontwikkelingen geweest op het vlak van "user experience" en interaction design. Een interessante ontwikkeling in dit werkgebied is het ontstaan van "interaction design pattern catalogs". Dit zijn catalogi die een generieke beschrijving geven van een interactie patroon. Via deze hyperlink (<http://www.welie.com/patterns/showPattern.php?patternID=shopping-cart>) wordt een (welbekend) voorbeeld gegeven van de opzet van een interactiepatroon. In dit artikel worden twee catalogi gebruikt te weten die van Martijn van Welie [Welie, 2008] en van Jennifer Tidwell [Tidwell, 2005].

Interactiepatronen bieden de mogelijkheid om structuur te brengen in de architectuur van één of meerdere applicaties. Bijvoorbeeld door een selectie van interactiepatronen de voorkeur te geven binnen deze applicaties waardoor deze applicaties eenduidiger worden in interface en gedrag. Anderzijds door herbruikbare (parameteriseerbare) componenten te ontwikkelen voor één of meerdere interactiepatronen.

Vanuit de applicatiearchitectuur dient echter wel rekening gehouden te worden met de gekozen technologische oplossing. Sommige interactiepatronen stellen non functionele eisen (veelal performance) waarmee deze niet bruikbaar zijn in elke technische situatie. In onderstaande paragrafen gaan we in op het selecteren van interactiepatronen en passen dit toe op een scenario.

Interactiepatroon selectie

Voor het opstellen van een applicatiearchitectuur voor één of meer informatiesystemen kunnen interactiepatronen goed gebruikt worden om structuur te brengen in de interacties tussen informatiesysteem en gebruiker. Veelal wordt deze structuur bereikt door een aantal Interactiepatronen te selecteren en hiermee voor de scenario's een beperkt aantal te gebruiken. In de afbeelding hieronder ziet u een voorbeeld op basis van een aantal interactiepatronen uit de catalogus [Tidwell, 2005].



Figuur 3 Interactiepatroon selectie

Bij het selecteren van interactiepatronen is het goed om rekening te houden met het feit dat er enkelvoudige en samengestelde interactiepatronen zijn. Enkelvoudige interactiepatronen hebben betrekking op één besturingselement en beschrijven het gedrag van dit ene element. Bijvoorbeeld het "jump to item patroon" waarbij in een keuzelijst door het intikken van de eerste letters automatisch gezocht wordt naar het bijbehorende element in de lijst van opties.

Een samengesteld interactiepatroon bestaat uit een groep van besturingselementen die door hun combinatie en gedrag gezamenlijk het interactiepatroon bepalen. Bijvoorbeeld het "Two-Panel selector" patroon waarbij het scherm is ingedeeld in twee vlakken, één voor het overzicht van een groep items en één voor de detaillering van een gekozen item.

Houdt rekening met het feit dat een samengesteld interactiepatroon opgebouwd kan worden uit meerdere enkelvoudige interactiepatronen. Daarnaast kan een interactiescenario bestaan uit meerdere (enkelvoudige- en meervoudige) interactiepatronen. Binnen een interactiescenario kunnen al deze interactiepatronen getoond worden in één enkel scherm maar ook een sequentie van meerdere schermen kan tot één interactiescenario behoren. In de volgende paragraaf ziet u hiervan een aantal voorbeelden.

Uitwerken van een scenario

Op basis van een catalogus van interactiepatronen kan een scenario uitgewerkt worden op basis van een selectie van deze interactiepatronen. Dit geeft een beeld wat onderliggende eisen zijn die aan een interactie gesteld kan worden.

In dit artikel gebruiken we het volgende scenario:

Het interactiescenario bestaat uit een urenregistratie. De toepassing biedt voor medewerkers de mogelijkheid om uren te boeken op één of meerdere projecten. Aan het einde van een periode dient een secretariaatmedewerker de uren te controleren en heeft deze persoon de

mogelijkheid om eventueel fouten te corrigeren. Scenario bestaat ruwweg uit de volgende stappen:

- *Selecteer medewerker*
- *Selecteer de projecten waarop deze medewerker uren geboekt heeft.*
- *Toon de activiteiten geboekt door deze medewerker op de projecten*
- *Biedt de mogelijkheid om eventueel activiteiten te corrigeren*
- *Sla de correcties op in het bestand (indien niet automatisch).*

Ter illustratie is voor dit scenario een eenvoudig voorbeeldprogramma ontwikkeld in MS-Access dat een viertal interactiepatronen uitwerkt als voorbeeld. Dit voorbeeld is te downloaden via de volgende hyperlink

<http://www.interactory.nl/frmFormManager.aspx?webpage=Interactiepatroon>. Deze voorbeeldtoepassing laat duidelijk een aantal verschillen zien tussen de interactiepatronen. Een voorbeeldapplicatie als deze kan gebruikt worden als een prototype in een evaluatiesessie waarbij in samenspraak met de gebruikers wordt geëvalueerd welk interactiepatroon het beste past bij de situatie.

Toepassen van Interactiepatronen

In de vorige hoofdstukken zijn we ingegaan op applicatie-architectuur en interactiepatronen. In dit hoofdstuk gaan we in op hoe Interactiepatronen toegepast kunnen worden in de praktijk.

Applicatie-ontwikkeling

Bij het ontwikkelen van applicaties kunnen interactiepatronen een bijdrage leveren aan de interne structuur van de toepassing. Worden een aantal interactiepatronen geselecteerd die toegepast gaan worden in de applicatie dan is het mogelijk om hiervoor bij de ontwikkeling van de interne structuur rekening mee te houden.

Zo kunnen voor één of meerdere interactiepatronen reeds bestaande besturingselementen of libraries aanwezig zijn. Deze kunnen het implementeren van een interactiepatroon sterk vereenvoudigen. Bijvoorbeeld het implementeren van het patroon "Movable panels", wat de gebruiker de mogelijkheid biedt om bepaalde elementen over een scherm te verplaatsen waardoor een eigen indeling gemaakt kan worden, is in Asp.Net eenvoudig te introduceren door het gebruik van webpart besturingselementen.

Een andere implementatie van interactiepatronen in applicatie-ontwikkeling bestaat uit het selecteren van scenario's op basis van interactiepatronen en deze op een dusdanige wijze te ontwikkelen dat er generieke formulieren en of pagina's ontstaan welke het scenario uitwerken zonder de detaillering van de specifieke situatie. Deze detaillering wordt bij implementatie op basis van parametrisering of data gedreven inrichting ingevuld. Een voorbeeld toepassing is het gebruik van master pages in Asp.Net. Deze zijn bedoeld voor het standaardiseren van de opmaak van webpagina's maar kunnen ook voor het introduceren van interactiepatronen gebruikt worden. Er kunnen namelijk naast opmaak ook op eenvoudige wijze complexere entiteiten toegevoegd zoals besturingselementen, gedrag en gebeurtenisafhandeling toegevoegd worden.

Een minder voor de hand liggende toepassing van interactiepatronen is het uitwerken van deze interactiepatronen tot besturingselementen en of componenten. De producten die ontstaan kunnen beschikbaar worden gesteld als commercieel product. Zie bijvoorbeeld producten als devexpress waarbij al een aantal producten aanwezig zijn die een implementatie zijn van één of meerdere interactiepatronen.

Houdt er rekening mee dat de gekozen technologie en de gekozen interactiepatronen sterk aan elkaar gerelateerd kunnen zijn. Sommige interactiepatronen kunnen alleen geïmplementeerd worden als aan specifieke eisen van hardware en of besturingssysteem voldaan wordt.

Bijvoorbeeld het interactie patroon "Animated transition" is in een besturingssysteem als MacOS of Windows eenvoudig te realiseren. Echter dit toepassen in een web based applicatie kan een uitdaging zijn voor de applicatie ontwikkelaars.

Het kan dan ook handig zijn om bij de introductie van scenario's op basis van interactiepatronen samen met een aantal applicatie ontwikkelaars om tafel te gaan zitten en een score te maken van haalbaarheid en complexiteit van deze scenario's. Dit kan voorkomen dat tijdens de ontwikkeling van generieke componenten of de applicatie inrichting veel tijd verloren gaat met het technisch inrichten van dit patroon. Terwijl de inrichting van het scenario met een eenvoudiger te implementeren interactiepatroon ook volstaan had.

Architectuurrichtlijnen

Niet alleen bij applicatie-ontwikkeling kunnen interactiepatronen een rol spelen. Ook op een hoger abstractieniveau is het mogelijk om interactiepatronen in te zetten. Patronen bieden structuur aan de wijze waarop gebruikers met informatiesystemen interacteren. Ook architectuurrichtlijnen geven hiervoor in een aantal gevallen kaders af. Bijvoorbeeld in de NORA worden een aantal bouwstenen beschreven die op adequate wijze uitgewerkt kunnen worden op basis van interactiepatronen. Denk bijvoorbeeld aan het uitwerken van bouwstenen als e-formulieren en een persoonlijke internet-pagina voor burgers.

Echter ook het uitwerken van richtlijnen die gesteld worden in een architectuur kunnen bij gebruik van een interactiepatroon catalogus goed ingevuld worden. Wederom een voorbeeld op basis van NORA. Hierbij worden richtlijnen gegeven omtrent transparantie. Voor een aantal van deze richtlijnen kunnen scenario's op basis van interactiepatronen uitgewerkt worden. Voordeel is dat hierdoor bij implementatie van deze scenario's door verschillende overheden de opzet voor de burger herkenbaar zal zijn wat weer tegemoet komt aan de eis van administratieve lastenverlichting voor de burger.

Interactieve ontwerpessies

Interactieve ontwerpessies zijn een krachtig middel om in een iteratief en interactief ontwerptraject applicatie requirements op te stellen in samenspraak met de toekomstige gebruikers. Toekomstige gebruikers vinden het veelal moeilijk om vooraf aan te geven waaraan applicatie interacties moeten voldoen. Maar men is wel goed in staat wat fout is in een opzet van bestaande applicatie interacties.

Dit uitgangspunt van negatief valideren kan ingezet worden met behulp van interactiepatronen. Zo is het mogelijk om voor een interactieve ontwerpessie een aantal patronen te selecteren en te combineren tot een scenario. Dit kan meestal eenvoudig van opzet zijn mits het toegelicht wordt in de ontwerpessie. Door de uitgewerkte scenario in een groepsdiscussie te evalueren en bij te stellen (zie de paragraaf modelleer- en werkwijze) ontstaat de gebruikersinterface iteratief.

Ervaringen bij deze werkwijze zijn veelal verrassend maar positief. Enerzijds blijkt dat men de interactiepatronen kiest die vanuit het perspectief van een applicatie ontwikkelaar niet verwacht zijn. Opvallend is dat gebruikers meestal voor eenvoudige interactiepatronen kiezen die herkenbaar zijn. Denk hierbij aan patronen als "extra on demand", "titled sections" en "responsive disclosure".

Positief aspect is dat de deelnemende gebruikers zich betrokken voelen bij het ontwerpproces. Dat blijkt uit de bereidheid om te zoeken naar alternatieven, acceptatie van eventuele beperkingen maar ook uit het vervullen van een ambassadeursrol in de gebruikersorganisatie.

Tot slot

Een applicatie-architectuur op basis van interacties biedt bij een iteratief ontwikkeltraject een aantal duidelijke voordelen. Met name het gebruik van interactiepatronen en het betrekken van toekomstige gebruikers bij het ontwikkeltraject dragen bij aan een betere acceptatie bij implementatie.

Er is echter een aandachtspunt bij het gebruik van interactiepatronen. Met name de kans op stilstand in de ontwikkeling van interacties en interactiepatronen is vermeldenswaardig. Evaring leert dan men eerder voor bekende interactiepatronen kiest, voornamelijk omdat men deze goed kan matchen met de gewenste oplossing. Bij vernieuwende interactiepatronen is

deze matching moeilijker. Om dit te ondervangen kan men overwegen om deze interacties in een prototype in detail uit te werken en de gebruikers de mogelijkheid te bieden dit te evalueren in een testomgeving.

Referenties

- [Cooper, 2003] Alan Cooper: *About face 2.0, essentials of interaction design*, Wiley publishing, 2003
- [Dingemans, 2008] Bert Dingemans, *www.interactory.nl*.
- [Hoogendoorn, 2009] Sander Hoogendoorn: *Serviceorientatie vormgeven met smart use cases*, Sdu, 2009.
- [Tidwell, 2005] Jennifer Tidwell: *Designing interfaces* (<http://designinginterfaces.com>), o'Reilly, 2005.
- [Welie, 2008] Martijn van Welie: <http://www.welie.com/patterns> (*A pattern library for Interaction Design*, 2008).

Genoemde interactiepatronen

- Movable panels: http://www.designinginterfaces.com/Movable_Panels
- Animated transition: http://www.designinginterfaces.com/Animated_Transition
- Extras on demand: http://www.designinginterfaces.com/Extras_On_Demand
- Titled sections: http://www.designinginterfaces.com/Titled_Sections
- Responsive disclosure: http://www.designinginterfaces.com/Responsive_Disclosure
- Jump to Item: http://www.designinginterfaces.com/Jump_to_Item