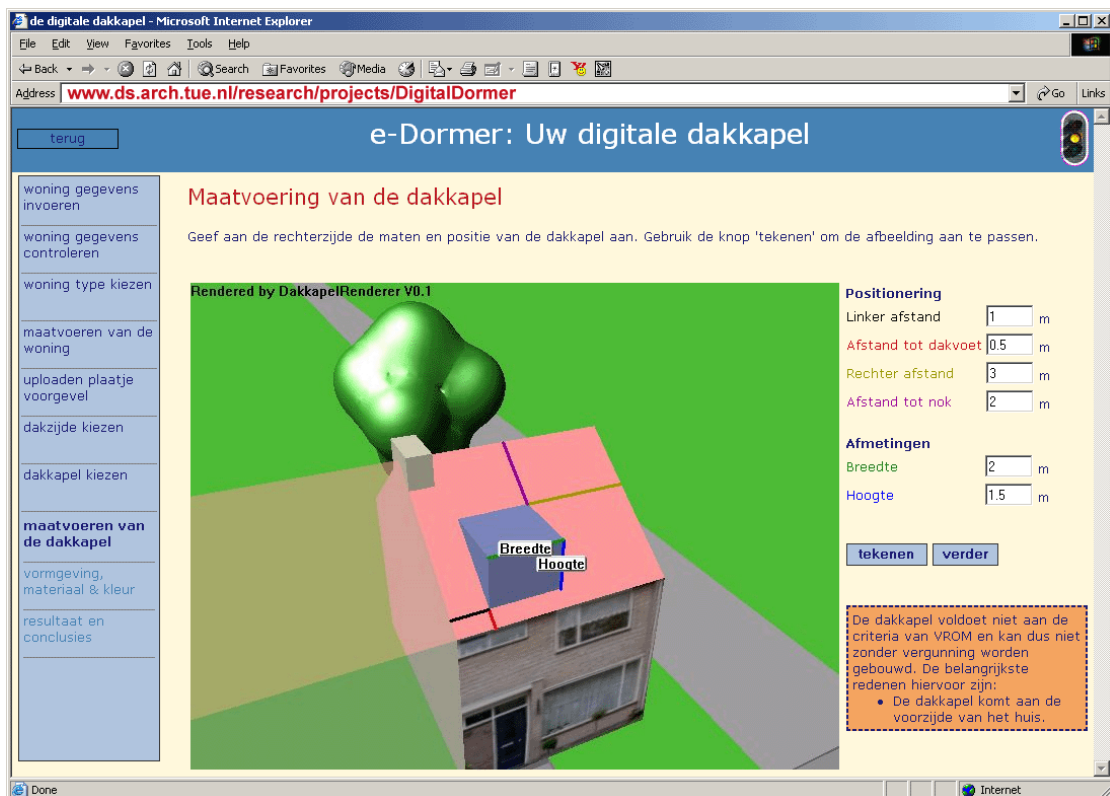


De Digitale Dakkapel

een online toets voor dakkapellen

Eindrapport over fase 1 van het project
16 juni 2004



De Digitale Dakkapel

een online toets voor dakkapellen

Eindrapport over fase 1 van het project
16 juni 2004

Auteurs: dr.ir. Jos P. van Leeuwen en ir. Joran Jessurun

Technische Universiteit Eindhoven
Faculteit Bouwkunde
Leerstoel Ontwerp Systemen
Postbus 513 – VRT 9.11
5600 MB Eindhoven

<http://www.ds.arch.tue.nl>

Opdrachtgevers: ing. Edward de Wit
Gemeente Rotterdam

Gemeente Zoetermeer

Projectleiding: Ok van Megchelen
Stichting SAV€

Financiering: InAxis – commissie innovatie openbaar bestuur
Ministerie van Binnenlandse Zaken en Koninkrijksrelaties

TU/e technische universiteit eindhoven





De Digitale Dakkapel?

De Digitale Dakkapel is een webapplicatie die bestaat uit een verzameling webpagina's met dynamische functionaliteit die het de bezoeker van de pagina's mogelijk maakt om met eenvoudige interactie een dakkapel te ontwerpen op zijn woning. De gebruiker van de applicatie kan de dakkapel plaatsen op een vereenvoudigd 3D geometrisch model van zijn woning. Het geometrische model van de woning wordt gegenereerd op basis van selectie door de gebruiker van een basistype woning en het invoeren van een klein aantal afmetingen van de woning. De applicatie toont de woning met de dakkapel in een visualisatie. Gaande het 'ontwerpproces' geeft de applicatie, op basis van de gegevens die de gebruiker verstrekt, informatie over (a) de noodzaak tot het aanvragen van een bouwvergunning voor deze dakkapel en (b) of aan de criteria wordt voldaan die zijn gesteld in de gemeentelijke welstandsnota.

Dit document bevat de eindrapportage van fase 1 (*proof of concept*) van het project. De Digitale Dakkapel applicatie die in fase 1 is ontwikkeld vindt u gedurende de looptijd van het project op de volgende website:

<http://bwds04.campus.tue.nl>

Daar kan ook een demonstratie van de applicatie worden gedownload.

Inhoudsopgave

1.	Doelstellingen	5
2.	Ontwikkelingen in fase 1	5
3.	Levering en beschikbaarheid	6
4.	Functionaliteit	7
4.1	Bepalen van de welstandscriteria op basis van de postcode	7
4.2	Selecteren van het woning/daktype	7
4.3	Dimensioneren van de woning	7
4.4	Selecteren en plaatsen van de dakkapel	7
4.5	Visualiseren van woning en dakkapel	8
4.6	Toetsen van de ingevoerde gegevens	8
4.7	Functionaliteit die geen onderdeel uitmaakte van de opdracht	9
5.	Implementatie & documentatie	10
5.1	Ontwikkelingsproces van de applicatie	10
5.2	Uitgangspunten voor de applicatie	11
6.	Aanbevelingen voor vervolgfases	12
6.1	Koppeling met gemeentelijke gegevensbanken	12
6.2	Aanvullende functionaliteit voor vergunningsaanvraag	12
6.3	Verbetering van de gebruikersinterface	13
6.4	Uitbreidingen van de doelstelling	15
6.5	Verspreiding, exploitatie en beheer	15
7.	Toekomstperspectief	18
7.1	De toekomst van digitaal toetsen	18
7.2	De rol van overheid en commercie	19
8.	Bijlage	21
8.1	Parametriseren van geometrie	21
8.2	Onderlinge afhankelijkheden van de parameters	24
8.3	Digitale toetsingscriteria	25
8.4	Referenties	26

1. Doelstellingen

De applicatie heeft als doelgebruiker de burger die voornemens is een vergunning aan te vragen voor het plaatsen van een dakkapel op zijn woning. Met behulp van de applicatie kan de burger snel een beeld krijgen van hoe de woning er met dakkapel uit zal zien. Daarnaast geeft de applicatie direct informatie over de mate waarin de gewenste dakkapel voldoet aan de criteria van respectievelijk VROM en de gemeentelijke welstandsnota. Op basis van de VROM criteria kan worden bepaald of een bouwvergunning voor de dakkapel moet worden aangevraagd. Op basis van de criteria uit de welstandsnota (in het geval van Rotterdam zijn dat de zogenaamde loketcriteria), kan *indicatief* worden aangegeven of de welstandscommissie het voorstel zal aanvaarden.

Uitgaande van bovenomschreven doelstelling zijn voor de applicatie die in dit project is ontwikkeld een aantal hoofdeigenschappen gedefinieerd.

- De applicatie is eenvoudig toegankelijk via Internet, zonder ingewikkelde installatieprocedures.
- De dakkapel wordt grafisch zichtbaar gemaakt, maar de applicatie is toch eenvoudig door leken te bedienen, zonder kennis over tekensoftware.
- De applicatie is flexibel van opzet opdat de grenzen waarbinnen de applicatie werkt kunnen worden verruimd en aangepast, betreffende de soorten woningen en dakkapellen en de inhoud van de criteria.
- Er is een goede balans tussen het realisme van de representatie van de woning met dakkapel enerzijds en de mate van benodigde interactie van de gebruiker met de applicatie anderzijds.
- Er wordt zo direct mogelijk getoetst tijdens de interactie met de applicatie met heldere feedback over het resultaat van de toetsing op de individuele criteria.
- Voor zover mogelijk wordt de applicatie ontwikkeld op basis van open standards en met middelen die als open source beschikbaar zijn.

2. Ontwikkelingen in fase 1

De doelstelling van fase 1 kan worden aangeduid met de term *proof of concept*. Dit betekent dat in fase 1 op overtuigende wijze wordt aangetoond dat de ten doel gestelde applicatie technisch realiseerbaar is en in praktisch zin haalbaar. Normaliter wordt zo'n proof of concept geleverd, nadat een functioneel en technisch ontwerp van de applicatie is gemaakt, door middel van een zogenaamde *mock-up*, een demonstratie van de interface van de te ontwikkelen software, met daarbij afdoende technische onderbouwing van de wijze van implementatie.

In dit project is een andere benadering gekozen die geleid heeft tot een resultaat dat verder ligt dan een proof of concept. De volgende activiteiten zijn ontplooid in fase 1:

1. Beschrijving van het functioneel/technisch ontwerp dat tevens als basis diende voor de subsidieaanvraag (zie hoofdstuk 4)
2. Het functioneel ontwerp is besproken met direct en indirect betrokken partijen:
 - Opdrachtgevende gemeenten
 - Stichting SAVE
 - Stichting Dorp, Stad en Land
 - Software ontwikkelaars die geïnteresseerd zijn in vervolgfases (De Twee Snoeken) en die betrokken zijn bij aan het onderwerp gerelateerde

projecten (De Twee Snoeken en UGS PLM Solutions in het project 'Centrale Server Bouwaanvragen')

3. Het technische ontwerp is besproken met De Twee Snoeken in vrijblijvend overleg over fase 1 en vervolgfases.
4. Er is onderzoek verricht naar de technische mogelijkheden en de beschikbaarheid van open source software ter ondersteuning en ontwikkeling van het project.
5. Door middel van prototyping en exploratief ontwikkelen van deelfunctionaliteit is gekomen tot de implementatie van een softwaretechnisch raamwerk dat de basis vormt voor de applicatie. Dit raamwerk bestaat uit een tweetal server applicaties: de DakkapelRenderServer en de DakkapelWebServer (zie hoofdstuk 5).
6. Voorts is de gebruikersinterface ontwikkeld en in een aantal sessies met de direct en indirect betrokken partijen geëvalueerd en bijgesteld.
7. Tot slot is middels demonstraties en discussies tijdens diverse gelegenheden terugkoppeling gevonden vanuit bouwpraktijk en overheden. Onder meer is het prototype gedemonstreerd tijdens de BouwRAI te Amsterdam en tijdens een bezoek aan het Ministerie van VROM. De ontvangen terugkoppeling is verwerkt in de aanbevelingen in hoofdstuk 6.

3. Levering en beschikbaarheid

De oplevering van fase 1 van dit project bestaat uit:

1. De functionaliteit van de volledige applicatie in de vorm van een online webserver die operationeel zal zijn tot 31 december 2004.
2. Een CD-Rom met daarop:
 - a. een volledige kopie van broncode en gecompileerde versies van de software.
 - b. alle projectgegevens die tijdens de uitvoering van het project zijn verzameld dan wel gegenereerd
 - c. alle documentatie die bij de broncode is geschreven voor de overdracht van het project aan derden ten behoeve van de uitvoering van vervolgfases
3. Dit eindrapport.

Over alle hierboven genoemde onderdelen berust het copyright bij de opdrachtgevers. De Technische Universiteit Eindhoven behoudt zich het recht voor om:

- De applicatie voor eigen doeleinden operationeel te houden en te demonstreren.
- De broncode voor doelstellingen van onderzoek en onderwijs te hergebruiken.

Bovendien worden toekomstige ontwikkelingen door derden op basis van de geleverde software voor bovengenoemde doeleinden beschikbaar gesteld aan de Technische Universiteit Eindhoven.

4. Functionaliteit

De functionaliteit van de applicatie is beschreven in een zestal onderdelen welke het uitgangspunt hebben gevormd voor de implementatie van het proces van interactie tussen gebruiker en webapplicatie.

4.1 *Bepalen van de welstandscriteria op basis van de postcode*

De gebruiker voert de postcode en huisnummer in van zijn woning. De applicatie zoekt hiermee op welke welstandscriteria van toepassing zijn. Dit onderdeel van de functionaliteit wordt geleverd door een nader te ontwikkelen webservice welke geen deel uitmaakt van deze opdracht. Deze webservice levert, op basis van de aangeleverde postcode en huisnummer, de criteria in een digitaal formaat dat wel binnen deze opdracht is vastgesteld.

In deze eerste stap kunnen ook algemene gegevens van de woning gevraagd. In fase 1 is dit beperkt tot de vraag of de woning is aangemerkt als beschermd monument.

4.2 *Selecteren van het woning/daktype*

De gebruiker selecteert uit een beperkt aantal woning/daktypen het type dat het meest lijkt op de eigen woning. De aanname hierbij is dat met behulp van zeven woning/daktypen ongeveer 80% van het woningbestand kan worden gemodelleerd. Het keuzeaanbod wordt in grafische vorm aan de gebruiker getoond. Na deze keuze kan door de applicatie op basis van vooraf bepaalde parameters per woningtype een eenvoudig driedimensionaal geometrisch model worden gegenereerd en aan de gebruiker worden getoond.

4.3 *Dimensioneren van de woning*

Terwijl het geometrisch model wordt getoond, geeft de gebruiker de dimensies van zijn woning aan. Dit doet hij door gevraagde maten numeriek in te voeren waarna de grafische representatie van de woning door de applicatie wordt aangepast. Hoewel in fase 1 van deze ontwikkeling 1 woningtype is uitgewerkt, is het met deze methode mogelijk meerdere en complexe types te realiseren. Het woningtype dat in fase 1 is uitgewerkt heeft een rechthoekige plattegrond en een symmetrisch zadeldak.

4.4 *Selecteren en plaatsen van de dakkapel*

Wanneer de maten van de woning zijn ingevoerd kan een type dakkapel worden geselecteerd, geplaatst en gedimensioneerd:

1. Selecteren van het type dakkapel. Er kan een keuze worden gemaakt uit een beperkt aantal eenvoudige typen. In fase 1 is 1 type dakkapel uitgewerkt, namelijk een dakkapel met een plat dak.
2. Selecteren van het dakvlak waarop de dakkapel moet worden geplaatst.
3. Positioneren en maatvoeren vinden plaats door numerieke invoer van maten. De grafische representatie wordt door de applicatie aangepast.
4. Invoeren van andere parameters, zoals materiaalgebruik en kleuren, door de gebruiker. In fase 1 zijn niet alle parameters uitgewerkt, wel is van elk te verwachten type parameter er minimaal 1 opgenomen.

4.5 Visualiseren van woning en dakkapel

De dakkapel en het model van de woning worden steeds gevisualiseerd op een stilistische manier die in architectonische presentaties wel “artist’s impression” wordt genoemd. Omdat het gehele model van de woning bekend is, kan de applicatie meerdere aanzichten genereren. Om het realisme van de representatie te verhogen maakt de applicatie het mogelijk aan de gebruiker om een digitale foto van de voorgevel van de woning te sturen naar de webapplicatie, waarna deze kan worden ‘geplakt’ op het 3-dimensionale geometrische model.

4.6 Toetsen van de ingevoerde gegevens

Op basis van de ingevoerde gegevens worden, voor de gebruiker op de achtergrond, continu drie toetsen uitgevoerd:

1. De basis toets
Dit is een geometrische toets waarin wordt gecontroleerd of de maatvoering van woning en dakkapel logischerwijs correct is (bv. maten > 0 , goothoogte $<$ nokhoogte, breedte dakkapel $<$ breedte woning, etc.).
2. De VROM toets
Deze toets bestaat uit een combinatie van geometrische en niet-geometrische criteria. Wanneer aan deze criteria wordt voldaan, geeft de applicatie aan dat de dakkapel zonder aanvraag van bouwvergunning kan worden gebouwd. Indien niet wordt voldaan aan de criteria wordt, met opgave van de reden, gemeld dat een vergunning moet worden aangevraagd.
3. De Welstandstoets
Deze toets is vergelijkbaar met de VROM toets, maar bevat meer criteria. De gemeente Rotterdam noemt dit de loketcriteria. Deze criteria zijn in fase 1 van het project geïmplementeerd. Wanneer aan deze criteria wordt voldaan meldt de applicatie dat de welstandscommissie waarschijnlijk akkoord zal gaan met het verlenen van een bouwvergunning. Wanneer niet aan de criteria wordt voldaan meldt de applicatie dat een akkoord van deze commissie niet zeker is.

Deze drie toetsen zijn onderling afhankelijk zoals in onderstaande tabel is aangegeven.

	<i>voorwaarde voor uitvoering</i>	<i>indien voldaan</i>
Basis toets	geen voorwaarde, deze toets wordt altijd uitgevoerd	de gegevens zijn logischerwijs correct
VROM toets	de ingevoerde gegevens voldoen aan de basis toets	de dakkapel kan vergunningsvrij worden gebouwd
Welstandstoets	de ingevoerde gegevens voldoen <i>niet</i> aan de VROM toets (de dakkapel is niet vergunningsvrij)	Welstand gaat waarschijnlijk akkoord

Om de toetsing mogelijk te maken zijn de diverse criteria in een digitale vorm geformuleerd. In bijlage 8.3 is een overzicht van de geïmplementeerde criteria opgenomen.

Tijdens het doorlopen van het ‘ontwerpproces’ met de applicatie zullen de drie toetsen in genoemde volgorde bij elke stap worden uitgevoerd, indien relevant.

De applicatie geeft een visuele indicatie van de uitkomst van de drie toetsen middels een verkeerslicht:

Groen	VROM toets doorstaan, dakkapel is vergunningsvrij
Oranje	VROM toets niet doorstaan, Welstandstoets van toepassing
Oranje knipperend	Basis toets niet doorstaan, logische fout in invoer
Rood	Welstandstoets niet doorstaan

4.7 Functionaliteit die geen onderdeel uitmaakte van de opdracht

De applicatie zoals ontwikkeld in fase 1 biedt geen functionaliteit om de gegevens van bezoekers van de pagina’s te administreren. Dit betekent dat tijdens de sessie alle gegevens wel beschikbaar zullen zijn, maar dat deze data niet kunnen worden opgeslagen voor hergebruik tijdens een volgend bezoek.

Het vinden van de welstandscriteria op basis van de ingevoerde postcode en huisnummer is een service die niet binnen deze opdracht ontwikkeld is, maar waarvan wordt geacht dat deze in de toekomst beschikbaar zal zijn.

Verder ontbreekt de daadwerkelijke doorloop van de verzamelde informatie naar het aanvraagformulier, ter beperking van onnodig dubbelwerk. De mogelijkheid van een gestructureerde output is wel aanwezig. Zodra het formaat van het digitale aanvraagformulier beschikbaar is, en wellicht gestandaardiseerd, kan hierop worden aangesloten.

5. Implementatie & documentatie

Het prototype van de applicatie zoals in fase 1 ontwikkeld bestaat uit twee deelapplicaties:

1. De DakkapelRenderServer
Dit is een intern proces waarin de berekeningen, toetsingen, en visualisaties worden uitgevoerd.
2. De DakkapelWebServer
Dit is een webapplicatie die de webpagina's genereert en de interactie met de gebruiker verzorgt.

Deze twee applicaties communiceren onderling om de benodigde en gegenereerde gegevens te delen.

In de ontwikkeling is voornamelijk gebruik gemaakt van de volgende software en technologieën:

• Microsoft Windows 2000 Server operating system	C
• Apache webserver met PHP programmeeromgeving voor dynamische webpagina's	O
• XML bestandsformaat voor gegevensuitwisseling	O
• 3D Studio Max van Autodesk voor het genereren van de basisgeometrieën	C
• Open Scene Graph met Perl script omgeving voor het parametriseren van de geometrieën en het produceren van de grafische representaties	O
• Microsoft Visual Studio 2003 voor ontwikkeling van de DRS applicatie	C

C = Commercieel beschikbare software

O = Open source software of open standaard

In fase 1 is alleen voor de ontwikkeling commerciële software gebruikt. De resulterende applicaties kunnen met alleen Open Source software functioneren.

5.1 Ontwikkelingsproces van de applicatie

Het ontwikkelingsproces heeft in grove lijnen de volgende stappen doorlopen:

1. Ontwikkeling van een algemene methode voor parametrische beschrijvingen en grafische representatie van geometrieën met behulp van Open Scene Graph (OSG).
2. Modelleren van woningtype en dakkapeltype in 3D Studio Max.
3. Parametrisering van de geometrieën in OSG.
4. Parametrisering van niet-geometrische beschrijvingen.
5. Ontwikkeling van de render functionaliteit.
6. Ontwikkeling van de logica voor interactie tussen gebruiker en geometrie.
7. Definitie van configuratiebestanden en het formaat voor gegevensuitwisseling tussen de twee applicaties.
8. Ontwikkeling van de gebruikersinterface en het 'ontwerpproces' in de webapplicatie.
9. Definitie van het formaat voor de digitale criteria.

10. Ontwikkeling van de digitale toetsingsmodule.

5.2 Uitgangspunten voor de applicatie

Een belangrijk uitgangspunt bij alle onderdelen van de implementatie is de flexibiliteit van de software. Een hoge mate van flexibiliteit is bereikt doordat de processen welke zijn geïmplementeerd in de software los staan van de benodigde gegevens en, belangrijker nog, los staan van allerlei vooraf gedefinieerde aannames. Door deze benaderingswijze is het nu mogelijk om zonder aanpassing van de software een groot deel van de functionaliteit te wijzigen, zoals:

- De geometrieën en hun grafische representatie, qua textuur, belichting, vorm, camerastandpunt, etc.
- De parametrisering van de geometrieën en de niet-geometrische beschrijvingen.
- De onderlinge afhankelijkheden van de parameters.
- De inhoud van de toetsingscriteria en de bijbehorende teksten voor terugkoppeling.

Bovendien is het hiermee betrekkelijk eenvoudig om in de gebruikersinteractie wijzigingen aan te brengen of nieuwe onderdelen toe of in te voegen.

6. Aanbevelingen voor vervolgfases

6.1 Koppeling met gemeentelijke gegevensbanken

1. Opvragen van adresgegevens op basis van postcode

De postcode en het huisnummer worden in de huidige versie opgevraagd, maar nog niet gebruikt om de adresgegevens van de gebruiker te controleren. Hiertoe moet een koppeling tot stand worden gebracht met een (landelijke) databank van adresgegevens. Deze zijn waarschijnlijk commercieel beschikbaar. Indien adresgegevens ook beschikbaar zijn in de gemeentelijke databestanden zou ook daar een controle kunnen plaats vinden (zie punt 2 hieronder).

2. Inlezen van gemeentelijke welstandscriteria op basis van postcode.

Afhankelijk van hoe de applicatie uiteindelijk zal worden ingezet (landelijk of gemeentelijk) moet op basis van de postcode worden bepaald:

- In welke gemeente de gebruiker woont (zie punt 1 hierboven).
Dit is van belang wanneer de applicatie landelijk wordt ingezet. Afhankelijk van de postcode wordt de gemeentelijke set criteria ingelezen. De huidige versie van de software gaat uit van gemeentelijke inzet van de applicatie met één enkele set welstandscriteria.
- In welk gebied binnen de gemeente de gebruiker woont.
De informatie over het gebied binnen de gemeente zal uit gemeentelijke databases moeten komen. In de gemeente Rotterdam zijn aanvullende bepalingen gekoppeld aan gebiedstypen. Deze bepalingen zijn echter niet geformuleerd in digitaal toetsbare vorm. Waarschijnlijk is het dan ook beter de gebruiker alleen op deze aanvullende bepalingen te wijzen en geen poging te doen hier digitaal te toetsen.

3. Inlezen van bestemmingsplangegevens op basis van postcode en huisnummer.

In de huidige applicatie is er nog geen koppeling met gegevens uit een gemeentelijk bestemmingsplan. Een dergelijke koppeling is wel relatief eenvoudig toe te voegen, bijvoorbeeld in de vorm van een vierde type toets. Hiervoor is aanpassing van de software weliswaar noodzakelijk, maar er zijn daarbij geen complicaties te verwachten. Voorwaarde hiervoor zijn:

- De eisen betreffende dakkapellen zijn in het bestemmingsplan digitaal aanwezig zijn.
- Deze eisen kunnen op basis van postcode en huisnummer (direct of indirect) worden opgevraagd.
- Om de informatie uit bestemmingsplan op te vragen op basis van de postcode en huisnummer is het nodig dat er een koppeling wordt gemaakt tussen een postcodebestand en de GIS data in het bestemmingsplan.

6.2 Aanvullende functionaliteit voor vergunningsaanvraag

1. Aanleveren van tekeningen voor de vergunningsaanvraag

Op basis van het geparmetriseerde model van woning en dakkapel is het in principe mogelijk om geschematiseerde bouwtekeningen van de dakkapel aan te leveren. Het

verdient aanbeveling om hierbij te streven naar vereenvoudiging, omdat het maken van een volledig geparametriseerde detail-dakkapel waarschijnlijk niet haalbaar is. Een werkbaar alternatief is het gebruik van principe details. Op basis van door de gebruiker aangegeven parameters kan een selectie uit vooraf gedefinieerde details worden gemaakt. De parameters die in de huidige versie al worden bepaald zijn hiertoe nog niet toereikend. Extra benodigde parameters zijn in ieder geval:

- de hoogte van de bovenkant van de verdiepingsvloer waarop de dakkapel wordt geplaatst;
- de opbouw van de dakconstructie;
- afgeleid daarvan: de locatie van de vrije hoogtelijn onder de kap.

Voor het aanleveren van een gewijzigde plattegrond van de betreffende verdieping kan gebruik worden gemaakt van een ingescande versie van een bestaande plattegrond die door de gebruiker via upload naar de applicatie wordt gestuurd (analoog aan de upload van de gevel-foto).

2. Opslag van gebruikersgegevens in server database

In de huidige versie worden de gegevens van de gebruiker en de ontworpen dakkapel niet bewaard. Ook is er geen mogelijkheid om deze gegevens op de client op te slaan. Het is echter wel wenselijk dat gebruikers hun ‘ontwerp’ tijdelijk kunnen bewaren. De eenvoudigste oplossing is een database te definiëren aan server-zijde waarin de sessiegegevens worden opgeslagen. Het verdient aanbeveling om voor deze ontwikkeling op termijn aansluiting te zoeken bij een eventuele algemene service voor het indienen van bouwvergunningsaanvragen.

Wanneer een koppeling tot stand wordt gebracht met een server voor het indienen van de vergunningsaanvraag, zal door die server authenticatie van de aanvrager moeten worden uitgevoerd, bijvoorbeeld door middel van een elektronische handtekening.

3. Invoer van gegevens van de dakkapel van de burelen

Eén van de welstandscriteria betreft de hoogte van de dakkapel in vergelijking met die van de burelen. Dit criterium wordt nu niet getoetst omdat de benodigde informatie niet wordt opgevraagd. Overwogen moet worden of en op welke wijze deze informatie wel gevraagd kan worden. Toetsing is bij aanwezigheid van deze informatie eenvoudig zonder aanpassing van de toetsingssoftware.

6.3 Verbetering van de gebruikersinterface

1. Harmonisering van de gebruikte terminologie met de wetsteksten

Bij de evaluaties bleek dat op een aantal plaatsen in de interface termen zijn gebruikt die afwijken van de begrippen die in de wet worden gehanteerd. Een voorbeeld hiervan is de term goothoogte. In het bouwbesluit is van goothoogte geen sprake maar wordt de term dakvoet gehanteerd. Hoewel in sommige gevallen de formele term niet gebruiksvriendelijker is in de context van deze applicatie, lijkt het toch verstandig om analoog te blijven met de wetstekst en door middel van uitleg in de interface e.e.a. toe te lichten waar dit nodig wordt geacht.

2. Meer dynamische camera standpunten

In de huidige versie van de software hebben de camera's van waaruit de geometrie wordt gevisualiseerd vaste standpunten. Het is wenselijk dat deze camerastandpunten

en de gebruikte lens automatisch worden aangepast afhankelijk van de grootte en plaats van de woning en dakkapel. Hiertoe is het nodig de software aan te passen en de eigenschappen van de camera te bepalen op basis van de reeds bekende parameters.

3. Verbeterd stappen menu

De plaats en vorm van het menu van stappen dat de gebruiker doorloopt (nu in verticale vakken aan de linkerkant van de webpagina's) moet worden herbezien. Onderzocht moet worden of een horizontaal menu meer ruimte schept maar nog voldoende leesbaar en begrijpbaar kan worden vormgegeven.

4. Positionering van de terugkoppeling

De terugkoppeling over het resultaat van de diverse toetsen wordt in de huidige applicatie geplaatst onder de invoervelden. Onderzocht moet worden of een andere positionering meer helderheid schept en wellicht ook of een meer beknopte omschrijving van de terugkoppeling mogelijk is zonder verlies aan duidelijkheid en eenduidigheid.

5. Grafische interactie voor bepalen van maten woning en dakkapel

De parameters worden in de huidige applicatie alleen door middel van numerieke interactie of selectie van opties bepaald. Een verbetering is mogelijk wanneer de bepaling van maten en posities grafisch kan plaatsvinden. Een technische oplossing hiervoor is onderzocht maar niet uitgewerkt, welke bestaat uit het absoluut positioneren via javascript van 1x1 pixel DIV's met een zwarte achtergrond in de HTML pagina.

6. Upload foto van achtergevel

In de huidige versie kan alleen een foto van de voorgevel in de geometrie worden geplakt. Wenselijk is om ook foto's van achter- en eventueel zijgevels te kunnen plakken, vooral wanneer de dakkapel aan de achterzijde wordt geplaatst.

7. Tekening van begane grondvloer uploaden en plakken in de geometrie

Een gescande tekening van de bestaande plattegrond van de verdiepingsvloer, zoals omschreven bij punt 1 in sectie 6.2, kan ook worden ingeplakt in de geometrie. Vervolgens kan op basis hiervan de dakkapel in de nieuwe situatie schematisch worden ingetekend. Ook kunnen hoogtelijnen worden aangegeven, op basis van de parameters van de woning en de gegevens over de dakopbouw.

8. Tekenen van lijnen in het dakvlak die de grenzen van de welstandscriteria aangeven

In het gekozen dakvlak kunnen lijnen worden getekend die aangeven waarbinnen een dakkapel nog voldoet aan de welstandscriteria. De posities van deze lijnen zijn afhankelijk van:

- de parameters van de woning
- de bebouwing of het gebruik van begrenzendende percelen

6.4 Uitbreidingen van de doelstelling

1. Best practices

Om te komen tot een verbetering van wat men zoal wenst te bouwen als dakkapel zou door middel van de dakkapel website vanuit welstand een set aanbevelingen kunnen worden aangeboden aan de aanvrager. Het aanbieden van een aantal algemene aanbevelingen is waarschijnlijk al mogelijk zonder afhankelijkheid van de door de gebruiker ingevoerde gegevens. Mochten sommige aanbevelingen wel afhankelijk zijn van keuzes die de gebruiker maakt, dan kan de applicatie worden uitgebreid met een onderdeel dat deze aanbevelingen relateert aan de ingevoerde parameters.

2. Trendzeters

Een reeds bestaande vorm van de bij punt 1 genoemde aanbevelingen is in feite het principe van de trendzetter zoals in de gemeente Rotterdam momenteel wordt gehanteerd. Op basis van de postcode en huisnummer kan, indien de benodigde gegevens over trendzeters digitaal voorradig zijn, worden bepaald welke trendzeters in de omgeving van de woning bekend zijn. De applicatie kan vervolgens de mogelijkheid bieden aan de gebruiker om deze trendzetter te volgen. Naar mate er meer informatie over de trendzetter beschikbaar is (bouwtekeningen, parameters, vergunningsaanvraag) kunnen deze gegevens worden gebruikt om het proces van vergunningsaanvraag te versnellen. Of en hoe het mogelijk is om te toetsen of een dergelijke trendzetter wel past in de bestaand situatie van de gebruiker dient nog onderzocht te worden.

6.5 Verspreiding, exploitatie en beheer

(Door Ok van Megchelen, Stichting SAV€)

1. algemeen

De oplevering van deze applicatie luidt een volgende fase in, waarbij ter voorbereiding van fase 2 en 3, in het bijzonder wordt stilgestaan bij de verspreiding, de exploitatie en het beheer.

Het vertrekpunt hierbij is dat de voor ogen staande ketengerichte applicatie, in combinatie met een aantal aangrenzende digitale voorzieningen, is gepositioneerd op het openbare en dynamische speelveld tussen overheid en bedrijfsleven. Deze positie is in hoge mate bepalend voor de wijze, waarop het beste vorm en inhoud kan worden gegeven aan de organisatie, de financiering, de verspreiding, de exploitatie en het beheer ervan.

2. Verspreiding (adoptie)

Het uiteindelijke nut van de op te leveren applicatie wordt niet alleen afgemeten aan de schoonheid van de oplossing en het enthousiasme waarop hierop wordt gereageerd. Het succes wordt uiteindelijk bepaald door de mate waarin deze in de praktijk wordt ingezet bij het voorbereiden en indienen van vergunningaanvragen.

Het succes van deze ketengerichte applicatie wordt bepaald en beperkt door het draagvlak en de adoptie door partijen als VROM, de gemeenten en het bedrijfsleven. Hierbij is het op de eerste plaats van belang, dat gemeenten bereid zijn om deze verandering te gaan inzetten.

De verwachting is dat de snelheid waarmee gemeenten deze vernieuwing kunnen oppakken om uiteenlopende redenen niet hoog zal zijn.

Dit proces kan worden versneld als onder regie van het Ministerie van VROM een definitieve applicatie ter beschikking wordt gesteld van gemeenten en bedrijfsleven. Deze faciliteit kan dan worden gezien als aanvulling op de huidige digitale dienstverlening van VROM rondom bouwvergunningen, die is gericht op het verlagen van de administratieve lastendruk.

Verder kan de applicatie ook worden ingezet bij het uitvoeren van kleinschalige experimenten met de VROM vergunning.

Ook kan worden overwogen om de applicatie, onder voorwaarden, aan te bieden aan partijen uit het bedrijfsleven, die de commerciële mogelijkheden ervan omarmen. Voornoemde scenario's hoeven elkaar niet uit te sluiten.

3. Exploitatie

In het kader van de exploitatie is het van belang stil te staan bij onderwerpen als:

- De markt: ontwikkelingen worden gevolgd om maximaal aan te blijven sluiten op gestelde wettelijke kaders en de eisen en wensen van de ketenspelers;
- Communicatie: informeren en vertrouwen winnen van de doelgroepen;
- Plaats en functies van de applicatie;
- Toekomstperspectief;
- Financieel model;
- Prijsstelling en voordelen.

Vanuit het perspectief van bedrijfszekerheid, is het van belang om duidelijkheid en zekerheid te bieden in de vorm van een SLA over onderwerpen zoals:

- De vertrouwelijkheid;
- De functionaliteit;
- De service;
- De techniek;
- De mogelijkheden voor koppeling met aangrenzende applicaties en gegevensbestanden;
- De continuïteit en bedrijfszekerheid / hosting;
- Prijsstelling.

4. Beheer

De volgende beheergebieden onderscheiden:

- Functioneel: procesmodellen, schermopbouw, berichten, inhoud informatie;
- Technisch: standaards, open source software;
- Juridisch: wettelijke regels en reglementen, SLA;
- Services: services aan gemeenten, gebruikersgroepen en ICT aanbieders.

Het is van belang, dat de doelgroepen vanuit hun rollen, zoals deze zijn vernoemd in paragraaf 7.2, continu en tijdig worden betrokken en geïnformeerd over ontwikkelingen op het vlak van het beheer. De organisatie en uitvoering van het beheer kan worden ondergebracht bij de stichting SAV€ als ketenplatform.

5. Prijsbeleid en financiering

Zoals gesteld neemt de applicatie een plaats in op het “openbare” speelveld tussen overheid en bedrijfsleven. In het verlengde hiervan ligt het voor de hand om de prijsstelling voor gebruikers te baseren op een transparant kostprijs principe.

De volgende scenario's kunnen worden overwogen:

1. Betaling per aanvraag: de gebruiker (aanvrager en of / verlener) betaalt per aanvraag (al dan niet op basis van een abonnement / lidmaatschap);
2. Financiering op basis van afdracht door gemeenten op basis van percentage van de totale omzet aan leges;
3. Financiering op basis van “sponsoring/bijdragen” door direct en indirect belanghebbende partijen;
4. Financiering op basis van commerciële exploitatie.

Ter bevordering van het gebruik is het van belang om de kosten voor de gebruiker in de aanloopfase (de eerste 2 tot 3 jaar) tot “een minimum” te beperken.

De kosten zouden voor deze periode dan moeten worden gefinancierd op basis van bijdragen van overheid en / of bedrijfsleven. Hiervoor wordt in het bijzonder gedacht aan het Ministerie van VROM en de brancheorganisaties.

7. Toekomstperspectief

Naar aanleiding van de ervaringen die zijn opgedaan met dit project en eerdere projecten op het gebied van digitaal toetsen zijn in dit hoofdstuk een klein aantal observaties en verwachtingen samengevat.

7.1 De toekomst van digitaal toetsen

De mogelijkheden om bouwkundige ontwerpen digitaal te toetsen zijn reeds jaren onderwerp van onderzoek in nationale en internationale context. Onder de term Building Code Checking kunnen in de internationale wetenschappelijke literatuur tal van voorbeelden worden gevonden van onderzoek om, ten behoeve van bijvoorbeeld brandveiligheid of constructieve veiligheid, digitale modellen van gebouwontwerpen te toetsen aan de voorschriften. Enkele voorbeelden hiervan zijn Woodbury et al. (2000), Yang and Xiang (2001) en Vanier (1995).

In Nederland is een goed voorbeeld van digitaal toetsen ontwikkeld door De Twee Snoeken (<http://www.tweesnoeken.nl/automatisering>) onder de naam Normworm. In deze applicatie kan een gebouwontwerp worden getoetst aan regelgeving ten aanzien van: de energie prestatie coëfficiënt; het minimaal vereiste raamoppervlak; de minimaal vereiste ventilatie; normen voor geluidsisolatie; en diverse eisen uit het bouwbesluit voor woningbouw en utiliteitsbouw.

Uit het voortschrijdend inzicht dat blijkt uit deze nationale en internationale ontwikkelingen mogen we de volgende drie conclusies trekken:

1. Het formaliseren van toetsen is in principe geen probleem

Hoewel er een afhankelijkheid is van de eenduidigheid van begripsbepalingen blijkt dat regelgeving over het algemeen zodanig kan worden geformuleerd dat een formele toets aan de regels mogelijk is. De kwaliteit van de toets wordt daarbij bepaald door de mate van exactheid van de bepalingen in de regels. Naarmate de begrippen meer exact worden gedefinieerd kan een meer gedetailleerde en exacte toets worden uitgevoerd. Een toets als 'hoogte dakrand ≤ 0.25 m' kan digitaal exact worden getoetst wanneer het begrip 'hoogte dakrand' exact is gedefinieerd. Wanneer het begrip niet of niet eenduidig is gedefinieerd kan ook digitaal worden getoetst, maar dan slechts in een eenvoudige vorm waarin door de aanvrager wordt gesteld of aan de toets wordt voldaan (ja/nee). Een ander voorbeeld van deze laatste situatie is: 'geen zichtbare bevestigingsmiddelen en naden'.

Bepaalde regels lenen zich niet goed voor digitale toetsing. Bijvoorbeeld een regel als 'materiaal, kleurgebruik en detaillering zijn afgestemd op bestaande architectuur' zal niet digitaal kunnen worden getoetst. Hiervoor zijn de begrippen onvoldoende concreet gedefinieerd. Op het gebied van welstandscriteria betreft het hier veelal toetsen van esthetische aard waarbij een relatie wordt gelegd met bestaande bouw. De belangrijkste moeilijkheid is echter dat de beoordeling of aan de toets wordt voldaan op subjectieve wijze (lees: door menselijke interpretatie) plaats moet vinden door de beoordelende instantie.

Volledigheid van toetsen is niet noodzakelijk een issue. Onvolledige toetsen kunnen acceptabel zijn zolang er duidelijkheid bestaat over de mate van onvolledigheid en de gevolgen hiervan voor de procedure van goedkeuring.

2. Het uitvoeren van digitale toetsen is geen probleem

Zodra regels zijn geformaliseerd kan er digitaal worden getoetst. De snelheid van het uitvoeren van digitale toetsen neemt toe met de rekenkracht van computers en hoeft in de huidige praktijk geen belemmering te vormen om gaande het ontwerpproces op zeer regelmatige basis het ontwerp te toetsen aan een veelheid van normen.

3. Het aanleveren van gegevens als input voor toetsen is wel een probleem

Cruciale succesfactor voor het digitaal toetsen blijkt in dit project, zoals ook in de literatuur en de huidige praktijk, te liggen bij de kwaliteit van de gegevens die ter toetsing worden aangeleverd. In het digitale dakkapel project is deze input volledig gestandaardiseerd: in parametrische modellen van woning en dakkapel is vooraf bepaald met welke gegevens het ontwerp wordt beschreven. Dit maakt het mogelijk om deze gegevens eenduidig en compleet aan de ‘ontwerper’ te vragen en een directe relatie te leggen met de digitale toetsen.

Zonder een dergelijke standaardisatie is het verkrijgen van de juiste informatie voor het uitvoeren van digitale toetsen veel moeilijker. In de literatuur blijkt de grootste inspanning dan ook te liggen bij het formaliseren van de input voor de toetsen, namelijk het standaardiseren van gebouwmodellen. De kwaliteit van een gebouwmodel wordt bepaald door de mate van eenduidigheid van de gegevens en de compleetheit van het model. Het is daarbij onontkoombaar dat strikte definities worden gehanteerd voor de manier waarop bouwkundige concepten worden vastgelegd. Op tegenstrijdige voet hiermee staat de eis die door ontwerpers gesteld wordt aan modelleergereedschappen: voldoende flexibiliteit en ontwerpvrijheid. De uitdaging voor ontwikkelaars van bouwkundige ontwerpsoftware ligt in het vinden van de gulden middenweg.

7.2 De rol van overheid en commercie

In het ‘spel’ van ontwerpen en toetsen kan in de huidige praktijk de volgende rolverdeling worden waargenomen:

- Overheid: normstellende rol;
- Bouwkundige experts: ontwerpende en toetsende rol;
- Software industrie: faciliterende rol;
- Burger: consument: heeft wensen en maakt keuzes.

Wanneer we deze rolverdeling doortrekken naar een situatie waarin regelgeving digitale vormen aanneemt zullen deze spelers in het veld de volgende activiteiten (moeten) ontplooiën.

- Overheid
 - Als wetgever heeft de overheid de verantwoordelijkheid over de kwaliteit van de normen en zal deze verantwoordelijkheid bij zich moeten houden, ook in geval van digitale normen. De bron van de digitale norm zal dus bij de overheid liggen en het is dan ook haar taak om regels in digitale vorm ter beschikking te stellen aan de partijen die geacht worden volgens de regels te ontwerpen en aan de regels te toetsen. Gezien de ontwikkelingen op het gebied van Internettechnologie, bijvoorbeeld het Semantic Web en Webservices, kan voor de vorm waarin digitale regels beschikbaar worden gesteld gedacht worden aan geautomatiseerde online diensten geleverd door de overheid. Daarnaast is er een taak bij de overheid om software die op correcte wijze van

deze diensten gebruik maakt te certificeren en duidelijkheid te verschaffen over de acceptatie van het gebruik van de software.

- **Bouwkundige experts**
De bouwkundige heeft de verantwoordelijkheid om een ontwerpdocument op een eenduidige en complete wijze inhoud te geven. De bouwpraktijk heeft dan ook een gezamenlijke verantwoordelijkheid om zich als actieve markt op te stellen voor het afzetten van software die dit mogelijk maakt.
- **Software industrie**
De software industrie zal altijd inspelen op de vraag vanuit de markt, maar daarbij grenzen stellen wat betreft aansprakelijkheid voor de prestaties van de geleverde software. Zij zal deze aansprakelijkheid laten hangen van de voorwaarden die gesteld worden aan de certificering van de software van overheidswege. De correctheid van de uitvoering van de digitale toets is een verantwoordelijkheid van de software leverancier, de correctheid van regels waaraan wordt getoetst echter niet. Het is daarom van belang dat de software industrie voldoende vraag neerlegt bij de overheid om de regels in de vorm van (online) diensten aan te bieden, opdat men kan beschikken over actuele regels en correctheid daarvan wordt gegarandeerd.
- **Burger:**
Burgers hebben twee drijfveren om in het digitale spel mee te spelen: lagere kosten en meer keuze, inspraak en zeggenschap.

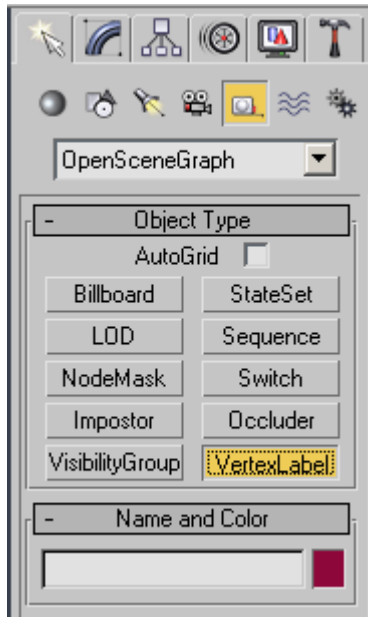
In een scenario samengevat kan men zich deze rolverdeling als volgt voorstellen. Een bouwkundige ontwerper gebruikt CAD software waarin een ontwerp wordt vastgelegd in objecten waarvan de bouwkundige betekenis eenduidig is gedefinieerd. De software bevat tevens een module die het mogelijk maakt om het gegevensmodel van het ontwerp digitaal te toetsen. Daartoe maakt de software gebruik van online diensten van de overheid die de regels digitaal aanleveren waarmee het model kan worden getoetst. De correctheid van de regels blijft daarmee de verantwoordelijkheid van de overheid, de correctheid van de toets die van de softwareleverancier. Het is de verantwoordelijkheid van de bouwkundige om gegarandeerde dan wel gecertificeerde software te gebruiken voor het toetsen.

Burgers zullen zich in toenemende mate willen opstellen als deskundige. Door de toenemende beschikbaarheid van informatiebronnen en beslissingsondersteunende gereedschappen zullen burgers steeds meer een goed geïnformeerde bijdrage willen en kunnen leveren aan ontwerp en planning van de gebouwde omgeving. Onder de voorwaarde dat de grenzen van de speelruimte goed zijn onderzocht en vastgelegd is het een wenselijke situatie dat burgers zonder tussenkomst van experts besluiten nemen over de gebouwde omgeving. Vooral als het onderwerpen betreft waarbij door hun beperkte belang en omvang de betrokkenheid van experts beperkt moet worden.

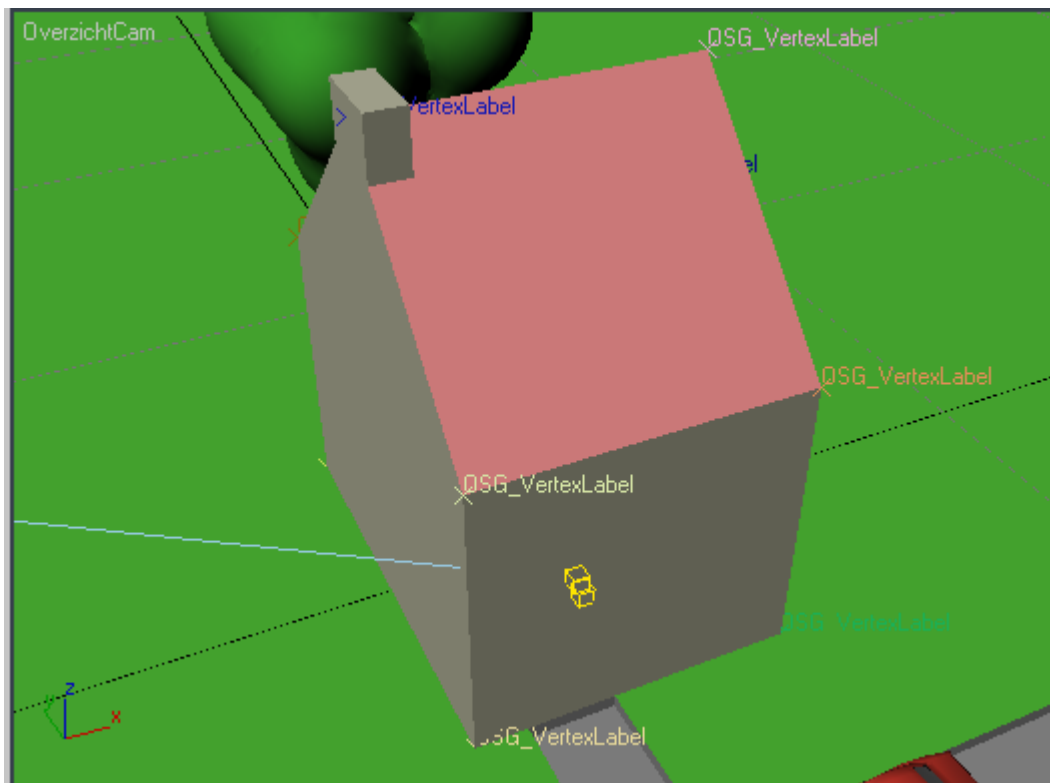
8. Bijlage

In deze bijlage zijn een aantal onderdelen van de ontwikkeling van de digitale dakkapel nader toegelicht ter verkrijging van een begrip van de totstandkoming en werking van de applicaties in fase 1 en ter indicatie van te verwachten werkzaamheden in vervolgfases.

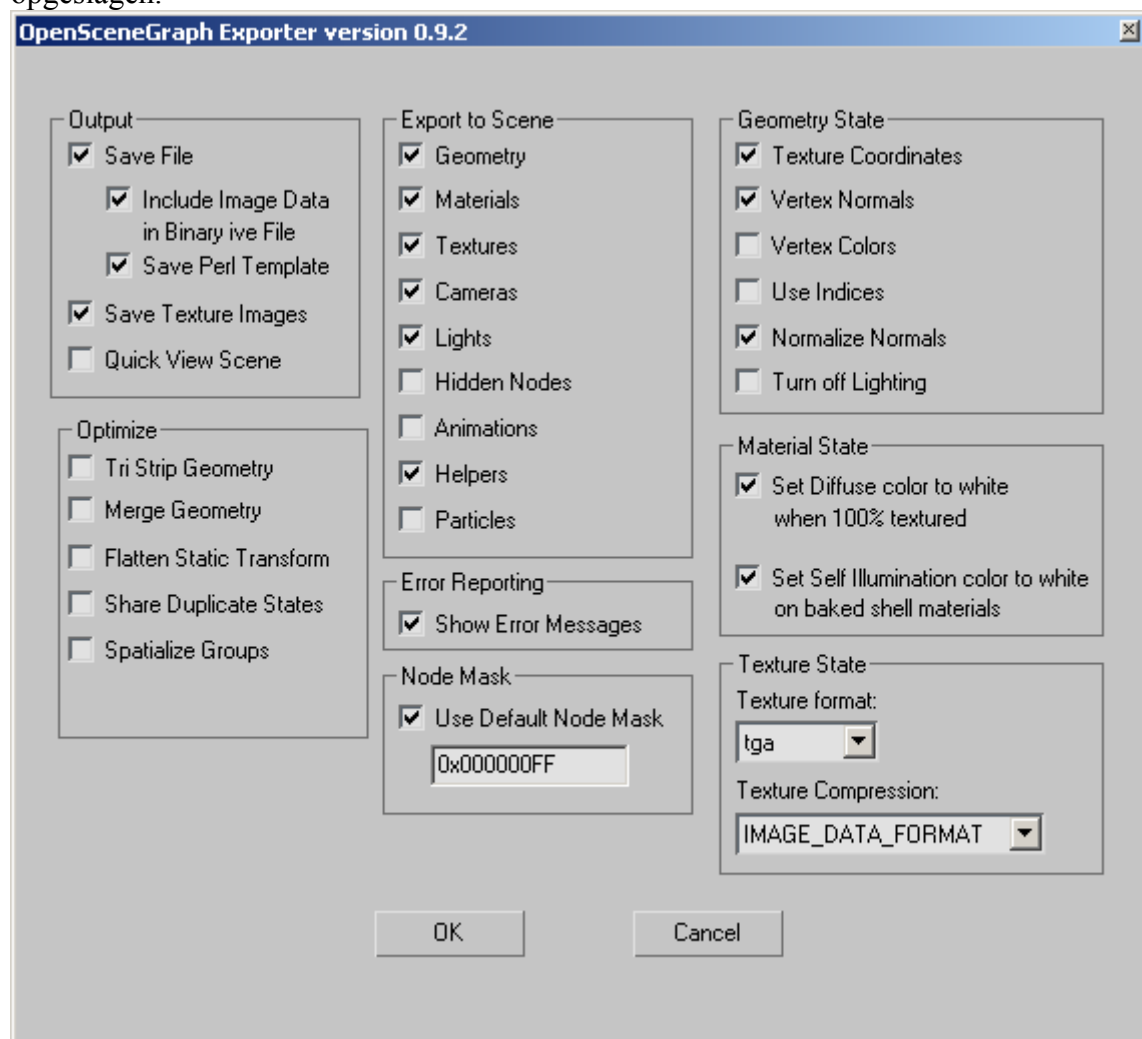
8.1 Parametriseren van geometrie



De dakkapel renderer is gebaseerd op geparametriseerde geometrie. De gangbare manier om geparametriseerde geometrie te maken is door de geometrische vorm in een scripttaal uit te drukken. Voor dit project is een tussenvorm bedacht die het parametriseren eenvoudiger maakt. Dit werkt op de volgende manier: In 3D Studio Max kan een model gemaakt worden. Dit model moet niet te ingewikkeld zijn en je moet je aan een aantal regels houden, maar verder kan het normaal gemodelleerd worden. Vervolgens kan er met een speciaal VertexLabel helper object (een plugin die beschikbaar is via de OpenSceneGraph helper dropdown) met behulp van 3D Vertex Snap, een aantal vertices gemarkeerd worden met een naam. Een model ziet er in Max dan uit als bijvoorbeeld in de afbeelding hieronder.



Vervolgens kan het model geëxporteerd worden. Dit wordt gedaan met de exporter die gebaseerd is op OSGExp (<http://osgexp.vr-c.dk/>). Aan dit component is in het kader van dit project functionaliteit toegevoegd zodat er ook een Perl script en een configuratie file bij de export worden gegenereerd. De scene moet geëxporteerd worden als een .osg file. Dit is een text file die een scene beschrijft voor open scene graph. Verder moet de optie “Save Perl Template” aangevinkt zijn om de extra configuratie files te genereren. De camera’s moeten ook geëxporteerd worden omdat die gebruikt worden voor de verschillende aanzichten om het model te renderen. De positie van de camera wordt geëxporteerd naar de .osg file, de camera lens informatie wordt in een .osgcfg file bewaard. De Perl template wordt in een .osgpl file opgeslagen.



In de Perl template zullen alle vertices die met behulp van de VertexLabel helper gemarkeerd zijn vervangen worden door een variabele. Een voorbeeld van de geëxporteerde Perl file is hieronder opgenomen.

```
$PuntA{"Muren"}=[-3,-4,0];
$PuntB{"Muren"}=[3,-4,0];
$PuntD{"Muren"}=[-3,4,0];
$PuntE{"Muren"}=[-3,-4,5.5];
$PuntE{"Dak"}=[-3,-4,5.5];
```

```

$PuntF{"Muren"}=[3,-4,5.5];
$PuntF{"Dak"}=[3,-4,5.5];
$PuntH{"Muren"}=[-3,4,5.5];
$PuntH{"Dak"}=[-3,4,5.5];
$PuntI{"Muren"}=[-3,0,9.5];
$PuntI{"Dak"}=[-3,0,9.5];
$PuntJ{"Muren"}=[3,-1.19209e-007,9.5];
$PuntJ{"Dak"}=[3,-1.19209e-007,9.5];
$PuntC{"Muren"}=[3,4,0];
$PuntG{"Muren"}=[3,4,5.5];
$PuntG{"Dak"}=[3,4,5.5];
print <<MARKER
MatrixTransform {
  DataVariance STATIC
  name "Scene Root"
  nodeMask 0xff
  cullingActive TRUE
  StateSet {
    UniqueID StateSet_0
    DataVariance STATIC
    rendering_hint DEFAULT_BIN
    ----- knip -----
    useDisplayList TRUE
    useVertexBufferObjects FALSE
    PrimitiveSets 1
    {
      DrawArrays TRIANGLES 0 36
    }
    VertexArray Vec3Array 36
    {
      $PuntA{"Muren"}[0] $PuntA{"Muren"}[1] $PuntA{"Muren"}[2]
      $PuntD{"Muren"}[0] $PuntD{"Muren"}[1] $PuntD{"Muren"}[2]
      $PuntC{"Muren"}[0] $PuntC{"Muren"}[1] $PuntC{"Muren"}[2]
      $PuntC{"Muren"}[0] $PuntC{"Muren"}[1] $PuntC{"Muren"}[2]
      $PuntB{"Muren"}[0] $PuntB{"Muren"}[1] $PuntB{"Muren"}[2]
      $PuntA{"Muren"}[0] $PuntA{"Muren"}[1] $PuntA{"Muren"}[2]
      $PuntA{"Muren"}[0] $PuntA{"Muren"}[1] $PuntA{"Muren"}[2]
      $PuntB{"Muren"}[0] $PuntB{"Muren"}[1] $PuntB{"Muren"}[2]
      $PuntF{"Muren"}[0] $PuntF{"Muren"}[1] $PuntF{"Muren"}[2]
      $PuntF{"Muren"}[0] $PuntF{"Muren"}[1] $PuntF{"Muren"}[2]
      $PuntE{"Muren"}[0] $PuntE{"Muren"}[1] $PuntE{"Muren"}[2]
      $PuntA{"Muren"}[0] $PuntA{"Muren"}[1] $PuntA{"Muren"}[2]
      $PuntB{"Muren"}[0] $PuntB{"Muren"}[1] $PuntB{"Muren"}[2]
      $PuntC{"Muren"}[0] $PuntC{"Muren"}[1] $PuntC{"Muren"}[2]
      $PuntG{"Muren"}[0] $PuntG{"Muren"}[1] $PuntG{"Muren"}[2]
      $PuntB{"Muren"}[0] $PuntB{"Muren"}[1] $PuntB{"Muren"}[2]
      $PuntG{"Muren"}[0] $PuntG{"Muren"}[1] $PuntG{"Muren"}[2]
      $PuntJ{"Muren"}[0] $PuntJ{"Muren"}[1] $PuntJ{"Muren"}[2]
      $PuntB{"Muren"}[0] $PuntB{"Muren"}[1] $PuntB{"Muren"}[2]
      $PuntJ{"Muren"}[0] $PuntJ{"Muren"}[1] $PuntJ{"Muren"}[2]
      $PuntF{"Muren"}[0] $PuntF{"Muren"}[1] $PuntF{"Muren"}[2]
      $PuntC{"Muren"}[0] $PuntC{"Muren"}[1] $PuntC{"Muren"}[2]
      $PuntD{"Muren"}[0] $PuntD{"Muren"}[1] $PuntD{"Muren"}[2]
      $PuntH{"Muren"}[0] $PuntH{"Muren"}[1] $PuntH{"Muren"}[2]
      $PuntH{"Muren"}[0] $PuntH{"Muren"}[1] $PuntH{"Muren"}[2]
      $PuntG{"Muren"}[0] $PuntG{"Muren"}[1] $PuntG{"Muren"}[2]
      $PuntC{"Muren"}[0] $PuntC{"Muren"}[1] $PuntC{"Muren"}[2]
      $PuntD{"Muren"}[0] $PuntD{"Muren"}[1] $PuntD{"Muren"}[2]
      $PuntA{"Muren"}[0] $PuntA{"Muren"}[1] $PuntA{"Muren"}[2]
      $PuntE{"Muren"}[0] $PuntE{"Muren"}[1] $PuntE{"Muren"}[2]
      $PuntD{"Muren"}[0] $PuntD{"Muren"}[1] $PuntD{"Muren"}[2]
      $PuntE{"Muren"}[0] $PuntE{"Muren"}[1] $PuntE{"Muren"}[2]
      $PuntI{"Muren"}[0] $PuntI{"Muren"}[1] $PuntI{"Muren"}[2]
      $PuntD{"Muren"}[0] $PuntD{"Muren"}[1] $PuntD{"Muren"}[2]
      $PuntI{"Muren"}[0] $PuntI{"Muren"}[1] $PuntI{"Muren"}[2]
      $PuntH{"Muren"}[0] $PuntH{"Muren"}[1] $PuntH{"Muren"}[2]
    }
    NormalBinding PER_VERTEX
    NormalArray Vec3Array 36
    {
      0 0 -1
      0 0 -1
      0 0 -1
    }
    ----- knip -----

```

```

    }
    num_children 1
    Geode {
        DataVariance STATIC
        nodeMask 0xff
        cullingActive TRUE
        num_drawables 0
    }
}
}
MARKER

```

Vervolgens kan met behulp van deze file de geometrie geparametriseerd worden. Hierin heeft men bij het schrijven van het script alle vrijheid. Een fragment voor het script van de woning is hieronder gegeven:

```

#Parametriseren
$PuntA{"Muren"}=[-3,-4,0];
&setorigin($PuntA{"Muren"}); #Werkt ook voor dak
$PuntB{"Muren"}=&offsetpoint($breedte,0,0);
$PuntD{"Muren"}=&offsetpoint(0,$diepte,0);
$PuntE{"Muren"}=&offsetpoint(0,0,$goothoogte);
$PuntE{"Dak"}=$PuntE{"Muren"};
$PuntF{"Muren"}=&offsetpoint($breedte,0,$goothoogte);
$PuntF{"Dak"}=$PuntF{"Muren"};
$PuntH{"Muren"}=&offsetpoint(0,$diepte,$goothoogte);
$PuntH{"Dak"}=$PuntH{"Muren"};
$PuntI{"Muren"}=&offsetpoint(0,$diepte/2.0,$nokhoogte);
$PuntI{"Dak"}=$PuntI{"Muren"};
$PuntJ{"Muren"}=&offsetpoint($breedte,$diepte/2.0,$nokhoogte);
$PuntJ{"Dak"}=$PuntJ{"Muren"};
$PuntC{"Muren"}=&offsetpoint($breedte,$diepte,0);
$PuntG{"Muren"}=&offsetpoint($breedte,$diepte,$goothoogte);
$PuntG{"Dak"}=$PuntG{"Muren"};

#           -2.65305 -0.0281586 7.99112 1
@SchoorsteenFix=($PuntJ{"Dak"}[0]-3-2.65305,$PuntJ{"Dak"}[1]-1.19209e-007-
0.0281586,$PuntJ{"Dak"}[2]-9.5+7.99112);

```

8.2 Onderlinge afhankelijkheden van de parameters

De parameters van de dakkapel zijn onderling afhankelijk. Wanneer de breedte wordt aangepast moeten de linker- en/of rechterafstand ook een andere waarde krijgen. Deze onderlinge afhankelijkheid is gedefinieerd in het script waarin de geparametriseerde dakkapel is gedefinieerd. Een voorbeeld van de manier waarop deze afhankelijkheden zijn vastgelegd volgt hierna.

```

#calculate the parameters
@changed=split(/[ ,]+/, $changed);
if(changed("right") && changed("dakkapelwidth") && changed("left")) {
    ignorechange("dakkapelwidth");
}
if(!changed("right")) { $right=$dakvlakbreedte-$dakkapelwidth-$left; }
if(!changed("dakkapelwidth")) { $dakkapelwidth=$dakvlakbreedte-$left-$right; }
if(!changed("left")) { $left=$dakvlakbreedte-$dakkapelwidth-$right; }
if(changed("top") && changed("dakkapelheight") && changed("bottom")) {
    ignorechange("dakkapelheight"); }
if(!changed("top")) { $top=$dakvlaklengte-$dakkapellengte-$bottom; }
if(!changed("dakkapelheight")) {
    $dakkapellengte=$dakvlaklengte-$top-$bottom;
    $dakkapelheight=$dakkapellengte*sin($angle);
}
if(!changed("bottom")) { $bottom=$dakvlaklengte-$dakkapellengte-$top; }

```

8.3 Digitale toetsingscriteria

De criteria voor de VROM toets en de Welstandstoets zijn in XML formaat geformaliseerd. De criteria zoals in fase 1 geïmplementeerd zien er in dat formaat als volgt uit.

```
<checks>
  <check id="100" expression="$monumentaal==0" type="vrom" stage="gg">
    <fail>Het huis is een monumentaal pand.</fail>
  </check>
  <check id="101" expression="$onvoordakvlak==0" type="vrom" stage="hg">
    <fail>De dakkapel komt aan de voorzijde van het huis.</fail>
  </check>
  <check id="102" expression="$stoopenspace==0" type="vrom" stage="dp">
    <fail>De dakkapel is naar een weg of openbaar groen gekeerd.</fail>
  </check>
  <check id="103" expression="$left>=0.5" type="vrom" stage="dg">
    <fail>De afstand tussen de dakkapel en de linkerkant van het huis is kleiner dan
50 centimeter.</fail>
  </check>
  <check id="104" expression="$top>=0.5" type="vrom" stage="dg">
    <fail>De afstand tussen de dakkapel en de nok van het huis is kleiner dan 50
centimeter, gemeten in het dakvlak.</fail>
  </check>
  <check id="105" expression="$right>=0.5" type="vrom" stage="dg">
    <fail>De afstand tussen de dakkapel en de rechterkant van het huis is kleiner dan
50 centimeter.</fail>
  </check>
  <check id="106" expression="$bottom>=0.5" type="vrom" stage="dg">
    <fail>De afstand tussen de dakkapel en de dakgoot van het huis is kleiner dan 50
centimeter, gemeten in het dakvlak.</fail>
  </check>
  <check id="107" expression="$bottom<=1" type="vrom" stage="dg">
    <fail>De afstand tussen de dakkapel en de dakgoot van het huis is groter dan 1
meter.</fail>
  </check>
  <check id="108" expression="$dakkapelheight<1.500001" type="vrom" stage="dg">
    <fail>De hoogte van de dakkapel is groter dan 1.5 meter.</fail>
  </check>
  <check id="109" expression="$did eq 'd1'" type="vrom" stage="dp">
    <fail>De dakkapel heeft geen plat dak.</fail>
  </check>
  <check id="110" expression="$dakkapelsidetrans==0" type="vrom" stage="dg">
    <fail>De wanden aan de zijkant van de dakkapel zijn doorzichtig.</fail>
  </check>
  <check id="200" expression="$top>=0.5" type="loket" stage="dg">
    <fail>De afstand van de dakkapel tot de nok moet minimaal 50 centimeter zijn,
gemeten in het dakvlak.</fail>
  </check>
  <check id="201" expression="$bottom>=0.5" type="loket" stage="dg">
    <fail>De afstand van de dakkapel tot de dakgoot moet minimaal 50 centimeter zijn,
gemeten in het dakvlak.</fail>
  </check>
  <check id="202" expression="$bottom<=1" type="loket" stage="dg">
    <fail>De afstand van de dakkapel tot de dakgoot mag niet groter zijn dan 1 meter,
gemeten in het dakvlak.</fail>
  </check>
  <group>
    <check id="203" expression="$leftopen==0 || $left>=2" type="loket" stage="dg">
      <fail>Omdat er aan de linkerkant een weg of openbaar groen is moet de dakkapel
minimaal 2 meter van de linkerkant van het huis geplaatst worden.</fail>
    </check>
    <check id="204" expression="$left>=0.5" type="loket" stage="dg">
      <fail>De afstand van de dakkapel tot de linkerkant moet minimaal 50 centimeter
zijn.</fail>
    </check>
  </group>
  <group>
    <check id="205" expression="$rightopen==0 || $right>=2" type="loket" stage="dg">
      <fail>Omdat er aan de rechterkant een weg of openbaar groen is moet de dakkapel
minimaal 2 meter van de rechterkant van het huis geplaatst worden.</fail>
    </check>
  </group>

```

```

<check id="506" expression="$right>=0.5" type="loket" stage="dg">
  <fail>De afstand van de dakkapel tot de rechterkant moet minimaal 50 centimeter
zijn.</fail>
</check>
</group>
<check id="207" expression="$dakkapelrandvorm==1" type="loket" stage="dp">
  <fail>De hoogte van de dakrand mag niet groter zijn 25 centimeter zijn en een
overstek van 5 tot 10 centimeter is verplicht.</fail>
</check>
<check id="208" expression="$dakrandmat<3" type="loket" stage="dp">
  <fail>De dakrand moet gemaakt zijn van houten plaatmateriaal, houten delen, of
zink.</fail>
</check>
</checks>

```

8.4 Referenties

Woodbury, R., Burrow, A., Drogenuller, R. and Datta, S. (2000) Code checking by representation comparison , In: *Proceedings of CAADRIA2000*, Singapore, pp. 235-244.

Yang, QZ and Li, Xiang (2001) Representation and Execution of Building Codes for Automated Code Checking, In: De Vries, Van Leeuwen, Achten (eds.) *Proceedings of the Ninth International Conference on Computer Aided Architectural Design Futures* [ISBN 0-7923-7023-6] Eindhoven, 8-11 July 2001, pp. 315-329.

Vanier, D.J. (1995) Canada and computer representations of design standards and building codes, *The Int. Journal of Construction IT* **3**(1), pp.1-12.