

SALIX-2

SALIX-2

Simulatie Agenten voor Landschapsarchitectonisch Design in virtual reality (X)

**R. van Lammeren
V. Clerc (SERC)
H. Kramer**

**Alterra-rapport 715
CGI-rapport 2003-1**

Alterra, Research Instituut voor de Groene Ruimte, Wageningen, 2003

REFERAAT

Lammeren, R. van, V. Clerc (SERC), H. Kramer , 2003. *SALIX-2. Simulatie Agenten voor Landschapsarchitectonisch Design in virtual reality (X)*. Wageningen, Alterra, Research Instituut voor de Groene Ruimte. Alterra-rapport 715. 67 blz. 18 fig.; 8 tab.; 28 ref.

De ontwikkeling van een driedimensionale computerapplicatie ten behoeve van de landschapsarchitectuur wordt beschreven. De applicatie, een voorbeeld van *virtual reality*, wordt zowel functioneel als naar implementatie besproken. Ingegaan wordt op de digitale weergave van beplantingen, het beplantingsplan als de specifieke geo-informatie van beplantingen, de groeisimulatie van beplantingen, het onderscheid tussen beplantingsobjecten en landschapsarchitectonische objecten alsmede de transformatie van die beplantingsobjecten in architectonische objecten en natuurlijk de wijze waarop een gebruiker van deze applicatie in deze virtuele wereld kan navigeren. De technische bouw van deze applicatie door middel van Java, Javascript, Vrmml en Access wordt besproken alsmede de wijze waarop een gebruiker de applicatie kan bedienen.

Trefwoorden: beplantingsplan, virtual reality, groeisimulatie, landschapsarchitectuur, geo-informatiekunde, drie-dimensionale ontwerpomgeving, vrmml

ISSN 1566-7197

Dit rapport kunt u bestellen door €22,- over te maken op banknummer 36 70 54 612 ten name van Alterra, Wageningen, onder vermelding van Alterra-rapport 715. Dit bedrag is inclusief BTW en verzendkosten.

© 2003 Alterra, Research Instituut voor de Groene Ruimte,
Postbus 47, NL-6700 AA Wageningen.
Tel.: (0317) 474700; fax: (0317) 419000; e-mail: info@alterra.nl

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van Alterra.

Alterra aanvaardt geen aansprakelijkheid voor eventuele schade voortvloeiend uit het gebruik van de resultaten van dit onderzoek of de toepassing van de adviezen.

Inhoud

Inhoud	5
Woord vooraf	7
Samenvatting	9
1 Inleiding	11
2 Beplantingsmethoden en beheer	15
3 Beplanting en Landschapsarchitectuur	19
4 Digitale representatie van beplanting	21
5 Inzet ICT-componenten	23
6 SALIX-2 ontwerp	27
7 Functionaliteit SALIX-2	31
7.1 Opstarten en afsluiten	31
7.2 Ondersteunde objecttypen	31
7.3 De gebruikersinterface	32
7.4 Navigatie	33
7.5 Werken met beplantingsplannen	35
7.6 Werken met objecten	36
7.7 Groei van objecten	38
7.8 Visualisatie van overlappende objecten	38
7.9 Overige functies	41
8 Technische beschrijving	43
8.1 Drie-lagen architectuur	43
8.2 Gegevenslaag	43
8.3 Applicatielogica	46
8.4 Visualisatie	49
9 Conclusies en vervolg	53
Literatuur	57
Bijlagen	
1. Installatie	61
2. overzicht van 'multi-agents systems' concepten	65

Salix: geslacht van bomen en heesters (wilg), met zeer buigzame takken, die veel voor bindrijs, vlechtwerk en hoephout gebruikt worden (van Dale)

Woord vooraf

Zomer 2000 werd het projectvoorstel SALIX in het kader van KUS-MOOK geselecteerd als een subsidiabel project. Met een multidisciplinair, gemotiveerd team moest uitvoering worden gegeven aan een ambitieus plan: een virtuele ruimte applicatie van een groeisimulatie van beplanting op basis van een beplantingsplan, die bij voorkeur functioneert in een internet-omgeving.

De resultaten van dit Mook-project zijn gepubliceerd in een rapport van de leerstoelgroep Landschapsarchitectuur van Wageningen Universiteit. Vervolgens kon via middelen uit de onderwijs-innovatie van het ministerie van Landbouw, Natuur en Visserij (project STUWWAL) en het SEO-programma Virtuele Groene Ruimte (2001/2002) een vervolg worden gegeven aan een nadere uitwerking van het SALIX-prototype zoals dat in het rapport uit 2001 is beschreven. Onderhavig rapport beschrijft de resultaten van dit vervolg. Om de lezer van dit rapport de gehele context van het SALIX-project te verduidelijken zijn de hoofdstukken uit de eerdere publicatie opgenomen in dit rapport.

De in dit rapport behandelde Salix-2 applicatie is *freeware* en op te vragen bij de Geodesk van het Centrum Geo-Informatie (email: geodesk@wur.nl en url: www.geo-informatie.nl/cgi/geodesk) van Wageningen Universiteit en Research.

Het projectteam in deze tweede fase van dit SALIX-experiment was klein maar slagvaardig. Marc Evers (SERC) en Henk Kramer (WUR-Alterra) vormden een solide basis voor Viktor Clerc (Serc) en mij om geregeld op terug te vallen. Viktor, met name, heeft zich, als ware ICT'er, uitstekend ingeleefd in het landschapsarchitectonische denken en doen.

Een ieder dit in dit project geparticipeerd heeft wil ik bedanken. In de eerste fase hebben oa. Klaas Kerkstra (hoogleraar Landschapsarchitectuur), Bert Annevelink (WUR-IMAG) en Frits Ruyten (Integralis bv.), Michiel Uiterwijk (WiSL), Gerd Weitkamp (student WU-landschapsarchitectuur), Henk van Aggelen (WU-Landschapsarchitectuur) en Monica Wachowicz (WUR-CGI) bijgedragen aan een bijzondere pilot. In dit tweede opvolgende project moet met name Arend Ligtenberg bedankt worden voor de nodige nuchtere gesprekken en het projectbeheer door Mies Gerbrands (WU-landschapsarchitectuur).

Wageningen, februari 2003

Ron van Lammeren
Projectleider SALIX-2,
WUR - Centrum Geo-Informatie / Landschapsarchitectuur

Samenvatting

SALIX-1 is een computerapplicatie die in 2001 is ontwikkeld om diverse beplantings- en beheersmethoden te vergelijken. De applicatie was een voorbeeld van een zogenaamde *virtual reality* applicatie waaraan een groeisimulatie van beplantingsobjecten was gekoppeld.

Het doel van het onderhavig onderzoek is het ontwikkelen van een driedimensionale, digitale werkomgeving voor landschapsarchitecten. Deze werkomgeving wordt SALIX-2 genoemd en vormt een nadrukkelijk vervolg op SALIX-1. In SALIX-1 is vooral aandacht besteed aan de omzetting van een twee-dimensionale weergave van groeisimulatie (nl. SIMGRO-PP van IMAG) in een drie-dimensionale weergave. SALIX-2 richt zich vooral op de nadere koppeling met geo-data en de verfijning van de applicatie tbv. het ontwerpproces. Impliciet is via dit project gezocht naar meer generieke aspecten van virtual reality (VR) ten behoeve van de visualisatie van geo-informatie ten behoeve van communicatie. Naar inhoud vormt dit project dan ook een onderdeel van het Virtuele Groene Ruimte onderzoek.

Weergave beplantingsobjecten

Met betrekking tot de weergave van beplanting, een belangrijke bouwsteen van de landschapsarchitectuur, in een virtuele werkelijkheid is het nodig om duidelijk te maken op welke wijze beplanting in een dergelijke virtuele wereld wordt gerepresenteerd. Want ook voor de digitale representatie geldt uiteindelijk dat het object moet worden waargenomen.

Waarnemen is mogelijk wanneer er gelijkheid in verschil is te ontdekken. Digitale objecten en hun onderlinge verschillen kunnen worden waargenomen op grond van verschillende grafische kenmerken (oa. kleur, grootte, textuur en vorm). Door middel van geometrische en grafische abstracties kunnen representaties van de werkelijkheid gemaakt worden, die geschikt zijn voor het doel waarvoor ze afgebeeld worden. Wanneer we naar beplantingsobjecten kijken, valt op dat deze uit complexe vormen (natuurlijk, onregelmatig) bestaan, waardoor ze in essentie moeilijk zijn te representeren.

Wanneer een aantal traditionele afbeeldingen van landschapsarchitectonische ontwerpen bekeken wordt, is te zien dat op veel verschillende manieren in twee dimensies getracht wordt een ruimtelijk beeld weer te geven met gebruik van kenmerken van de beplantingsobjecten. De belangrijkste beperkingen daarvan zijn het gebruik van slechts één projectie en het afbeelden van een specifieke moment in de tijd.

Digitale representatie

Bepantingsobjecten staan echter niet op zichzelf maar kennen een onderlinge relatie alsmede (uiteindelijk) een locatie op aarde. In dit laatste ligt de link met geo-informatie besloten. Het betekent uiteindelijk dat er een 3D-scene wordt gebouwd die de beplantingsobjecten, afkomstig van een beplantingsplan, weergeven in een bepaalde ruimtelijke context.

SALIX-2 biedt de mogelijkheid om dergelijk 3D-scenes eenvoudig te maken via een ontwerpproces, al dan niet gebaseerd op een bestaand beplantingsplan in combinatie met geo-data. Dit houdt bovendien in dat deze digitale modellen op een relatief eenvoudige manier aangepast kunnen worden. De ruimtelijke effecten die een verandering in het landschap, op basis van de aanpassing van beplantingsobjecten, kan hierdoor goed bestudeerd worden.

Landschapsarchitectonische objecten

Behalve op het werken met en het simuleren van de groei van beplanting is in dit project ook gezocht naar een mogelijkheid om de koppeling te leggen naar ruimtevorming. Daartoe is aansluiting gezocht bij de landschapsarchitectonische objecten zoals die door Simonds zijn geïntroduceerd. De ruimte wordt gevormd deze architectonische objecten. De ruimte vormt als het ware de contramal hiervan. De omvorming van beplantingsobjecten in landschapsarchitectonische objecten vormt een deelonderzoek van deze applicatiebouw.

Uiteindelijk wordt in deze studie aangegeven welke functionaliteiten SALIX-2 heeft gekregen en op welke wijze deze zijn geïmplementeerd. Daarbij gaat het om:

- database beheer;
- navigatie in het virtual reality –model (VR-model);
- interactie met het VR-model;
- definitie van solitaire en niet-solitaire beplantingsobjecten;
- transformatie van beplantingsobjecten in architectonische objecten.

Met behulp van SALIX-2 lijkt meer informatie over ruimtevorming en beleving te verkrijgen dan via het platte vlak het geval is. Met behulp van navigatie door een gebied ontstaat een nog beter ruimtelijk inzicht

Toekomstige ontwikkelingen

Het uiteindelijk doel van dit project betreft het vervaardigen van een digitale driedimensionale werkomgeving, waarmee ontwerp van het landschap via de digitale aanplant van bomen en struiken wordt ondersteund. Deze werkomgeving kan via internet toegankelijk zijn. Op termijn kan men denken aan uitbreidingen via een uitgebreide set aan groeicurven van bomen en planten, om daarmee de digitale database met beplantingsobjecten uit te breiden.

Ook de omzetting van beplantingsobjecten naar architectonische objecten zal op termijn om een verbetering vragen. Deze verbetering kan te maken hebben met de visualisatie enerzijds en met de geometrie anderzijds.

Het toepassen van deze mogelijkheden in de onderwijs- en ontwerppraktijk is natuurlijk een 'must' om te achterhalen op welke wijze deze vorm van ontwerpen bijdraagt aan de kennis en ervaring van ruimtevorming.

De voordelen die de uiteindelijke digitale driedimensionale werkomgeving zullen bieden voor landschapsarchitecten en groenbeheerders, zijn veelbelovend en de verwachting is, dat de invloed hiervan in de architectonische denk- en werkwereeld, waarin bomen en struiken belangrijke 'bouwstenen' zijn, niet lang zal uitblijven.

1 Inleiding

Aanleiding

Drie-dimensionale modelrepresentatie van beplanting vormt een van de meest complexe digitale computertoepassingen. De complexiteit komt voort uit het nastreven van een realistische weergave van beplanting. Het streven naar een zo groot mogelijk realisme, via de detaillering van het 3-dimensionale-model (3D-model), alsmede die van de digitale foto die op dat 3D-model wordt geplaatst, zorgt tevens voor omvangrijke bestanden. Daarnaast zorgt groei-simulatie van de beplanting eveneens voor een toenemende complexiteit. Wereldwijd zijn er op dit gebied momenteel slechts een beperkt aantal aansprekende voorbeelden¹. Op zich is deze complexiteit en de omvang van bestanden niet direct bezwaarlijk. Deze wordt pas bezwaarlijk indien het doel is om in grote mate interactief met een dergelijk 3D-model te werken. Het project SALIX-2 richt zich nu met name op het interactief werken met 3-dimensionale modelrepresentaties van de werkelijkheid.

Om de resultaten van dit project te verduidelijken wordt in dit rapport op vele, uiteindelijk met elkaar samenhangende aspecten, ingegaan. Het gaat daarbij om o.a. landschapsarchitectonische concepten, principes van *virtual reality* en digitale visualisatie, alsmede de code waarmee deze concepten en principes zijn geoperationaliseerd. Ook worden voorbeelden gegeven van het mogelijk gebruik van de ontwikkelde software

De uitvoering van het onderzoek is verricht in het kader van de tweede fase van het project 'simulatie agenten voor het landschapsarchitectonisch en interactief virtueel groenbeheer (x)'. Het project wordt in het kort met het acroniem SALIX, de latijnse naam voor de wilg - een boom die bekend staat als een belangrijke pioniersoort-, aangeduid.

In dit project zijn een aantal vragen beantwoord die verbonden zijn aan de pogingen om een landschapontwerp-proces, waarin de invloed van de tijd zichtbaar, te simuleren.

Daartoe wordt voortgebouwd op de ervaringen en vindingen uit het SALIX-1 project

In het SALIX-1 project [LAM1] stonden vragen die o.a. betrekking hadden op kenmerken van het groenbeheer centraal. Daarbij moet duidelijk zijn dat groenbeheer ten doel heeft de beplanting zodanig te begeleiden dat op termijn een bepaald landschapsarchitectonisch beeld wordt gerealiseerd. Groenbeheer vormt daarmee enerzijds een schakel tussen ontwerp en realiteit en tussen aanplant en beoogde sfeer anderzijds.

Om die schakel minder afhankelijk te maken van de werkelijke groei- en ontwikkelingstijd lijken digitale middelen een uitstekend mogelijkheid te bieden om

¹ <http://www.cirad.fr/presentation/en/program-eng/amap.shtml> een simulatie programma van CIRAD geeft een voorbeeld van een zeer complex en niet interactief te gebruiken groeisolimulatie. Bovendien is deze simulatie gebaseerd op botanische en genetische kenmerken van gewassen.

virtueel die groei en ontwikkeling te simuleren. Een simulatie die daardoor het ontwerp met de realiteit verbindt en inzicht geeft in de mogelijke ontwikkeling van de aanplant op basis van een beplantingsplan in de richting van het beoogde landschapsarchitectonisch idee.

Deze intentie roept natuurlijk een veelheid aan nieuwe vragen op. Vragen die te maken hebben met de beplanting zelf: hoe ontwikkelt de beplanting zich in de tijd; welk beeld kent deze per seizoen en op welke wijze ontstaat vanuit de beplanting een architectonisch element. Hiertoe verwijzen wij naar het werk van F. Ruyten [RUY1, RUY3].

Vragen die te maken hebben met het gebruik van beplantingsobjecten in de landschapsarchitectuur. Om deze vragen te beantwoorden wordt vooralsnog gebruik gemaakt van de concepten van F. Simonds [SIM1].

Daarnaast zijn er vragen die te maken hebben met de digitale wereld: hoe simuleer je de groei van beplanting, op welke wijze visualiseer je de verschijningsvorm en op welke manier kunnen architectonische ingrepen in een digitale omgeving uitgevoerd worden. Vragen die impliciet weer vele nieuwe vragen oproepen.

In het Salix-2 project is geprobeerd om het oorspronkelijke prototype zodanig uit te bouwen dat studenten landschapsarchitectuur op relatief eenvoudige wijze ontwerp ingrepen aan de hand van beplantingselementen kunnen uitvoeren en vervolgens het ruimtelijk effect van die ingrepen kunnen bestuderen. De belangrijkste van die technologieën zijn gericht op navigatie in en interactief gebruik van een drie-dimensionale virtuele werkelijkheid.

Doelstelling

Doel van dit SALIX-2 project betreft het verbeteren van het SALIX-1 prototype in een applicatie die in het Landschapsarchitectonisch onderwijs kan worden ingezet en waarmee tevens de meerwaarde van de zogenaamde Virtual Reality applicaties valt aan te tonen.

Het SALIX-1 project richtte zich op het 'aanvullen' van een 2D-groeisimulatie met een 3D-groeisimulatie. Voor de visualisatie van de groei werd gebruik gemaakt van een digitale beschrijvingsvorm van boom- en struiktypen die gebaseerd is op een 'trade-off' tussen de detaillering van 3D-model en bitmap, waarbij de groeisimulatie aan de hand van eenvoudige geometrische objecten voorop stond. Vooralsnog wordt daarbij gedacht aan het ontwikkelen van 'agents', waarbij iedere boomsoort moet worden voorgesteld als een 'agent' die groeikarakteristieken (aanpassing geometrie) en seizoenskenmerken meekrijgt (verandering bitmap). Via de applicatie werden beheersproblemen ('overlap' tussen beplantingselementen) zichtbaar.

Het oorspronkelijke prototype was gebaseerd op het proefterrein Prins Bernhardbos grenzend aan het Floriade 2002 tentoonstellingsterrein. Via dit proefterrein werd een koppeling gemaakt met geo-informatie.

In dit onderhavig project vormde het oorspronkelijke prototype het vertrekpunt, maar is uiteindelijk een compleet nieuwe computerapplicatie ontwikkeld. Veranderingen, dan wel uitbreidingen, t.o.v. het oorspronkelijke prototype zijn:

- database beheer;
- verbeterde navigatie in het *virtual reality*-model (VR-model);
- interactie met het VR-model;
- definitie van solitaire en niet-solitaire beplantingsobjecten;
- transformatie van beplantingsobjecten in architectonische objecten.

In dit SALIX-2 project zijn voor het laatste aspect 'agents'-concepten gebruikt.

Toepassingsmogelijkheden

Primair is de SALIX-2 applicatie bedoeld om studenten landschapsarchitectuur inzicht te verschaffen in de ruimtelijke werking van hun ontwerp-ideeën. De applicatie is mbt. de groei van de beplantingselementen gebaseerd op het praktijkonderzoek naar groenbeheer in stedelijke gebieden (praktijkonderzoek kostenvergelijking 'wijker-blijver' en 'integrale beplantingsmethode' IMAG en Integralis pp). Veelal zijn beplantingsplannen in de stedelijke sfeer gebaseerd op het 'wijker-blijver' principe. Afhankelijk van groei-ontwikkeling en architectonische doelstelling zal tijdens het groeiproces keuzen moeten worden gemaakt mbt. de uitkap van boomsoorten en -individuen. Door gebruik te maken van de 3D groeisimulatie is het mogelijk om op basis van empirische meetgegevens een virtuele indruk te krijgen van het groeigedrag en de te verwachten effecten van de beheersingreep.

Ook bestaat er een mogelijkheid om de beplantingsontwikkeling volgens het 'wijker-blijver' principe te vergelijken met die van de 'integrale beplantingsmethode'. Hiervoor is er in het SALIX-1 project een relatie gelegd met het promotie-onderzoek van F. Ruyten en de onderzoeksresultaten van Integralis pp. en IMAG. Deze vergelijking biedt interessante mogelijkheden voor architecten, beheerders en de boomkwekerijsector via bijvoorbeeld Plant Propaganda Holland.

De landschapsarchitectuur krijgt hiermee mogelijkwijze een middel in handen waarmee de ontwikkeling van een beoogde architectonisch concept kan worden gesimuleerd in relatie tot aanleg- en onderhoudskosten. Daar een ontwerp veelal is gericht op een tijdshorizon van 30 jaar, kan nu via een dergelijke applicatie ook het ontwerp in zijn ontwikkeling worden gevolgd en aangepast. Dit biedt goede kansen voor inzet in het onderwijs bij bijvoorbeeld de opleidingen Landschapsarchitectuur van Wageningen Universiteit en IAHL Larenstein.

Tenslotte biedt SALIX-2 nieuwe inzichten in de mogelijkheden van Virtual Reality en 'agent' technologie voor projecten binnen de onderzoeksprogramma's Virtuele Groene Ruimte en Delta onderzoek. Zo zijn via dit project meer generieke vraagstukken met betrekking tot de relatie tussen geo-data en VR-model enerzijds alsmede de koppeling tussen VR-model en tijd-ruimte simulatie modellen anderzijds scherper geworden.

Leeswijzer

De nu volgende hoofdstukken geven inzicht in de achtergronden, de technische realisatie en de gebruiksmogelijkheden van SALIX-2. Hoofdstuk 2 biedt een korte bespreking van de beplantingsmethoden en het bijbehorende beheer en vormt als zodanig een copie van de hoofdstukken 2 en 3 van het SALIX-1 rapport [LAM1]. Vervolgens geeft hoofdstuk 3 een korte beschouwing over beplanting en architectonische objecten. Hoofdstuk 4 geeft een overzicht van digitale representatie van beplanting in een 3-dimensionale, digitale werkomgeving, waarna hoofdstuk 5 ingaat op de gebruikte ICT-middelen.

Vervolgens wordt ingegaan op de kenmerken van de SALIX-applicatie. Allereerst wordt het functioneel ontwerp besproken (hoofdstuk 6). Dit functioneel ontwerp is uitgewerkt in een applicatie. De werking van de SALIX-2 applicatie komt in hoofdstuk 7 aan de orde. In hoofdstuk 8 wordt inzicht gegeven in sommige van de technische achtergronden van deze applicatie.

Tenslotte worden vervolgmogelijkheden voor toepassing en onderzoek in hoofdstuk 9 benoemd.

2 Beplantingsmethoden en beheer

Planontwerpen voor de aanleg van tuinen, parken en herinrichtingsgebieden worden veelal vertaald in beplantingsplannen, waarin de aanleg van de beplanting wordt gespecificeerd. Die beplantingsplannen zijn veelal gebaseerd op een traditionele opvatting over aanplant en onderhoud van de beplanting. Die traditionele opvatting staat bekend als de 'blijvers-wijkers' methode. Een methode die zich typeert door hoge onderhoudskosten en veelal afwijkt van de beoogde architectonische doelstellingen per groeifase. Als een reactie daarop is de integrale beplantingsmethode Ruyten ontwikkeld.

De traditionele of 'blijvers-wijkers' beplantingsmethode

Deze beplantingsmethode is van oorsprong ontstaan uit de bosbouw met als doel houtproductie te realiseren. Daarvoor werden in rijen zaailingen geplant met een onderlinge afstand van 1 tot 5 m. Als gevolg van deze dichte plantafstand ontstaat een concurrerende groei waardoor de gewenste rechte takvrije stammetjes worden verkregen.

In de jaren 60 werd dit plantsysteem op grote schaal voor recreatiebossen maar ook in het stedelijk milieu toegepast, vanwege het voordeel van de goedkope aanleg. Door dunningen van de wijkers dienen de blijvende of duurzame planten de ruimte te krijgen om uit te groeien. In tegenstelling tot de bosbouwmethode, waarbij op kaprijke leeftijd de bomen worden verwijderd.

Door de toenemende bezuinigingen ontstond er een druk om op het onderhoud te bezuinigen. Dit leidde tot achterstallig onderhoud en de toepassing van chemische middelen (round-up) om hergroei van afgezette planten te voorkomen. Door grootschalige ingrepen probeerde men toch aan het dunningsregime te kunnen voldoen. Deze werkmethode staat thans algemeen ter discussie. Door ingrijpende en kostbare omvormingen probeert men nu een definitieve wijziging van het onderhoud door te voeren. Het beeld van de beplanting wordt gedomineerd door op rijen geplante bomen en/of struiken met een sterk opgaande vorm.

Bij het blijver- wijkerssysteem staan struik- en boomvormers in groepen van 10 tot 30 stuks in een vak. Hoewel er geen onderhoudsplan is zal er naar alle waarschijnlijkheid gedund gaan worden op het moment, dat struiken of bomen elkaar verdringen in hun groeifase. Er is geen sprake van houtproductie. Deze beplantingsmethode wordt algemeen toegepast in het stedelijk milieu. Het sortiment bestaat daarom uit meerdere soorten in tegenstelling tot het sortiment van een productiebos.

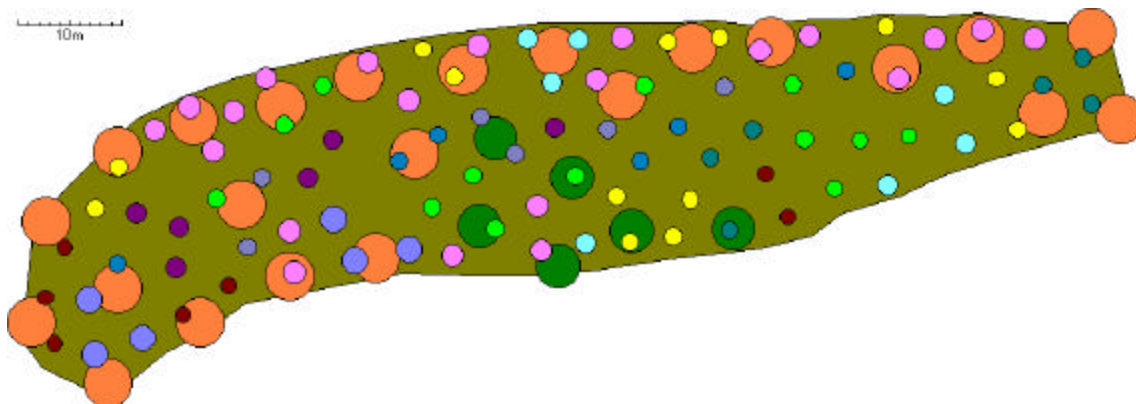
De Integrale beplantingsmethode Ruyten pp

De Integrale beplantingsmethode Ruyten is een alles omvattend en samenhangend plan, waarbij de volgende factoren de basis vormen voor een duurzaam plantsysteem:

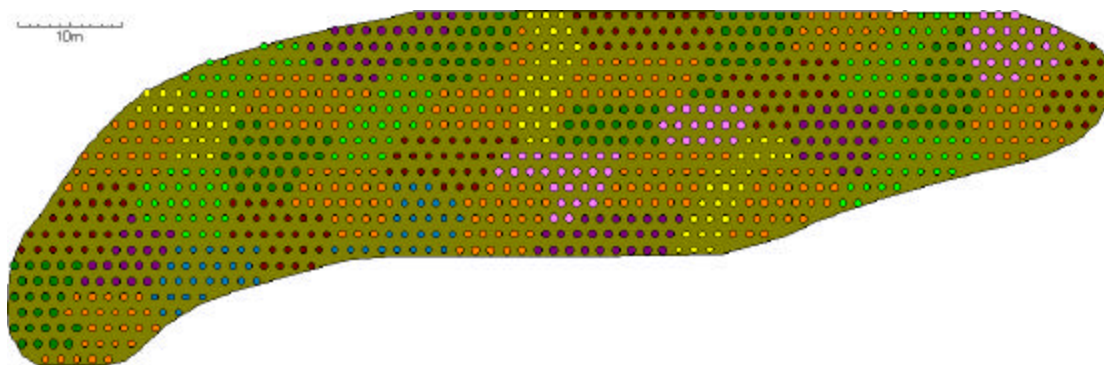
- plantgebruik	de juiste plaats gelet op zon, schaduw en grondsoort
- architectonische waarde	de uiterlijke kenmerken, de bouwwijze en de manier waarop de plant zich ontwikkelt
- natuurlijke ontwikkeling	de verschillende groeifasen waarin de beplanting zich ontwikkelt van jong tot volwassen uitgroei
- beheer	de plantverzorging in overeenstemming met de kenmerken van de beplanting

De integrale benadering houdt in dat een beplantingssamenstelling niet alleen gericht is op het eindbeeld, maar ook op de beelden van de tussenliggende beplantingsstadia: na 2 tot 3 jaar na aanleg een gesloten bodembegroeiing, vervolgens na 6 tot 8 jaar bladverliezende struiken in maximale uitgroei, na 10 jaar bereiken de bladhoudende struiken en coniferen dat stadium en na weer 10 jaar zijn de bomen zover. Er wordt als het ware een scenario ontworpen van bewegende beelden.

Om bij de aanleg de bedoelingen van het ontwerp zichtbaar te hebben, wordt direct met grotere planten begonnen, die als zodanig daarvoor gekweekt zijn: bomen gemiddeld 7 m hoog - 3 m breed en struiken 2 m hoog - 1,5 m breed op onregelmatige en definitieve eindafstanden. Daarmee groeien de planten op in vrijstand. Pas in de fase van maximale uitgroei kunnen de planten elkaar gaan raken. Omdat er geen onderlinge concurrentie is kunnen de planten breed uitgroeien. Daardoor ontwikkelen zich vroegtijdig bloemen en vruchten. De open bosstructuur zorgt voor veel zoomvegetatie met ontwikkelingsmogelijkheden voor een grote diversiteit aan flora en fauna. Het onderhoud in de plantvakken bestaat uit het tweemaal per jaar pleksgewijs maaien met een bosmaaier van de boom- en struikzaailingen, distels of andere ongewenste kruiden. Het maaisel kan blijven liggen. Omdat met dit type onderhoud de successie wordt geremd treedt er geen spontane verjonging op maar kunnen de bomen en struiken worden vervangen in de fase van aftakeling (de geschatte omlooptijd voor eiken 80 - 120 jaar, essen 60 - 80 jaar en de struiken 50 - 70 jaar). Bij deze methode zijn geen chemische bestrijdingsmiddelen nodig.



Figuur 2.1: Voorbeeld beplantingsplan volgens de Integrale beplantingsmethode Ruyten.



Figuur 2.2: Voorbeeld beplantingsplan volgens de blijvers-wijkers methode.

Kostenvergelijking en 2D-groeisimulatie met SIMGRO PP

Voor een vergelijking van de kosten van de twee beplantingsmethoden is bij IMAG in samenwerking met Integralis PP een prototype ontwikkeld van een 2D-groeisimulatie programma genaamd SIMGRO PP. Dit programma legt een relatie tussen de groeiontwikkeling en de daarvan afhankelijke onderhoudswerkzaamheden bij gegeven plantafstanden. De grootte van de beplanting wordt door de seizoenen heen gesimuleerd. Als planten een omvang bereiken die dunning noodzakelijk maakt, wordt dit door het programma gesignaleerd. De dunningen kunnen dan al of niet (virtueel in het programma) worden uitgevoerd. De benodigde tijd voor de dunningen en andere noodzakelijke onderhoudswerkzaamheden werden op basis van data uit het boek 'Tijdnormen voor groenvoorzieningen en buitensportaccomodaties' (IMAG, 1994) berekend. Op basis van de benodigde werktijd en de van de werkmethode afhankelijk ingezette werktuigen kan een kostprijs van de onderhoudswerkzaamheden worden berekend. Samen met de kosten van gebruikte hulpmiddelen en de kosten van de oorspronkelijke aanplant maakt dit een vergelijking over een langere looptijd van bijvoorbeeld 20 jaar mogelijk. De grafische weergave van de twee-dimensionale projectie van de boomspiegels geeft een indruk van het bereikte eindresultaat.

Simulatie van groei en beheer

In het 2-dimensionale groei- en beheersmodel (SIMGRO PP) wordt gebruik gemaakt van plattegrond waarop de beplanting wordt weergegeven als een cirkel met een middelpunt (zie figuur 2.1 en 2.2). Het middelpunt geeft de stam aan en de cirkel de omtrek van de kroon bij een bepaalde leeftijd. De kleur van cirkel verwijst naar de soort beplanting.

De groei wordt gesimuleerd doordat een functie via een logistische groeicurve de mate van transformatie van de cirkel per jaar berekent. De transformatie van de functie is niets anders dan een opschaling van de cirkel.

Op het moment dat cirkels elkaar voor een bepaald (van te voren in te stellen) gedeelte overlappen, worden de betreffende cirkels rood gekleurd. De kleur rood geeft in dit verband aan dat er 'iets' moet worden gedaan; een beheersmaatregel moet worden uitgevoerd.

Het uitvoeren van een beheersmaatregel houdt bij SIMGRO PP in dat er een 'individueel' beplantingsobject (boom of struik) moet worden verwijderd. Dit gebeurt door op het scherm aan te geven welke rode cirkel moet verdwijnen. Nadat voldoende concurrerende cirkels zijn verwijderd, verandert de rode kleur van de resterende cirkels weer terug in de soortspecifieke kleuren en kan de groei worden gecontinueerd.

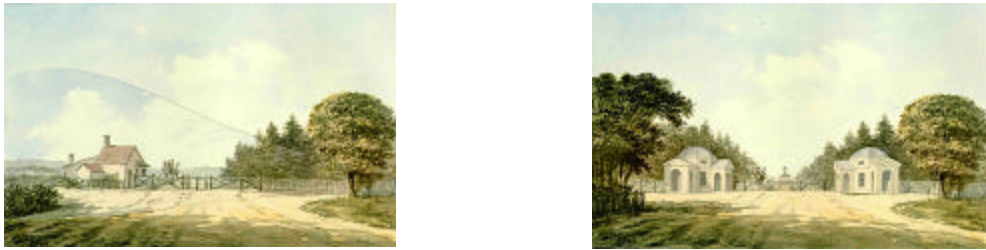
Deze ingreep staat in het traditionele groenbeheer (aangeplant volgens de 'blijvers-wijkers' methode) bekend als dunnen. Via uitkap van het oorspronkelijk aangeplante bosplantsoen wordt gestreefd naar het realiseren van het beoogde architectonische ruimtelijke 'wensbeeld'.

De andere beplantingsmethode betreft de door Ruyten [RUY1] ontwikkelde 'Integrale beplantingsmethode Ruyten'. In deze aanpak wordt geen 3-jarig bosplantsoen aangeplant, maar bomen en struiken van een hogere leeftijd, waarmee in hoofdlijnen het beoogde ruimtelijke 'wensbeeld' al meteen bereikt wordt. De groei van de beplanting vraagt dan ook om veel minder onderhoud, daar de individuele beplantingssoorten al op juiste positie en onderlinge afstand staan. Groei betekent hier vooral uitgroeien. Terwijl groei in de 'blijvers-wijkers' methode geheel afhankelijk is van de mate van uitkappen van concurrerende bomen en struiken.

Simulatie van de Integrale beplantingsmethode Ruyten vereist in tegenstelling tot de 'blijvers-wijkers' methode vooral meer inzicht de groei van het architectonisch concept, waarbij de beheers-ingrepen minimaal zijn.

3 Beplanting en Landschapsarchitectuur

Het landschapsarchitectonisch ontwerpen is een fascinerende activiteit, omdat via een modelmatige aanpak (het ontwerp) een gedachteconstructie over een toekomstig ruimtebeeld wordt gematerialiseerd. In het materialiseren of concretiseren van deze ruimtebeelden houdt men rekening met de driedimensionale werking van de objecten waarmee dat ruimtebeeld kan worden gerealiseerd. Die objecten kennen, dit in tegenstelling tot de architectuur of stedenbouw, de fascinerende eigenschap dat het levende materialen zijn. Deze materialen, oa. bomen en struiken, vormen de primaire bouwstenen van de architectonische objecten waarmee landschapsarchitecten buitenruimten vormen. In dit opzicht zijn oa. de Red Books van Repton zeer bepalend geweest. 'Voor'- en 'na'-afbeeldingen tonen de ruimtelijke effecten van een ontwerpingsgreep aan de hand van oa. beplantingselementen in het bestaande landschap (figuur 3.1).



Figuur 3.1: Voor en na afbeeldingen van Repton

Over de relatie tussen beplantingselementen en landschapsarchitectonisch objecten heeft oa. Simonds [SIM1] gepubliceerd. Simonds introduceerde een primair aantal ruimtevormende, architectonische objecten:

- *base plane*: een massa-element waarmee het grondvlak wordt verhoogd. Veelal wordt dit element gebruikt om geleiding van de ruimte te creëren;
- *overhead plane*: een massa-element waarmee een 'dak' wordt gemaakt. Dit element wordt gebruikt om beslotenheid of 'vensters' op de omgeving te creëren;
- *vertical plane*: een massa-element waarmee een 'wand' wordt gemaakt. Dit element wordt gebruikt om ruimtelijke scheiding te realiseren.

Ieder van deze objecten is gebaseerd op het gebruik van beplantingselementen. Figuur 3.2 toont, via de grijze balken, de relaties tussen deze beplantingselementen en architectonische objecten. De beplantingselementen zijn kruiden (1.1), struiken (2.1, 2.2 en 2.3) en bomen (3.1 en 3.2) met een verschillende habitus (via de getalsaanduiding .1, .2 en .3). De architectonische objecten zijn de *base plane* (B.1 en B.2), de *vertical plane* (C.1 en C.2) en de *overhead plane* (D.1 en D.2). De getalsaanduiding .1 of .2 verwijst naar respectievelijk een enkelvoudige, smalle vorm van het object of de meervoudige, brede vorm van het object.

Belangrijk is het architectonische object 'solitaire' (A). Via dit object, dat in principe uit een beplantingselement bestaat, wordt een eenvoudige ruimtebepaling of -markering (bijv. *landmark*) nagestreefd. Simonds geeft in zijn boek diverse voorbeelden van de werking van een dergelijk solitaire element.

In het ontwerpproces gaat het niet alleen om het bedenken van de individuele architectonische objecten, maar ook om de relaties tussen die architectonische objecten. In deze studie is vooral aandacht besteed aan de 'bouwstenen' van het landschapsarchitectonisch ontwerp (de beplantingselementen) en de wijze waarop deze kunnen overgaan in de genoemde architectonische objecten.

Aandacht aan deze overgang van element naar object is bijzonder, daar de meeste architecten vooral op het niveau van objecten werken en die vervolgens via beplantingsplannen vertalen naar de 'bouwstenen'. Terwijl in de werkelijkheid het beplantingsplan wordt uitgevoerd en na verloop van een aantal jaren, en passend groenbeheer (via een beheersplan), het architectonisch bedoelde ruimtelijk beeld moet worden behaald. Dit betekent dat realisatie van het ontwerp gebaseerd is op groei en beheer van de individuele beplantingselementen.

Voor de ontwikkeling van SALIX-2 betekent deze relatie tussen beplantingselementen en architectonische objecten dat gezocht is naar een oplossing waarbij de gesimuleerde groei van beplantingselementen op bepaalde momenten leidt tot inzicht in het ontstaan van architectonische objecten.

Architectural objects	A solitaire	B.1 base plane	B.2	C.1 vertical plane	C.2	D.1 overhead plane	D.2
Planting objects							
1.1							
2.1							
2.2							
3.1							
3.2							

Figuur 3.2: Relatie tussen beplantingselementen (*plantings objects*) en architectonische objecten (*architectural objects*) (uit: [LAM2])

4 Digitale representatie van beplanting

Het weergeven van beplanting is letterlijk een kunst op zich. Een willekeurige wandeling langs schilderijen uit diverse perioden toont een variëteit aan realistische en abstracte vormen van grafische weergave van bomen, struiken en planten. Een weergave die gebaseerd is op het gebruik van grafische symbolen en variabelen, die gerealiseerd worden met traditionele tekenmaterialen. Bovendien gaat die variëteit gepaard aan een projectievorm van waaruit de werkelijkheid wordt bekeken, geschilderd cq. beschreven.

Porter [POR1] geeft aan hoe in de loop der eeuwen door architecten getracht is de werkelijkheid te visualiseren om daarmee de ruimtewerking van gebouwen in hun omgeving te verduidelijken. Het gebruik van verschillende projectievormen heeft daarbij veranderingen ondergaan. Oorspronkelijk werd gewerkt met de plattegrond als projectievorm, maar in de loop der eeuwen wordt deze in de West-Europese wereld aangevuld met andere vormen zoals het perspectief, de doorsnede en het aanzicht. Al deze vormen werden gehanteerd om het inzicht de ruimtelijke elementen en de door hen gevormde ruimte te verbeteren.

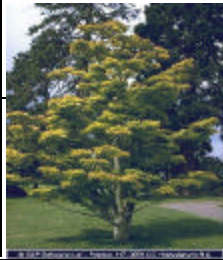
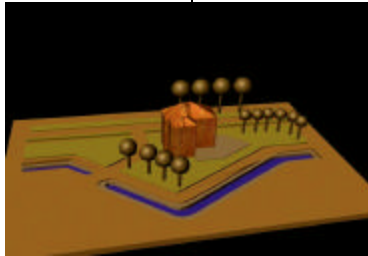

Typerend voor al deze projectievormen is het statische en gefixeerde karakter van de visualisatie. De representatie van de werkelijkheid is die vanuit een gezichtspunt en vanuit een bepaalde projectie. Met de intrede van de computer en met name het onderzoek naar vluchtsimulatoren zoals dat door NASA is geïnitieerd heeft geleid tot hedendaagse visualisaties met de computer. Deze visualisatie bieden in principe de mogelijkheid om dynamisch en interactief de bestaande of de bedachte werkelijkheid te aanschouwen en wellicht zelfs op in te grijpen.

Daartoe zal van die werkelijkheid een digitale, drie-dimensionale weergave moeten worden gemaakt: een virtuele werkelijkheid (*Virtual Reality* (VR)).

Een digitale representatie van beplanting in een VR kan op verschillende wijzen worden gerealiseerd (tabel 4.1). Elementair daarbij is echter de scheiding tussen de beschrijving van de geometrie en de grafische weergave van de beplanting. De geometrische beschrijving bestaat uit een coördinatenset bestaande uit X, Y en Z waarden en de onderlinge relaties tussen de daarmee beschreven punten: 3D model. De grafische beschrijving bestaat uit digitale plaatjes: bitmap.

De 3D modellen en bitmaps kunnen verschillende niveaus van detail hebben. Een eenvoudig 3D-model, bijvoorbeeld een plat vlak, kan aan de voor- en achterzijde digitaal beplakt worden met een zeer gedetailleerde bitmap. Daar tegenover staat dat een zeer gedetailleerd 3D-model, dat bestaat uit zeer vele kleine 3D objecten, voorzien kan zijn van vele bitmaps met slechts een enkele, homogene kleur. Overigens wordt zo een uiterst gedetailleerd 3D model vaak een 'compound object' genoemd.

Tabel 4.1 Digitale vormen van 3D-weergave van beplanting: een combinatie van geometrische objecten (verticaal) en bitmap-techniek (horizontaal)

Model	Bitmap	a.Elementary color	b.Pictures	c.Multi-layer map
Model description 2/3d				
Planes [3D studio Max]				1.c
Primitives [3D studio Max]		2.a 	2.b	2.c
Compound Objects [http://www.mediamachine.nl/prod/Treer.html]		3.a	3.b 	3.c

Het 3D-model kent dus een verbijzondering door van een simpel object (een vlak of 'plane'), via elementaire 3d objecten of 'primitives' met behulp van die eenvoudige objecten en primitives zeer gedetailleerde samengestelde objecten of 'compound objects' te maken.

Vergelijkbaar kunnen de bitmaps slechts één kleur representeren (elementary color), dan wel een complex samenstel van vele kleuren en daardoor vormen weergeven (pictures) en zelfs gecombineerd worden (multi-layer map).

In SALIX-2 is gekozen voor het gebruik van *primitives* met een eenvoudige bitmap (type 2a in tabel 4.1) om de beplantingselementen van het type 2.1, 2.2 en 3.2 (zie figuur 3.2) weer te geven, waarbij per soort een unieke kleur wordt toegekend. Weergave van de architectonische objecten gebeurt eveneens via *primitives*. Deze krijgen eveneens een unieke kleur.

5 Inzet ICT-componenten

Voor de 3D-representatie en groei-simulatie van beplantingen zijn diverse technologieën beschikbaar. In dit project wordt de uitdaging vooral gezocht in het ontwikkelen van een virtuele werkelijkheid die zowel 'non-immersive' als 'immersive' kan worden gebruikt.

'Non-immersive' wil zeggen dat de gebruiker van het systeem nog steeds de virtuele werkelijkheid bekijkt via het scherm en daarin met behulp van een '3D-besturingsmiddel', bijvoorbeeld een 2D-muis of een stuurwiel, kan navigeren in en interacteren met het model.

'Immersive' wil zeggen dat de gebruiker als het ware onderdeel uitmaakt van de virtuele werkelijkheid en daarin kan rondlopen. De gebruiker is dan uitgerust met bijvoorbeeld een stereo-bril en kan daarmee oa. via een 'cave' volledig onderdeel worden van de virtuele omgeving.

Daartoe is in dit project gekozen voor een ontwikkelomgeving die bekend staat als Virtual Reality Modelling Language (VRML). VRML biedt enerzijds het voordeel dat zij een opstap biedt voor zowel 'immersive' als 'non-immersive' applicaties en zich kenmerkt doordat het een standaard vormt. Anderzijds kunnen de applicaties via internet en client-server applicaties worden verspreid en gebruikt (daartoe is wel een VRML-viewer nodig) en de VRML ontwikkel omgeving is open en goedkoop.

Om de relatie te kunnen leggen met digitale geografische gegevens, ten behoeve van de koppeling aan andere terreingegevens zoals relief, bodem, grondwater, kadaster kan ook gebruik worden gemaakt van Geo-VRML.

Daar VRML een extensie is van HTML dan wel XML, is ook een koppeling mogelijk met HTML- en XML-code, alsmede met Java en Javascript.

VRML technologie

De VRML-standaard is in 1994 geïntroduceerd ten behoeve van de constructie en weergave van statische, 3-dimensionale, geometrische modellen. Deze eerste VRML-versie was in grote lijnen gebaseerd op OpenInventor format van Silicon Graphics. Latere versies hebben een afwijkend karakter gekregen. In 1997 was de specificatie van VRML 2.0 of VRML 97 gereed. Met deze versie is het ook mogelijk om animaties uit te voeren, om dynamische elementen te gebruiken en om interactie met de 3D-wereld uit te voeren via scripting of JAVA-programma's [URL9, URL10].

In de meest recente versie van VRML zijn nieuwe componenten toegevoegd o.a. ten behoeve van animatie, interactie en uitbreiding van reeds beschikbare functionaliteiten en capaciteiten. Bovendien zijn er VRML-'browsers' beschikbaar voor bijna alle computer-platforms, voornamelijk in de vorm van web-browser 'plugin'.

Het gedrag van het model kan worden gecontroleerd via daartoe te schrijven VRML-scripts of JAVA-code gebaseerde programmatuur. Deze uitbreidingen bieden nieuwe, bruikbare perspectieven zoals het modelleren van object-gedrag en 'hot objects' [GAG1]. Helaas ontbreken momenteel nog de verbeteringen van de 'render'-faciliteiten [WAT1]. Daarnaast is VRML nog steeds alleen beschikbaar als een zgn.

interpretatie-taal ('interpreted language') en daardoor is de werking van de applicaties gemaakt met VRML verhoudingsgewijs traag.

De mogelijkheden van de beoogde applicatie kunnen voor een belangrijk deel met bekende database technologie (o.a. in een client-server omgeving) in relatie tot VRML en de bijbehorende programmeer-omgeving (JAVA) worden gerealiseerd. Toch kent dit project een tweede technologische uitdaging en deze betreft het gebruik van 'agent' technologie.

Agent technologie - Multi-Agents Systems

Een Multi-Agent System wordt gebruikt om een verzameling van samenwerkende 'agents' in een bepaalde reken-omgeving te duiden. Een 'agent' is een computerprogramma dat specifieke taken verricht. Die taken kunnen zijn het aanpassen van zowel de omgeving via communicatie, besluitvorming en realisatie, dan wel het aanpassen van zichzelf via perceptie, redeneren en leren [FER1, WEI1]. Multi-agent systems variëren dus in de aard van de mogelijkheden van de agents, de wijze van interactie tussen agents, de taak van de agents in een computersysteem en de relatie van de 'agent' tot de omgeving waarin zij actief moeten zijn. Bijlage 1 geeft daarvan een Engelstalig overzicht.

Momenteel zijn populaire toepassingsvelden die van E-commerce en opsporen van Web-gebaseerde informatie [KLU1]. Ontwikkelingen zijn gaande op de terreinen van digitale bibliotheken, telecommunicatie en maatproductie. De belangrijkste ondersteunende technologieën omvatten mens-agent interactie en visualisatie van informatie-'ruimten' ('information spaces') oa. in de vorm van 'visual data mining' [CAR1, CHE1, KEI1, WAC1].

Mens-agent interactie is in hoofdlijnen gebaseerd op de 'indirecte management' metafoor. Deze metafoor geeft aan dat de gebruiker betrokken is in een co-operatief werkproces waarin zowel de mens als de agent het initiatief kunnen nemen in communicatie, monitoren en bijsturen van processen dan wel in de directe uitvoering van [LEW1, MAY1]. Een dergelijke interactie vereist:

- Verwerking en analyse van de gebruikers input door middel van bijv. spraak en andere signalen;
- Onderhouden van het interactie-proces doordat de agent kennis heeft over het toepassingsveld, de context van de toepassing, de middelen die gebruikt worden voor communicatie en de taken, dan wel methoden, die kunnen worden ingezet;
- Presentatie van de output van agents via grafische, gesproken of nonverbale middelen.

Visualisatie van 'informatie-ruimten' kan oa. via 'virtual reality' (VR) technieken, zoals mogelijk via VRML of haar opvolger X3D. Tele-immersieve omgevingen, zoals CAVE, ImmersaDesk3, CAVE6D en TIDE, bieden al mogelijkheden op de gebruiker met behulp van agents te begeleiden door het virtuele landschap. De eerste voorbeelden daarvan vindt men vooral in 3D-omgevingen (oa. landschappen, winkelcentra [ISH1]).

Opvallend is dat er vele cyber-figuren zijn ontwikkeld (denk aan Lara Croft) door oa. NetSage, Extempo Imp, Microsoft, DFKI, and MIT Media Lab. Deze 'cybers' zijn

veelal gebaseerd op karaktereigenschappen die via agents zijn toegekend aan deze cybers [KLU1].

Iedere cyber simuleert een andere rol en vanuit deze rol probeert de cyber een gebruiker van het systeem (dat kan dus een internet-applicatie zijn) te overtuigen om bijvoorbeeld een produkt te kopen.

De belangrijkste doorbraak van 'multi-agent systems' hangt uiteindelijk af van hardware en software ontwikkelingen ten behoeve van aanzienlijke verbeteringen van 3D graphics, opslagcapaciteit, superbreedband verbindingen voor Internet en standaardisatie ten behoeve van multi-media applicaties op het Web (denk aan SMIL) [URL8].

De 'information-agent' en de 'interface-agent' geven een beeld van de mogelijkheden. Het eerste type heeft toegang tot veel en verschillende (naar inhoud en locatie) informatie. Hun taak is gericht op het actief zoeken voor bruikbare informatie ten behoeve van de gebruiker dan wel andere agents. Dit houdt in opsporen, opvragen, analyseren, waarderen, aanpassen, aanmelden en doorgeven.

De 'interface-agent' helpt de gebruiker bij het uitvoeren van een (complexe) taak. Deze 'agent begeleidt, traint en ondersteunt de gebruiker dan wel een andere 'agent'. Vooral in coöperatieve omgevingen en leeromgevingen lijkt dit type agent zinvol.

6 SALIX-2 ontwerp

Het oorspronkelijke SALIX-1 prototype is gebaseerd op een conceptueel ontwerp dat zo goed mogelijk aansluit bij SIMGRO. Dit houdt in dat een digitaal beplantingsplan wordt ingevoerd en vervolgens op soortafhankelijke groeikenmerken wordt doorgerekend. Beplanting die met elkaar lijkt te vergroeien wordt vervolgens opgespoord.

Het conceptueel ontwerp (fig. 6.1) van SALIX-1 kent de volgende kenmerken.

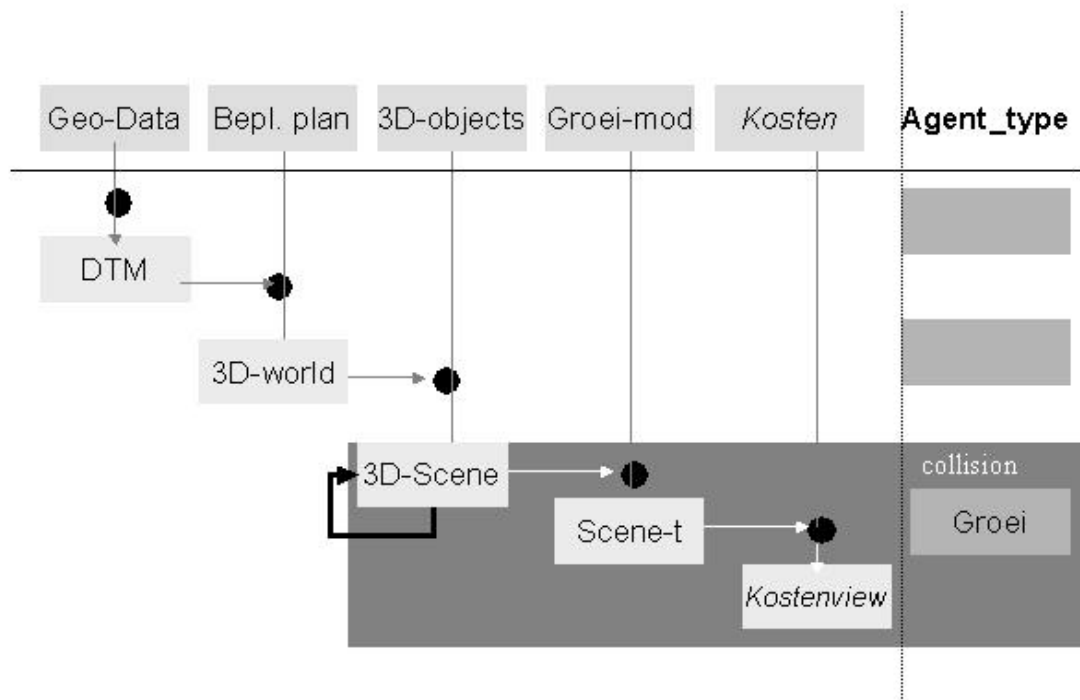
De input van de applicatie bestaat uit:

- terreingegevens: deze gegevens vormen een beschrijving van de terreinopbouw zonder opgaande beplanting. Bij voorkeur zijn deze gegevens via hun georeferentie gekoppeld aan het RijksDriehoekstelsel (RD), waardoor een relatie met geo-informatie (topografische, kadastrale en hoogte-gegevens) is te realiseren;
- beplantingsplan: digitale beschrijving van het beplantingsplan die veelal afkomstig zal zijn uit een CAD of 3D-visualisatie omgeving.
- beplantingsdata: gegevens waarmee een beplantingsplan kan worden gemaakt. Het gaat hierbij met name om soortkenmerken;
- groei-modellen: op basis van empirisch onderzoek zijn voor een aantal soorten, onder verschillende fysieke omstandigheden groeigegevens verzameld. Deze groei-gegevens zijn omgezet naar groei-curven.
- onderhoudgegevens: deze gegevens zijn primair afkomstig van WUR-IMAG en geven via de soort, architectonische rol en omvang van beheer en onderhoud (beheerpakket) een kostenindicatie voor het beplantingsplan.

De handelingen die met deze gegevens verricht worden bestaan uit:

- opbouw 3D model mbt. terreingegevens, resultaat 3D-model van het terrein [3D-terrein];
- opbouw beplantingsplan in 3D, resultaat: 3D model van terrein met beplanting [3D-beplanting];
- aanpassing 3D beplanting; resultaat: aangepast 3D model van terrein op basis van de locatie van de beplanting[3D-beplanting] ;
- groei-simulatie: resultaat: aangepast 3D-model van terrein en beplanting op basis van de temporele kenmerken van de beplanting [3D-beplanting & collision];
- kostenberekening: berekening van actueel beplantingsplan, resultaat: weergave kosten [kostenview].

De agent-technologie kan op diverse wijzen in de applicatie worden aangewend. In figuur 6.1 zijn deze mogelijkheden aangeduid als *agents* 1 t/m 5. Uiteindelijk is in het prototype gekozen voor een *agent*, aangeduid in figuur 6.1 als agent 4, die continu de mate van overlap (*distortion*) tussen zelfstandige beplantingselementen, controleert.



Figuur 6.1 Concept systeemontwerp SALIX-1 [zie LAM1])

Het systeem-ontwerp van SALIX-2 (figuur 6.2) geeft aandacht aan de volgende onderdelen:

1. database beheer;
2. verbeterde navigatie in het VR-model;
3. interactie met het VR-model;
4. definitie van solitaire en niet-solitaire beplantingsobjecten;
5. transformatie van beplantingsobjecten in architectonische objecten.

Het databasebeheer richt zich op de opslag van de verschillende ontwerpen in VRML en in beplantingsdatabases en het onderscheid in de beplantingsdatabase (Bepl. Plan in fig. 6.2) tussen solitaire en niet-solitaire beplantingselementen. Dit onderscheid is van belang met het oog op de transformatie van beplantingsobjecten in architectonische objecten. Het databasebeheer wordt oa. gerealiseerd via aanpassingen in Access- en het schrijven van Java-code. In figuur 6.2 is dat mede terug te vinden in de terugkoppeling.

Verbetering van navigatie komt voort uit een ontwerpbehoefte. Het blijkt tijdens het ontwerpen van groot belang dat er continue gevarieerd kan worden van beschouwersperspectief. Voor het bekijken en beschouwen van het ruimtelijk effect (3D) van een ontwerp wil men bij voorkeur in het ontwerp staan en via een perspectivische projectie dit effect bestuderen. In verband met het bekijken en beschouwen van het 2-dimensionale patroon wil men als het ware boven de kaart

hangend, via een orthogonale projectie, het ontwerp bestuderen. In de SALIX-2 applicatie is dan ook een mogelijkheid ingebouwd om deze twee perspectieven alternerend te gebruiken.

Ook bleek er behoefte te bestaan om een willekeurig punt in het plan aan te wijzen en vervolgens naar dat punt te gaan en als het ware staande in het model via een perspectivische projectie vanaf dit punt het ontwerp te bekijken. Om dit daadwerkelijk snel te kunnen doen moet behalve de locatie ook de kijkrichting worden opgegeven.

De interactie met het model beperkte zich in Salix-1 vooral tot het schuiven met beplantingsobjecten. In Salix-2 is, met het oog op de inzet in het ontwerpproces, gekozen voor een uitbreiding van de interactie via het toevoegen en verwijderen van beplantingsobjecten. Bij het toevoegen kan tevens de aard van het beplantingsobject (solitair of niet-solitair) worden opgegeven. Deze definitie van het beplantingsobject wordt vastgelegd in de beplantingsdatabase. Dit aspect van navigatie en interactie is niet expliciet terug te vinden in figuur 6.2 .

Tenslotte is de transformatie van beplantingsobjecten in architectonische objecten ingevoegd. Leidraad voor deze transformatie zijn de relaties uit figuur 3.2 geweest. Concreet gaat het daarbij om de omzetting van de geometrie van de 'primitives' van de beplantingsobjecten in 'primitives' die de architectonische objecten duiden.

Uitgaande van de twee soorten objecten (bomen en struiken) zijn drie mogelijke combinaties van overlappende objecten mogelijk. Een boom overlapt met een boom, een boom overlapt met een struik, of een struik overlapt met een struik.

Per definitie is het niet mogelijk dat een boom of een struik overlapt met een architectonisch object, omdat overlap altijd bepaald wordt voor de objecten afzonderlijk.

Wanneer twee bomen met elkaar overlappen, zal het architectonisch object gevormd worden rond het loof van beide bomen omdat er geen wezenlijke overlap tussen hun stammen bestaat. Twee overlappende bomen vormen als architectonisch object een **dak** (overhead plane).

Struiken hebben geen stam. Wanneer er zodoende twee struiken met elkaar overlappen vormen beide objecten in hun geheel een architectonisch object: een **vlak** (horizontal plane).

Een bijzonder geval treedt op wanneer een boom overlapt met een struik. Het architectonisch object dat hier gevormd reikt van de grond tot aan het hoogste punt van het loof van de boom: er ontstaat een **wand** (vertical plane).

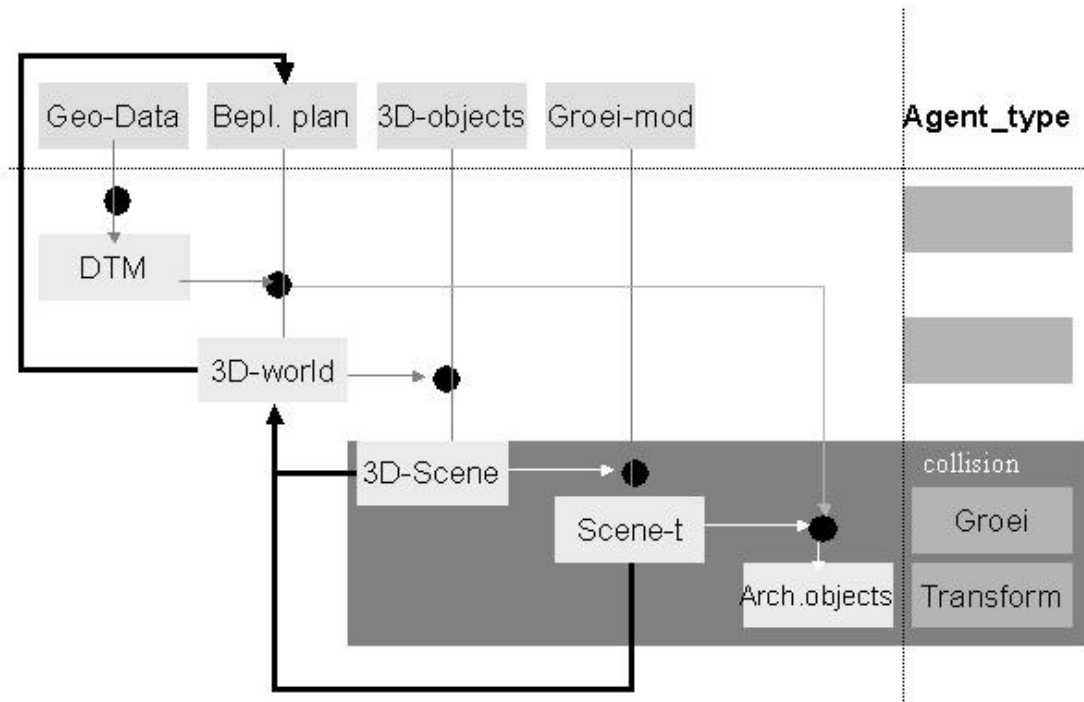
Om architectonische objecten te visualiseren moet van ieder object bekend zijn of zij zich solitair of niet-solitair moet gedragen in het geval dat het object overlapt met andere objecten. Om deze reden is het mogelijk om bij het toevoegen van objecten aan het landschap aan te geven wat het gedrag (solitair of niet-solitair) van dit object moet zijn.

Het is op verschillende manieren mogelijk te controleren of er overlappende objecten in het landschap bestaan:

- Op een willekeurig tijdstip kan bepaald worden of er overlap bestaat.
- Tijdens het groeiproces van de objecten wordt voortdurend gecontroleerd of er overlappende objecten ontstaan tengevolge van de groei van de in het landschap aanwezige objecten.

Dit geheel met betrekking tot de transformatie van beplantingsobjecten in architectonische objecten is in figuur 6.2 terug te vinden in de overgang van Scene naar Scene-t (een volgende tijdstip) naar Arch.obj, (de omzetting van de beplantingsobjecten uit Scene-t in architectonische objecten in Scene-t). Via specifieke 'agents' vindt de detectie en de transformatie plaats.

Het conceptueel ontwerp richt zich daarmee vooral op de ondersteuning van de architectonische activiteiten en niet op ondersteuning van het groenbeheer zoals dat in Salix 1 aan de orde was.



Figuur 6.2 Conceptueel ontwerp Salix-2

7 Functionaliteit SALIX-2

Dit hoofdstuk bevat een beschrijving van de functionaliteit die Salix 2 biedt. De beschrijving is vanuit het oogpunt van de gebruiker van Salix 2 opgesteld. Daartoe is het hoofdstuk in paragrafen opgedeeld.

Paragraaf 7.1 beschrijft hoe het Salix 2 programma gestart en afgesloten kan worden. In paragraaf 7.2 wordt ingegaan op de typen objecten die Salix 2 ondersteund. Vervolgens volgt een overzicht van de gebruikersinterface met daarbij een grove indeling in de geboden functionaliteit. Op de functionaliteit wordt nader ingegaan in de paragrafen 7.4 (navigatie), 7.5 (werken met beplantingsplannen), 7.6 (werken met objecten) 7.7 (groei van objecten) en 7.8 (visualisatie van overlappende objecten). Tenslotte beschrijft paragraaf 7.9 de overige functies van Salix 2.

7.1 Opstarten en afsluiten

Het Salix programma draait in de browser Microsoft Internet Explorer, versie 5.

Salix is het best te bekijken met de volgende instellingen:

- Resolutie: 1024 x 768
- Kleurendiepte: 32 bits
- Browser:
 - Scherm gemaximaliseerd
 - Knoppenbalk, adresbalk en 'links'-balk verwijderd

Het programma is op te starten door het bestand [Index.html](#) te laden.

Het programma kan afgesloten worden door het browservenster te sluiten. Voor het venster verdwijnt, wordt eerst het geheugen vrijgemaakt. Hierdoor kan het zijn dat het venster niet direct verdwijnt.

7.2 Ondersteunde objecttypen

Salix 2 onderscheidt 5 verschillende objecttypen. Het betreft twee bomen en drie struiken:

Bomen

Fraxinus excelsior

Quercus robur

Struiken

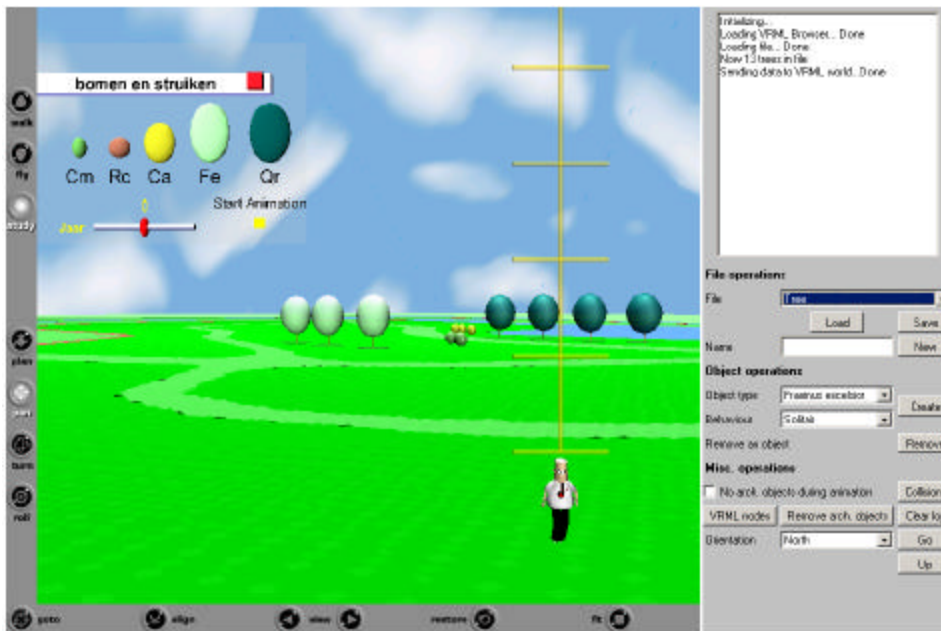
Corylus avellana

Rosa canina

Cornus mas

7.3 De gebruikersinterface

De gebruikersinterface van Salix-2 bestaat uit twee delen. In het linkergedeelte van het browservenster is een driedimensionale omgeving te zien waar de beplantingsplannen gevisualiseerd zullen worden. Rechts is het besturingsgedeelte weergegeven waarmee de beplantingsplannen bestuurd kunnen worden. Figuur 7.1 toont gebruikersinterface van Salix-2:



Figuur 7.1 Gebruikersinterface Salix-2

Het linker gedeelte van de figuur hierboven toont de driedimensionale weergave van een landschap. U ziet hier verschillende bomen en struiken in het landschap geplaatst. Verder treft u een legenda aan (linksboven), als ook een representatie van een persoon om de grootte van de objecten aan te staven.

In het rechter gedeelte van de figuur hierboven kunnen vier gedeelten worden onderscheiden. Hieronder wordt kort de functionaliteit van ieder van de gedeelten opgesomd.

- Tekstveld gedeelte

In het tekstveld (rechtsboven in het scherm) verschijnt een korte beschrijving van de door de gebruiker uitgevoerde handelingen.

- Gedeelte File operations

In dit gedeelte (2^{de} deel van bovenaf in het rechterschermddeel) is het mogelijk bestaande beplantingsplannen te laden of te bewaren. Daarnaast is het mogelijk nieuwe beplantingsplannen te maken.

- Gedeelte Object operations

In dit gedeelte is het mogelijk objecten aan het beplantingsplan toe te voegen, of objecten van het beplantingsplan te verwijderen (3^{de} deel van bovenaf in het rechterschermdedeel).

- Gedeelte Misc. operations

In dit laatste gedeelte is het onder andere mogelijk de plaats van de gebruiker in het landschap te preciseren en het tekstveld leeg te maken (4^{de} deel van bovenaf in het rechterschermdedeel).

7.4 Navigatie

De driedimensionale representatie maakt het de gebruiker mogelijk door het landschap te navigeren alsof hij zich daadwerkelijk in het landschap bevindt.

In Salix 2 is het op verschillende manieren mogelijk door het landschap te navigeren en zo de beplantingsplannen gedetailleerd te bestuderen: het is mogelijk door het landschap te bewegen, of het landschap vanaf voorgedefinieerde gezichtspunten te observeren. Tenslotte is het mogelijk naar een zelf op te geven plaats in het landschap te springen om bijvoorbeeld op die plek de locatie van bepaalde objecten te wijzigen.

Navigatie door het landschap

De functionaliteit om de gebruiker door het landschap te laten lopen of vliegen wordt geleverd door de zogenaamde browser. Zo is het mogelijk te 'wandelen' of te 'vliegen' door het landschap, als ook het landschap vanaf een afstand te bekijken en het als het ware te roteren. Deze opties kunnen aan de linkerkant van het linkergedeelte van Salix 2 geselecteerd worden. De exacte werking van navigatie door het driedimensionale landschap is afhankelijk van de gebruikte 3D browser. Voor Salix-2 wordt de Cortona VRML client aanbevolen (zie paragraaf 7.1).

Gezichtspunten in het landschap

In het landschap zijn verscheidene posities gedefinieerd van waaruit het landschap bekeken kan worden. Onder andere de volgende gezichtspunten (*view points*) worden ondersteund:

- Start = perspectivische projectie: ooghoogte 1.70 meter
- Top = orthogonale projectie
- West = perspectivische projectie
- Noord = perspectivische projectie: vogelvlucht

De gebruikte 3D browser stelt de gebruiker van Salix 2 in staat een gewenst gezichtspunt te selecteren. Dit kan door op de rechter muisknop te klikken en vervolgens de optie 'Viewpoints' te selecteren. Figuur 7.2 toont ter voorbeeld het selecteren van het gezichtspunt 'noord':



Figuur 7.2 Gezichtspunten kiezen

Navigatie naar op te geven plaats in het landschap

Tijdens het werken met beplantingsplannen kan bij de landschapsarchitect de behoefte ontstaan om bepaalde objecten van zeer dichtbij te bekijken. Hoewel het mogelijk is om op de hierboven beschreven wijze naar de gewenste plek te navigeren, duurt dit vaak lang en vergt dit enige training. Daarom is het in het besturingsgedeelte (4^{de} deel van boven aan de rechterkant van het scherm) van Salix-2 mogelijk om naar een zelf te definiëren plaats in het landschap te navigeren. Figuur 7.3 toont het betrokken gedeelte in de gebruikersinterface.



Figuur 7.3 Navigatie naar een plaats in het landschap

Allereerst is het noodzakelijk de gewenste kijkrichting te bepalen. Dit kan door in het pull-down menu een van de vier windrichtingen (North, East, South en West) te kiezen.

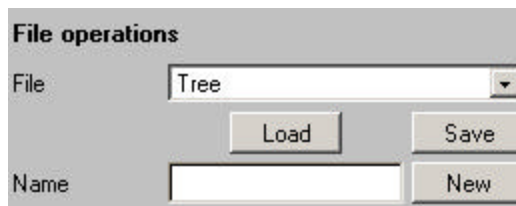
1. Vervolgens klikt u op de knop Go.
Het opschrift van de knop Go verandert in Click world ten teken dat u nu op een plek in de VR-wereld kunt klikken.
2. Klik op de gewenste plek in het landschap (VR-wereld).
Let op dat u op een plek op het landschap klikt, dus niet daarbuiten of op een geplaatst object.

Het gezichtspunt op het landschap verandert nu: de gebruiker bevindt zich nu op de aangeklikte positie in de wereld, kijkend in de vooraf opgegeven richting.

De gebruiker kan nu van nabij bijvoorbeeld de positie van een object wijzigen (zie paragraaf 7.6). Als de gebruiker naar tevredenheid wijzigingen heeft aangebracht, is het mogelijk de plek waar wijzigingen zijn aangebracht van iets grotere hoogte te bekijken. Met de knop Up kan de gebruiker vanaf de huidige positie in de wereld het gezichtspunt omhoog verplaatsen en vanaf dat punt de wereld vanuit top-view bekijken. Op deze wijze hoeft de gebruiker niet geheel uit te zoomen om de plek van wijzigingen in meer context te bekijken.

7.5 Werken met beplantingsplannen

Om landschapsarchitecten in staat te stellen te experimenteren met verschillende beplantingsplannen voor een landschap moet het programma de invoer en het ontwerp van meerdere beplantingsplannen ondersteunen. Salix 2 biedt de functionaliteit om beplantingsplannen te laden, te bewaren, en nieuwe beplantingsplannen te maken. Figuur 7.4 toont het betrokken gedeelte van de gebruikersinterface.



Figuur 7.4 Werken met beplantingsplannen

De volgende paragrafen gaan hier nader op in.

Beplantingsplannen laden

Om een beplantingsplan te laden, selecteert u in het pull-down menu File het gewenste beplantingsplan. Vervolgens klikt u op de knop Load. Het beplantingsplan wordt ingelezen, waarbij eventuele bestaande objecten van het landschap verwijderd worden. In het tekstveld verschijnt informatie over de voortgang van het inleesproces. Na enige tijd (afhankelijk van het aantal objecten in het beplantingsplan) verschijnen de objecten op het landschap in het linkergedeelte van het scherm

Beplantingsplannen opslaan

Indien u een bepaald beplantingsplan wilt opslaan, handelt u als volgt:

1. Selecteer in het pull-down menu File het gewenste beplantingsplan.
Indien het gewenste beplantingsplan niet in de lijst voorkomt, dient u een nieuw beplantingsplan te maken.
2. Klik op de knop Save.
Het huidige beplantingsplan wordt onder de geselecteerde naam opgeslagen.

Nieuwe beplantingsplannen aanmaken

Indien u wilt werken met een nog niet bestaand beplantingsplan, dient dit eerst te worden aangemaakt. Op de volgende wijze maakt u een nieuw beplantingsplan aan:

1. Voer in het tekstveld Name de naam van het te creëren beplantingsplan in.
2. Klik op de knop New.

Het nieuwe beplantingsplan wordt aangemaakt. Het nieuwe beplantingsplan wordt meteen geselecteerd in het pull-down menu File. Dit houdt in dat eventuele inlees- of bewaaracties uitgevoerd worden op dit nieuwe beplantingsplan. Deze acties worden echter niet automatisch uitgevoerd!

7.6 Werken met objecten

Beplantingsplannen worden gevormd door objecten. Het is mogelijk het aantal en de locatie van de objecten te variëren om zo het gewenste beplantingsplan te creëren. In deze sectie wordt besproken hoe objecten te verplaatsen, toe te voegen en te verwijderen zijn. Het gedeelte **Object operations** uit de gebruikersinterface biedt de benodigde functionaliteit.

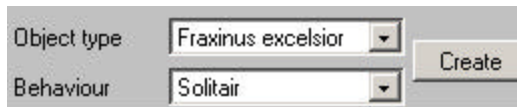
Objecten verplaatsen

Het is mogelijk de positie van objecten op het landschap te variëren. Om een object te verplaatsen dient u het object naar de gewenste positie te 'slepen'. Dit gaat als volgt:

6. Bepaal het object dat u wilt verplaatsen
7. Klik met de linkermuisknop op het object.
Dit kan op zowel de schaduw op de grond, de stam (in het geval van een boom), of de bladeren.
3. Beweeg uw muis nu met de linkerknop ingedrukt naar de gewenste positie.
Het object volgt de beweging van de muis.
8. Indien u de gewenste positie bereikt heeft, laat u de linkermuisknop los.
9. In het tekstveld verschijnen de coördinaten van de nieuwe positie van het object.

Objecten toevoegen

Om het gewenste beplantingsplan te creëren, kan het nodig zijn objecten van een bepaald type toe te voegen aan het landschap. Figuur 7.5 bevat het betrokken gedeelte van de gebruikersinterface.



Figuur 7.5 Objecten toevoegen aan het landschap

Om objecten toe te voegen handelt u als volgt:

1. Selecteer in het pull-down menu Object type het gewenste type van het toe te voegen object.
2. Indien noodzakelijk kunt u de legenda raadplegen die u linksboven in het landschapsgedeelte aantreft.
3. Selecteer in het pull-down menu Object behaviour het gewenste gedrag van het toe te voegen object.
4. Het gedrag van een object (solitair of niet-solitair) komt tot uiting wanneer een object met andere objecten overlapt tijdens zijn levensduur. Zie hoofdstuk 3 voor een gedetailleerde beschrijving.
5. Klik op de knop Create.
6. Het opschrift van de knop Create verandert in Click world ten teken dat u de plek in de wereld waar u het object geplaatst wilt hebben, kunt aanklikken.
7. Klik op de gewenste plek in het landschap.
8. Let erop dat u op een plek op het landschap klikt, dus niet daarbuiten of op een geplaatst object.

Op de aangeklikte positie verschijnt een object van het geselecteerde type. U kunt dit object nu net als ieder ander object manipuleren.

Objecten verwijderen

Het is mogelijk objecten uit het landschap te verwijderen. Figuur 7.6 toont het betrokken gedeelte van de gebruikersinterface.



Figuur 7.6 Objecten verwijderen uit het landschap

Om een object te verwijderen uit het landschap doet u het volgende:

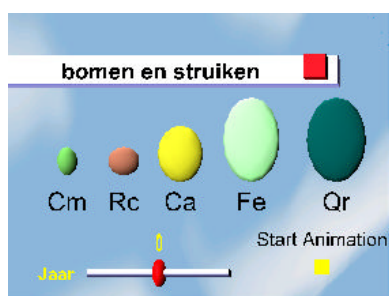
1. Klik op de knop Remove.
Het opschrift van de knop Remove verandert in Click item ten teken dat u het object kunt aanklikken dat u wilt verwijderen.
2. Klik op het te verwijderen object.
Het object wordt uit het landschap verwijderd.

7.7 Groei van objecten

De grootte van geplaatste objecten is afhankelijk van de leeftijd van die objecten. Tijdens het opstellen van een beplantingsplan heeft een architect soms niet voldoende inzicht in wat de plaatsing van bepaalde objecten voor uitwerking heeft over enkele jaren. Met Salix-2 is het mogelijk dit te bepalen; ieder object dat aan het beplantingsplan wordt toegevoegd kent een groeifunctie die variabel is met de tijd. Salix 2 stelt de architect in staat de leeftijd van de beplanting te variëren. Dit kan op twee manieren: met een animatie beginnend bij de oorsprong van het beplantingsplan tot aan het tijdstip waarop alle objecten volgroeid zijn, of door het exacte tijdstip te bepalen waarop de architect een beplantingsplan wenst te observeren.

Animatie van de groei van objecten

Figuur 7.7 toont de legenda van Salix-2:



Figuur 7.7 Legenda van boomtypen en hun groei in Salix-2

In het onderste gedeelte van de legenda ziet u in de schuifbalk Jaar het huidige jaar van het beplantingsplan. Wanneer u rechts op de tekst of gele knop Start Animation klikt, wordt het groeiproces vanaf het eerste begin tot het punt dat alle objecten volgroeid zijn, gevisualiseerd.

Het is niet mogelijk de groeianimatie tussentijds te stoppen of pauzeren, of de groeianimatie vanaf een willekeurig jaar te starten.

Beplantingsplan op een specifiek tijdstip

Indien u het beplantingsplan op een specifiek tijdstip gevisualiseerd wilt hebben, maakt u gebruik van de schuifbalk in de legenda (zie Figuur 7.7). U kunt het rode knopje op de schuifbalk slepen totdat het gewenste jaar in de gele tekst boven het knopje verschijnt.

7.8 Visualisatie van overlappende objecten

Wanneer objecten een bepaalde mate van overlap vertonen, is er sprake van overlappende objecten. Om de overlap aan de landschapsarchitect kenbaar te maken wordt deze overlap gevisualiseerd.

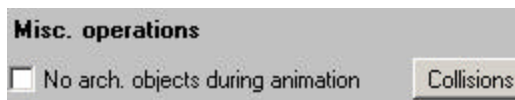
Bij de visualisatie van overlap wordt onderscheid gemaakt tussen overlap tussen niet-solitaire objecten en een overlap waarbij een solitair object betrokken is.

Figuur 7.8 toont twee momentopnames (bovenaanzicht) van het groeiproces van een aantal objecten (jaar 0 versus jaar 24). De blauwe objecten zijn niet-solitair, de witte objecten zijn solitair.



Figuur 7.8 Groei van objecten tot overlap (links: jaar 0 – rechts: jaar 24)

Figuur 7.9 bevat het gedeelte van de gebruikersinterface dat gebruikt wordt om te bepalen of er voldoende mate van overlap is:



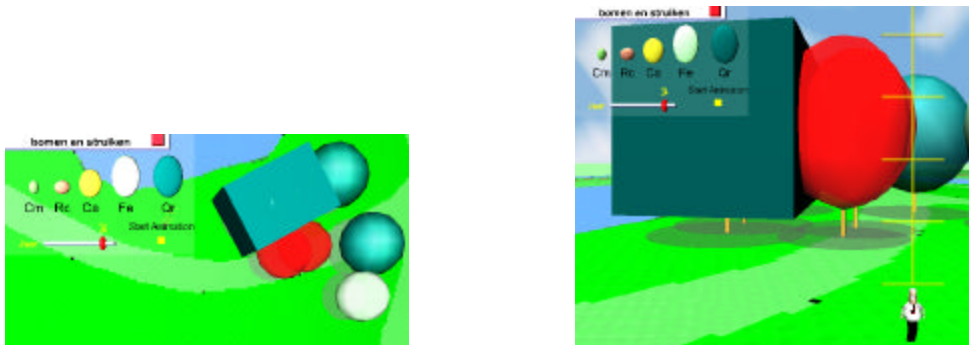
Figuur 7.9 Overlappende objecten detecteren

Wanneer de knop Collisions aangeklikt wordt, worden de posities en diameters van alle objecten gebruikt om de mate van overlap te bepalen. In het tekstveld verschijnt de output van dit proces:

```

Checking diameter and color....Done
Checking collisions...
Tree(1) collides with tree(2)!
Tree(5) collides with tree(6)!
Done checking collisions
Color the nodes....Done
    
```

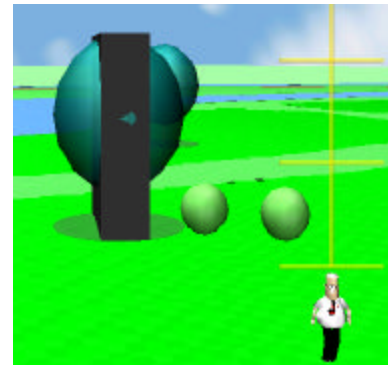
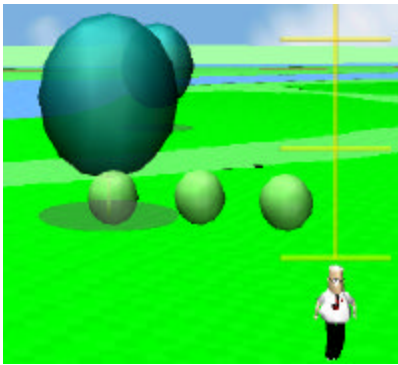
Figuur 7.10 bevat het beplantingsplan in boven- en zijaanzicht:



Figuur 7.10 Overlappende solitaire en niet-solitaire objecten

De twee blauwe objecten zijn opgegaan in een architectonisch object. De afzonderlijke boomvormen zijn vervangen door een groot blok met dezelfde kleur.

De twee witte solitaire objecten zijn roodgekleurd, ten teken dat de overlap plaats heeft gevonden tussen solitaire objecten.



Figuur 7.11 Vorming van wanden

Figuur 7.11 bevat een voorbeeld van wandvorming. Dit treedt op wanneer een niet-solitaire struik overlapt met een niet-solitaire boom. De kleur van het architectonische object is grijs.

De afmetingen van architectonische objecten volgen de volgende richtlijnen:

- De diameter van het kleinste niet-solitaire object bepaalt de breedte van het architectonische object.
- De ondergrens van het architectonische object is gelijk aan de laagste stamhoogte van de niet-solitaire objecten.
- De bovengrens van het architectonische object is gelijk aan het hoogste punt van de niet solitaire objecten.

Om de locatie van de objecten te veranderen is het wellicht handig de geplaatste architectonische objecten te verwijderen. Figuur 7.12 toont de knop Remove arch. objects in de gebruikersinterface

Remove arch. objects

Figuur 7.12 Architectonische objecten verwijderen

Daarnaast worden een architectonische object automatisch verwijderd wanneer een van de niet-solitaire objecten waaruit het is opgebouwd, wordt verwijderd.

Een andere mogelijkheid om te controleren over er overlappende objecten zijn is *tijdens het groeiproces*. Wanneer het vinkje No arch. objects during animation uitgevinkt staat, zal de controle op overlappende objecten tijdens het groeiproces uitgevoerd worden. Op deze manier kan een landschapsarchitect gemakkelijk bepalen op welk tijdstip de vorming van architectonische objecten begint.

Deze functionaliteit vergt wel rekencapaciteit en kan enigszins ten koste gaan van de performance van Salix-2. Daarom kan de gebruiker zelf aangeven of hij deze vorm van controle wil of niet.

7.9 Overige functies

VRML representatie aanwezige objecten

Het is mogelijk een VRML representatie van de in het beplantingsplan aanwezige objecten weer te geven. Hiervoor kunt u op de knop VRML nodes klikken. In het tekstveld verschijnen regels met de volgende structuur:

```
FraxExc { leeftijd 12 rdPosition -13.078682 0.0 -21.927742 }  
FraxExc { leeftijd 12 rdPosition -20.390285 0.0 -19.975468 }  
FraxExc { leeftijd 12 rdPosition -27.791458 0.0 -17.679482 }  
CorMas { leeftijd 6 rdPosition -34.37444 0.86034846 -0.9581585 }  
CorMas { leeftijd 6 rdPosition -30.969091 2.5118637 -15.340019 }  
CorMas { leeftijd 6 rdPosition -33.785885 2.5118637 -9.101123 }  
QueRob { leeftijd 12 rdPosition -25.122343 0.0 -3.6424942 }  
QueRob { leeftijd 12 rdPosition -32.625427 0.0 20.60745 }
```

Op deze manier is het mogelijk de bestaande configuratie te exporteren naar een bestand of op te nemen in een rapportage.

Wissen van tekstveld

Wanneer u op de knop Clear log klikt, wordt de inhoud van het tekstveld gewist. Deze handeling kan niet ongedaan gemaakt worden.

8 Technische beschrijving

In dit hoofdstuk wordt ingegaan op de gebruikte technieken voor de realisatie van Salix-2. In dit hoofdstuk wordt allereerst de drielagen-architectuur van Salix-2 besproken. In de overige paragrafen worden de volgende zaken besproken: de gegevenslaag, de applicatielogica en de visualisatielaag.

8.1 Drie-lagen architectuur

Salix-2 is opgezet volgens een drie-lagen architectuur: de programmatuur die interactie met de visualisatie van het landschap bewerkstelligt is gescheiden van de logica die bepaalt hoe objecten in het landschap geplaatst dienen te worden en hoe bijvoorbeeld overlappende objecten gedetecteerd worden. Vervolgens is dit weer gescheiden van de programmatuur die het opslaan en laden van beplantingsplannen realiseert.

Bovengenoemde architectuur maakt het mogelijk de visualisatie van Salix-2 te veranderen zonder dat hiervoor aanpassingen gemaakt hoeven te worden aan de wijze waarop beplantingsplannen opgeslagen worden. Daarnaast is het bijvoorbeeld mogelijk om het opslagmechanisme zoals bijvoorbeeld het type database te vervangen zonder dat dit invloed heeft op de gegevens die opgeslagen dienen te worden.

De programmatuur is ontwikkeld in de programmeertaal Java™. De visualisatie van de beplantingsplannen wordt gerealiseerd in VRML (Virtual Reality Modeling Language).

In de volgende secties wordt ingegaan op de wijze waarop ieder van bovengenoemde lagen gerealiseerd is en wordt een opsomming gegeven van de functionaliteit met bijbehorende methode-aanroepen. Verder wordt een en ander met voorbeelden uit de code verduidelijkt.

8.2 Gegevenslaag

De gegevens die Salix-2 nodig heeft voor de beplantingsplannen zijn tweeledig:

- hoeveel objecten van een bepaald type een beplantingsplan bevat,
- wat de positie is van ieder object in het landschap en
- wat het gedrag is van ieder object.

Daarnaast zijn er gegevens beschikbaar over de ondersteunde objecttypen. Hieronder vallen onder andere:

- de volledige (Latijnse) naam van het type,
- de soort van het type (boom of struik) en
- een visualisatie-gerelateerde definitie

Bovengenoemde gegevens zijn opgeslagen in een Microsoft Access database. Tabel 8.1 bevat de belangrijkste kolomnamen, hun type en een beschrijving:

Tabel 8.1 Belangrijke kolommen uit de tabel *TreeType*

Kolomnaam	Datatype	Omschrijving
Type	String	De korte benaming van het type
Kind	String	Indicatie of dit objecttype een boom of een struik is
Definition	String	De definitie van het type in termen van onder andere kleur en verhoudingen
LatinName	String	De volledige Latijnse naam van het type

Zoals eerder vermeld worden de beplantingsplannen ook in de database opgeslagen. Per beplantingsplan bestaat er een tabel in de database. De naamgeving van de tabel is gerelateerd aan de naam van het beplantingsplan; een beplantingsplan met de naam *plan* wordt in de database opgeslagen onder de naam *USER_plan*. Op deze wijze is de uniciteit van de naamgeving gewaarborgd; het aanmaken van een beplantingsplan met de naam *TreeType* levert door de prefix geen problemen op.

Tabel 8.2 bevat de kolommen uit een tabel met beplantingsplannen en hun betekenis.

Tabel 8.2 Belangrijke kolommen uit de tabel *Tree*

Kolomnaam	Datatype	Omschrijving
TreeID	Integer	Een unieke identificatie van een object in een beplantingsplan
TreeType	String	Het type van dit object
TreePosX	Float	De waarde van de x-positie van het object in het landschap
TreePosY	Float	De waarde van de y-positie van het object in het landschap
TreePosZ	Float	De waarde van de z-positie van het object in het landschap
TreeAge	Integer	De initiële leeftijd van het object

De functionaliteit om te interacteren met de database is gerealiseerd in de Java class *DBHandler*. Tabel 8.3 bevat een omschrijving van de geboden functionaliteit en de naam van de methode.

Tabel 8.3 Geboden functionaliteit door class *DBHandler*

Omschrijving	Methode in <i>DBHandler</i>
Starten van database interface	<code>void start()</code>
Stoppen van de database interface	<code>void stop()</code>
Verkrijgen van namen van opgeslagen beplantingsplannen	<code>Vector getConfigurations()</code>
Aanmaken van een nieuw leeg beplantingsplan	<code>void newConfiguration(String)</code>
Het selecteren van een beplantingsplan	<code>void setConfiguration(String)</code>
Verkrijgen van namen van ondersteunde objecttypen	<code>Vector getTypes()</code>
Laden van een beplantingsplan	<code>Vector loadObjectsFromConfiguration()</code>
Wissen van een beplantingsplan	<code>void clearConfiguration()</code>
Toevoegen van objecten aan beplantingsplan	<code>void insertInConfiguration(Vector)</code>

Met betrekking tot deze functionaliteit wordt opgemerkt:

- Bij het starten van de database interface wordt een connectie gelegd met de database. Deze connectie blijft actief totdat de database interface gestopt wordt. Dit gebeurt alleen als Salix-2 beëindigd wordt.
- Er kunnen verschillende soorten foutmeldingen optreden. Om uniformiteit af te dwingen zijn deze foutmeldingen geïmplementeerd in de class SalixException.

Code listing 1 bevat de uitwerking van de methode `newConfiguration(String)` ter illustratie van de wijze waarop de database geïmplementeerd is:

```
/**
 * Create a new configuration
 * @param name The name of the new configuration
 * @throws SalixException if the configuration could not be created or
 * if there already exists a configuration with the given name
 */

public void newConfiguration(String name) throws SalixException {
    Statement stmt = null;
    String query = 'CREATE TABLE ' + USER_TABLE_PREFIX + name +
        "(TreeID NUMBER PRIMARY KEY, TreeType STRING, " +
        "TreePosX DOUBLE, TreePosY DOUBLE, TreePosZ DOUBLE, " +
        "TreeAge NUMBER, Solitair BIT)";

    if(c == null)
        c = getConnection();
    try {
        stmt = c.createStatement();
        stmt.executeUpdate(query);
    }
    catch(SQLException e) {
        System.err.println(e.getMessage());
        if(e.getMessage().indexOf("already exists") != -1)
            throw new SalixException("newConfiguration(): Database " + name +
                " already exists");
        if(e.getMessage().indexOf("syntax error") != -1)
            throw new SalixException("newConfiguration(): Syntax error " +
                "in query or configuration name");
    }
}
```

Code listing 1 DBHandler class newConfiguration(String)

8.3 Applicatielogica

De applicatielogica bepaalt op welke wijze de opgehaalde gegevens geïnterpreteerd worden. Verder wordt hier de actieve staat van de beplantingsplannen (het aantal en de locatie van de objecten) geregistreerd. Op deze wijze kunnen bijvoorbeeld overlappings gedetecteerd worden.

De applicatielogica is in de volgende Java classes gerealiseerd:

OutputApplet

De gebruiker van Salix-2 kan het beplantingsplan manipuleren via een applet. Deze class bevat de definities van de gebruikersinterface van de applet. Verder zorgt deze class voor het starten van de database en de visualisatiecomponent. Ook is het algoritme waarmee overlap gedetecteerd wordt, in deze class geïmplementeerd.

Dit algoritme doet een paarsgewijze vergelijking: voor ieder object wordt nagegaan of hij met een van de overige objecten overlap vertoont.

Tabel 8.4 bevat een overzicht van de belangrijkste functionaliteit:

Tabel 8.4 Geboden functionaliteit door class OutputApplet

	Methode in OutputApplet
Initialiseren van gebruikersinterface	<code>void init()</code>
Events vanuit visualisatie	<code>void callback(EventOut, double, Object)</code>
Verkrijgen van namen van opgeslagen beplantingsplannen	<code>Vector getConfigurations()</code>
Aanmaken van een nieuw leeg beplantingsplan	<code>void newConfiguration(String)</code>
Het selecteren van een beplantingsplan	<code>void setConfiguration(String)</code>
Verkrijgen van namen van ondersteunde objecttypen	<code>Vector getTypes()</code>
Laden van een beplantingsplan	<code>Vector loadObjectsFromConfiguration()</code>
Wissen van een beplantingsplan	<code>void clearConfiguration()</code>
Toevoegen van objecten aan beplantingsplan	<code>void insertInConfiguration(Vector)</code>

Het volgende wordt opgemerkt met betrekking tot de functionaliteit:

- De layout van de gebruikersinterface is een GridBagLayout. Op deze wijze is het gedetailleerd mogelijk om de verschillende elementen te positioneren.
- De methode `callback` wordt automatisch aangeroepen door de standaard VRML interface die Java biedt: de zgn. External Authoring Interface (EAI). Het is mogelijk om aan te geven dat bepaalde events afkomstig uit de visualisatie afgevangen worden. Code listing 2 bevat de callback methode die de implementatie van de functionaliteit voor het event afvangt:

```
EventOutSFVec3f translation =
    (EventOutSFVec3f) result[i].getEventOut("pos_changed");
EventOutSFBool click =
    (EventOutSFBool) result[i].getEventOut("clicked");
translation.advise(applet, new Integer(nrCreatedObjects));
click.advise(applet, new Integer(nrCreatedObjects++));
```

Code listing 2 Het aanmelden voor bepaalde events (class VRMLHandler)

Bovenstaand code fragment vangt voor een object (aangeduid met result[i]) twee events op: een translation event (wanneer de positie van het object verandert) en een click event (wanneer op het object geklikt wordt). Vervolgens wordt met de advise() aanroep aangegeven dat de OutputApplet class (aangeduid met applet) de events af zal vangen. Verder wordt er nog een nieuw Integer object meegegeven, wat zorgt voor de unieke identificatie van het object in het beplantingsplan.

```
else if(who instanceof EventOutSFVec3f) {
    //A tree is _being_ moved
    //Save new position each time object is moved
    float[] pos      = vrml.getPosition(i);
    saveTreePosition(i, pos[0], pos[1], pos[2]);

    if(lastTreeMoved == i) return;
    //We already did actions below for this object
    //Remove architectural objects
    removeArchObjectsOfTree(i);
    tree.setAlert(false);
    vrml.setColor(i, tree.getAlert());
    lastTreeMoved = i;
}
```

Code listing 3 Afvangen van events door callback mechanisme

Code listing 3 bevat een fragment van de callback methode in de class OutputApplet. Het fragment implementeert de functionaliteit voor het verplaatsen van een object. Dit wordt vanuit de visualisatie geïnitieerd (getriggerd) door een EventOutSFVec3f event, afkomstig van het verslepen van een object. Aan de VRML interface wordt de positie gevraagd. Deze wordt vervolgens in de interne administratie opgeslagen (saveTreePosition). Vervolgens wordt een controle uitgevoerd of het vorige EventOutSFVec3f event betrekking had op hetzelfde object. Tijdens een sleepactie wordt dit event namelijk meerdere malen gegenereerd. De code onder de controle dient maar één keer uitgevoerd te worden: indien het versleepte object deel uitmaakt van een architectonisch object, wordt dit object verwijderd; het zal immers opnieuw berekend moeten worden aan de hand van de nieuwe coördinaten. Indien het object ook nog betrokken was bij een overlap met een solitair object, dan wordt de rode kleur van het object verwijderd.

Behaviour Orientation

Bovengenoemde classes geven een implementatie van alle mogelijke gedragingen van objecten en ondersteunde oriëntaties. Voorsnog worden alleen solitaire en niet-solitaire objecten ondersteund ter onderscheiding van de vorming van architectonische objecten of niet. De ondersteunde oriëntaties zijn op dit moment Noord, Oost, Zuid en West. Voor ondersteuning van meerder oriëntaties dient bovengenoemde class aangepast te worden.

MUTree

MUTreeType

Bovengenoemde classes representeren een object dat in het beplantingsplan geplaatst wordt, en het type van dit object.

De Java-objecten vormen directe representaties van de gegevens in de gegevenslaag. De objecten kennen voor elke in die paragraaf genoemde eigenschap *getter- en settermethodes*. Van een MUTree is het mogelijk een VRML-representatie op te vragen die zodoende gemakkelijk gevisualiseerd kan worden.

ArchitecturalObject

Bovengenoemde *class* implementeert de vorming van een architectonisch object. In de huidige implementatie wordt per tweetal niet-solitaire overlappende objecten een architectonisch object gedefinieerd.

Deze *class* is echter zo opgezet dat op dit moment al met Vectoren gewerkt wordt. Zodoende is ondersteuning voor meerdere niet-solitaire objecten per architectonisch object reeds gedeeltelijk aanwezig. De enige methode die ervan uitgaat dat een architectonisch object slechts uit twee niet-solitaire objecten bestaat, is de methode die de precieze afmetingen van het object berekent.

Tabel 8.5 toont de belangrijkste functionaliteit:

Tabel 8.5 Geboden functionaliteit door class ArchitecturalObject

Omschrijving	Methode in ArchitecturalObject
Controle of een architectonisch object een gegeven niet-solitaair object bevat	<code>boolean contains(int)</code>
Zichtbaarheid van het object opvragen/bepalen	<code>boolean getVisibility()</code> <code>void setVisibility()</code>
De kleur van het object bepalen	<code>void setColor(float[])</code>
Opvragen van VRML representatie	<code>String getVRMLRepresentation()</code>

Het volgende wordt opgemerkt met betrekking tot de functionaliteit:

- Wanneer een architectonisch object van het beplantingsplan verwijderd moet worden (tengevolge van verplaatsing van een van de niet-solitaire objecten of een kleinere omvang van deze objecten), wordt het object onzichtbaar gemaakt in plaats van totaal verwijderd. Dit omdat het herordenen van uitgegeven *identifiers* onnodig veel administratie met zich meebrengt.
- Het is mogelijk de kleur van het architectonische object te bepalen. Er wordt een *float array* met RGB waarden meegegeven. Op deze wijze is het mogelijk een architectonisch object dat uit niet-solitaire objecten bestaat van slechts één type de kleur van dat type te geven. Wanneer objecten van een verschillend type overlappen, wordt de kleur van het architectonische object grijs.

8.4 Visualisatie

De visualisatie van de beplantingsplannen geschiedt met VRML (Virtual Reality Modeling Language). VRML is met het volgende doel ontworpen [URL 11]:

“The Virtual Reality Modeling Language (VRML) is a file format for describing interactive 3D objects and worlds. VRML is designed to be used on the Internet, intranets, and local client systems. VRML is also intended to be a universal interchange format for integrated 3D graphics and multimedia. VRML may be used in a variety of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds.”

VRML is een modelleertaal, geen programmeertaal. De 3D objecten die geplaatst kunnen worden, worden **Nodes** genoemd. Via URL 12 is een lijst met alle *Nodes* en hun eigenschappen beschikbaar.

Het is mogelijk applicatielogica in een VRML omgeving te realiseren. Dit gaat met de **Script** node. Hier is het mogelijk met behulp van *ECMAScript* (een gestandaardiseerde scripttaal, lijkend op JavaScript) te interacteren met de VRML omgeving.

ECMAScript ontbeert echter de kracht van een volledige programmeertaal zoals Java. Daarom is expliciet aandacht besteed aan de koppeling van Java met VRML. Dit heeft geresulteerd in de *External Authoring Interface* (EAI). Zie hiervoor onder andere URL 13.

Salix-2 maakt gebruik van de EAI. De relatie tussen de Java-programmatuur en de VRML omgeving is gegroepeerd in de class *VRMLHandler*.

Tabel 8.6 bevat een overzicht van de belangrijkste geboden functionaliteit tezamen met de methodeaanroepen:

Tabel 8.6 Geboden functionaliteit door class *VRMLHandler*

Omschrijving	Methode in <i>VRMLHandler</i>
Starten van VRML interface	<code>void start()</code>
Stoppen van de VRML interface	<code>void stop()</code>
Toevoegen van objecten aan het beplantingsplan	<code>void createAndAddObjects(String)</code>
Toevoegen van een architectonisch object aan het beplantingsplan	<code>void createAndAddArchitecturalObject(ArchitecturalObject)</code>
Opvragen VRML-tekstrepresentatie van objecten	<code>String getText()</code>
Verwijderen van een object van een beplantingsplan	<code>void removeObjectFromScene(int)</code>
Verwijderen van een architectonisch object van een beplantingsplan	<code>void removeArchitecturalObjectFromScene(int)</code>
Het opnieuw initialiseren (leegmaken) van het beplantingsplan	<code>void resetScene()</code>
Opvragen positie van een object	<code>float[] getPosition(int)</code>
Opvragen van een klik-positie	<code>float[] getClickLocation()</code>
Opvragen diameter van een object	<code>float getDiameter(int)</code>
Opvragen hoogte van een object	<code>float getHeight(int)</code>
Opvragen stamhoogte van een object	<code>float getTrunkHeight(int)</code>
Opvragen kleur van een object	<code>float[] getColor(int)</code>
Bepalen van de kleur van een object	<code>void setColor(int, boolean)</code>
Positie in de wereld bepalen	<code>void setLocation(float[], orientation, boolean)</code>
Opvragen van huidige positie	<code>float getUserDefinedLocation()</code>

Code listing 4 op de volgende bladzijde bevat ter illustratie de implementatie van de methode

```

createAndAddArchitecturalObject(ArchitecturalObject):
    public int createAndAddArchObject(ArchitecturalObject o) {
        Node[] result = null;
        try {
            result = browser.createVrmlFromString(o.VRMLRepresentation());
        }
        catch(Exception e) {
            System.err.println(e.getMessage());
        }
        storeNodes(result, true);
        addNodesToScene(result);
        return nrCreatedArchObjects++;
    }

```

Code listing 4 Toevoegen van architectonische objecten aan de VRML wereld

Eerst wordt van het meegegeven architectonische object de VRML representatie gevraagd. Van deze representatie wordt getracht één of meerdere Nodes te maken die daadwerkelijk aan de VRML wereld kunnen worden toegevoegd (*createVrmlFromString(String*). De gecreëerde *nodes* worden in de interne administratie opgeslagen; op deze manier kunnen de objecten later vanuit Java gemanipuleerd

worden. De aanroep *addNodesToScene(Node[])* zorgt ervoor dat de *Nodes* daadwerkelijk in de VRML wereld geplaatst worden. Vervolgens wordt een unieke *identifier* voor het architectonische object bepaald en geretourneerd.

9 Conclusies en vervolg

In dit onderhavig project vormde het oorspronkelijke prototype SALIX 1 het vertrekpunt, maar is uiteindelijk een compleet nieuwe computerapplicatie ontwikkeld. Veranderingen, dan wel uitbreidingen, t.o.v. het oorspronkelijke prototype zijn:

- database beheer, dat wil zeggen dat de egbruiker nu de mogelijkheid heeft om bestaande beplantingsplannen aan te passen dan wel nieuwe beplantingsplannen te genereren via het VR-model. Alle resultaten kunnen tussentijds dan wel als zelfstandige versies worden opgeslagen. Dit biedt een groot aantal voordelen m.b.t. het ontwerp van verschillende alternatieven;
- verbeterde navigatie in het *virtual reality* -model (VR-model). Het is via Salix-2 mogelijk om snel te navigeren naar een belangrijk punt in het VR-model. Vervolgens vanuit dat punt ontwerpingrepen uit te voeren en dit te doen met behulp van alternerende perspectivische en orthogonale projecties;
- interactie met het VR-model. Alle beplantingsobjecten kunnen worden toegevoegd, verwijderd en verplaatst via interactie met het VR-model. Nadeel van de huidige versie is dat het invoeren, verwijderen en verplaatsen per beplantingsobject moet plaatsvinden.
- definitie van solitaire en niet-solitaire beplantingsobjecten. Salix-2 maakt het mogelijk om in de database de aard van het beplantingsobject te definiëren;
- transformatie van beplantingsobjecten in architectonische objecten is mogelijk. Daarbij is gebruik gemaakt van een principe waarbij de samenstelling van het architectonisch object via de kleurcode van dit object altijd bekend is. De achterliggende transformatie is gebaseerd op de berekening van een rechte lijn (euclidische afstand) tussen twee beplantingsobjecten en een omhullende kubusvorm tbv. de twee aan elkaar gerelateerde objecten. Via de combinatie van struikachtig en boomachtig beplantingsobjecten kunnen architectonisch schermobjecten (wand) worden gecreëerd. Deze berekeningswijze en presentatievorm hebben tot gevolg dat architectonische objecten langs bochten veelal een hoekig onregelmatig karakter hebben.

De doelstellingen van het Salix-2 project zijn gerealiseerd. Toch zijn er een groot aantal nieuwe vragen bijgekomen:

- directe invoer van geo-data: in de huidige versie is de geo-data voorbewerkt en omgezet naar een digitale afbeelding. In een volgende versie moet het mogelijk worden om geo-data direct in te voeren. Een belangrijk aandachtspunt daarbij vormt de transformatie van de referentiesystemen. Deze verschillen in een VRML-model sterk van een geo-data model;
- koppeling met simulatie modellen: in Salix-2 zijn de simulaties nadrukkelijk in de vrml-omgeving ingebouwd. Bij meer uitgebreide en complexe simulatiemodellen (denk hierbij bijvoorbeeld aan hydrologische modellen) is het zinvol om een bruikbare koppeling tussen vrml-model en bestaand simulatiemodel te ontwikkelen. In hoeverre hier 'open standaarden' voor aanwezig zijn, is vooralsnog onduidelijk;

- eenvoudige invoer van meerdere objecten tegelijkertijd: zoals in de conclusies al is vermeld biedt de huidige applicatie uitsluitend invoer, verwijdering en verplaatsingsmogelijkheden voor een enkel object. Het werkt in een groot aantal gevallen eenvoudiger indien meerdere objecten tegelijkertijd kunnen worden ingevoerd, verwijderd of verplaatst;
- geo-objectgebaseerde beperking in het VR-model: in veel gevallen is het weinig zinnig om een beplantingsobjecten op een bestaande weg of een bestaande waterpartij te plaatsen. De geografische objecten in het VR-model kunnen worden voorzien van een beperking (*constraint*) waardoor het niet mogelijk wordt om beplantingsobjecten op dergelijke geo-objecten te plaatsen;
- uitbreiding van beplantingsobjecten en bijbehorende simulatiemodellen;
- invoer van architectonische objecten en transformatie naar beplantingsobjecten: de 'reverse design' benadering is wellicht nog interessanter voor een architect. Eerst wordt een ruimtelijk ontwerp gemaakt met behulp van de architectonische objecten, waarna deze objecten uiteen worden gelegd in beplantingsobjecten. Deze omgekeerde methode vraagt een ander functioneel ontwerp;
- meer realisme in de boomrepresentatie: veel beschouwers van salix-2 reageren op het beperkte realisme van de grafische presentatie. Deze kan verbeterd worden via oa. een uitgebreide objectgeometrie of het gebruik van bitmaps (tabel 4.1). Een dergelijke uitbreiding heeft effect op de rendering-snelheid. Dit kan het interactief werken en het navigeren negatief beïnvloeden.

Daarnaast is het van groot belang om meer inzicht te krijgen in de gebruiksmogelijkheden van een dergelijke applicatie in het veld van de landschapsarchitectuur. Hierbij denken we in eerste instantie aan het onderwijs. Eind 2002 hebben enkele 5^{de} jaars studenten landschapsarchitectuur gebruik gemaakt van de software. Deze studenten waren unaniem enthousiast over het alternerend gebruik van verschillende beeldprojecties om op die wijze de ruimtelijke effecten van het ontwerp te kunnen bestuderen. Eveneens vonden zij dat de groeisimulatie in belangrijke mate bijdroeg tot de ontwerpqualiteit.

Wel waren er enkele praktische verzoeknummers. Men vroeg oa. om virtuele meetmiddelen (bijv. zwevende meetbalken) die in het vr-model op verzoek kunnen verschijnen en meer inzicht geven in maten en verhoudingen. Met het oog op voorbeelden werd ook gevraagd om mogelijkheden om meer informatie over beplantingsobjecten te krijgen via links in het vr-model naar bijvoorbeeld web-sites.

Een belangrijk nadeel van deze eerste onderwijs exercitie bleek de gevoeligheid van de software voor een ander *operating system*. Met name de database-link en de interfacing met een andere versie van Internet explorer bleek bij het gebruik van Windows 98, NT, 2000 of XP te verschillen.

Opvallend uit dit studentenwerk was wel dat de database-functionaliteit daadwerkelijk verbeterd is en dat ontwerpen waarin meer dan 400 objecten worden gebruikt geen probleem meer vormt. 'Rendering' blijft een aandachtspunt in deze VRML-omgeving. Wellicht dat de geo-vrml "ask-group"[url 7] daarbij tot aanbevelingen kan komen. Ondanks deze denkbare beperking lijkt VRML in

combinatie met een internet-taal een omgeving te kunnen bieden die daadwerkelijk *immersive* is en via een koppeling met agent-technologie de gebruiker op vele aspecten inzicht kan geven in zijn of haar keuzen.

In meer generieke zin lijkt het waardevol om een dergelijke programmeeromgeving ook toe te passen in andere vakgebieden waarin objecten in de tijd en ruimte kunnen veranderen. Te denken valt o.a. aan vegetatiekunde, ecologie, populatiedynamica, waterhuishouding, recreatiestudies en teeltrichtingen.

De voordelen die de uiteindelijke digitale driedimensionale werkomgeving kunnen bieden aan landschapsarchitecten lijken ons veelbelovend en wellicht dat hiermee ook een basis wordt gelegd voor de interesse vanuit andere vakgebieden.

Literatuur

[ANN1] Annevelink, E., F. Ruyten, K. Albers & E.J.J. Lamaker, 2000 Kostenvergelijking Bernhardsbos. Een vergelijking van de integrale beplantingsmethode met de blijvers-wijkers-methode. IMAG, Wageningen, IMAG-nota, V 2000-18, 73 pp.

[BUL1] Bulens, J., ea., Another view of the landscape. Digital Creativity, symposium proceedings. 2000, pp.379-386

[BUC1] Buckley, D., Ullbricht, C., Berry, J. The virtual forest: advanced 3-D visualization techniques for forest management and research in: Proceedings ESRI conference 1998, San Diego, CA

[CAR1] Card, S., Mackinlay, J. and Shneiderman, B., 1998. Information visualization. In: *Readings in Information Visualization* (Eds. Card, S., Mackinlay, J. and Shneiderman, B.), San Francisco: Morgan-Kaufmann, pp. 1-34.

[CEN1] Centen, J.W.H.M., 1998 Computer ondersteunde kostenberekening van beplantingsplannen, IMAG-nota 1998-40, Wageningen

[CHE1] Chen, C., 1999. *Information Visualisation and Virtual Environments*. Springer-Verlag, London.

[FER1] Ferrand, N., 1996. Modelling and Supporting Multi-Actor Spatial Planning using Multi-Agents Systems. NCGIA Conference on GIS and Environmental Modelling, <http://www-leibniz.imag.fr/MAGMA/ferrand/papiers/SantaFe.html>

[GAG1] Gagegan, M., 1999. Four barriers to the development of the effective exploratory visualisation tools for the geosciences, *International Journal of Geographical Information Science*, 13(4), 289-310.

[IMAG1] IMAG-DLO, 1994 Tijdnormen groenvoorziening en buitensportaccomodaties. IMAG-DLO en CNG, Wageningen, 64 pp + bijlagen.

[ISH1] Ishida, T. and Isbister, K. 2000. Digital Cities-Technologies, Experiences, and Future Perspectives. *Lecture Notes in Computer Science*, vol. 1765, Springer, Berlin.

[KEI1] Keim, D. and Kriegel, H.-P., 1996. Visualization techniques for mining large databases: a comparison. *IEEE Transactions on Knowledge and Data Engineering* (Special Issue on Data Mining).

[KLU1] Klusch, M., 2001. Information agent technology for the Internet: A Survey. *Data and Knowledge Engineering* 36, 337-372.

[LAM1] Lammeren, R. van, Annevelink, B., Kramer, H., Ruyten, F., Uiterwijk, M. and Wachowicz, M. (2001). SALIX – Simulatie Agenten voor Landschapsarchitectonisch en Virtueel Groenbeheer (x). Onderzoeksrapport 24|05|01, Wageningen Universiteit.

[LAM2] Lammeren, R. van, Clerc, V., Kramer, H., Ligtenberg, A., 2002. Virtual Reality in the Landscape design process. *In: Landscape planning in the era of globalisation. Ed.: Ogrin, D., Portoroz.*

[LEW1] Lewis, M., 1998. Designing for human interaction, *AI Magazine* 19(2).

[MAY1] Maybury, M.T. and Wahlster W. 1998. *Readings in Intelligent User Interfaces*, Morgan Kaufmann, CA.

[MUH1] Muhar, A., Pflanzen aus dem computer. *Garten + Landschaft*, 1999, 11, pp.10-13

[POR1] Porter, T., The architect's eye – visualization and depiction of space in architecture. E & FN Spon, 1997

[PPH1] Plant Publiciteit Holland, Prins Bernhard Bos, proefproject voor beplanting met economische meerwaarde. 1999, PPH, Boskoop

[REP1] Repton, H. ; Daniels, S. ; (1994), Humphrey Repton : the red books for Brandsbury and Glemham Hall, Washington : Dumbarton Oaks Research Library and Collection, [72] p.

[RUY1] Ruyten, F., J. Centen & D. Goense, 1999 De Integrale beplantingsmethode en de blijvers- en wijkersmethode: een kostenvergelijking. Groen, juli/augustus, 42-47.

[RUY2] Ruyten, F. Architectonische beplantingstypen en elementen met het bijbehorende onderhoud. Poster, 2000.

[RUY3] Ruyten, F., E. Annevelink, K. Albers & E.J.J. Lamaker, 2000 De integrale beplantingsmethode: Een stap verder. Groen, September, 43-47.

[SIM1] Simonds, J.O., (1997) Landscape Architecture, 3rd edition, McGraw-Hill, Chapter 11: Site volumes

[WAC1] Wachowicz, M. (2001). GeoInsight: an approach for developing a knowledge construction process based on the integration of GVis and KDD methods. To appear in: *Geographic knowledge discovery and spatial data mining* (Eds. Miller, H. J. and Han, J.), London: Taylor & Francis.

[WAC2] Wachowicz , M., Bulens, J.D., Kramer, H., Lammeren, R.J.A., Ligtenberg, A., Rip, F. (2002) GeoVR construction and use: the seven factors. In: proceedings of the 5th Agile conference on geographic information science by Ruiz, M., Gould, M., Ramon, ., Palma, pp.417-422

[WAT1] Waters, N. 1997. Is GIS prepared for the virtual reality age? *GISWord*, March 1997, p.58.

[WEI1] Weiss, G., 1999. Multi-Agents Systems. The MIT Press, Cambridge, USA

Internetadressen

(20 november 2002)

- [URL1] <http://www.cirad.fr/produits/amap/pepinieres/categori.htm>: Groei-gegevens
- [URL2] <http://www.cirad.fr/presentation/en/program-eng/amap.shtml> CIRAD
- [URL3] http://www.cryo-networks.com/uk/frame_center.htm virtual communities
- [URL4] <http://www.3dcafe.com/asp/plants.asp> voorbeelden van 3D-modellen
- [URL5] <http://lms.cfr.washington.edu/lmsdown.html> groeisimulatie in de bosbouw
- [URL6] http://www.bbc.co.uk/thegarden/html/logon_page.shtml BBC VRML
- [URL7] <http://www.ai.sri.com/geovrml> GeoVRML group
- [URL8] <http://www.w3.org/TR/REC-smil> SMIL-site
- [URL9] <http://www.best.com/~rikk/Book> VRML 97
- [URL10] <http://www.web3d.org>
- [URL11] <http://www.web3d.org/technicalinfo/specifications/vrml97/part1/introduction.html>
- [URL12] <http://www.web3d.org/technicalinfo/specifications/vrml97/part1/nodesRef.html>
- [URL13] <http://www.vrml.org/WorkingGroups/vrml-eai/Specification/>

Verwijzingen in de tekst naar:

Literatuur [auteur1]

URL [URL1]

Bijlage 1 Installatie

Systemvereisten

Salix-2 is ontwikkeld voor gebruik onder de volgende systeemconfiguratie:

- Microsoft Windows 2000 Professional
- Salix 2 is tevens succesvol getest met het besturingssysteem Windows '98.
- Microsoft Internet Explorer versie 5.0 en hoger
- Cortona VRML Client versie 3.1
- Java Development Kit 1.3.1

Installatieprocedure

In deze sectie wordt beschreven welke handelingen uitgevoerd dienen te worden om Salix 2 te gebruiken. Bij deze beschrijving wordt uitgegaan van het besturingssysteem Windows 2000 Professional.

1. Uitpakken van de bestanden naar de harde schijf

De Salix applicatie wordt gedistribueerd in een ZIP-bestand. ZIP-bestanden zijn te openen met het programma WinZip. Een evaluatieversie van WinZip is gratis verkrijgbaar via www.winzip.com. Deze zip file kan uitgepakt worden naar een willekeurige directory.

Voor het uitpakken van de bestanden naar de harde schijf verwijzen wij naar de handleiding of het help-bestand van WinZip.

2. Aanmaken ODBC Data Source

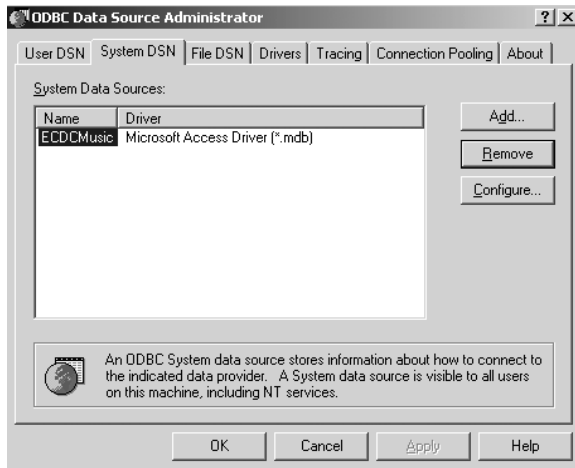
De gegevens over beplantingsplannen betreft Salix 2 uit een Microsoft Access database. Deze database wordt meegeleverd in het ZIP-bestand (Tree.mdb). Om het systeem de database te laten benaderen, dient een aanpassing aan het Windows 2000 systeem gemaakt te worden: de database moet namelijk benaderbaar gemaakt worden voor andere applicaties. De uit te voeren handelingen worden hieronder beschreven:

N.B. Voor deze aanpassingen is het vereist dat u als 'Administrator' op uw computer bent ingelogd. Vraag indien nodig uw systeembeheerder de installatieprocedure uit te voeren.

1. Ga naar Settings, Control Panel, Administrative Tools, Data Sources (ODBC).

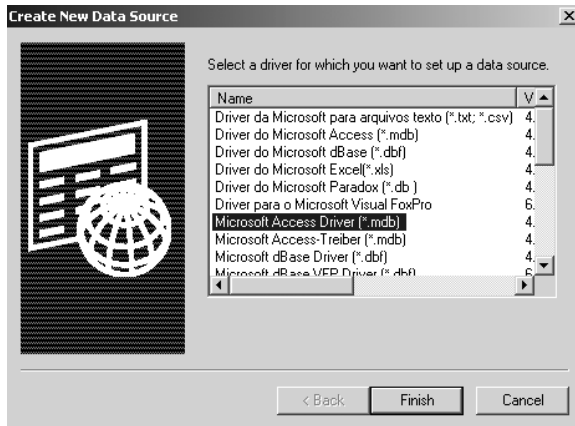
Het scherm *ODBC Data Source Administrator* verschijnt.

2. Ga naar het tabblad System DSN:



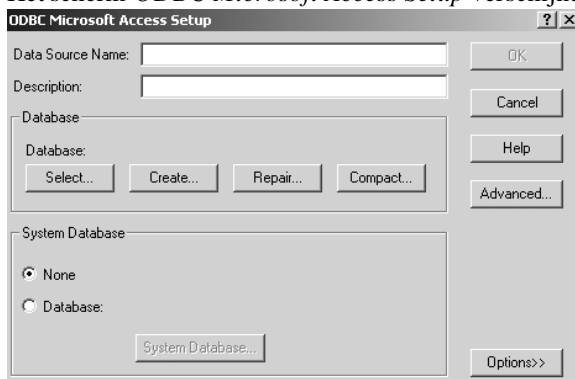
3. Klik op de knop Add...

Het scherm *Create New Data Source* verschijnt:



4. Selecteer 'Microsoft Access Driver (*.mdb)' en klik op de knop Finish.

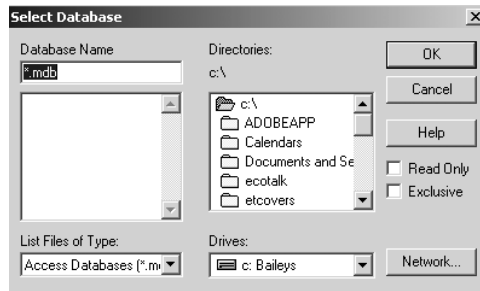
Het scherm *ODBC Microsoft Access Setup* verschijnt:



5. Voer bij Data Source Name de tekst 'myTrees' (let hierbij op het gebruik van hoofdletters en kleine letters).

6. Klik in de sectie Database op de knop Select...

Het scherm *Select Database* verschijnt:



7. Selecteer nu de directory waar het bestand Tree.mdb is geplaatst (zie **uitpakken van de bestanden naar de harde schijf**), en klik op de knop OK.
8. Klik in het scherm *ODBC Microsoft Access Setup* op de knop OK.
9. Klik in het scherm *ODBC Data Source Administrator* op de knop OK

Salix 2 is nu klaar voor gebruik.

Bijlage 2 Overzicht van ‘multi-agents systems’ concepten (uit LAMI)

Concept	Definition	Main Characteristics
AGENTS	They are interacting computing elements (e.g. an active object, or a bounded process, or a programming language), which encapsulate <i>Design goals</i> (what objectives we want to achieve), <i>Behaviour</i> (perception, actions, intentions, and tasks carried out by an agent), <i>Agent State</i> (a situation, position, condition of an agent at a specific point in time), <i>Process</i> (such as problem solving, planning, search, decision-making, and learning).	<i>Autonomous</i> (agents to some extent they have control over their behaviour and can perform their tasks without the intervention of humans or other systems) <i>Intelligent</i> (agents pursue their goals and execute their tasks such that they optimise some given performance measures) <i>Reactive</i> (agents are capable to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design goals) <i>Pro-Active</i> (agents take the initiative in order to satisfy their design goals)
INTERACTIONS	Everything that occurs between agents (<i>agent-agent interaction</i>), between agents and their environment (<i>agent-environment interaction</i>) and between agents and humans (<i>agent-human interaction</i>).	<i>Frequency</i> (low ...high) <i>Persistence</i> (short-term ... long-term) <i>Level</i> (signal-passing ... knowledge intensive) <i>Pattern</i> (flow of data and control) <i>Variability</i> (fixed ... changed) <i>Purpose</i> (competitive ... co-operative)
AGENTS ARCHITECTURE	A particularly methodology for building agents (e.g. structures, algorithms, and control flows)	<i>Logical-based architectures</i> (in which a process is achieved via logical deduction) <i>Reactive architectures</i> (in which a process is achieved via simple mapping from perception to action) <i>Belief-desire-intention architectures</i> (in which a process is the explicit representation of the beliefs, desires, and intentions that determine the agent's actions) <i>Layered architectures</i> (in which a process is realised via the interaction of tasks accomplishing layers)
ENVIRONMENT	communication and interaction protocols; <i>Static environments</i> (that can only change via the action of the agent in it) <i>Deterministic environments</i> (in which there is no uncertainty about the effect an action will have. Few environments are deterministic). <i>Discrete Environments</i> (in which perceptions and actions are discrete, as opposed to continuous) <i>Episodic environments</i> (in which tasks are divided into a number of discrete episodes. They relieve the agent of the need to reason about the interaction between current and future behaviour)	<i>Knowable</i> (to what extent is the environment known to the agent) <i>Predictable</i> (to what extent can it be predicted by the agent) <i>Controllable</i> (to what extent can the agent modify the environment) <i>Historical</i> (Do future states depend on the entire history, or only current state?) <i>Teleological</i> (Are there other agents?) <i>Real-time</i> (Can the environment change while the agent is deliberating?)