



## **ONTWIKKELING VAN EEN VISUEEL VOCABULAIRE VOOR MODELPRIMITIEVEN EN OPERATIES IN GARP3**

**DIRK BERTELS**

Studentnummer 9350616

Augustus 2007

**DOCTORAAL SCRIPTIE EN STAGEVERSLAG  
SOCIAAL WETENSCHAPPELIJKE INFORMATICA**

Begeleiding: **BERT BREDEWEG**

Onderwijsinstituut Informatiewetenschappen  
Human-Computer Studies Laboratory  
Faculteit der Natuurwetenschappen, Wiskunde en Informatica  
Universiteit van Amsterdam  
Matrix I, Kruislaan 419, 1098 VA Amsterdam



## **Samenvatting**

Qualitative Reasoning (QR) is een formele taal voor het beschrijven van onderdelen van situaties in de wereld op een kwalitatieve manier. Deze taal biedt de mogelijkheid kennis over systemen in de wereld op te slaan in een artefact, zoals de Garp software. Een tweetal systemen zijn in het verleden ontwikkeld waarin een meer grafische representatie wordt gehanteerd voor de gebruikersinteractie met het artefact: VisiGarp voor simulatie en inspectie van kwalitatieve modellen en Homer als modelbouw omgeving. Recente ontwikkeling van Garp, Homer en VisiGarp is de integratie van deze drie systemen in het kader van het NaturNet-Redime project tot een enkel raamwerk: Garp3.

Dit werkstuk beschrijft de visuele taal die ontwikkeld is voor het Garp3 raamwerk (een gemeenschappelijk visueel vocabulaire). Het bestaat uit een vertaling van het QR vocabulaire naar een geïntegreerde taal van zowel tekst als beeld. Naast dit visuele vocabulaire worden ook de operaties op de QR ingrediënten weergegeven in dit zelfde vocabulaire om bij te dragen aan de aansluiting van de besturing van de software op de model ingrediënten.

## **Dankwoord**

Mijn dank gaat uit naar Bert Bredeweg voor zijn doorzettingsvermogen in het begeleiden van mijn werkstuk/stage en het aanwakkeren van het vertrouwen in mijn wetenschappelijke kunnen. Zijn onderkoelde maar duidelijke manier van motiveren zal mij nog lang bij blijven. Verder wil ik mijn vrienden Mark-Jan Sizoo en Marc Dahmen bedanken voor de goede gesprekken over het onderwerp en vooral de motivatie voor dit werkstuk. Bovenal wil ik mijn geliefde Godelinde Andrea bedanken voor de steun en het geloof. Zij heeft vele keren de kinderen opgevangen, mijn leven draaiend gehouden en zo af en toe mijn hoofd weer in mijn romp geduwd. Dit alles in het vertrouwen dat het haar lief helpt afstuderen.



## Inhoudsopgave

<b>1</b>	<b>INLEIDING</b>	<b>6</b>
1.1	Representeren	6
1.2	Variatie in artefacten	6
1.3	Taken uitvoeren	7
1.4	Kwalitatief redeneren	7
1.5	Kwalitatieve simulaties	8
1.6	Visualisatie van QR-primitieven en operaties voor Garp3	9
<b>2</b>	<b>GARP</b>	<b>9</b>
2.1	Ontologie	9
2.2	Gedrag	10
2.3	Causaliteit	10
2.4	Modelfragmenten	10
2.5	Modelingrediënten	11
2.5.1	Entiteiten, attributen en structurele relaties	11
2.5.2	Kwantiteiten, waardenruimte, waarde en afgeleide	11
2.5.3	Afhankelijkheden tussen kwantiteiten	11
2.6	Model constructies	12
<b>3</b>	<b>GARP3</b>	<b>12</b>
3.1	Homer	13
3.2	VisiGarp	14
3.3	Onderweg naar Garp3	16
<b>4</b>	<b>GRAFISCHE PRINCIPES</b>	<b>16</b>
4.1	Iconen en grafische representatie	17
4.2	Samengesteld visualiseren	19
4.3	Kleur	21
4.4	Data-inkt	21
4.5	Operaties	22
<b>5</b>	<b>GRAFISCHE REPRESENTATIE GARP3</b>	<b>25</b>
5.1	Visualisatie modelingrediënten	26
5.1.1	Model type, klasse en instantie	28
5.1.2	Elementaire grafische objecten	28
5.2	Relaties	36
5.2.1	Relaties voor gedrag	37
5.2.2	Structurele relaties	39
5.2.3	Overige relaties	39
5.3	Afgevallen alternatieven	40
5.4	Visualisatie operaties	42

5.5	Operaties op het model	43
5.6	Operaties op modelingrediënten	44
5.7	Operaties op model views	45
5.8	Operaties voor simulatie	46
5.9	Overige operaties	48
<b>6</b>	<b>IMPLEMENTATIE</b>	<b>48</b>
6.1	Hoofdscherm	48
6.2	Modelbouw	50
6.2.1	Window layout van lijst van modelingrediënten	51
6.2.2	Window layout van lijst en definitie	51
6.2.3	Hiërarchische window layouts	54
6.2.4	Vrije layout van modelingrediënten	55
6.3	Modelsimulatie	56
6.4	Alternatieve layouts	59
6.5	Grafisch vocabulaire van Garp3	62
<b>7</b>	<b>PILOT EVALUATIE VAN OPERATIES</b>	<b>62</b>
7.1	‘Slechte pictogrammen’	67
<b>8</b>	<b>CONCLUSIE EN DISCUSSIE</b>	<b>67</b>
<b>9</b>	<b>LITERATUUR</b>	<b>70</b>
<b>10</b>	<b>BIJLAGEN</b>	<b>71</b>
10.1	Bijlage 1: Vragenlijst	71
10.2	Bijlage 2: Statistische analyse	82

## 1 Inleiding

In toenemende mate spelen pictogrammen een rol in het leven van mensen. Veel van deze pictogrammen, zoals op verkeersborden, mobiele telefoons en computerschermen, merken we amper meer op vanwege de grote bekendheid die we hebben met het gebruikte visuele vocabulaire. Komen we echter in contact met een nieuw vocabulaire dan worden we bewust van de interpretaties die we maken en de betekenis die we geven aan pictogrammen. In het buitenland zijn we bijvoorbeeld een stuk minder trefzeker in het verkeer, en de besturing van een nieuw computerprogramma gaat zelden soepel. Dit werkstuk is de beschrijving van een nieuw vocabulaire waarin in eerste instantie uiteen wordt gezet *wat* weergegeven gaat worden in het visuele vocabulaire. Vervolgens worden grafische principes besproken die ten grondslag liggen aan de visuele vertaling, ofwel *hoe* de vertaling gebeurt. Ten derde wordt de invulling van het vocabulaire gepresenteerd. Tenslotte volgt een evaluatie van de bruikbaarheid van het nieuwe vocabulaire.

### 1.1 Representeren

Mensen gebruiken geluiden, gebaren en symbolen om te refereren aan objecten, dingen en concepten. Hierdoor maken we gebruik van representaties en niet meer van de dingen zelf. Elke representatie heeft twee ingrediënten. Ten eerste het gerepresenteerde en ten tweede de representatie. De elementen uit de representatie 'staan voor' elementen uit het gerepresenteerde. De kracht van menselijke cognitie bestaat uit het vermogen om situaties, waarnemingen en gedachten te representeren in een ander medium dan waarin ze zijn ontstaan. Afhankelijk van de taak hanteren we een representatie waarin de irrelevante details naar de achtergrond worden geabstraheerd zodat alleen overblijft wat belangrijk is. Vervolgens is het mogelijk dat uit deze representatie nieuwe ervaringen, inzichten of creaties ontstaan. Dit is de essentie van intelligentie volgens Norman (1993).

Externe representaties bestaan buiten degene die de representatie gebruikt en zijn daartoe vastgelegd in een bepaald mechanisme, een cognitief artefact. Dit kan een simpel artefact zijn als een stokje en strandzand voor tekeningen waarbij de representatie ter plekke eventueel kan worden toegelicht door de maker. Het artefact kan ook veel geavanceerder zijn en vele representaties bevatten die met een druk op de knop zijn op te roepen en welke zijn te delen met anderen zoals de huidige digitale en interactieve artefacten. De objecten die we gebruiken in de wereld om ons heen zijn niet allemaal artefacten voorzien van een representatie van informatie. Veel objecten zijn simpelweg extensies van onze fysieke mogelijkheden zoals een vergrootglas of schaar.

### 1.2 Variatie in artefacten

Ten eerste kunnen artefacten van elkaar verschillen in dat wat gerepresenteerd is. Cognitieve artefacten kunnen zelf onderdeel zijn van een representatie, ofwel een representatie zijn van representaties. Vanuit deze metarepresentaties zijn nieuwe inzichten en verbanden af te leiden en is nieuwe kennis op te doen. Dit type artefact verschilt van artefacten waarbij het de representatie van een fysiek object betreft. De benzinemeter in een auto is hier een voorbeeld van. De meter is een artefact met een grafische representatie van 'leeg' tot 'vol' als representatie van de benzinetank van de auto zelf.

Ten tweede kunnen artefacten verschillen, niet in wat gerepresenteerd wordt maar hoe dit gebeurt, ofwel verschillen in de mate waarin ze analoog zijn aan dat wat ze representeren. Het betreft het verschil tussen

representaties waarbij een duidelijke overeenkomst bestaat tussen de structuur van de syntax van de representatie en de semantiek van het gerepresenteerde en representaties waarbij dat niet het geval is (Kulpa, 1994). Een analoge representatie is te bezien als een voorbeeld of illustratie van het gerepresenteerde terwijl het tegenovergestelde, de propositionele representatie een beschrijving is van het gerepresenteerde. Dit heeft gevolgen voor de manier waarop informatie ontleend wordt aan de representatie. In het eerste geval kan deze vaak simpelweg worden geobserveerd of gemeten terwijl deze in het tweede geval moet worden afgeleid. Er is een cognitieve vertaalslag nodig om te begrijpen wat de representatie behelst.

### **1.3 Taken uitvoeren**

Ondanks het feit dat er verschillende representaties mogelijk zijn in de uitvoer van een taak is de geschiktheid ervan relatief. De benzinemeter kan bestaan uit een enkel getal dat aangeeft hoeveel liter benzine er nog in de auto zit. Om te bepalen hoe vol de tank is, moet eerst uit het geheugen worden gehaald hoeveel liter er in de tank past en vervolgens het verschil worden berekend van deze twee getallen. De kracht van een cognitief artefact wordt bepaald door de geschiktheid van de representatie gehanteerd in het artefact voor de taak die met het artefact moet worden uitgevoerd (Norman, 1993). Dezelfde informatie kan op verschillende manieren gerepresenteerd worden voor de uitvoer van verschillende taken.

Elk cognitief artefact is een instrument dat mensen gebruiken voor de uitvoer van een bepaalde taak. Om deze taak uit te kunnen voeren, interacteren mensen met het artefact. Deze interactie vindt plaats in een bepaalde modaliteit. Een modaliteit bestaat uit het sensorische systeem waardoor we activiteit waarnemen of waarneembaar maken. Bijvoorbeeld spraak, klanken en muziek nemen mensen waar door hun auditieve modaliteit en tekst en beeld door de visuele modaliteit. Binnen elke modaliteit bestaan één of meer kanalen van communicatie; elk kanaal bevat een manier om informatie in een modaliteit te encoderen, zoals deze tekst een voorbeeld is van een Nederlandstalig tekstkanaal in de visuele modaliteit. Daarnaast moet het mogelijk zijn het communicatiekanaal te decoderen, dit vindt nu bij u plaats terwijl u deze tekst leest.

Een artefact dat bestaat uit een set communicatiekanalen waarbij sprake is van onderlinge coördinatie is een zogenaamd medium (Elsom-Cook, 2001). Televisie maakt bijvoorbeeld gebruik van bewegend beeld, geluid, spraak en ondertitels. Deze vormen samen één geheel en zijn onderdeel van de interpretatie taal van televisie. De ondertitels hebben veelal betrekking op de vreemde taal die gesproken wordt in geluid en behoort toe aan de mensen wiens mond op dat moment bewegen.

Alle informatie die mensen encoderen in een communicatiekanaal is een representatie van wat zij willen overbrengen. Deze representatie wordt vervolgens gedecodeerd door de ontvanger, hopelijk tot de informatie die bedoeld was te worden overgebracht. De ontwerper van de eerder genoemde benzinemeter maakt een representatie van de benzinetank en hoopt dat later de bestuurder deze representatie weer kan vertalen naar de benzinetank. We gebruiken een taal voor de representatie in de artefacten. Deze taal kan variëren van formeel met een vast lexicon en syntax tot grotendeels ongedefinieerd, zoals vaak bij veel visuele talen.

### **1.4 Kwalitatief redeneren**

Een van de belangrijkste eigenschappen van mensen is de capaciteit te redeneren over de fysieke wereld. Dit stelt ons in staat problemen op te lossen, problemen te voorzien voordat ze ontstaan, gereedschappen



te ontwikkelen, etc. Dit ‘common sense’ redeneren is in het dagelijkse leven veelal kwalitatief, aangezien de mathematische grondslagen niet beschikbaar of geheel irrelevant zijn voor het redeneerproces. De observatie dat wetenschappelijke, mathematische en technische modellen en formalisaties maar een deel vormen van de intellectuele activiteiten van mensen in hun omgang met de wereld, heeft aanleiding gegeven tot onderzoek naar kwalitatief redeneren (Bredeweg en Struss, 2003). Qualitative Reasoning (QR) is de verzamelterm voor onderzoek naar de manier waarop kwalitatieve representaties en redeneren mogelijk zijn ten behoeve van kennisoverdracht in het algemeen en onderwijs in het bijzonder (Bredeweg en Forbus, 2003).

QR is een formele taal voor het op kwalitatieve wijze beschrijven van onderdelen van situaties in de wereld. Daarnaast is het een beschrijving van hoe te redeneren over de elementen in die taal om tot bepaalde conclusies te komen. Deze beschrijvingen zijn expliciete representaties aan de hand van een ontologie. De meest voorkomende typen ontologie zijn ofwel gebaseerd op onderling verbonden componenten van een systeem, of gebaseerd op interacterende processen binnen het systeem (Bredeweg en Struss, 2003). De representaties in QR zijn in de vorm van een kwalitatief model bestaande uit relevante objecten met bepaalde eigenschappen en interacties tussen deze objecten en kunnen worden gebruikt in simulaties. De gehanteerde modellen zijn kwalitatief van aard waarbij twee dingen van belang zijn (Forbus, 1996): Ten eerste de resolutie of ‘qualitativeness’. Het model moet de nauwkeurigheid hebben om een intuïtief resultaat op te leveren om een bepaald probleem op te lossen of taak uit te voeren. Ten tweede ‘compositionality’, het combineren van representaties van verschillende onderdelen of aspecten van een fenomeen tot een representatie van het fenomeen als geheel.

Voorbeelden van artefacten waarbij QR wordt ingezet als representatie taal zijn onder andere: Betty’s Brain (Biswas e.a., 2001) waarbij leerlingen leren door ‘concept maps’ te onderwijzen aan een systeem en daar intuïtieve feedback op krijgen; Vmodel (Forbus, 2001) een visuele kwalitatieve modelleeromgeving waarin studenten worden uitgedaagd zoveel mogelijk herbruikbaarheid tussen onderdelen (o.a. processen) van modellen te ontdekken; en Garp (Bredeweg, 1992), een kwalitatieve simulatie omgeving voor het vastleggen van conceptuele domeinkennis van experts.

## **1.5 Kwalitatieve simulaties**

Een onderdeel van QR onderzoek is kwalitatieve simulatie waarin het gedrag van systemen, aan de hand van een model, wordt gesimuleerd en kan worden verklaard in kwalitatieve termen. Het formuleren van een kwalitatief model is waardevol in een onderwijsomgeving. Ten eerste omdat het domein beschreven kan worden in conceptuele termen met een natuurlijke notie van causaliteit en ten tweede omdat het formuleren van een kwalitatief model een goed raamwerk biedt voor kwantitatieve en mathematische modellen van hetzelfde domein. Daarnaast is het gebruik van een kwalitatief model in een simulatie context zinvol in domeinen waarvan geen kwantitatieve gegevens of modellen bekend zijn en voor het articuleren van kennis om inzicht te krijgen in het gedrag van een bepaald systeem. Het formuleren, testen en reviseren van domeinkennis in kwalitatieve modellen is een manier om kennis te articuleren en een goed conceptueel begrip te krijgen van het gedrag van het gemodelleerde systeem. Garp is gebaseerd op een ontologie waarin de component-gebaseerde en de proces-gebaseerde aanpak zijn gecombineerd. Aan de hand van modelfragmenten worden uit een initieel scenario meerdere toestanden gegenereerd met een alternatief verloop. Deze toestanden geven causale verklaringen voor het gedrag van het gesimuleerde systeem.

Ondanks de toepasbaarheid van QR in verschillende domeinen en de low-resolution ambitie zijn de modellen lastig 'leesbaar'. Het zijn reflectieve of propositionele representaties omdat de gehanteerde QR-taal domein onafhankelijk is. De taal kan geen eenduidige visuele relatie vertonen met dat wat gerepresenteerd wordt, omdat het voor diverse domeinen kan worden gebruikt. Het gevolg is dat meer mentale inzet nodig is om het model te begrijpen. Een tweetal systemen waarin een meer grafische representatie wordt gehanteerd: VisiGarp (Bouwer, 2005) voor simulatie en inspectie van kwalitatieve modellen en Homer (Jellema, 2000) als modelbouw omgeving. In beide systemen worden in veel gevallen de QR ingrediënten in meerdere kanalen, zowel tekst als beeld, weergegeven.

Recente ontwikkeling van Garp, Homer en VisiGarp is de integratie van deze drie systemen in het kader van het NaturNet-Redime project (zie [www.naturnet.org](http://www.naturnet.org)) tot één enkel raamwerk: Garp3. Dit nieuwe systeem wordt ingezet in het toepassingsdomein van duurzame ontwikkeling waarin mensen zonder expliciete kennis van QR, onder andere ecologen, te maken krijgen met het vocabulaire en het maken, testen en reviseren van modellen.

## **1.6 Visualisatie van QR-primitieven en operaties voor Garp3**

De integratie van de drie QR systemen, Garp, VisiGarp en Homer tot Garp3 is behalve een technische implementatie ook een ontwikkeling van een gemeenschappelijk visueel vocabulaire voor de drie onderdelen. Dit werkstuk beschrijft de visuele taal die ontwikkeld is voor het Garp3 raamwerk, waarbij wordt uitgegaan van de bestaande systemen, en deze zoveel mogelijk stroomlijnt om tot een naadloos systeem te komen. Deze visuele taal bevat visuele woorden in de vorm van pictogrammen en beoogt makkelijker 'leesbaar' te zijn, ofwel meer zeggen met minder woorden. De gevolgen van deze leesbaarheid bestaan uit een snel begrip van het abstracte QR vocabulaire en de mogelijkheid dit vocabulaire te onthouden tijdens het werken met de software en in herhaaldelijk gebruik.

Tevens is dit werkstuk de beschrijving van de ontwikkeling van een medium. De twee communicatie kanalen die gehanteerd worden in de twee modules, voor modelbouw en modelsimulatie, worden zoveel mogelijk geïntegreerd. Deze twee kanalen zijn onlosmakelijk met elkaar verbonden waardoor een 'cross channel' taal van interpretatie ontstaat waarin het geheel enkel te begrijpen is door de interpretatie van gezamenlijke kanalen. De domeinspecifieke elementen blijven gecodeerd in het bestaande tekstkanaal terwijl de QR taal in het nieuwe grafische kanaal wordt weergegeven. Naast dit visuele vocabulaire worden de operaties op de QR ingrediënten weergegeven in dit zelfde vocabulaire om bij te dragen aan de naadloze aansluiting van de onderdelen van Garp3.

## **2 Garp**

Garp (Bredeweg, 1992) is een interactieve simulatie omgeving gebaseerd op QR voor het formuleren en testen van kwalitatieve modellen. Een kwalitatieve simulatie is een beschrijving van de beredeneerde gevolgen vanuit een uitgangssituatie. Aan de hand van een scenario, ofwel initiële structurele situatiebeschrijving is het in staat gedrag af te leiden en vervolgt toestanden te genereren. De beschrijving van dit gedrag is in kwalitatieve termen. De waarden van grootheden zijn geen getallen maar bestaan uit een geordende verzameling van labels met elk een specifieke conceptuele betekenis.

### **2.1 Ontologie**

De kwalitatieve modellen zijn representaties van systemen. Een belangrijk onderdeel is een beschrijving van elementen die voorkomen in dat systeem en de onderlinge relaties, samengevat in de vorm van een

ontologie. Een ontologie is als een zienswijze op het systeem dat gerepresenteerd wordt; van de complexe werkelijkheid wordt maar een deel opgenomen in het kwalitatieve model. De ontologie bepaalt welke elementen uit het systeem relevant zijn en de mate van detail van het gerepresenteerde in de representatie.

## **2.2 Gedrag**

Van alle relevante onderdelen van het gerepresenteerde systeem is al het relevante gedrag weergegeven in het model. Van elk onderdeel is afzonderlijk het gedrag beschreven en opgenomen in het model. Afhankelijk van de structurele relaties tussen deze onderdelen kan een onderdeel verandering veroorzaken in een ander onderdeel. Het totale gedrag van het systeem bestaat uit het gedrag van de onderling verbonden onderdelen. Uit de structurele samenstellingen van verschillende onderdelen tot een compleet systeem wordt het totale gedrag van het systeem als geheel afgeleid in de vorm van een toestand. Garp leidt alle toestanden af uit het scenario en blijft dit doen met elke toestand tot er geen verandering meer is in gedrag. Elke kwalitatieve toestand kan een abstracte representatie zijn van oneindig veel kwantitatieve toestanden (Forbus, 1996). In kwalitatieve termen bestaat een toestand totdat er een significante verandering is. In kwantitatieve simulaties is dit anders, hierin is sprake van continue verandering en niet van duidelijke toestanden.

## **2.3 Causaliteit**

Een belangrijke eigenschap van QR is het representeren van causale verbanden: A is van invloed op B en B is weer van invloed op C. Als men B wil veranderen, kan dit door A te veranderen en als een verandering in B wordt waargenomen, is een verandering in A daar waarschijnlijk de oorzaak van. Deze causale noties zijn relaties in één enkele richting ofwel relaties in de vorm van oorzaak en gevolg. Het beschrijven van gedrag in termen van oorzaak-gevolg relaties is dat het sterk overeenkomt met de natuurlijke manier waarop mensen redeneren over systemen. De verschillende toestanden samen vormen de 'state-graph' van het gesimuleerde systeem waarbij de afzonderlijke toestanden de zogenaamde 'causal maps' van het gedrag zijn.

## **2.4 Modelfragmenten**

De kwalitatieve modellen in Garp bestaan uit gedeeltelijke beschrijvingen van onderdelen van het gerepresenteerde systeem. In deze beschrijvingen wordt zowel de structuur als het gedrag van het onderdeel beschreven. Meerdere deelbeschrijvingen kunnen delen van het gedrag beschrijven en samen vormen ze de gehele gedragsbeschrijving. Deze modelfragmenten zijn van een drietal typen (Bredeweg, 1992): Ten eerste de statische modelfragmenten waarin eigenschappen van één element of een configuratie van elementen worden gedefinieerd. Ten tweede de proces-modelfragmenten waarin veranderingen in het gedrag van elementen als gevolg van een ongelijkheid wordt beschreven. Een proces wordt geactiveerd door een ongelijkheidsrelatie tussen twee kwantiteiten. Tenslotte bestaan er agent-modelfragmenten waarin elementen worden gedefinieerd die een verandering bij een element in het systeem veroorzaken alsof ze van buiten het systeem komen.

Simulaties van gedrag van een systeem in Garp worden uitgevoerd aan de hand van een scenario van een toestand van dat systeem. De modelfragmenten bestaan uit twee delen, ten eerste het conditionele deel en ten tweede het resultaat. Afhankelijk van de match tussen het conditionele deel van het

modelfragment en de huidige toestand bepaald Garp of het modelfragment toegepast kan worden. Vervolgens worden daarbij instanties gemaakt van de toepasbare modelfragmenten en probeert Garp recursief met de nieuwe informatie opnieuw modelfragmenten toe te passen.

## **2.5 Modelingrediënten**

### **2.5.1 Entiteiten, attributen en structurele relaties**

Entiteiten representeren de (fysieke) objecten die onderdeel zijn van het domein en zijn van een bepaald type. Deze typen zijn georganiseerd in een hiërarchie van subtype relaties. Entiteiten, bijvoorbeeld een auto, kunnen een attribuut hebben in de vorm van een attribuut relatie, zoals 'heeft kleur' en een attribuut waarde. Hierdoor is te representeren dat het een rode auto betreft. Structurele relaties tussen entiteiten bestaan uit een relatie en twee entiteiten. Deze relaties komen vaak overeen met de fysieke structuur van de entiteiten. De entiteiten hoeven niet een directe representatie te zijn van een fysieke entiteit. Ze kunnen een functionele abstractie van de fysieke werkelijkheid zijn.

De elementen komen in twee typen voor: Generieke concepten worden weergegeven in een hiërarchie van generieke beschrijvingen als onderdeel van de domeinspecifieke ontologie om de focus van het redeneerproces te richten op bepaalde entiteiten. Van deze generieke concepten worden instanties gemaakt om de specifieke objecten uit het systeem mee weer te geven, van het concept auto kunnen instanties worden gemaakt zoals de 'Saab van mama'.

### **2.5.2 Kwantiteiten, waardenruimte, waarde en afgeleide**

Aan elke entiteit kunnen meerdere kwantiteiten toebehoren. Deze kwantiteiten kunnen in de loop van de simulatie een bepaalde waarde hebben afhankelijk van welke waarden mogelijk zijn. Hetzelfde geldt voor de afgeleide van de kwantiteit: deze geeft aan of de waarde van de kwantiteit stijgt, daalt, gelijk blijft of onbekend is. De veranderingen in de waarde van een kwantiteit, in combinatie met de veranderingen in relaties tussen entiteiten, beschrijven het gedrag in de simulatie. De 'quantity space' (Forbus 1984) of waardenruimte van een grootheid geeft aan welke waarden een kwantiteit kan krijgen in de vorm van een ordinale verzameling waarden. De waarden representeren de specifieke invulling of resolutie die een kwantiteit kan hebben op een bepaald tijdstip. In het geval van Garp kunnen deze waarden zich op één punt ofwel ijkwaarde in de waardenruimte bevinden of een waarde hebben die tussen twee ijkwaarden in zit. Deze tussenwaarden zijn expliciet in de waardenruimte gerepresenteerd. In de reeks 'zero, plus, max' representeert de waarde 'plus' een waarde die ligt tussen de ijkwaarden 'zero' en 'max'. Zodra een waarde overgaat van een oude naar een nieuwe waarde, ontstaat een nieuwe toestand.

### **2.5.3 Afhankelijkheden tussen kwantiteiten**

Afhankelijkheden zijn relaties tussen kwantiteiten; ze zijn onder te verdelen naar causale afhankelijkheden en mathematische afhankelijkheden. Causale afhankelijkheden zijn de functionele relaties en maken verandering in de simulatie mogelijk; een verandering van een kwantiteit kan een keten aan veranderingen tot gevolg hebben bij andere kwantiteiten. Er bestaan twee verschillende soorten die elk een positieve en negatieve variant hebben. Ten eerste een 'proportionaliteit' waarmee een relatie kan worden aangegeven tussen de twee afgeleiden van kwantiteiten. De ontwikkeling van de

waarde van een kwantiteit is afhankelijk van de ontwikkeling van een andere variabele: stijgend, dalend of constant. Ten tweede de 'invloed' waarbij de afgeleide van een kwantiteit afhankelijk is van de waarde van een andere kwantiteit.

Mathematische afhankelijkheden tussen kwantiteiten zijn beperkingen op de waarde of de afgeleide tussen twee kwantiteiten. Deze afhankelijkheden zijn geen causale relaties maar beperkingen van het gedrag van de kwantiteit. Normaliter worden ze ingezet om condities te formuleren voor de toepasbaarheid van een modelfragment op een situatie (Bredeweg, 1992). Zogenaamde 'inequalities' bestaan in de vorm van mathematische vergelijkingen ( $<$ ,  $\square$ ,  $=$ ,  $\square$  en  $>$ ) waarmee bijvoorbeeld aangegeven kan worden dat de waarde van een variabele groter is dan de waarde van een andere variabele.

Een tweede type mathematische afhankelijkheid is de 'correspondence' waarmee specifieke kwalitatieve waarden aan elkaar gerelateerd worden. Door een correspondentie van waarden is aan te geven dat bijvoorbeeld het maximum van een variabele samengaat met het maximum van een andere variabele.

## **2.6 Model constructies**

De hierboven beschreven modelingrediënten worden gebruikt om model constructies samen te stellen: Scenario's, modelfragmenten en terminatieregels. Een scenario is een gedeeltelijke situatiebeschrijving die als input dient voor de simulatie. Het is de initiële beschrijving van het systeem waarvan de elementen en onderlinge functionele relaties dienen als conditie voor het redeneer proces.

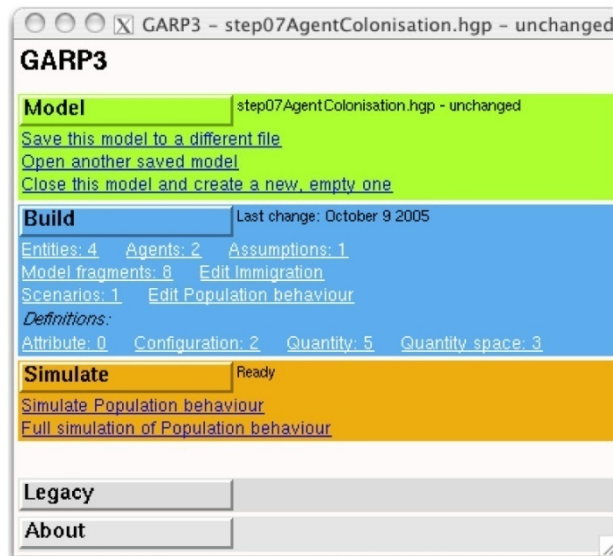
Modelfragmenten zijn generieke beschrijvingen van stukken kennis bestaande uit een conditioneel deel en een resultaat. Indien elementen of functionele samenstellingen van elementen uit de huidige toestand, of scenario, overeenkomen met de algemene beschrijving uit de conditie van het modelfragment dan is het mogelijk het resultaat toe te voegen aan de huidige toestand. Hierdoor kunnen entiteiten, kwantiteiten of attributen worden toegevoegd, structurele of causale relaties worden aangegeven of kwantiteiten en afgeleiden van een waarde worden voorzien. Daarnaast kunnen processen starten en stoppen. Als dit tot gevolg heeft dat een kwantiteit van waarde verandert binnen de waardenruimte of wanneer een kwantiteit verandert ten opzichte van een andere kwantiteit ontstaat een nieuwe kwalitatieve toestand.

Deze nieuwe toestand is het gevolg van terminatieregels. Dit zijn veelal generieke regels waarin bepaald wordt wat het gedrag van de waarde is van een kwantiteit gegeven de huidige waarde en afgeleide. Wanneer meerdere kwantiteiten van waarde veranderen, zijn meerdere terminatie regels van toepassing op dezelfde toestand.

Een belangrijke eigenschap van kwalitatieve modellen vormt de ambiguïteit. Indien zowel positieve als negatieve invloeden gelden is het netto resultaat ambigu en ontstaan meerdere toestanden. Dit onvoorziene resultaat kan inzichten bieden over het gedrag van het systeem.

## **3 Garp3**

Recent werd in het Garp framework een geïntegreerd systeem voor kennis modelleren ontwikkeld in het kader van het project NaturNet-Redime. In dit project staat leren over duurzame ontwikkeling door het modelleren en simuleren van kwalitatieve modellen centraal. Enerzijds bestaat het uit de ontwikkeling van een geïntegreerd systeem van Garp met twee eerder ontwikkelde systemen, anderzijds wordt domein specifieke kennis ontwikkeld door experts in duurzame ontwikkeling.

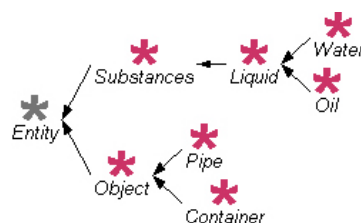


Figuur 1. Initieel Garp3 hoofdmenu

In het NaturNet-Redime project zijn 3 tools geïntegreerd tot een omgeving die alle functionaliteiten biedt van de afzonderlijke tools als zijnde een enkel computerprogramma (Bredeweg e.a., 2005). Naast de functionele integratie is er een uniform gebruik van grafische elementen als iconen, diagrammen, het gebruik van kleur en vocabulaire voorzien. De ingrediënten van Garp3 bestaan uit: Garp voor het simuleren van de kwalitatieve modellen, Homer (Jellema, 2000) en VisiGarp (Bouwer, 2005).

### 3.1 Homer

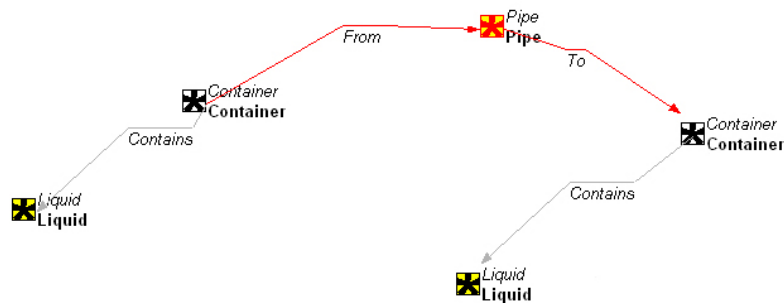
Homer is een systeem voor de bouw van modellen die in Garp gesimuleerd kunnen worden. Door het gebruik van een grafische interface is er geen kennis meer nodig van de onderliggende 'tekstuele' prolog code waarin Garp is geïmplementeerd. De modellen gecreëerd in Homer kunnen worden geëxporteerd naar prolog-bestanden, die vervolgens kunnen worden ingelezen in Garp. Het modelbouw proces bestaat uit een aantal zogenaamde 'Editors' waarin verschillende modelbouw taken worden uitgevoerd. Voor elk van de afzonderlijke onderdelen van het QR vocabulaire van Garp bestaat een editor. Het formuleren van een model gebeurt initieel in een sequentie van editors, omdat bepaalde modelingrediënten zoals entiteiten moeten bestaan voordat ze in samengestelde elementen als scenario's kunnen worden gebruikt.



Figuur 2. Isa-hiërarchie in Homer

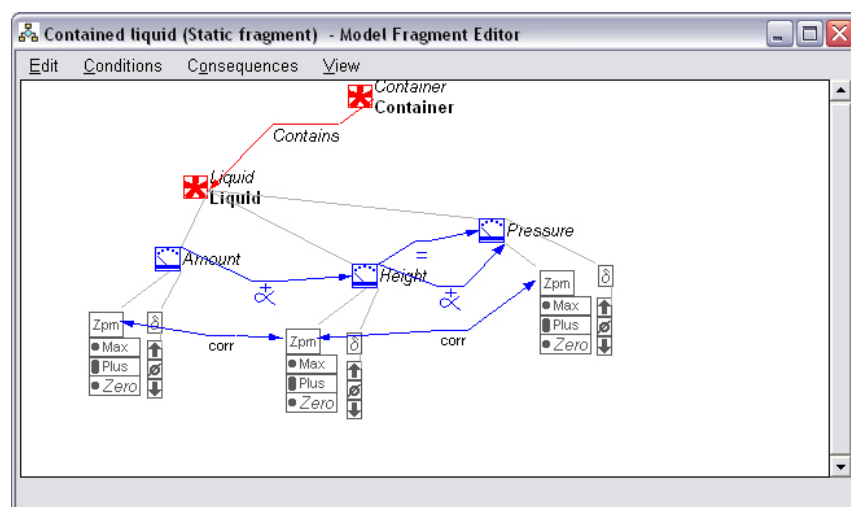
Ter illustratie van grafische representatie hierbij enkele voorbeelden in de vorm van de entiteiten in de isa-hiërarchie (zie Figuur 2) en een weergave van structurele relaties (zie Figuur 3). Voor entiteiten wordt een icon gebruikt met een arbitraire vorm, zodat deze domeinonafhankelijk kan worden ingezet.

Vervolgens worden variaties hiervan gebruikt in kleur en omlijning om visuele variatie aan te brengen tussen klassen en instanties en het gebruik binnen de simulatie omgeving. Structurele relaties worden weergegeven als een gerichte graaf.



Figuur 3. Structurele relaties in Homer

De afzonderlijke editors zijn windows voorzien van een 'context-sensitive' menubalk bestaande uit knoppen met een tekstuele label. Afhankelijk van de huidige selectie van elementen zijn alleen operaties mogelijk vanuit het menu die relevant zijn voor die elementen. Het toevoegen van relaties tussen elementen is alleen mogelijk indien twee elementen geselecteerd zijn. In het onderstaande voorbeeld van de modelfragment editor illustreert het gebruik van kleur: rood voor de conditionele elementen van het modelfragment en blauw voor toe te voegen resultaat. Op 'belongs-to' relaties (grijze lijnen in Figuur 4) na worden alle relaties weergegeven voorzien van een label om het type aan te geven.

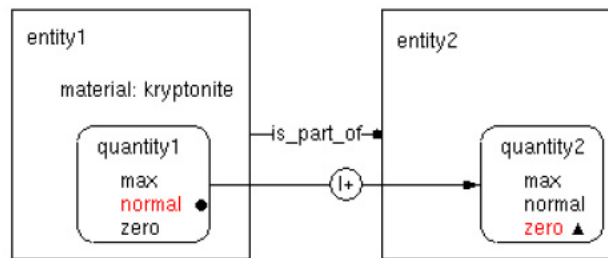


Figuur 4. Modelfragment in Homer

### 3.2 VisiGarp

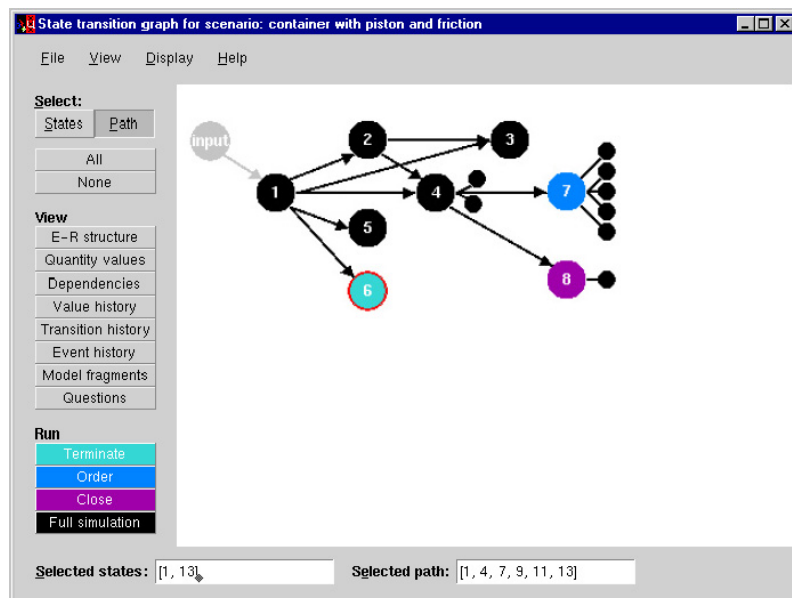
VisiGarp (Bouwer, 2005), het derde onderdeel van het nieuwe Garp3 raamwerk hangt als een visuele laag over Garp heen, het is een grafische representatie voor de controle en inspectie van simulaties en modellen. Deze representatie bestaat uit een serie domein onafhankelijke en automatisch gegenereerde 'views' die veelal tegelijk open staan om vergelijking van de verschillende onderdelen mogelijk te maken. In tegenstelling tot de grafische representatie van Homer worden belongs-to relaties in VisiGarp weergegeven door de visuele primitieven te combineren door deze in of naast elkaar weer te geven (zie Figuur 5). Alle kwantiteiten die tot een entiteit behoren worden in de entiteit weergegeven en de

waardenruimte wordt onder de kwantiteit weergegeven waartoe deze behoort. In plaats van het gebruik van pictogrammen voor het aanduiden van het type zoals bij Homer wordt in dit geval het type aangeduid door de geometrische vorm van het element: Vierkant voor entiteiten en vierkant met afgeronde hoeken voor kwantiteiten. Behalve dat dit een representatie is die goed aansluit bij de manier waarop mensen denken over de belongs-to relatie, heeft deze representatie een tweetal voordelen: Ten eerste is de grafische representatie domeinonafhankelijk en ten tweede biedt het studenten de mogelijkheid hetzelfde soort diagrammen te tekenen, waardoor ze niet afhankelijk zijn van het systeem om hun gedachten op papier te formuleren.



Figuur 5. Voorbeeld van grafische representatie VisiGarp

Het hoofdvenster van VisiGarp bestaat uit een aantal onderdelen. Ten eerste een serie dropdown menu's voor algemene operaties zoals het laden van modellen en aanpassen van de interface. Ten tweede de onderdelen die betrekking hebben op de huidige simulatie: enerzijds, in het weergave vlak, een weergave van de toestanden van de huidige simulatie en anderzijds, in de balk links, operaties die hierop mogelijk zijn: instellen van een selectietool voor het selecteren van individuele toestanden, paden tussen toestanden, alle of geen toestanden. Aan de hand hiervan kunnen de geselecteerde toestanden geïnspecteerd worden met de verschillende 'viewers'. Tevens is het mogelijk om met de simulatieoperaties de geselecteerde toestanden te simuleren door Garp ('run').



Figuur 6. Kleurgecodeerde toestanden en operaties hierop in VisiGarp



De visualisatie van de toestanden (Figuur 6) is kleurgecodeerd waarbij een arbitraire keuze van kleuren terugkomt in het menu van simulatie operaties. Hierdoor is in één oogopslag zichtbaar in welke fase elke toestand binnen de simulatie is.

### **3.3 Onderweg naar Garp3**

Van de drie bestaande systemen bezitten er twee een interface waarbij een grafische representatie wordt gebruikt voor de primitieven uit het QR vocabulaire. In de ontwikkeling van een naadloos systeem zijn er een aantal knelpunten. Ten eerste liggen er geheel verschillende grafische principes ten grondslag aan de visualisatie van de QR-primitieven. In het geval van VisiGarp is er geen sprake van een iconografische weergave van kwantiteiten of entiteiten; deze zijn herkenbaar door de grafische stijl waarin ze worden weergegeven. Ten tweede is er een verschil in het gebruik van kleur. Ondanks dat in beide systemen op overeenkomstige manier gebruik wordt gemaakt van rood en blauw voor conditionele elementen en resultaten, bevat VisiGarp verder geen kleur, terwijl Homer een aantal kleuren hanteert voor het aangeven van het verschil in verschijningsvormen van instanties: type, klasse of instantie. Naast de verschillen in de manier van visualiseren is er verschil in de wijze waarop interactie plaatsvindt; de manier waarop operaties moeten worden uitgevoerd verschilt per systeem. In beide gevallen komt de weergave van de mogelijke operaties wel overeen: alle operaties worden uitgevoerd door op een tekstueel label te drukken voor de uitvoer van een bepaald commando.

In de ontwikkeling naar één platform vanuit een aantal bestaande systemen is een samenhangende grafische taal van belang; hierdoor is men niet genooddacht cognitieve vertaalslagen te maken tussen de verschillende onderdelen van het systeem. In zowel VisiGarp als Homer worden meerdere grafische kanalen gebruikt voor de modelingrediënten en in enkele gevallen ook voor de operaties daarop. Deze visualisaties zijn in de nieuwe Garp3 aanpak, voor zover de implementatie en onderliggende grafische principes van de systemen het toestonden, gestroomlijnd tot een enkele grafische taal voor zowel de operaties als de modelingrediënten. Hierbij is zoveel mogelijk gebruik gemaakt van twee kanalen, bestaande uit een variabel tekst kanaal en een type-aanduidend visueel kanaal. Vervolgens is deze grafische taal zoveel mogelijk doorgevoerd in het gehele Garp3 systeem zoals het hoofdmenu en onderliggende windows en editors.

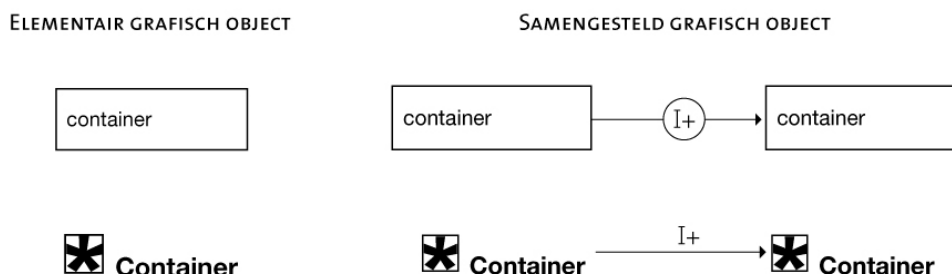
## **4 Grafische principes**

In de eerder genoemde systemen worden visualisatie gebruikt voor de primitieven uit het QR vocabulaire van Garp door middel van een eigen grafische syntax. Aan de hand van het werk van Engelhardt (2002) kunnen deze primitieven in een tweetal groepen worden onderverdeeld: Ten eerste elementaire grafische objecten. Dit zijn de grafische objecten die na zogenaamde syntactische decompositie overblijven. Hierbij gaat het niet zozeer om lijnstukken die samen één enkel vierkant vormen maar om het vierkant als de uiteindelijke afbeelding van een bepaalde betekenis. Een icon op een computerscherm is eerder een elementair object dan de pixels waaruit het is opgebouwd.

Ten tweede de samengestelde grafische objecten. Dit zijn samengestelde afbeeldingen van meerdere elementaire grafische objecten in één vlak. Alle elementen in een samengestelde afbeelding zijn onderdeel van één of meer onderlinge grafische relaties. Deze kunnen bestaan uit relaties tussen elementen en het vlak waarin ze voorkomen of als relaties tussen de elementen onderling. Ook is het

goed mogelijk dat deze samengestelde afbeeldingen weer uit één of meer samengestelde afbeeldingen bestaan totdat de elementaire grafische objecten overblijven.

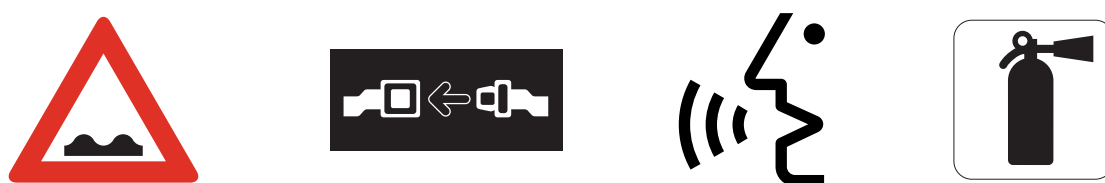
Opvallend aan de grafische syntax van de ontwikkelde visualisatie en modelleer tools voor het Garp framework is dat er weinig overeenkomsten zijn tussen de elementaire grafische objecten terwijl er wel veel overeenkomsten zijn in de samengestelde grafische objecten (zie Figuur 7).



Figuur 7. Verschil en overeenkomst Homer en VisiGarp

#### 4.1 Iconen en grafische representatie

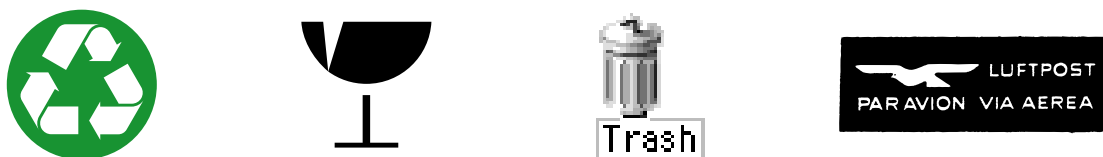
Het definiëren van een visuele taal bestaat uit het vinden van een goede overeenkomst tussen enerzijds wat weergegeven wordt en wat er bedoeld wordt anderzijds. De belangrijkste vraag is of de representatie inderdaad het gerepresenteerde weergeeft. In ons geval gaat het om de relatie tussen de weergave die we gebruiken voor de modelingrediënten uit het QR vocabulaire en hun betekenis. In navolging van Engelhardt en eerder literatuuronderzoek (Bertels, 2001) is deze relatie op een aantal manieren vorm te geven: Ten eerste kan de relatie tussen de weergave en wat er bedoeld wordt, gebaseerd zijn op letterlijke overeenkomst. De relatie is letterlijk als dat wat getoond wordt overeenkomstigheden vertoont met het fysieke object of de fysieke structuur van wat bedoeld wordt. Dit kan ook een overeenkomst zijn met een prototypisch voorbeeld van het fysieke object dat bedoeld wordt. Zie Figuur 8 voor enkele voorbeelden van letterlijke overeenkomsten. De hobbels in de weg waarvoor gewaarschuwd wordt, worden weergegeven als prototypische oneffenheden en hoeven niet overeen te komen met de werkelijke hoogte van de hobbels in de weg. Hetzelfde geldt voor de stoelriemen en de brandblusser. De afbeelding voor 'spraak', de derde in de rij, geeft aan dat letterlijke overeenkomsten meer zijn dan fysieke objecten alleen maar ook voor algemenere fysieke concepten bruikbaar is.



Figuur 8. Voorbeelden letterlijke overeenkomst

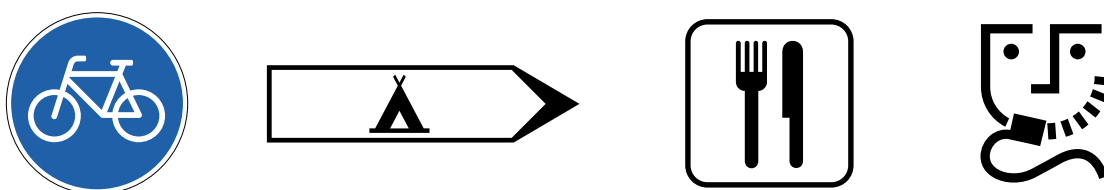
Ten tweede kan de overeenkomst tussen weergave en betekenis gebaseerd zijn op metaforische overeenkomst. Hiervan is sprake als er aangenomen wordt dat er een analogie geldt tussen wat wordt weergegeven en wat wordt bedoeld. Veel voorbeelden van metaforische relaties zijn terug te vinden in icons gebruikt in computer programma's. Hierbij valt te denken aan het huisje voor de homepage of start pagina van een gebruiker, de vuilnisbak voor het weggooien van bestanden of de folder voor het verzamelen ervan, zie Figuur 9. In verpakkingen komen ook geregeld metaforische relaties voor: wanneer

een produkt van gerecycled materiaal is gemaakt wordt daarvoor de afbeelding gebruikt van drie pijlen die samen een cyclus vormen. Indien iets breekbaar is wordt dat dikwijls op de verpakking aangegeven door een analogie te maken met de breekbaarheid van glas.



**Figuur 9. Voorbeelden van metaforische overeenkomst**

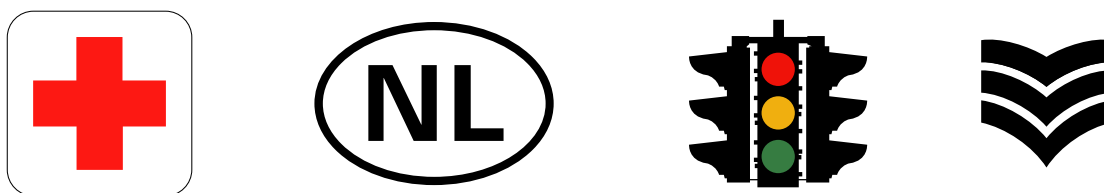
Ten derde kan gebruik worden gemaakt van metonymische overeenkomst. Dit type is gebaseerd op een mentale associatie die het gevolg is van een relatie tussen fysieke betrokkenheid tussen wat wordt getoond en wat wordt bedoeld. Dat wat wordt afgebeeld kan bijvoorbeeld 'een onderdeel zijn van' of 'een mogelijk resultaat zijn van' wat wordt bedoeld. Dit is wezenlijk verschillend van het vorige type overeenkomst omdat er nu geen vergelijking wordt gemaakt of een analogie wordt verondersteld maar er sprake is van een fysieke relatie. In de voorbeelden in Figuur 10 worden in de weergave fysieke elementen getoond die een rol spelen in dat waarnaar verwezen wordt (fietsers op het fietspad, camping, restaurant en stopcontact in de badkamer). In de eerste drie gevallen zijn de afgebeelde elementen een fysiek onderdeel van dat wat bedoeld wordt. In het laatste geval betreft het een mogelijk resultaat; afgebeeld als het scheren dat is mogelijk geworden door het stopcontact.



**Figuur 10. Voorbeelden van metonymische overeenkomst**

Ondanks dat Engelhardt een duidelijk verschil ziet tussen de metaforische en metonymische overeenkomst, is dat verschil minder duidelijk tussen de letterlijke en metonymische overeenkomst: er staan tenten op de camping dus het bord in Figuur 10 is in dat geval een letterlijke overeenkomst met wat bedoeld wordt. Het voorbeeld is een uiting van letterlijke overeenkomst als het een tentencamping betreft maar metonymisch als er ook caravans kunnen staan. Vanwege het feit dat de letterlijke overeenkomsten ook prototypisch kunnen zijn is het onderscheid niet altijd duidelijk. Een voorgestelde oplossing hiervoor is het onderscheid in volledigheid van de beschreven situatie. In het geval van letterlijke overeenkomst wordt een bepaalde situatie weergegeven en vaak een actie geïmpliceerd. Bijvoorbeeld de aanwijzing om de vliegtuiggordels te sluiten: Deze beschrijving van actoren in die situatie is altijd volledig bij een letterlijke overeenkomst. Om de gordels te kunnen sluiten moeten die twee onderdelen in elkaar geschoven worden en niet meer dan dat. Er is een directe relatie tussen wat bedoeld wordt en wat getoond wordt. Dit in tegenstelling tot de metonymische overeenkomst waarin situaties worden geschetst die niet volledig zijn. Hierbij wordt ruimte gelaten aan de interpretatie van de kijker over wat gerepresenteerd is in de grafische representatie.

Tenslotte kan een relatie gebaseerd zijn op arbitraire conventie (Figuur 11). In dit geval wordt de betekenis van dat wat is afgebeeld alleen duidelijk door conventie. Volgens Engelhardt hebben alle grafische objecten een herkomst in de vorm van een verklaring in termen van één van de andere typen relaties. Indien deze herkomst verloren gaat maar de grafische relatie blijft bestaan tussen een afbeelding en wat wordt bedoeld, dan is de relatie van dit type. Ondanks dat het een enorme verzameling betreft en de originele verklaring vergeten is voor de gekozen visualisatie en de bedoelde betekenis, kan het een bruikbare manier zijn voor een grafische representatie: de arbitraire conventie kan erg universeel zijn en dan door veel mensen begrepen worden.



Figuur 11. Voorbeelden van overeenkomst op basis van arbitraire conventie

In het geval van de visualisatie van primitieven uit het QR domein is er geen aanpak die zondermeer aansluit. Hoewel een letterlijke en metonymische overeenkomst een goede overeenkomst mogelijk maakt tussen wat weergegeven en wat bedoeld wordt, is dit onbruikbaar voor de QR primitieven. Alle onderdelen zijn domein onafhankelijk en daarmee van nature zonder overeenkomst met een specifiek element in de werkelijke wereld. Daarnaast is er geen domein in diezelfde werkelijke wereld waarbij het gehele vocabulaire weergegeven kan worden door middel van metaforische relaties. Hierbij valt te denken aan katrollen en touwen waarbij de entiteiten de katrollen zijn, de kwantiteiten de touwen en de lengte van het touw de waardenruimte. De touwen staan dan in verbinding met andere touwen op andere katrollen waarbij door bepaalde kracht op de touwen uit te oefenen de katrollen gaan draaien en de lengtes van de touwen veranderen, ofwel de waarde verandert in de waardenruimte. In dit soort analogieën is het echter niet mogelijk de rijkdom van het QR vocabulaire uit te drukken of om complexe modellen weer te geven met de grote variatie aan relaties dat er kunnen gelden tussen modelingrediënten.

Ondanks dat geen van de besproken overeenkomsten toepasbaar is voor de grafische representatie van het gehele QR vocabulaire is het mogelijk de metaforische overeenkomst te gebruiken voor afzonderlijke pictogrammen in het vocabulaire. Een letterlijke of metonymische overeenkomst is uitgesloten omdat de QR primitieven allen abstracte elementen zijn waarbij geen fysieke overeenkomst mogelijk is. Als gevolg zijn overeenkomsten gebaseerd op arbitraire conventie bruikbaar maar deze zijn dan altijd te herleiden tot een metaforische relatie.

## 4.2 Samengesteld visualiseren

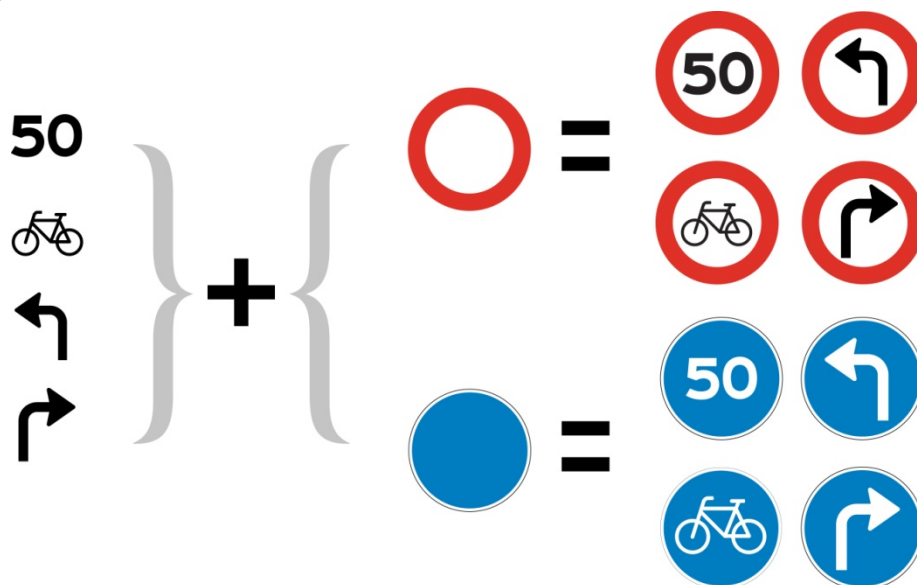
Door een onderlinge samenhang tussen afbeeldingen te hanteren is het mogelijk een consistente grammatica te maken gebaseerd op zo veel mogelijk hergebruik van de visuele elementen. Deze grammatica kan bijdragen aan sneller begrip van het vocabulaire als geheel. Er zijn twee verschillende manieren waarop dit mogelijk is: Ten eerste door het gebruik van zogenaamde 'modifiers' en ten tweede door samenstellingen te maken van bestaande elementen uit het grafische grammatica.



Figuur 12. Twee voorbeelden van samengesteld visualiseren

Aan de linker zijde van Figuur 12 staan twee verkeersborden met dezelfde inhoud, namelijk de pijl naar rechts. Ze zijn echter door verschillende ‘modifiers’ (Engelhardt, 2002) omsloten waardoor ze een complementaire betekenis krijgen. Ook al kent iemand slechts één van de twee borden is de betekenis van de andere vrijwel meteen duidelijk omdat de inhoud al bekend is en alleen de modifier een interpretatie behoeft. Hiernaast is de interpretatie van de betekenis van borden met andere inhoud, bijvoorbeeld pijlen een andere kant op, snel gemaakt. In dat geval is de betekenis van de ‘modifier’ bekend en moet enkel de inhoud van de afbeelding worden geïnterpreteerd.

Door het gebruik van modifiers is het mogelijk twee verschillende grammatica’s met een eigen semantiek te gebruiken waardoor de visuele taal als geheel sneller te leren is. Er zijn in dat geval minder visuele primitieven nodig om hetzelfde uit te drukken (zie Figuur 13). Paradoxaal genoeg is dit alleen het geval als er voldoende primitieven zijn. Bij vier grafische primitieven, namelijk ‘naar links’, ‘naar rechts’, ‘toegestaan’ en ‘verboden’ zijn deze slechts te combineren naar vier verschillende samengestelde visualisaties.



Figuur 13. 4 representaties + 2 modifiers = 8 resultaten

Het definiëren van een grammatica bestaat uit het zoveel mogelijk visueel hergebruik van elementen op een consistente manier. Dit hergebruik kan het begrip van een vocabulaire cumulatief laten verlopen. Aan de rechterkant in Figuur 12 staan een floppydisk afgebeeld en een pictogram voor het invoeren van een floppydisk in de computer. Ondanks dat de tweede afbeelding vrij onduidelijk is in wat er moet gebeuren met de floppydisk, staat onmiskenbaar vast dat er ‘iets’ moet gebeuren met de floppydisk en niet met een cd-rom of de muis. Deze manier van samengestelde visualisaties heeft niet tot gevolg dat er minder visuele primitieven nodig zijn in het vocabulaire als geheel, het enige voordeel is dat leerbaarheid van nieuwe grafische elementen wordt vergroot.

### 4.3 Kleur

Het gebruik van kleur is op een viertal manieren mogelijk in een grafische representatie (Tufte, 1990). Ten eerste kan kleur worden gebruikt om elementen van elkaar te laten verschillen. Dit is een nominale verdeling van elementen in kleur. Ten tweede kan kleur worden gebruikt voor het kwantificeren van informatie. Een voorbeeld van dit gebruik van kleur zijn hoogtekarten waarbij verschillende hoogtes een kleur krijgen. In veel gevallen wordt kleur op ordinale manier ingezet, herkenbaar aan het representeren van een kwantiteit in een kleurverloop. Daarnaast is het mogelijk kleur te gebruiken voor het representeren of imiteren van de realiteit. Tenslotte is kleur te hanteren om een grafische representatie te verfraaien of verlevendigen.

Het gebruik van donkere iconen op een lichte achtergrond, zie de linkerkant van Figuur 14, gaan als silhouet werken waardoor het door gebruikers meer met het oorspronkelijke object wordt vereenzelvigd dan in het geval van diapositieve visualisatie (Ruisch & Theys, 1998).

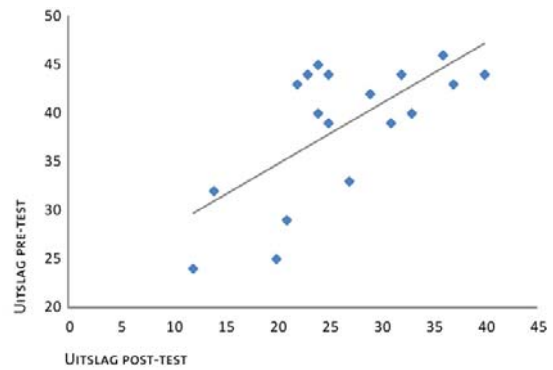
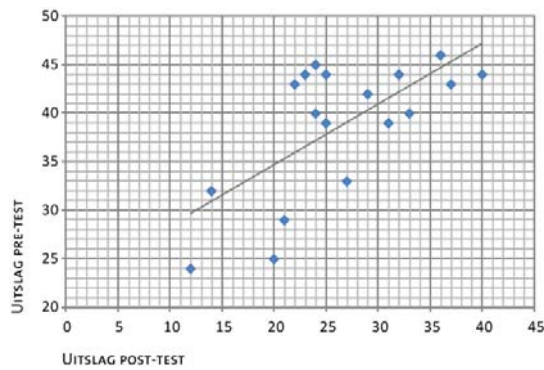


Figuur 14. Positieve en diapositieve weergave van zelfde pictogram (Hora, 1996)

### 4.4 Data-inkt

Bij de weergave van statistische gegevens is het van belang er zorg voor te dragen dat de aandacht van de gebruiker uitgaat naar de representatie van de data en niet naar overbodige elementen in de weergave. Het fundamentele principe van goede statistische weergaven luidt dan ook: 'above all else, show the data' (Tufte, 1984). Hiervoor introduceert Tufte de data-inkt ratio, een verhouding tussen de hoeveelheid inkt in een grafische representatie die bestaat uit data-informatie en overbodige informatie. De term inkt is inwisselbaar voor pixels in het geval van een computerscherm. Deze data-informatie betreft niet alleen de grafische representatie van de data maar ook de noodzakelijke informatie om de data te kunnen interpreteren, zoals de assen waarlangs deze data wordt weergegeven en labels voor de data, assen of legenda. Een hoge verhouding data-inkt is herkenbaar aan de hoge mate waarin de grafische representatie als geheel verandert wanneer de data verandert.

In Figuur 15 zijn twee voorbeelden weergegeven van dezelfde statistische gegevens. In de linker tabel is, volgens Tufte de verhouding data-informatie lager dan in de rechter tabel. Door het grid te verwijderen ontstaat een grafische weergave waarin de verhouding data-informatie hoger is. Hoewel dit vaak het geval is voor gridlijnen in een tabel is dit niet altijd het geval. De definities van welke onderdelen van de weergave data-informatie zijn, zijn geen eigenschappen van de weergave maar van de gebruiker hiervan. Deze gebruiker hanteert de weergave in de uitvoer van een bepaalde taak. Aan de hand van de gridlijnen in de tabel is in minder tijd beter te bepalen wat de exacte waarde van afzonderlijke elementen is dan in een tabel zonder deze lijnen. Echter, bij het identificeren van een puntenwolk zijn deze lijnen overbodig en bevatten ze geen data-informatie. Afhankelijk van de taak zijn de lijnen data-informatie of niet. De data-inkt ratio verhogen is volgens Tufte mogelijk door ofwel zoveel mogelijk non-data inkt te verwijderen ofwel door overbodige data-inkt te verwijderen.



**Figuur 15. Twee tabellen met verschillende hoeveelheid data-inkt**

De inkt-data verhouding is ook toepasbaar in de context van pictogrammen. Elk pictogram bestaat uit een collectie van lijnen en vlakken, al dan niet in een kleur of grijs tint als grafische representatie van een bepaald concept. In de ontwikkeling van pictogrammen is het zaak zorg te dragen spaarzaam te zijn met de hoeveelheid inkt. Dit is echter complementair aan de eerdere opmerkingen met betrekking tot het hergebruik van grafische primitieven omdat hierin juist het belang van hergebruik van al bekende elementen van een pictogram in een ander pictogram wordt geïllustreerd. Dit zijn bij uitstek redundante elementen omdat elke combinatie van modifier en ingrediënt of combinatie van ingrediënten een bepaalde mate van redundantie impliceert. Deze redundantie is echter gerechtvaardigd door grotere leerbaarheid van het pictogram. Zonder het hergebruik van visuele elementen zijn alle pictogrammen uniek en daarom lastiger leerbaar.

#### **4.5 Operaties**

Het uitvoeren van taken door gebruikers in een interface gebeurt door middel van objecten voor het activeren van operaties. Voor deze objecten is de zichtbaarheid en functionele indicatie ('affordance') van belang (Norman, 1988). Een deurknop is een voorbeeld van een functionele indicatie voor het openen van de deur. Objecten hebben bepaalde functionele eigenschappen die een indicatie zijn van de operaties en manipulaties die mogelijk zijn met dat object. Hierbij is de waargenomen indicatie ('perceived affordance') van groot belang. De functionele indicatie is niet werkelijk een eigenschap van een object maar een beeld dat de gebruiker van het object heeft. De functionele indicatie bestaat uit de mogelijke operaties die de gebruiker er mee denkt uit te kunnen voeren.

Voor het uitvoeren van operaties of acties in een computer applicatie worden vaak knoppen gebruikt. Deze knoppen zijn voorzien van een grafische of tekstuele label. Deze label dient ten eerste om de knop te onderscheiden van de andere knoppen en ten tweede als representatie van de actie die er via de knop wordt uitgevoerd. In het geval van de bediening van analoge apparaten bestaat de volledige interface vaak uit een beperkte hoeveelheid acties waarvan wordt verondersteld dat deze bekend zijn bij de gebruiker. In deze gevallen is de mogelijkheid de knoppen van elkaar te onderscheiden voldoende en worden knoppen vaak niet voorzien van een representatie van de actie (zie de parkeermeter in Figuur 16).





Figuur 16. Knoppen zonder uitleg

Het representeren van een actie kan op verschillende manieren. Ten eerste is het mogelijk de actie zelf te representeren. In dat eerste geval wordt een gehele of gedeeltelijke representatie weergegeven van de actie die uitgevoerd wordt op het moment van het activeren van de knop. Hierbij moet opgemerkt worden dat een gedeeltelijke representatie vaak bestaat uit een functionele abstracte beschrijving. In het geval van de tweede parkeermeter aan de linkerkant van Figuur 17 wordt de actie voor het uitprinten van het kaartje enkel gerepresenteerd als een groene knop. Er is in dit geval geen sprake van een volledige representatie van de actie maar een abstracte representatie van een stoplicht waarmee enkel het doorgaan of niet doorgaan met het proces van het kopen van een kaartje is gerepresenteerd. Een voorbeeld van een meer volledige representatie van acties is weergegeven in de liftknoppen voor het openen en sluiten van de deuren in de rechterkant van Figuur 17. In deze afbeelding wordt de ruimte tussen de deuren weergegeven (door de verticale lijn) en geven de pijlen de actie aan die uitgevoerd kan worden op deze ruimte.

De overeenkomst tussen een representatie van een actie en de actie kan op verschillende manieren. In veel gevallen worden pijlen gebruikt als symbool voor de actie in combinatie met een letterlijke overeenkomst met de elementen waarop deze actie dan wordt uitgevoerd. In het geval van computerprogramma's wordt vaak gebruik gemaakt van pijlen in combinatie met het visuele vocabulaire van de rest van het programma, ongeacht het soort overeenkomst dat dit vocabulaire heeft met het gerepresenteerde, bijvoorbeeld een enveloppe in het geval van e-mail.

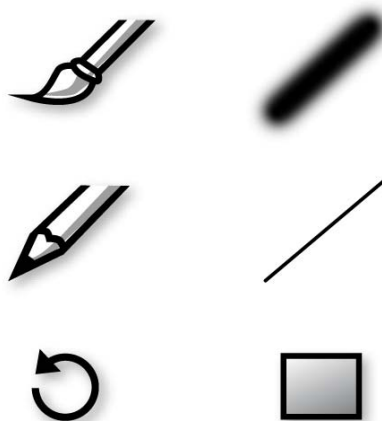


Figuur 17. Knoppen met grafische representatie van de actie, links parkeermeter, rechts lift

Ten tweede kan het resultaat van de actie, of een deel daarvan, worden gerepresenteerd. Dit zijn representaties gebaseerd op de letterlijke overeenkomst tussen de weergave van de actie en het resultaat van de actie. Figuur 18 geeft voorbeelden van acties uit tekenprogramma's: Aan de linkerkant drie voorbeelden van representaties van acties en rechts drie representaties van resultaten van acties. Verschillen in lijnstijl waarmee getekend kan worden, een dikkere wazige lijn of een dunne scherpe, kunnen worden gerepresenteerd als actie waarbij de dikke lijn weergegeven is als het tekenen met een



penseel en de dunne lijn als tekenen met een potlood. Alternatief hiervoor is een representatie in de vorm van een voorbeeld van het resultaat. Het voorbeeld van de dikke of dunne lijn zijn letterlijke overeenkomsten omdat de lijn die je ziet precies de lijn is die je krijgt. De laatste afbeelding uit Figuur 18 is een rechthoek. Los van de context is onduidelijk wat deze actie tot gevolg heeft; het is goed mogelijk dat het computerscherm of window-eigenschappen betreft. Als echter duidelijk is dat het een representatie betreft van het resultaat, bijvoorbeeld doordat het gehele vocabulaire daarop is gebaseerd, dan is duidelijk dat het resultaat simpelweg een vierkant is. Dit is alleen mogelijk in het geval dat acties op een consistente manier worden gerepresenteerd.



**Figuur 18. Linkse representaties van acties, rechts representaties van resultaten van acties**

Ten derde kan de representatie van een actie bestaan uit een representatie van de huidige toestand waarop de actie uitgevoerd kan worden. Een veel gebruikt icon in websites dient om ofwel het geluid 'uit' ofwel het geluid 'aan' te zetten. Hierin worden de operaties op één van twee manieren gedaan vormgegeven. Ten eerste kan een variant van de linker afbeelding uit Figuur 19 zijn weergegeven, waarin is aangegeven dat het resultaat is dat het geluid uit gaat. Ten tweede kan een variant van de rechter afbeelding zijn weergegeven om aan te geven dat het geluid nu 'aan' staat en dat het 'uit' gaat als de knop is gedrukt. In dat geval is sprake van een representatie van de huidige toestand waarop de actie wordt uitgevoerd. Deze manier van representeren is alleen in zeer zeldzame gevallen bruikbaar omdat in veel gevallen uit de initiële situatie niet is af te leiden wat het resultaat zal zijn. Dit impliceert een slechte functionele indicatie van de mogelijkheden van de buttons.



**Figuur 19. Representatie van actie 'geluid aan' of 'geluid uit'**

Een belangrijke overweging in de weergave van operaties is dat het voor mensen makkelijker is informatie te herkennen dan deze uit het geheugen te halen (Preece, 1994). In het geval van operaties is het makkelijker de operatie te herkennen die voldoet aan de taak die uitgevoerd moet worden dan deze in te moeten vullen. In het verlengde hiervan bestaat het verschil tussen 'kennis in de wereld' en 'kennis in het hoofd'. In het eerste geval betreft het kennis die geen onderdeel uit maakt van het geheugen maar 'in de wereld' bestaat. Een voorbeeld hiervan zijn de toetsen op het toetsenbord. Onervaren gebruikers

zoeken de letter die ze nodig hebben uit alle beschikbare toetsen. Het voordeel van kennis in de wereld is dat de operaties niet eerst geleerd hoeven worden en onervaren gebruikers makkelijker hun weg vinden. Door veelvuldig gebruik verschuift de kennis van 'de wereld' naar 'in het hoofd' of geheugen van de gebruiker ervan. Dit neemt niet weg dat mensen geneigd zijn te onthouden wat van belang is voor de uitvoer van hun taken. Hierdoor zal veel niet geleerd worden omdat deze kennis toch 'in de wereld' blijft bestaan. Door een groot deel van de informatie en functionaliteiten onderdeel van de interface te maken is de kennis van de operaties die uitgevoerd kunnen worden in die interface 'in de wereld'. Op deze wijze kan de interface dienen als geheugen voor de gebruiker, zolang deze de kennis nog niet zelf in zijn geheugen heeft. Om deze reden is het raadzaam zoveel mogelijk acties altijd zichtbaar te houden voor de gebruiker. In de eerder ontwikkelde systemen, Homer en VisiGarp, worden veel operaties opgeborgen in 'drop-down' menu's terwijl deze veel beter als extern geheugen kunnen dienen wanneer ze altijd in beeld zijn.

Deze overwegingen over zowel het grafisch representeren van informatie en het representeren van acties liggen ten grondslag aan de nieuwe grafische representatie van Garp3. De ontwikkeling van dit vocabulaire bestaat ten eerste uit de definitie van de modelingrediënten. Voor zover mogelijk is binnen de representatie van het vocabulaire van modelingrediënten hergebruik gemaakt van de elementen. Aan de hand van deze ingrediënten is het vervolgens mogelijk operaties hierop te visualiseren. Vanwege het feit dat de operaties in de eerder ontwikkelde systemen bestaan uit tekstuele representaties was er geen noodzaak deze in groepen onder te verdelen. Door de operaties te verdelen naar de elementen waarop ze van toepassing zijn is het mogelijk dat de operaties beter te begrijpen en leerbaar zijn.

## 5 Grafische representatie Garp3

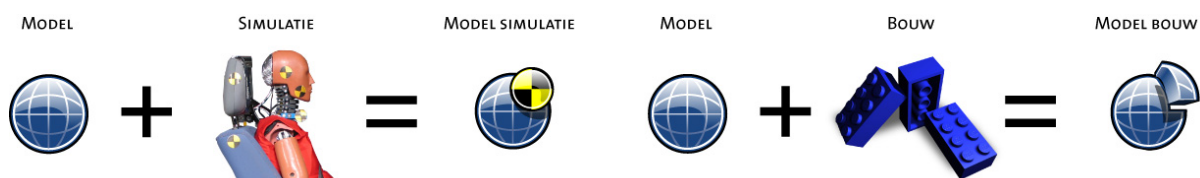
Het model wordt opgebouwd uit entiteiten, eigenschappen, kwantiteiten, modelfragmenten, scenario's, onderlinge relaties, etc. Dit zijn allemaal afzonderlijke modelingrediënten. Het visuele primitief refereert aan een primitief uit het QR vocabulaire. Elk van deze typen moet in het grafische vocabulaire terugkomen. Hoewel dit een abstracte definitie is van de modelleertaal, is het juist door visualisatie mogelijk hier handvatten te bieden om deze ingrediënten duidelijk van elkaar te kunnen onderscheiden en aan te geven wat de rol is van het ingrediënt in het model als geheel. Door een entiteit een ander voorkomen te geven dan een kwantiteit is duidelijk dat deze elementen van een andere soort zijn. Als vervolgens de modelfragmenten worden weergegeven als bakjes waar andere elementen in worden gestopt, wordt duidelijk dat dit modelfragment weer van een derde soort is en een andere rol speelt dan de eerste twee elementen. Zonder te weten wat de elementen betekenen is op deze manier duidelijk dat het laatste element met de eerdere twee elementen te maken heeft en de rol van container vervult. Op deze manier worden handvatten geboden om de modelleertaal te begrijpen.



Figuur 20. Model

In QR wordt gebruik gemaakt van kwalitatieve modellen van een specifiek domein. Dat domein is niet van te voren bepaald. Om die reden is er geen visueel aanknopingspunt of analogie te verzinnen voor de visualisatie van 'een model'. Er is gekozen voor een grafische figuur van een wereldbol zonder daarin landmassa op te nemen, zie Figuur 20. Op deze manier is een model altijd een wereld zonder specifieke relatie met een geologisch domein.

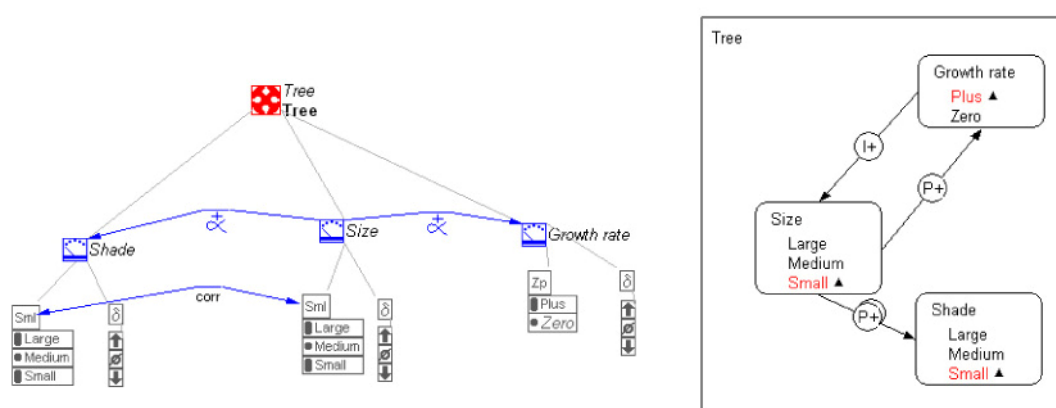
De onderdelen van Garp3 bestaan uit een simulatie en een modelbouw gedeelte. Voor deze twee onderdelen zijn twee verschillende visualisaties gemaakt. Hierbij is het model als uitgangspunt genomen voor het cumulatief visualiseren. Bij modelsimulatie is de typische geel-zwarte sticker voor test opstellingen gebruikt als metafoor voor simulatie. Daarnaast is voor de modelbouw omgeving de maakbaarheid van het model geïllustreerd door de visualisatie van het model uit twee onderdelen te laten bestaan: het model en een element dat wordt toegevoegd. Zie voor een illustratie hiervan Figuur 21. Dit drietal grafische weergaven staat aan de basis van vrijwel alle operaties die in Garp3 kunnen worden uitgevoerd.



Figuur 21. Visuele rationale voor model simulatie en modelbouw afbeelding

## 5.1 Visualisatie modelingrediënten

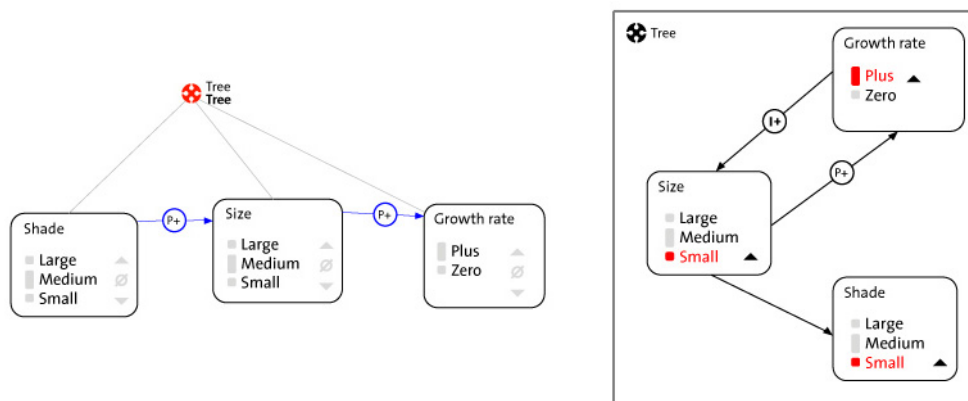
Het Garp3 platform bevat verschillende manieren voor het visualiseren van informatie, die grotendeels afhankelijk zijn van de context waarin de informatie wordt opgevraagd. De inhoud van een modelfragment wordt in de modelbouw context anders gerepresenteerd dan in de simulatie context. Om verwarring tussen de twee grafische talen zoveel mogelijk te beperken, is getracht deze twee visuele vocabulaires zoveel mogelijk op elkaar te laten aansluiten.



Figuur 22. Voorbeeld visuele primitieven uit Homer en VisiGarp

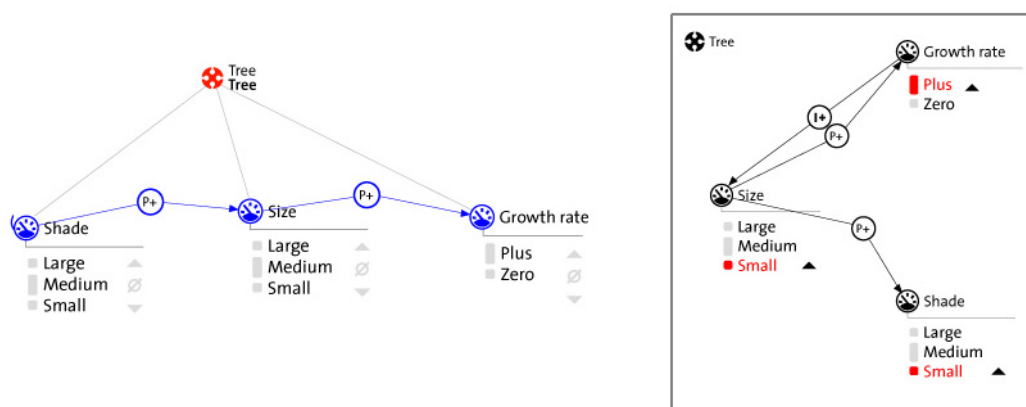
Ter illustratie van de verschillen een illustratie in Figuur 22 van ongeveer dezelfde deelbeschrijving van het gedrag van de ontwikkeling van een boom. Uit dit voorbeeld komen duidelijk de verschillen in grafische principes naar voren die ten grondslag liggen aan de grafische representaties; de weergave van de relatie

‘behoort toe aan’ wordt ofwel weergegeven door middel van een expliciete lijn ofwel door de elementen te omsluiten vanuit een container metafoor.



Figuur 23. Voorbeeld van hybride VisiGarp visualisatie

Voor onervaren gebruikers kan dit dubbele visuele vocabulaire verwarrend zijn en kan het raadzaam zijn meer overlap aan te brengen. In het extreemste geval biedt dit twee mogelijkheden bestaande uit een volledige adaptatie van het andere vocabulaire. Door kleine ingrepen in de representatie van QR-primitieven ontstaat echter al meer visuele overeenkomst, hetgeen bijdraagt aan de leerbaarheid van het systeem als geheel. De twee afbeeldingen die hier gepresenteerd worden verschillen van elkaar in de representatie van de kwantiteiten. In Figuur 23 wordt de container metafoor gebruikt vanuit het VisiGarp vocabulaire in het Homer vocabulaire en in Figuur 24 gebeurt hetzelfde andersom.



Figuur 24. Voorbeeld van hybride Homer visualisatie

Ondanks dat hierbij een duidelijkere visuele overeenkomst is tussen de twee vocabulaires gaat in beide gevallen veel verloren. In het eerste geval waarbij de kwantiteiten als containers dienen voor de waardenruimte en afgeleide blijft weinig ruimte over voor de maker van het model om relaties tussen elementen uit de waardenruimte en afgeleide van relaties te voorzien zonder voor opeenhopingen van relaties te zorgen. In het tweede geval wordt de weergave onnodig onduidelijk en is niet meer leesbaar ‘in één oogopslag’. Hieruit valt op te maken dat meer overlap tussen de grafische representaties wenselijk is vanuit het perspectief van onervaren gebruikers maar onwenselijk in de context waarin de afzonderlijke representaties worden gebruikt. In de uitvoer van taken is de vrijheid van het creëren van een eigen layout in de context van modelbouw en de duidelijkheid in de context van simulatie van groter belang. Deze fundamentele verschillen zijn in dit licht geen tekortkomingen van Garp3 als platform maar ondersteunen

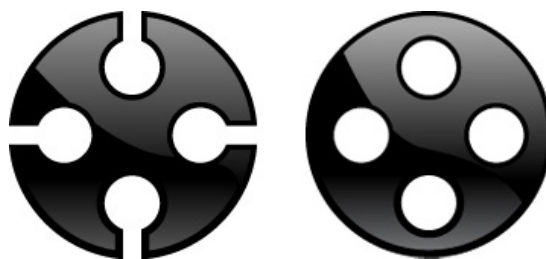
gebruikers in de taak die ze op dat moment uit te voeren hebben. Dit heeft tot gevolg dat in de grafische representatie van primitieven uit het QR vocabulaire in Garp3, dat hieronder beschreven is, rekening is gehouden met de verschillen in grafische principes die ten grondslag liggen aan de weergave in de context van modelbouw of model simulatie en inspectie.

### 5.1.1 Model type, klasse en instantie

Vanwege de domeinonafhankelijke natuur van Garp3 is het niet mogelijk visualisaties te maken van verschillende klassen van objecten of specifieke instanties. Machado (Machado, 2004) illustreert dat visualisatie alleen mogelijk is voor de modelingrediënten vanwege het feit dat ze de enige, volledig bepaalde set elementen van de simulatieomgeving vormen. De klassen en instanties variëren afhankelijk van het domein, model en modelleerkeuzes die gemaakt zijn door de maker van het model. Om die reden bestaan de visualisaties van de modelingrediënten enkel uit de visualisatie van de verschillende typen elementen die voorkomen en worden tekstelementen gebruikt voor de weergave van klassen en instanties. Dit neemt niet weg dat het zinvol kan zijn visuele variatie aan te brengen tussen de drie manieren waarop modelingrediënten kunnen voorkomen. Door dit visuele verschil te maken, is het voor de gebruiker van de software als geheel inzichtelijker welke rol het element speelt. In Garp3 zijn veel verschillende editors tegelijk te openen waardoor bij onervaren gebruikers verwarring kan ontstaan over de rol van klassen in het geval van modelbouw en van instanties in het geval van simulatie.

### 5.1.2 Elementaire grafische objecten

Zoals eerder besproken representeren de entiteiten de, meestal fysieke, objecten uit het domein. In niet-QR termen is een entiteit 'iets' waarvan mensen achten dat het relevant is in de simulatie. In het geval van Homer is er sprake van een iconische weergave in de vorm van een arbitraire grafische representatie die vervangbaar is door een willekeurige andere vorm zolang deze niet in relatie staat tot andere primitieven uit de visualisatie. VisiGarp hanteert echter een andere visualisatie van entiteiten. Hierbij is geen sprake is van een icon maar een grafische stijl die ten grondslag ligt aan het gehele grafische vocabulaire van VisiGarp; een entiteit is van variabele grootte afhankelijk van de hoeveelheid kwantiteiten die het bevat en wordt altijd weergegeven als een rechthoek. Om deze reden is besloten een alternatief voor de grafische weergaven van Homer te ontwikkelen en deze te incorporeren in het VisiGarp raamwerk.



Figuur 25. Initieel ontwerp en uiteindelijk versie van entiteit

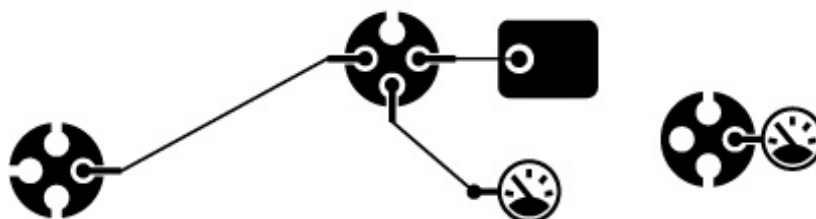
Voor de grafische representatie van een entiteit, ofwel 'iets', bestaat geen metafoor of conventie. Vanwege de domeinonafhankelijke grondslag van Garp3 is het niet mogelijk een invulling te geven aan wat het iets zou kunnen zijn. Er zijn een tweetal geschikte manieren voor de visualisatie: Ten eerste kan een geschikte vorm worden gezocht die de betekenis 'iets' kan overbrengen. Eerdere ontwerpen hiervoor, zie Figuur 26, bestonden uit een geabstraheerde kartonnen doos (tot een ruimtelijke kubus) omdat 'iets'

ongeacht de grootte of substantie in een doos gestopt kan worden. Deze visualisatie werd in een later stadium losgelaten vanwege het feit dat een kubus te veel veronderstelt dat entiteiten altijd fysiek zijn terwijl deze ook logische abstracties kunnen betreffen.



**Figuur 26. Visuele rationale voor alternatieve entiteit**

De alternatieve aanpak is een grafische vorm vinden die geen duidelijke relatie met het QR-primitief heeft, maar ingezet wordt op een manier waardoor andere elementen uit het zelfde visuele vocabulaire makkelijker te begrijpen zijn. Het impliceert dat dit element door nieuwe gebruikers geleerd moet worden maar dat daarmee het begrip van andere elementen sneller of beter verloopt. Hier is vorm aan gegeven door de grafische representatie te laten bestaan uit een cirkel waarin een viertal vormen zijn uitgespaard, zie Figuur 25. De reden hiervoor vormen de overige elementen in de samengestelde afbeelding van structurele relaties en de relatie tussen een kwantiteit en een entiteit. De uitsparingen bieden de mogelijkheid andere elementen te koppelen aan dit element als puzzelstukken of sleutel en slot.



**Figuur 27. Structurele relaties door koppelingen**

Ten eerste is op deze manier visueel duidelijk welke elementen aan elkaar gekoppeld kunnen worden aangezien ze voorzien zijn van verbindingstukken die passen in de uitsparingen. Ten tweede valt visueel af te leiden wat de mogelijke relaties zijn; aan een entiteit kunnen meerdere dingen worden gekoppeld en een relatie bestaat altijd tussen twee entiteiten. Een attribuut en een kwantiteit kunnen maar aan één enkele entiteit worden gekoppeld. Zie voor voorbeelden hiervan Figuur 27.



**Figuur 28. Ongewenste associaties van grafische representatie**

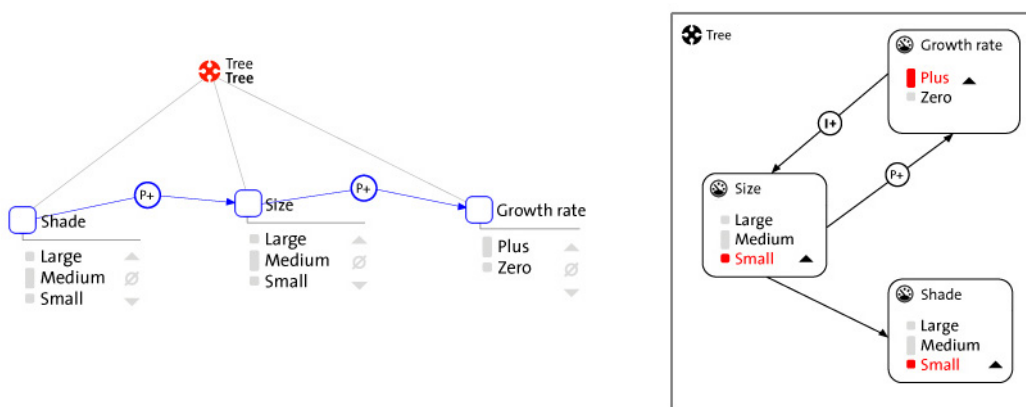
In de uiteindelijke implementatie van Garp3 zijn deze koppelingen niet opgenomen om twee redenen. Ten eerste hebben de visuele koppelingen tot gevolg dat elke entiteit meerdere verschijningsvormen heeft: zonder enige koppeling of met één of meer koppelingen. De toegevoegde waarde van de koppelingen weegt niet op tegen de implementatie van het exponentieel toenemen van het aantal verschillende verschijningsvormen van de entiteit. Daarnaast is er een culturele overweging om de

representatie niet te gebruiken in bovenstaande vorm. Ondanks dat de vorm niet geheel overeenkomt met een Keltisch kruis en het 45 graden geroteerd is, ontstaat toch de associatie hiermee, zie Figuur 28. Behalve dat het een domein impliceert dat niet relevant is voor het QR vocabulaire is het een associatie die gezien het historische gebruik van het symbool geheel ongewenst is.



Figuur 29. Kwantiteit

Een kwantiteit, zie Figuur 29, is de representatie van een hoeveelheid. De invulling van deze hoeveelheid wordt gegeven in de vorm van de mogelijke kwantiteitwaarden. Een natuurlijke metafoer is de analoge meter zoals deze ook in Homer wordt gehanteerd. Een alternatieve visualisatie die bijdraagt aan de consistentie van Garp3 als geheel is het hanteren van de grafische stijl vanuit VisiGarp. Hierdoor zou de kwantiteit als icon worden weergegeven in de vorm van een vierkant met afgeronde hoeken, zie voor een voorbeeld de linker zijde Figuur 30. Ondanks de toename in consistentie is deze aanpak niet gekozen vanwege het gebrek aan zeggingskracht van het afzonderlijke pictogram. Onervaren gebruikers krijgen in hun contact met Garp3 snel te maken met een kwantiteit in de modelbouwomgeving en zijn dan niet in staat er betekenis aan te ontlelen. Een alternatief voor een uniforme aanduiding van kwantiteiten is het opnemen van een visueel geheugensteuntje aan de grafische representatie van de elementen in de simulatie context. Dit is echter niet consistent met de grafische principes van dat vocabulaire als geheel omdat de containers de representatie *zijn* van de QR-primitieven waardoor dubbel wordt aangegeven van welk type de elementen zijn. Deze grafische toevoeging is voor onervaren gebruikers eventueel handig maar voor alle andere gebruikers redundant.



Figuur 30. Links kwantiteit als vierkant met afgeronde hoeken, rechts entiteit en kwantiteiten met type aanduiding





Figuur 31. Afgeleide

De grafische representatie van een afgeleide komt overeen met de kwantiteit, zie Figuur 31. Hierbij is er niet zozeer sprake van een mogelijke waarde maar betreft het de richting van de ontwikkeling van de waarde. Voor de visualisatie is dezelfde meter metafoor gehanteerd maar met een andere invulling van de meter zelf. De indicatie van de waarde staat verticaal en middels pijlen wordt aangegeven dat het de afgeleide betreft, hiermee wordt duidelijk dat de afgeleide kan variëren tussen dalend, constant en stijgend. In Garp3 wordt invulling gegeven aan een kwantiteit door de kwantiteitruimte en de afgeleideruimte. Deze laatste is altijd positief, nul of negatief. Vanwege het feit dat de kwantiteitruimte alleen wordt aangeduid door de naam is besloten geen grafische representatie te gebruiken voor de afgeleide zodat deze visueel in overeenstemming blijven. Er is geen reden meer 'inkt' te gebruiken voor dit ingrediënt zonder dat hier additionele informatie wordt aangeboden. De afgeleidenruimte wordt nu tekstueel weergegeven in de vorm van de tekstuele weergave van de 'delta' ofwel '□'.



Figuur 32. Waardenruimte

Elke kwantiteit heeft een waardenruimte waarmee de mogelijke waarden worden gerepresenteerd die de kwantiteit kan aannemen. Deze waardenruimtes worden samengesteld door de maker van het model. Daarnaast bevat elke afgeleide een standaard waardenruimte die altijd dezelfde waarden bevat; negatief, nul en positief. Omdat deze niet varieert, worden de onderdelen niet voorzien van een tekstlabel en hebben een eigen grafische representatie, specifiek voor de afgeleide. De ruimtes voor de kwantiteiten zijn geordende reeksen van punten en intervallen. Deze bouwstenen uit het QR vocabulaire worden weergegeven als een vierkant met afgeronde hoeken en altijd voorzien van een tekstuele label.

In de grafische representatie voor waardenruimte als pictogram, zoals afgebeeld in Figuur 32, wordt een voorbeeld gebruikt van twee punten en een interval. De reden hiervoor is dat op deze manier een archetypische waardenruimte ontstaat die niet te verwarren is met een 'i' of 'l', wat wel het geval is bij een enkele punt in combinatie met een interval. De volledige representaties van de kwantiteitruimte bestaan uit de tekstuele weergave van de naam en de elementen waaruit de waardenruimte bestaat, dus een nieuwe grafisch element voor dit geheel heeft tot gevolg dat het vocabulaire onnodig uitgebreid wordt.

In Figuur 33 staan twee toepassingen van de waardenruimte weergegeven. In het eerste geval in de simulatie context en in het tweede geval in de context van de modelbouw. Hierbij valt op te merken dat er enige verschillen zijn. In de modelbouw omgeving wordt de naam van de waardenruimte

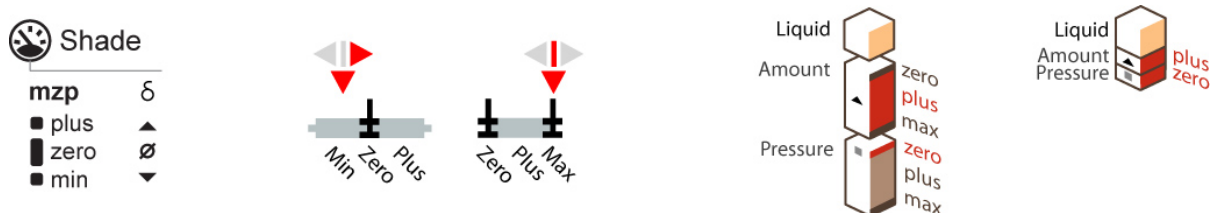


weergegeven. In deze context is dat zinvol omdat de maker van het model kan besluiten een andere waardenruimte te hanteren voor deze kwantiteit, terwijl de gebruiker in het eerste geval die keuze niet heeft wat het label overbodig maakt. In de tweede toepassing zijn de waardenruimte en afgeleide los van elkaar weergegeven en met de kwantiteit verbonden door een 'standaard relatie', terwijl ze onlosmakelijk verbonden zijn in de simulatie context.



**Figuur 33. Twee verschillende representaties van waardenruimte**

In alternatieve representaties uit Figuur 34 is overwogen om de waardenruimte op dezelfde manier naast de afgeleide weer te geven. Deze aanpak is niet voortgezet vanwege het feit dat de onderdelen van de waardenruimte en afgeleide vaak verbonden zijn met verschillende andere onderdelen in de vorm van ongelijkheden en correspondenties. Als het dan niet mogelijk is deze elementen verder uit elkaar te spatieren ontstaat onnodige opeenhoping van elementen. Verder zijn er alternatieven gebaseerd op andere manier van visualiseren van de waardenruimte waaruit naar voren komt dat een horizontale organisatie als voordeel heeft dat het een natuurlijke representatie is van een toenemende schaal (Machado, 2004) overeenkomstig de cartesische weergave van een vlak. Daar tegenover staat het nadeel dat de tekstlabels nooit recht onder de elementen kunnen worden weergegeven omdat deze variabel zijn in tekstlengte.



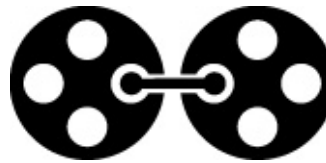
**Figuur 34. Alternatieve representaties voor waardenruimte**

Het derde alternatief is een weergave waarbij elke kwantiteit wordt weergegeven als element onder de entiteit waartoe deze behoort. In deze pseudo 3d weergave wordt de linkerzijde gebruikt voor de weergave van de afgeleide en de andere zijde voor de waardenruimte. Het belangrijkste kenmerk van deze visualisatie is dat deze alleen in uitgeklapte versie voorkomt als de entiteit of één van de kwantiteiten is geselecteerd. In alle andere gevallen wordt de minimale weergave gebruikt waarin alleen zichtbaar is wat de huidige waarde is in de waardenruimte en de afgeleide. Ondanks dat een minimale weergave veel scherm opeenhoping voorkomt, biedt het niet de mogelijkheid relaties inzichtelijk te maken doordat deze teveel opeenhopen aan het element zelf. Relaties hebben ruimte nodig zodat visueel kan worden onderscheiden waarmee deze precies een relatie vormen.



Figuur 35. Attribuut

Een attribuut is een representatie van een structurele eigenschap van een entiteit, zie Figuur 35. Het bestaat uit een attribuut-type en -waarde. De visuele representatie bestaat uit een bagagelabel in het systeem van gekoppelde structurele elementen. In deze initiële visuele opzet bestond de grafische weergave enkel uit het bagagelabel. In de uiteindelijke implementatie zijn de gekoppelde elementen niet doorgevoerd, wat tot gevolg heeft dat de representatie bestaat uit het volledige pictogram zoals hier is aangegeven. Hierdoor is aan de visualisatie af te leiden dat het alleen bij entiteiten mogelijk is een attribuut toe te voegen.



Figuur 36. Structurele relatie

Een structurele relatie, Figuur 36, kan alleen bestaan tussen twee entiteiten. Dat neemt niet weg dat een entiteit onderdeel kan zijn van een grotere constellatie van structurele relaties. Voor de implementatie van dit element geldt hetzelfde als bij het hierboven beschreven attribuut: doordat de grafische representatie van de entiteit is opgenomen in de weergave van de relatie is duidelijk dat alleen dit type element gerelateerd kan worden.



Figuur 37. Aanname

De 'assumption', of aanname is een element in Garp3 waarmee het gedrag van een model beperkt wordt. Het maakt geen onderdeel uit van structurele onderdelen noch van gedragsonderdelen van het model. Door het model te beperken is het mogelijk een specifieke richting of sturing aan de simulatie te geven. Het is een beperking van het gehele gedrag tot een deel van het gedrag dat relevant wordt geacht. Om deze reden is de afbeelding van de vuurtoren in de grafische representatie verwerkt, zie Figuur 37. De vuurtoren staat metafoor voor het beperkte zicht dat mogelijk is in het licht van de vuurtoren. Daarnaast kan de baan elke kant op schijnen en is het altijd een bewuste keuze om een bepaalde richting op te schijnen.



Figuur 38. Alternatieven voor het modelingrediënt 'aanname'

Alternatieve visualisaties voor het 'assumption' ingrediënt in Figuur 38 bestonden uit een abstracte afbeelding van een vizier en zaklamp. Een vizier is de beste metafoor voor het ingrediënt omdat in de andere twee varianten dingen zogenaamd blijven onbelicht blijven, terwijl dat in de simulatie niet het geval is. Het pictogram is niet gebruikt vanwege het gebrek aan herkenbaarheid van het vizier. De zaklantaarn is afgefallen omdat het op de kleine schaal van het computerscherm waarop het gebruikt wordt niet herkenbaar meer is als zaklamp.



Figuur 39. Agent

Een agent is de representatie van een entiteit buiten het gemodelleerde systeem. Net als entiteiten kunnen agents kwantiteiten bevatten die van invloed zijn op de rest van het systeem. Zoals te zien in Figuur 39 is aan de hand van een pijl die een zwarte figuur binnendringt vorm gegeven aan dit type ingrediënt. Vanwege de speciale status van bovenstaande twee elementen zijn ze van overeenkomstige vorm. In beide gevallen is er sprake van een invloed op de simulatie die van buiten het model zelf komt.



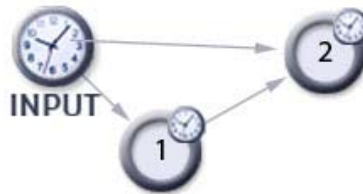
Figuur 40. Toestand

Een toestand beschrijft een specifieke situatie van het gemodelleerde systeem. Vanwege het domeinonafhankelijke karakter is gekozen voor de abstracte figuur van de voorkant van een fotografische lens, zie Figuur 40. In eerdere varianten is het aantal toepasbare modelfragmenten als waaier om de toestand weergegeven om een verschil aan gewicht aan te geven tussen verschillende toestanden. Een toestand waar veel modelfragmenten toepasbaar zijn, is potentieel interessanter vanwege een grotere verandering in gedrag of structuur. Om de volgende redenen werd deze aanpak niet voortgezet: Ten eerste is het mogelijk dat er meer modelfragmenten van toepassing zijn dan er aan visuele elementen om de toestand heen passen waardoor elementen over elkaar heen komen te liggen. Ten tweede is het een grafische representatie van een modelfragment dat niet consistent is met de andere verschijningsvorm. Ten derde viel de implementatie buiten de werkzaamheden voor een functioneel Garp3.



Figuur 41. Verschijningsvormen van toestanden

Zoals weergegeven in Figuur 41 komen toestanden op vier verschillende manieren voor, afhankelijk van de staat van de simulatie van die specifieke toestand. Het viertal gekozen kleuren lijkt te werken als kleurverloop waarbij de kleuren in de ontwikkeling van een nieuwe toestand steeds donkerder worden. Ondanks dat de gekozen kleuren als een verloop ogen, van licht naar donker, zijn het vier verschillende kleuren zodat ze goed te differentiëren zijn van elkaar.



Figuur 42. Alternatieve representatie voor scenario en toestanden

Een alternatief voor de representatie van toestanden is gebaseerd op een analogie met klokken (Figuur 42). Een toestand bestaat op een bepaald tijdstip, voor of na een ander tijdstip. Een scenario wordt gerepresenteerd als een volledige klok omdat in een simulatie het scenario het 'begin van de tijd' is. Vanwege de grote stijlbreuk met de rest van het visuele vocabulaire en het gevaar dat gebruikers menen dat de tijd op de klokken overeenkomt met de tijd van de toestand, is deze representatie niet voortgezet.



Figuur 43. Modelfragment

Een modelfragment is een generiek stukje kennis opgebouwd uit verschillende modelingrediënten. Om deze reden bestaat de visualisatie uit een map of container waaruit een aantal modelingrediënten steken. Op deze wijze wordt aangegeven dat een modelfragment in staat is andere elementen te bevatten. In dit elementaire grafische object worden andere elementaire grafische objecten gebruikt, zie Figuur 43.



Figuur 44. Scenario

Een scenario is een gedeeltelijke initiële situatie beschrijving. Het komt overeen met het modelfragment behalve dat het geen conditioneel deel bevat. Zoals te zien in Figuur 44 bestaat de visualisatie uit dezelfde

container als in het geval van het modelfragment en de inhoud bestaat uit een grafische representatie van een toestand.

## 5.2 Relaties

Voor het weergeven van relaties bestaan een aantal grafische randvoorwaarden voor het QR domein (Machado, 2004) aan de hand waarvan de weergave tot een minimum beperkt kan worden.

Ten eerste is het formeel onnodig een grafische weergave te maken van elementen uit een lijst als duidelijk is gemaakt dat de lijst van elementen van hetzelfde modeltype zijn. Een aanduiding van het type volstaat voor de lijst als geheel en voor de lijst volstaat de naam van de elementen. Ten tweede zijn veel relaties van een bepaald type, zonder dat hiervoor een klasse hoeft te bestaan of instanties worden gemaakt. Elk van deze relaties is uniek in combinatie met de elementen waartussen de relatie bestaat. Ondanks dat het formeel mogelijk is, is het alleen relevant als er maar één relatie bestaat van een bepaald type tussen twee elementen: In Garp3 is het niet zinvol meerdere keren dezelfde ongelijkheid, bijvoorbeeld '>' toe te voegen tussen twee kwantiteiten. Het is dan niet nodig het element van een naam te voorzien, om het element uniek identificeerbaar te maken, en volstaat de naam of visualisatie van het modeltype. In het geval van structurele relaties geldt hetzelfde voor de instantie naam. Onderdeel van het model is de volledige set mogelijke structurele relaties die kunnen bestaan tussen entiteiten. Deze aanduiding van het type structurele relatie in combinatie met de gecombineerde entiteiten is voldoende voor de weergave van deze relatie. Ondanks dat meerdere structurele relaties kunnen bestaan tussen dezelfde entiteiten is het niet relevant om dezelfde relatie meerdere keren aan te geven waardoor het niet nodig is de relaties uniek te benoemen om ze van elkaar te onderscheiden.

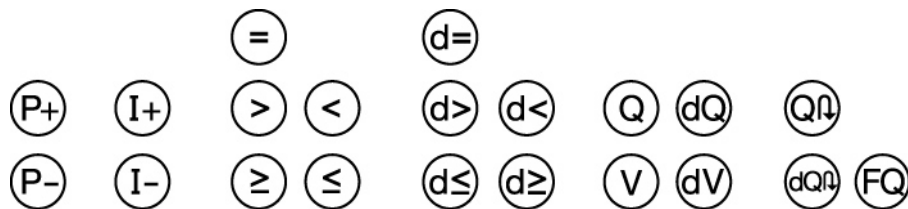
Naast optimalisaties voor de weergave van relatietypen bij een relatie bestaan er ook gevallen waarbij dit type helemaal weggelaten wordt. Machado differentieert tussen impliciete en expliciete relaties. In het geval van het laatste bestaan er primitieven in het QR vocabulaire om de relatie te duiden, hiervan bestaat vervolgens een grafische representatie in het visuele vocabulaire. Daarnaast bestaan er relaties die impliciet zijn, er bestaat in dat geval geen modelingrediënt om de relatie mee aan te duiden. Een voorbeeld hiervan is de relatie tussen een kwantiteit die toebehoort aan een entiteit zonder dat er een relatietype in het vocabulaire voor bestaat. Dat neemt niet weg dat er een visuele representatie van de relatie moet zijn om de relatie weer te kunnen geven. Hiervoor wordt standaard een grijze of zwarte lijn zonder tekstueel label gehanteerd. Een uitzondering is de manier waarop VisiGarp de impliciete relatie visualiseert tussen entiteiten en kwantiteiten. Deze relatie wordt op basis van een 'container' metafoor weergegeven: de entiteit bevindt zich altijd in de entiteit waardoor geen expliciete grafische representatie van de relatie nodig is zoals in de modelbouw omgeving.

Ondanks dat er in het QR vocabulaire een primitief bestaat om de relatie tussen de entiteiten te beschrijven in de isa-hiërarchie wordt er geen type aanduiding gehanteerd in de visualisatie hiervan. De reden is dat van de relatie verondersteld wordt dat de onderlinge relatie duidelijk is en tot de standaard kennis van de gebruiker behoort. Voor de definitie van impliciete relaties is het goed mogelijk dat het niet relevant is of er een primitief voor bestaat in het QR vocabulaire. In elke visualisatie van één of meer typen relaties kan één relatie, mits verondersteld kan worden dat de relatie tot de standaard kennis van de gebruiker behoort, worden weergegeven als een impliciete relatie. Hierdoor zijn veel relaties spaarzaam op het scherm weer te geven en ontstaat een goede visuele hiërarchie tussen verschillende relaties.

### 5.2.1 Relaties voor gedrag

Tussen modelingrediënten kunnen verschillende soorten relaties bestaan in eenzelfde weergave. Ten eerste bestaan er overzichten in de vorm van ongerelateerde elementen van het zelfde type, veelal in een lijst. In dit geval kan de visualisatie bestaan uit een aanduiding van het modeltype en een lijst tekstuele labels, een icon voor de afzonderlijke elementen in de lijst kan weggelaten worden om ruimte op het scherm te besparen. Een extensie hiervan is het overzicht van elementen in een hiërarchisch overzicht. In het geval van modeltypes en modelfragmenten bestaat het overzicht uit alle elementen van dat type en de onderlinge hiërarchische relatie tussen deze elementen.

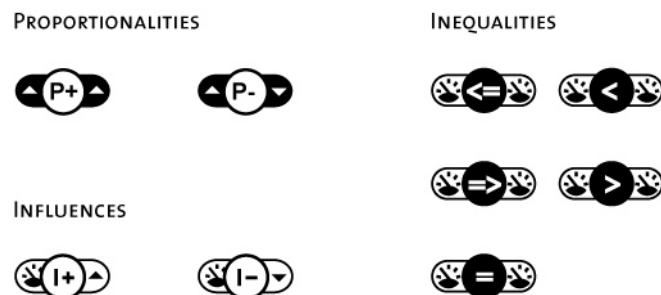
Ten derde kunnen er relaties bestaan tussen modelingrediënten die onderdeel zijn van de gedrags- of structuurbeschrijving van het gemodelleerde systeem. In het geval van structurele relaties betreft het de relaties tussen entiteiten waarmee veelal de fysieke structuur van het systeem wordt beschreven. In het geval van gedrag bestaan de relaties uit mathematische en causale afhankelijkheden tussen kwantiteiten. Voor deze twee typen relaties bestaan primitieven in het QR vocabulaire die onderdeel zullen zijn van de grafische representatie van Garp3.



Figuur 45. Relaties in Garp3

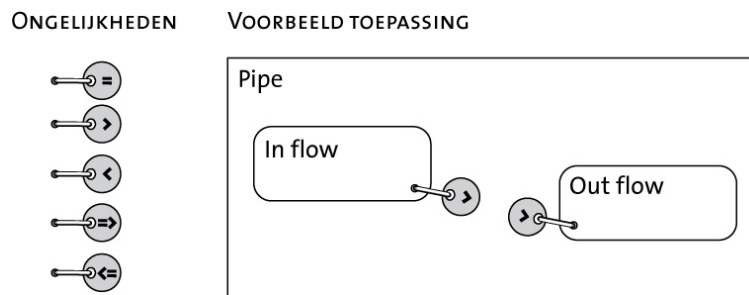
Eén van de doorslaggevende criteria voor de grafische representatie van relaties is spaarzaam visualiseren. De causale en mathematische relaties komen veelvuldig samen voor in hetzelfde scherm dus is het noodzakelijk een visualisatie te hanteren waarbij het mogelijk is veel relaties weer te geven. Deze hoeveelheid moet niet beperkt worden door technische beperkingen in termen van kleur of pixel afmetingen enerzijds en visueel/cognitieve beperkingen in termen van scherm clutter. Een spaarzame visualisatie maakt het mogelijk dat alle informatie in hetzelfde scherm is weer te geven.

In plaats van een aparte visualisatie te ontwikkelen voor elk van de verschillende typen is voor één visualisatie gekozen voor alle typen, zie Figuur 45. Op deze manier is visueel altijd duidelijk dat het een gedragsrelatie betreft en is het aan de lezer van de visualisatie de afzonderlijke relaties te interpreteren. Deze manier van 'high density design' heeft tot gevolg dat meer elementen op het scherm mogelijk zijn maar heeft tevens tot gevolg dat de interpretatietask van de kijker een meer cognitieve taak wordt.



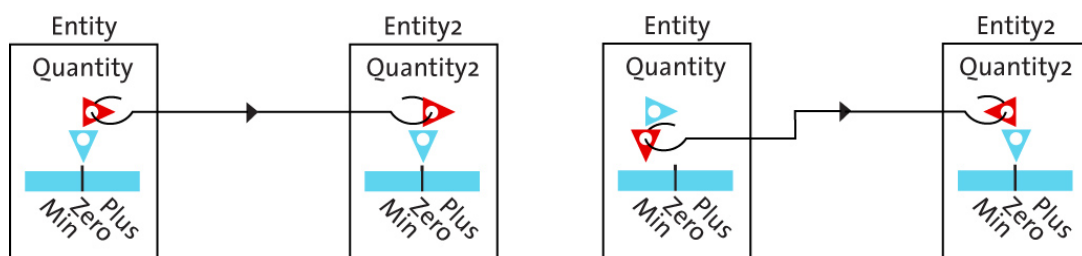
Figuur 46. Alternatieve representatie van relaties

Voor de grafische representatie van relaties zijn een aantal verschillende varianten ontwikkeld. In de versie in Figuur 46 is getracht visueel het gedrag van de relatie weer te geven. Op deze manier ontstaat voor onervaren gebruikers een visuele legenda voor de relatie waarvoor het element wordt ingezet. In het geval van een positieve proportionaliteit is zichtbaar dat het om de afgeleide gaat waarbij beide stijgend zijn. Visueel wordt weergegeven dat een invloed bestaat tussen een kwantiteit en een afgeleide en mathematische afhankelijkheden bestaan altijd tussen twee kwantiteiten. Een aantal redenen geven aan waarom deze representatie geen gevolg heeft gehad. De primitieven uit het QR domein moeten geleerd worden alvorens ze zijn in te zetten in een model. In dat geval is het in principe onnodig een visuele geheugensteun te gebruiken voor deze elementen. Ten tweede is de vraag of het gerechtvaardigd is om meer visuele elementen toe te voegen voor een specifieke gebruikersgroep, in dit geval gebruikers die niet of nauwelijks thuis zijn in het QR domein. Tenslotte impliceren de invloeden een bepaalde leesrichting terwijl de onderdelen van de relatie op het scherm zijn georganiseerd op een manier die hiermee niet overeen hoeft te komen.



Figuur 47. Alternatie voor ongelijkheden

Een exoot in de grafische representatie van ongelijkheden bestaat uit de relatie weergegeven als twee afzonderlijke grafische elementen aan de elementen die de relatie verbindt. Zoals zichtbaar in het voorbeeld aan de rechterkant van Figuur 47 bestaat de relatie niet uit een lijn maar wordt duidelijk door de richting van de grafische elementen. Op deze manier wordt bespaard op lijnen die het scherm doorkruisen. Een belangrijke voorwaarde voor deze representatie is de interactiviteit van de elementen: de ongelijkheden moeten gericht blijven langs een onzichtbare lijn op elkaar, ook als de kwantiteiten over het scherm worden herschikt door de gebruiker. Deze interactie maakt het namelijk mogelijk in een simpele handeling te zien tussen welke kwantiteiten de relatie geldt. Vooral wanneer meerdere ongelijkheden bestaan is het middels interactie mogelijk inzicht te krijgen in de relaties.



Figuur 48. Ketting weergave van causale relaties

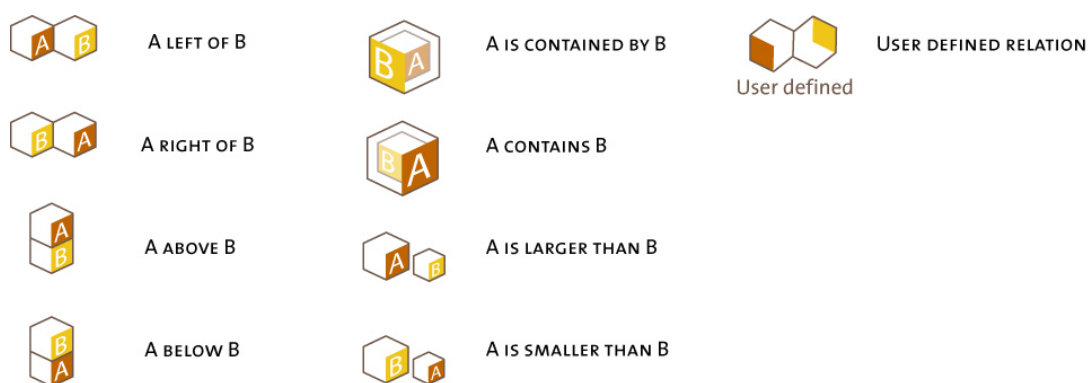
Voor causale relaties is een grafische representatie gemaakt waarbij interactie een zelfde rol speelt. In Figuur 48 staan voorbeelden van de proportionaliteit en invloed afgebeeld tussen verschillende kwantiteiten. De relaties bestaan uit haken tussen elementen uit de horizontaal georganiseerde



kwantiteitruimtes: in het geval van een proportionaliteit tussen de twee afgeleiden en in het geval van de invloed tussen de waarde en afgeleide. Door interactie met de haken wordt het gedrag van de causale relaties duidelijk. In het geval van de onafhankelijke kwantiteit is zichtbaar dat het variëren in waarde of afgeleide gevolgen heeft bij de afhankelijke kwantiteit. Een geheel causaal pad van meerdere causale relaties verandert visueel door de te interacteren met de waarde of afgeleide in het begin van deze keten. Zonder enige kennis van het QR vocabulaire is duidelijk dat verandering in het begin van de keten gevolgen heeft aan het einde van de keten.

### 5.2.2 Structurele relaties

Zoals eerder besproken worden structurele relaties in Garp3 weergegeven als een lijn zonder aanduiding van type of instantie maar enkel door de klasse naam.



Figuur 49. Grafische representatie van meest voorkomende structurele relaties

In het geval van structurele relaties is een poging ondernomen om visueel invulling te geven aan veel gebruikte relaties, zie Figuur 49. Ondanks dat het in de modelbouw omgeving mogelijk is oneindig veel verschillende structurele relaties aan te maken, betreft het in de meeste modellen een vrij beperkte set van invullingen. Door hergebruik van model elementen door modelbouwers blijft de groep structurele relaties beperkt. Vanuit die gedachte is het mogelijk een set van structurele relaties grafisch te representeren en deze vervolgens te gebruiken in de simulatie omgeving. Deze set is niet bedoeld als exclusieve verzameling maar voorziet de meeste modelbouwers van een adequaat vocabulaire. Deze representatie is alleen mogelijk indien er een 'catch all' representatie is voor structurele relaties die geen deel uitmaakt van de standaard set.

### 5.2.3 Overige relaties

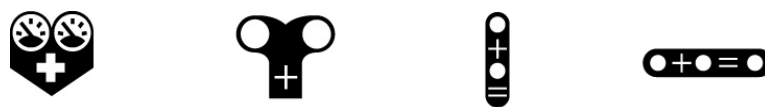
Onderdeel van het QR vocabulaire van Garp3 is de 'math operator' waarmee kwalitatieve mathematische operaties kunnen worden uitgevoerd. Dit element bevat drie relaties. Enerzijds de twee relaties met de twee kwantiteitwaarden die de input of 'left hand side' van de vergelijking vormen en waarop de operator wordt uitgevoerd, anderzijds de relatie met de kwantiteitwaarde waaraan het resultaat wordt toegewezen. De grafische representatie bestaat uit twee balken voorzien van uitsparingen, zie Figuur 50. De buitenste balk bevat het verbindingspunt van de resultaatrelatie en door de binnenste balk volledig te omsluiten wordt visueel duidelijk dat het hier het resultaat betreft. Om die reden is het onnodig een '=' teken op te nemen in de visualisatie. Voor de verbindingen van de relaties is het van belang dat deze volgens de normale leesrichting, horizontaal worden weergegeven zodat een visuele overeenkomst ontstaat met de tekstuele manier van mathematische operaties weergegeven.





Figuur 50. Ingrediënt voor mathematische operaties in verschillende verschijningsvormen

In drie van de alternatieven in Figuur 51 voor de mathematische operatie is deze overeenkomst niet doorgevoerd waardoor minder duidelijk is wat deze representeert. Ondanks dat in de eerste variant duidelijk is dat er kwantiteiten mee verbonden moeten worden, is niet duidelijk dat er een uitkomst is. De vierde variant komt zeer overeen met de uiteindelijke variant maar door de container metafoor te gebruiken is niet alleen het '=' teken overbodig maar is het element ook goed kleur te coderen voor de verschillen in context waarin het voorkomt.

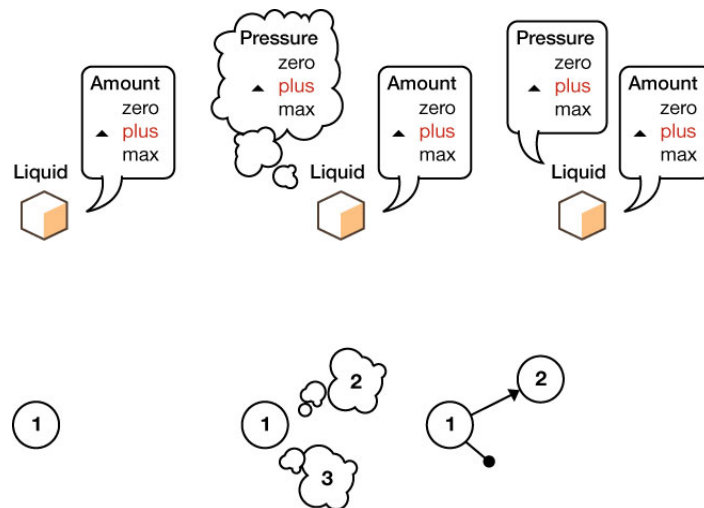


Figuur 51. Alternatieven voor de mathematische operatie

### 5.3 Afgevalen alternatieven

Tijdens de ontwikkeling van het visuele vocabulaire zijn een drietal visualisaties ontstaan die apart zijn opgenomen omdat ze zeer afwijkend zijn van de opbouw van de componenten van Garp3. Ten eerste een visualisatie van elementen waarin onzekerheid een rol speelt, ten tweede een visualisatie gebaseerd op vloeistoffen en stromen hiervan tussen vaten en tenslotte een meer diagrammatische weergave.

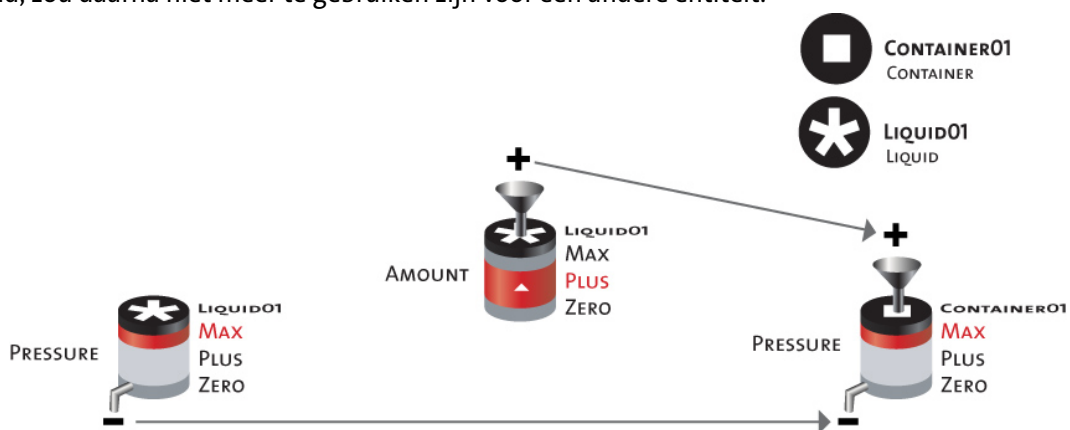
Er bestaat een zekere overeenkomst tussen het resultaat gedeelte van modelfragmenten en de ontwikkelingen van toestanden. In beide gevallen zijn er elementen waarvan onzeker is of ze ook werkelijk zullen gaan bestaan. Een modelfragment kan niet van toepassing zijn en een terminatie kan niet resulteren in een vervolg toestand. Een mogelijke analogie in dit geval is het visuele verschil tussen 'denken' en 'zeggen' waarbij de gedachte vooraf gaat aan het uiteindelijke zeggen. Op deze wijze wordt visueel het verschil duidelijk tussen elementen die aanwezig zijn en elementen die mogelijk aanwezig zijn. De grafische representatie bestaat uit cartooneske spraak- en gedachteballonnen waarvan voorbeelden staan in Figuur 52. Om twee redenen is deze aanpak niet voortgezet. Ten eerste is de grafische representatie niet erg spaarzaam en biedt daarmee niet de mogelijkheid veel verschillende elementen op het scherm te zetten. Ten tweede zijn de vormen van de ballonnen moeilijk schaalbaar te implementeren zodat elementen van variabele grootte op dezelfde manier van een kader worden voorzien.



Figuur 52. Cartoon-achtige weergave van mogelijke ontwikkelingen

De tweede afwijkende visualisatie, zie Figuur 53, bestaat uit een visuele representatie van kwantiteiten in de vorm van vaten. Elk vat is voorzien van een verticale weergave van de waardenruimte. Afhankelijk van hoe vol het vat is, wordt de bijbehorende waarde uit de waardenruimte actief. De entiteiten worden plat weergegeven, in tegenstelling tot de kwantiteiten die in een zogenaamd 2,5d of pseudo perspectief worden weergegeven. Causale relaties worden visueel weergegeven als lijnen tussen de bovenkant (positieve invloed) of de onderkant van de kwantiteit (negatieve invloed) met de gedachte dat er vloeistof aan het vat wordt toegevoegd of er uitgehaald.

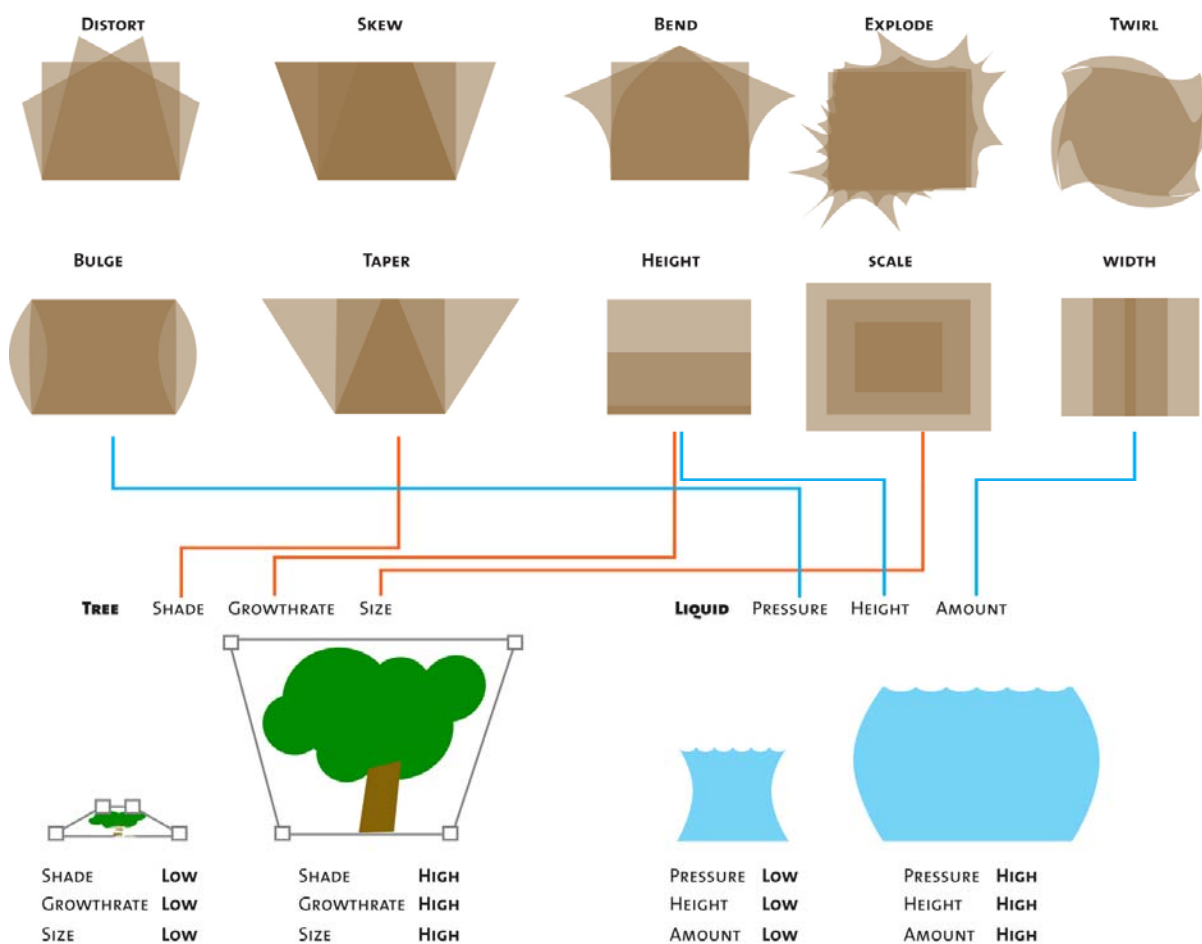
Elk vat heeft een deksel met daarop weergegeven het pictogram van de entiteit. Ondanks dat de visuele representatie domeinonafhankelijk is, is het mogelijk een set pictogrammen te ontwikkelen die als enig doel hebben de elementen visueel van elkaar te kunnen differentiëren. Dit komt overeen met de weergave van de structurele relaties uit Figuur 49. Dit zou betekenen dat elke entiteit in de isa-hiërarchie niet alleen een naam heeft en een plaats in de hiërarchie maar ook wordt voorzien van één van de arbitraire figuren als visueel herkenningsteken. Elk element dat uit de bibliotheek van figuren wordt gehaald, zou daarna niet meer te gebruiken zijn voor een andere entiteit.



Figuur 53. Representatie van kwantiteiten gebaseerd op de 'volheid' van waardenruimte

De derde afwijkende visualisatie is gebaseerd op het gebruik van afbeeldingen voor entiteiten uit de isa-hiërarchie en het cumulatief vervormen van de grafische representatie van de entiteit. Zoals schematisch in het voorbeeld in Figuur 54 is weergegeven is het mogelijk een serie vervormingen te bedenken die door de modelbouwer aan een kwantiteit worden toegewezen. Elke waarde in de waardenruimte komt

overeen met bepaalde mate van vervorming. Vervolgens worden entiteiten voorzien van kwantiteiten en worden de toegewezen vervormingen toegepast op de representatie van de entiteit. Door de visuele overeenkomst ontstaat een meer diagrammatische weergave tussen de representatie en het gerepresenteerde. Een aantal problemen doen zich voor bij deze aanpak die opgelost moeten worden voordat deze aanpak werkelijk bruikbaar is. Ten eerste moet een oplossing worden gevonden voor het nulpunt van de waardenruimte. Indien een kwantiteit de waarde 'nul' heeft en toegewezen is aan de breedte van een entiteit, dan wordt deze gevisualiseerd als een entiteit zonder breedte. Op dat moment is de entiteit niet meer zichtbaar en is zeer de vraag in hoeverre dit voor gebruikers nog inzichtelijk is. Ten tweede moet duidelijk zijn of de verschillende kwantiteiten nog van elkaar te onderscheiden zijn in de cumulatieve visualisatie van de entiteit. Naast een bestaande bibliotheek van vervormingen moet het tenslotte mogelijk zijn voor gebruikers om eigen vervormingen te definiëren van deze elementen. Hiermee worden additionele operaties toegevoegd die geen onderdeel zijn van het QR vocabulaire die het modeleren eventueel nodeloos ingewikkeld maken.



Figuur 54. Diagrammatische representatie door 'entiteit vervorming'

## 5.4 Visualisatie operaties

De visualisatie van operaties is onder te verdelen naar vier verschillende soorten taken die uitgevoerd kunnen worden, en een restgroep. Als eerste de operaties die betrekking hebben op het model als geheel. Vervolgens zijn er operaties op modelingrediënten vanuit de modelbouwomgeving. Ten derde operaties

die betrekking hebben op de modelsimulatie omgeving van Garp3. Ten vierde operaties voor de organisatie van de interface. Dit betreft operaties voor de besturing van de layout van modelingrediënten of operaties op verschillende windows. Tenslotte operaties die niet tot één van deze groepen behoren en heel specifiek zijn voor het QR domein of juist heel algemeen.

Voor het visualiseren van operaties is zoveel mogelijk gebruik gemaakt van het ontwikkelde visuele vocabulaire van de QR-primitieven. Door dit hergebruik van deze primitieven in de verschillende onderdelen van het systeem zijn de operaties beter te begrijpen voor gebruikers die minder ervaren zijn in het QR vocabulaire. Het hergebruik bestaat uit een combinatie van een QR-primitief en een in 4.2 besproken modifier zodat de acties bestaan uit een combinatie van een onderwerp waarop de actie wordt uitgevoerd en een representatie van de actie.

## 5.5 Operaties op het model

Operaties die uitgevoerd kunnen worden op het model als geheel bestaan vooral uit operaties voor file management van het model: het openen, opslaan en sluiten van het model. Voor zover mogelijk zijn dezelfde modifiers gebruikt zoals deze gehanteerd worden op de modelingrediënten. Daar waar het nieuwe functionaliteit betreft zijn nieuwe modifiers toegevoegd: in het geval van ‘openen’ en ‘opslaan’ is gebruik gemaakt van een pijl die in het geval van ‘openen’ naar het model wijst. De operatie bestaat uit het inladen van gegevens van buiten de software in wat het huidige model is binnen de software. Het tegenovergestelde geldt voor het opslaan van een model waar de pijl buiten het model wijst; ter illustratie van het buiten de software plaatsen van het huidige model. In het geval van ‘opslaan als..’ wordt een additioneel grafische element getoond om te illustreren dat deze operatie iets extras nodig heeft boven het normale opslaan, namelijk de nieuwe naam voor het bestand.

Een vaak voorkomende operatie binnen Garp3 is het opslaan van veranderingen binnen het model. Dit lijkt overeen te komen met het opslaan van het bestand, maar dit is niet het geval. Een verandering wordt opgeslagen in het model maar komt te vervallen als het model later niet opgeslagen wordt. Vanwege deze reden is deze operatie vormgegeven als het pictogram van modelbouw, zie Figuur 21, waarbij het losse element in de blauwe bol van het model ‘schuift’. Deze subtiele grafische representatie maakt een alternatieve interpretatie van de operatie mogelijk. Het pictogram kan ook geïnterpreteerd worden als het terugkeren naar de modelbouw omgeving. In principe is dit correct, omdat alleen in de modelbouw omgeving veranderingen zijn aan te brengen aan een model die opgeslagen dienen te worden.



Figuur 55. Model operaties

Alternatieven voor de bestandsoperaties zijn weergegeven in Figuur 56. In deze alternatieven is getracht in de weergave van de operaties associaties te gebruiken met fysieke objecten, die te maken hebben met bestandsbeheer. In het eerste geval in de vorm van een ouderwetse floppy disk in combinatie met een holle weergave van een model als representatie van de huidige toestand waarin nog geen model aanwezig is. In het tweede alternatief is gebruik gemaakt van een grafische weergave van een doos als representatie van een harde schijf of computer. Beide alternatieven zijn minder geschikt dan de meer abstracte varianten vanwege het feit dat de associaties met de fysieke objecten ofwel gedateerd zijn, ofwel een onduidelijke associatie hebben en onnodig veel elementen bevatten.



Figuur 56. Alternatieven voor bestands operaties

## 5.6 Operaties op modelingrediënten

Naast het hergebruik van de QR-primitieven in combinatie met een modifier is een visueel primitief ontworpen dat als algemeen modelingrediënt dient. De grafische weergave bestaat uit het losse deel van het modelbouw pictogram uit Figuur 21, vanwege het feit dat het overeenkomt met het concept 'een deel van een model'. Dit primitief komt op dezelfde manier voor in de representatie van acties; zoveel mogelijk in combinatie met een modifier. Dit algemene element is ten eerste toepasbaar in situaties waarin de operatie mogelijk is op verschillende typen modelingrediënt. In dat geval volstaat een afbeelding van het primitief in combinatie met een modifier niet. Daarnaast is het overbodig om meerdere varianten van dezelfde operatie weer te geven voor elk type waarop de operatie mogelijk is; dit heeft een onnodige hoeveelheid knoppen tot gevolg. Verder is het algemene element bruikbaar in situaties waarbij uit de context is af te leiden op welk ingrediënt de operatie betrekking heeft. Deze context kan bestaan uit een window waarin maar een enkel type ingrediënt mogelijk is, zoals de isa-hiërarchie die alleen uit entiteiten kan bestaan. In dat geval is er geen alternatief type mogelijk waarop operaties uitgevoerd kunnen worden en volstaat het algemene element in combinatie met een modifier. Het voordeel van dit algemene element boven de modelingrediënten is dat algemene en vaak terugkerende operaties op een uniforme manier worden weergegeven zodat deze in variabele context kunnen terug komen terwijl de gebruiker ze maar één keer hoeft te begrijpen.



Figuur 57. Overzicht operaties op QR-primitieven

Voor modelingrediënten zijn een vijftal, vooral veel gebruikte, operaties ontwikkeld die algemeen toepasbaar zijn op alle ingrediënten (zie de bovenste rij modifiers uit Figuur 57): Ten eerste het maken van een nieuw element waarbij de representatie is gebaseerd op de conventie dat nieuwe dingen glimmen en is daarmee een representatie van het resultaat van de actie. De overige modifiers zijn respectievelijk: het verwijderen van een element door het visueel 'uit te gummen', de mogelijkheid tekstuele eigenschappen van een element aan te passen, een tekstuele weergave in het 'oude' Prolog vocabulaire, zogenaamde 'legacy', wordt gerepresenteerd door een perkamentrol of oude printer uitdraai en tenslotte het bewerken van een element aan de hand van een mechanisch tandwiel ter illustratie van de toegang die het verschaft tot de onderliggende machinerie van het element. In Figuur 58 is deze modifier weergegeven in combinatie met de modelingrediënten.



Figuur 58. Voorbeeld 'bewerken' modifier op QR-primitieven

Het eerste element uit de tweede rij van Figuur 57 is het pictogram voor 'kopiëren', bestaande uit een dubbele weergave van het element. De overige elementen zijn operaties die bestaan uit variaties van het algemene ingrediënt in combinatie met modifiers.

## 5.7 Operaties op model views

De functionaliteiten van Garp3 hebben niet alleen betrekking op modelingrediënten of het model als geheel maar bestaan ook uit operaties op onderdelen van de interface van de software. De weergave bestaat uit een grijze ruit als representatie van een window in combinatie met een modifier, zodat visueel duidelijk is dat het altijd operaties op windows betreffen. Voor een deel wordt hierin gebruik gemaakt van dezelfde modifiers uit Figuur 57. Tevens bestaan er functionaliteiten die betrekking hebben op de organisatie van elementen binnen deze windows, weergegeven als een zelfde grijze ruit met een raster. De modifiers die hierbij gebruikt worden, zijn modelingrediënten die als gevolg op een andere manier in het window worden georganiseerd.

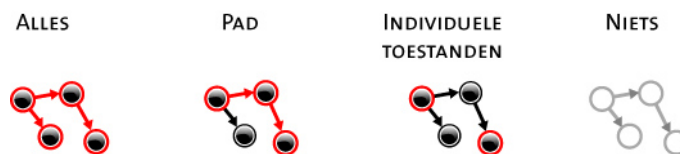


Figuur 59. Operaties op interface

## 5.8 Operaties voor simulatie

In de simulatie omgeving van Garp3 bestaat de interface uit een hoofdscherm voor de besturing van de simulatie en de verschillende manieren van inspectie van onderdelen daarvan. In dit scherm worden tevens alle toestanden weergegeven in de zogenaamde state graph. De operaties die kunnen worden uitgevoerd zijn onderverdeeld naar drie groepen: Ten eerste operaties die betrekking hebben op het selecteren van toestanden en paden van toestanden. Ten tweede de inspectie operaties waarin verschillende facetten van de toestanden worden weergegeven. Tenslotte bestaan er operaties voor het uitvoeren van verschillende stappen van de simulatie en operaties voor het nauwgezet instellen van deze stappen.

Op drie verschillende manieren is het mogelijk toestanden te simuleren (zie Figuur 60), deze operaties zijn gerepresenteerd als rode lijnen voor het geselecteerde in een voorbeeld state-graph. Het gebruik van de rode kleur komt overeen met de manier waarop in de eigenlijke state-graph de geselecteerde onderdelen worden weergegeven.

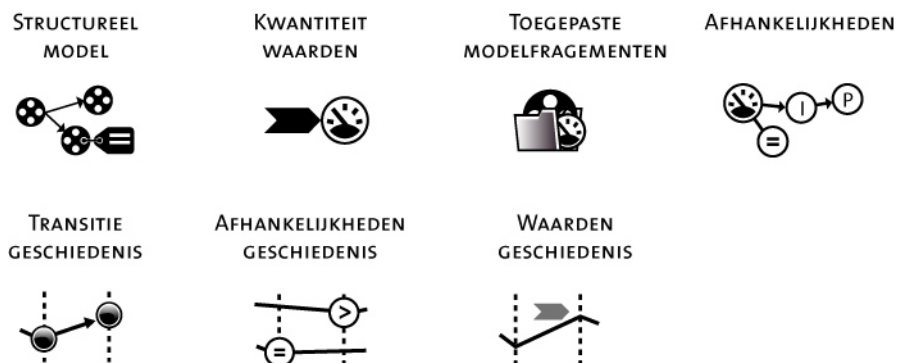


Figuur 60. Selectie operaties op toestanden

Na het selecteren van één of meerdere toestanden is het mogelijk hiervan verschillende aspecten in te zien. De operaties zijn vormgegeven door gebruik te maken van de modelingrediënten die deel uitmaken van het getoonde resultaat. Hierbij moet worden opgemerkt dat het niet alleen archetypische voorbeelden van het resultaat zijn zoals in het geval van het structurele model of de toegepaste modelfragmenten in Figuur 61 maar dat het ook een samenstelling is die in het QR vocabulaire niet mogelijk is. De pijl in combinatie met de grafische representatie van de kwantiteit is niet mogelijk omdat de pijl enkel een waarde kan geven aan een element in de kwantiteitruimte en niet aan de kwantiteit als



zodanig. Er is hierbij gebruik gemaakt van visuele abstractie om binnen zo min mogelijk ruimte zoveel mogelijk te vertellen over het soort inspectie de operatie behelst.

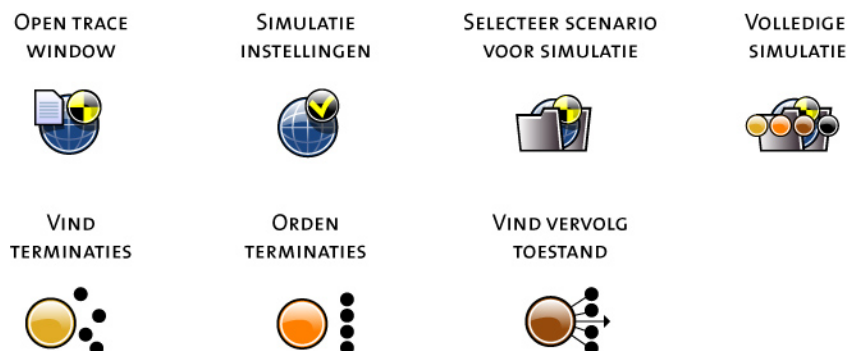


Figuur 61. Operaties voor de inspectie van elementen in geselecteerde toestanden

De inspectie operaties in Figuur 61 bestaan uit twee groepen. Ten eerste operaties voor de inspectie van afzonderlijke toestanden, de bovenste rij in de figuur en ten tweede operaties voor de vergelijking van onderdelen van verschillende toestanden onderling, weergegeven in de onderste rij. Een grafische representatie van verschillen in tijd tussen de toestanden is weergegeven in de vorm van verticale stippellijnen. De combinatie van de modelingrediënten en de verticale tijdslijnen maakt duidelijk dat deze drie operaties alle betrekking hebben op veranderingen in deze elementen tussen toestanden.

Het derde type operaties heeft betrekking op het uitvoeren van een simulatie. Het terugkerende element in de eerste rij in Figuur 62 is het pictogram voor simulatie: in het eerste geval in combinatie met een 'tekstuele eigenschappen'-modifier voor het openen van een tekstuele representatie van de simulatie. In het tweede geval is een variatie op het simulatie pictogram gebruikt voor het uitvoeren van instellingen in de simulator. In het derde en vierde geval is een combinatie gemaakt met de grafische representatie van het scenario, waarbij in het laatste geval ook het verloop van de gehele simulatie is weergegeven. Deze weergave van een volledige simulatie bestaat uit een serie toestanden in verschillende fases van de simulatie.

De Garp3 simulator doorloopt een drietal stappen om tot één of meerdere nieuwe toestanden te komen. Voor de visualisatie van die verschillende stappen is de visualisatie van Haverkort (Haverkort, 2000) gekozen die een goede representatie is van wat er in de simulator plaats vindt.



Figuur 62. Operaties voor het uitvoeren van simulatie



## 5.9 Overige operaties

Binnen Garp3 bestaat een aantal operaties dat niet van één van de vier bovengenoemde types is. De eerste drie operaties in Figuur 63 komen op een enkele plek in de interface voor en zijn daarmee zo specifiek voor het uitvoeren van een bepaalde taak dat hiervoor speciale pictogrammen zijn ontwikkeld. In het geval van selectie operaties voor alle of juist geen enkele van de elementen is gebruik gemaakt van het algemene element pictogram.



Figuur 63. Overige operaties

Een speciale plaats in de interface wordt ingenomen door de 'help'-functionaliteit (Figuur 64). In elk onderdeel is deze op te roepen, waarna een context gevoelige internet pagina wordt getoond. Deze manier van 'tips' geven, gaf aanleiding de hulp functionaliteit te personaliseren met een karakter. Door deze personalisatie zijn in een later stadium varianten te ontwikkelen voor het soort hulp dat wordt geboden: hulp in het modelleren in het QR vocabulaire of hulp bij het uitvoeren van bepaalde taken in de interface.



Figuur 64. Help personalisatie

## 6 Implementatie

Dit hoofdstuk bestaat uit een beschrijving van de geïmplementeerde onderdelen van Garp3. Hierin wordt zichtbaar op welke wijze de eerder besproken modelingrediënten en operaties hierop, terug komen in de uiteindelijke software.

### 6.1 Hoofdscherm

Na het opstarten van Garp3 verschijnt het hoofdscherm van de software. De opbouw van dit scherm bestaat uit een verdeling in drie groepen operaties, die aangeduid worden met een pictogram voor het model, modelbouw en modelsimulatie (zie Figuur 65). Deze volgorde is analoog aan het modelbouwproces waarin de gebruiker begint met een (leeg) model waar elementen aan worden toegevoegd in de modelbouw omgeving welke vervolgens als input dienen voor de simulatieomgeving. Elk van deze groepen is te openen of te sluiten door op het pictogram van die groep te klikken. Binnen elke groep worden afhankelijk van de toepasbaarheid van de operatie op de huidige toestand van het model de operaties als actief of inactief weergegeven.



Figuur 65. Hoofdscherm Garp3

De eerste groep bestaat uit operaties voor bestandsbeheer van het model, zoals eerder besproken in sectie 5.5. Aan de linkerkant van Figuur 65 is de initiële schermindeling na het opstarten van de software weergegeven. Vanwege het feit dat alleen het openen van een bestaand model of het maken van een nieuw model relevante acties zijn, zijn alleen de bestandsbeheer operaties opengeklapt.

Bij de modelbouw operaties (zichtbaar in de rechterzijde van Figuur 65) worden alle modelingrediënten weergegeven die een rol spelen in het modeleren van een kwalitatief model. Gezien eerdere overwegingen met betrekking tot modifiers voor het combineren van elementen uit het QR vocabulaire met operaties hierop (zoals besproken in 5.4), is het mogelijk deze ingrediënten weer te geven met een 'nieuw' modifier. Deze is echter niet toegepast op deze elementen omdat de operatie bestaat uit het openen van een window voor het toevoegen van elementen en niet uit het toevoegen van een nieuw element zelf. Daarnaast zijn er nog twee alternatieve visualisaties mogelijk, zie Figuur 66. Ten eerste door middel van een modifier in de vorm van een window met als betekenis 'open editor'. Het alternatief is juist andersom waarin het modelingrediënt als modifier dient. In beide gevallen kan het window een representatie zijn van het soort window dat zich opent. Ondanks dat beide visualisaties correct zijn, worden ze niet gebruikt in het hoofdmenu. De reden hiervoor is dat van elk van de modelingrediënten een getal is opgenomen voor het weergeven van het aantal dat er van dat ingrediënt bestaan. Om verwarring te voorkomen over wat dit getal voorstelt is alleen het modelingrediënt weergegeven.

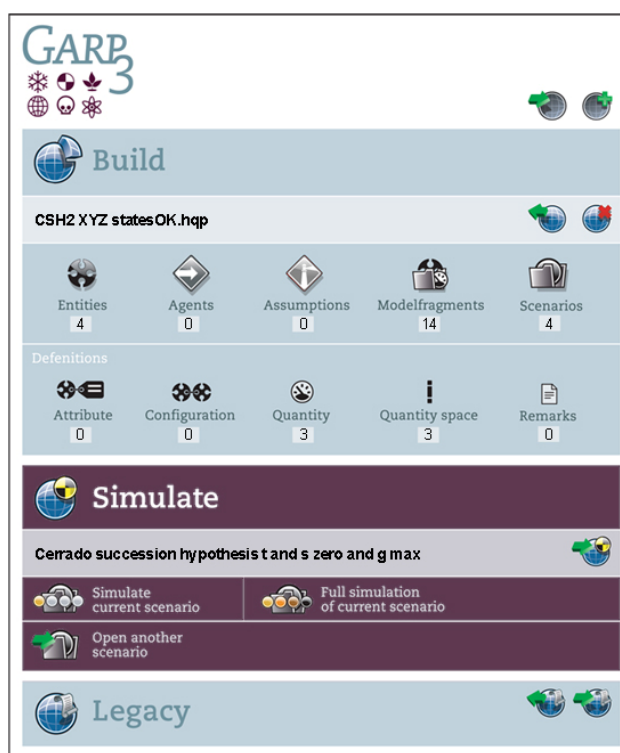


Figuur 66. Alternatieven voor het openen van de editors

De laatste groep operaties van het hoofdmenu heeft betrekking op het uitvoeren van simulaties. Er zijn twee operaties weergegeven die ook voorkomen in de simulatieomgeving, omdat een simulatie vanaf dit

punt gestart kan worden en deze instellingen voorafgaande aan de simulatie nodig zijn. De vier simulatie operaties bestaan uit het selecteren van een scenario als input voor de simulator, het starten van de simulator met het huidige scenario, het starten en volledig simuleren van het huidige scenario en tenslotte het openen van de simulator in de huidige toestand. Deze laatste operatie is dan ook alleen mogelijk indien er een simulatie heeft plaatsgevonden.

Een alternatieve weergave organisatie van het hoofdmenu is weergegeven in Figuur 67. In dit geval zijn tekstuele labels weergegeven bij de modelingrediënten. De reden hiervoor is dat de pictogrammen niet zogenaamd ‘zelf uitleggend’ zijn; er wordt niet van de grafische representatie van de modelingrediënten verwacht dat een nieuwe gebruiker de taal meteen begrijpt en deze gebruiker zal tijd nodig hebben om het vocabulaire te leren kennen. Omdat het hoofdscherm het eerste contact is voor nieuwe gebruikers kan het dienen als legenda voor de rest van de software. Deze legenda functie van het hoofdscherm is echter losgelaten omdat deze overbodig is door een weergave van het tekstlabel indien de gebruiker over één van de pictogrammen beweegt met de muis.



Figuur 67. Alternatief hoofdscherm van Garp3

## 6.2 Modelbouw

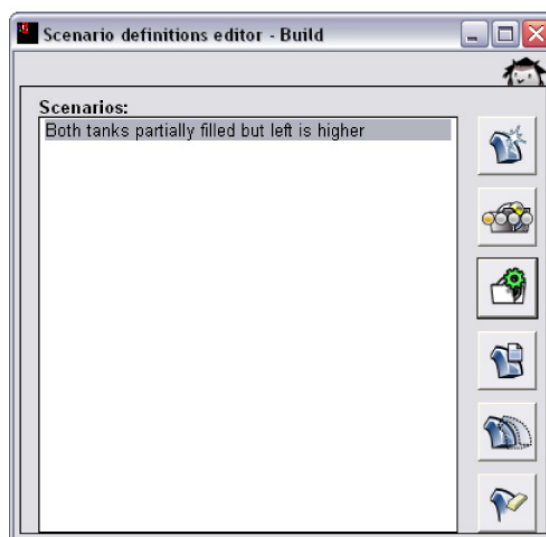
De modelbouw omgeving bestaat uit een serie windows of editors voor het aanmaken van klassen van modelingrediënten. Elk van deze windows is specifiek voor een type ingrediënt en is gebaseerd op één van vier mogelijke layout schema's. Deze layout schema's zijn afhankelijk van de manier waarop invulling wordt gegeven aan de ingrediënten. Bijvoorbeeld in het geval van entiteiten moet elk nieuw element niet alleen van een naam worden voorzien maar ook een plaats krijgen binnen de hiërarchie van entiteit typen.

De vier verschillende layout schema's zijn: Ten eerste de in 6.2.1 besproken layout, gebaseerd op een lijst van elementen waar nieuwe elementen aan kunnen worden toegevoegd en van waaruit de elementen bewerkt kunnen worden in een vervolg editor, zoals in het geval van scenario's. Het tweede schema

bestaat uit een lijst van aanwezige elementen in combinatie met operaties om direct additionele definities in te kunnen stellen, zie 6.2.2. Vervolgens worden twee overeenkomstige schema's besproken waarin de modelingrediënten worden weergegeven als een grafische representatie in een werkvlak. In het eerste geval wordt een plaatsing afgedwongen waardoor het element altijd onderdeel uitmaakt van een hiërarchie terwijl het voor de gebruiker mogelijk is in het laatste schema om de modelingrediënten zelf een plaats in het werkvlak te geven.

### 6.2.1 Window layout van lijst van modelingrediënten

De lijstweergave bestaat uit een verticaal georganiseerde lijst van tekstueel gerepresenteerde modelingrediënten, zie Figuur 68. De operaties worden aan de rechterkant weergegeven. Dit in tegenstelling tot de layouts gebaseerd op een grafische representatie, waarbij de operaties altijd aan de linkerkant worden weergegeven. Vanwege het feit dat alle ingrediënten van hetzelfde type zijn, is het overbodig een pictogram op te nemen voor de individuele elementen in de lijst. Dit is echter alleen mogelijk indien de lijst als geheel is voorzien van een aanduiding van het type elementen dat het bevat (Machado, 2004). De titel van de lijst bestaat uit een tekstuele representatie van het modeltype. Deze titel had ook kunnen bestaan uit de grafische representatie van het ingrediënt om zo consistent mogelijk het vocabulaire te volgen, het woord 'scenarios' komt namelijk verder niet terug in Garp3.



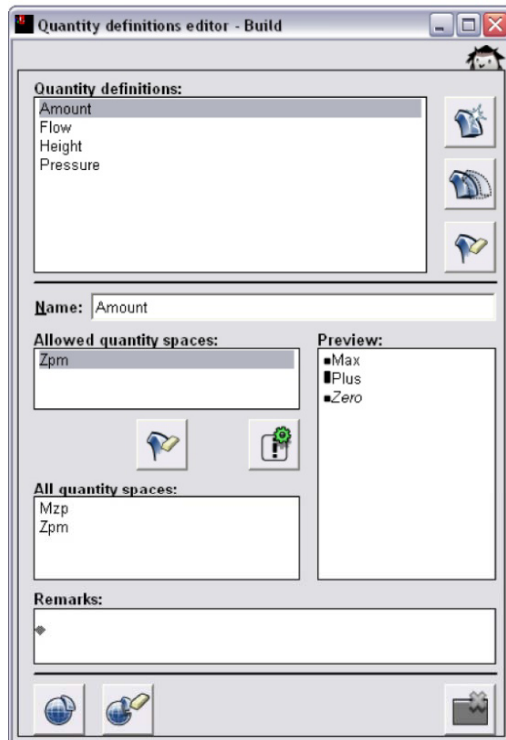
Figuur 68. Lijst van scenario's

### 6.2.2 Window layout van lijst en definitie

Een aantal editors bestaat uit een overzicht van al bestaande modelingrediënten in combinatie met de tools voor het definiëren van de inhoud van het ingrediënt (zie Figuur 69). Elk van deze editors bestaat uit een overzicht van bestaande elementen van het type specifiek voor de editor. Dit overzicht is een lijst van tekstuele elementen in combinatie met een aantal operaties hierop voor het toevoegen, verwijderen en kopiëren van een element. Onder dit overzicht is het mogelijk invulling te geven aan de elementen uit het overzicht. Zowel voor de operaties op de elementen uit de lijst als op de invulling van deze elementen is zoveel mogelijk gebruik gemaakt van de het algemene modelingrediënt in de weergave van de operaties. Indien het verschil tussen deze twee onderdelen duidelijk genoeg is, is er geen reden te verwachten dat onduidelijkheid bestaat voor gebruikers over het meerdere keren voorkomen van een bepaalde operatie

in hetzelfde window. Bij de editor voor het definiëren van structurele relaties is er geen verdere invulling van het ingrediënt mogelijk behalve additionele informatie in de vorm van 'remarks'. Alternatieve weergave voor deze editor is de lijst zoals besproken in 6.2.1.

### 1: DEFINITIE VAN KWANTITEITEN



**Quantity definitions editor - Build**

**Quantity definitions:**

- Amount
- Flow
- Height
- Pressure

**Name:** Amount

**Allowed quantity spaces:**

- Zpm

**Preview:**

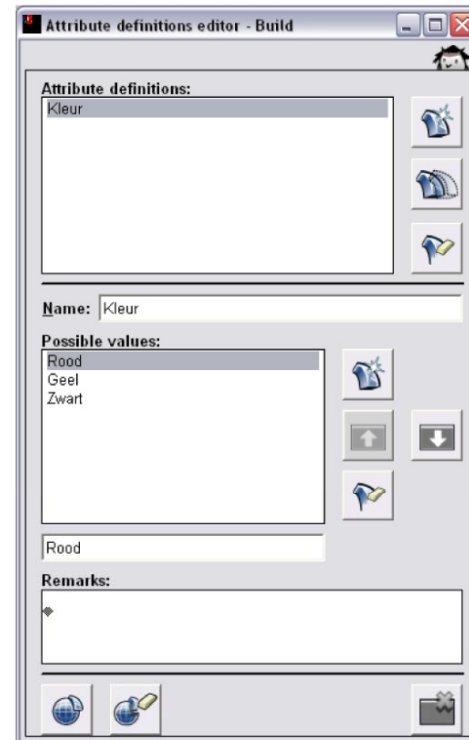
- ☐ Max
- ☒ Plus
- ☐ Zero

**All quantity spaces:**

- Mzp
- Zpm

**Remarks:**

### 2: DEFINITIE VAN ATTRIBUTEN



**Attribute definitions editor - Build**

**Attribute definitions:**

- Kleur

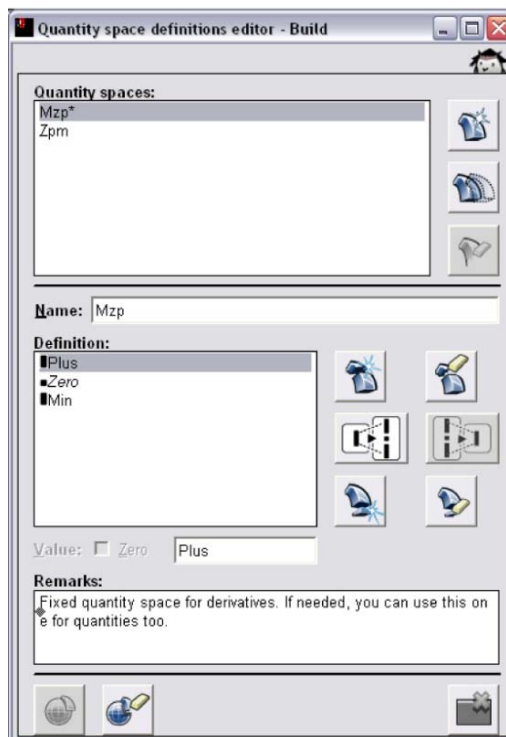
**Name:** Kleur

**Possible values:**

- Rood
- Geel
- Zwart

**Remarks:**

### 3: DEFINITIE VAN WAARDENRUIMTES



**Quantity space definitions editor - Build**

**Quantity spaces:**

- Mzp
- Zpm

**Name:** Mzp

**Definition:**

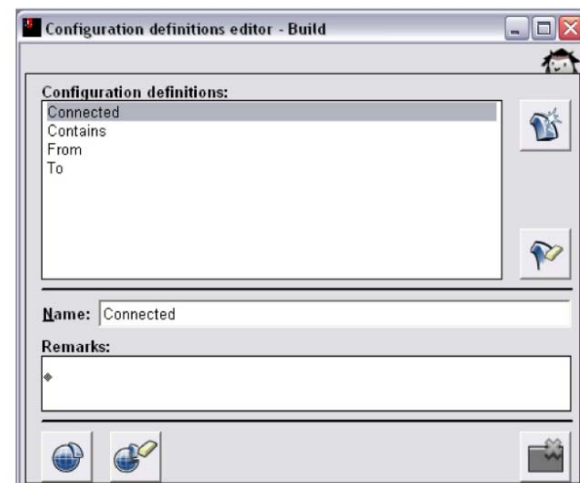
- ☒ Plus
- ☐ Zero
- ☐ Min

**Value:** ☐ Zero ☒ Plus

**Remarks:**

Fixed quantity space for derivatives. If needed, you can use this one for quantities too.

### 4: DEFINITIE VAN STRUCTURELE RELATIES



**Configuration definitions editor - Build**

**Configuration definitions:**

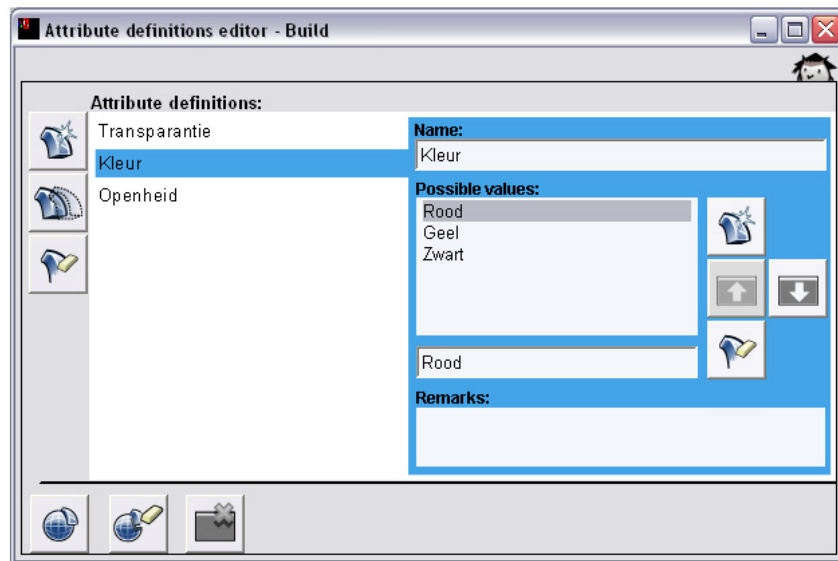
- Connected
- Contains
- From
- To

**Name:** Connected

**Remarks:**

Figuur 69. Vier editors voor modelingrediënten

Een alternatief voor de verticaal georganiseerde opbouw van de editors is een horizontaal georganiseerde opbouw waarbij gebruik wordt gemaakt van een (meestal verticaal) tabblad metafoor in horizontale vorm (zie Figuur 70). Door het gebruik van de tab in combinatie met een tintverschil is de hiërarchie van elementen goed duidelijk te maken: De invulling van de elementen is nu duidelijk gescheiden van de operaties op de elementen zelf.

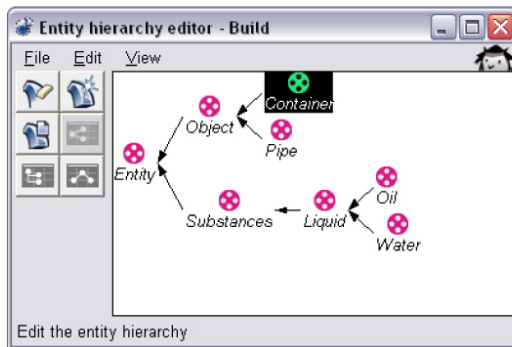


Figuur 70. Tab gebaseerde editor

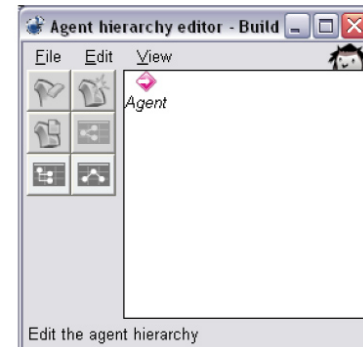
### 6.2.3 Hiërarchische window layouts

In vier gevallen worden de nieuwe elementen in een hiërarchie gezet op het moment dat deze gecreëerd worden. Voor elk van deze gevallen worden de modelingrediënten in een paarse kleur weergegeven. Deze kleur is gekozen voor alle klassen van objecten die in de vorm van een pictogram wordt gebruikt, zodat, binnen de context, de rol van het element duidelijk is. Het betreft hier namelijk klassen van elementen waarvan later instanties worden gemaakt in de simulatie omgeving. In die context zijn deze elementen zwart.

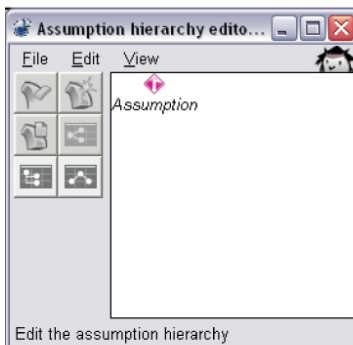
### 1: DEFINITIE VAN ENTITEITEN



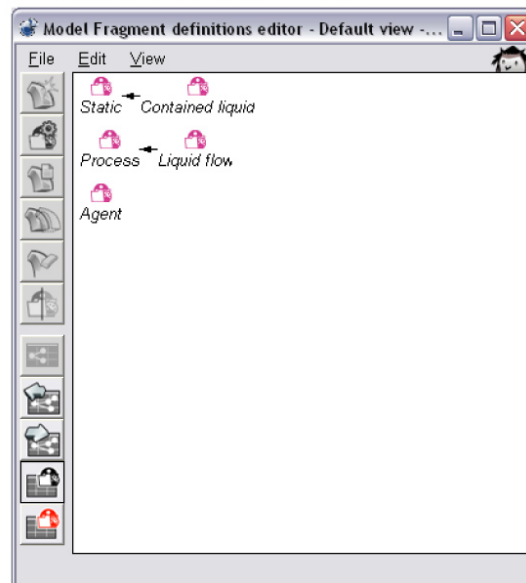
### 2: DEFINITIE VAN AGENTS



### 3: DEFINITIE VAN AANNAMES



### 4: DEFINITIE VAN MODELFRAGMENTEN



Figuur 71. Definitie van hiërarchische ingrediënten

In het geval van entiteiten, agents en aannames wordt dezelfde weergave gehanteerd (zie 1, 2 en 3 in Figuur 71). Indien een element in de hiërarchie is geselecteerd is het mogelijk hier een nieuw element 'onder' te hangen, zonder deze selectie blijft de knop inactief. Hierdoor is op natuurlijke wijze duidelijk dat een nieuw element alleen 'onder' een ander element kan worden toegevoegd ondanks het feit dat de originele 'nieuw' modifier gehanteerd wordt waarin deze eis niet visueel is verwerkt.

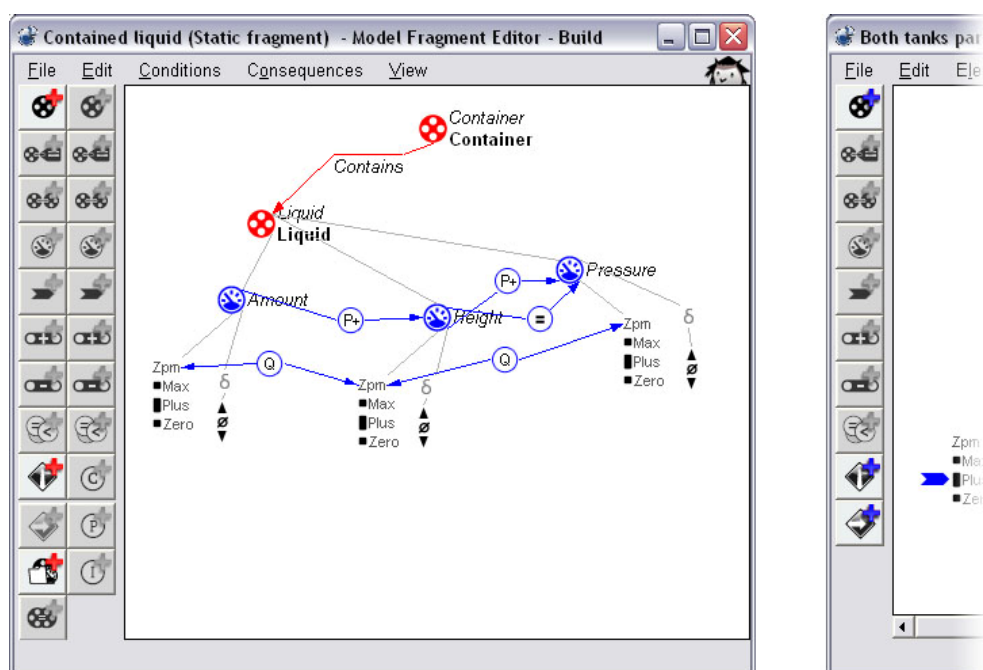
Het aanmaken van nieuwe modelfragmenten gebeurt op overeenkomstige manier, maar bevat enkele additionele functionaliteiten voor het bewerken van de inhoud van een modelfragment en voor de weergave van de modelfragmenten in de hiërarchie (zie 4 in Figuur 71).

## 6.2.4 Vrije layout van modelingrediënten

In de modelbouw omgeving is het bij het samenstellen van modelfragmenten en scenario's mogelijk de modelingrediënten zelf te positioneren binnen het werkvlak. In de linkerkant van Figuur 72 is een modelfragment weergegeven. Het samenstellen gebeurt aan de hand van de 'buttonbar', deze balk bevat afhankelijk van de geselecteerde elementen, een selectie aan mogelijke operaties. Het uitvoeren van een operatie bestaat uit het toevoegen van het geselecteerde ingrediënt aan het werkvlak. In de situatie van de figuur is niets geselecteerd en daarom is het alleen mogelijk een conditionele entiteit, agent of aanname toe te voegen. De gehele buttonbar bestaat uit twee verticaal georganiseerde sets knoppen: Aan de linkerkant de operaties voor het toevoegen van conditionele modelingrediënten en aan de



rechterkant operaties voor het toevoegen van ingrediënten aan het resultaat. In de organisatie van deze operaties zijn de ingrediënten die in beide sets voorkomen naast elkaar neergezet waardoor het makkelijker is ingrediënten terug te vinden bij herhaald gebruik.



Figuur 72. Modelbouw omgeving voor een modelfragment en 'buttonbar' voor scenario

In de implementatie van deze module is om twee redenen afgeweken van de eerder besproken modifiers: Ten eerste is getracht de knoppen zo klein mogelijk weer te geven om ruimte te besparen voor de weergave van de inhoud van het modelfragment, waardoor de bestaande 'nieuw' modifier minder geschikt is. Ten tweede is een kleur codering aangebracht in de modifier (rood voor nieuwe conditionele elementen en blauw voor het toevoegen van elementen aan het resultaat) dat niet goed toepasbaar is op de blauw met witte bestaande modifier. Een rode variant van de 'nieuw' ster heeft meer weg van een verwijder operatie dan het toevoegen van een conditioneel element, zie ter illustratie Figuur 73.

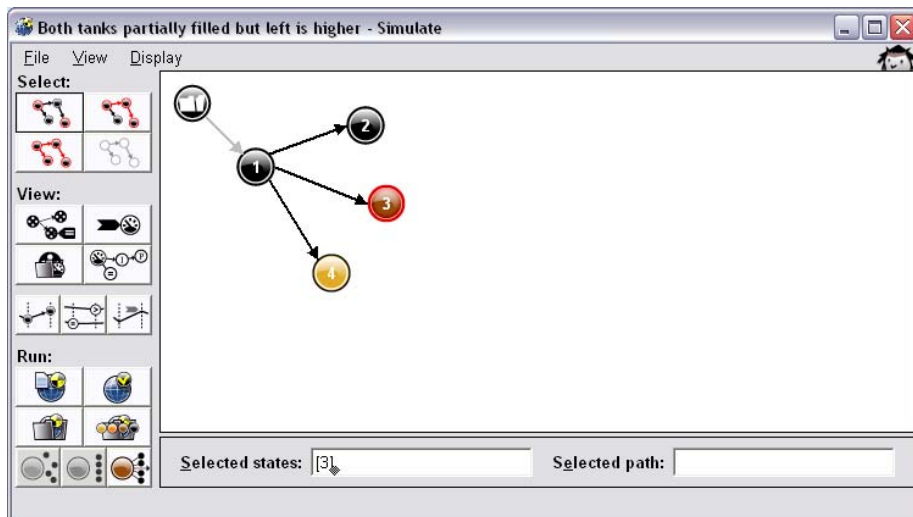


Figuur 73. Onduidelijke modifier door variatie in kleur

De module voor het vervaardigen van een scenario bestaat uit eenzelfde werkveld en een enkele buttonbar. Een scenario is de initiële situatiebeschrijving en kan daarom geen conditionele elementen bevatten. De elementen uit een scenario worden blauw weergegeven, overeenkomstig het resultaat gedeelte van het modelfragment omdat deze kleur in Garp3 gebruikt wordt voor nieuw toe te voegen modelingrediënten tijdens de simulatie.

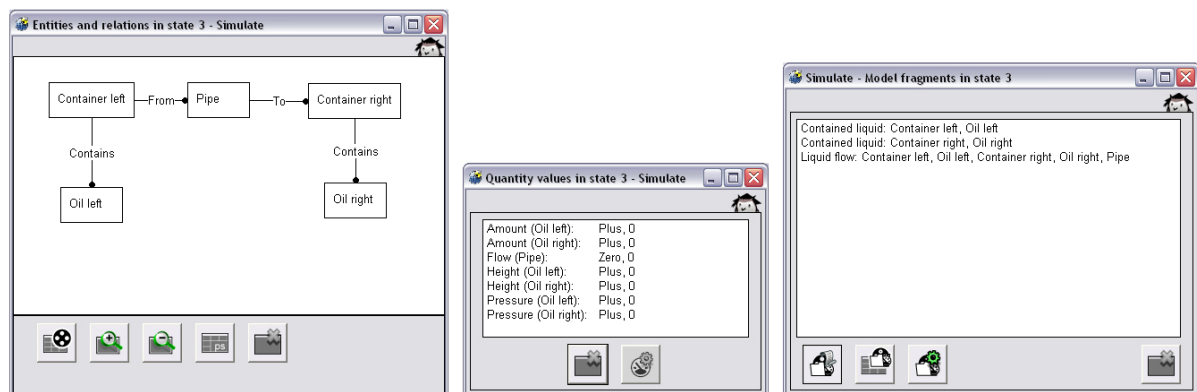
### 6.3 Modelsimulatie

Het hoofdscherm (zie Figuur 74) van de simulatie omgeving bestaat uit de state-graph met de toestanden in de huidige simulatie (waarvan de weergave is besproken in 5.1.2) en een serie operaties voor de besturing en inspectie van de simulatie (zoals besproken in 5.8).



Figuur 74. Hoofdscherm in simulatie omgeving

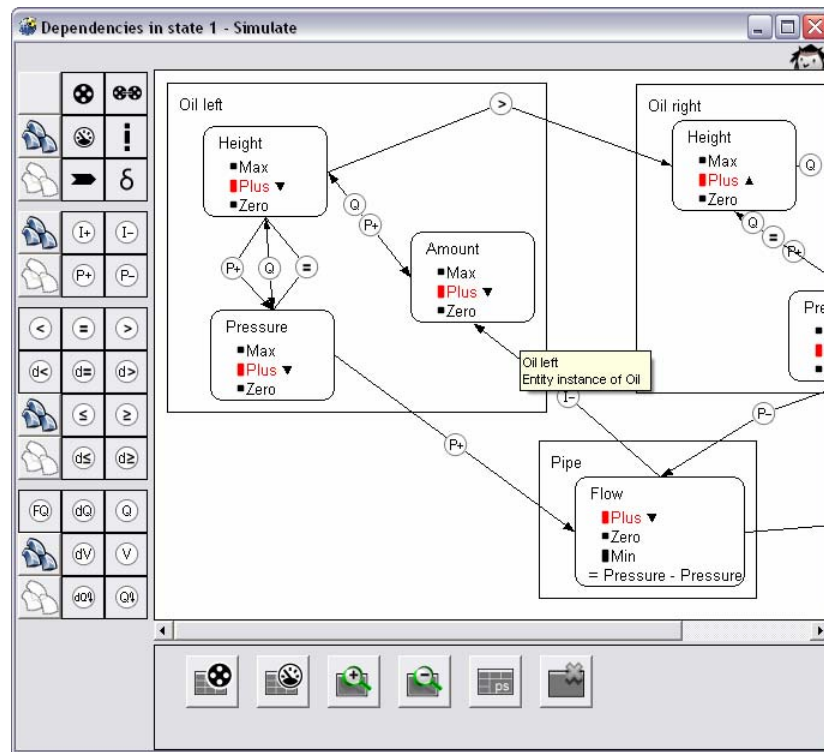
De vier beschikbare inspectie tools, onder 'view' in Figuur 74, betreffen de inspectie van de structurele elementen en relaties daartussen, de waarden van de kwantiteiten, de toegepaste modelfragmenten en tenslotte een window voor de inspectie van causale relaties en afhankelijkheden. De eerste drie hebben een overeenkomstige indeling van de mogelijke operaties op de weergegeven elementen of het window als geheel. Voor het overgrote deel bestaan deze editors uit de weergave afkomstig uit de originele VisiGarp software. De operaties zijn wel weergegeven in het nieuwe grafische vocabulaire waarbij moet worden opgemerkt dat de plaatsing hiervan weinig onderling consistent is. Op verschillende manieren worden de operaties uit verschillende groepen naast elkaar weergegeven of juist gescheiden.



Figuur 75. Inspectie tools in simulatie omgeving

In de simulatieomgeving is de inspectie van de afhankelijkheden en causale relaties een belangrijk onderdeel voor het begrip van het gedrag van het systeem als geheel. Figuur 76 is een voorbeeld van hoe deze worden weergegeven voor een bepaalde toestand. Deze weergave is een zogenaamd 'high density design' (Tufte 1990). In een high density design waarbij de kijker een grote hoeveelheid informatie krijgt aangeboden maar in staat is deze te laten presenteren naar eigen inzicht behoort de controle over de informatie toe aan de kijker in plaats van de ontwerper van de grafische weergave. De gebruiker is vrij alle elementen in het werkvlak te verplaatsen en is in staat de weergave aan te passen aan de eigen informatiebehoefte. In dit window zijn twee soorten operaties mogelijk. Ten eerste operaties op de

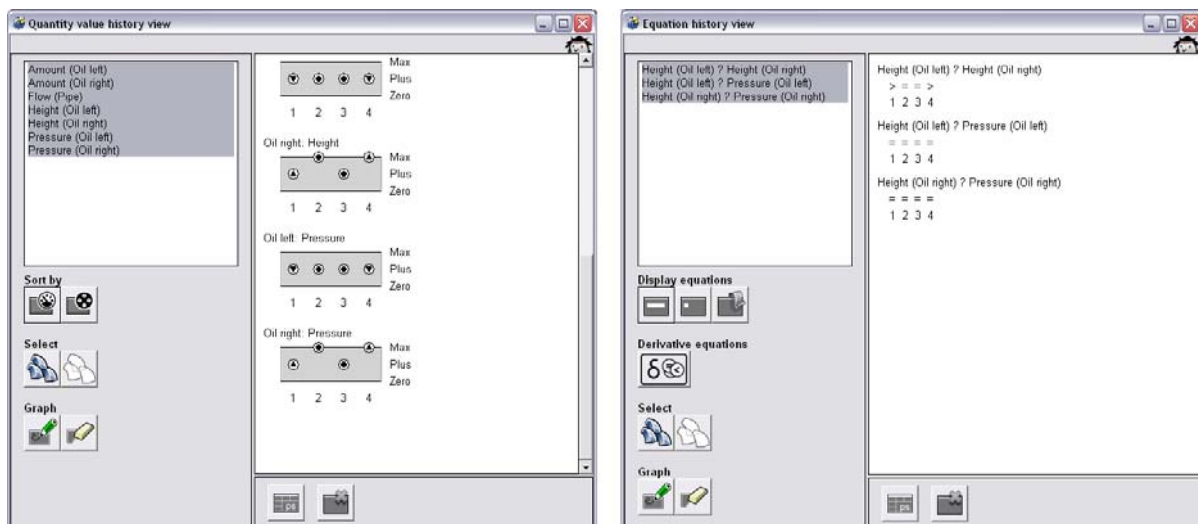
weergave van de ingrediënten van de toestand en ten tweede operaties op de layout van het werkveld als geheel. De buttonbar aan de linkerkant bestaat uit operaties voor het selecteren van elementen die, indien aanwezig in de toestand, ofwel getoond of juist verborgen moeten worden. Voor de elementen uit deze buttonbar geldt hetzelfde als voor de ingrediënten van het hoofdscherm. Er is geen modifier toegevoegd voor de operatie 'selecteer' aan de ingrediënten omdat het de enige operatie is die hier uitgevoerd kan worden.



Figuur 76. Inspectie van afhankelijkheden binnen een toestand

Vanuit de eerdere versie van de software is de plaatsing van de layout operaties overgebleven. Hierdoor staan deze horizontaal onder het werkveld. Op deze manier wordt op zeer natuurlijke wijze duidelijk dat de operaties van verschillende orde zijn. Deze verdeling is echter niet in alle windows toe te passen. In de windows voor het definiëren van elementen in een hiërarchie is maar een zeer beperkte hoeveelheid operaties mogelijk waardoor in deze situaties de operaties dicht bij elkaar zijn gezet om onnodig veel muis beweging te voorkomen.

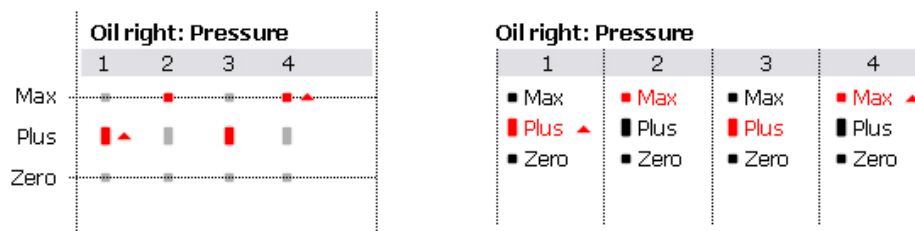
Naast de tools voor de inspectie van informatie binnen een bepaalde toestand bestaan er drie tools voor de inspectie van elementen tussen toestanden. Hiervan zijn er twee weergegeven in Figuur 77, de derde bestaat uit een tekstuele weergave van de geïnstantieerde terminatie regels en is overeenkomstig de rechter twee windows uit Figuur 75.



Figuur 77. Transitie inspectie editors in simulatie omgeving

De tools voor de inspectie van veranderingen in waarden van kwantiteiten en ongelijkheden tussen kwantiteiten is weergegeven in Figuur 77. In beide gevallen is er een verticale scheiding, waarbij aan de linkerkant de lijst met kwantiteiten is weergegeven in combinatie met de mogelijke operaties. Deze operaties zijn onderverdeeld in operaties op de manier waarop de elementen in de lijst zijn weergegeven, operaties voor de selectie van alle of geen van de elementen in de lijst en tenslotte de operaties voor het uitvoeren van de weergave operatie op het rechterscherm of het leegmaken hiervan.

De invulling van de waardengeschiedenis bestaat uit een grafische weergave van horizontaal georganiseerde toestanden in combinatie met lijnen (voor punten in de waardenruimten) en vlakken (voor intervallen in de waardenruimten) waarlangs pictogrammen zijn weergegeven. Ondanks dat deze grafische representatie niet consistent is met het vocabulaire van de modelingrediënten is de werking heel overzichtelijk. Twee voorbeelden, zie Figuur 78, van dezelfde informatie met de grafische representatie van de modelingrediënten tonen aan dat deze misschien herkenbaarder zijn binnen de context maar niet duidelijker.

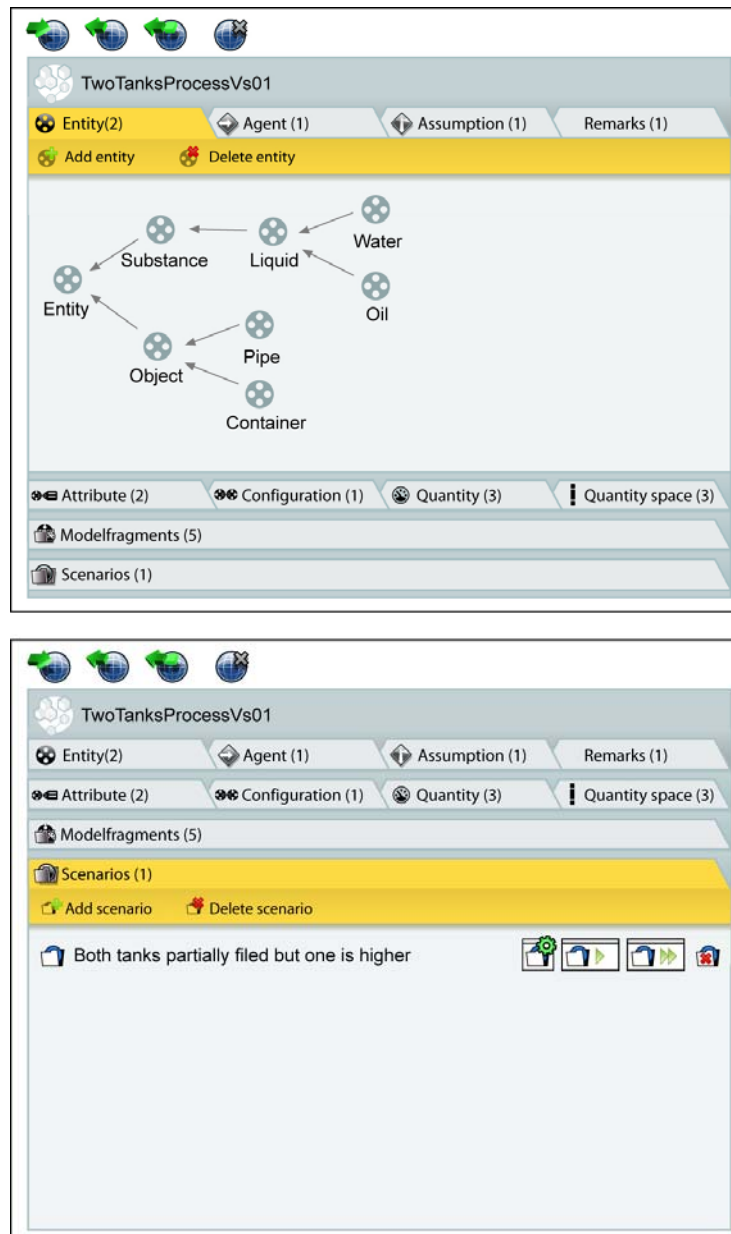


Figuur 78. Alternatieve representatie van de waardengeschiedenis

## 6.4 Alternatieve layouts

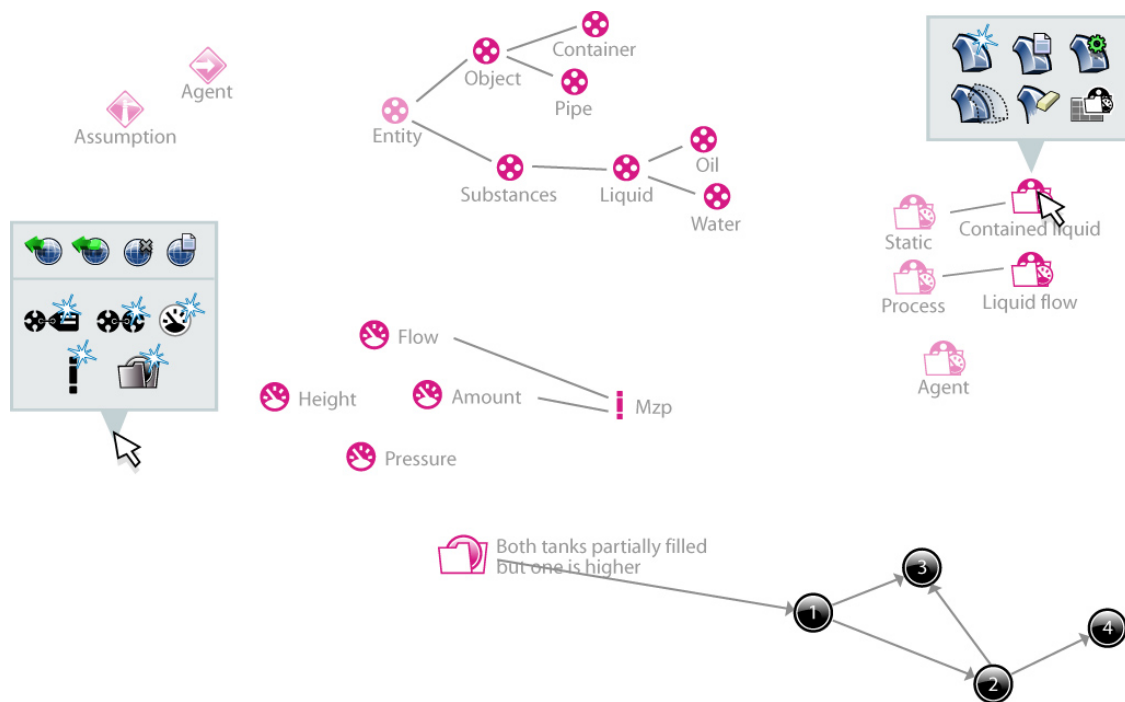
De gehanteerde aanpak in Garp3 voor de verschillende editors en inspectietools is gebaseerd op verschillende onafhankelijke windows. Het voordeel is dat de gebruiker zelf kan besluiten welke informatie zichtbaar is. De nadelen zijn de hoeveelheid tijd die een gebruiker nodig heeft voor het organiseren van de windows in een voor de gebruiker zinvolle manier en het wisselen tussen de editors (Machado, 2004). Een alternatief hiervoor is een tab-gebaseerde aanpak waarin de editors worden weergegeven in een andere tab van hetzelfde window van een vast formaat. Het gevolg is dat altijd maar één editor zichtbaar en actief is. In Figuur 79 is een tab-gebaseerde interface voor Garp3 weergegeven. De

opbouw van het window bestaat uit een weergave van tabbladen voor alle modelingrediënten waarvan er één altijd 'open' staat ofwel actief is. Het actieve tabblad is visueel duidelijk geactiveerd en bestaat uit een horizontale buttonbar voor operaties voor het toevoegen en verwijderen van ingrediënten van het type dat van behoort tot dat tabblad en een werkvlak. Het werkvlak bestaat uit de hiërarchische organisatie van elementen, een lijst of een combinatie van lijst en editor zoals eerder besproken in 6.2. Het is mogelijk een visueel onderscheid aan te brengen tussen operaties voor het toevoegen en verwijderen van elementen en de operaties voor de invulling van deze elementen. Door de eerste weer te geven in de buttonbar en de laatste achter het element, zie het overzicht van scenario's in de onderste helft van Figuur 79. Op deze manier is het mogelijk visueel duidelijk te maken waarop de operaties betrekking hebben. De overzichten van scenario's en modelfragmenten bestaan uit een hiërarchie of lijst waardoor het mogelijk is het tab window relatief klein te houden. Dit is namelijk van belang omdat de simulatie van een scenario en de invulling van scenario en modelfragment in een nieuw window moet gebeuren omdat deze relatief veel ruimte nodig heeft.



Figuur 79. Tab-gebaseerde layout van software

Een alternatief voor de aanpak gebaseerd op overlappende windows en de tab-gebaseerde aanpak is de desktop metafoor waarin alle modelingrediënten worden weergegeven in één groot werkvlak, zie Figuur 80. De gebruiker is in staat met de muis een aantal operaties uit te voeren. Ten eerste is het mogelijk meerdere elementen te selecteren, ongeacht het type modelingrediënt, door hier een selectie vierkant omheen te trekken. Vervolgens kunnen deze geselecteerde elementen worden verplaatst naar een plaats binnen of buiten het scherm. In dat laatste geval zal de interface het werkvlak opschuiven om de selectie in beeld te houden. Eventuele editors die open staan schuiven mee uit beeld. Daarnaast is het mogelijk een contextsensitief menu op te roepen voor het uitvoeren van operaties op het geselecteerde element of elementen. Zie de mogelijke operaties op het geselecteerde modelfragment in het voorbeeld. Door op het werkvlak zelf te klikken verschijnt ook een menu voor het toevoegen van elementen en filemanagement.



Figuur 80. Layout gebaseerd op desktop

## 6.5 Grafisch vocabulaire van Garp3

Ondanks de ontwikkeling van een gemeenschappelijk grafische vocabulaire zijn er een aantal oorzaken aan te wijzen waardoor er geen naadloze integratie van de eerder ontwikkelde componenten is. Zoals eerder besproken liggen er verschillende grafische principes ten grondslag aan modelbouw en simulatie omgeving (onder andere in 5.1). De onderliggende grafische principes van de modelbouw omgeving sluiten beter aan op de het ontwikkelde vocabulaire omdat er gebruik wordt gemaakt van pictogrammen voor de modelingrediënten. Dit in tegenstelling tot de representatie van de ingrediënten in de simulatieomgeving waarbij de representatie onlosmakelijk verweven is met de weergave. In dit geval is duidelijke sprake van een diagrammatische weergave (Kulpa, 1994). Dit neemt echter niet weg dat het vocabulaire een rol speelt in het uitvoeren van operaties. Door deze in beide omgevingen consistent door te voeren ontstaat een visuele herkenbaarheid. Door de layout schema's van de twee omgevingen meer met elkaar in overeenstemming te brengen is dit nog te verbeteren. Bijvoorbeeld de inspectie editor van structurele relaties in de simulatieomgeving (zie linker editor in Figuur 75) komt overeen met het layout schema waarin onderdelen van het werkvlak vrijelijk geplaatst kunnen worden in het werkvlak, zie Figuur 72. Hieruit is de blauwdruk van de originele componenten terug te zien. Deze verschillen tussen de modelbouw en simulatie omgeving zijn ook te interpreteren als natuurlijke manier van functionaliteiten scheiden binnen de software. Zoals eerder besproken zijn de gehanteerde visualisaties binnen de omgevingen specifiek geschikt voor de taken die daar moeten worden uitgevoerd. Dan is voor de gebruikers aan de gehanteerde visualisatie af te leiden welk soort operaties, in die context, kunnen worden uitgevoerd.

## 7 Pilot evaluatie van operaties

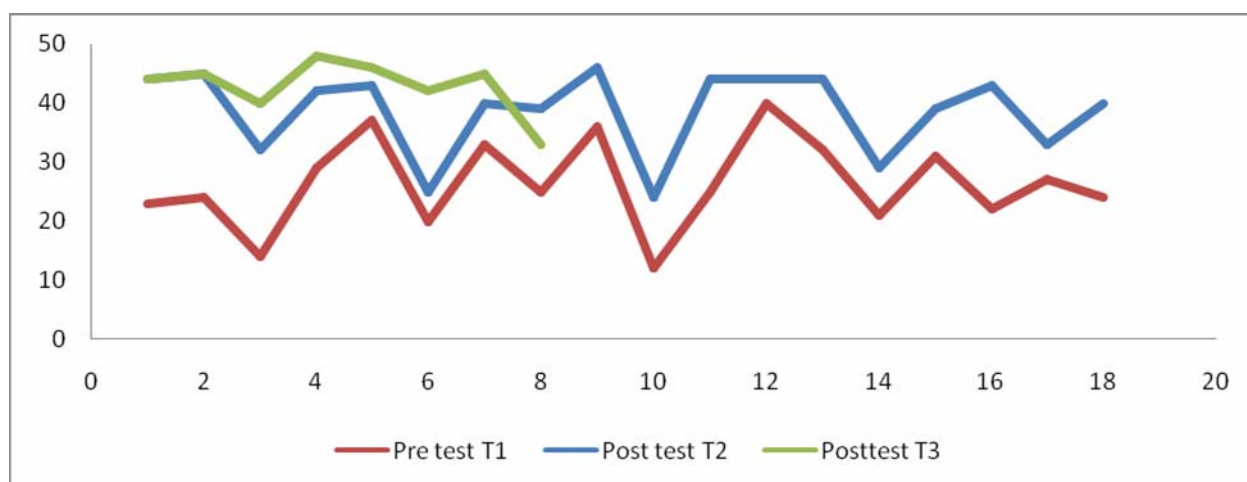
Een pilot evaluatie is uitgevoerd met een pretest-posttest-follow up design. Hiertoe is een meerkeuze vragenlijst van 50 vragen samengesteld met betrekking tot het uitvoeren van taken in Garp3, waarbij één



van de vier mogelijke antwoorden goed is. De vragen bestaan uit een pictogram dat gebruikt wordt voor het uitvoeren van een specifieke operatie uit de modelbouw omgeving of simulatieomgeving van de software. Daarnaast zijn er vragen die betrekking hebben op algemene operaties, bijvoorbeeld filemanagement, of die betrekking hebben op het model als geheel of zowel van toepassing zijn in de modelbouw als simulatie omgeving en tenslotte zijn er vragen die betrekking hebben op operaties voor het manipuleren van de interface.

Deze vragenlijst is bij twee groepen respondenten afgenomen die beide weinig kennis van QR en geen kennis van Garp3 hadden ten tijde van de pretest. Ten eerste een groep deelnemers aan een QR workshop in het kader van Naturnet-Redime in Oktober 2005 en ten tweede een groep studenten aan de faculteit Wiskunde en Informatica in September 2005 tijdens de 'QR Summer School' in Wales.

De onderzoeksopzet bestaat uit een pretest nadat de respondenten een algemene inleiding in QR hebben gekregen. Vervolgens wordt, in ongeveer zes uur, een serie opdrachten uitgevoerd met de software. Tenslotte wordt aan het einde van de dag dezelfde vragenlijst nog een keer afgenomen. Er is geen gebruik gemaakt van een controlegroep omdat het een pilot evaluatie betreft en het aantal respondenten laag is.



Tabel 1. Positieve scores op de drie vragenlijsten

Het aantal goede antwoorden op de pretest is bijna 26 van de 50 vragen. In het geval van de post-test is deze score 38 en bij de follow-up 42. Omdat de uitslagen op de post-test en follow-up niet normaal verdeeld zijn, wordt er een non-parametrische test gedaan; de zogenaamde Wilxon signed ranks toets, zie bijlage 2. Hieruit blijkt dat het interacteren met de software een positieve invloed heeft op het benoemen van de pictogrammen in de post-test (aantal goed op pre-test;  $25,89 \pm 7,31$  range 12 – 39; aantal goed op post-test;  $37,89 \pm 6,78$  range 24-45; Wilcoxon gepaarde waarnemingen:  $p < 0,001$ ). Dit betekent dat het visuele vocabulaire van Garp3 bruikbaar en leerbaar is.

Bij de follow-up is het aantal correct geïdentificeerde pictogrammen weliswaar hoger dan bij de post-test (aantal goed op post-test;  $37,89 \pm 6,78$  range 24-45; aantal goed bij follow-up;  $42,0 \pm 4,03$  range 33-47), maar dit verschil is niet significant (Wilcoxon gepaarde waarnemingen: sig. = 0,108). Daarnaast zijn de resultaten tussen pre-test en de follow-up ook significant beter (sig = 0,007). Dat het aantal correct geïdentificeerde pictogrammen bij de follow-up test niet is afgenomen en significant verschilt met de pre-test, betekent dat het vocabulaire goed is onthouden.

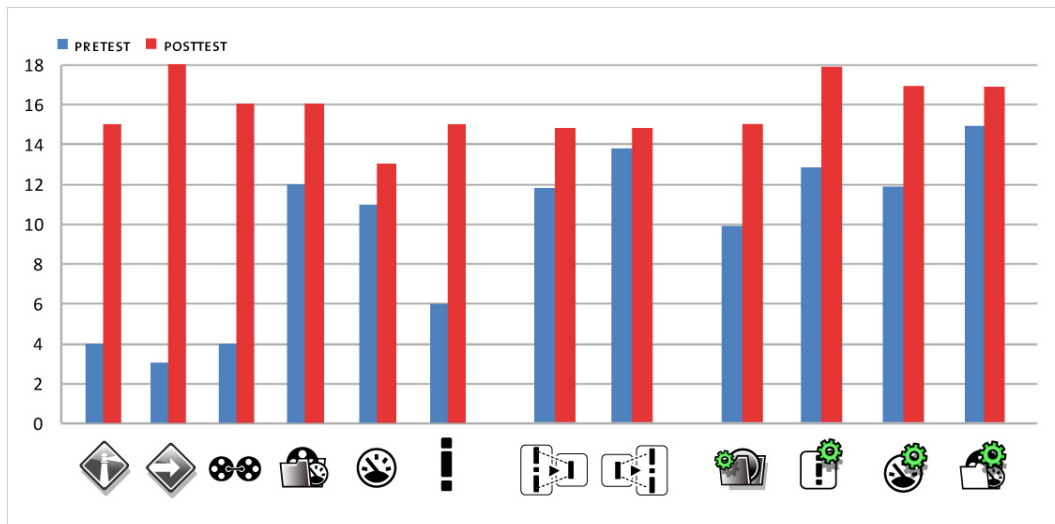


Naast dat uit het onderzoek blijkt dat de leerbaarheid van het vocabulaire als geheel voldoet aan de uitgangspunten worden hier de afzonderlijke scores op de vragenlijst uiteen gezet (zie Figuur 81 tot en met Figuur 84). Hieruit is op te maken dat de uitslagen verschillen laten zien tussen pictogrammen. Deze zijn niet geheel toe te schrijven aan het visuele vocabulaire omdat niet is af te leiden in hoeverre de abstractheid of complexiteit van de operatie een negatief effect heeft gehad op het begrip van het pictogram. Daarnaast is er verschil tussen de mate waarin de gebruiker in contact komt met het pictogram uit de vragenlijst in de uitvoer van de opdrachten. In deze opdrachten zijn de operaties moeilijk te balanceren zodat elke operatie even vaak voorkomt. Een mogelijk gevolg hiervan is dat gebruikers hierdoor een beter begrip hebben van operaties die vaker zijn gebruikt in de uitvoer van de opdrachten. Dit neemt niet weg dat de grafische representatie van de operatie een rol speelt in de verschillen. Een deel van deze verschillen en dan vooral in de resultaten op de pre-test, wordt hieronder in het licht van de grafische representatie toegelicht.



Figuur 81. Scores van pre-test en post-test op vragenlijst

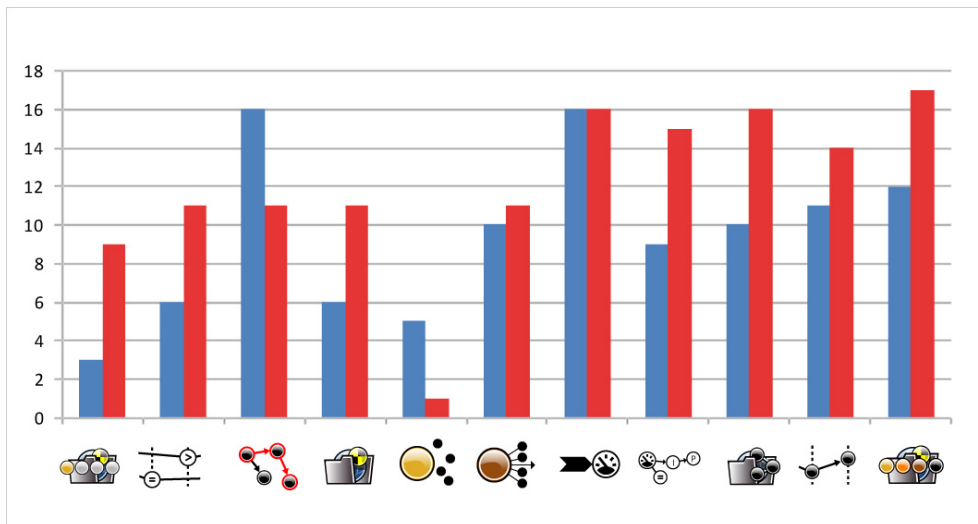
In Figuur 81 zijn de laatste vijf operaties op het model als geheel. In de pre-test is de score van deze operaties ongeveer even laag en ze laten allen (de eerste drie meer dan de laatste twee) een grote stijging zien op de post-test. Hieruit is vrijwel zeker af te leiden dat het aan de grafische representatie ligt dat de lage score veroorzaakt. Het overeenkomstige element dat hierin is gebruikt is dat van het model, zoals eerder besproken, in verschillende verschijningsvormen en met verschillende modifiers. De vormgeving van deze pictogrammen is sterk bepaald door het grafische primitief voor het model uit het vocabulaire. Ondanks dat dit initieel volledig onbekend is, illustreert deze groep pictogrammen dat het snel geleerd kan worden door met de software te interacteren.



Figuur 82. Vervolg scores op vragenlijst

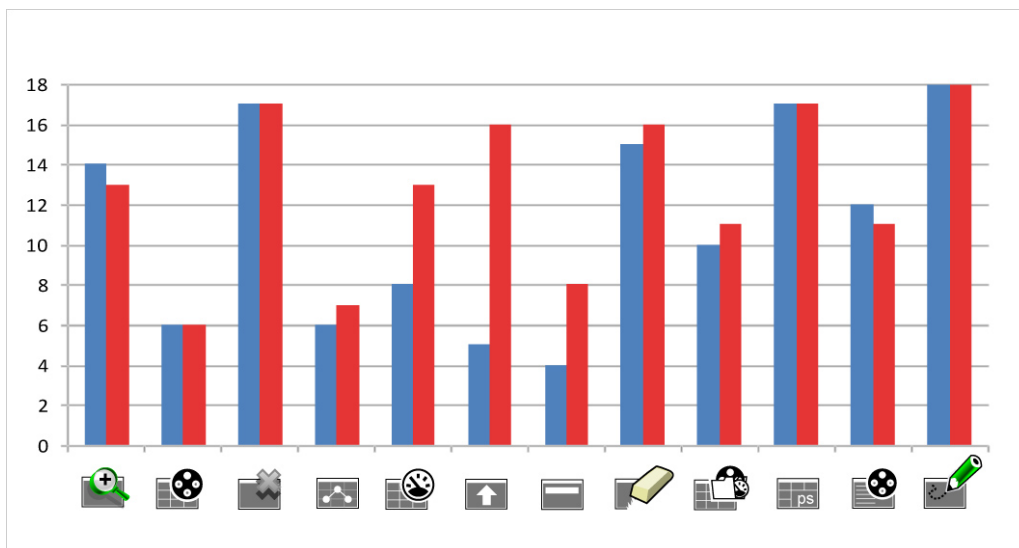
De eerste groep operaties uit Figuur 82 komt visueel overeen met de modelingrediënten uit het vocabulaire en dient om de desbetreffende editor te openen. Hierin verschillen vooral het modelfragment en de kwantiteit van de rest in de pre-test. Een verklaring hiervoor kan zijn dat de representatie van de pictogrammen overeenkomt met het mentale beeld van het concept dat de respondenten hadden. In het eerste geval is de representatie gebaseerd op hergebruik van modelingrediënten in een folder dat overeen blijkt te komen met het concept 'modelfragment'. In het tweede geval blijkt de grafische weergave van de analoge meter een bruikbare representatie van een kwantiteit.

Een opmerkelijke set operaties uit Figuur 82 zijn de operaties voor het openen van de waardenruimte editor (6<sup>e</sup> in de rij) en de operaties voor het opsplitsen of samenvoegen van elementen uit deze waardenruimtes (7<sup>e</sup> en 8<sup>e</sup> in de rij). Ondanks dat ze op de post-test gelijke score krijgen, verschillen ze in de pre-test. De representatie van de operatie is in de laatste twee gevallen vrij letterlijk gebaseerd op het samenvoegen en opsplitsen van een waardenruimte. Ondanks dat de respondenten het concept waardenruimte niet kenden (gezien de score op de eerste van de drie) zijn de operaties op de nog onbekende waardenruimte wel meteen duidelijk. Dit kan tevens de verklaring zijn voor het grote verschil tussen de score van deze 'open waardenruimte editor' en de alternatieve weergave van dezelfde operatie (10<sup>e</sup> in de rij). In het laatste geval is de waardenruimte voorzien van een modifier ('bewerken') en een lijn om het modelingrediënt. Vanwege deze additionele grafische elementen is waarschijnlijk voor de respondenten duidelijk dat het een 'bewerken' operatie betreft.



Figuur 83. Vervolg scores op vragenlijst: Simulatie operaties

De pictogrammen uit de vragenlijst die betrekking hebben op modelsimulatie zijn weergegeven in Figuur 83. Hier zijn twee operaties die een afwijkende score hebben van de andere resultaten. Het pictogram voor de selectie van paden in de state-graph (3<sup>e</sup> in de rij) en voor het vinden van terminaties voor de geselecteerde toestand (5<sup>e</sup> in de rij) hebben een lagere score op de post-test dan op de pre-test. In het geval van de selectie operatie is het mogelijk dat er verwarring ontstaat bij de respondenten door de context waarin het pictogram wordt gebruikt. Deze context bestaat uit een drietal andere selectie operaties op toestanden, zie Figuur 74. Deze verklaring gaat echter niet op voor de slechte score van het tweede pictogram omdat een operatie uit dezelfde context (namelijk het sluiten van de terminaties, de 6<sup>e</sup> in de rij) een stuk beter scoort.



Figuur 84. Vervolg scores op vragenlijst: Window/layout operaties

Operaties voor de organisatie van een window of de weergave van elementen in een window zijn weergegeven in Figuur 84. Ondanks de grote variatie in resultaten vallen een aantal pictogrammen op door de hoge scores. Ten eerste zijn dit de operaties voor het sluiten en leeg maken van het window (de 3<sup>e</sup> en 8<sup>e</sup> in de rij). Beide zijn zeer begrijpelijk ondanks de verschillen die ten grondslag liggen aan de representatie. In het eerste geval is de representatie gebaseerd op conventie en in het tweede geval op

een metaforische relatie met de gum om potlood strepen te verwijderen. Ten derde scoort de operatie voor het exporteren van de inhoud van een window naar een postscript bestand hoog. Deze score is te verklaren door de gebruikelijke extensie voor postscript bestanden: ps. De teken operatie scoort zowel op de pre-test als op de post-test volledig. Dit impliceert dat niet alleen het pictogram duidelijk is maar waarschijnlijk ook dat de software voldoet aan het verwachte resultaat.

### **7.1 'Slechte pictogrammen'**

Uit de bovenstaande grafieken komt duidelijk naar voren dat er grote variatie in de uitslagen per pictogram bestaat. Sommige van de pictogrammen scoren niet alleen laag op de pre-test maar verbeteren nauwelijks in de post-test, of verslechteren zelfs in een paar gevallen. Een logische conclusie is dat het slechte pictogrammen zijn. Hiervoor zijn een aantal mogelijke oorzaken aan te wijzen. Ten eerste kan een 'slecht' pictogram simpelweg een slechte representatie zijn; het is geen goede match tussen hoe het eruit ziet en wat het doet. Ten tweede is het mogelijk dat de gebruiker geen verschil ziet in de software na het uitvoeren van de actie (bijvoorbeeld het verbergen of tonen van een relatie die niet voorkomt in een bepaalde toestand). In dat geval is een pictogram 'slecht' omdat de achterliggende functionaliteit van de operatie niet klopt. Tenslotte wordt een deel van de operaties initieel beter begrepen voorafgaande aan het interacteren met de software. Een potentiële oorzaak hiervan is de context waarin de operaties voorkomen. Binnen deze context worden vaak operaties gegroepeerd in functionele groepen die visueel overeenkomstig zijn. Ondanks dat deze visueel onderscheidend zijn van elkaar kan het aanbod van meerdere operaties die 'ongeveer' hetzelfde doen, initieel verwarrend zijn voor gebruikers. Als gevolg volstaat het dan waarschijnlijk niet om alleen het 'slechte' pictogram te vervangen. Een alternatieve weergave voor deze groep operaties als geheel kan tot gevolg hebben dat de afzonderlijke operaties initieel beter worden begrepen.

## **8 Conclusie en discussie**

In het kader van het NaturNet-Redime project zijn drie systemen (Garp, VisiGarp en Homer) geïntegreerd tot een enkel raamwerk: Garp3; een raamwerk voor de bouw, simulatie en inspectie van kennismodellen gebaseerd op Qualitative Reasoning (QR). Dit werkstuk presenteert in die context een visuele taal die is ontwikkeld voor de modelingrediënten uit dit QR vocabulaire. Deze is zodanig dat dit resulteert in een eenvoudig leerbare taal van zowel tekst als beeld die bovendien goed onthouden wordt. Naast de ingrediënten zijn de operaties op deze ingrediënten weergegeven in hetzelfde vocabulaire waardoor duidelijk is op welke ingrediënten een operatie van toepassing is. Hiervoor zijn principes uiteengezet over grafische representaties in het algemeen en representaties van operaties in het bijzonder die ten grondslag liggen aan de ontwikkeling van deze taal. Een pilot-evaluatie ondersteunt de uitgangspunten met betrekking tot de leerbaarheid van het ontwikkelde vocabulaire. Bovendien is Garp3 volledig operationeel en de gebruikers groep vanuit het NaturNet-Redime project groeit. Daarnaast is Garp3 opgenomen in een handboek over 'Ecological modelling' (Bredeweg et al, 2006). Dit alles ondersteunt de hypothese dat de software en bijbehorende grafische aspecten voldoende bruikbaar zijn. Aan de software zijn in een later stadium nog additionele functionaliteiten en pictogrammen toegevoegd om bijvoorbeeld zogenaamde 'concept maps' te kunnen maken. In totaal bestaat Garp3 uit iets minder dan 400 pictogrammen: 155 modelingrediënten, 83 causale relaties en ongelijkheden en 152 operaties.

Omdat het QR vocabulaire een abstracte representatie is van concepten zijn er in het initiële gebruik van dit vocabulaire nauwelijks aanknopingspunten met bekende fysieke objecten of ander bekend vocabulaire. Dit heeft tot gevolg dat veel modelingrediënten zijn gebaseerd op arbitraire vormen waarvan verschillende varianten zijn ontwikkeld. In het geval van operaties bestaat een groot deel uit redelijk algemene operaties, waarvoor evenknieën te vinden zijn in het grafische vocabulaire van andere domeinen. In de ontwikkeling van het vocabulaire van Garp3 is op deze manier ‘geleend’ uit gangbaar vocabulaire, voornamelijk in het geval van zogenaamde modifiers. Dit draagt bij aan het initiële begrip van de operaties, in het gebruik van de software. Om het begrip van het vocabulaire te vergroten zou toekomstig onderzoek naar andere visuele vocabulaires, waaruit geleend kan worden, wellicht in nog meer herkenbare visualisaties resulteren.

Uit het evaluatieonderzoek is naar voren gekomen dat er een aantal pictogrammen voor operaties zijn gemaakt die de respondenten lager scoren na het werken met de software, bijvoorbeeld de pictogrammen voor het selecteren van toestanden. Deze hebben als overeenkomst dat ze in de software voorkomen in een functionele groep van vergelijkbare pictogrammen. Hieruit is op te maken dat de pictogrammen afzonderlijk begrijpelijk zijn maar dat verwarring ontstaat door de nuance verschillen tussen pictogrammen in de context waarin ze voorkomen. Het ligt in de verwachting dat na enig gebruik van de software deze nuances duidelijk worden en de verwarring zich niet meer voor zal doen. Ondanks dat dit een vanzelfsprekende ontwikkeling is in het leren van een onbekend vocabulaire, kan nader onderzoek inzicht geven in de manier waarop deze verwarring voorkomen of ondersteund kan worden.

In de simulatieomgeving van de software wordt gebruik gemaakt van een diagrammatische representatie: De kwantiteiten die behoren bij een entiteit worden ‘in’ die entiteit weergegeven. Voor deze relatie is het mogelijk alternatieve visualisaties te gebruiken gebaseerd op het vervormen van de entiteit, zoals besproken in 5.3. Elk type kwantiteit past een andere vervorming toe op de bijbehorende entiteit, zoals het roteren of schalen van het element. De mate van vervorming is afhankelijk van de waarde in de waardenruimte. De weergave van deze entiteit bestaat dan uit een vierkant waarop cumulatief vervormingen zijn uitgevoerd. Nader onderzoek moet uitwijzen in welke mate gebruikers in staat zijn deze vervormingen terug te vertalen naar de waardes van de verschillende kwantiteiten en wat dit betekent voor het weergeven van relaties tussen kwantiteiten.

Bij vervolg onderzoek is het ook van belang naar de functionaliteit van de software zelf te kijken. In het ontwikkelen van de pictogrammen speelt de data-inkt ratio een rol, analoog hieraan speelt de klik-inkt ratio een rol bij het uitvoeren van operaties. Dit is de hoeveelheid inkt (of pixels) dat zinvol of relevant op het scherm weergegeven is per muis klik. Het is van belang dat alle inkt, ofwel het gehele scherm in zo min mogelijk muisklikken zo zinvol mogelijk gebruikt wordt. In Garp3 organiseert de gebruiker zelf de verschillende windows over het scherm. Hierdoor worden er muiskliks gedaan om het scherm inzichtelijk te krijgen met als gevolg dat de klik-inkt ratio afneemt. Potentieel biedt de in 6.4 besproken tabgebaseerde layout hier een oplossing voor omdat de layout van het systeem bepaald wordt door de software. Deze aanpak gecombineerd met een initieel observatie onderzoek naar de manier waarop gebruikers verschillende windows organiseren kan een goede aanzet zijn voor Garp4.

Mogelijke ontwikkelingen hierin waarin de visuele taal een grote rol speelt zijn het werken met meerdere modellen tegelijkertijd of het nemen van model ‘snapshots’ (versies van hetzelfde modellen waarin

wijzigingen zijn aangebracht). Aan de hand hiervan kunnen meerdere simulaties worden gestart van deze modellen. Het visueel mogelijk maken deze taken te ondersteunen, vergt niet alleen inzicht in manieren van visualiseren maar ook in de cognitieve capaciteiten van de gebruikers en de technische mogelijkheden van het systeem.

Eén belangrijke eigenschap van pictogrammen is volledig buiten beschouwing gelaten in dit werkstuk. Dit neemt niet weg dat het een rol heeft gespeeld in de ontwikkeling van het vocabulaire en het afwegen van de alternatieven. Naast de leerbaarheid, begrijpelijkheid, expressief en onderscheidend vermogen, zijn de pictogrammen ook mooi.

## 9 Literatuur

- Bertels, D.** (2001) 'Praatjes en Plaatjes'-Een inventarisatie in gebruikte visualisatie technieken en een beschouwing van de relatie tussen vorm en betekenis. Literatuurscriptie, University of Amsterdam, Amsterdam, The Netherlands.
- Bouwer, A., Bredeweg, B.** (2001) VisiGarp: Graphical Representation of Qualitative Simulation Models. (in book) *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future*. (eds) J.D. Moore, G. Luckhardt Redfield, and J.L. Johnson, pages 294-305, 2001, IOS-Press/Ohmsha, Japan, Osaka.
- Bouwer, A.** (2005) Explaining Behaviour - Using Qualitative Simulation in Interactive Learning Environments, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Bredeweg, B.** (1992) Expertise in qualitative prediction of behaviour. PhD thesis, University of Amsterdam.
- Bredeweg, B., Salles, P., en Neumann, M.** 2006. Ecological Applications of Qualitative Reasoning. *Ecological Informatics, Scope, Techniques and Applications*, F. Recknagel (ed.), pages 15-47, 2nd Edition, Springer, Berlin.
- Bredeweg, B., Struss P.** (2003) Current Topics in Qualitative Reasoning. ....
- Bredeweg, B., Bouwer, A. en Liem, J.** (2005) Single-user QR model building and simulation workbench
- Dreyfus, H.** (1972) Symbol Sourcebook, an Authoritative Guide to International Graphic Symbols.
- Engelhardt, Y.** (2002) The Language of Graphics – A Framework for the analysis of syntax and meaning in maps, charts and diagrams, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Elsom- Cook, M.** (2001) Principles of Interactive Multimedia, McGraw-Hill Publishing Company.
- Forbus, K. D.** (1984). Qualitative Process Theory. *Artificial Intelligence*, 24, 85-168.
- Forbus, K. D.** (1996) Qualitative Reasoning. (in book) *CRC Handbook of Computer Science*. CRC Press.
- Haverort, N.** (2000). Graphical Representation of Qualitative Simulations, Master thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Hora, M.** (1996) Official Signs & Icons. Ultimate Symbol.
- Jellema, J.** (2000) Ontwerpen voor Ondersteuning - De rol van taakkennis bij ondersteuningsontwerp, Master thesis, University of Amsterdam, Amsterdam, The Netherlands
- Kleer, J. de, Brown, J. S.** (1984). A Qualitative Physics Based on Confluences. *Artificial Intelligence*, 24, 7-83.
- Kulpa, Z.** (1994). Diagrammatic representation and reasoning. *Machine GRAPHICS & VISION* 3(1/2), (1994), 77-103.
- Machado, V. B.** (2004) Supporting the Construction of Qualitative Models, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Norman, D.** (1993) Things that make us smart. Addison-Wesley.
- Preece, J.** (1994) Human-Computer Interaction. Addison-Wesley.
- Ruisch, P. en Theys, E.** (1998) Pictogrammen & Iconen. [Z]OO Productions.
- Tufte, E. R.** (1990) Envisioning Information. Graphics Press.
- Tufte, E. R.** (2001) The Visual Display of Quantitative Information. Graphics Press.

## 10 Bijlagen

### 10.1 Bijlage 1: Vragenlijst

Investigating the meaning of icons  
*University of Amsterdam – June 25<sup>th</sup>, 2007*

In computer software icons are often used to create a graphical interface. The icons refer to actions the software will perform when pressing the icon with the mouse (thus by clicking on the icon). With this questionnaire we want to investigate the meaning of icons. You are therefore asked to select the most appropriate meaning for each icon enumerated below. You should select your answer out of the four options given for each icon. Give your answers by writing/typing '1', '2', '3', or '4' in the boxes on the right hand side.

Be aware that we will be using these icons for building and simulating qualitative models. The following main tasks are typical for that:

- File manipulations (opening, saving, etc.)
- Building the model
- Simulating the model
- Inspecting the simulation results

Notice that answers are 'good' or 'wrong'. The goal is to find out the ideas people associate to icons.

Thank you very much for participating!

*Please fill out:*

Name:


---

Date:

---


*Choose one answer for each of the following questions:*

Question 1


Icon	What action do you associate with this icon?	Your answer:
	Delete selected item Remove above Undo changes Layout list	




Question 2

Icon	What action do you associate with this icon?	Your answer:
	Open model fragments editor Find successors for selected states Save current model to a different file Start a new model	


Question 3

Icon	What action do you associate with this icon?	Your answer:
	Open quantity space editor Undo changes Open configurations definitions editor Split interval	


Question 4

Icon	What action do you associate with this icon?	Your answer:
	Open model fragments editor Draw Add a quantity Close model fragments editor	


Question 5

Icon	What action do you associate with this icon?	Your answer:
	Add selected item to list Add value low Open agent hierarchy editor Close current model	


Question 6

Icon	What action do you associate with this icon?	Your answer:
	Show item properties Simulate 'current' scenario Full simulation 'current' scenario Select all items	


Question 7

Icon	What action do you associate with this icon?	Your answer:
	Show value history Open assumption hierarchy editor Show transition history List by quantities	


Question 8

Icon	What action do you associate with this icon?	Your answer:
	Open quantity definitions editor Select a path Select all states Layout hierarchy: vertical	


Question 9

Icon	What action do you associate with this icon?	Your answer:
	Full simulation 'current' scenario Open model fragment editor Clear screen Select a scenario to simulate	


Question 10

Icon	What action do you associate with this icon?	Your answer:
	Terminate selected states Open model fragment editor Change layout entities Find successors for selected states	


Question 11

Icon	What action do you associate with this icon?	Your answer:
	Select all items Add new item Open model from file Save changes to model	


Question 12

Icon	What action do you associate with this icon?	Your answer:
	Start a new model Copy selected item Combine intervals Show long names	


Question 13

Icon	What action do you associate with this icon?	Your answer:
	Show transition history Save changes to model Cancel changes Clear screen	


Question 14

Icon	What action do you associate with this icon?	Your answer:
	Zoom out Zoom in Save changes to model Open de simulator to its current state or a saved simulation	


Question 15

Icon	What action do you associate with this icon?	Your answer:
	Change layout entities List by entities Clear screen List by quantities	


Question 16

Icon	What action do you associate with this icon?	Your answer:
	Print diagram to file (postscript) Show dependencies Add value high Combine intervals	


Question 17

Icon	What action do you associate with this icon?	Your answer:
	Open quantity definitions editor Copy selected item Start a new model Close window	


Question 18

Icon	What action do you associate with this icon?	Your answer:
	Delete value low Show dependencies Delete dependencies Delete value high	


Question 19

Icon	What action do you associate with this icon?	Your answer:
	Show model fragment in context Select all states Layout hierarchy: vertical Select all items	


Question 20

Icon	What action do you associate with this icon?	Your answer:
	Add value high Delete quantities List by quantities Change layout quantities	


Question 21

Icon	What action do you associate with this icon?	Your answer:
	Clear screen Start a new model Terminate selected states Add new item	


Question 22

Icon	What action do you associate with this icon?	Your answer:
	Show quantity values Clear screen Open quantity space editor Select all items	


Question 23

Icon	What action do you associate with this icon?	Your answer:
	Find successors for selected states Add new item Edit current / selected scenario Open quantity space editor	


Question 24

Icon	What action do you associate with this icon?	Your answer:
	Find successors for selected states Cancel changes Save changes to model Start a new model	


Question 25

Icon	What action do you associate with this icon?	Your answer:
	Delete value high Add value high Add selected item to list Split interval	


Question 26

Icon	What action do you associate with this icon?	Your answer:
	Show long names Add selected item to list Draw Close current model	


Question 27

Icon	What action do you associate with this icon?	Your answer:
	Delete value low Terminate selected states Delete value high Add new item	


Question 28

Icon	What action do you associate with this icon?	Your answer:
	Edit current / selected scenario Start a new model Show transition history Find successors for selected states	


Question 29

Icon	What action do you associate with this icon?	Your answer:
	Save changes to model Delete quantity Show quantity values Save changes to model	


Question 30

Icon	What action do you associate with this icon?	Your answer:
	Add new item Zoom in Open quantity space editor Close window	


Question 31

Icon	What action do you associate with this icon?	Your answer:
	Change layout quantities Start a new model Clear screen Close current model	


Question 32

Icon	What action do you associate with this icon?	Your answer:
	Delete value low Add value high Find successors for selected states Cancel changes	


Question 33

Icon	What action do you associate with this icon?	Your answer:
	Layout hierarchy: vertical Terminate selected states Show model fragment in context Open assumption hierarchy editor	


Question 34

Icon	What action do you associate with this icon?	Your answer:
	Open quantity definitions editor Select a path Select all states Show dependencies	


Question 35

Icon	What action do you associate with this icon?	Your answer:
	Open quantity definitions editor Combine intervals Show long names Zoom in	


Question 36

Icon	What action do you associate with this icon?	Your answer:
	Cancel changes Clear screen Show dependencies Print diagram to file (postscript)	


Question 37

Icon	What action do you associate with this icon?	Your answer:
	Change layout entities Open model from file Zoom in List by entities	


Question 38

Icon	What action do you associate with this icon?	Your answer:
	Copy selected item Find successors for selected states Start a new model Open de simulator to its current state or a saved simulation	


Question 39

Icon	What action do you associate with this icon?	Your answer:
	Open model from file Add selected item to list Show transition history Select a path	

Question 40


Icon	What action do you associate with this icon?	Your answer:
	Start a new model Terminate selected states Clear screen Close current model	

Question 41


Icon	What action do you associate with this icon?	Your answer:
	Select all items Select a scenario to simulate Open assumption hierarchy editor Find successors for selected states	




Question 42

Icon	What action do you associate with this icon?	Your answer:
	Show item properties Open agent hierarchy editor Show quantity values Open model fragment editor	


Question 43

Icon	What action do you associate with this icon?	Your answer:
	Select a scenario to simulate Select a path Full simulation 'current' scenario Add new item	


Question 44

Icon	What action do you associate with this icon?	Your answer:
	Add value low Open quantity definitions editor Change layout quantities Select all items	


Question 45

Icon	What action do you associate with this icon?	Your answer:
	Select a scenario to simulate Draw Open assumption hierarchy editor Show details	


Question 46

Icon	What action do you associate with this icon?	Your answer:
	Show item properties Save current model to a different file Show value history Add new item	


Question 47

Icon	What action do you associate with this icon?	Your answer:
	Open configurations definitions editor Add value low Terminate selected states Delete value low	


Question 48

Icon	What action do you associate with this icon?	Your answer:
	Clear screen Select all states Draw Delete selected item	

Question 49

Icon	What action do you associate with this icon?	Your answer:
	Combine intervals Remove above Split interval Open quantity definitions editor	

Question 50

Icon	What action do you associate with this icon?	Your answer:
	Edit current / selected model fragment Cancel changes Close model fragments editor Layout list	

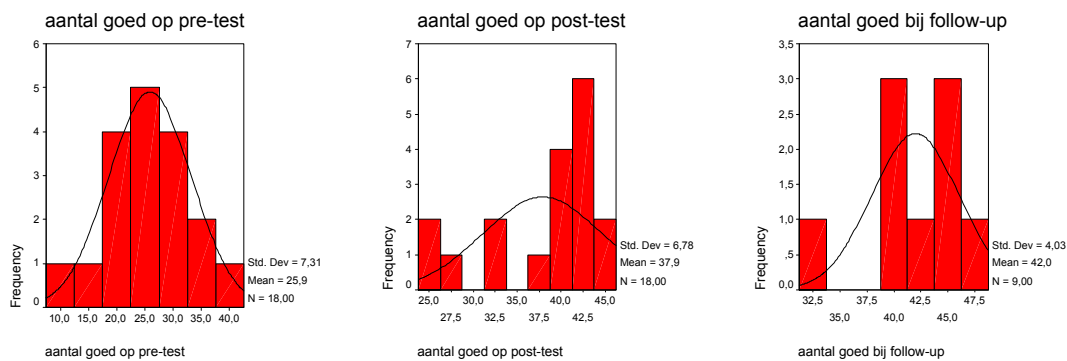
## 10.2 Bijlage 2: Statistische analyse

### Frequencies

Statistics

		aantal goed op pre-test	aantal goed op post-test	aantal goed bij follow-up
N	Valid	18	18	9
	Missing	0	0	9
Mean		25,8889	37,8889	42,0000
Std. Deviation		7,30744	6,78137	4,03113
Skewness		-,060	-1,136	-1,398
Std. Error of Skewness		,536	,536	,717
Kurtosis		-,382	,078	2,852
Std. Error of Kurtosis		1,038	1,038	1,400

### Histogram



### NPar Tests

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum
Pretest	18	25,89	7,307	12	39
Posttest	18	37,89	6,781	24	45
Followup	9	42,00	4,031	33	47

## Wilcoxon Signed Ranks Test

**Ranks**

		N	Mean Rank	Sum of Ranks
Posttest - Pretest	Negative Ranks	0(a)	,00	,00
	Positive Ranks	18(b)	9,50	171,00
	Ties	0(c)		
	Total	18		
Followup - Posttest	Negative Ranks	2(d)	2,25	4,50
	Positive Ranks	5(e)	4,70	23,50
	Ties	2(f)		
	Total	9		
Followup - Pretest	Negative Ranks	0(g)	,00	,00
	Positive Ranks	9(h)	5,00	45,00
	Ties	0(i)		
	Total	9		

- a Posttest < Pretest  
 b Posttest > Pretest  
 c Posttest = Pretest  
 d Followup < Posttest  
 e Followup > Posttest  
 f Followup = Posttest  
 g Followup < Pretest  
 h Followup > Pretest  
 i Followup = Pretest

**Test Statistics(b)**

	Posttest - Pretest	Followup - Posttest	Followup - Pretest
Z	-3,728(a)	-1,609(a)	-2,677(a)
Asymp. Sig. (2-tailed)	,000	,108	,007

- a Based on negative ranks.  
 b Wilcoxon Signed Ranks Test