

# Het creëren van inzicht in multi-user applicaties door visualisatie van inzichtsinformatie

Willemijn van Poppel  
Studentnummer: 9657177  
augustus 2004

Doctoraalscriptie  
Kunstmatige Intelligentie

Begeleiding: Bert Bredeweg

Faculteit der Natuurwetenschappen, Wiskunde en Informatica  
Universiteit van Amsterdam  
Onderwijsinstituut Informatiewetenschappen  
Laboratorium Sociaal Wetenschappelijke Informatica  
Roeterstraat 15, 1018 WB Amsterdam

## **Samenvatting**

In deze scriptie wordt verslag gedaan van een onderzoek naar het creëren van inzicht bij gebruikers van multi-user applicaties. Uit de literatuur blijkt dat dit inzicht te bereiken is door visualisatie van informatie over de groepsactiviteit in de interface. De twee belangrijkste vragen die bij dit proces gesteld dienen te worden zijn: (1) welke informatie is relevant en (2) hoe kan die effectief gevisualiseerd worden in de interface? Er heeft een praktijkonderzoek plaatsgevonden naar de multi-user applicatie Keyworx om te onderzoeken in hoeverre de aspecten genoemd in de theorie op het gebruik van deze applicatie van invloed zijn. Uit het gebruikersonderzoek bleek dat de gebruikers wel degelijk behoefte hebben aan visualisatie van aanvullende informatie. Zij willen echter voornamelijk informatie over de bewerkte objecten zien en niet zozeer informatie over de andere gebruikers. Naar aanleiding van de wensen van de gebruikers is een prototype van een visualisatie gemaakt dat vervolgens aan enkele van hen ter evaluatie is voorgelegd. Het huidige prototype is een goed begin voor verdere ontwikkeling.

**Voorwoord**

De beginfase van dit onderzoek heeft plaatsgevonden bij Waag society / maatschappij voor oude en nieuwe media te Amsterdam (<http://www.waag.org>) waar ik stage heb gelopen van maart tot september 2003. Daar werd mij volledige vrijheid geboden om een eigen onderzoek op te zetten en vervolgens gebruik te maken van aanwezige personen als bron van kennis en als proefpersonen. Daarvoor wil ik met name Tom Demeyer en Sher Doruff bedanken die mij hierin de weg hebben gewezen. En natuurlijk wil ik Dirk, Pit, Isabelle, Tanja, Louise, Lodewijk en Fokke bedanken voor hun medewerking aan mijn gebruikersonderzoek en evaluatie zonder welke dit onderzoek onmogelijk was geweest. Bijzonder dankbaar ben ik Jos en Simon voor de hulpdeskfunctie die zij vervulden tijdens het implementeren van het prototype. Verder veel dank aan alle vrienden en familie om mij heen die mij deze tijd hebben ondersteunt en al mijn ups and downs hebben doorstaan. En natuurlijk dank aan Bert Bredeweg, voor de heldere begeleiding die ik nodig had, en het enorme geduld met mijn eigenwijsheid...!

## Inhoudsopgave

<b>1 Inleiding</b>	6
1.1 Opbouw van deze scriptie	6
<b>2 Inleiding interface design en HCI</b>	7
2.1 Wat is een interface?	7
2.2 Wat is HCI?	7
2.3 Verschillende taakniveaus	7
2.4 Conceptualiseren van interactie	8
2.4.1 Conceptuele modellen gebaseerd op activiteiten	9
2.4.1.1 Instrueren	9
2.4.1.2 Converseren	10
2.4.1.3 Manipuleren en navigeren	11
2.5 conclusie	13
<b>3 Multi-user applicaties</b>	14
3.1 Wat is groupware?	14
3.2 Aandachtspunten groupware	15
3.2.1 Sociale factoren	15
3.2.2 Conflictbeheer	16
3.2.3 Groepsinterfaces	17
3.3 Inzicht	18
3.4 Conclusie	19
<b>4 Visualisatie</b>	20
4.1 Kennisrepresentatie	20
4.1.1 Diagrammatische representatie	21
4.1.2 Visuele talen	21
4.1.3 Een voorbeeld	22
4.2 Invloeden op visualisatie	23
4.3 Veel informatie in één afbeelding	24
4.3.1 Micro / macro ontwerpen	24
4.3.2 Aanbrengen van lagen	24
4.3.3 Vermenigvuldigen van afbeeldingen	25
4.3.4 Gebruik van kleur	25

4.3.5 Weergeven van beweging en tijd	26
4.4 Visuele eigenschappen	27
4.5 Visualisatie technieken	28
4.6 Vervormingstechnieken	28
4.7 Conclusie	30
<b>5 Visualisatie in multi-user systemen</b>	<b>31</b>
5.1 Het kennisgevingsproces	31
5.2 Selectie	32
5.2.1 Informatie oorsprong	32
5.2.2 Relevantie	32
5.3 Presentatie	33
5.3.1 Statische en dynamische aspecten	34
5.3.2 Medium	34
5.3.3 Tijdelijke vervorming	34
5.3.4 Ruimtelijke vervorming	35
5.3.5 Presentatiemethoden	36
5.4 Conclusie	37
<b>6 Keyworx</b>	<b>38</b>
6.1 Wat is Keyworx?	38
6.2 Keyworx opbouw	38
6.3 Een keyworx sessie	41
6.4 Gebruikersonderzoek - opzet	45
6.5 Analyse gebruikersonderzoek – video's	46
6.5.1 Gebruiker A	46
6.5.2 Gebruiker B	46
6.5.3 Gebruiker C	47
6.5.4 Gebruiker D	48
6.5.5 Gebruiker E	49
6.5.6 Gebruiker F	49
6.5.7 Gebruiker G	50
6.5.8 Alle gebruikers	50
6.5.9 Verdere analyse gebruikersonderzoek – video's	57
6.6 Analyse gebruikersonderzoek – vragen achteraf	58
6.7 Keyworx volgens de theorie	59

6.8 Conclusie	60
<b>7 Prototype</b>	<b>62</b>
7.1 Ontwerp	62
7.2 Implementatie	64
7.3 Evaluatie	65
7.4 Conclusie	66
<b>8 Conclusie</b>	<b>68</b>
8.1 Toekomstig onderzoek	69
<b>Bibliografie</b>	<b>70</b>
<b>Bijlage 1</b> Vragen vooraf	<b>72</b>
<b>Bijlage 2</b> Vragen achteraf	<b>73</b>
<b>Bijlage 3</b> Antwoorden op vragen achteraf	<b>75</b>
<b>Bijlage 4</b> Vragen over het prototype	<b>76</b>

## 1 Inleiding

Door het gebruik van internet en mobiele telefoon hebben mensen steeds meer de mogelijkheid om over grote afstand en op ieder moment met elkaar te communiceren. Grenzen en tijdszones worden hierdoor onbelangrijk, fysieke nabijheid is niet meer noodzakelijk. In de toekomst zullen mensen steeds meer gebruik gaan maken van computer, mobiele telefoon, PDA en andere apparaten om met elkaar te communiceren of samen te werken. De meeste bestaande computerapplicaties laten echter één persoon met zijn computer werken, ze zijn niet geschikt voor gebruik door meerdere personen tegelijk, zoals zogenaamde multi-user applicaties. Aangezien de behoefte aan deze applicaties steeds groter wordt, zullen de huidige methodes uit de systeemontwikkeling uitgebreid en aangepast moeten worden om wel te voldoen aan deze nieuwe behoefte.

Wanneer meerdere personen vanaf verschillende lokaties optimaal willen samenwerken, zullen zij inzicht moeten hebben in het groepsproces en de andere gebruikers (o.a. Berlage en Sohlenkamp, 1999; Dourish en Bellotti, 1992). Dit kan bereikt worden door het aanbieden van informatie over de andere personen en het systeem, zoals wat de verschillende gebruikers aan het doen zijn, wat er op dat moment allemaal gaande is. Deze informatie wordt ook wel inzichtsinformatie genoemd. Om de gebruikers niet teveel af te leiden van hun werk, moet de wijze waarop deze informatie aangeboden wordt discreet en niet te opvallend zijn. Er bestaan verschillende methodes voor. In dit onderzoek wordt visualisatie in de interface nader bekeken. De centrale vraag van dit onderzoek luidt: welke informatie over de groepsactiviteit moet in de interface van een multi-user applicatie worden gevisualiseerd om beter inzicht te creëren bij de gebruikers? En hoe moet deze visualisatie vervolgens vormgegeven worden?

De uitvoering van dit onderzoek is als volgt opgezet. Om te beginnen is een literatuurstudie uitgevoerd om een theoretisch kader te verschaffen van de materie en inzicht in de algemene kwesties die spelen bij het gebruik van multi-user applicaties. Hierbij gaat het om literatuur over mens-computer interactie en interface-design in het algemeen, over multi-user applicaties of groupware en samenwerking via de computer, zogenaamde computer supported cooperative work, over het belang van inzicht en hoe dat te creëren door middel van visualisatie, oftewel data-visualisatie. De volgende stap in dit onderzoek richt zich op een praktijksituatie. Hiervoor is gekozen voor de applicatie Keyworx. Dit is een multi-user applicatie die door Waag society / maatschappij voor oude en nieuwe media (Amsterdam) is ontwikkeld. Het is een DJ/VJ tool waarmee beeld, geluid en tekst door meerdere mensen tegelijk vanaf verschillende lokaties gemanipuleerd kan worden. Er is een gebruikersonderzoek uitgevoerd om te achterhalen of er problemen veroorzaakt worden door het multi-user aspect van Keyworx en zo ja, wat die dan zijn. Vervolgens is onderzocht welke informatiebehoefte vervuld moet worden om deze problemen op te lossen en hoe deze informatie dan te visualiseren. De laatste stap van dit onderzoek is het ontwikkelen van een prototype waarin de informatievoorziening wordt geïmplementeerd om deze informatievoorziening vervolgens te kunnen evalueren.

### 1.1 Opbouw van deze scriptie

Hoofdstuk 2 geeft een inleiding in interface ontwerp en onderzoek naar mens-computer interactie. Het gebruik van conceptuele modellen in een ontwerpproces wordt beschreven. Hoofdstuk 3 beschrijft multi-user applicaties. Wat zijn de mogelijkheden van multi-user applicaties en welke specifieke issues ontstaan daardoor? Het belang van inzicht in de activiteiten van anderen voor gebruikers van groupware wordt besproken. Hoofdstuk 4 gaat over visualisatie. Verschillende manieren om informatie in één afbeelding te representeren worden besproken. Hoofdstuk 5 bespreekt visualisatie in multi-user systemen door middel van het kennisgevingsproces waarin twee cruciale stappen te onderscheiden zijn, namelijk informatie-selectie en -presentatie. Hoofdstuk 6 richt zich op de applicatie Keyworx. De werking en mogelijkheden ervan worden uitgelegd aan de hand van een taakanalyse. De in het gebruikersonderzoek verzamelde data wordt geanalyseerd. Hoofdstuk 7 gaat over het prototype. Waarom worden bepaalde acties gevisualiseerd en hoe wordt dat gedaan? De mening van de Keyworx gebruikers over het prototype wordt hier besproken. Hoofdstuk 8 is de conclusie van deze scriptie met bovendien suggesties voor toekomstig onderzoek.

## 2 Inleiding interface design en HCI

### 2.1 Wat is een interface?

In dit onderzoek worden gebruikersinterfaces bekeken. Volgens webopedia is een gebruikersinterface: “de verbinding tussen een gebruiker en een computerprogramma. Een interface is een set van commando’s of menu’s door welke een gebruiker met een programma communiceert” (<http://www.webopedia.com>).

Anders gezegd bestaat een gebruikersinterface uit zowel de muis, het toetsenbord en de joystick, als uit de keuzemenu’s, iconen, commandolijnen, touch screens, etcetera. Dit alles stelt de gebruiker in staat met het besturingssysteem van de computer en de op de computer geïnstalleerde applicaties te communiceren (Simons, 2002).

In het vervolg van dit stuk, wordt de term ‘interface’ in de betekenis van ‘gebruikersinterface’ gebruikt. De interface is een van de belangrijkste onderdelen van een computerapplicatie, omdat het bepaald hoe eenvoudig een applicatie bediend kan worden. Een uitgebreide computerapplicatie met een slecht ontworpen interface heeft weinig nut. De interface vormt hoe de computergebruiker de hele computer bekijkt (Manovich, 2001). Of zoals Norman zegt: “vanuit de gebruiker gezien is de interface het systeem” (Norman, 1986).

De meeste computerapplicaties worden gebruikt door één enkele gebruiker. Er bestaan echter ook applicaties waarmee meerdere mensen tegelijk kunnen (samen)werken. Zij kunnen dan bijvoorbeeld tegelijkertijd een tekstbestand bewerken, ieder van achter een eigen computer. Deze applicaties worden multi-user applicaties genoemd, en de interface waar zij gebruik van maken is een multi-user interface.

### 2.2 Wat is HCI?

HCI is de afkorting van human-computer interaction. Dit is een discipline die zich bezighoudt met het bestuderen, ontwerpen en implementeren van interactieve computer systemen voor menselijk gebruik. De interactie tussen mens en computer verloopt via de gebruikersinterface, maar HCI gaat verder dan het vormgeven van schermen en menu's; het bestudeert de redenering achter specifieke functionaliteiten van computers en de lange-termijn effecten die deze systemen zullen hebben op mensen (<http://www.webopedia.com>).

HCI speelt een rol bij het ontwerpen van allerlei verschillende soorten systemen, van luchtverkeersleiding waarbij veiligheid voorop staat, tot kantoorssystemen (office systems) waarbij productiviteit en voldoening het voornaamste doel zijn, tot computerspellen die plezier moeten opleveren. Het doel van HCI is onderzoek doen naar en het verbeteren van de kwaliteit van de interactie tussen mens en machine. Om dit te bereiken, is kennis over computertechnologie noodzakelijk. Verder is het essentieel om iets van menselijke (cognitieve) psychologie in het algemeen te begrijpen en van de specifieke karakteristieken van de gebruikers op de hoogte te zijn, zoals achtergrondkennis, leeftijd en sociale en organisatorische omgeving (o.a. Grudin, 1990 en 1993; Preece, 1994). De grote uitdaging van het ontwerpen van computers voor menselijk gebruik is het bedenken welke taken ondersteund kunnen worden (dat wil zeggen, de functionaliteit) en hoe dat kan gebeuren in overeenstemming met de gebruiker's behoeften (dat wil zeggen, bruikbaarheid). Bij het ontwikkelen van door mensen gebruikte computersystemen komt dus erg uiteenlopende kennis aan bod, uit de sociale wetenschappen, informatica en vormgeving.

### 2.3 Verschillende taakniveaus

Binnen een applicatie zijn als het ware twee niveau's te onderscheiden waar de HCI zich mee bezig houdt. Aan de ene kant staat het niveau van acties die uitgevoerd kunnen worden, aan de andere kant het niveau van de objecten waarop die acties uitgevoerd worden.

Op het eerste niveau moet een applicatie bepaalde functionaliteiten aanbieden om een gebruiker te ondersteunen bij het uitvoeren van taken. De vragen die hier van belang zijn, zijn: welke taken worden ondersteunt, welke manipulaties kunnen worden uitgevoerd, en hoe wordt dat gedaan? Kan een gebruiker bijvoorbeeld van een plaatje de achtergrondkleur veranderen, en zo ja, hoe gebeurt dat, welke (deel)handelingen moeten daarvoor verricht worden? Het verschil tussen single- en



multi-user applicaties ligt op dit niveau voor de hand: een multi-user applicatie moet extra functionaliteit bieden om samenwerking tussen meerdere personen te ondersteunen.

Op het tweede niveau speelt zich de informatievoorziening af. Hierbij gaat het om de objecten die gemanipuleerd kunnen worden; hoe worden die gevisualiseerd? Het is nodig om informatie aan te bieden in de interface, zodat een gebruiker kan zien wat de input of output van een handeling kan zijn. Een voorbeeld van zo'n object is het eerder genoemde plaatje, of alleen de achtergrondkleur. Het gaat hierbij niet alleen om weergave van de verschillende objecten, maar ook om informatie over hoe een actie verloopt en wat de applicatie op dat moment aan het doen is. Hierdoor krijgt de gebruiker inzicht in wat er gebeurt en een idee van de gevolgen van zijn acties, zodat een taak beter uitgevoerd kan worden (o.a. Norman, 2002). Dit is bij zowel single- als multi-user applicaties van belang. Wat er wordt afgebeeld en hoe, hangt af van de taak die met de applicatie wordt uitgevoerd. Het zoeken in Google gebeurt bijvoorbeeld in een statisch weergegeven interface, terwijl een luchtverkeersleider in de door hem gebruikte interface de bewegingen van vliegtuigen kan bekijken, en zodoende met een dynamische visualisatie in de interface werkt.

Wanneer er sprake is van een multi-user applicatie, is het ook nodig informatie aan te bieden over de samenwerking, over de verschillende gebruikers en hun acties in de gedeelde werkruimte, zodat de eigenlijke samenwerking beter kan verlopen (o.a. Berlage en Sohlenkamp, 1999).

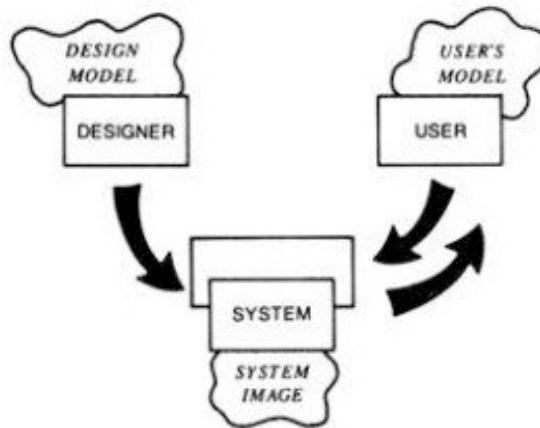
In de interface komen de twee niveau's samen. Aan de ene kant kan de gebruiker hierin de applicatie bedienen, via menu's en knoppen die bepaalde acties tot gevolg hebben. Aan de andere kant wordt informatie, feedback, aangeboden over wat er op dat moment gebeurt, welke objecten gemanipuleerd worden, door bijvoorbeeld direct de achtergrondkleur van het zichtbare plaatje te veranderen.

#### **2.4 Conceptualiseren van interactie**

Een goede manier om met het ontwerpen van een nieuwe applicatie te beginnen, is om te pogen de aard van de probleemruimte te begrijpen, voordat er iets gebouwd gaat worden. Hiermee wordt bedoeld het conceptualiseren van wat gecreëerd moet worden en goed uitdrukken waarom dit zo is. Dit vereist het bedenken van hoe het ontwerp mensen in hun dagelijks leven of werk kan ondersteunen. In het bijzonder moet bedacht worden of het beoogde produkt zal bereiken wat gehoopt wordt. Dit kan later veel tijd en moeite besparen (Preece, 2002). Wanneer dit namelijk niet gebeurt, kunnen kritieke bruikbaarheidsdoelen en gebruikers behoeften over het hoofd worden gezien. Het verhelderen van de probleemruimte geschiedt door het ontwikkelen van een conceptueel model.

Met conceptueel model wordt bedoeld (Preece, 2002): "een beschrijving van het voorgestelde systeem in termen van een set van geïntegreerde ideeën en concepten waarin staat wat het zou moeten doen, hoe het zich zou moeten gedragen en hoe het eruit moet zien, zodat het begrijpelijk is voor beoogde gebruikers."

Een goed conceptueel model stelt ons in staat te voorspellen wat de effecten van onze acties zullen zijn. Zonder goed model functioneren mensen machinaal, blind; we voeren handelingen uit zoals ons verteld wordt; we kunnen niet geheel bevatten waarom, welke effecten te verwachten, of wat te doen als dingen fout gaan. Zolang alles goed werkt, kunnen we het aan. Als dingen echter misgaan, of wanneer we op een nieuwe situatie stuiten, hebben we een dieper begrip nodig (Norman, 2002). Voor dagelijkse gebruiksvoorwerpen hoeven conceptuele modellen niet erg ingewikkeld te zijn. Tenslotte zijn scharen, pennen, en lichtschakelaars vrij eenvoudige apparaten. Er is geen noodzaak de preciese werking van elk apparaat dat we gebruiken te begrijpen, de relatie tussen de bediening en de uitkomsten daarvan volstaat. Norman (Norman, 2002) heeft een kader opgesteld om de relatie tussen het ontwerp van een conceptueel model en een gebruikers begrip daarvan duidelijk te maken (zie figuur 2.1).



figuur 2.1: conceptuele modellen (uit: Norman, 2002)

Er zijn drie interacterende componenten aanwezig: de ontwerper, de gebruiker en het systeem. Deze bezitten drie bijbehorende conceptuele modellen. Het ontwerpmodel is het conceptuele model dat de ontwerper heeft van hoe het systeem zou moeten werken. Het gebruikersmodel is het model dat de gebruiker ontwikkeld heeft door interactie met het systeem, het conceptualiseert hoe het systeem volgens hem werkt. Het systeembeeld komt voort uit de fysieke structuur die gebouwd is (inclusief documentatie en instructies), het is het zichtbare gedeelte van een apparaat. De ontwerper verwacht dat het gebruikersmodel identiek is aan het ontwerpmodel. Maar de ontwerper communiceert niet rechtstreeks met de gebruiker, alle communicatie verloopt via het systeembeeld. Gebruikers zouden in staat moeten zijn hun taken uit te voeren op de door de ontwerper bedoelde manier, door te interacteren met het systeembeeld, welke duidelijk maakt wat te doen. Als het systeembeeld het ontwerpmodel echter niet duidelijk maakt aan de gebruikers, is het waarschijnlijk dat deze een onjuist conceptueel model ontwikkelen, waardoor ze een verkeerd beeld hebben van de werking van het systeem, waardoor ze vervolgens het systeem inefficiënt gaan gebruiken en fouten gaan maken.

#### 2.4.1 Conceptuele modellen gebaseerd op activiteiten

Er bestaan verschillende uitgangspunten om een conceptueel model op te stellen. Eén methode gaat uit van de verschillende soorten acties die een gebruiker kan verrichten.

De meest algemene typen activiteiten die een gebruiker van een systeem uitvoert zijn (Preece, 2002):

1. instrueren
2. converseren
3. manipuleren en navigeren

Een eerste opmerking hierbij is dat deze activiteiten niet exclusief zijn, aangezien ze tegelijk uitgevoerd kunnen worden. Ze hebben echter wel specifieke eigenschappen en vereisen een andere manier van omgang.

##### 2.4.1.1 Instrueren

Dit soort conceptueel model beschrijft hoe gebruikers hun taken uitvoeren door het systeem op te dragen wat te doen. Dit kan gebeuren door verschillende manieren van interactie: intypen van commando's, selecteren van opties in menu's in een windows-omgeving of op een touch screen, hardop uitspreken van commando's, indrukken van knoppen, of gebruik maken van een combinatie van functietoetsen. Hierdoor wordt het systeem geïnstrueerd om operaties uit te voeren als tijdlezen, een bestand printen, of de gebruiker herinneren aan een afspraak.

Besturingssystemen als Unix en DOS zijn command-based systemen, waar de gebruiker instructies geeft door commando's in te typen achter een prompt. In Windows en andere grafische user interface-systemen worden instructies gegeven via functietoetsen of het selecteren van menu-opties

met een muis. Deze grafische user interfaces (GUI's) zijn opgebouwd volgens het zogenaamde WIMP-principe, wat staat voor windows, icons, menus and pointers. Command-based en andere instructie-gevende systemen kunnen dus variëren in de vorm van de commando's (bijvoorbeeld het gebruik van afkortingen, volledige namen, iconen, en/of labels), hun syntax (hoe verschillende commando's het beste te combineren), en hun organisatie (bijvoorbeeld hoe opties in verschillende menu's te structureren).

Een van de voordelen van een instructie-gebaseerd conceptueel model is dat het snelle en efficiënte interactie ondersteunt. Het is vooral geschikt voor zich herhalende acties uitgevoerd op meerdere objecten. Voorbeelden vormen de zich herhalende acties als bewaren, weggooien, en organiseren van email berichten of bestanden.

#### **2.4.1.2 Converseren**

Dit conceptueel model is gebaseerd op het idee van een persoon die met een systeem converseert, waarbij het systeem functioneert als gesprekspartner. In het bijzonder is het systeem ontworpen om te reageren zoals een persoon zou doen in een conversatie met iemand anders. Gebruikers praten tegen het systeem of typen vragen in, waarop het systeem reageert middels tekst- of spraakoutput. Het verschil met de vorige categorie van instrueren is de bedoeling een tweerichtings communicatie proces weer te geven, waarbij het systeem meer optreedt als partner dan als een machine die simpelweg bevelen opvolgt (Simons, 2002).

Dit soort conceptueel model is het meest bruikbaar voor applicaties waarin de gebruiker specifieke informatie moet uitvinden of kwesties wil bespreken (Preece, 2002). Voorbeelden zijn adviessystemen, help faciliteiten, en zoekmachines. De ondersteuning bij dit soort conversaties loopt uiteen van simpele stem-herkennings menu-gebaseerde systemen, die via telefoon bedient worden, tot complexere natuurlijke-taal-gebaseerde systemen waarbij het systeem ingetypte gebruikersvragen ontleedt en daarop reageert. Voorbeelden van de eerste zijn bankieren, kaartjes reserveren, en treininlichtingen, waarbij de gebruiker in enkele-woord zinnen (zoals 'ja', 'nee', 'drie') tegen het systeem praat in reactie op vragen die worden gesteld. Voorbeelden van de laatste zijn zoekmachines en helpsystemen, waarbij de gebruiker een specifieke vraag intypt waar het systeem vervolgens op reageert door meerdere antwoorden te geven.

Een groot voordeel van dit conceptueel model is dat het gebruikers, vooral beginners, toestaat met een systeem te interacteren op een wijze waaraan ze al gewend zijn. Een nadeel is echter dat er onbegrip kan ontstaan als het systeem op gestelde vragen geen antwoord kan geven op de manier die de gebruiker verwacht. Een ander probleem dat kan ontstaan is dat bepaalde soorten taken getransformeerd worden in omslachtige en eenzijdige interacties. Dit is vooral het geval bij geautomatiseerde, telefoon-gebaseerde systemen die auditieve menu's gebruiken om dieper in het systeem te komen. Gebruikers moeten eerst luisteren naar een stem die verschillende mogelijkheden aanbiedt, dan een selectie maken, en dit verschillende malen herhalen voordat hun doel bereikt wordt (bijvoorbeeld een echte persoon te spreken of het betalen van een rekening).

In deze conceptualisering wordt de computer beschouwd als partner, al dan niet gelijkwaardig. Dit heeft vaak geleid tot sciencefiction voorstellingen van supercomputers zoals HAL uit de film "2001: A Space Odyssey" (S. Kubrick, UK/USA, 1968). In minder futuristische vorm zijn ze te vinden als smart agents of bots die voor hun gebruikers nieuws selecteren, de temperatuur in huis reguleren of boodschappenlijsten samenstellen. Een recente ontwikkeling zijn geanimeerde agenten. Verschillende soorten karakters, uiteenlopend van 'echte' personen die in de interface verschijnen tot tekenfilm personages, zijn ontworpen om op te treden als partner in de conversatie met het systeem. Hierdoor is de gebruiker in staat de dialoogpartner te zien, te horen, en soms zelfs aan te raken, waarbij het lijkt alsof deze zich gedraagt en praat als een mens. Veel agenten zijn uitgerust met wenselijke menselijke eigenschappen, zoals vrolijkheid en enthousiasme, die tot uiting komen in gezichtsuitdrukkingen en levensechte fysieke bewegingen. Een nadeel van een fysieke aanwezigheid van gesprekspartners in de interface, is dat ze erg irritant kunnen worden, bijvoorbeeld door aanwijzingen te geven terwijl de gebruiker daar helemaal niet op zit te wachten. Volgens Preece (Preece, 2002) kunnen geanimeerde agenten die praten en beschikken over

menselijk of dierlijk gedrag, geloofwaardiger zijn. Het onderliggende conceptuele model wordt veel explicieter overgebracht wanneer het systeem via een zichtbare agent handelt en praat. Een voordeel zou zijn dat mensen eenvoudiger doorhebben dat de interface agent (of fysiek speelgoed) waarmee ze converseren geen mens is, maar een synthetisch karakter dat bepaalde menselijke karaktertrekken heeft meegekregen. Wanneer de gesprekspartner aan het oog is onttrokken, is het moeilijker te achterhalen wat erachter zit, en hoe intelligent dat eigenlijk is, waardoor een persoon kan denken dat het intelligenter is dan het in werkelijkheid is. Als de gesprekspartner dan niet in staat blijkt vragen of commentaar te begrijpen, verliezen gebruikers waarschijnlijk hun geduld. Bovendien zijn ze in zo'n geval waarschijnlijk minder vergevingsgezind dan bij een gesprekspartner die wordt gerepresenteerd als tekenfilm karakter in de interface (ervan uitgaande dat het een simpele partner was).

Daarentegen denkt Simons (Simons, 2002) dat zich in deze conceptualisering twee problemen voordoen. Volgens hem wordt de metaforische aard van de voorstelling over intelligente machines vaak over het hoofd gezien, en worden metaforen juist te letterlijk genomen. Wanneer gebruikers de metafoor te letterlijk nemen, kan dit ertoe leiden dat zij hun geanimeerde agent meer capaciteiten toedichten dan deze in werkelijkheid heeft, waardoor ze vervolgens teleurgesteld worden als blijkt dat deze niet aan hun verwachtingen voldoet. Bovendien gaan veel van de futuristische voorstellingen voorbij aan het gegeven dat computers nog lang niet de intelligente wezens zijn die in deze voorstellingen figureren. Hedendaagse computers hebben bijvoorbeeld nog niet de vermogens van HAL tot spraakherkennig, laat staan liplezen.

Deze twee problemen, gebruikers die de metafoor te letterlijk nemen en de te hoge verwachtingen van de huidige computercapaciteit, leiden er volgens Simons toe dat in plaats van als een hulpvaardige en intelligente partner, veel gebruikers de computer ervaren als een domme en eigengereide bediende die de instructies van zijn opdrachtgever niet kan of wil begrijpen en vaak over een beperkter repertoire aan vaardigheden en kennis beschikt dat de gebruiker lief is.

#### **2.4.1.3 Manipuleren en navigeren**

Dit conceptueel model beschrijft de activiteit van het manipuleren van objecten en navigeren door virtuele omgevingen door gebruik te maken van gebruikerskennis over hoe ze dat doen in de fysieke wereld. Virtuele objecten kunnen bijvoorbeeld gemanipuleerd worden door bewegen, selecteren, openen, sluiten, en in- of uitzoomen. Deze acties kunnen ook uitgebreid worden door bijvoorbeeld manipulaties en navigaties mogelijk te maken die in de fysieke wereld niet mogelijk zijn.

Een bekende instantie van dit conceptueel model is directe manipulatie. Volgens Ben Shneiderman, die deze term bedacht, beschikken directe manipulatie interfaces over drie fundamentele eigenschappen (uit: "interview with Ben Shneiderman" in Preece, 1994):

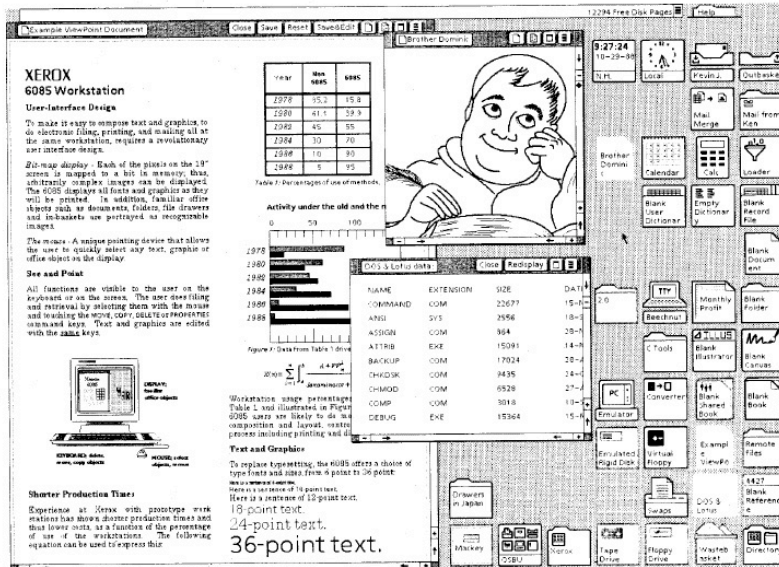
- continue representatie van de objecten en acties die interessant zijn
- snelle omkeerbare acties met onmiddellijke feedback over het object van interesse
- fysieke acties als selecteren en klikken in plaats van commando's geven met complexe syntax

Voordelen van een directe manipulatie interface zijn:

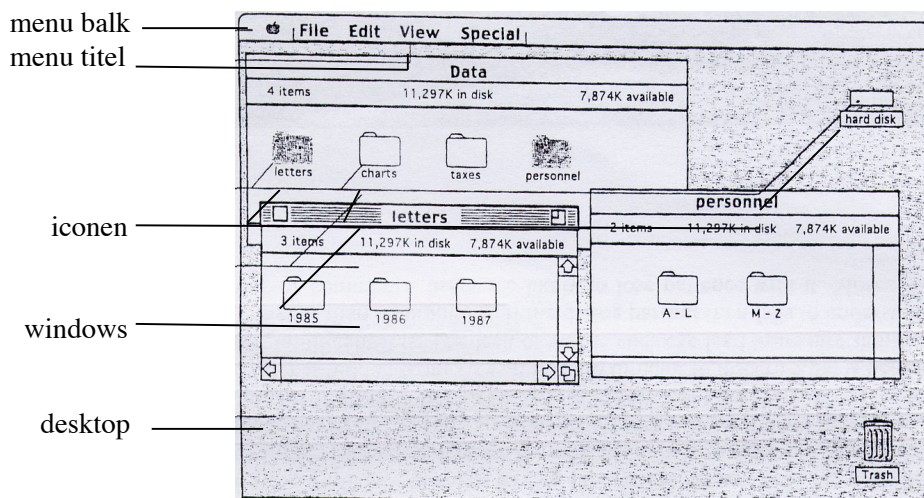
- helpt beginners de basisfunctionaliteit snel te leren
- ervaren gebruikers kunnen snel werken aan een breed scala van taken
- onregelmatige gebruikers kunnen onthouden hoe operaties uit te voeren
- vrijwel geen foutmeldingen nodig
- gebruikers kunnen meteen zien of hun acties hen dichterbij hun doel brengt
- gebruikers ervaren minder angst
- toename van gebruikers vertrouwen en gevoel van controle

In 1981 introduceerde Xerox het 8010 "Star" system (zie figuur 2.2), wat een revolutie betekende voor het interface-ontwerp voor personal computing. Hoewel het commercieel niet zo'n succes was, zijn veel van de achterliggende ideeën geleend en aangepast door andere bedrijven, zoals Apple en Microsoft, die vervolgens erg succesvol werden. Star was ontworpen als kantoorstelsel, bedoeld voor gebruikers die niet zozeer in het computeren zelf geïnteresseerd waren (Preece, 2002). Een belangrijk ontwerpdoel was dan ook om de computer zo onzichtbaar als mogelijk voor de

gebruikers te maken en om applicaties te ontwerpen die voor hen geschikt waren. De ontwerpers hadden gekozen voor een conceptueel model gebaseerd op een fysiek kantoor. Ze wilden dat de computergebruikers de computer zagen als een kantooromgeving, waar ze konden werken met elektronische equivalenten van wat ze kenden uit de fysieke wereld. De aanname was dat dit de elektronische wereld zou versimpelen en verduidelijken, waardoor het bekender overkwam, en eenvoudiger te leren zou zijn.



figuur 2.2: de Star interface



figuur 2.3: de originele Macintosh interface

Apple computers inc. was een van de eerste computerbedrijven die een besturingsomgeving ontwierp die gebruik maakte van directe manipulatie als voornaamste vorm van interactie. Zij introduceerden de succesvolle bureaublad (desktop)-metafoer, dat nog steeds het meest voorkomende type interface is (zie figuur 2.3) (Simons, 2002).

De succesvolle Apple Macintosh desktop demonstreert de belangrijkste principes van directe manipulatie. Om gebruik te maken van het begrip dat mensen hebben van wat er met fysieke objecten gebeurt in de echte wereld, gebruikten ze een aantal visuele en auditieve aanwijzingen in de interface die bedoeld waren om ze te evenaren. Een van hun aannames was dat mensen verwachten dat hun fysieke acties fysieke gevolgen hebben, zodat wanneer een tekentool wordt gebruikt, er een bijbehorende lijn verschijnt, en wanneer een bestand in de prullenbak wordt

gegooid, er een bijbehorende geluids- of visuele aanwijzing wordt gebruikt, waarmee wordt aangegeven dat de taak succesvol is uitgevoerd. Veel van dit ontwerp was erop gericht de gebruiker aanwijzingen te verschaffen zodat deze wist wat te doen, zich op zijn gemak voelde, en het werken met de interface als plezierig ervoer.

Veel andere soorten directe manipulatie interfaces zijn ontwikkeld, zoals videogames, data visualisatie tools en CAD systemen. Virtuele omgevingen en virtual reality hebben op dezelfde wijze gebruik gemaakt van een reeks van interactie mechanismen die de gebruiker in staat stellen te interacteren met en te navigeren door een gesimuleerde 3D fysieke wereld.

Hoewel directe manipulatie en virtuele omgevingen voorzien in een erg veelzijdige manier van interactie, zijn er ook een aantal nadelen aan verbonden. Op een conceptueel niveau kunnen sommige personen het conceptuele model misschien te letterlijk nemen en verwachten dat bepaalde dingen in de interface gebeuren zoals ze zouden verwachten in de fysieke wereld. Een bekend voorbeeld hiervan zijn nieuwe Mac gebruikers die bang zijn om het icoon van hun floppy of cd naar het prullenbak-icoon op de desktop te slepen om het te verwijderen, uit angst het daadwerkelijk weg te gooien. Deze verwarring ontstaat omdat de ontwerpers ervoor gekozen hebben dezelfde actie (slepen) op hetzelfde object (prullenbak) te gebruiken voor twee totaal verschillende acties, namelijk het verwijderen van media en het weggooien van bestanden.

Een ander nadeel aan deze metafoer is dat het ontwerpers vaak blind maakt voor de nieuwe mogelijkheden die computers bieden, ze zijn gefixeerd op bestaande, oude ideeën. Hierdoor dreigen computers meer van hetzelfde en, erger nog, imitaties van achterhaalde objecten en praktijken te worden. De bekende desktopinterface veronderstelt bovendien dat het kantoor de meest voor de hand liggende omgeving voor de computer en zijn gebruiker is. Inmiddels heeft de computer echter zijn weg naar huiselijke kring gevonden, waar deze niet alleen voor werk wordt gebruikt, maar ook voor amusement en informatievoorziening. Bovendien zijn er naast de desktopcomputers laptops, palmtops en PDA's (personal digital assistants), die de vaste werkplek mobiel hebben gemaakt. Wat betreft omgeving en functionaliteit is de computer dus allang niet meer voornamelijk aan het kantoor gebonden.

## **2.5 conclusie**

In de interactie tussen mens en computer wordt gebruik gemaakt van een interface, een representatie van het interne van de computer waar een persoon mee kan werken. Wanneer meerdere mensen tegelijkertijd met één applicatie werken, is er sprake van een multi-user applicatie met een multi-user interface. Bij het ontwerpen van elk computersysteem, dus ook een multi-user systeem, wordt gebruik gemaakt van kennis over human-computer interaction. Dit is een vakgebied dat onderzoek doet naar de interactie tussen mens en computer, en de opgedane kennis toepast om deze te verbeteren.

Binnen een applicatie zijn twee niveaus te onderscheiden waar de HCI zich mee bezig houdt, namelijk het niveau van mogelijke actie, en het niveau van de objecten waarop die uitgevoerd kunnen worden en de informatie die daarover aangeboden wordt.

Wanneer een nieuwe applicatie ontworpen wordt, wordt een conceptueel model gemaakt van de applicatie en zijn functionaliteiten. Er bestaan verschillende methoden om een conceptueel model op te stellen. Eén benadering baseert zich op de activiteiten die een gebruiker kan uitvoeren. Deze activiteiten zijn instrueren, converseren, en manipuleren en navigeren. Bij instrueren geeft een gebruiker een applicatie opdrachten om bepaalde taken uit te voeren. Bij converseren functioneert het systeem als gesprekspartner die reageert op de gebruiker, waardoor meer tweerichtings communicatie ontstaat. Het derde conceptueel model beschrijft de activiteit van het manipuleren van objecten en navigeren door virtuele omgevingen. Het gaat uit van gebruikerskennis over hoe dat gebeurt in de fysieke wereld, en wordt eventueel uitgebreid door manipulaties en navigaties te ondersteunen die in de fysieke wereld niet mogelijk zijn. Een bekende instantie van dit conceptueel model is directe manipulatie. Interfaces volgens dit principe beschikken over een continue representatie van interessante objecten en acties, omkeerbare acties met onmiddellijke feedback, en worden aangestuurd door fysieke acties als het slepen met bestanden. De huidige GUI-systemen zoals Apple Macintosh en Microsoft Windows maken hier nog steeds succesvol gebruik van.

### 3 Multi-user applicaties

#### 3.1 Wat is groupware?

Het gebruik van multi-user applicaties verloopt anders dan het gebruik van single-user applicaties, aangezien er sprake is van samenwerking. Het ontwerpen van multi-user applicaties ter ondersteuning van groepen en organisaties vereist gevoeligheid voor issues als het groepsproces, sociale conventies, rolverdeling en individualiteit binnen groepen en organisaties. De kennis en werkwijze van deze gebieden wordt gecombineerd met informatica, waardoor een nieuw multidisciplinair onderzoeksgebied is ontstaan (Grudin, 1994): *computer-supported cooperative work* (CSCW). Hierin wordt onderzocht hoe groepen functioneren en bekeken hoe technologie, en dan met name computers, hen bij het uitvoeren van werk kan ondersteunen.

CSCW producten worden vaak *groupware* genoemd. Ellis e.a. (Ellis e.a., 1991) definiëren de term *groupware* als “computergebaseerde systemen die ondersteuning bieden aan groepen mensen die verbonden zijn in een gezamenlijke taak (of doel) en die voorzien in een interface naar een gedeelde omgeving”. De begrippen ‘gezamenlijke taak’ en ‘gedeelde omgeving’ zijn hierin cruciaal. De mate van gezamenlijkheid en gedeeldheid kan nogal verschillen van geval tot geval. Groupware kan gebruikt worden om zowel groepen verspreid over verschillende lokaties te ondersteunen, als groepen die zich in één omgeving bevinden. Daarnaast kan het gebruikt worden bij real-time, synchrone communicatie, alsook bij asynchrone communicatie. Door dit te combineren ontstaan er vier gevallen waarin groupware van dienst kan zijn, zoals te zien is in figuur 3.1; face-to-face interactie, waarbij sprake is van een zelfde plaats en tijd; asynchrone interactie, waarbij sprake is van een gedeelde ruimte, maar verschillende tijdstippen; synchrone gedistribueerde interactie, waarbij men op dezelfde tijd maar vanaf verschillende lokaties communiceert; en asynchrone gedistribueerde interactie, waarbij men zowel op een verschillende lokatie als op een verschillend tijdstip communiceert (Ellis e.a., 1991).

	Same time	Different times
Same Place	face-to-face interaction	asynchronous interaction
Different Places	synchronous distributed interaction	asynchronous distributed interaction

figuur 3.1: verschillende tijd-plaats situaties

Groupware is breed inzetbaar. Enkele voorbeelden zijn een message system, waarbij mensen asynchroon berichten kunnen uitwisselen; multi-user editors, waarmee meerdere personen gezamenlijk een document kunnen maken en bewerken; group decision support systems, die van pas komen bij het versnellen van het besluitproces; computer conferenties, waarbij personen synchroon vanaf verschillende lokaties samen komen; coördinatie systemen, waarmee een groep collega's het totale werkproces in de gaten kan houden. Deze verschillende functionaliteiten worden steeds vaker geïntegreerd in één systeem.

Bij samenwerken voeren de groepsleden niet alleen tegelijkertijd verschillende acties uit, ze communiceren ook met elkaar, en ze moeten het werk dat zij uitvoeren coördineren. Groupware moet dus als het ware drie vormen van interactie ondersteunen. Bij het ontwerpen van een dergelijk systeem moet nagedacht worden over wat te doen bij situaties waarin deelnemers conflicterende acties uit willen voeren. Ook het centraliseren of decentraliseren van de data, dat van invloed is op reactietijden, of hoe het te bewerken object en de groep te presenteren in de interface zijn issues waarover nagedacht moet worden. Op het gebied van communicatie zijn veel verschillende vormen te onderscheiden, zowel formeel als informeel, die al dan niet door de computer ondersteund gaan worden. Om ten slotte het werk op elkaar af te stemmen, moet het gehele proces gecoördineerd



worden. Dit kan gebeuren door de groepsleden expliciet een agenda te laten invullen, maar kan ook door het systeem op een implicietere wijze gedaan worden, door bijvoorbeeld bestanden op bepaalde momenten voor bepaalde personen af te sluiten. Dit zijn in het kort enkele van de issues die een rol spelen bij het ontwerpen van multi-user applicaties. In paragraaf 3.2 zal hier dieper op ingegaan worden.

### **3.2 Aandachtspunten groupware**

Het ontwerpen en ontwikkelen van multi-user applicaties vraagt om de aanpak van een aantal lastige kwesties die bij single-user applicaties niet of in mindere mate spelen. Ellis e.a. (1991) en Grudin (1994) noemen drie gebieden waar bij het ontwikkelen van groupware problemen ontstaan. Dit zijn sociale factoren, conflictbeheer en groepsinterfaces. Deze worden in de paragrafen 3.2.1, 3.2.2 en 3.2.3 behandeld.

Bij groupware systemen zijn twee factoren van essentieel belang en ieder systeemontwerp dient daar dan ook rekening mee te houden. Deze factoren zijn reactiesnelheid en continuïteit.

*Reactiesnelheid.* Om synchrone processen goed te ondersteunen, zijn twee eigenschappen noodzakelijk: een korte reactietijd, dat is de tijd die nodig is om in een deelnemers interface zijn eigen acties weer te geven, en een korte kennisgevingtijd, de tijd die nodig is voordat deze acties op alle deelnemende computers weergegeven worden.

*Continuïteit.* Alle deelnemers aan het groepsproces moeten continu op de hoogte gebracht worden van de toestand van het systeem en veranderingen daarin, zodat zij op alle tijden over dezelfde informatie beschikken.

#### **3.2.1 Sociale factoren**

Bij het ontwerpen van systemen voor meerdere gebruikers komen alle al bekende interfaceontwerp-problemen kijken. Het grootste probleem hierbij is echter dat verschillende gebruikers met verschillende achtergronden, ervaringen en voorkeuren allemaal gebruik moeten maken van dezelfde groupware applicatie. Deze personen hebben allemaal verschillende functies die op hun eigen manier ingevuld worden, waardoor het lastig is om in het ontwerpproces uit te gaan van een standaard werkprocedure. Een applicatie ontworpen naar een standaard zal mogelijk te star zijn en wellicht niet om kunnen gaan met de verschillende functies, toepassingen en werkmethodes waar in groepsprocessen vaak sprake van is. De ene persoon kan er bijvoorbeeld de voorkeur aan geven met meerdere open werkvensters te werken, terwijl iemand anders liever maar één venster open heeft staan en de rest minimaliseert.

Een andere complicatie voor het ontwerpen van groupware is dat in situaties waarin groupware wordt gebruikt, veel verschillende taken bij elkaar of zelfs door elkaar worden uitgevoerd, die allemaal in één systeem verenigd moeten worden. Zo wordt er bij een vergadering niet alleen gepraat, maar ook gepresenteerd, gebrainstormd en gestemd.

Een ander probleem is dat groupware de subtiele en complexe sociale dynamica eigen aan groepen kan verstoren, en daardoor verzet kan oproepen. De computer kan omgaan met expliciete, concrete informatie. Bij groepsactiviteit zijn echter andere factoren van belang, die minder expliciet of stabiel zijn. Menselijke acties worden, vaak onbewust, geleid door sociale conventies en door kennis van de omringende persoonlijkheden (Grudin, 1994). Het is lastig zulke kennis voor de computer beschikbaar te maken.

Groepsprotocollen zijn overeengekomen manieren om met elkaar om te gaan. Deze protocollen kunnen in de hardware of software zijn ingebouwd, zogenaamde technologische protocollen, of worden overgelaten aan de deelnemers, zogenaamde sociale protocollen.

Een voorbeeld van een technologisch protocol is het 'floor control' mechanisme in conferentiesystemen, waarbij input van maximaal één gebruiker tegelijk kan worden verwerkt, waardoor het proces van om de beurt werken ('turn-taking') opgelegd wordt. Sociale protocollen vind je in de vorm van sociale etiquette waarover overeenstemming bestaat, maar die niet is opgelegd door de groupware. Hierbij gaat het om afspraken die groepsleden onderling maken, bijvoorbeeld om 'track changes' aan te zetten tijdens het bewerken van een tekst.



Zowel de technologische als de sociale benadering van groepsprocessen hebben voor- en nadelen. De processen overlaten aan sociale protocollen moedigt samenwerking aan: de groep moet zijn eigen protocollen vormen. Aan de andere kant kunnen sociale protocollen ook oneerlijk, afleidend en inefficiënt zijn. Daarentegen zorgt het vastleggen van technologische protocollen in software ervoor dat het proces wordt gevolgd, dat er meer structuur aanwezig is, en biedt het assistentie aan minder ervaren gebruikers. Technologische protocollen kunnen echter star zijn, ze dwingen een bepaalde manier van werken af.

### 3.2.2 Conflictbeheer

Groupware systemen hebben behoefte aan conflictbeheer wanneer groepsleden tegelijkertijd conflicterende operaties willen uitvoeren. Het kan bijvoorbeeld voorkomen dat een persoon een zin wil wissen, terwijl een ander persoon er juist een woord aan wil toevoegen. Een normale uitvoering van de operaties kan op zo'n moment niet, maar wat moet het systeem dan wel doen? Er bestaan verschillende manieren om met conflicten om te gaan.

*Data replicatie.* Omdat een real-time groupware systeem behoefte heeft aan korte reactietijden, kan de data naar elke gebruiker gekopieerd worden; veel eventuele 'dure' operaties kunnen lokaal uitgevoerd worden. Een probleem hierbij is echter dat ze hierdoor lastiger te synchroniseren zijn.

*Afscherming.* Hierbij wordt data afgeschermd wanneer het door iemand bewerkt wordt. Het systeem kan afgeschermd bronnen bijvoorbeeld visueel aangeven.

Deze methode levert drie problemen op. Ten eerste veroorzaakt het aanvragen en behouden van een afscherming een vertraging in reactietijd. Ten tweede is het niet altijd duidelijk welk deel afgeschermd moet worden. Bijvoorbeeld bij een tekstverwerker is het niet duidelijk om welk gedeelte het gaat; moet een zin of een paragraaf afgeschermd worden, of enkel het woord of slechts het karakter? Het derde probleem heeft te maken met de timing van de afschermverzoeken en het vrijgeven ervan. Moet de afscherming in een tekstverwerker geplaatst worden wanneer de cursor bewogen wordt, of wanneer een toets is aangeslagen? Het systeem moet de gebruiker niet lastig vallen met dit soort vragen, maar het is moeilijk om automatisch afschermen in tekstverwerkers toe te passen. Het kan problemen opleveren bij de gebruikers wanneer zij opeens geen toegang hebben tot een bestand of gedeelte daarvan zonder dat meteen duidelijk is waarom dat het geval is.

*Beurten.* Bij het om de beurt werken kunnen alle gebruikers aan de beurt komen, waardoor het een goede methode is. Het voornaamste probleem met deze benadering is echter dat het niet geschikt is voor sessies met veel parallelle acties, wanneer meerdere personen tegelijk aan iets willen werken. Een flexibelere vorm van conflictbeheer is het aanvragen van toestemming. Wanneer iemand aan een bestand wil werken, vraagt hij toestemming daarvoor aan. Als niemand anders het bestand gebruikt, krijgt hij toestemming en het bestand, of een deel ervan, wordt afgesloten voor anderen. Wanneer een andere persoon aan het bestand wil werken, vraagt ook hij toestemming aan. De huidige gebruiker wordt hiervan op de hoogte gebracht, en kan de aanvrager laten weten hoe lang hij nog aan het bestand zal werken, waarna zij daarover kunnen onderhandelen. Deze vorm van beheer laat meer controle over aan de personen over hoe zij willen samenwerken, in plaats van een rigide systeemcontrole. Maar ook dit kan op den duur fout gaan. Mensen maken vaak fouten in het volgen van protocollen of willen het simpelweg niet volgen, waardoor meerdere mensen zich kunnen gedragen alsof zij aan de beurt zijn.

*Gecentraliseerde controle.* Hier wordt een gecentraliseerd controle proces ingevoerd. Stel dat alle data gekopieerd is naar de werkstations van alle groepsleden. De controleur ontvangt verzoeken voor operaties, en verzendt deze verzoeken naar alle deelnemers. Aangezien dezelfde operatie uitgevoerd wordt in dezelfde volgorde voor alle deelnemers, blijven alle kopieën van de data hetzelfde. Deze methode werpt echter ook een aantal problemen op. De operaties worden pas uitgevoerd wanneer ze terugkomen van de controleur en niet op het tijdstip van het verzoek, waardoor de reactiesnelheid achteruit gaat. De interface van een deelnemer die een verzoek indient, moet vastgezet worden zolang het verzoek wordt verwerkt; anders bestaat er de mogelijkheid dat een ander verzoek dat refereert naar een specifieke datatoestand uitgevoerd gaat worden terwijl de data alweer in een andere toestand is.

*Afhankelijkheidsdetectie.* Dit speurt naar conflicterende operaties, die dan door de groepsleden onderling opgelost moeten worden. Het grote voordeel van deze methode is dat synchronisatie niet nodig is, niet-conflicterende operaties worden gewoon meteen uitgevoerd, en bovendien dat de reacties van gebruikers erg goed zijn (Ellis e.a., 1991). Mechanismen die de deelnemers betrekken zijn over het algemeen waardevol in groupware applicaties. Zij zijn echter ook kwetsbaar voor menselijke fouten.

*Omgekeerde uitvoering.* Hierbij worden operaties onmiddellijk uitgevoerd, maar er wordt informatie over onthouden zodat de operaties later ongedaan gemaakt kunnen worden, mocht dat noodzakelijk zijn. Veel methodes voor conflictbeheer vallen in deze categorie. Zulke methodes definiëren een globale tijdsordering voor operaties. Wanneer twee of meer conflicterende operaties gelijktijdig zijn uitgevoerd, wordt één (of meer) van deze operaties ongedaan gemaakt, en opnieuw in de juiste volgorde uitgevoerd. De noodzaak om operaties globaal te ordenen is echter een nadeel, evenals de vervelende mogelijkheid dat een operatie bij een deelnemer op een scherm verschijnt, om even later, wanneer het ongedaan gemaakt moet worden, weer te verdwijnen.

*Operatie transformatie.* Deze methode kan beschouwd worden als een afhankelijkheidsdetectie-oplossing met automatische, in plaats van handmatige, conflict oplossing. Ook hier wordt gezocht naar conflicterende operaties, en als daar sprake van is worden operaties getransformeerd. De specifieke transformatie is afhankelijk van operatietype en een log van al uitgevoerde operaties. Het grote nadeel van deze methode is dat het moeilijk te implementeren is.

### **3.2.3 Groepsinterfaces**

Groepsinterfaces verschillen van single-user interfaces omdat ze groepsactiviteit weergeven, en omdat ze bestuurd worden door meerdere gebruikers. Het belangrijkste ontwerpprobleem is hoe om te gaan met complexiteit: meerdere gebruikers kunnen een hoger niveau van activiteit en een grotere graad van complexiteit produceren dan één enkele gebruiker, en de interface moet dit complexe gedrag ondersteunen.

Een goede groepsinterface moet de totale groepsactiviteit weergeven, en tegelijkertijd niet afleidend zijn. Wanneer een persoon bijvoorbeeld stukken tekst uit een bestand gaat verwijderen waarin op dat moment ook andere gebruikers aan het werk zijn, kan dit afleidend zijn. Dit toont een fundamenteel verschil tussen single-user en multi-user interfaces. Bij single-user systemen hebben gebruikers meestal de mentale context om een verandering in het scherm te interpreteren als een gevolg van hun eigen acties. Bij multi-user systemen daarentegen zijn gebruikers over het algemeen niet zo bewust van de acties van hun medegebruikers, en daardoor minder goed in staat plotselinge schermveranderingen die het resultaat zijn van andermans acties, te interpreteren. Wanneer er dan meerdere dingen tegelijk gebeuren, kan dat verwarrend zijn; “wie doet dit?” en “waarom wordt dit veranderd?”

Op zo'n moment is er behoefte aan contextuele aanwijzingen over de groepsactiviteit. Een simpele manier om dit op te lossen is dat deelnemers hoorbaar aankondigen wat hun intenties zijn voordat ze een actie ondernemen. Simpel te implementeren, maar lastig uit te voeren en bovendien ook weer afleidend. Een alternatief is het gebruik van real-time animatie om veranderende groepsactiviteit vloeiend weer te geven. Echter, ook aan deze oplossing kleven problemen. Ten eerste vereist animatie veel rekenkracht van de computer, en bovendien is het lastig herkenbare, visuele metaforen te vinden die geschikt zijn om te animeren.

Een ander probleem in multi-user interfaces is de schermruimte. Schermruimte is een schaars goed in single-user applicaties, maar het is een nog groter probleem in groepsinterfaces waarin elke gebruiker een venster kan creëren die vervolgens ook op de schermen van andere gebruikers verschijnen. Technieken voor het beheren van vensters zijn nodig.

Er bestaan verschillende manieren om een interface geschikt te maken voor gebruik door groepen. Hierboven is het gebruik van animatie al even genoemd, andere methodes zullen later aan bod komen, in paragraaf 5.3.5.

### 3.3 Inzicht

Wanneer een groep mensen een gezamenlijke taak wil uitvoeren, wordt de kwaliteit van de samenwerking beter als de groepsleden inzicht hebben in de andere deelnemers en de gedeelde omgeving waarin de samenwerking plaatsvindt. Besef van andermans werkzaamheden stelt groepsleden in staat hun werk te coördineren en te structureren. Bovendien kunnen mensen bekijken hoe beschikbaar en toegankelijk een ander is wanneer ze met die persoon willen communiceren. Om deze bewustwording te ondersteunen moet het systeem op interface niveau inzicht bieden, in de vorm van extra informatie, zogenaamde inzichtsinformatie. Dit gecreëerde inzicht kan gedefinieerd worden als “een begrip van de activiteiten van anderen waardoor een context voor je eigen activiteit geschapen wordt” (Dourish en Bellotti, 1992). Inzicht is een sleutelfactor voor CSCW systemen; het is essentieel voor het coördineren van het werkproces en het delen van informatie (o.a. Berlage en Sohlenkamp, 1999; Dourish en Bellotti, 1992; Gutwin and Greenberg, 1996; Beaudouin-Lafon en Karsenty, 1992; Ackerman en Starr, 1995; Schmidt, 2002).

Inzichts-informatie is elke soort van meta-informatie over de staat van het systeem: informatie over de wie, waarom, wanneer, waar, wat en hoe-vragen gerelateerd aan veranderingen in de toestand van het systeem. Gutwin en Greenberg (Gutwin and Greenberg, 1996) zetten op een rij welke elementen van belang zijn om inzicht te verkrijgen, en tonen welke vragen een persoon tijdens een gezamenlijke activiteit kan stellen.

Element	Relevante vragen
Aanwezigheid	Wie neemt deel aan de activiteit?
Lokatie	Waar zijn zij aan het werk?
Activiteit niveau	Hoe actief zijn zij?
Acties	Wat zijn zij aan het doen? Wat zijn hun huidige activiteiten en taken?
Intenties	Wat zullen ze vervolgens doen? Waar zullen ze zijn?
Veranderingen	Welke veranderingen zijn ze aan het maken, en waar?
Objecten	Welke objecten gebruiken zij?
Uitgebreidheid	Wat kunnen ze zien? Hoe ver kunnen zij reiken?
Bevoegdheden	Wat kunnen ze doen?
Gebied van invloed	Waar kunnen ze veranderingen aanbrengen?
Verwachtingen	Wat willen ze dat ik hierna doe?

*figuur 3.2: vragen tijdens een gezamenlijke activiteit*

Deze elementen bieden een basis woordenlijst voor het denken over besefvereisten bij groupware ondersteuning. Dit kan door ontwerpers gebruikt worden om bestaande coöperatieve situaties te analyseren en om ondersteuning te bieden bij het ontwerpen van nieuwe systemen.

Een systeem dat inzicht mogelijk wil maken heeft behoefte aan extra functionaliteit: aan de ene kant moet alle informatie die nodig is om aanwijzingen aan groepsleden te verschaffen, verworven worden; aan de andere kant moet de informatie gepresenteerd worden aan andere groepsleden wanneer dit nodig is, zonder informatieoverdaad en afleiding te veroorzaken (Berglage en Sohlenkamp, 1999). Er zijn verschillende benaderingen mogelijk voor de voorziening van de inzichtsinformatie. De verschillende methodes van selectie en presentatie worden besproken in hoofdstuk 5.

### 3.4 Conclusie

Het gebruik van computers in samenwerkingsverbanden zal steeds vaker voorkomen. Het duidelijkste verschil tussen een single-user en een multi-user applicatie, is dat de laatste door meer mensen gebruikt wordt. Meerdere mensen tegelijkertijd aan het werk, betekent meer te doen voor de computer. Daarnaast zijn al die mensen verschillend, met hun eigen achtergronden en ideeën, en zullen zij verschillende manieren van werken hebben. Ook dit moet allemaal ondersteund worden door die ene applicatie. Bovendien spelen er sociale factoren mee in elk groepsproces. Een groep gedraagt zich volgens bepaalde sociale conventies, die vaak impliciet zijn. Een groupware systeem kan deze niet negeren, aangezien ze fundamenteel zijn voor het groepswerk. Er moet dus een manier gevonden worden waarbij de sociale conventies in stand gehouden kunnen worden en waarbij tegelijk de groupware wel zijn ondersteunende taak kan uitvoeren.

Een belangrijke aspect van groupware is conflictbeheer. Het komt voor dat mensen tegelijkertijd conflicterende acties willen uitvoeren, wat vertraging kan opleveren in het systeem waardoor ook de doorgifte van informatie aan gebruikers vertraagd wordt, waarbij de mogelijkheid bestaat dat de verschillende deelnemende computers zich in een andere toestand bevinden, waardoor het hele groepsproces teniet wordt gedaan. Mogelijke oplossingen voor conflictbeheer zijn in dit hoofdstuk besproken, waarbij elke methode zijn voor- en zijn nadelen heeft.

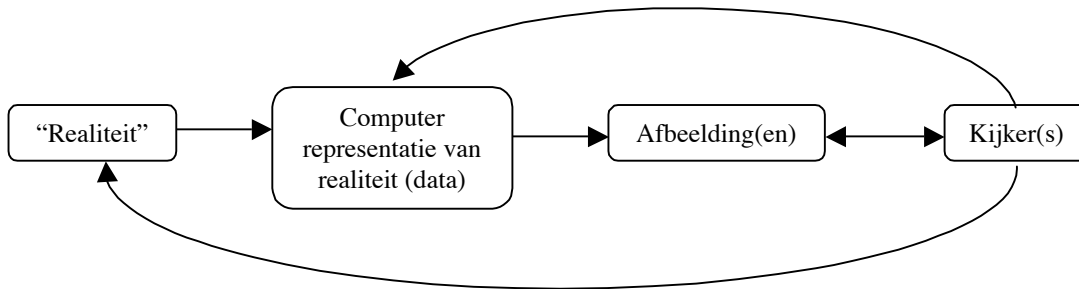
Een ander gebied dat problemen oplevert bij multi-user applicaties is de interface. In een multi-user interface moet niet alleen die informatie weergegeven worden waardoor een gebruiker met het systeem kan werken, maar moet ook het groepswerk worden ondersteund. Bij het weergeven van informatie in het scherm moet rekening gehouden worden met de beperkte ruimte daarin. Bij single-user gebruik van een computer is het vaak al krap, als er dan ook nog extra informatie over het groepsproces moet worden weergegeven, zal dat nog erger worden. Er moet een manier gevonden worden om hier mee om te gaan.

Bij het samenwerken van een groep mensen is het van belang dat er inzicht bestaat in elkaars werkzaamheden. Dit inzicht kan verkregen worden door bepaalde informatie omtrent die personen en het werkproces in de interface aan te bieden. Deze informatie zal leiden tot besef over de omgeving en zijn gebruiker, waardoor de samenwerking beter kan verlopen.

#### 4 Visualisatie

Er bestaan verschillende methodes om informatie te representeren. Een manier om dit te doen is visueel. Visualiseren betekent volgens het Kramers woordenboek “aanschouwelijk maken” of “een beeld in de geest vormen van een zaak waaraan men sterk denkt”. In deze context is de eerste betekenis van belang, het aanschouwelijk maken, het visualiseren van data en informatie, het representeren in een visuele vorm.

In deze scriptie wordt met visualisatie in essentie een proces van het in kaart brengen van computer representaties naar waarneembare representaties bedoeld, daarbij gebruik makend van encodeertechnieken, om menselijk begrip en communicatie te maximaliseren (zie figuur 4.1) (Domik, 1999).



Figuur 4.1: visualisatie

In paragraaf 4.1 volgt een bespreking van diagrammatische representaties en het belang van het kiezen van een geschikte visuele taal daarvoor. Verschillende aspecten die van invloed zijn op de uiteindelijke visualisatie worden in paragraaf 4.2 doorgenomen. Paragraaf 4.3 bespreekt hoe veel informatie overzichtelijk gepresenteerd kan worden in één afbeelding en hoe beweging en tijd tweedimensionaal weergegeven kunnen worden. Vervolgens volgt een opsomming van algemeen voorkomende visuele eigenschappen, zoals kleur en structuur, en hun toepasbaarheid voor informatieve visualisatie in paragraaf 4.4. Tenslotte worden verschillende visualisatie technieken en vervormingstechnieken besproken in paragraaf 4.5 en 4.6.

##### 4.1 Kennisrepresentatie

Om data inzichtelijk te maken voor mensen zodat zij dit kunnen interpreteren, is het noodzakelijk de data op een wijze te representeren die door mensen makkelijker te interpreteren is dan zijn oorspronkelijke vorm. Binnen kennisrepresentatie is een onderscheid te maken tussen twee types van representatie; zogenaamde *analoge* (of directe) representaties en *propositionele* (of Fregeaanse) representaties (Kulpa, 1994). De structuur van een analoge representatie correspondeert met de structuur van het gerepresenteerde. Relatieve posities en afstanden op een landkaart bijvoorbeeld hebben een directe overeenkomst met relatieve posities en afstanden tussen de steden die ze representeren. Een propositionele representatie heeft een structuur die geen overeenstemming hoeft te hebben met het probleem domein. De informatie wordt hierbij in logische formules, vaak predikaatlogica, gepresenteerd. Propositionele representaties kunnen makkelijk universeel gemaakt worden, waardoor ze echter ook minder effectief kunnen worden. Analoge representaties zijn vaak gespecialiseerd voor een specifiek probleem. Anders gezegd, een analoge representatie *modelleert* het gerepresenteerde ding, waar een propositionele representatie het eerder *beschrijft*. Een soortgelijk onderscheid kan gemaakt worden met betrekking tot de wijze van terugvinden van informatie uit de representatie. De benodigde informatie kan vaak simpelweg geobserveerd (of gemeten) worden in een analoge representatie, terwijl het afgeleid moet worden uit de beschrijving van de feiten en axioma's waar de propositionele representatie uit opgebouwd is. Een analoge representatie is dan ook makkelijker interpreteerbaar door mensen, terwijl een computer beter kan werken met een propositionele representatie.

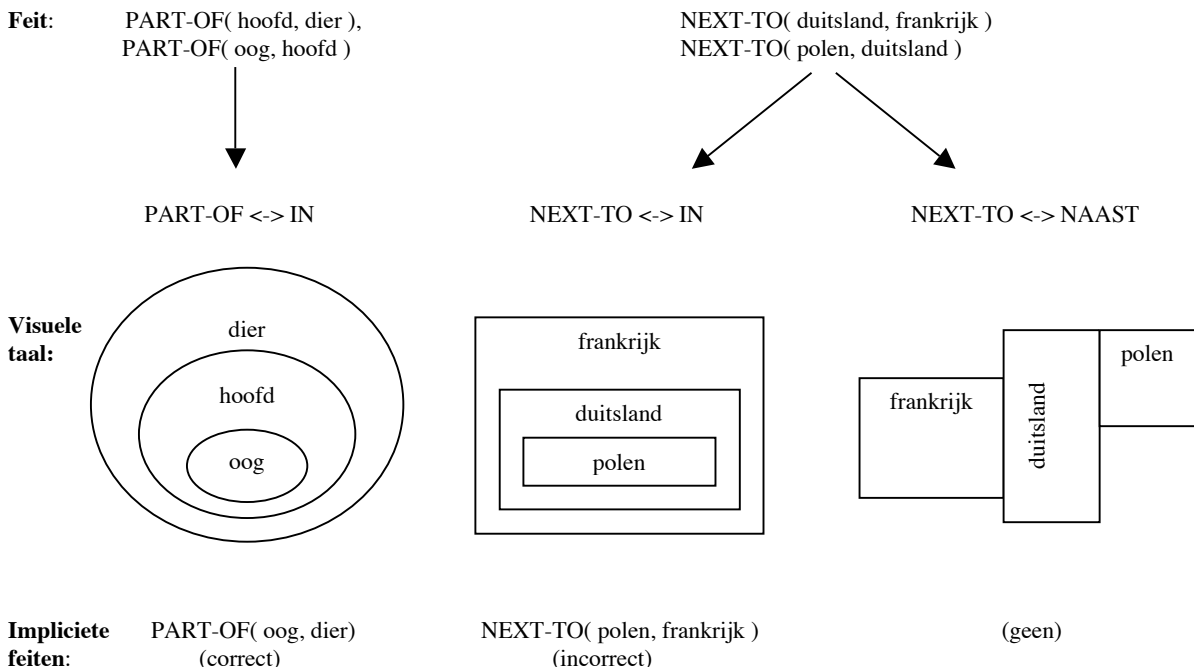
#### 4.1.1 Diagrammatische representatie

Een vorm van analoge representatie is een diagrammatische representatie. Een diagrammatische representatie maakt gebruik van diagrammen om data en kennis te representeren (Kulpa, 1994). Een diagram is een visuele wijze van representeren van informatie. De term ‘diagrammatische representatie’ vertaalt zich dus als ‘representatie gebruik makend van visuele middelen van representatie’. Het is lastig een visuele representatie te bedenken die geen diagrammatische representatie genoemd kan worden. Een voorbeeld is een foto van de werkelijkheid, dat is namelijk een letterlijke weergave, niet kunstmatig gemaakt met de bedoeling informatie te representeren. Diagrammatische representaties maken gebruik van *beeldende* en *grafische* elementen. Beelden worden beschouwd als realistische (representatieve) afbeeldingen van de wereld, in tegenstelling tot meer gesimplificeerde, gestileerde of symbolische grafische afbeeldingen.

#### 4.1.2 Visuele talen

Propositionele representaties kunnen niet woord voor woord vertaald worden naar visuele. Geschikte visuele representatie vereist een eigen, onderscheidende visuele taal, bestaande uit zowel een geschikte visuele woordenlijst van grafische symbolen, als uit “diagrammatische” regels voor compositie van betekenisvolle en effectief leesbare representaties.

Drie belangrijke criteria worden onderscheiden om te bepalen of een visuele taal geschikt is voor de presentatie van bepaalde data, namelijk *expressiviteit*, *effectiviteit* en *presentatiedoel* (Kulpa, 1994). Expressiviteit betekent dat de visuele taal in staat moet zijn om alle feiten uit te drukken en alleen die feiten die gevisualiseerd moeten worden. Het eerste deel spreekt voor zich, het tweede is wat lastiger. Sommige data representatietalen hebben namelijk de eigenschap om wanneer zij expliciet iets representeren, ook impliciete feiten te beweren. Dit ‘impliciete feiten’ fenomeen heeft zowel voor- als nadelen. Aan de ene kant kunnen bepaalde feiten efficiënter geëncodeerd worden en wordt het diagram eenvoudiger te gebruiken in redeneringen – door de impliciete feiten worden bepaalde afleidingen automatisch weergegeven zonder expliciete representatie. Aan de andere kant kunnen de impliciete feiten onwaarheden weergeven.



figuur 4.2: impliciete feiten voortkomend uit gerepresenteerde expliciete feiten in verschillende visuele talen

Een voorbeeld hiervan is te zien in figuur 4.2. In het eerste geval wordt een “part-of” relatie gerepresenteerd door een “in” relatie tussen geometrische figuren. In het diagram is te zien dat het

“oog” figuur aanwezig is in het “dier” figuur, waardoor het impliciete feit dat het oog een onderdeel is van het dier wordt weergegeven. Wanneer de “in” relatie echter gebruikt wordt om een “next-to” relatie weer te geven, wordt impliciet beweerd dat Polen naast Frankrijk ligt. Het is dus belangrijk de juiste visuele taal te kiezen om de feiten in een gegeven domein uit te drukken.

Expressiviteit van een visuele taal houdt dus rekening met de impliciete feiten factor - de visuele representatie moet niet, als bijproduct van het representeren van de gewilde feiten, impliciet ongewilde (incorrecte) feiten representeren.

De effectiviteit van een visuele taal wordt bepaald door hoe eenvoudig het is om de feiten weer te geven in de taal en hoe eenvoudig het is om de feiten uit de representatie af te leiden. Effectiviteit is relatief ten opzichte van de gebruiker van de representatie, andere regels zijn bijvoorbeeld van toepassing wanneer dit een mens is dan wanneer dit een computer is.

Sommige onderzoekers maken ook gebruik van het presentatiedoel criterium dat rekening houdt met wat voor soort operaties de gebruiker uit wil voeren met de representatie, of wat voor soort taken de representatie zal gaan ondersteunen. Voorbeelden van verschillende taken zijn het nauwkeurig opzoeken van waarden, waarbij een representatie van de waarden op een numerieke schaal is aan te raden, versus een algemene vergelijking van groottes, waarbij de waarden door grootte van grafische entiteiten gerepresenteerd kunnen worden.

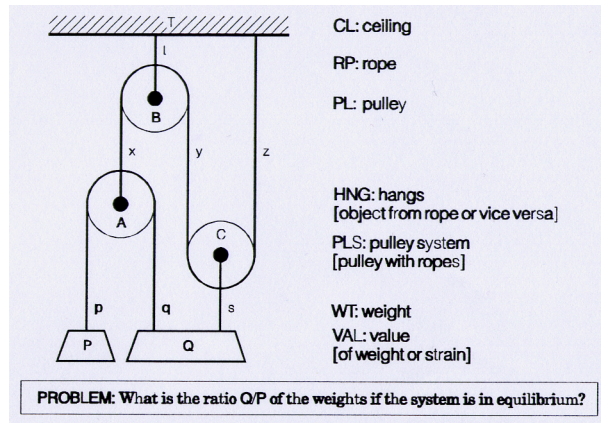
### 4.1.3 Een voorbeeld

Kulpa bespreekt een voorbeeld waarmee het verschil in bruikbaarheid tussen propositionele en diagrammatische representaties wordt geïllustreerd. Figuur 4.3 beschrijft een simpel mechanisch probleem in predikaat logica. Als eerste beschrijft het de feiten, de entiteiten die betrokken zijn, hun eigenschappen en onderlinge relaties. Vervolgens wordt het doel beschreven, namelijk een bewering die waar moet zijn met juist geïnstantieerde variabelen (aangegeven met ?n). De “werking” van het systeem wordt tenslotte aangegeven met een set van regels voor de afleiding van nieuwe feiten uit de bestaande feiten. Deze set van regels en feiten tezamen vormt de kennisbank van het probleem.

<b>FACTS:</b> structural description and parameters
WT(P), WT(Q) RP(p), RP(q), RP(s), RP(x), RP(y), RP(z), RP(t), PL(A), PL(B), PL(C), CL(T) PLS(p, A, q), PLS(x, B, y), PLS(y, C, z), HNG(P, p), HNG(Q, q), HNG(Q, s), HNG(A, x), HNG(B, t), HNG(s, C), HNG(t, T), HNG(z, T) VAL(P, 1)
<b>GOAL:</b>
VAL(Q, ?n)
<b>RULES:</b>
Rule 1: if WT(w <sub>1</sub> ) & RP(r <sub>1</sub> ) & HNG(w <sub>1</sub> , r <sub>1</sub> ) & ~HNG(w <sub>1</sub> , r <sub>2</sub> ) & VAL(w <sub>1</sub> , n <sub>1</sub> ) then VAL(r <sub>1</sub> , n <sub>1</sub> )
Rule 2: if PL(p <sub>1</sub> ) & RP(r <sub>1</sub> ) & RP(r <sub>2</sub> ) & PLS(r <sub>1</sub> , p <sub>1</sub> , r <sub>2</sub> ) & VAL(r <sub>1</sub> , n <sub>1</sub> ) then VAL(r <sub>2</sub> , n <sub>1</sub> )
Rule 3: if PL(p <sub>1</sub> ) & RP(r <sub>1</sub> ) & RP(r <sub>2</sub> ) & RP(r <sub>3</sub> ) & PLS(r <sub>1</sub> , p <sub>1</sub> , r <sub>2</sub> ) & {HNG(r <sub>3</sub> , p <sub>1</sub> ) ∨ HNG(p <sub>1</sub> , r <sub>3</sub> )} & VAL(r <sub>1</sub> , n <sub>1</sub> ) & VAL(r <sub>2</sub> , n <sub>2</sub> ) then VAL(r <sub>3</sub> , n <sub>1</sub> +n <sub>2</sub> )
Rule 4: if WT(w <sub>1</sub> ) & RP(r <sub>1</sub> ) & RP(r <sub>2</sub> ) & HNG(w <sub>1</sub> , r <sub>1</sub> ) & HNG(w <sub>1</sub> , r <sub>2</sub> ) & ~HNG(w <sub>1</sub> , r <sub>3</sub> ) & VAL(r <sub>1</sub> , n <sub>1</sub> ) & VAL(r <sub>2</sub> , n <sub>2</sub> ) then VAL(w <sub>1</sub> , n <sub>1</sub> +n <sub>2</sub> )

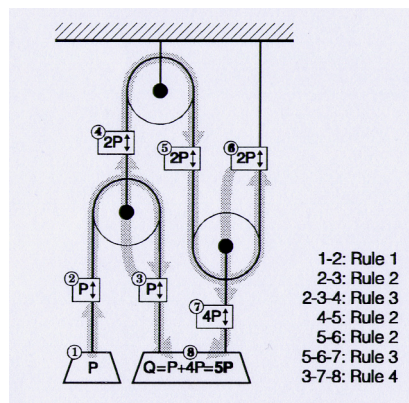
figuur 4.3: een simpel mechanisch probleem in predikaat logica notatie

Zonder verdere hulp is het lastig het mechanische probleem dat hier beschreven wordt op te lossen. Een mogelijke regel moet telkens de hele set van bekende feiten afgaan om te kijken of de regel toegepast kan worden, zo verschillende instantiaties proberend om uiteindelijk hopelijk bij de juiste oplossing uit te komen. Hoeveel makkelijker is het niet om een diagram te gebruiken. Kijk naar figuur 4.4, zo ziet het mechanische probleem er een stuk begrijpelijker uit.



figuur 4.4: een diagram van het “katrollen” probleem. De legenda beschrijft de betekenissen van predikaat symbolen uit de logische representatie van figuur 4.3.

In deze representatie is de oplossing eenvoudig af te leiden door gebruik van diagrammatische redenering. Naamgeving van entiteiten zoals touwen is niet nodig, net zo min als het expliciet benoemen van relaties als ‘hangen’. Hieronder is tenslotte te zien hoe de oplossing makkelijk gevonden kan worden.



figuur 4.5: diagrammatische redenering leidt tot de oplossing van het katrollen probleem; genomen redenatiestappen zijn genummerd, de legenda toont gebruikte regels bij elke stap.

## 4.2 Invloeden op visualisatie

De vorm van de visualisatie wordt door bepaalde eigenschappen van zowel de gebruiker als de data en de toepassing van de visualisatie een zekere richting in gestuurd (Domik, 1999). Hieronder volgt een bespreking van die eigenschappen, met eventuele vermelding van visualisatie mogelijkheden.

*Interpretatie doelen* worden gedefinieerd door de gebruiker, bijvoorbeeld voor wetenschappelijke visualisatie (vergelijken van waarden, identificeren, onderscheiden of categoriseren van objecten), software visualisatie (nadruk op tekst, data structuren, prestatie, algoritme), of informatie visualisatie (nadruk op detail met overzicht, tonen van relaties).

*Vermogens en wensen van gebruiker* kunnen beperkingen opleveren door onvoldoende vermogen tot kleurwaarneming, kleur geheugen, mentale rotatie, fijne motoriek, of door intuïtieve kleur rangschikking of voorkeuren.

*Beschikbare software en hardware* kunnen beperkingen opleveren door output apparatuur (bijvoorbeeld monitor versus beamer), input apparatuur (bijvoorbeeld mechanische of optische muis, toetsenbord), tekort aan computer/grafisch vermogen (bijvoorbeeld bij real-time performances), beschikbaarheid van benodigde software.

*Data karakteristieken* zijn verder onder te verdelen:



- aantal dimensies (bijvoorbeeld 1,2,3)
- relaties tussen data variabelen (onafhankelijk, hiërarchisch, netwerk (Wenzel e.a., 2001), visualiseren door de rangschikking van de elementen ten opzichte van elkaar, bijvoorbeeld naast of boven elkaar)
- continuë data (representatie door een functie  $y = f(x)$ ) of niet-continuë data (bijvoorbeeld moleculen, databases en binaire bomen)
- data type zoals rangtelwoord ([klein, medium, groot], effectieve visuele attributen zijn helderheid, grootte en in mindere mate kleur/tint), nominaal (leden van een bepaalde klasse zoals [eik, beuk, kastanje], effectieve visuele attributen zijn kleur/tint en symbool) of kwantitatief (precieze numerieke waarden, bijvoorbeeld [2.3, 4.56, 0.8, 2.5E-35], effectieve visuele attributen zijn positie, lengte en in mindere mate kleur/tint)
- data type als punt (elk data-element wordt beschouwd als een positie in een n-dimensionale ruimte. Voorbeeld: meting van bladeren: [lengte, breedte, boomsoort, leeftijd], bijvoorbeeld [2.3, 1.2, B, 1], [4.3, 2.2, B, 3], [1.5, 1.5, M, 1], [3.0, 2.9, M, 3]. Expressieve visualisaties zijn stipdiagrammen en iconen), schaal (elk data-element heeft een numerieke expressie. Voorbeeld: topografie van terrein met hoogteverschillen, temperatuur, golfhoogte. Expressieve visualisaties zijn lijngrafieken en beschaduwde oppervlak) en vector (elk data-element wordt beschouwd als een rechte, gerichte lijn met een bepaalde lengte (grootte) in n-dimensionale ruimte. Voorbeeld: richting van deeltjesstroom in kanaal, snelheid, kracht. Expressieve visualisaties: pijlen, stroomlijnen, deeltjesspoor).
- data betrouwbaarheid is niet altijd optimaal, er kan zelfs data missen. Er kan in dat geval interpolatie toe worden gepast, maar dat moet met uiterste zorgvuldigheid gebeuren. Mogelijke expressieve visualisaties zijn het aangeven van foutbalken, of het aangeven van onderscheid tussen 'echte' en missende data.

### 4.3 Veel informatie in één afbeelding

Het is lastig veel informatie op een overzichtelijke wijze weer te geven in één afbeelding die beperkt wordt door slechts twee dimensies. Een taak van informatievisualisatie is dan ook het ontsnappen aan deze beperking (Tuft, 1990). De manieren om dit te doen zijn het verhogen van (1) het aantal dimensies dat op vlakke oppervlakken gerepresenteerd kan worden en (2) de data dichtheid (hoeveelheid informatie op bepaald gebied). Methodes hiervoor zijn micro/macro ontwerpen, lagen aanbrengen in data, vermenigvuldigen van afbeeldingen en gebruik van kleur.

#### 4.3.1 Micro / macro ontwerpen

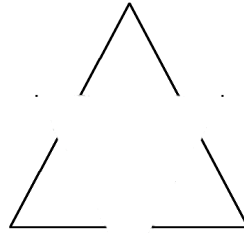
Tuft (1990) onthult een hoogst onconventionele ontwerpstrategie: om te verhelder, voeg detail toe. Een micro/macro ontwerp toont zowel micro als macro informatie en kan veel detail weergeven, daarmee organisatie aanbrengend in complexiteit door meerdere en vaak hiërarchische lagen. Wanneer de visuele taak is contrasteren, vergelijken en kiezen, wat vaak het geval is, dan geldt: hoe meer relevante informatie binnen het blikveld, hoe beter. Open, niet dichte vertoningen, verspreid over meerdere pagina's, vereisen dat de kijker vertrouwt op visueel geheugen om een contrast, een vergelijking, een keuze te maken. Het visueel geheugen is echter een zwakke eigenschap, waardoor dit vaak niet goed zal gaan.

#### 4.3.2 Aanbrengen van lagen

Om ruis te reduceren en de inhoud van vertoningen te verrijken is het visueel in lagen brengen van verschillende aspecten van de data een krachtige methode.

Effectief lagen aanbrengen in informatie is vaak moeilijk: de verschillende, verzamelde elementen interacteren, daarbij bijkomstige patronen en structuren creërend. Dit effect wordt aangeduid als  $I + I = 3$  of meer; twee elementen worden samen getoond waardoor bijkomstige producten ontstaan – soms een basis voor plezierige esthetische effecten, maar een continu gevaar voor datavertoningen vanwege de kans op allerlei ongeplande en rommelige, interacterende combinaties. Kleine, subtiele zetten kunnen resulteren in doorslaggevende resultaten, zoals te zien is in de demonstratie van

illusionaire randen van subjectieve contouren in figuur 4.6. Visuele activering van negatieve gebieden van witte ruimte illustreert de contextuele en interactieve natuur van visuele elementen.



figuur 4.6: visuele illusie

Wat van belang is, is de juiste relatie tussen informatielagen. De lagen moeten met elkaar in harmonie zijn, zodat de informatie goed wordt uitgedrukt. Een voorbeeld waarbij dat niet het geval is, is een landkaart waarop alle informatie (contouren, rivieren, wegen, namen) op dezelfde wijze worden weergegeven, bijvoorbeeld door zwarte lijnen. Dit resulteert in een niet-onderscheidend, wazig en chaotisch vlak.

#### 4.3.3 Vermenigvuldigen van afbeeldingen

Bij kwantitatieve redenering wordt vaak de vraag gesteld: vergeleken met wat? Bij het veelvuldig tonen van afbeeldingen wordt visuele vergelijking direct afgedwongen. Voor veel problemen in data presentatie, zijn kleine veelvuldig getoonde afbeeldingen de beste ontwerp oplossing. Dit kan gebeuren door illustraties in postzegelformaat te tonen, opeenvolgend in de tijd zoals frames in een film, of geordend door een kwantitatieve variabele die niet in de afbeelding zelf gebruikt wordt. Aangezien veel stukken informatie getoond worden binnen het blikveld, kunnen kijkers in staat zijn contrasten en overeenkomsten in één oogopslag op te merken. Deze wijze van vertoning onthult, allemaal in eens, een mogelijkheid aan alternatieven, een meervoud aan opties.

#### 4.3.4 Gebruik van kleur

Hoe kunnen we in het representeren en communiceren van informatie voordeel halen uit kleur? Menselijke ogen zijn uiterst gevoelig voor kleurvariaties: ongeveer 20,000 kleuren zijn toegankelijk voor veel kijkers, waarbij de beperkingen voor praktische toepassingen bepaald worden door de beperkingen van het menselijk visueel geheugen in plaats van de capaciteit om lokaal tussen twee aangrenzende tinten te onderscheiden. Voor het encoderen van abstracte informatie produceren meer dan 20 of 30 kleuren echter vaak negatieve resultaten. Zelf een juiste kleur kiezen om op een juiste plek te gebruiken, is een complexe zaak.

De fundamentele toepassingen van kleur in informatievormgeving zijn: om te labelen, om te meten, om te representeren of realiteit te imiteren, om te verlevendigen of decoreren. Tufte (1990) illustreert dit aan de hand van een Zwitserse bergkaart: hier labelt kleur door water van steen te onderscheiden en gletsjer van veld, meet door aangeven van hoogte met contourlijnen en snelheid van verandering door verdonkeren, imiteert realiteit met rivierblauw en schaduw, en verlevendigt de topografie op een visuele wijze die niet bereikt zou worden met zwart en wit alleen.

Tufte (1990) bespreekt strategieën voor het gebruik van kleur bij informatiepresentatie uit de cartografie die beschreven worden in Eduard Imhof's *cartographic Relief Presentation*.

- *Eerste regel*: pure, heldere en sterke kleuren hebben felle, ondragelijke effecten wanneer ze onafgebroken over grote gebieden naast elkaar staan, maar mooie effecten kunnen bereikt worden wanneer ze spaarzaam gebruikt worden op of tussen rustige achtergrondtinten. Wanneer echter alles, vooral grote gebieden, verblindende, rijke kleuren krijgt, hebben de afbeeldingen ongeordende, verwarrende en onplezierige effecten.
- *Tweede regel*: het plaatsen van lichte, heldere kleuren gemixt met wit naast elkaar levert onplezierige resultaten, vooral wanneer de kleuren voor grote gebieden worden gebruikt.

- *Derde regel:* grote stukken achtergrond of basiskleuren moeten hun werk stil doen, de kleinere, heldere gebieden in staat stellend om levendig af te steken, wanneer de eerste dof, grijsachtig of neutraal zijn. Sterk gedempte kleuren, gemixt met grijs, zorgen voor de beste achtergrond voor het gekleurde thema.
- *Vierde regel:* als een afbeelding is opgebouwd uit twee of meer grote, omringde stukken in verschillende kleuren, valt de afbeelding uit elkaar. Eenheid wordt echter behouden wanneer de kleuren van een gebied herhaaldelijk vermengd worden in het andere, wanneer de kleuren op een tapijtmanier door de andere heen geweven worden. Alle kleuren van het hoofdthema moeten als eilanden gestrooid worden in de achtergrondkleur.

Welk palet van kleuren moeten we kiezen om informatie te representeren en te verlichten? Een goede strategie is om kleuren uit de natuur te gebruiken, vooral die aan de lichtere kant, zoals blauwen, gelen en grijzen van lucht en schaduw. Natuurlijke kleuren zijn bekend en samenhangend, harmonieus en hebben een zekere definitieve autoriteit. Lokale nadruk van data kan dan gegeven worden door plekken van hoogtepunten van felle kleuren geweven door de serene achtergrond.

Aan de hand van zeekaarten illustreert Tufte (1990) het verschil tussen twee soorten kleurgebruik. In de ene kaart wordt blauw gebruikt om diepte aan te geven en bruin voor hoogte. Hier wordt het verloop aangegeven door een *waarde* schaal, lopend van licht naar donkerblauw. Een alternatief is een schaal van regenboogkleuren, waarmee de duidelijke visuele opvolging van licht naar donker vervangen wordt door het ongeordende rood, oranje, geel, groen, blauw, indigo en violet. In deze opeenvolging is moeilijk orde aan te brengen, bij het zien van deze kleuren moeten kijkers zich beroepen op andere aanwijzingen (contouren, randen, labels) om de data te kunnen zien en interpreteren. Uiteindelijk is de mens dus beter in staat om opvolging door gebruik van waardeschalen waar te nemen, dan door gebruik van (in menselijke ogen) ongeordende, felle kleuren.

#### 4.3.5 Weergeven van beweging en tijd

Informatie over beweging, proces, mechanisme, oorzaak en gevolg is lastig te presenteren in een statische, tweedimensionale ruimte. Tufte (1997) beschrijft een aantal methodes om beweging en tijd op een juiste manier te presenteren in een tweedimensionaal vlak.

Beweging kan weergegeven worden door gebruik van:

- Meerdere posities in één plaatje: door bijvoorbeeld een hand één keer met doorgetrokken lijn en één keer met gestippelde lijn te tekenen wordt duidelijk dat het twee opvolgende posities zijn.
- Meerdere plaatjes naast/onder elkaar: verschillende plaatjes tonen delen van activiteit; gestippelde lijnen geven beweging binnen plaatje aan; een opeenvolging met verschillende posities van hetzelfde object geeft beweging aan.
- Pijlen van beweging in één plaatje of tussen plaatjes.
- Objecten van links naar rechts tekenen in één afbeelding.

In paragraaf 4.3.3 is het vermenigvuldigen van afbeeldingen, waardoor een kijker in één oogopslag tussen de verschillende afbeeldingen kan vergelijken, ook al besproken. Dit kan dus niet enkel gebruikt worden om veel informatie op één moment weer te geven, maar ook om veranderingen door de tijd weer te geven. Hierdoor ontstaat parallellisme. Visueel parallellisme is makkelijk te bereiken door twee plaatjes naast elkaar te zetten, eventueel met een pijl of iets dergelijks ertussen. Meerdere plaatjes tonen herhaling en verandering, patroon en afwisseling. Ruimtelijk parallellisme maakt gebruik van menselijke vermogen om te vergelijken en redeneren over meerdere plaatjes die simultaan in ons blikveld verschijnen.

Parallele plaatjes kunnen ook tijdelijk verspreid worden, met één plaatje volgend op een ander, parallel in tijd. Bij tijdgebonden sequenties worden vergelijkingen gemaakt tussen een herinnerd plaatje met een momenteel zichtbaar plaatje, een soms lastige taak.

Parallellisme verbindt visuele elementen. Verbindingen worden gemaakt tussen plaatjes door positie, oriëntatie, overlap, synchronisatie en overeenkomsten in inhoud.

#### 4.4 Visuele eigenschappen

Elke afbeelding heeft een aantal visuele eigenschappen, zoals kleur en structuur. In een visualisatieproces is het kiezen van de juiste visuele attributen essentieel. Volgens Salomon (1994) bestaan er aangeboren en aangeleerde reacties op visuele eigenschappen. Bij de eerder besproken zeekaart is sprake van een aangeboren reactie. Het gebruik van waardeschalen, waarbij donkerblauw een dieper gedeelte aanduidt dan lichtblauw, is erg effectief, omdat de kijker automatisch die rangorde aan de kleuren geeft en ze dus juist interpreteert. Daarnaast bestaan er aangeleerde reacties op bepaalde visuele attributen. Dit is bijvoorbeeld het geval bij regenboogschalen en verkeersborden. Een ander voorbeeld is de relatie tussen gebruik van bepaalde kleuren en bepaalde gebeurtenissen, zoals gebruik van de kleuren rood en groen voor kerst, of gebruik van oranje en bruin voor 'jaren 70'.

Hieronder volgt een bespreking van de belangrijkste visuele eigenschappen volgens Domik (1999) die voor representatie van bepaalde datatypes zeer geschikt zijn.

*Kleur.* Dimensies van kleur: tint, verzadiging, helderheid. Kunnen gezamenlijk of onafhankelijk gevarieerd worden.

*Tint.* Kleuren van de regenboog (gerelateerd aan golflengte). Kan effectief gebruikt worden voor nominale data types en rangtelwoorden (kleurschaal).

Hints:

- Blauw (koele kleuren): toepasselijk voor verder weg gelegen, koelere, lagere of negatieve waarden.
- Rood (warme kleuren): toepasselijk voor nabije, warmere, hogere en positieve waarden; gevaar.
- Vorm van object getoond in regenboogsschaal zal niet makkelijk waarneembaar zijn.
- Tinten zien er anders uit op verschillende achtergronden.

*Verzadiging.* "Bleekheid" van kleur is gebrek aan verzadiging (gerelateerd aan spectrale distributie). Effectief te gebruiken voor rangtelwoorden. Voorzichtig bij onafhankelijke interpretatie van verzadiging en helderheid. Tweedimensionale kleurschalen voor effectiviteit.

*Helderheid.* Lichte/donkere kleuren (gerelateerd aan hoeveelheid licht die het oog binnen komt). Effectief gebruikt voor rangtelwoorden (en kwantitatieve data types).

Hints:

- Heldere objecten op donkere achtergrond lijken groter dan donkere objecten op heldere achtergrond.
- Vervagende helderheid geeft impressie van afstand/diepte.
- Contrast beïnvloedt waarneming van helderheid.

*Structuur.* Effectief te gebruiken voor nominale datatypes. Voeg legenda toe.

Hints:

- Voorzichtig met overlappende structuren
- Structuren kunnen ook andere impressie geven, zoals dichtheid.

*Oriëntatie.* Gebruik verschillende oriëntaties om juiste zicht op objecten te verzekeren.

Hints:

- Bekendheid van vorm vaak verbonden met oriëntatie
- Bij voorkeur symmetrie rond verticale as.

*Dimensie.* De presentatie kan één, twee, twee en half of drie dimensies hebben. Twee en halve dimensies worden gebruikt bij een 3D-presentatie die uit een 2D-presentatie is ontwikkeld (Wenzel e.a., 2001).

*Diepte attributen.* Gebruik diepte attributen om waarneembaarheid van 3D-objecten te vergroten.

Hints:

- Vervaag helderheid om toenemende diepte weer te geven.
- Gebruik perspectief om toenemende diepte weer te geven.
- Occlusie om voor/achter te onderscheiden.
- Transparantie om voor/achter te onderscheiden.
- Verandering in helderheid om oppervlakken te simuleren.

- Rotatie om 3D waarneming te verbeteren

*Beweging.* Voorbeelden: animatie; verwissel twee of meer afbeeldingen om verschillen of overeenkomsten aan te geven.

#### 4.5 Visualisatie technieken

Hoe de data uiteindelijk te visualiseren hangt af van de toepassing van de visualisatie en het type van de data. Hier volgt een opsomming van verschillende visualisatietechnieken (Domik, 1999; Harris, 1999).

*Diagrammen:* zijn voornamelijk gemaakt van geometrische vormen als cirkels, rechthoeken en driehoeken die verbonden zijn door pijlen of lijnen. Hiermee kan getoond worden hoe dingen met elkaar in verband staan. Een voorbeeld is een flow diagram (veel gebruikt bij software visualisatie), of een organisatie diagram.

*Histogrammen:* tonen de frequentie waarmee specifieke waarden of waarden binnen een interval voorkomen in een set van data. Aantal klassenintervallen kan variëren. In een eendimensionaal histogram geeft de lengte van de balk het aantal elementen in een subcategorie aan. In een tweedimensionaal histogram geeft helderheid of kleur het aantal elementen in een subcategorie aan.

*Punt/stipgrafiek:* positie van punten geeft lokatie van data-elementen aan. Animatie kan gebruikt worden voor 3D-effect.

*Grafieken:* zijn grafische weergaves van kwantitatieve relaties tussen twee of meer groepen van informatie, bijvoorbeeld steden en bevolking, autosnelheid en efficiëntie, koopkracht van de dollar over de tijd. Grafieken hebben een, twee of drie rechte assen. Grote hoeveelheden data kunnen in één oogopslag bekeken worden; afwijkingen in data worden snel zichtbaar.

*Kaarten:* bieden de mogelijkheid om informatie visueel te tonen in relatie tot zijn fysieke (ruimtelijke) lokatie, waardoor ze in het bijzonder geschikt zijn om geografische verdeling te tonen.

*Iconen:* worden gebruikt als symbolen bij grafieken en kaarten. Groepen van iconen worden ook gebruikt om meerdere entiteiten te vergelijken op drie of meer variabelen. Hierbij is een legenda nodig. Pijlen kunnen ook iconen zijn, die informatie weergeven door richting, lengte, breedte en kleur.

*Contourlijnen (isolijnen):* verbinden punten met dezelfde waarde in 2D-raster.

*Taartdiagrammen:* tonen relatieve groottes van onderdelen ten opzicht van elkaar en het geheel.

*Proportionele diagrammen:* worden gebruikt om verschillen in grootte, waarde of aantal grafisch te tonen zonder gebruik van schalen. Dit wordt gedaan door delen grafisch in dezelfde verhouding tot elkaar te tonen als de verhouding tussen de waardes die ze representeren. Als de grootte van A twee keer de grootte van B is, dan is de waarde van A ook twee keer zo groot als de waarde van B.

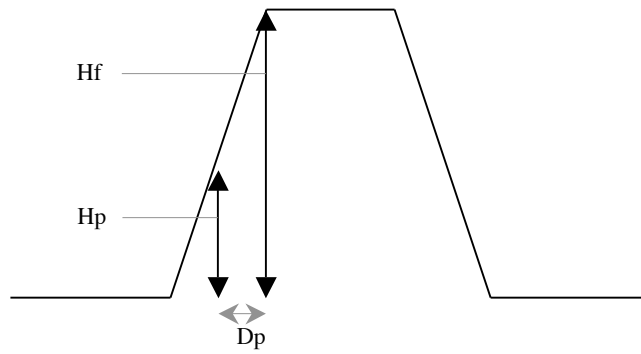
#### 4.6 Vervormingstechnieken

Wanneer informatie op een computer bekeken wordt, gebeurt het maar al te vaak dat de grootte van het scherm de beperkende factor is. Er bestaan verschillende manieren om de ruimte van een monitor optimaal te benutten door verandering van presentatiewijze.

*Fisheye presentatie:* wordt mogelijk gemaakt door het gebruik van *drop-off functies* (monotoon afnemend). Geselecteerde regio's kunnen met een specifieke graad van vergroting getoond worden en integratie tussen de 'focusregio's' en hun omringende context wordt bereikt door het gebruik van drop-off functies (Carpendale e.a., 2001) (zie figuur 4.7).

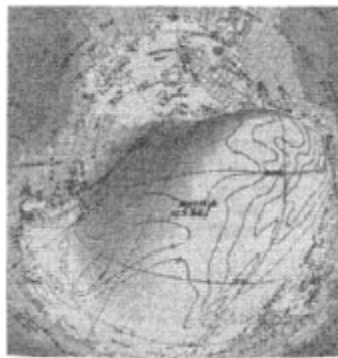
Elk punt op de basislijn dat in de focusregio valt wordt opgehoogd naar hoogte  $H_f$ . De hoogte  $H_p$  van elke overig punt wordt vervolgens bepaald door de waarde van de drop-off functie toegepast op de afstand tussen het punt en de focusregio  $D_p$ .

Het kan ook zo zijn dat juist de focusregio onveranderd blijft en op normale grootte wordt afgebeeld, maar dat de contextregio wordt verkleind, dus dat alle punten buiten de focusregio worden verlaagd.



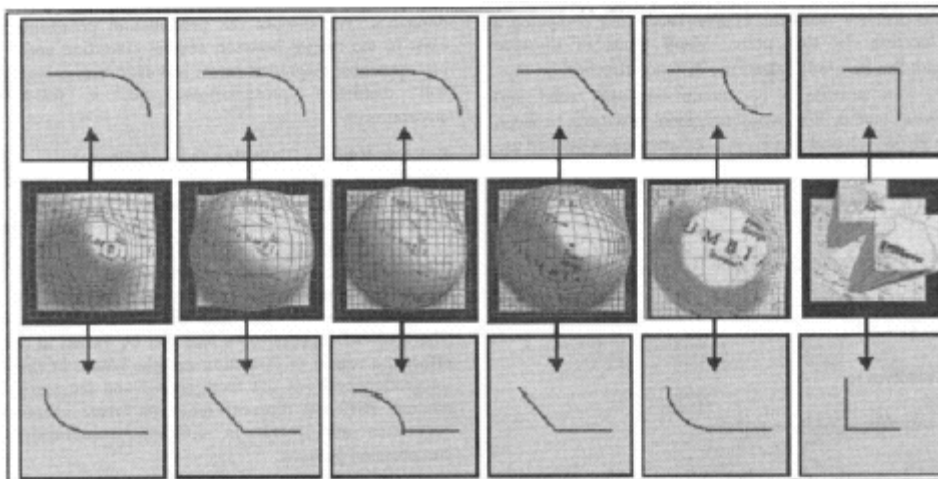
figuur 4.7: gebruik van een lineaire drop-off functie

Wanneer de focusregio verplaatst wordt zodat het over de contextregio heen komt te staan, spreekt men van *folding* (zie figuur 4.8). Hierdoor is een deel van de contextregio niet meer zichtbaar, maar kan de focusregio wel verplaatst worden zonder geïsoleerd te raken van de rest van de afbeelding.



figuur 4.8: *folding*

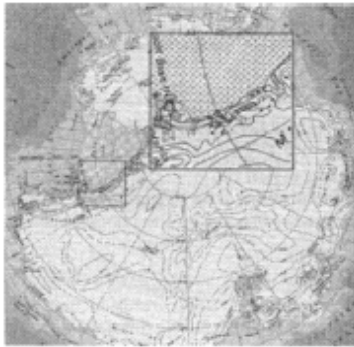
Bij het creëren van een fisheye presentatie kunnen de overgangen, van focusregio naar de vervorming (door drop-off functie) en van vervorming naar de contextregio, op verschillende wijze verlopen. Ook de drop-off functie kan variëren. Er kan voor alle gevallen bijvoorbeeld een sinus functie of een lineaire functie gebruikt worden. Verschillende mogelijkheden zijn te zien in figuur 4.9.



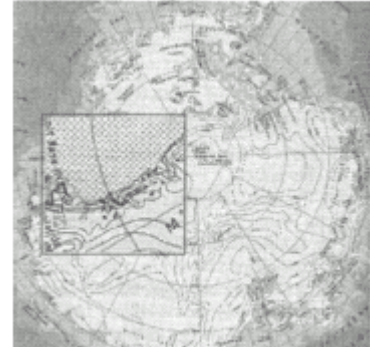
figuur 4.9: verschillende functies: van links naar rechts Gaussiaans, cosinus, bol, lineair, inverse cosinus en Manhattan. De bovenste rij toont de overgang van focusregio naar vervorming, de onderste rij van vervorming naar contextregio.

*Inset / offset.* Een inset is een geselecteerde subregio van een afbeelding die op zijn plek wordt vergroot (zie figuur 4.10). Hierdoor wordt de omringende context niet zichtbaar. Een offset is een

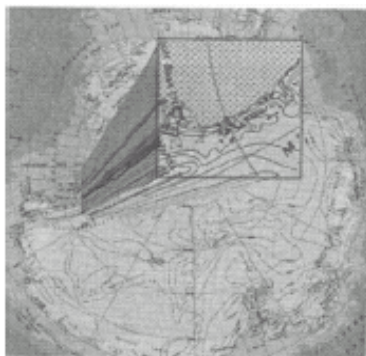
inset die verplaatst is in de afbeelding (zie figuur 4.11). Hierbij kan de maker ervan zelf de plek kiezen om de vergroting te tonen, wanneer bijvoorbeeld de omringende context zichtbaar moet blijven. Soms wordt de offset door lijnen verbonden met zijn oorspronkelijke lokatie om visuele duidelijkheid te scheppen (zie figuur 4.12). Een zogenaamde Manhattan lens verbindt de offset met zijn oorspronkelijke lokatie door de randen op te rekken (zie figuur 4.13). Hierin is weer de drop-off functie te herkennen.



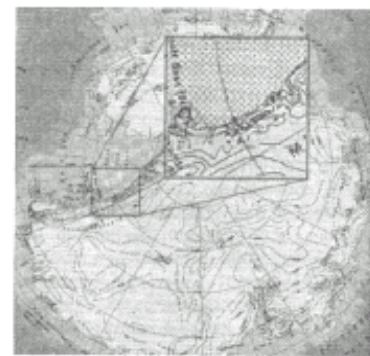
*figuur 4.10: inset*



*figuur 4.11: offset*



*figuur 4.12: offset met verbindinglijnen*



*figuur 4.13: Manhattan*

#### **4.7 Conclusie**

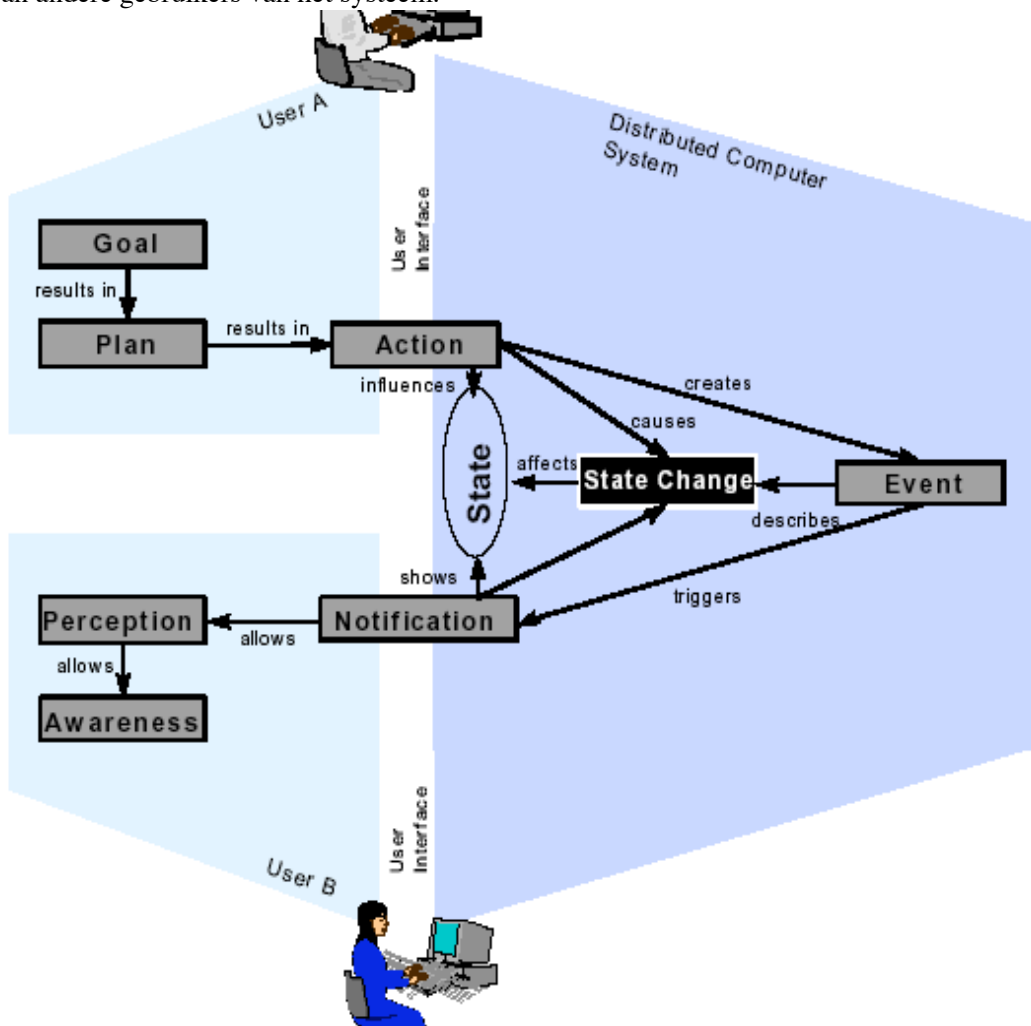
Een effectieve wijze om data te tonen, is het visueel te representeren. Een diagrammatische representatie is voor mensen eenvoudiger te interpreteren dan een propositionele. Bij informatievisualisatie dient rekening gehouden te worden met verschillende eigenschappen van het visualisatieproces, zoals data-eigenschappen en het doel van de visualisatie. Hierdoor wordt de uiteindelijke visualisatie een bepaalde richting in gestuurd. Om veel informatie weer te geven in een beperkte, tweedimensionale ruimte kunnen verschillende vervormingen toegepast worden op de informatie. Zo kan de informatie in lagen weergegeven worden, of kan het gebruik van kleur extra diepte geven aan een afbeelding. Het weergeven van tijd of beweging in een statische, tweedimensionale ruimte vraagt ook om specifieke methodes. In de uiteindelijke visualisatie kunnen verschillende visuele eigenschappen en technieken ingezet worden om de informatie zo expressief en effectief mogelijk te tonen.

## 5 Visualisatie in multi-user systemen

### 5.1 Het kennisgevingsproces

Een multi-user systeem moet informatie presenteren aan zijn gebruikers om hen inzicht te bieden in de groepsdynamiek. Dit inzicht in computerondersteunde samenwerking wordt in meerdere stappen bereikt, zoals te zien in figuur 5.1.

Beschikbare informatie over een gebruikers activiteit moet gecommuniceerd worden naar en verwerkt worden door de andere gebruikers. Dit proces wordt geïnitieerd door een gebruiker met een doel ('goal'), wat resulteert in een plan ('plan'), wat resulteert in een gebruikersactie ('action'), wat een toestandsverandering ('state change') van het systeem veroorzaakt. De toestandsverandering wordt als gebeurtenis ('event') naar alle gebruikers verspreid. De laatste stap is de kennisgeving ('notification'), door Berlage en Sohlenkamp (Berglage en Sohlenkamp, 1999) gedefinieerd als "het waarneembaar maken van gemeenschappelijke activiteiten". De waarneming ('perception') van de kennisgeving maakt gebruikers bewust ('allows awareness') van de gedane actie. Dit proces vormt een feedbackcyclus: besef van andermans acties beïnvloedt toekomstige acties van andere gebruikers van het systeem.



figuur 5.1: creëren van inzicht in computerondersteunde samenwerking

Deze beschrijving identificeert twee cruciale punten in het proces van kennisgeving. Ten eerste moet de noodzakelijke informatie aanwezig zijn in een vorm die de computer kan verspreiden naar alle deelnemers. Ten tweede moet de kennisgeving op het juiste moment en op de juiste manier plaatsvinden, afhankelijk van de huidige situatie van elke deelnemer. Deze twee stappen worden



door zowel Berlage en Sohlenkamp (1999) als Gutwin en Greenberg (1996) in een kader geplaatst om het totale proces van kennisgeving vorm te geven. De twee hierboven genoemde, cruciale onderdelen van dit kader, selectie en presentatie, zullen in de paragrafen 5.2 en 5.3 besproken worden. Deze twee onderdelen worden in dit kader verder onderverdeeld in aspecten als relevantie, informatieoorsprong, presentatie medium, tijdelijke en ruimtelijke vervorming, en integratie in bestaande interfaces. Tabel 5.2 vat deze aspecten samen en toont hun relatie met de verschillende stappen van het kennisgevingsproces.

Aspect	Beschrijving
Oorsprong	Hoe wordt inzichtsinformatie gecreëerd en geselecteerd?
Relevantie	Hoe kan de relevantie van een gebeurtenis voor een gebruiker ingeschat worden?
Medium	Welke informatie moet waarneembaar worden gemaakt? Welk informatiemedium is het meest geschikt voor een kennisgeving?
Tijdelijke vervorming	Welke tijdelijke vervormingstechnieken moeten toegepast worden?
Ruimtelijke vervorming	Welke ruimtelijke vervormingstechnieken moeten toegepast worden?
Integratie	Hoe worden de kennisgevingstechnieken geïntegreerd in de interface?

*tabel 5.2: aspecten van het kennisgevingsproces*

## 5.2 Selectie

De eerste stappen in het kennisgevingsproces selecteren de informatie voor een specifieke gebruiker: welke informatie uit de beschikbare pool van gebeurtenissen is geschikt voor presentatie?

### 5.2.1 Informatie oorsprong

Inzichtsinformatie moet gecreëerd en verzameld worden. Een belangrijke issue omtrent het kennisgevingsproces is de hoeveelheid werk die gebruikers moeten investeren om de informatie te genereren. Actieve en passieve vormen zijn mogelijk. In het geval van actieve creatie van inzichtsinformatie (waarbij 'actief' verwijst naar de gebruiker en niet naar het systeem), moeten gebruikers zelf informatie verschaffen, bijvoorbeeld door het bijhouden van aantekeningen of het versturen van e-mail. Passieve creatie van inzichtsinformatie daarentegen wordt door het systeem gedaan als bijeffect van gebruikersacties. Dit is lastiger te implementeren. De passieve benadering biedt echter veel voordelen. Gebruikers worden niet gedwongen zelf informatie te verschaffen, ze hoeven geen extra werk te doen. Daarbij komt dat bij actieve informatievergaring de verzender van de informatie bepaald welke informatie verzonden wordt en op wel moment. Het kan zijn dat informatie die op deze manier verkregen is niet specifiek genoeg is voor de ontvangers, of op dat moment niet relevant is (Dourish en Bellotti, 1992) waardoor problemen kunnen ontstaan. Desondanks kan actieve creatie een aanvulling zijn om informatie te verschaffen die anders niet voorhanden zou zijn; bijvoorbeeld om de bedoelingen van een gebruiker duidelijk te maken. In dat geval is het belangrijk dat de gebruikersinterface ondersteuning biedt aan een eenvoudige, niet storende invoer van de informatie zodat deze functie wordt geaccepteerd door de gebruikers.

### 5.2.2 Relevantie

Het selecteren van gebeurtenissen voor kennisgeving hangt af van één cruciale kwestie: inschatten hoe relevant informatie op dat moment is voor een gebruiker. In het ideale geval wordt alleen informatie gepresenteerd waar op dat moment behoefte aan is. Relevantie hangt af van de individuele persoon en verandert in overeenstemming met de werksituatie, waardoor het moeilijk automatisch te bepalen is. Ook voor het bepalen van relevantie zijn actieve en passieve benaderingen mogelijk. Personen kunnen expliciet hun interesse in bepaalde gebeurtenissen aangeven, mogelijk in relatie met specifieke werksituaties, bijvoorbeeld door in een lijst met

mogelijke gebeurtenissen de interessante te selecteren. Hoewel dit een nauwkeurige selectie van informatie toestaat, past het zich niet automatisch aan aan veranderende werksituaties. Bovendien is het vaak een vervelende en moeilijke taak voor de gebruiker.

Als alternatief kunnen systemen automatisch gegevens bijhouden die gerelateerd zijn aan een gebeurtenis, zoals het soort van gebeurtenis, zijn maker, of zijn lokatie. Automatische selectie houdt echter meestal geen rekening met individuele wensen van een groepslid. Daarom werkt een combinatie van beide benaderingen, met het gemak van automatische selectie en de flexibiliteit van filtering door de gebruikers, het beste om te bepalen of informatie relevant is voor een persoon. Applicaties die beide benaderingen ondersteunen, kunnen bijvoorbeeld gebruikers laten aangeven op welk niveau samengewerkt moet worden, waardoor wordt bepaald hoe gedetailleerd andere gebruikers op de hoogte worden gebracht van gebeurtenissen. In het geval van een losse samenwerking worden alleen die gebeurtenissen die een toestandverandering veroorzaken geselecteerd, in een nauwe samenwerking worden alle gebeurtenissen gevisualiseerd. Neem het geval waarin een rapport door een groep personen gezamenlijk bewerkt wordt. Wanneer personen de taak krijgen bepaalde hoofdstukken te schrijven, wil de persoon die verantwoordelijk is voor een bepaald hoofdstuk zeker op de hoogte gebracht worden van elke wijziging die daarin wordt aangebracht. Aan de andere kant zijn wijzigingen in stukken waar andere mensen verantwoordelijk voor zijn, niet zo interessant, hierbij telt alleen het eindresultaat.

Een simpele doch effectieve methode om de relevantie van een gebeurtenis te bepalen, is het meten van de afstand tussen de lokatie die geassocieerd wordt met de gebeurtenis en een lokatie die geassocieerd wordt met de gebruiker. In andere woorden; er wordt een interessegebied toegewezen aan elke gebruiker en aan elke gebeurtenis. Wanneer een interessegebied van een gebruiker en een gebeurtenis elkaar overlappen, wordt de bijbehorende informatie aan de gebruiker gepresenteerd. Ook de vorm van de presentatie kan afhangen van de afstand tussen gebeurtenis en gebruiker.

Het enkel weergeven van die gebeurtenissen die op dat moment voor de gebruiker zichtbaar zijn, is hier een variant op: het interessegebied van de gebruiker is de set van zichtbare delen op zijn scherm, en het interessegebied van een gebeurtenis wordt bepaald door de regio's die beïnvloed worden door de bijbehorende toestandsverandering. Het is beter wanneer de interessegebieden preciezer bepaald kunnen worden. Voor een gebeurtenis kan het een bepaald gebied zijn rond het punt in de interface waar veranderingen zichtbaar worden of waar een visualisatie of animatie plaats zou vinden. Het bepalen van interessegebieden van een gebruiker is lastiger, omdat het zowel statische interesse in bepaalde delen van een document moet bevatten, alsook de veranderlijke werksituatie. Om dit te bereiken kan een gebruikers interessegebied gedefinieerd worden als de som van verschillende gebieden: het momenteel zichtbare deel in de interface; de delen die de gebruiker expliciet heeft aangegeven als interessant; en de delen waar de gebruiker recent in heeft gewerkt. Een tijdsgrens bepaald dan hoe lang die laatste delen van invloed blijven.

### **5.3 Presentatie**

Na het proces van informatieselectie moet de informatie gepresenteerd worden. Er bestaat een veelvoud aan presentatievormen met verschillende eigenschappen en verschillende invloeden op de gebruiker. Mechanismen die minder opvallend zijn, vergroten ook de kans dat een gebruiker simpelweg informatie over het hoofd ziet. Daarom is gebruikersontvangst van informatie een belangrijk aspect. Ontvangst kan verdeeld worden in actieve en passieve vormen. In het actieve geval moeten gebruikers een actie ondernemen om de informatie te ontvangen; bijvoorbeeld door het openen van een informatievenster of door het klikken op een knop. In het passieve geval wordt de informatie op het scherm weergegeven zonder tussenkomst van de gebruiker; het staat gebruikers vrij de informatie op te nemen of te negeren.

Het passieve geval biedt voordelen: gebruikers kunnen de informatie waarnemen met perifeer zicht zonder afgeleid te worden van hun werk en systemen kunnen een continue stroom van informatie aanleveren. Natuurlijk zijn er gevallen waarin de actieve benadering de voorkeur heeft, bijvoorbeeld wanneer het om informatie gaat die anderen absoluut moeten zien, afgezien van of ze op dat moment naar het scherm kijken.

### **5.3.1 Statische en dynamische aspecten**

Er wordt onderscheid gemaakt tussen dynamische en statische aspecten van de presentatie van inzichtsinformatie. Statische informatie visualiseert de toestand van het systeem op elk moment. De presentatie is in zoverre statisch dat deze stabiel blijft zolang de toestand van het systeem onveranderd blijft. Daarnaast kan dynamische presentatie gebruikt worden om de dynamiek van het samenwerkingsproces weer te geven. Een dynamische presentatie visualiseert een toestandsverandering voor een bepaalde tijd; bijvoorbeeld in een animatie of een popup venster.

Deze twee basistechnieken zijn sterk gerelateerd aan elkaar: een verandering in de statische toestandspresentatie impliceert dat daarvoor een toestandsverandering en dus een dynamische presentatie heeft plaatsgevonden. Een dynamische kennisgeving heeft echter niet altijd tot gevolg dat de statische toestand verandert. Een kennisgeving kan informatie willen aangeven die simpelweg niet beschikbaar is in de toestandspresentatie. Wanneer een gebruiker bijvoorbeeld wordt gewezen op inkomende e-mail terwijl de mail-inbox niet wordt gerepresenteerd, blijft de statische presentatie van de toestand ongewijzigd.

Statische presentatie is bedoeld om tijdens het normale werk inzicht in het systeem te verkrijgen, met als belangrijkste doel een snelle en niet-storende informatie overdracht zonder overdaad. Dynamische presentatie daarentegen wijst gebruikers op specifieke gebeurtenissen, waarbij het niveau van intensiteit verhoogd wordt om ervoor te zorgen dat informatie op de juiste manier wordt opgenomen, zonder storend te zijn.

Beide technieken verschillen in hun ontwerpeisen. Statische presentatie is in feite een probleem van informatievevisualisatie. Dit is een bekend probleem waarvoor verschillende oplossingen bestaan, zoals venster systemen, 3D-omgevingen, of vervormingmethoden. Al deze technieken zijn ontwikkeld om de grote hoeveelheid informatie die een document bevat, weer te geven. Inzichtsinformatie is echter meta-informatie die niet in het document aanwezig is. Daarom wordt de selectie van te presenteren informatie een belangrijke kwestie: hoewel normaal gesproken informatie niet zomaar weggegooid of gewijzigd kan worden, is dat geen enkel probleem bij dit soort informatie. Dynamische presentatie is over het algemeen complexer te ontwerpen, omdat deze niet vaak is geïntegreerd in de algemene systeempresentatie, en dus geen gebruik kan maken van dezelfde technieken en metaforen. Daarnaast biedt dynamische presentatie natuurlijk meer problemen vanwege zijn dynamische karakter.

### **5.3.2 Medium**

De impact van een kennisgeving hangt voor een groot deel af van het presentatiemedium. Hoewel de mens over verschillende zintuigen beschikt, zijn er maar twee daarvan daadwerkelijk relevant voor kennisgeving: zicht en gehoor (Faulkner, 1998). De andere vormen - smaak, reuk, tast - hebben een aantal eigenschappen waardoor ze niet geschikt zijn als presentatiemedium. Ten eerste kunnen ze maar weinig inzichtsinformatie bevatten vergeleken met optische en geluidssignalen. Ten tweede worden ze in het dagelijks leven ook niet veel gebruikt. En ten slotte wordt hun gebruik niet ondersteund door computerhardware.

Aangezien de menselijke waarneming is geoptimaliseerd voor optische input en aangezien computers de meeste van hun informatie in visuele vorm weergeven, zijn visualisatietechnieken de meest belangrijke en gangbare methode voor kennisgeving. Natuurlijk kan gebruik van een zorgvuldig geselecteerd audiosignaal handig zijn, voornamelijk omdat ze geen visuele aandacht van de gebruiker eisen. Het is echter moeilijk ingewikkelde informatie alleen via audio signalen over te brengen. Bovendien wordt geluid vaak als storender ervaren dan beeld (Faulkner, 1998).

De meeste systemen gebruiken het scherm dan ook als voornaamste kennisgevingmedium, waarbij vaak ook gebruik wordt gemaakt van geluid om extra informatie weer te geven, bijvoorbeeld wanneer een deelnemer toetreedt tot een sessie.

### **5.3.3 Tijdelijke vervorming**

Vervorming is een sleutelbegrip voor kennisgeving. Technisch gesproken is vervorming een verandering van een inputsignaal om een aangepast outputsignaal te creëren. In het kennisgevingsproces is het inputsignaal de geselecteerde informatie. Het outputsignaal bestaat uit

de visualisatie van inzichtsinformatie. Er zijn twee vormen van vervorming te overwegen: ruimtelijke en tijdelijke. Ruimtelijke vervorming komt in paragraaf 5.3.4 aan bod.

Tijdelijke vervorming betekent dat de tijdelijke reeks of tijdinformatie van het inputsignaal wordt gewijzigd. Dit is het geval wanneer informatie asynchroon wordt gepresenteerd (bijvoorbeeld een geschiedenis presentatie), of wanneer de presentatie langer of korter duurt dan de oorspronkelijke actie (bijvoorbeeld bij een samengestelde animatie). Asynchrone kennisgeving is verbonden met gebruikersacties die in het verleden plaatsvonden. Synchrone kennisgeving is vooral bedoeld ter ondersteuning van het lopende werkproces.

Asynchrone kennisgeving is om verschillende redenen van belang:

*Voor herhaling:* deelnemers kunnen hun eigen of andermans vroegere acties herhalen om de huidige staat van het systeem te begrijpen.

*Om bij te werken:* niet elke deelnemer is voortdurend beschikbaar voor samenwerking. Daarom moeten deelnemers die afwezig zijn geweest kunnen scannen wat er gebeurd is tijdens hun afwezigheid.

*Voor conflictoplossing:* veranderingen kunnen bewaard worden om in geval van conflict te bepalen welke van de conflicterende acties te behouden.

Aangezien synchrone kennisgeving door het systeem wordt gecreëerd zonder expliciet gebruikersverzoek, is het voorkomen van gebruikersstoring en -afleiding hierbij de belangrijkste kwestie. Daarentegen wordt asynchrone kennisgeving meestal verwacht door de gebruiker; of omdat het actief wordt aangeroepen of omdat het gerelateerd is aan bepaalde gebruikersacties. In deze gevallen is het hoofdzakelijk te zorgen voor beknopte informatie terwijl informatieoverdaad wordt vermeden.

Hoewel dezelfde basistechnieken gebruikt kunnen worden, zal de hoeveelheid informatie die gebruikt wordt voor een asynchrone kennisgeving meestal groter zijn. Natuurlijk kan dit substantieel verminderd worden door technieken die bij synchrone kennisgeving niet beschikbaar zijn, zoals het groeperen van soortgelijke gebeurtenissen.

Het voornaamste tijdelijke vervormingsmechanisme is animatie. Omdat een animatie vaak is samengesteld, kan de presentatie ervan langer duren dan de oorspronkelijke actie. Animatie is bijzonder geschikt om toestandsveranderingen aan de gebruiker duidelijk te maken, aangezien beweging snel gezien wordt. Animatie zorgt voor een gevoel van continuïteit en kan makkelijk geïntegreerd worden in de toestandspresentatie. Het kost een programmeur echter veel moeite om animatie voor de verschillende toestandsveranderingen te implementeren.

Tijdelijke vervorming kan gebruikt worden als een alleenstaande techniek om toestandsveranderingen waarneembaar te maken. Het werkt echter het beste wanneer het geïntegreerd wordt in de achtergrond presentatie.

### **5.3.4 Ruimtelijke vervorming**

Ruimtelijke vervorming is een wijziging van de ruimte van de inzichtsinformatie. Aangezien dit een abstracte ruimte is, beperkt ruimtelijke vervorming zich niet tot veranderingen van de daadwerkelijke ruimtelijke lay-out van objecten op het scherm; het gaat ook om technieken als verandering van het presentatiemedium, het weglaten van informatie of het wijzigen van informatie. Systemen kunnen ervoor kiezen betrouwbare kennisgevingen over andermans activiteiten weer te geven, wat betekent dat een exacte kopie van de oorspronkelijke acties getoond wordt (WYSIWIS, zie paragraaf 5.3.5). Het kan echter nodig zijn een gebeurtenis te veranderen of samen te voegen om gegevens te tonen die anders niet zichtbaar zouden zijn, of om de informatie in een overzichtelijkere vorm te kunnen presenteren. Bovendien is het in veel gevallen niet de gebeurtenis die interessant is, maar de bedoeling die daaruit blijkt.

Verschillende benaderingen zijn mogelijk. Ten eerste kan een systeem onbelangrijke details weglaten om te voorkomen dat de gebruiker van belangrijke feiten wordt afgeleid. Het verminderen van de hoeveelheid informatie heeft twee voordelen: gebruikers hoeven minder informatie waar te nemen, en systemen hebben minder informatie te verwerken. Wanneer bijvoorbeeld in een animatie een beweging van een muis wordt nagespeeld, kan het pad hiervan berekend worden met alleen het begin- en eindpunt van de beweging, het is niet nodig letterlijk elke tussenstap weer te geven.

Ten tweede kan een systeem extra stappen toevoegen om dingen duidelijker te maken. Wanneer een actie bijvoorbeeld met een sneltoets is uitgevoerd, kan het systeem wel de uitgebreidere menuselectie met de muis laten zien, om duidelijk te maken hoe de actie is uitgevoerd.

Tenslotte is het heel goed mogelijk volledig samengestelde informatie te laten zien, die niet is gebaseerd op één specifieke gebeurtenis. Een systeem kan bijvoorbeeld een icoon van een gebruiker aan de telefoon laten zien, wanneer deze druk bezig is, al dan niet met telefoneren. Presentatie van samengestelde informatie is erg moeilijk te implementeren, aangezien het systeem tot op zekere hoogte moet afleiden wat de bedoelingen en acties van gebruikers zijn. Daarbij bestaat het gevaar dat gebruikers misleidende informatie ontvangen over wat anderen aan het doen zijn, wat juist niet de bedoeling is van het tonen van inzichtsinformatie.

Het bovenstaande deel is toepasbaar op zowel synchrone als asynchrone kennisgeving. Asynchrone kennisgeving biedt echter extra mogelijkheden. Gebeurtenissen kunnen namelijk opnieuw geschikt worden. Ze kunnen gehergroepeerd worden volgens een structuur die handig is in een bepaalde situatie. Zo kunnen bijvoorbeeld alle tekstveranderingen in een tekstverwerker op volgorde van voorkomen in de tekst worden geplaatst, onafhankelijk van in welke volgorde ze zijn uitgevoerd. Op dezelfde manier kunnen gebeurtenissen per gebruiker worden geordend, waardoor inzicht kan worden verkregen in een specifieke persoon.

### **5.3.5 Presentatiemethoden**

Ruimtelijke vervorming kan gebruikt worden om de aandacht van de gebruiker op een specifiek deel van de interface te richten, terwijl ondertussen aanvullende informatie getoond wordt, bijvoorbeeld door het oplichten van gebieden die gewijzigd zijn. Verschillende methoden om uiteindelijk de interface vorm te geven worden hier besproken. De in paragraaf 4.6 besproken fisheye methode valt hier overigens ook onder.

*WYSIWIS.* Een methode om groepsinterfaces vorm te geven staat bekend als WYSIWIS, wat staat voor “what you see is what I see”. Hierbij verschijnt de interface gegarandeerd hetzelfde bij alle deelnemers op het scherm, er wordt dus geen tijdelijke of ruimtelijke vervorming toegepast. De voordelen van WYSIWIS zijn een sterk gevoel van gedeelde context (mensen kunnen bijvoorbeeld naar iets verwijzen door zijn positie) en simpele implementatie. Het grote nadeel is dat het onflexibel kan zijn. Onderzoek heeft uitgewezen dat gebruikers graag zelf invloed willen uitoefenen op zaken als vensterplaats en grootte, of willen beschikken over aangepaste informatie in het venster.

WYSIWIS kan op verschillende punten variëren: de objecten waarop het wordt toegepast, de tijd van vertoning (wanneer schermen worden gesynchroniseerd), subgroep populatie (groep deelnemers betrokken of beïnvloed), en overeenstemming van het scherm (graad van visuele overeenstemming van de getoonde informatie)

*Gegroepeerde vensters.* Een methode om om te gaan met de beperkte schermruimte is om vensters te groeperen in functionele sets die elk corresponderen met een bepaalde taak. Als iemand dan een taak gaat uitvoeren, worden alle vensters die met die taak geassocieerd worden geopend. Een andere methode is één van de gebruikers belasten met het onderhouden van vensterorde.

*Overlappende vensters.* Een natuurlijke benadering is het openen van een apart venster om daarin inzichtsinformatie weer te geven. Dit heeft echter als groot nadeel dat de informatie in andere vensters geblokkeerd wordt wanneer het extra venster ze overlapt. Daarnaast is er in deze methode geen directe verbinding tussen de getoonde informatie en de inhoud van de werkruimte in de interface. Om te voorkomen dat gebruikers hun aandacht moeten verdelen tussen verschillende informatiebronnen is het beter om de informatie als achtergrondinformatie in de interface te presenteren (Dourish en Bellotti, 1992). Deze methode is echter alleen haalbaar wanneer de interface ruimte biedt voor een vloeiende en naadloze integratie van de aanvullende informatie, zonder van gebruikte interfacemetaforen af te wijken.

*Semi-transparante informatie lagen.* Een uitbreiding van de venterbenadering is het gebruiken van semi-transparante vensters als informatielaag. Dit vergroot de zichtbaarheid van het systeem, tegen de kosten van het afleiden van gebruikers van hun werk. Vanwege het speciale visuele uiterlijk is

het informatie venster snel te onderscheiden van de rest van de werkruimte. Door de graad van transparantie te veranderen, kan het venster aangepast worden aan de huidige werksituatie.

*Toekennen van betekenis.* Aan objecten kan een betekenis toegekend worden in de interface, zodat deze niet verder expliciet hoeft te worden gemaakt. De eerste methode om dit te doen is *ruimtelijk*, door het toewijzen van een bepaalde betekenis aan een bepaald gebied in de interface. Hierdoor krijgen objecten of delen daarvan die zich in dat gebied bevinden de bijpassende betekenis. Lokaties kunnen bepaald worden door insluiting (bijvoorbeeld in een kamer zijn), door nabijheid (bijvoorbeeld in de buurt van een bureau zijn), of figuratief (bijvoorbeeld het weergeven van documenten in een rechte lijn om een volgorde aan te geven).

Een andere methode is *symbolisch*. Door het toewijzen van een symbool aan een object krijgt het een betekenis. Symbolische relaties kunnen uitgedrukt worden door het gebruik van labels (namen, kleuren of iconen) of als attributen.

De derde methode is *referentieel*, door een object met een ander object te verbinden. Zulke verbindingen kunnen als expliciete visuele verbindingen worden weergegeven (door lijnen) of als bijpassende eigenschappen (door kleuren).

In een voorbeeld dat Berlage en Sohlenkamp (1999) bespreken (DIVA), wordt de inzichtsinformatie geïntegreerd in de gebruikte metafoor voor het systeem, in dit geval een kantoor met kamers en bureaus. Door weer te geven dat een actor een bepaalde taak uitvoert, kunnen andere gebruikers gemakkelijk opmerken wie waar wat aan het doen is. Elke gebruiker wordt gerepresenteerd door een icoon met zijn beeld en naam. Bovendien worden gebruikers geassocieerd met kleuren en worden deze kleuren gebruikt om acties van de bijbehorende gebruiker aan te geven.

#### **5.4 Conclusie**

Het kennisgevingsproces dat gebruikers van een multi-user systeem in staat stelt inzicht te verkrijgen in wat er gaande is, bestaat uit twee onderdelen; de selectie van de relevante informatie, en de presentatie van die informatie op een niet-storende manier in de interface (Bergel en Sohlenkamp, 1999; Gutwin en Greenberg, 1996). De beste kennisgevingmethode blijkt een passieve selectie door het systeem, aangezien de gebruikers hierbij geen extra werk hoeven te verrichten, en automatische presentatie van de informatie als achtergrondinformatie in de werkruimte in de interface, waardoor de gebruiker kan bepalen welke informatie tot zich te nemen op een zelfgekozen manier en tijdstip, zonder teveel afgeleid te worden (Dourish en Bellotti, 1992).

## 6 Keyworx

Om na te gaan in hoeverre de ideeën uit de literatuur in de werkelijkheid van toepassing zijn, is een bestaande multi-user applicatie nader onderzocht. In het bijzonder wordt onderzocht in welke mate de drie aspecten die voor problemen zorgen bij multi-user applicaties, sociale factoren, conflictbeheer en multi-user interfaces, het gebruik van de applicatie beïnvloeden. Ook kan zo onderzocht worden aan welke specifieke informatie gebruikers van multi-user applicaties behoefte hebben en hoe die informatie weergegeven kan worden.

In deze scriptie wordt de applicatie Keyworx onderzocht. Dit is een multi-user applicatie die door Waag Society / maatschappij voor oude en nieuwe media (Amsterdam) is ontwikkeld (<http://www.waag.org> en <http://www.keyworx.org>). In 1996 is een voorstel om deze applicatie te creëren aangenomen, vanaf 1998 zijn zij aangevangen met de daadwerkelijke ontwikkeling. Op dit moment wordt het verspreid over de wereld gebruikt en vinden op verschillende plekken regelmatig bijeenkomsten plaats waar de applicatie door meerdere personen wordt gebruikt en waar men de mogelijkheden ervan onderzoekt. Een van deze bijeenkomsten is Anatomic, die vrijwel elke zaterdagmiddag plaatsvindt in het theatrum anatomicum van de Waag. Deze anatomic-bijeenkomsten zijn voor dit onderzoek regelmatig bezocht en enkele anatomic-deelnemers hebben meegewerkt aan een hier beschreven gebruikersonderzoek.

### 6.1 Wat is Keyworx?

Keyworx is een applicatie waarmee meerdere gebruikers in staat worden gesteld om beeld, geluid en tekst te genereren, samen te voegen en te bewerken. Dit alles synchroon binnen een gedeelde virtuele werkruimte. Keyworx combineert multi-user omgevingen met real-time media processing tools, zogenaamde cross media synthesis. Het verschaft de instrumenten voor uitgebreide vormen van telecommunicatie en samenwerking. Deelnemers kunnen samenwerken in dezelfde fysieke ruimte door verbinding via een lokaal netwerk, of verspreid over de wereld door verbinding via het internet. Keyworx is in staat tot dynamisch samenvoegen van media van maximaal vijf gebruikers in een gedeelde werkruimte, waardoor het een krachtige live en/of performance tool is voor interactief en interdisciplinair werk.

De controle van de gebruikte media-modules is uitwisselbaar, zodat elke module door elke deelnemer bewerkt kan worden. Zo kan een persoon de grootte van een tekst bepalen met de positie van zijn muis, terwijl een ander de kleur van diezelfde tekst verandert met een toetsenbord. Er kan zelfs besloten worden de muis van een andere gebruiker te koppelen aan de tekst-grootte of een andere eigenschap.

Keyworx is ontworpen voor media professionals en artiesten uit uiteenlopende disciplines zoals muziek, theater, dans, performance art, video art, grafisch ontwerp en architectuur, die synchroon over lokale en niet-lokale netwerken in een samenwerkingsverband willen werken. De output van een performance of samenwerkingssessie kan worden geprojecteerd op een groot scherm in een theateromgeving, uitgezonden op televisie, of bekenen op het internet. De eindresultaten zijn flexibele, creatieve multi-user omgevingen en uitgebreide mogelijkheden voor genetwerkte, interdisciplinaire samenwerking en/of communicatie.

Mogelijke scenario's zijn:

- artiesten op verschillende lokaties creëren en besturen een performance die geprojecteerd wordt in een theateromgeving
- VJ's / DJ's besturen beeld en geluid via een lokaal netwerk in een club
- installaties die door buitenstaanders via sensoren op het internet gemanipuleerd worden
- telematische en live performance events
- multi-user gaming

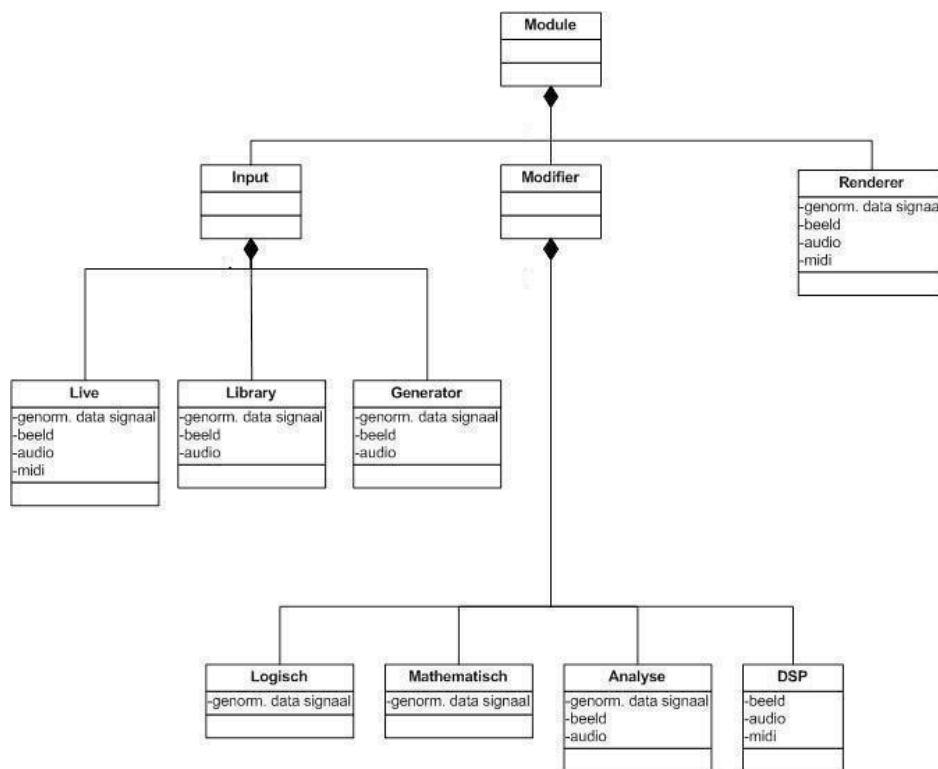
### 6.2 Keyworx opbouw

De keyworx applicatie bestaat uit twee onderdelen, de patcher en de realizer. In de patcher wordt het eigenlijke werk gedaan, hier worden media ingebracht en bewerkt. In de realizer is slechts de output te zien. Dit is het scherm dat getoond wordt aan een publiek tijdens optredens.

Een keyworx sessie is opgebouwd uit modules, die aan elkaar verbonden kunnen worden om een zogenaamde patch te vormen. Elk media type is te zien als een module met een aantal eigenschappen die bepalen hoe het eruit ziet of klinkt. In Keyworx is media alles wat een interessante digitale waarde kan exporteren of wat daaruit gecreëerd kan worden. Een film-bestand heeft bijvoorbeeld de eigenschappen volume, positie en snelheid. Een oscillator heeft de eigenschappen golflengte, fase en tempo. Er zijn vier soorten media, namelijk beeld, audio, midi en genormaliseerd data signaal, dat wil zeggen modules met een getalswaarde als output, zoals een oscillator. Deze vier mediasoorten zijn te herkennen in het menu door de gekleurde bolletjes voor de module-naam. Rood staat voor beeld, blauw voor audio, paars voor midi en groen voor genormaliseerd data signaal.

De modules (zie figuur 6.1) zijn ingedeeld in drie categorieën, namelijk:

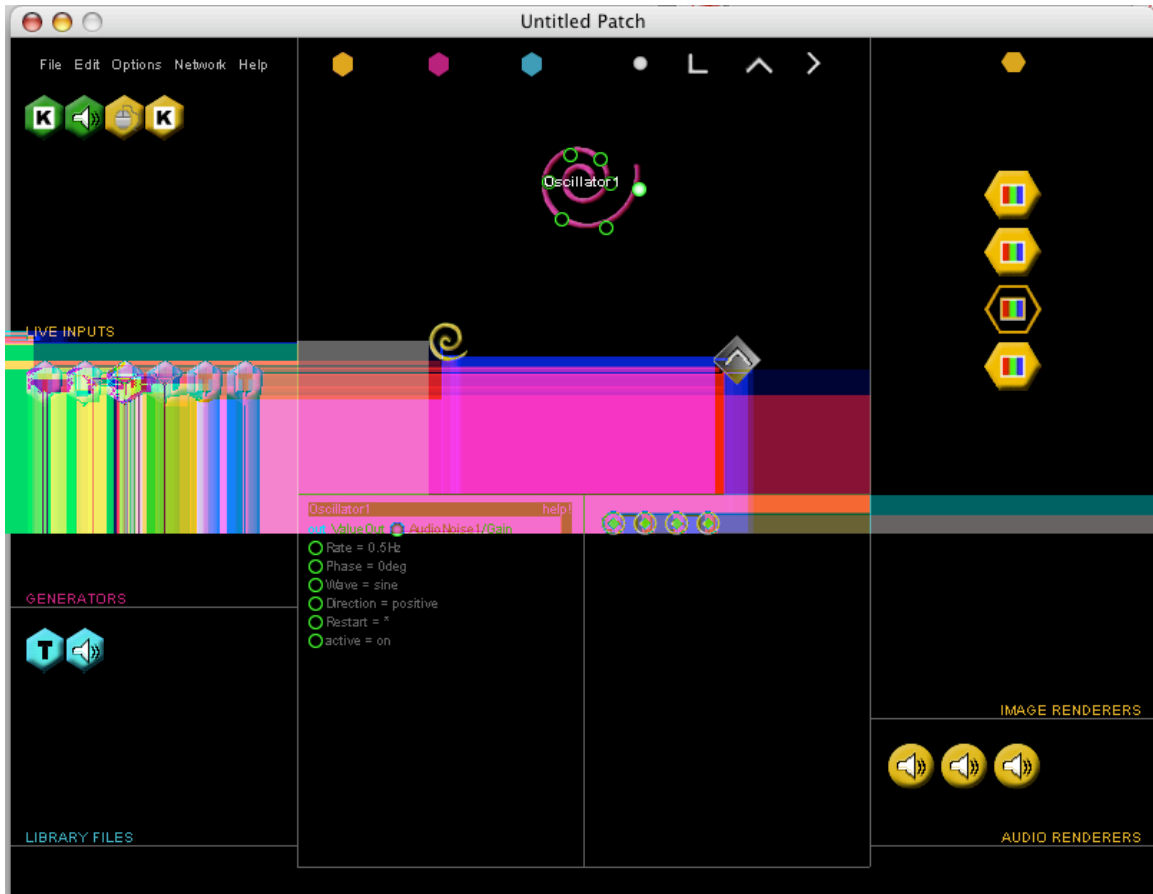
- Inputs: dit zijn media input bronnen, en kunnen zowel live als uit een library als gegenereerd zijn. Alle vier mediasoorten komen hier voor. Mogelijkheden zijn quicktime movies, tekst bestanden, audio bestanden, muis, toetsenbord, oscillator, random generator.
- Modifiers: deze functioneren als filters of modulatoren van inputs of andere modifier modules. Er is sprake van logische, mathematische, analyse en DSP (digital signal processing) modifier modules. De eerste drie zijn voornamelijk (wiskundige) formules met getalswaarden als output, de laatste zijn effecten en filters.
- Renderers : deze vertalen het signaal dat een input of modifier module aanvoert naar een weergave, waardoor het resultaat daadwerkelijk hoor- of zichtbaar wordt.



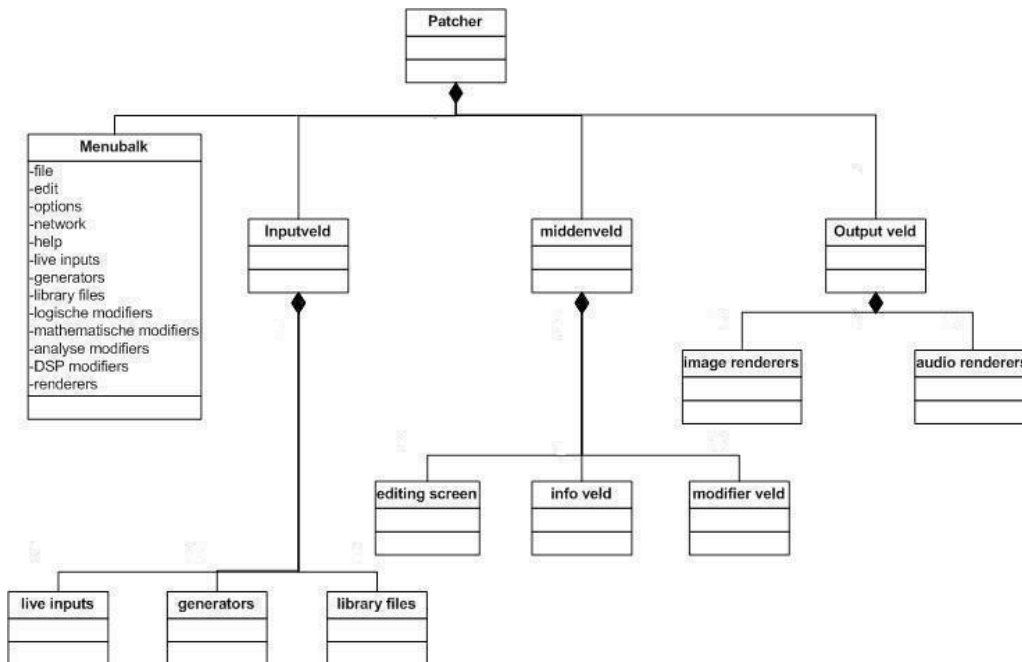
figuur 6.1: module opbouw

De interface van de patcher bevat verschillende onderdelen (zie figuur 6.2 en 6.3). Bovenin is een menubalk te vinden, met aan de linkerkant een conventionele menu-structuur en aan de rechterkant tekens waarmee de verschillende modules te selecteren zijn (gele zeshoek: live input, paarse zeshoek: generators, blauwe zeshoek: library files, ●: mathematische modifiers, L: logische, Λ: analyse, >: DSP, gele zeshoek: renderers).





figuur 6.2: screenshot van de patcher



figuur 6.3: onderdelen in de patcher

Aan de linkerkant van de patcher staan de input-modules, die zijn onderverdeeld in drie verschillende soorten, namelijk live inputs, generators en library files.

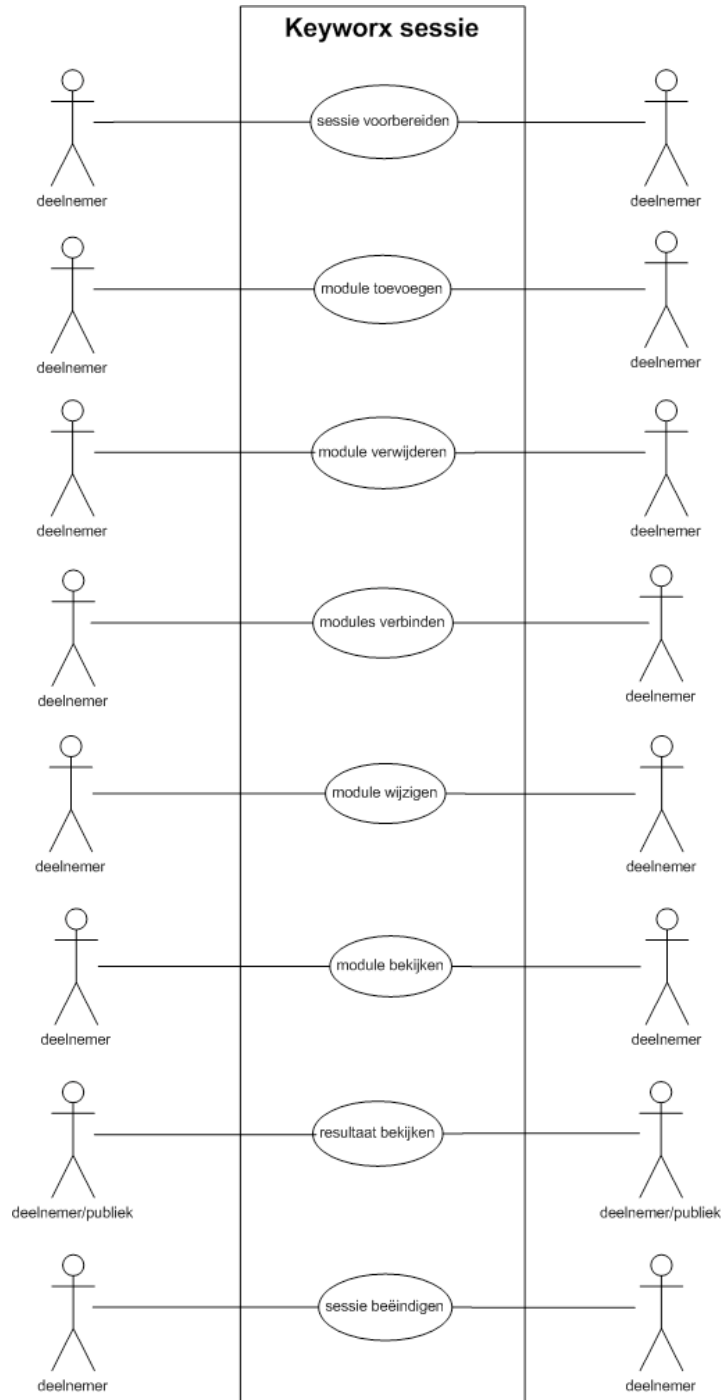
In het midden staat bovenaan het editing screen. Hierin worden de modules in uitgerolde spiraalvorm getoond na tweemaal erop klikken, om eventueel bewerkt te kunnen worden. Onder het editing screen is links het infoveld te vinden, waar eigenschappen van modules te zien zijn na eenmaal erop klikken om ook eventueel bewerkt te kunnen worden. Daarnaast zijn de modifiers te zien.

Aan de rechterkant van de patcher zijn de renderers te zien, bovenin de image renderers, onderin de audio renderers. Wanneer er meerdere outputs zijn kan het zijn dat de volgorde waarin ze staan niet klopt, waardoor een laag een andere blokkeert waardoor deze laatste niet zichtbaar is. In het image renderer veld kunnen de image renderers boven of onder elkaar worden gesleept om de volgorde van de visuele output-lagen eenvoudig te wijzigen wanneer dat wenselijk is.

### **6.3 Een keyworx sessie**

Een sessie vangt aan met het treffen van enkele voorbereidingen. Vervolgens kunnen in de patcher zeven soorten acties uitgevoerd worden (zie figuur 6.4). Een module kan worden toegevoegd, een module kan worden verwijderd, verbindingen tussen verschillende modules kunnen worden gemaakt, attribuutwaardes kunnen worden gewijzigd, eigenschappen van modules kunnen worden bekeken, het resultaat kan bekeken worden in het realizer-venster en de sessie kan beëindigd worden. Er moet minstens één module aangemaakt worden om überhaupt iets te kunnen doen. Vervolgens kunnen de verschillende acties, zoals het wijzigen, het verbinden of het toevoegen van modules, door elkaar heen lopen, de volgorde hiervan ligt niet vast.

Hiernaast kan het tijdens een sessie wenselijk zijn om op een directe manier met de andere deelnemers te kunnen communiceren. Er bestaat de mogelijkheid een chatvenster te openen en daarin berichtjes naar de andere gebruikers te typen. Ook kan het wenselijk zijn om het message window te bekijken. Hierin zijn alle deelnemers te zien en worden alle acties die de server uitvoert opgesomt. Hierdoor kan bijgehouden worden wat er op server-niveau allemaal gebeurt. Het chat- en messagevenster kunnen op elk ogenblik tijdens een sessie worden geopend.



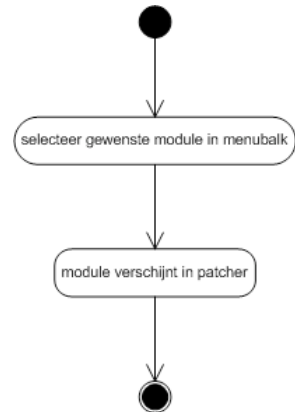
figuur 6.4: use-case diagram van een keyworx-sessie

*Vorbereiden sessie:*

Voordat met een sessie begonnen kan worden, zijn een aantal handelingen noodzakelijk (zie figuur 6.5). De tijdens een sessie te gebruiken media moet op één lokatie staan, waar de applicatie die kan vinden. Overigens is het handig wanneer iedere deelnemer dezelfde media op zijn computer heeft staan. De bestanden kunnen tijdens een sessie wel over en weer gezonden worden, maar dit vertraagt de sessie enorm. Vervolgens moet in overleg met alle deelnemers bepaald worden in welke ruimte gewerkt gaat worden. Er kan namelijk op de server gekozen worden uit verschillende ruimtes (KeySpaces). Wanneer dit allemaal gedaan is, kunnen de deelnemers met de server gaan verbinden.



figuur 6.5: voorbereiden van een sessie



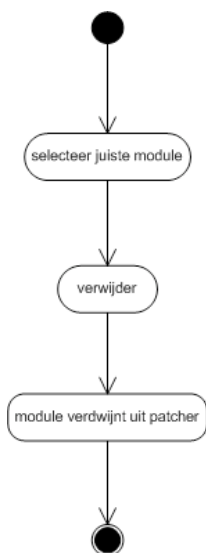
figuur 6.6: modules toevoegen

*Toevoegen van modules:*

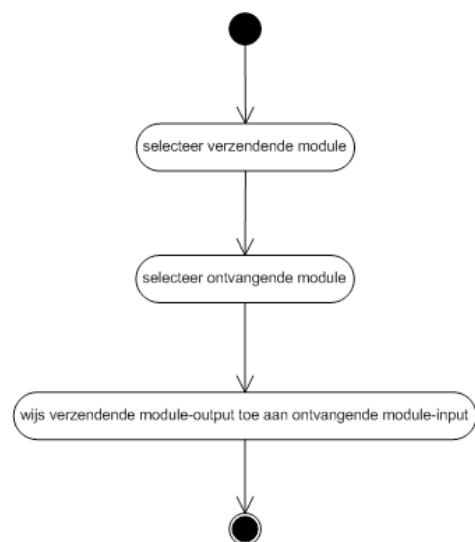
Het belangrijkste in een sessie is dat er iets te zien of horen is, er moet input aangemaakt worden. Hierboven werd al duidelijk dat er drie soorten input bestaan, namelijk live input, een bestand of een generator. Deze drie worden ieder op een vergelijkbare manier aangemaakt. De modules worden geselecteerd door middel van pulldown menu's waarna ze op de daarvoor bestemde plek in de patcher verschijnen (zie figuur 6.6).

*Verwijderen van modules:*

Het verwijderen van modules gebeurt heel eenvoudig door een module te selecteren en deze vervolgens te verwijderen (zie figuur 6.7).



figuur 6.7: modules verwijderen

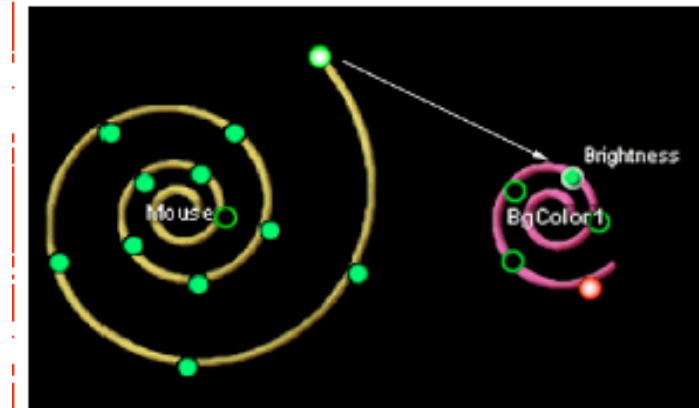


figuur 6.8: verbinden van modules

*Verbinden van modules:*

Elke module bestaat uit één of meerdere inputs en één of meerdere outputs. Ook de inputs en outputs zijn onderverdeeld naar soort media (beeld, audio, midi en genormaliseerd data signaal). Alleen dezelfde soorten media kunnen met elkaar verbonden worden, dus een audio-output kan alleen met een audio-input verbonden worden, wat te zien is aan de kleur van de bolletjes.

Modules worden met elkaar verbonden door de waarde van een output van de ene module toe te wijzen aan de input-waarde van een andere module (zie figuur 6.8). Dit gebeurt door het slepen van bolletjes, zoals te zien is in figuur 6.9. Hier wordt een output van een muis verbonden met de helderheid van de achtergrondkleur. Deze verbinding kan vervolgens verbroken worden door de output (het bolletje) van de ene module weer uit de input van de andere te slepen.



figuur 6.9: het verbinden van modules door het slepen van bolletjes

De volgorde van verbinden staat vast: als eerste een input, dan eventueel een of meerdere modifiers, en vervolgens een renderer.

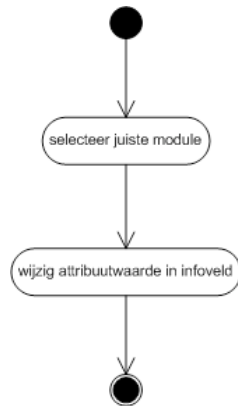
Wanneer als input een generator is gekozen, kan het zijn dat er niets gebeurt in de realizer. Dit hangt af van het soort generator, veel van de generators geven als output namelijk enkel een signaal dat op zichzelf niets doet. Wanneer je deze echter aan een module verbindt, wordt de generator pas zichtbaar. Een 'swarm' of een 'ImageWebCrawl' zijn generators waarvan het resultaat direct in de realizer zichtbaar wordt. Een swarm tekent als het ware een zwerm stipjes in de output, een ImageWebCrawl zoekt op Google Images naar plaatjes aan de hand van een in te voeren trefwoord en laat die een voor een zien. Een fractal of oscillator daarentegen zijn voorbeelden van generators die niet direct zichtbaar zijn, zij genereren een getalswaarde. Zo is het mogelijk de output van een oscillator te verbinden met de tekstgrootte van een keyboard, waardoor het effect van de oscillator eigenlijk pas zichtbaar wordt in de variërende tekstgrootte.

#### *Wijzigen van attribuutwaardes:*

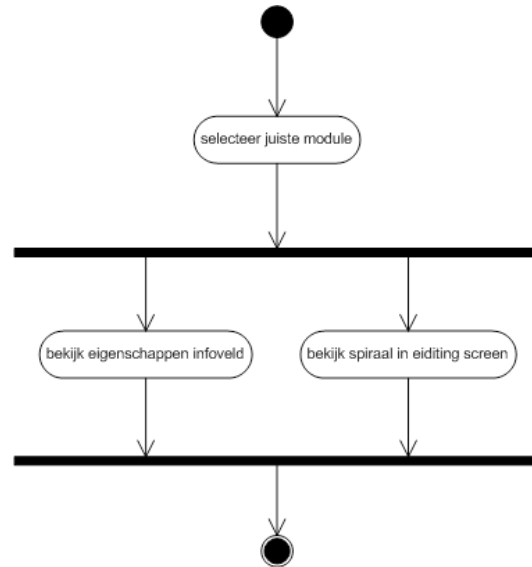
Elke module bevat attribuutwaardes, welke afhankelijk zijn van het type. Een voorbeeld is een tekst-renderer module. Het heeft als belangrijkste input een CharsIn (karakters in), maar daarnaast ook nog attributen als hue, horizontal offset, horizontal scale, brightness, saturation, mode, size, vertical offset en vertical scale. Deze eigenschappen kunnen op twee manieren gewijzigd worden. Er kan een output van een andere module aan verbonden worden (zie 'verbinden van modules' hierboven), of de waarde kan in het overzicht van de eigenschappen in het infoveld gewijzigd worden (figuur 6.10). De tekst size bijvoorbeeld kan op twee manieren gewijzigd worden. De output van een oscillator-module kan verbonden worden met de size-parameter, zoals hierboven besproken, of de gewenste grootte kan bij de size-parameter in het infoveld ingetypt worden.

#### *Bekijken van modules:*

Gedurende de hele sessie kunnen de aanwezige modules bekeken worden in het infoveld of het editing screen (figuur 6.11). Hierdoor kan een gebruiker inzicht krijgen in de eigenschap van de betreffende module of in de verbindingen die die module heeft met andere modules. Hierdoor kan dus inzicht ontstaan in de werking van aanwezige modules en de opbouw van een patch.



figuur 6.10: wijzigen attribuuwaares in infoveld



figuur 6.11: bekijken van modules

*Resultaat bekijken:*

Het resultaat van al het werk dat gedaan wordt in de patcher is in het realizer-venster te bekijken. Dit is enkel een venster met de output, meer niet. De patcher en de realizer kunnen eenvoudig afwisselend bekeken worden door op escape te drukken. Hierdoor wordt op dat moment het niet-vooraanstaande venster naar voren gehaald zodat daar in gewerkt of gekeken kan worden.

*Sessie beëindigen:*

Het beëindigen van een sessie gaat erg eenvoudig door het afsluiten van zowel de patcher als de realizer.

**6.4 Gebruikersonderzoek - opzet**

Het gebruikersonderzoek naar de applicatie Keyworx vindt in drie sessies plaats. Aan het onderzoek nemen in totaal zeven personen verdeeld over drie sessie deel. De volgende stappen worden hierbij doorlopen:

1. korte, algemene vragenlijst
2. sessie met hardop reflectie, vastgelegd door camera's
3. uitgebreide, specifieke vragenlijst
4. nabespreking

Om te beginnen krijgen de deelnemers een korte vragenlijst met algemene vragen (toegevoegd als bijlage 1). Deze dient om een inzicht te krijgen in hun achtergrond en computer-, keyworx- en multi-user ervaring. Vervolgens gaan de deelnemers in een multi-user setting zo'n 50 minuten werken met de applicatie. Hierbij wordt gevraagd of zij hardop kunnen reflecteren op de acties die zij uitvoeren, wat hun ideeën zijn en wat zij zien gebeuren op het scherm. Dit wordt geregistreerd door camera's. Dit is bedoeld om inzicht te krijgen in welke problemen gebruikers van Keyworx ervaren tijdens een multi-user sessie.

Na afloop van de multi-user sessie volgt een uitgebreide vragenlijst met specifieke vragen over de zojuist gehouden sessie, het groepsproces en visualisatie van informatie (toegevoegd als bijlage 2). Voorbeelden van vragen zijn: "Hoe vind je het dat anderen jouw dingen kunnen veranderen?", "Zou je willen zien wat de andere deelnemers doen?", "Heb je wel eens behoefte aan een overzicht van wat er tot nu toe is gebeurd?" De verwachting is dat, met de gehouden sessie nog vers in het geheugen, de antwoorden op deze vragen inzicht geven in de ervaringen en problemen die men tegen kan komen tijdens sessies.

Ten slotte volgt een discussie waarin de deelnemers uitgebreider aan kunnen geven wat hun

ervaringen met de applicatie zijn en wat voor ideeën zij hebben over het verbeteren van de informatievoorziening daarbinnen. Er wordt verwacht dat de gebruikers ideeën hebben over welke informatie zij gevisualiseerd willen zien en eventueel zelfs over een manier daarvoor.

Dit gebruikersonderzoek zal de volgende data opleveren:

- 7 korte, ingevulde vragenlijsten als inleiding
- 7 uitgebreide, ingevulde vragenlijsten
- 7 videotapes van de verschillende deelnemers tijdens de sessies
- 3 videotapes van de discussies
- datalogs van de verschillende sessies

### 6.5 Analyse gebruikersonderzoek – video’s

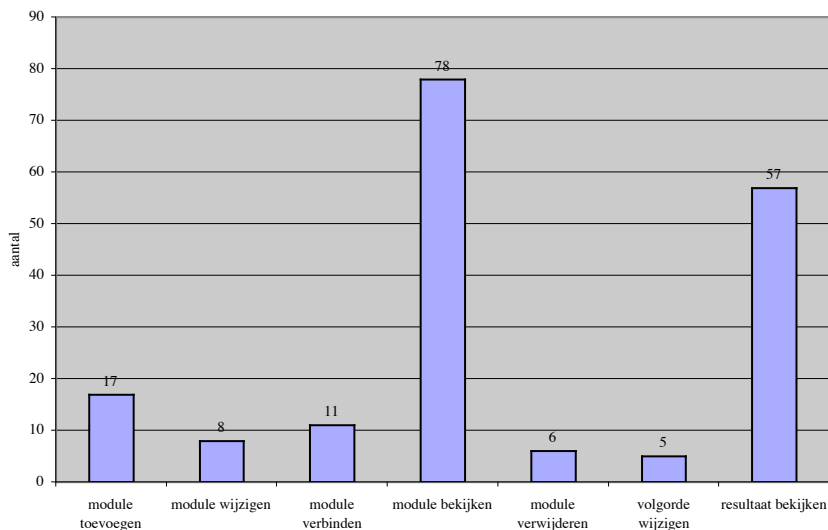
In de UML-schema’s is te zien welke acties een gebruiker moet uitvoeren om een bepaald doel te bereiken, zoals het verbinden van modules. Uit de videoanalyse blijkt echter dat dit nogal eens mis gaat. Gebruikers weten niet hoe ze een bepaald doel zo efficiënt mogelijk kunnen bereiken, of het systeem reageert anders dan zij verwachten.

Hieronder volgt een opsomming van alle gebruikers en hoe vaak zij alle mogelijke acties (uit figuur 6.4) hebben uitgevoerd. Op deze manier is een overzichtelijke, kwalitatieve data-analyse te maken door bij te houden hoe vaak gebruikers een bepaalde actie uitvoeren. Ook blijkt hieruit welke problemen daarbij voorkomen. Deze problemen worden vervolgens besproken in paragraaf 6.5.8.

#### 6.5.1 Gebruiker A

Gebruiker is 28 jaar, studeert in de media/technische richting, heeft veel computerervaring (5 op een schaal van 1 tot 5), heeft meer ervaring met multi-user applicaties (voornamelijk IRC-channels), heeft niet zo veel ervaring met Keyworx (2 op schaal van 1 tot 5), gebruikt Keyworx voornamelijk voor werk en heeft redelijk veel overeenkomsten in beroep/studie en de werkzaamheden met Keyworx. Gebruiker A deed mee aan een sessie met gebruikers C en F. De sessie liep eenmaal vast. In totaal heeft gebruiker A 184 acties uitgevoerd. De absolute verdeling daarvan over de verschillende acties is hieronder in figuur 6.12 te zien.

gebruiker A: alle acties



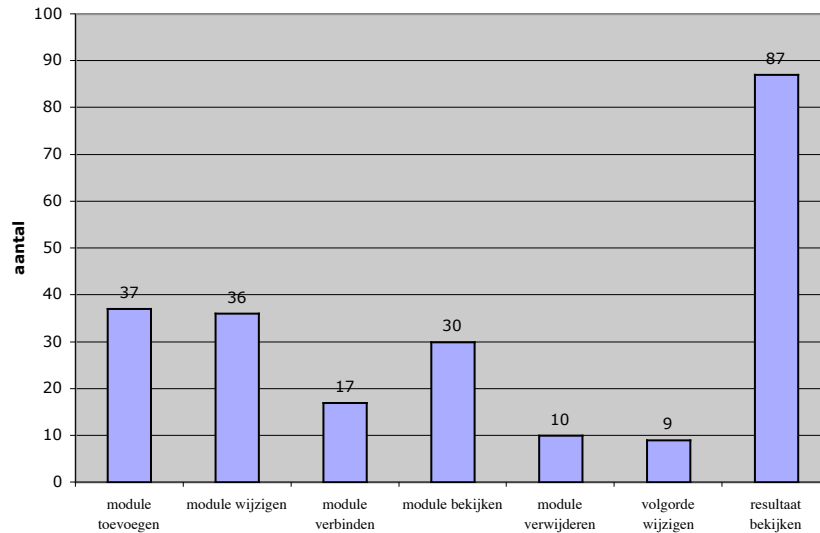
figuur 6.12: overzicht van alle door gebruiker A uitgevoerde acties

#### 6.5.2 Gebruiker B

Gebruiker B is 29 jaar en is software developer van onder andere Keyworx. De algemene computer-ervaring is veel (5 op schaal van 1 tot 5), meer ervaring met multi-user applicaties is aanwezig (zoals Ichat en games), multi-user ervaring met Keyworx is vrij veel (4 op schaal van 1 tot 5),

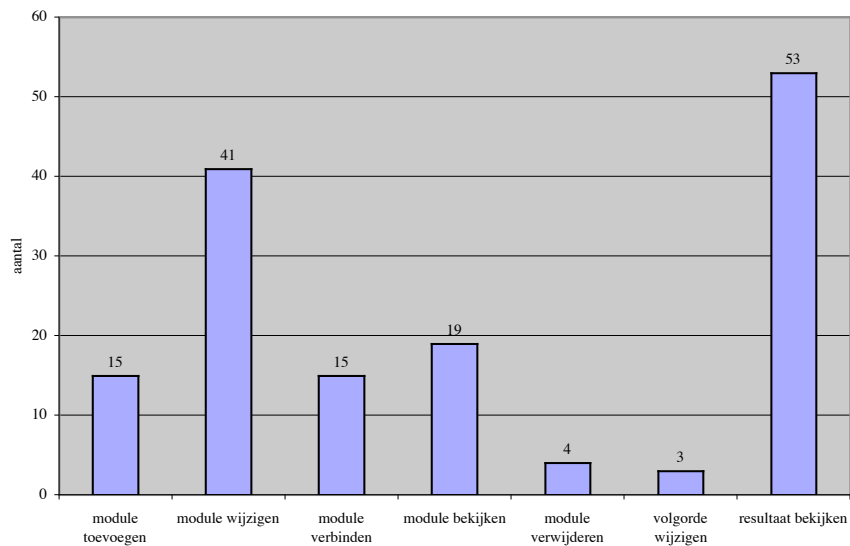
Keyworx wordt voor zowel werk als plezier gebruikt en beroep en studie komen veel overeen met de werkzaamheden met Keyworx. Gebruiker B deed mee aan een sessie met gebruiker D. Hierbij werd het messagewindow geopend om af te toe iets te kunnen bekijken, zoals de aanwezigheid van de andere gebruiker. Tijdens deze sessie werd tweemaal het programma herstart omdat het niet naar behoren werkte. Drie maal is het vastgelopen. In totaal heeft gebruiker B 226 acties uitgevoerd. De absolute verdeling daarvan over de verschillende acties is in figuur 6.13 te zien.

**gebruiker B: alle acties**



*figuur 6.13: overzicht van alle door gebruiker B uitgevoerde acties*

**gebruiker C: alle acties**



*figuur 6.14: overzicht alle door gebruiker C uitgevoerde acties*

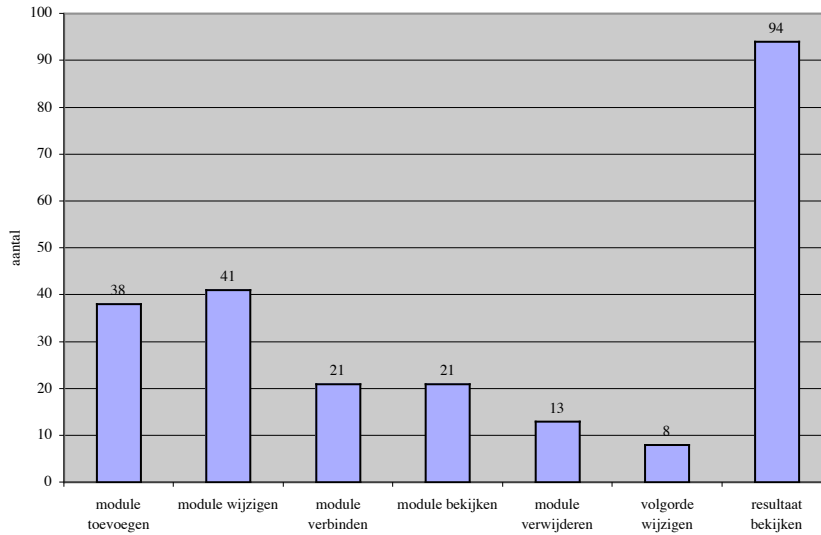
### 6.5.3 Gebruiker C

Gebruiker C is 32 jaar oud, kunstenaar, heeft vrij veel computer-ervaring (4 op schaal van 1 tot 5), heeft vrij veel ervaring met Keyworx (4 op schaal van 1 tot 5), geen verdere ervaring met multi-user applicaties, gebruikt Keyworx voor werk en heeft veel overeenkomsten tussen Keyworx en werk. Gebruiker C deed mee aan een sessie met gebruikers A en F. Hierbij werd ook het messagewindow



geopend. De sessie liep eenmaal vast. In totaal heeft gebruiker C 150 acties uitgevoerd. De absolute verdeling daarvan over de verschillende acties is in figuur 6.14 te zien.

**gebruiker D: alle acties**

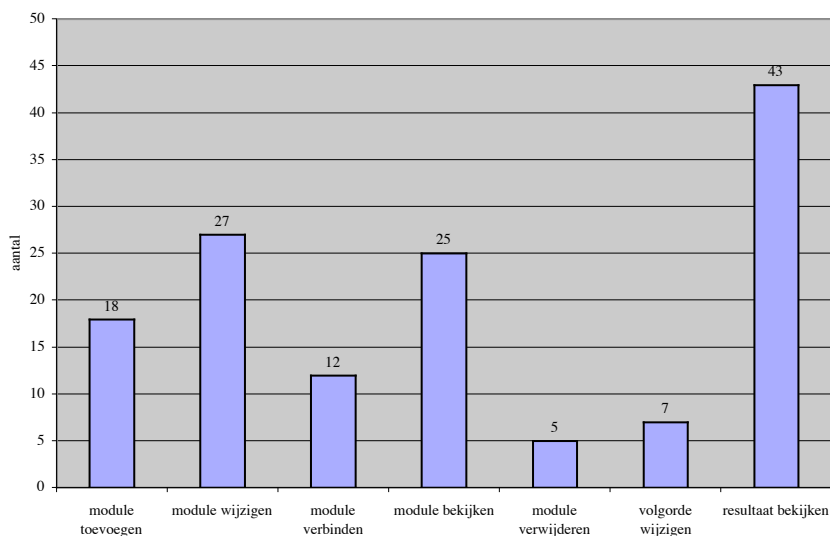


*figuur 6.15: overzicht van alle door gebruiker D uitgevoerde acties*

#### 6.5.4 Gebruiker D

Gebruiker D is 25 jaar oud, computer programmeur (onder andere van Keyworx), veel computer-ervaring (5 op schaal van 1 tot 5), heeft veel ervaring met Keyworx (5 op schaal van 1 tot 5), meer ervaring met multi-user applicaties is aanwezig (zoals chatten en games), gebruikt Keyworx voor werk en plezier en heeft 100% overeenkomst tussen Keyworx en werk. Gebruiker D deed mee aan een sessie met gebruiker B. Hierbij werd ook het messagewindow geopend, waarin vrij vaak werd gekeken (10 maal). De sessie liep drie maal vast en werd twee maal herstart. In totaal heeft gebruiker D 236 acties uitgevoerd. De absolute verdeling daarvan over de verschillende acties is in figuur 6.15 te zien.

**gebruiker E: alle acties**



*figuur 6.16: overzicht van alle door gebruiker E uitgevoerde acties*

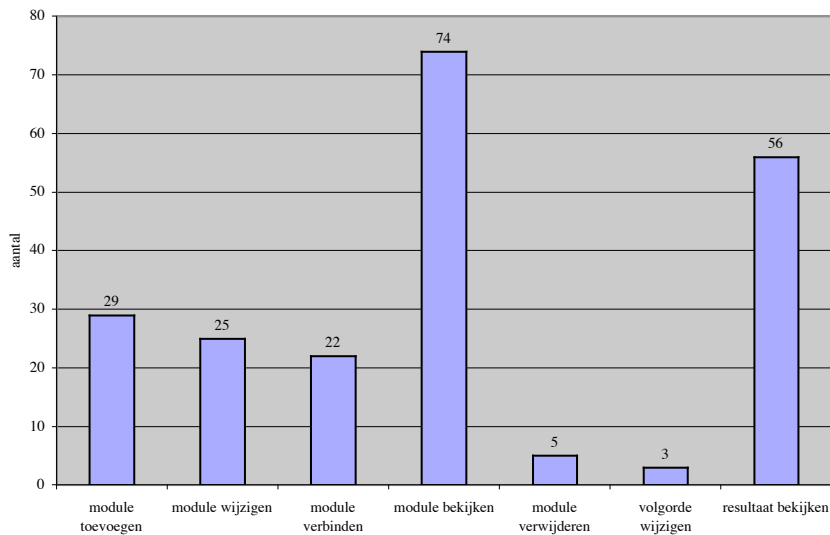
### 6.5.5 Gebruiker E

Gebruiker E is 26 jaar oud, student in de media/technische richting, veel computer-ervaring (5 op schaal van 1 tot 5), heeft weinig ervaring met Keyworx (1 op schaal van 1 tot 5), weinig ervaring met multi-user applicaties is aanwezig (wat games), gebruikt Keyworx voornamelijk voor werk (onderzoek), kan de studie toepassen op Keyworx, maar de eigenlijke werkzaamheden met Keyworx en in de studie vertonen niet veel overeenkomsten. Gebruiker E deed mee aan een sessie met gebruiker G en de onderzoeker (die meedeed om de sessie een groter multi-user aspect te geven). De sessie liep tweemaal vast. In totaal heeft gebruiker E 137 acties uitgevoerd. De absolute verdeling daarvan over de verschillende acties is in figuur 6.16 te zien.

### 6.5.6 Gebruiker F

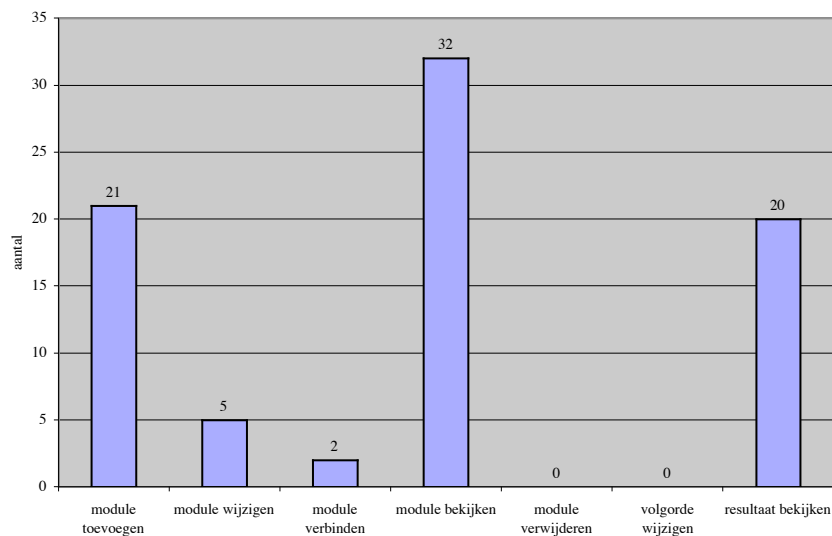
Gebruiker F is 26 jaar oud, heeft informatica gestudeerd, heeft veel computer-ervaring (5 op schaal van 1 tot 5), heeft weinig ervaring met Keyworx (1 op schaal van 1 tot 5), geen ervaring met multi-user applicaties, gebruikt Keyworx voornamelijk voor plezier, maar zeer weinig en heeft af en toe projecten die enige overeenkomst vertonen met de werkzaamheden met Keyworx. Gebruiker F deed mee aan een sessie met gebruikers A en C. De sessie liep eenmaal vast. In totaal heeft gebruiker F 214 acties uitgevoerd. De absolute verdeling daarvan over de verschillende acties is in figuur 6.17 te zien.

gebruiker F: alle acties



figuur 6.17: overzicht van alle door gebruiker F uitgevoerde acties

### gebruiker G: alle acties



figuur 6.18: overzicht van alle door gebruiker G uitgevoerde acties

#### 6.5.7 Gebruiker G

Gebruiker G is 40 jaar oud, media producent (voornamelijk theater), heeft wat computer-ervaring (3 op schaal van 1 tot 5), heeft weinig ervaring met Keyworx (1 op schaal van 1 tot 5), meer ervaring met multi-user applicaties (MUD's, chatten), gebruikt Keyworx voor zowel werk als plezier en heeft grote overeenkomst tussen de werkzaamheden met Keyworx en beroep. Gebruiker G deed mee aan een sessie met gebruiker E en de onderzoeker (die meedeed om de sessie een groter multi-user aspect te geven). De sessie liep tweemaal vast. In totaal heeft gebruiker G 80 acties uitgevoerd. De absolute verdeling daarvan over de verschillende acties is in figuur 6.18 te zien.

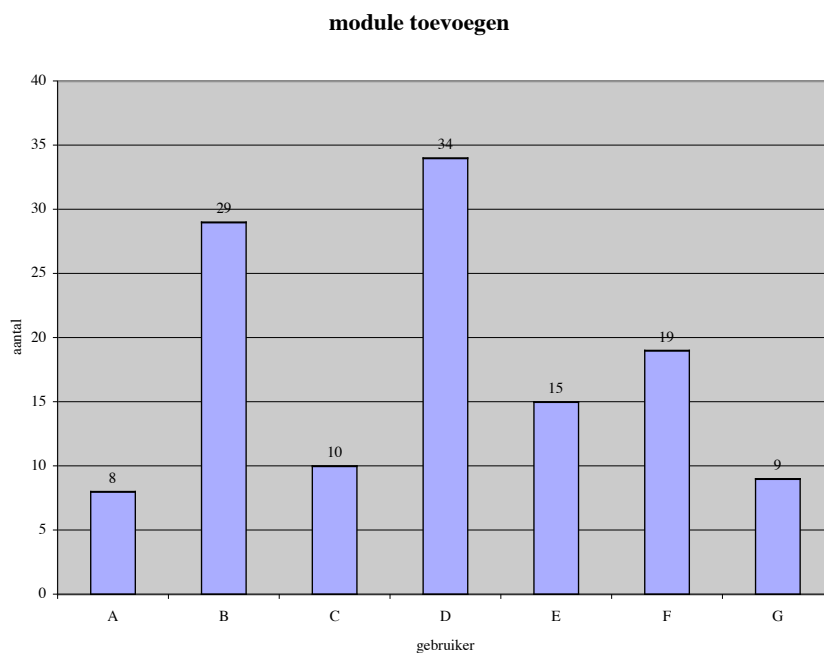
#### 6.5.8 Alle gebruikers

Wanneer alle gebruikers tezamen bekeken worden, is het interessant de ervaren en de minder ervaren gebruikers tegenover elkaar te zetten. De mate van ervarenheid kan bepaald worden door de algemene computer-ervaring op te tellen bij de Keyworx-ervaring. Op die manier ontstaat de volgende rangorde, van meer naar minder ervaring:

Gebruiker	Totale ervaring
D	10
B	9
C	8
A	7
E	6
F	6
G	4

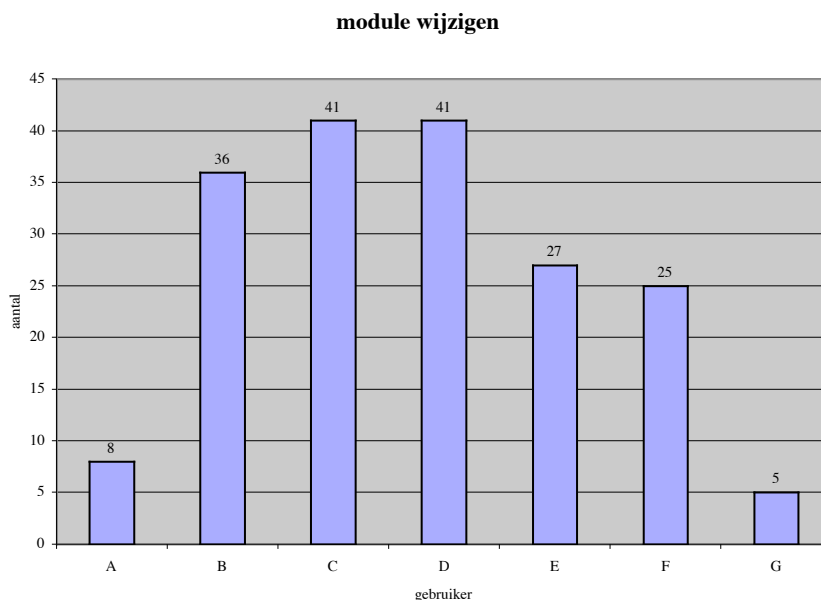
Figuur 6. 19: rangschikking van gebruikers naar totale ervaring

Wat opvalt, is dat de minst ervaren gebruiker G ook duidelijk de minste acties heeft uitgevoerd, namelijk 80. Dit kan veroorzaakt worden doordat alles voor deze persoon moeilijk te volgen was en elke actie dus veel moeite kostte.



*figuur 6.20: overzicht module toevoegen voor alle gebruikers*

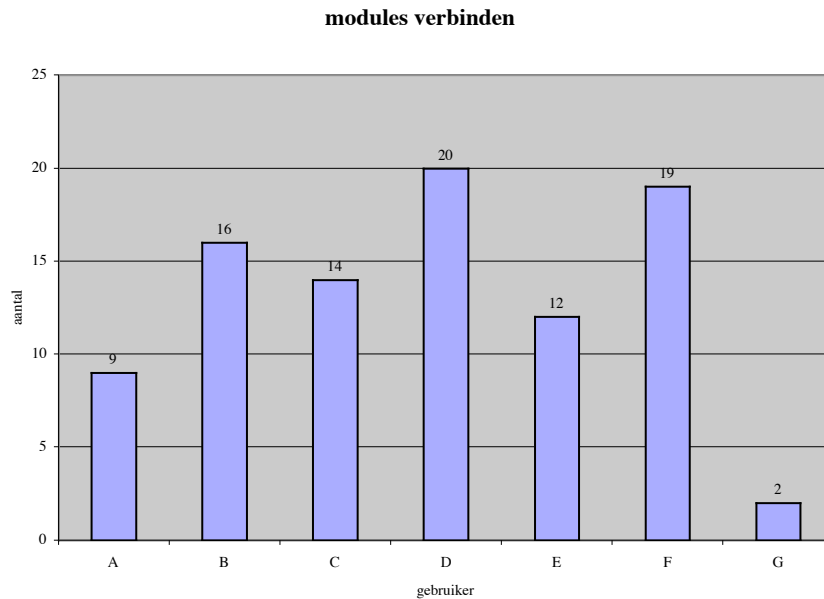
In figuur 6.20 is te zien hoe vaak elke gebruiker de actie ‘module toevoegen’ goed heeft uitgevoerd. Bij deze actie kan het voorkomen dat een gebruiker niet meteen weet waar een bepaalde module te vinden is, onder welk menu-item, zodat daar eerst naar gezocht moet worden in de menu’s. Wat ook wel eens voorkomt, is dat een gebruiker een module toevoegt, die dan even in de patcher verschijnt en vervolgens weer verdwijnt. Dit is een systeem-fout, de gebruiker kan hier niets aan doen, maar blijft wel verbaasd achter. In het diagram is te zien dat de meest ervaren gebruikers B en D ook de meeste modules toevoegen. De derde ervaren gebruiker, gebruiker C, wijkt echter af van deze trend.



*figuur 6.21: overzicht module wijzigen voor alle gebruikers*

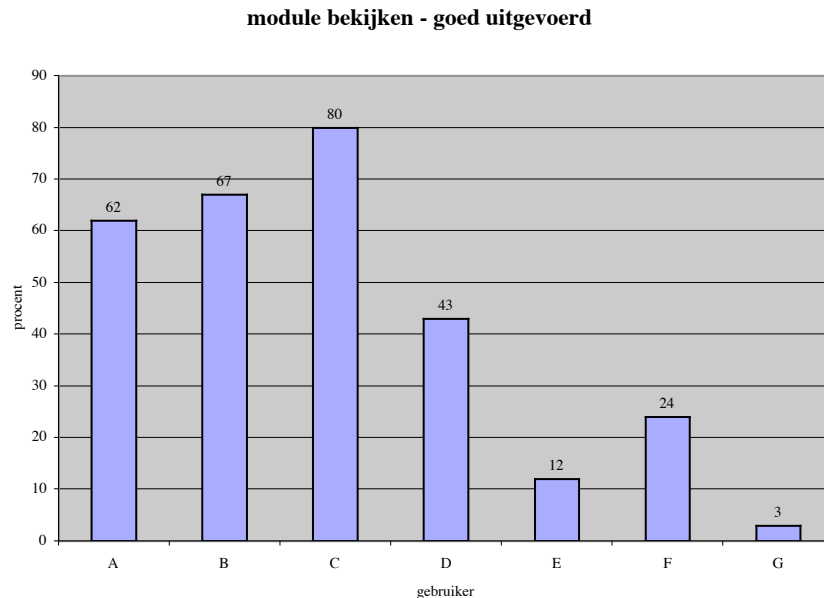
In figuur 6.21 is weergegeven hoe vaak alle gebruikers de actie ‘module wijzigen’ hebben uitgevoerd. Ook bij deze actie is te zien dat de meest ervaren gebruikers B, C en D de actie het

meest uitvoeren. Gebruiker A wijkt vervolgens af. Wel is het zo dat de minst ervaren gebruiker G de actie ook het minst vaak heeft uitgevoerd.



figuur 6.22: overzicht modules verbinden voor alle gebruikers

In figuur 6.22 is te zien hoe vaak de actie ‘modules verbinden’ juist is uitgevoerd. Verbindingen kunnen ook verwijderd worden of niet mogelijk zijn vanwege niet overeenkomende media-types. Dit komt echter vrijwel niet voor en is dan ook niet opgenomen in dit diagram. Uit de figuur is weinig af te leiden. De meest ervaren gebruiker voert de actie weliswaar het vaakst uit, maar daarna komt gebruiker F. Wel heeft gebruiker G wederom de actie het minst vaak uitgevoerd.



figuur 6.23: percentage van goed uitgevoerde acties ‘module bekijken’

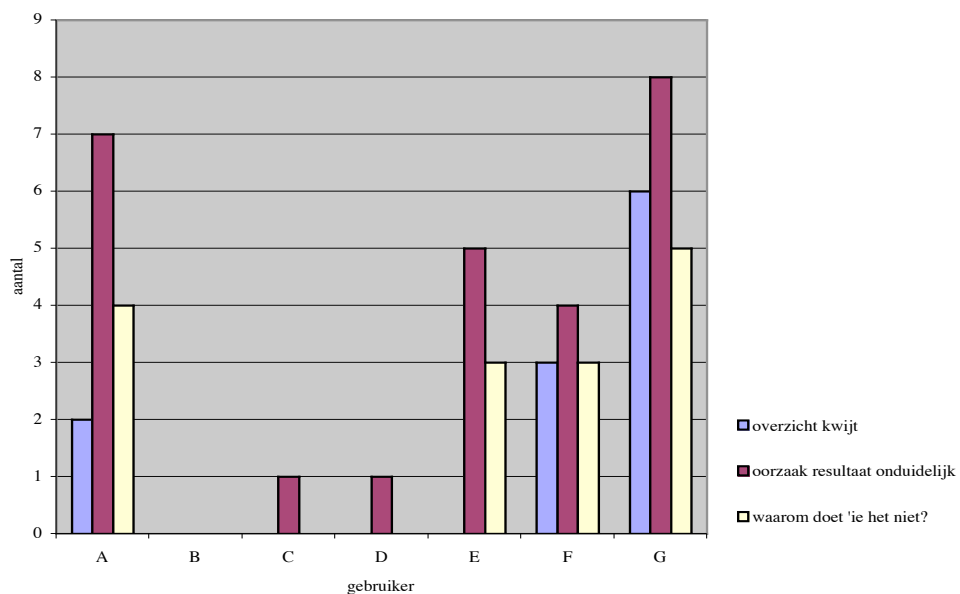
Uit de analyse van de actie ‘module bekijken’ vallen een aantal interessante dingen af te leiden. Uit de videoanalyses blijkt dat een duidelijk onderscheid te maken valt tussen het goed verlopen van de actie en het uitvoeren van de actie omdat gebruikers niet meer begrijpen wat er gaande is of waarom iets gebeurt. Het komt voor dat gebruikers de opbouw van een patch niet onmiddellijk doorzien. Een gebruiker kan dan de bestaande verbindingen nagaan, door van module naar verbonden module te

klikken. Uit de video's blijkt echter dat het voorkomt dat een gebruiker dan nog steeds het overzicht kwijt is, dat niet helder wordt hoe de patches in elkaar zitten. Het komt ook voor dat onduidelijk is waardoor een bepaald effect in de output veroorzaakt wordt, waarna er naar de veroorzakende module(s) op zoek gegaan wordt. Het gebeurt ook dat een gebruiker iets probeert te doen dat maar niet lukt en dat niet duidelijk wordt waarom dat dan niet werkt. Deze drie gevallen zijn dus met name interessant: 'overzicht kwijt', 'oorzaak resultaat onduidelijk' en 'waarom doet 'ie het niet?'

In figuur 6.23 is te zien hoe vaak de actie 'module bekijken' goed is uitgevoerd. Dit wordt relatief weergegeven, als percentage van hoe vaak die gebruiker de actie 'module bekijken' heeft uitgevoerd, aangezien hoe vaak de actie 'module bekijken' is uitgevoerd per gebruiker behoorlijk verschilt en daar rekening mee dient te worden gehouden. Hieruit is af te leiden dat de vier meest ervaren gebruikers A, B, C en D de actie ook relatief het vaakst juist uitvoeren. Vooral de minst ervaren gebruiker G behaalt een laag percentage.

In figuur 6.24 wordt uitgezet hoe vaak de gebruikers niet meer volledig konden volgen wat er gebeurde. De eerste staaf geeft aan hoe vaak men het overzicht kwijt was, dat zijn dus de gevallen waarbij na het bekijken van de modules of het nalopen van de verbindingen nog steeds niet duidelijk was wat er aan de hand was. Merk op dat dit eigenlijk helemaal niet voor zou moeten komen. De tweede staaf geeft aan hoe vaak de oorzaak van een resultaat niet duidelijk was en men daar vervolgens naar op zoek ging. De derde staaf geeft aan hoe vaak gebruikers niet begrepen waarom de applicatie niet deed wat zij verwachtten.

**module bekijken - onduidelijkheid**

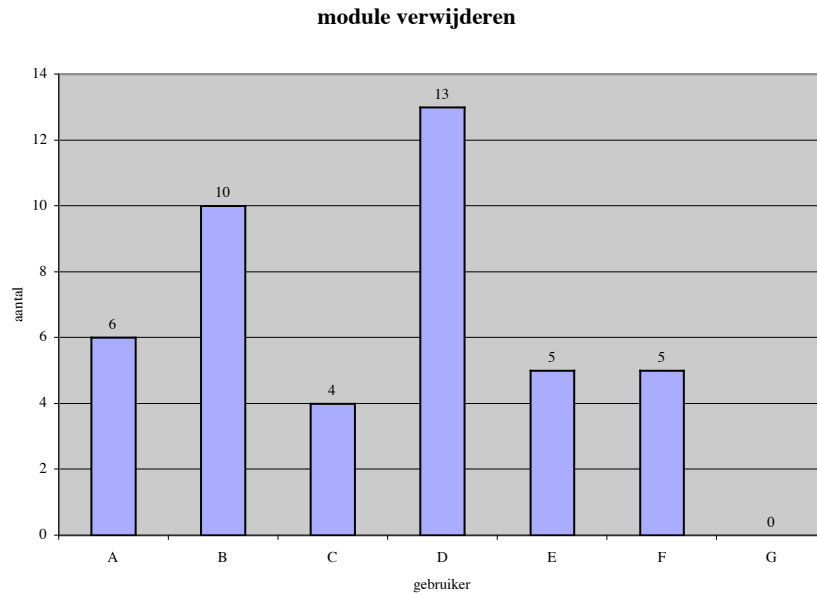


*figuur 6.24: onduidelijkheid per gebruiker*

Aan de hand van deze figuur is een duidelijk onderscheid te maken tussen de ervaren en minder ervaren gebruikers. Zo is opvallend dat de drie meest ervaren gebruikers niet één keer het overzicht kwijt zijn geraakt, terwijl de twee minst ervaren gebruikers F en G het vaakst het overzicht kwijt zijn. De oorzaak van een resultaat is bij twee van de drie ervaren gebruikers maar eenmaal niet duidelijk, bij de derde zelfs nooit, terwijl het bij de overige gebruikers veel vaker het geval is. Ook de vraag waarom de applicatie niet doet wat zij ervan verwachtten wordt door de drie ervaren gebruikers nooit gesteld, terwijl de andere gebruikers zich dat ieder minstens drie maal per sessie afvragen.

Samenvattend kan geconcludeerd worden dat de ervaren gebruikers duidelijk zicht hebben op wat er gaande is, hoe de patches zijn opgebouwd en waardoor dingen veroorzaakt worden. De minder

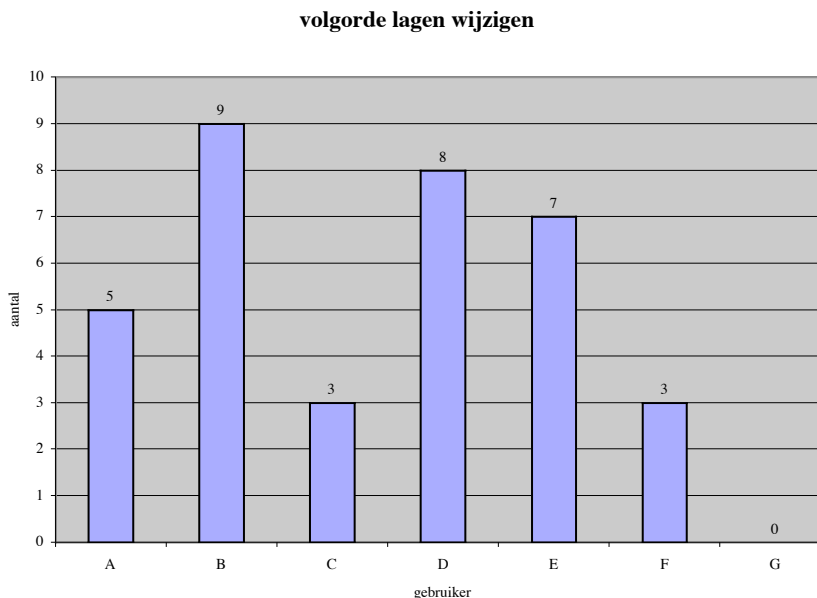
ervaren gebruikers hebben veel minder overzicht en vragen zich allemaal wel eens af waarom dingen gaan zoals ze gaan.



figuur 6.25: overzicht module verwijderen voor alle gebruikers

In figuur 6.25 is te zien hoe vaak iedereen de actie ‘module verwijderen’ heeft uitgevoerd. Hieraan is te zien dat de twee meest ervaren gebruikers B en D het vaakst modules hebben verwijderd. De derde ervaren gebruiker C valt echter uit de toon. Wel is opvallend dat de minst ervaren gebruiker G helemaal geen modules heeft weggegooid, wat waarschijnlijk veroorzaakt werd door een gebrek aan inzicht.

Figuur 6.26 geeft weer hoe vaak de volgorde van de outputlagen is gewijzigd. Hier valt niet veel uit af te leiden.



figuur 6.26: overzicht volgorde outputlagen wijzigen voor alle gebruikers

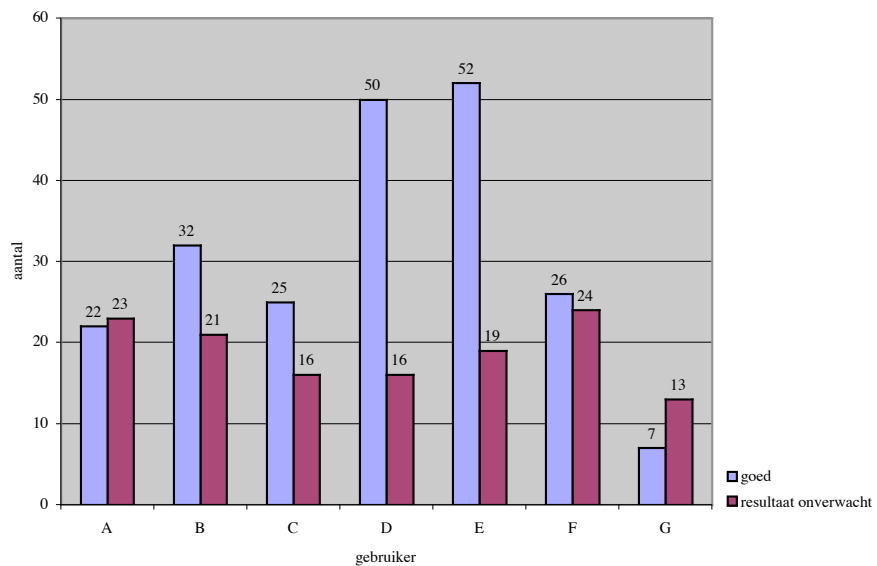
De actie ‘resultaat bekijken’ levert een interessante analyse. Uit de video’s valt op te maken dat een gebruiker een actie kan uitvoeren (module wijzigen, verbinden, verwijderen, volgorde lagen

wijzigen) waarvan bij het bekijken van de output in de realizer pas blijkt is dat het resultaat niet is wat men ervan verwachtte. Het komt ook voor dat een andere gebruiker een actie uitvoert, waardoor de output onverwacht veranderd. Op de video's valt ook te zien dat de realizer veel gebruikt wordt als communicatie-kanaal door er simpelweg in te typen. Hierdoor kunnen gesprekken tussen gebruikers ontstaan, waarin mensen rechtstreeks op elkaar reageren. Het komt echter ook voor dat een gebruiker een boodschap van een ander helemaal niet opmerkt en er dus ook niet op reageert.

Het is interessant om te bekijken hoe vaak de actie goed wordt uitgevoerd en hoe vaak het waargenomen resultaat inderdaad anders was dan verwacht. In figuur 6.27 worden deze beide mogelijkheden in één figuur getoond. In deze figuur is te zien dat het verschil tussen het goed uitvoeren van deze actie en het onverwachte resultaat bij de meest ervaren gebruiker D het grootst is. Ook bij de ervaren gebruikers B en C is dit verschil duidelijk aanwezig. Bij de minder ervaren gebruikers A, F en G is dit verschil nauwelijks aanwezig, of zelfs negatief. Een uitzondering vormt gebruiker E, die ondanks mindere ervaring niet zo vaak verrast wordt door het resultaat.

Een ander interessant fenomeen van deze actie is het typen in de realizer. Hieruit blijkt namelijk behoefte aan directe communicatie met de medegebruikers. In één van de sessies is hier uiteindelijk geen gebruik van gemaakt (wel mee aangevangen, maar niet doorgezet), namelijk de sessie met gebruikers E en G. In het geval van gebruiker G kwam dit omdat niet duidelijk was hoe dit voor elkaar te krijgen. De behoefte aan directe communicatie was wel degelijk aanwezig, maar kon niet worden bevredigd en werd toen opgegeven. Ook gebruiker E heeft het geprobeerd, maar na een aantal pogingen ook gestaakt. Figuur 6.28 toont het overzicht van alle gebruikers, hoe vaak getypt werd en daarnaast hoe vaak werd getypt als directe reactie op anderen, er dus als het ware een dialoog gaande was.

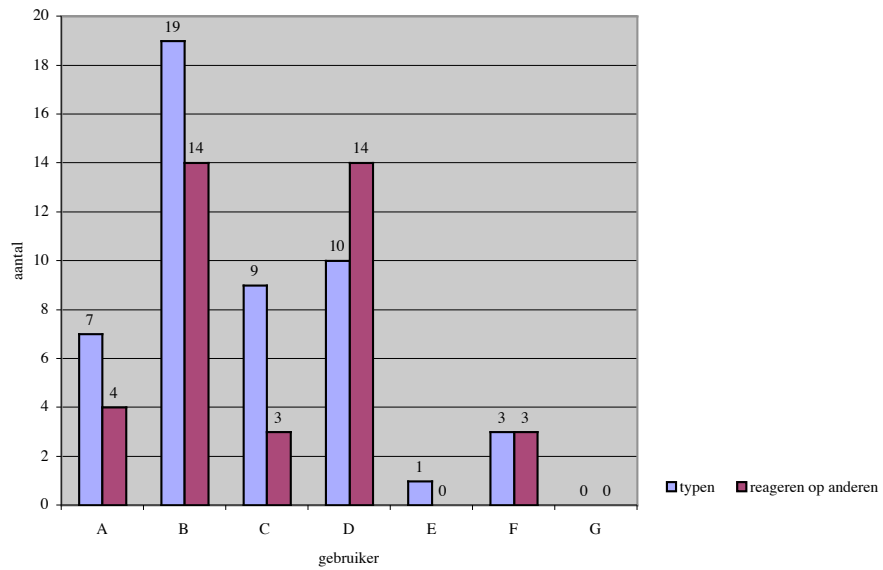
**resultaat bekijken - vergelijking**



*figuur 6.27: vergelijking resultaat bekijken per gebruiker*



### typen in realizer

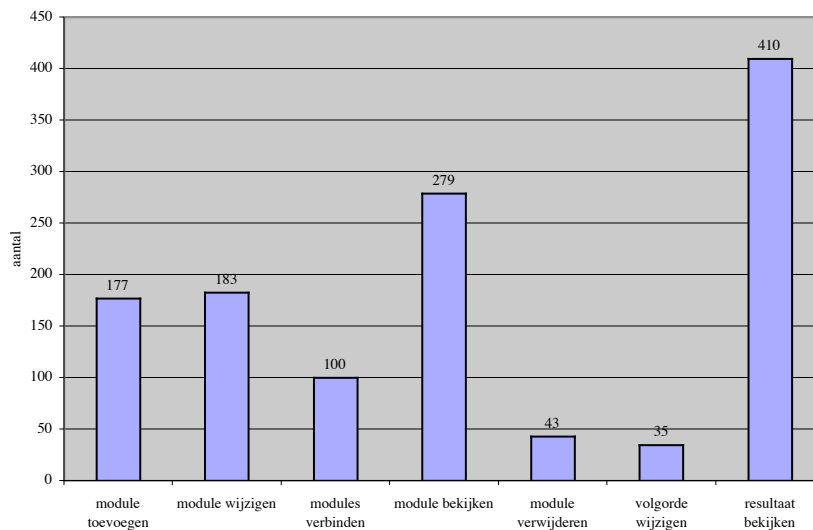


figuur 6.28: overzicht typen in realizer door alle gebruikers

In principe willen alle gebruikers gebruik maken van het typen in de realizer als extra communicatie-middel. Er wordt geen gebruik gemaakt van het aparte, daarvoor bestemde chatvenster. Waarschijnlijk weten de minder ervaren gebruikers niet eens dat die optie aanwezig is. Het blijkt dat de twee meest ervaren gebruikers B en D, die samen in één sessie zaten, het meest typten in de realizer, waarbij ook echt een dialoog gaande was. Ook de andere, minder ervaren gebruikers A, C en F, die samen in één sessie zaten, maakten gebruik van het typen, waarbij de meer ervaren gebruikers A en C dat dan weer vaker deden dan de minder ervaren gebruiker F. Zoals te zien heeft gebruiker E het wel geprobeerd, maar het uiteindelijk opgegeven. Om zichtbaar in de realizer te kunnen typen moeten namelijk een aantal acties uitgevoerd worden. Er moet eerst een keyboard worden toegevoegd. Vervolgens moet er een achtergrondkleur (BGColor) aangemaakt worden waarvan de outputlaag dan onder de keyboards geplaatst moet worden. Alle onderzochte gebruikers vangen hier direct mee aan in een sessie, waaruit geconcludeerd kan worden dat dit belangrijk is.

Ten slotte volgt in figuur 6.29 nog een overzicht van alle acties van alle gebruikers bij elkaar, waarin dus te zien is hoe vaak elke actie is uitgevoerd. Opvallend is de grote uitschieter die de actie 'resultaat bekijken' maakt. Hierna wordt de actie 'module bekijken' het meest uitgevoerd. Dit komt doordat gebruikers continu bezig zijn met dingen wijzigen, vervolgens het resultaat meteen bekijken en weer wijzigen, naast achterhalen hoe alles in elkaar steekt en wat er allemaal gaande is. Men is veel bezig met afstellen als het ware, het verfijnen van de output en beter inzicht verkrijgen in wat er gaande is in het systeem en elkaar. Tenslotte valt te zien dat twee ingrijpende acties 'module verwijderen' en 'volgorde wijzigen' het minst vaak worden uitgevoerd.

totaal alle acties



figuur 6.29: overzicht alle acties voor alle gebruikers

### 6.5.9 Verdere analyse gebruikersonderzoek – video's

Naast de meetbare analyse van de video's, volgt hier een bespreking van enkele opvallende algemene waarnemingen.

Er is een duidelijk verschil te bemerken tussen de verschillende gebruikers. De onervaren gebruikers zijn erg onzeker en kunnen niet goed, soms zelfs totaal niet, volgen wat er gebeurt. Zij zijn snel in de war wanneer iets niet gaat zoals zij hadden verwacht of wanneer grote veranderingen door anderen worden aangebracht. Een groot deel van de tijd zijn zij bezig met uitzoeken hoe iets werkt, hoe een patch in elkaar zit, wat met wat verbonden is en wat welk effect veroorzaakt. Wanneer zij iets proberen en het lukt na een aantal pogingen nog niet geven zij maar op. Er wordt bovendien lang naar de gewenste module gezocht door overal op te klikken. Wanneer een andere gebruiker iets in de realizer typt, duurt het soms lang voordat zij dit doorhebben en eventueel reageren.

De meer ervaren gebruiker doen alles veel doelbewuster, zelfverzekerder. Zij zoeken niet veel naar modules, maar gaan met de muis vrijwel direct naar de juiste plek in de interface. Zij durven beter eigenschappen te veranderen en zijn ook hierbij niet op zoek naar de juiste eigenschap, maar gaan daar direct op af, zij weten wat hun doel is. Wat opvalt is dat de ervaren gebruikers vaker modules weggooien, al dan niet ongebruikte, al dan niet door henzelf aangemaakt. Wat bovendien opvalt bij de twee meest ervaren gebruikers, B en D, is dat zij simpelweg modules weggooien en opnieuw beginnen wanneer iets niet meteen lukt. Zij gaan niet lang op zoek naar waarom iets niet werkt, zij proberen het gewoon nog eens, meestal met succes.

De ervaren gebruikers hoeven bijna nooit na te gaan hoe een patch in elkaar zit door van module naar verbonden module te klikken. Zij nemen ook veel vaker de tijd om even naar het resultaat in de realizer te kijken en verder niets te doen. De minder ervaren gebruikers komen hier helemaal niet aan toe, hebben hier de rust niet voor. Bovendien kunnen de ervaren gebruikers meer anticiperen op de andere deelnemers aan de sessie, zij zijn in staat in te schatten wat iemands bedoeling is. Wanneer het iemand niet lukt zijn doel te bereiken proberen zij te helpen door bijvoorbeeld benodigde verbindingen te maken. Op de video's is bijvoorbeeld te zien dat gebruiker C tijdens de sessie uit gaat leggen wat gebruiker F aan het proberen is.

De mogelijkheid om te typen in de realizer wordt door alle gebruikers gebruikt, of in ieder geval wordt het door iedereen geprobeerd. Deze communicatie wordt niet zonder reden gebruikt, het wordt voornamelijk benut om te informeren bij de andere gebruikers of zij hetzelfde resultaat waarnemen, of wijzigingen aankomen, of geluiden gehoord worden, om correcte werking van het

systeem na te gaan. Het wordt bijvoorbeeld niet veel gebruikt om individuele herkenning te krijgen, zoals “kijk, dat heb ik gemaakt, mooi he!” Het typen wordt ook niet gebruikt om te vragen hoe iemand iets heeft gedaan, bijvoorbeeld een bepaald effect voor elkaar heeft gekregen. Dit zou misschien wel in de verwachting liggen, maar blijktbaar is dat niet het kanaal voor dat soort boodschappen of vragen.

Tijdens twee sessies (niet in de sessie met gebruikers B en D aangezien zij slechts met zijn tweeën waren) is aan de gebruikers gevraagd of zij konden aangeven wie wat had veroorzaakt in de output, dus van welke gebruikers bepaalde media of effecten kwamen. Het is opvallend dat niemand precies kon zeggen wat door wie is gedaan. Sommige gebruikers twijfelen zelfs over hun eigen bijdrage. Er wordt enigszins gegokt op basis van kennis over de mede-gebruikers, zoals ervaring en sdergelijke, maar bij de sessie met gebruikers A, C en F kan men zelfs niet aanwijzen welke tekst door wie getypt wordt. Men weet dus niet wie wat doet.

### **6.6 Analyse gebruikersonderzoek – vragen achteraf**

De vragen die na de sessies aan de gebruikers zijn gesteld (zie bijlage 2), gaan dieper in op het multi-user aspect van een Keyworx-sessie en informeren op een directe wijze bij de gebruikers naar hun behoefte aan visualisatie van inzichtsinformatie. Alle antwoorden op de vragen zijn in het kort in een tabel uitgezet, die is opgenomen als bijlage 3. Hieronder worden de antwoorden uitgebreider besproken.

Vraag 2 of het eerder met elkaar samenwerken van invloed is op het verloop van deze sessie beantwoorden bijna alle gebruikers positief, alleen gebruiker E wijkt af. Het idee bestaat dus dat er een soort gedeelde ervaring opgedaan wordt waardoor de onderlinge samenwerking steeds beter verloopt. Uit vraag 5 blijkt bovendien dat het verschil in ervaring volgens sommige gebruikers (A, C, F en G) wel van invloed is op de sessie, maar dat dit niet perse negatief is. Het wordt veelal positief gevonden; het zorgt voor verassingen of leermogelijkheden. Eén gebruiker (F) vond het wel negatief, het wordt dan namelijk moeilijker alles te kunnen volgen.

Uit vraag 7 blijkt dat maar twee gebruikers (C en D) volledig konden volgen wat er gaande was, de rest kon het niet altijd volgen. Uit het tweede deel van de vraag blijkt echter wel dat de gebruikers zich niet buitengesloten voelden, er is dus wel een gevoel van onderlinge saamhorigheid.

Uit de vrijwel uitsluitend positieve antwoorden op vraag 8 en 9 valt af te leiden dat gebruikers van Keyworx gewoontes opbouwen. Dezelfde handelingen worden vaak uitgevoerd en er is sprake van voorkeur voor bepaalde media.

Vrijwel alle gebruikers weten waar de kleuren die gebruikt worden in de patcher om verschillende media-typen aan te geven, voor staan. De enige gebruiker die negatief antwoord, gebruiker A, geeft bovendien aan er altijd via trial and error wel weer achter te komen.

Vraag 11 en 12 tonen dat de meeste gebruikers het positief vinden dat iedereen elkaars media kan gebruiken en veranderen. Opvallend is overigens dat gebruiker B neutraal is wanneer anderen zijn dingen kunnen veranderen, maar wel erg positief wanneer dingen van anderen verandert kunnen worden! Maar al met al kan hieruit opgemaakt worden dat de vrije omgang en het multi-user aspect worden gewaardeerd.

Uit vraag 13 blijkt dat vrijwel alle gebruikers de mogelijkheid zouden willen hebben om even apart aan iets te werken, om iets even te ‘locken’, voor even of zolang diegene ermee bezig is.

Alle gebruikers beantwoorden vraag 14 negatief. Men is van mening dat wanneer toestemming moet worden gevraagd om een module te wijzigen, de boel dichtslibt, er nog meer data over en weer gestuurd moet worden, waardoor het nog ingewikkelder wordt.

Dat alle gebruikers behoefte hebben aan aanvullende, directe communicatie met de andere gebruikers wordt duidelijk uit de vrijwel alleen maar positieve antwoorden op vraag 15. De enige gebruiker die ‘nee’ antwoorde, gebruiker E, gaf aan het deze keer niet te willen omdat er sprake was van een jam. Uit vraag 16 blijkt bovendien dat de meeste gebruikers vaak ook gebruik maken van extra communicatiemiddelen, zoals Ichat of ICQ.

In vraag 17 tot en met 25 wordt direct aan gebruikers gevraagd welke informatie zij gevisualiseerd zouden willen zien. Uit de antwoorden valt op te maken dat er wel behoefte is aan visualisatie, maar vooral niet te veel of afleidend, aangezien er al zoveel te zien valt. Op één na alle gebruikers willen

zien met wie ze werken. Zij zouden ook willen zien waar iemand vandaan komt, niet zozeer omdat het van invloed is op het groepsproces, maar omdat men het wel interessant vindt om te zien waar mensen vandaan komen.

Op de vraag of men wil zien hoe actief de andere gebruikers zijn, antwoorden maar twee gebruikers volmondig ja. Eén gebruiker antwoordt misschien. Van drie gebruikers hoeft dit niet, dat blijkt wel uit de output. Eén gebruiker zegt dat het misschien wel interessant is om te visualiseren, maar dat de activiteit zelf niets hoeft te betekenen; je kan immers veel veranderen in de output door weinig handelingen uit te voeren of juist heel veel proberen maar niet succesvol zijn.

Vraag 20 en 21 lijken erg op elkaar. Vraag 20 vraagt of men zou willen zien wat de anderen doen, vraag 21 of men zou willen zien wat de huidige activiteit van anderen is. Het verschil hierbij is dat vraag 20 gaat over de algemenere activiteit van de gebruikers tijdens de gehele sessie, terwijl vraag 21 specifiek gaat over de huidige activiteit. Voor sommige gebruikers is het onderscheid tussen deze twee vragen echter niet duidelijk. Daarom antwoordt gebruiker A op vraag 21 dat je dat toch al ziet wanneer je vraag 17 en vraag 20 ziet (met wie je werkt en wat zij doen). Maar uiteindelijk antwoorden vrijwel alle gebruikers ja op deze twee vragen. Alleen gebruikers C denkt dat dat verwarrend zal zijn. Gebruiker D antwoordt op vraag 20 dat aan de output merkbaar zou moeten zijn wat de anderen doen.

Op de vragen 22, 23 en 24, zou je willen zien wanneer media wordt toegevoegd/verwijderd, verbindingen worden gemaakt, parameters worden gewijzigd, antwoorden de meeste gebruikers positief. Overigens dus wel met de vermelding dat het niet te druk of nadrukkelijk moet worden. Twee ervaren gebruikers, C en D, beantwoorden al deze vragen negatief, zij merken het wel aan de output en zijn bang voor een overdaad aan informatie. Hieruit blijkt duidelijk verschil met de minder ervaren gebruikers; deze laatste kunnen dit soort gegevens niet aan de output aflezen, dus hebben daarom wel behoefte aan een extra notificatie.

De meeste gebruikers zouden zichzelf niet duidelijker gerepresenteerd willen zien. Het gaat immers niet om individuen, maar om de groep. Gebruiker G daarentegen vindt juist dat het auteurschap niet zo anoniem moet zijn en vindt dit daarom wel wenselijk. Vrijwel iedereen vindt het een slecht idee om elke gebruiker een eigen stuk scherm toe te wijzen.

Bijna alle gebruikers zouden weer te geven informatie willen filteren door gradaties aan te kunnen geven zodat soms meer, soms minder informatie getoond wordt..

Op de vraag of men wel eens behoefte heeft aan een overzicht, een soort 'history' van wat er tot nu toe is gebeurd, antwoorden vijf van de zeven gebruikers ja. De 'history' zou men willen zien voor objecten, handelingen, modules, niet zozeer voor personen. Het is opvallend dat gebruiker F die nee antwoordt, iemand is die tijdens de sessie vaak het overzicht kwijt was. De andere nee-antwoorder, gebruiker E, vindt dat alles veel te snel gaat, dus dat een history te laat zou zijn.

Op de vraag of men nog ideeën heeft over hoe visualisatie van informatie gedaan zou kunnen worden, antwoorden veel mensen met ja. Benadrukt wordt dat de opbouw van patches, de verbindingen tussen modules, erg belangrijk zijn en dat men die wil zien. Het gaat er niet zozeer om wie wat doet, maar meer wat er gebeurt, waar verbindingen worden gemaakt, parameters worden gewijzigd, modules worden toegevoegd,

Tenslotte beoordelen de gebruikers de gehouden sessies voornamelijk als neutraal. Dit vanwege het veelvoorkomende vastlopen, de korte duur en de doelloosheid.

## **6.7 Keyworx volgens de theorie**

Uit de gebruikerstests valt ook af te leiden in welke mate met aspecten die in de literatuur genoemd zijn van toepassing zijn op Keyworx. Tijdens een sessie is sprake van synchrone, gedistribueerde interactie (besproken in paragraaf 3.1), namelijk op dezelfde tijd, vanaf verschillende lokaties, of in ieder geval vanaf verschillende computers, wat ook als zodanig op te vatten valt. In hoofdstuk 3 worden enkele problematische factoren van multi-user applicaties genoemd. Hieronder wordt besproken hoe deze zich in de context van Keyworx verhouden.

- sociale factoren: er wordt geen gebruik gemaakt van technologische protocollen. Keyworx is gericht op het 'vrijlaten' van de gebruikers en gaat ervan uit dat sociale conventies in acht worden genomen. Uit de videoanalyse blijkt dat dit inderdaad zo is; gebruikers helpen

elkaar graag, proberen anderen niet voor de voeten te lopen. Er is geen sprake van hiërarchische structuren of iets dergelijks, alle deelnemers zijn gelijk en kunnen dezelfde acties uitvoeren.

- Conflictbeheer: de server handelt de commando's sequentieel af. De verschillende gebruikers voeren bewerkingen lokaal uit. Op het moment dat enter wordt ingedrukt of de muis wordt losgelaten, wordt een bericht naar de server gestuurd. Hierdoor worden acties nooit tegelijkertijd uitgevoerd; zij worden op volgorde van binnenkomst uitgevoerd. Wanneer een gebruiker met een module werkt, blijft deze ook beschikbaar voor de andere gebruikers, er worden geen modules afgeschermd van anderen of iets dergelijks, dat gaat tegen het hele principe van Keyworx in, alles is immers van iedereen. Hierbij moet in acht worden genomen dat er een belangrijk verschil bestaat tussen Keyworx en bijvoorbeeld een multi-user tekstverwerkingsapplicatie. Bij het gezamenlijk bewerken van een tekst, kan het zo zijn dat door tegelijkertijd aan dezelfde zin te werken, er een totaal nietszeggende, onzinnige zin ontstaat, wat natuurlijk niet wenselijk is. Hiervoor moet dan aan conflictbeheer worden gedaan, om dit soort situaties te vermijden. Bij Keyworx echter bestaat er niets onzinnigs. Alles kan met alles verbonden worden, alles kan door elkaar gedaan worden. Op het moment dat een gebruiker aan een module iets wijzigt en ondertussen een andere gebruiker aan die module iets totaal anders toevoegt, dan wordt dit gewoon uitgevoerd, niets is immers fout of onwenselijk. Het kan zijn dat er een afschuwelijk kleurenpatroon ontstaat, of een hinderlijke piep, maar daar is niets fout aan. Bovendien kan dat zo verholpen worden. De kwestie conflictbeheer speelt dus in het geval Keyworx geen grote rol.
- Interface: Er is sprake van een WYSIWIS-interface. De interface bestaat uit twee delen, de realizer, waarin enkel de output te zien is, en de patcher, waarin het eigenlijke werk wordt gedaan. Door middel van kleuren is enig verschil aangegeven tussen verschillende gebruikers, maar dit wordt beperkt tot de live-inputs die de kleur van de gebruiker die ze heeft toegevoegd aannemen. Dit aangezien geen enkele module één persoon toebehoort, maar alles door iedereen gebruikt kan worden.

Samenvattend blijkt dat de algemene multi-user problemen als sociale factoren en conflictbeheer bij Keyworx geen rol spelen. Er wordt op deze punten geen bewust beleid gevoerd, het gaat vanzelf goed. Het grote probleem met Keyworx is de onoverzichtelijkheid van de patches en dat benodigde informatie niet af te leiden is uit de interface. Dit is echter een klassiek interface ontwerp probleem: hoe worden aanwezige objecten en de mogelijke bewerkingen daarop inzichtelijk weergegeven? De onoverzichtelijke interface wordt niet veroorzaakt door het multi-user aspect, het wordt er slechts lastiger door.

## **6.8 Conclusie**

Dit hoofdstuk presenteert de mogelijkheden van de applicatie Keyworx en hoe gebruikers daarmee omgaan. Er is een gebruikersonderzoek uitgevoerd met gebruikers van Keyworx. Dit leverde data op in de vorm van video's en ingevulde vragenlijsten. Naar aanleiding van een taakanalyse is bij het bekijken van de video's geteld hoe vaak de mogelijke acties uitgevoerd worden. Bovendien blijkt uit de video's tegen welke problemen de gebruikers bij het uitvoeren van die acties aanlopen. Er is ook bijgehouden hoe vaak deze problemen bij elke gebruiker voorkomen. Hieruit is af te leiden dat onderscheid gemaakt kan worden tussen ervaren en minder ervaren gebruikers. Deze laatste hebben moeite met het bewaren van het overzicht. Zij bekijken modules vaak om te kijken hoe verbindingen lopen, waardoor bepaalde effecten worden veroorzaakt en waarom de applicatie niet doet wat zij willen. De ervaren gebruikers hebben deze problemen vrijwel nooit. Ook wanneer het resultaat wordt bekeken, is onderscheid te maken tussen de ervaren en minder ervaren gebruikers. Voor de laatsten is het resultaat vaak anders dan verwacht, wat zowel door eigen acties als door acties van anderen veroorzaakt kan worden. Typen in de realizer gebeurt vaker door ervaren gebruikers, waarbij dan meer sprake is van een reactie op elkaar, een soort conversatie.

Samenvattend valt te concluderen dat ervaren gebruikers zelfzekerder zijn, doelgerichter handelen en minder twijfelen aan het hoe en waarom van het ontstane resultaat.

Uit de analyse van de ingevulde vragenlijsten blijkt dat de meeste gebruikers behoefte hebben aan visualisatie van extra informatie, echter niet te druk en afleidend aangezien er al genoeg te zien is. Er is vooral behoefte aan overzicht van de patches, hoe de verbindingen lopen, waar dingen gewijzigd worden. Men wil informatie over de modules die gebruikt worden en de acties die uitgevoerd worden, niet zozeer over de gebruikers die dat doen.

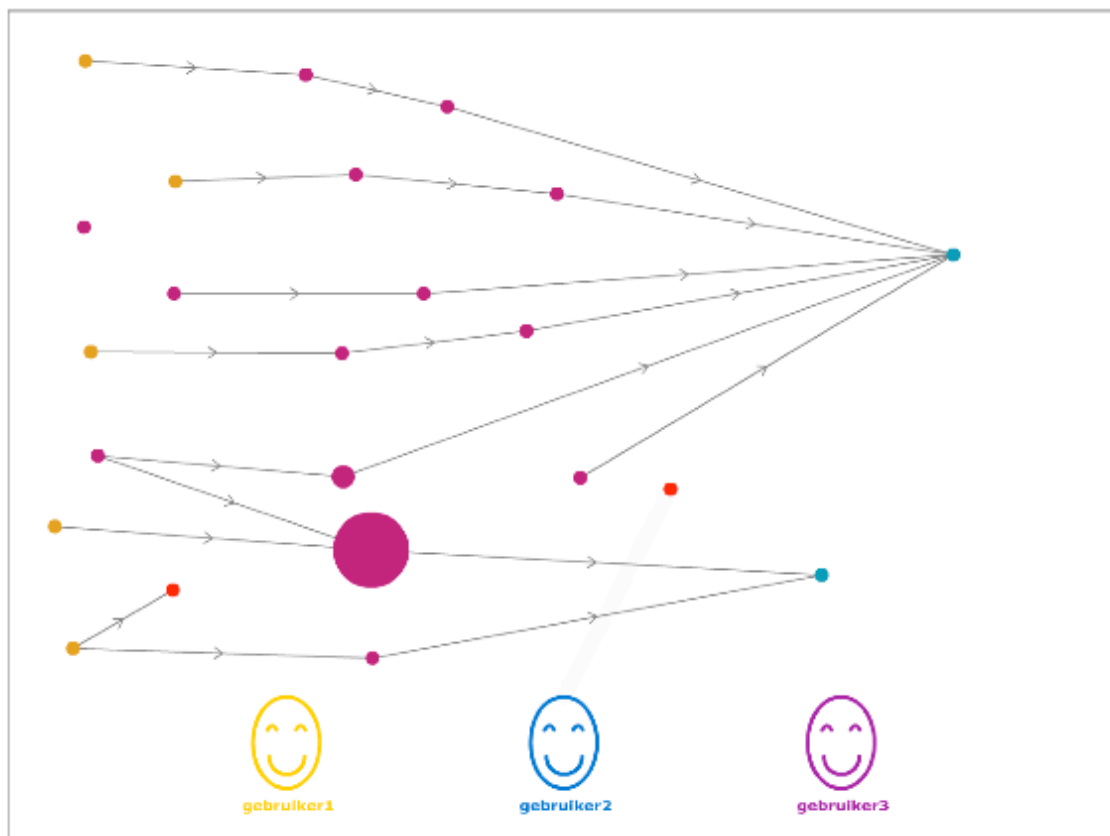
Tenslotte kan de applicatie worden getoets aan problematische factoren van multi-user applicaties. Er wordt geen gebruik gemaakt van technologische protocollen, de gebruikers worden vrijgelaten zelf sociale protocollen te vormen. Conflicterende acties komen wegens de aard van de applicatie en de sequentele afhandeling van acties door de server niet voor. De interface heeft te kampen met de klassieke interface ontwerp problemen, namelijk onoverzichtelijkheid. Dit wordt echter niet door het multi-user aspect veroorzaakt.

## 7 Prototype

### 7.1 Ontwerp

Naar aanleiding van de analyse van het gebruikersonderzoek (zowel de video's als de vragenlijsten) is een ontwerp gemaakt voor een prototype. Uit het gebruikersonderzoek blijkt dat men de meeste behoefte heeft aan een overzicht van de opbouw van de patches, hoe zijn de verschillende modules aan elkaar verbonden? Men wil graag zien wanneer modules worden toegevoegd of verwijderd, verbindingen worden aangebracht en attributen worden gewijzigd. Het is interessant om te zien wie de aanwezige gebruikers zijn. Om deze niet dwars te zitten, is het handig aan te geven aan welke module of patch gebruikers op dat moment werken, zodat bekend is dat daar iets mee wordt gedaan zodat ze bijvoorbeeld niet per ongeluk weggegooid worden. Hoe actief een gebruiker is, is niet zo belangrijk aangezien er geen direct verband bestaat tussen de activiteit in de patcher en het effect daarvan op het resultaat; het uitvoeren van weinige handelingen in de patcher kan de output in de realizer in grote mate beïnvloeden. Deze voornaamste wensen hebben geleid tot het prototype dat hieronder besproken wordt\*.

In figuur 7.1 is de gehele visualisatie in het prototype te zien. De verschillende aspecten worden vervolgens apart besproken.



figuur 7.1: visualisatie in het prototype

De gekleurde rondjes geven de verschillende modules weer. De kleur is afhankelijk van de soort module, waarbij de kleuren die gebruikt worden in de patcher grotendeels zijn aangehouden. De inputs, zowel de live inputs als de library files, zijn geel, de generators zijn paars, de modifiers zijn rood en de renderers zijn blauw. Er is gekozen om de generators te onderscheiden van de overige inputs, aangezien deze duidelijk een andere taak vervullen.

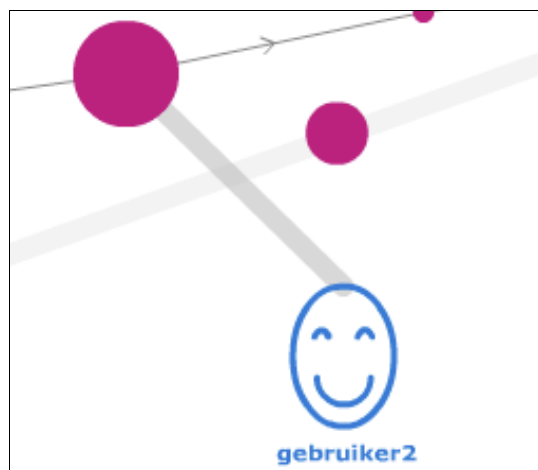
\* Het prototype is ook op internet te bekijken: <http://home.student.uva.nl/willeminj.vanpoppel/prototype.html>

Door met de muis over een module te gaan, verschijnt informatie over de betreffende module, namelijk de naam, het type en de gebruiker die hem heeft toegevoegd (zie figuur 7.2). Dit wordt bewust pas bij een 'mouse over' getoond omdat het venster anders te vol zou staan.



figuur 7.2: informatie over de module die bij een 'mouse over' verschijnt

De grootte van de module wordt bepaald door de hoeveelheid activiteit; wanneer een eigenschap wordt gewijzigd of een verbinding wordt gemaakt, wordt de module groter (zie figuur 7.3). Wanneer er niets mee gebeurt, wordt deze na verloop van tijd vanzelf weer kleiner.



figuur 7.3: gebruiker bewerkt module die vervolgens groter wordt

De verbindingen tussen modules worden weergegeven door lijnen (zoals te zien in figuur 7.1). Om ervoor te zorgen dat het geen warboel van lijnen wordt, kunnen de modules door de gebruiker verplaatst worden, zodat zelf een handige indeling gemaakt kan worden. De verbindingen zijn niet alleen zichtbaar, ook is een richting aangegeven door middel van pijlen waardoor duidelijk wordt wat een input en wat een output is.

De tot patches met elkaar verbonden modules worden van links naar rechts geplaatst, waarbij inputs links verschijnen, modifiers in het midden en renderers rechts. De verbindingen lopen dus horizontaal over het venster. Er is bewust voor gekozen de patches niet verticaal te plaatsen om te voorkomen dat dan het idee ontstaat dat de plek van de patches ook iets zegt over een eventuele bijbehorende gebruiker (dus patches aan de linkerkant horen bij gebruiker links) (zie paragraaf 4.1.2, over het ontstaan van onjuiste, impliciete feiten).

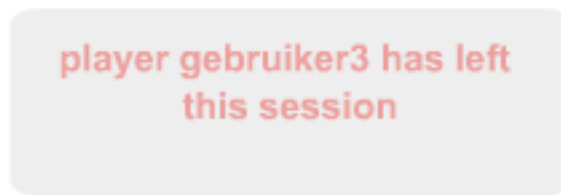
De aanwezige gebruikers verschijnen onderin het venster met hun inlognaam (zoals te zien in figuur 7.1). Wanneer een gebruiker een module toevoegt of een eigenschap wijzigt, verschijnt er een doorzichtige, vervagende streep van de gebruiker naar de betreffende module, waardoor te zien is welke gebruiker met welke module bezig is (zoals te zien in figuur 7.3). Door het verschijnen van deze strepen ontstaat ook inzicht in hoe actief elke gebruiker is. Bij het verbinden van modules wordt dit niet gedaan, om een overdaad aan strepen te voorkomen. Bovendien worden verbindingen vaak gemaakt door gebruikers die de module hebben toegevoegd of die iets eraan hebben gewijzigd, dus daaraan valt doorgaans ook af te leiden door wie de verbindingen worden gemaakt.



Verwijderde modules, verbindingen en gebruikers vervagen alvorens helemaal te verdwijnen, om het beeld niet te plotseling, schokkerig, te laten veranderen (zie figuur 7.4). Wanneer een gebruiker een sessie verlaat verschijnt bovendien een doorschijnend tekstblokje rechtsboven in het venster dat meldt dat gebruiker X de sessie heeft verlaten (zie figuur 7.5).



figuur 7.4: vervagende module en verbindingen



figuur 7.5: waarschuwing wanneer gebruiker sessie verlaat

## 7.2 Implementatie

De server van Keyworx legt alles wat er gebeurt met Keyworx vast in XML-logfiles. Voor de implementatie van het prototype is dan ook uitgegaan van deze data. Voor de visualisatie van de data is gekozen voor Flash.

De XML-logfiles zijn gefilterd op relevante informatie. Dit zijn de volgende acties: een gebruiker treedt toe tot sessie of verlaat deze, een module wordt toegevoegd of verwijderd, een verbinding wordt toegevoegd of verwijderd, een attribuut van een module wordt gewijzigd. De meldingen van overige acties zijn verwijderd, aangezien ze voor deze toepassing niet relevant zijn. Dit filteren is overigens van te voren gedaan en niet overgelaten aan Flash, aangezien de server-logs enorme bestanden zijn, die Flash niet aan zou kunnen (om een indruk te geven; de log van de eerste voor dit onderzoek gehouden gebruikerssessie beslaat 1915 pagina's, of 106.816 regels). In de bestanden wordt af en toe melding gemaakt van de tijd. Hierdoor is af te leiden op welke tijd de verschillende acties hebben plaatsgevonden. Deze tijd is vervolgens in de XML-data als extra attribuut aan de acties (de XML-nodes) meegegeven. Hieronder is een voorbeeld van een XML-node te zien, waarin een module wordt toegevoegd:

```
<patch uid="2" ><module Type="typeTextRender" id="14320002-Text" wrap="false" inputmode="text" name="Text1" t="7558" ></module></patch>
```

In Flash wordt de overgebleven, schone, XML-data eerst helemaal geparseerd. Aan de hand van de naam van de nodes wordt bepaald om welke Keyworx-actie het gaat (diegenen die hierboven zijn genoemd) en wat vervolgens gedaan moet worden. Er zijn verschillende objecten (prototypen in Flash, vergelijkbaar met klassen in Java) gedefiniëerd, namelijk 'players', 'modules', 'lijnen' en 'acties'. Aan de hand van de naam van de gelezen XML-node wordt bepaald welk van deze objecten moet worden aangemaakt. De objecten krijgen verschillende eigenschappen mee uit de XML-data, zoals naam, type, identificatie-nummer, identificatie-nummer van uitvoerende gebruiker en tijd van uitvoering. Lijnen krijgen bovendien mee tussen welke twee modules ze getrokken dienen te worden. In de classificatie van de objecten wordt een tekenfunctie gedefiniëerd die uiteindelijk de bijpassende figuren zal tekenen, met behulp van zogenaamde movieclips. De acties module toevoegen of verwijderen en attribuut wijzigen, resulteren bovendien ook in het aanmaken van een object 'actie', waarin te vinden is om welke module het gaat, welke gebruiker de actie heeft

uitgevoerd en op welk tijdstip. Wanneer de complete XML-data is geparseerd, wordt overgegaan op de eigenlijke visualisatie. Hierin wordt met behulp van een timer die telkens met 1 wordt opgehoogd, bekeken of de waarde van de timer overeenkomt met een tijdstip van één van de objecten. Is dat zo, dan wordt het betreffende object getekend. Bij een 'module', 'player' of 'lijn' resulteert dat in het verschijnen van het betreffende symbool. Bij een 'actie' betekent dat dat er een doorzichtige, vervagende streep wordt getrokken van de gebruiker die die actie uitvoert naar de module waar de actie betrekking op heeft. Wanneer het echter om een verwijder-actie gaat, vervaagt het betreffende object (zowel 'module', 'player' als 'lijn') eerst langzaam om vervolgens helemaal te verdwijnen. Wanneer een player verdwijnt, wordt een extra movieclip getoond, namelijk een tekstblokje die meldt dat een gebruiker de sessie heeft verlaten.

Aangezien Flash in één keer alle XML parst, dus zonder vertraging die voor tijdverloop zorgt, is het nodig om met een timer te werken. Zonder de timer zou alles in één keer uitgevoerd worden door Flash, waardoor alleen het eindresultaat van een sessie te zien zou zijn (de verwijderde modules zullen dan dus nooit getoond worden). Door met een timer te werken, wordt het tijdverloop van een sessie dus gesimuleerd.

### 7.3 Evaluatie

Het prototype is ter evaluatie op internet gezet, zodat dezelfde zeven gebruikers die hebben deelgenomen aan het Keyworx-gebruikersonderzoek, het konden evalueren. Hiervoor kregen zij de vragen die zijn opgenomen in bijlage 4 per email opgestuurd. Vier van de zeven proefpersonen hebben gereageerd, namelijk gebruikers A, D, E en F. Gebruiker D was de meest ervaren gebruiker, gebruiker A was een gemiddeld ervaren gebruiker, gebruikers E en F waren minder ervaren gebruikers. Hier volgt een bespreking van de antwoorden van deze gebruikers.

Op vraag 1, of het nuttig is de verbindingen tussen modules te zien, antwoorden alle gebruikers positief. Gebruiker E voegt toe dat ook de aangegeven richting erg nuttig is. Gebruiker A zou het handig vinden ook te zien welke attributen met elkaar verbonden. Zowel gebruiker A als D geven aan gradaties aan te willen geven in getoonde verbindingen, bijvoorbeeld alle verbindingen van de laatste 15 minuten of van één module.

De extra informatie over een module die getoond wordt bij een 'mouse over' wordt door gebruiker E voldoende gevonden. Hierbij wordt opgemerkt dat het nut van het tonen van de gebruiker die de module heeft toegevoegd, afhangt van de soort van de module; bij een generator is dit niet zo belangrijk, bij een live input echter wel. Gebruikers A en D vinden de getoonde informatie niet voldoende, er zou ook getoond moeten worden wat de waarden van de attributen zijn. Gebruiker D merkt op dat deze behoefte afhangt van de gebruiker en dat het nog beter zou zijn wanneer je zelf kunt instellen hoeveel informatie getoond wordt. Gebruiker F vindt de getoonde informatie voldoende, maar suggereert wel te tonen wat de output van een module is, bijvoorbeeld bij een keyboard wat er getypt is, of bij een geluidsinput een mogelijkheid tot het afspelen van het geluid.

Uit de antwoorden op vraag 3 blijkt dat het onderscheid tussen de modules door het gebruik van kleuren op zich duidelijk is, maar dat de plaatsing van de modules op het scherm willekeurig overkomt en dat dat overzichtelijker zou moeten gebeuren. Gebruiker D zou bovendien graag iconen zien zodat ook zichtbaar is wat voor soort generator het bijvoorbeeld het. Gebruiker A voegt toe dat het nuttig zou zijn om de modules qua plaatsing te relateren aan gebruikers, dus te plaatsen in de buurt van de gebruiker die die module heeft toegevoegd. Dit is erg opvallend, aangezien uit de eerdere gebruikersanalyse juist naar voren kwam dat men geen modules aan bepaalde gebruikers wil toewijzen en daarom in dit prototype bewust gekozen is om de modules niet ten opzichte van de gebruikers te plaatsen.

Vraag 4 wordt door iedereen positief beantwoord. Het is erg handig om de activiteit van een module te zien, hoewel het misschien ook door kleurverandering aangegeven zou kunnen worden.

Vraag 5 wordt ook positief beantwoord, men ziet graag de activiteit van een gebruiker, zodat men elkaar niet in de weg gaat lopen. Hierbij wordt overigens door de meeste gebruikers wel opgemerkt dat de doorzichtige, vervagende streep waarschijnlijk niet de optimale manier is om dit aan te geven.

Uit de antwoorden op vraag 6 blijkt dat de gebruikers tevens willen zien welke gebruiker een verbinding toevoegt, maar dat dit subtieler moet gebeuren dan het aangeven van de overige activiteiten, dus niet door middel van een doorzichtige, vervagende streep, aangezien het anders te druk wordt.

Het wordt door alle gebruikers noodzakelijk gevonden een extra melding te maken wanneer een gebruiker de sessie verlaat, het gaat tenslotte om een samenwerking, maar de wijze waarop dat nu gebeurt is in ieder geval niet duidelijk genoeg, het is te kort.

Uit de eerdere antwoorden blijkt dat er nog informatie gemist wordt in de visualisatie, met name de waardes van de attributen, de inputs en outputs van verbindingen en welke gebruiker verbindingen toevoegt. Gebruiker D suggereert dat er bij een 'mouse over' bij een verbinding eigenschappen getoond kunnen worden, zoals welke waardes er door de verbinding gaan.

Op vraag 9 wordt geantwoord dat de visualisatie van informatie erg nuttig is. Het zien van de verbindingen, de modules en de gebruikers en hun activiteiten is volgens de gebruikers de enige manier om te begrijpen wat er gaande is in een sessie. Er wordt vermeldt dat het huidige prototype nog wat rommelig is, maar dat het al een veel duidelijker beeld scheidt van wat gaande is dan de huidige Keyworx interface.

Tenslotte worden er nog wat suggesties ter verbetering gedaan. De voornaamste is dat het weergeven van de verbindingen snel rommelig wordt, ondanks de mogelijkheid de modules te verplaatsen. Hier moet zeker iets aan gedaan worden. Gebruiker A voegt nog toe dat de huidige weergave van de gebruikers wel leuk is, maar niet zo algemeen als noodzakelijk is. Of, wanneer er dan toch gebruik wordt gemaakt van een gezichtje, laat daaraan dan ook iets zien over de staat van de gebruiker, bijvoorbeeld verdrietig wanneer modules niet verschijnen. De activiteit van de gebruiker zou ook getoond kunnen worden, bijvoorbeeld zweetdruppels versus gesloten ogen als slapend.

#### **7.4 Conclusie**

Op het gemaakte prototype valt een en ander aan te merken. De plaatsing van modules zou efficiënter gedaan kunnen worden. Op dit moment worden ze afhankelijk van het type in een bepaald gebied geplaatst, maar daarbinnen nog willekeurig. Wanneer vervolgens verbindingen worden toegevoegd, kan een warboel ontstaan, ondanks het feit dat de gebruikers de modules kunnen verplaatsen. Automatische opbouw van veranderlijke diagrammen is echter een zeer lastig probleem dat niet eenvoudig op te lossen is.

In het huidige prototype verschijnt een doorschijnende, vervagende streep van de gebruiker die een bewerking uitvoert naar de module waarop die bewerking plaatsvindt. Dit is wel heel duidelijk, maar het zou subtieler kunnen. Op dit moment wordt er wel duidelijk aangegeven welke gebruiker een module toevoegt of verwijdert of een attribuut wijzigt, maar dit wordt bij het verbinden van twee modules niet gedaan. Wanneer er ook nog een doorschijnende streep vanuit de gebruiker naar een nieuwe lijn zou worden getrokken wordt het te druk en onoverzichtelijk. Toch is het wel wenselijk hier een notificatie van te geven op een rustige manier.

De informatie die bij een 'mouse over' bij een module verschijnt zou eventueel uitgebreid kunnen worden. Zo zou bijvoorbeeld de laatst ermee werkende gebruiker gemeldt kunnen worden of waardes van attributen. Het zou ook wenselijk zijn niet alleen aan te geven dat een attribuut van een module wordt gewijzigd, maar ook welk attribuut dat dan is. Dit wordt echter niet in de XML-data bijgehouden, dus is op dit moment niet te achterhalen.

Ten slotte gaven de gebruikers tijdens het gebruikersonderzoek aan het leuk te vinden de lokatie van de andere gebruikers te kunnen zien, om een 'globaal gevoel' te krijgen. Het zou dus een leuke, extra bijkomstigheid zijn wanneer dit ook op de een of andere manier af te leiden zou zijn uit de logs, zodat dit gevisualiseerd zou kunnen worden.

Een aantal door de gebruikers geuite wensen zijn niet opgenomen in het ontwerp. Zo werd aangegeven dat een overzicht, een soort 'history', wel gewaardeerd zou worden. Ook de mogelijkheid om gradaties aan te kunnen geven in getoonde informatie werd een goed idee gevonden. Het zou bijvoorbeeld handig kunnen zijn om enkel delen van de informatie te kunnen

zien, zoals één bepaalde patch of één bepaalde gebruiker. De reden dat dit in het prototype niet geïmplementeerd is, is de ingewikkeldheid hiervan.

Om de implementatie wat beter te laten verlopen zouden de XML-logfiles anders opgebouwd kunnen worden. Het zou handig zijn wanneer elke actie bijhoudt op welke tijd deze uitgevoerd wordt. Voor dit prototype is de tijd achteraf toegevoegd aan de XML-nodes, wat erg omslachtig is. Het type van de module bepaald zijn kleur en plek in het venster. Het is op dit moment echter vrij lastig te achterhalen wat het type van een module is. Het zou handig zijn wanneer dit in de XML op een duidelijke manier wordt bijgehouden.

Het voornaamste probleem van het gebruik van Flash is dat het niet real-time kan werken. Het is niet mogelijk real-time de XML-logfiles te laten uitbreiden en ze op dat moment in te lezen door Flash, die dan op dat moment de bijpassende actie uitvoert. Vandaar dus dat voor een uiteindelijke implementatie van dit prototype naar een andere manier gezocht zal moeten worden.

## 8 Conclusie

Het doel van dit project was te onderzoeken welke informatie over groepsactiviteit in de interface van een multi-user applicatie gevisualiseerd moet worden om beter inzicht te creëren bij de gebruikers van die applicatie. Hier is een algemeen onderzoek naar gedaan, om te achterhalen wat de literatuur zegt over multi-user applicaties en het creëren van inzicht, en er is een praktijkonderzoek uitgevoerd om deze theorie met de werkelijkheid te vergelijken.

Uit de literatuur blijkt dat bij het gebruik van multi-user systemen drie aspecten voor problemen kunnen zorgen. De eerste zijn sociale factoren; wat is de achtergrond van de gebruikers, hoe gaan de gebruikers met elkaar om en welke rollen vervullen zij. Het tweede aspect dat voor problemen kan zorgen is conflictbeheer, waar sprake van kan zijn wanneer twee of meerdere gebruikers conflicterende operaties willen uitvoeren. Het derde aspect dat problemen kan opleveren, is de interface. Multi-user interfaces moeten groepsactiviteit weergeven en kunnen door meerdere personen tegelijk gebruikt worden. Bovendien zorgen meerdere gebruikers voor meer activiteit en voor meer complexiteit. Een goede multi-user interface moet het gehele groepsproces weergeven zonder afleidend of storend te zijn.

Om inzichtsinformatie te kunnen presenteren aan de gebruikers van een multi-user applicatie moeten twee belangrijke stappen worden doorlopen. Eerst moet te presenteren informatie geselecteerd worden; welke informatie is relevant genoeg om weer te geven? Dit kan door de gebruikers zelf worden aangegeven, maar de voorkeur gaat uit naar een automatische selectie van de informatie. Na selectie moet overgegaan worden tot presentatie van de informatie. Om informatie te visualiseren op een wijze die eenvoudig door mensen te interpreteren is, is gebruik van diagrammatische representaties aan te raden. Het geniet de voorkeur de visualisatie van inzichtsinformatie plaats te laten vinden in de bestaande werkruimte en niet in een apart venster, om overdaad aan vensters te voorkomen.

Het uitgevoerde praktijkonderzoek richtte zich op de applicatie Keyworx, een multi-user DJ/VJ-tool. Er is een taakanalyse gedaan en een gebruikersonderzoek uitgevoerd. Hieruit blijkt dat gebruikers veel problemen tegenkomen bij het gebruik van Keyworx die veroorzaakt worden door onvoldoende informatievoorziening; het is niet duidelijk hoe de gemaakte patches zijn opgebouwd en waar bepaalde effecten in de output door veroorzaakt worden. Er is duidelijk verschil te merken tussen ervaren en minder ervaren gebruikers; de laatsten zijn vaak het overzicht kwijt, worden overrompeld door de ingewikkeldheid van de sessie en voelen zich onzeker. De meeste gebruikers typen ook berichten naar de anderen in de realizer, waaruit duidelijk een behoefte aan directe communicatie valt af te leiden. Uit de analyse van de antwoorden op de gestelde vragen blijkt dat de gebruikers wel degelijk behoefte hebben aan aanvullende visualisatie van informatie over wat er gaande is om beter overzicht te krijgen. Wat opvalt, is dat men niet zo geïnteresseerd is in de andere gebruikers, wat naar aanleiding van de literatuur wel verwacht werd, maar dat men simpelweg wil weten wat er gebeurt, welke modules met elkaar verbonden worden, welke attributen gewijzigd worden. Het zou bovendien op prijs worden gesteld wanneer de gebruikers de mogelijkheid zouden hebben gradaties aan te geven in hoeveelheid gevisualiseerde informatie. De gebruikers vinden het een goed idee een overzicht, een soort 'history', te tonen.

De drie eerder genoemde aspecten die vaak voor problemen zorgen bij multi-user applicaties, sociale factoren, conflictbeheer en groepsinterfaces, spelen bij Keyworx niet allemaal een rol. De sociale factoren worden overgelaten aan de gebruiker; er is geen onderscheid in rollen of mogelijkheden, en conflictbeheer wordt niet bewust toegepast omdat er door de aard van de applicatie en door de sequentiele afhandeling van acties door de server geen conflicten kunnen ontstaan. De groepsinterface zorgt wel voor problemen. Dit wordt echter veroorzaakt door een onoverzichtelijke weergave van de modules en acties, dat is een klassiek interface ontwerp probleem, dat niet zozeer te maken heeft met het multi-user aspect.

Naar aanleiding van de analyse van het gebruikersonderzoek is een prototype gemaakt dat een automatische visualisatie genereert om tegemoet te komen aan de wensen van de gebruikers van Keyworx. Uit de evaluatie van het prototype door de gebruikers blijkt dat hoewel er nog wel wat aspecten voor verbetering vatbaar zijn, het grotendeels voldoet aan de informatiebehoefte en het de inzichtelijkheid in de groepsactiviteit aanzienlijk vergroot.

### **8.1 Toekomstig onderzoek**

Het in dit onderzoek ontwikkelde prototype is een eerste opzet. Er zal verder onderzoek plaats moeten vinden naar de wensen van de gebruikers van Keyworx met betrekking tot informatievoorziening om de visualisatie nog beter daarop af te kunnen stemmen. Er zouden bovendien meer experimenten gedaan moeten worden met verschillende wijzen van visualisatie. Er zal ook gezocht moeten worden naar een wijze om de visualisatie real time te genereren en te integreren in de huidige Keyworx interface.

Verder zou het interessant zijn de applicatie Keyworx onder andere omstandigheden te testen en analyseren. Hoe zit het bijvoorbeeld in een live performance situatie? Heeft men dan behoefte aan andere informatievisualisatie? En wat als gebruikers elkaar niet kennen? Is dat van invloed op de informatiebehoefte? Is er dan nog meer behoefte aan aanvullende informatie?

Wat opvalt bij Keyworx is dat de gebruikers geen behoefte hebben aan inzicht in de andere gebruikers, maar in de modules en wat daarmee gebeurt. Het zou kunnen dat dit veroorzaakt wordt door de gelijke rolverdeling in Keyworx, iedereen kan hetzelfde zien en doen. Het zou interessant zijn onderzoek te doen naar multi-user applicaties waarbij dit niet het geval is, waar wel onderscheid te maken valt in rollen. Denk hierbij bijvoorbeeld aan multi-user tekstverwerkers, waarbij verschillende gebruikers verschillende taken uitvoeren. Heeft men daar misschien wel behoefte aan informatie over andere gebruikers?

## Bibliografie

- Ackerman, M.S. and Starr, B. (1995) - Social activity indicators: interface components for CSCW systems. In *Proceedings of the ACM symposium on user interface software and technology UIST'95*, ACM press, New York, p.159-168.
- Beaudouin-Lafon, M. and Karsenty, A. (1992) - Transparency and awareness in a real-time groupware system. In *Proceedings of the ACM symposium on user interface software and technology UIST'92*. ACM press, New York, p.171-180.
- Berlage, T. and Sohlenkamp, M. (1999) - Visualizing common artefacts to support awareness in computer-mediated cooperation. In *CSCW: the journal of collaborative computing* 8(3), p.207-238.
- Carpendale, M.S.T. and Montagnese, C. (2001) – A framework for unifying presentation space. In *Proceedings of the 14th annual ACM symposium on User interface software and Technology*, p.61-70.
- Domik, G. (1999) – Tutorial on visualization. Siggraph education materials. Te vinden op: <http://www.siggraph.org/education/materials/HyperVis/domik/folien.html>
- Dourish, P. and Belotti, V. (1992) - Awareness and coordination in shared workspaces. In *Proceedings of the conference on Computer supported cooperative work CSCW'92*. ACM press, New York, p. 107-114.
- Ellis, C.A., Gibbs, S.J. and Rein, G.L. (1991) - Groupware: some issues and experiences. In *Communications of the ACM* 34(1), p.38-58.
- Faulkner, C (1998) - The essence of human-computer interaction. Prentice Hall.
- Grudin, J. (1990) - The computer reaches out: the historical continuity of interface design. In *Proceedings of the conference on human factors in computing systems CHI'90*, ACM press, New York, p.261-268.
- Grudin, J. (1993) - Interface: an evolving concept. In *Communications of the ACM* 36(4), p.112-119.
- Grudin, J. (1994) - Groupware and cooperative work: problems and prospects. In *The art of human-computer interface design* - B. Laurel (eds.) Addison-Wesley, p.171 - 185.
- Gutwin, C. and Greenberg, S. (1996) - Workspace awareness for groupware. In *Proceedings of the conference on human factors in computing systems CHI'96*. ACM Press, New York, p.208-209.
- Harris, R. L. (1999) – Information Graphics: a comprehensive illustrated reference. Visual tools for analyzing, managing and communicating. Oxford University Press.
- Kulpa, Z. (1994) – Diagrammatic representation and reasoning. In *Machine Graphics & Visions* 3(1/2), p.77-103.
- Manovich, L. (2001) - The language of new media. The MIT Press.
- Norman, D.A. (1986) – Cognitive Engineering. In *User centered design* - D.A. Norman and S.W. Draper (eds). Lawrence Erlbaum, p.31-61.
- Norman, D.A. (2002) - The design of everyday things. Basic Books.
- Poltrock, P. and Grudin, J. (1994) - Computer supported cooperative work and groupware. *Conference companion on Human factors in computing systems CHI'94*, ACM Press, New York, p.355-356.
- Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T. (1994) - Human – Computer Interaction. Addison-Wesley.
- Preece, J., Rogers, Y. and Sharp, H. (2002) - Interaction design: beyond human - computer interaction. Wiley & sons.
- Rhyne, J.R. and Wolf, C.G. (1992) - Tools for supporting the collaborative process. *Proceedings of the ACM symposium on User interface software and technology UIST'92*, ACM Press, New York, p.161-170.
- Salomon, G. (1994) – New uses for Color. In *the art of human-computer interface design* - B. Laurel (eds.) Addison-Wesley, p.269-278.
- Schmidt, K. (2002) - The problem with 'Awareness': introductory remarks on 'awareness in

- CSCW'. In *CSCW: the journal of collaborative computing* 11(3), p.285-298.
- Simons, J. (2002) - Interface en cyberspace: inleiding in de nieuwe media. Amsterdam University Press.
- Stefik, M., Bobrow, D.G., Lanning, S., Tatar, D. and Foster, G. (1986) - WYSIWIS revised: early experiences with multi-user interfaces. *Reprinted from the proceedings of CSCW*.
- Tufte, E.R. (1990) - *Envisioning information*. Graphic Press.
- Tufte, E.R. (1997) - *Visual Explanations: design strategies for presenting information about motion, process, mechanism, cause and effect*. Graphic Press.
- Wenzel, S., Bernhard, J. and Jessen, U. (2001) – A taxonomy of visualization techniques for simulation in production and logistics. In *Proceedings of the 2003 Winter Simulation Conference*, p.729-736.

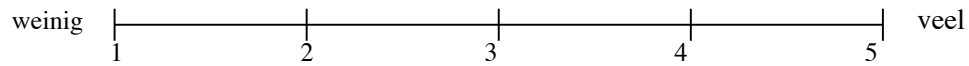


### Bijlage 1: Vragen vooraf

1. Wat is je leeftijd?

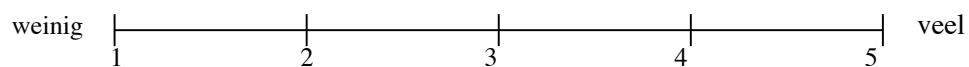
2. Wat is je beroep/studie?

3. Hoeveel computer-ervaring heb je? (geef aan op schaal van 1 (weinig) tot 5 (veel ervaring))



4. Heb je meer ervaring met multi-user applicaties? Zo ja, welke?

5. Hoeveel multi-user ervaring heb je met Keyworx? (geef aan op schaal van 1 (weinig) tot 5 (veel ervaring))

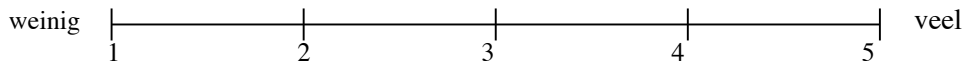


6. Waarvoor gebruik je Keyworx? (werk/plezier)

7. In hoeverre komt je beroep/studie overeen met wat je doet met Keyworx?

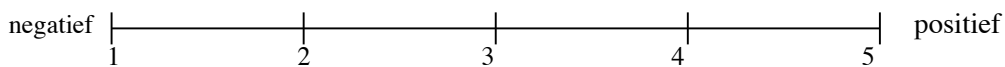
## Bijlage 2: Vragen achteraf

1. Heb je eerder met deze mensen aan een sessie deelgenomen?
2. Denk je dat het wel of niet eerder samenwerken van invloed was op het verloop van deze sessie?
3. Hoe groot was jouw rol/aandeel in deze sessie? Was je actief of vrij passief? (geef aan op schaal van 1 (passief) tot 5 (actief))

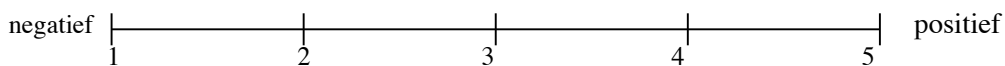


4. Hoeveel ervaring met Keyworx heb je ten opzichte van de andere deelnemers (of denk je te hebben)?  
minder / evenveel / meer
5. Speelde het verschil of overeenkomst in ervaring ten opzichte van andere deelnemers een rol in deze sessie? Zo ja, positief of negatief?
6. Had je tijdens de sessie duidelijk ideeën die je wilde uitvoeren, of bepaalde je op het moment zelf wat je ging doen, als reactie op wat er gebeurde?
7. Kon je altijd volgen wat er gaande was, of voelde je je weleens verloren of buitengesloten?
8. Voer je tijdens de sessie vaak dezelfde (combinaties van) handelingen uit? Zo ja, geef een voorbeeld.
9. Heb je 'favoriete' media-input en/of acties die je vaker toepast bij gebruik van keyworx? Zo ja, geef een voorbeeld.
10. Weet je waar de kleuren die gebruikt worden in Keyworx voor staan? (iedere deelnemer-input eigen kleur, elke soort medium (beeld, geluid) eigen kleur bolletje, alleen verbindingen met bolletjes van dezelfde kleur)

11. Hoe vind je het dat anderen jouw dingen kunnen veranderen? (geef aan op schaal van 1 (negatief) tot 5 (positief))



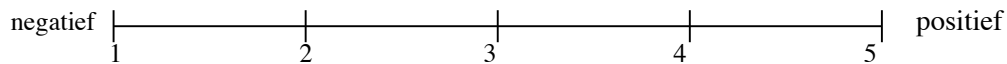
12. Hoe vind je het dat jij dingen van anderen kan veranderen? (geef aan op schaal van 1 (negatief) tot 5 (positief))



13. Wil je weleens dingen voor jezelf houden, bijvoorbeeld bepaald medium en bijbehorende effecten 'locken' zodat anderen er niets aan kunnen veranderen? Zo ja, hoe lang dan?

- a. nee
- b. ja, eventjes
- c. ja, zolang ik ermee bezig ben
- d. ja, gedurende de verdere sessie

14. Lijkt het je wenselijk als andere deelnemers toestemming aan jou moeten vragen wanneer ze iets aan jouw input willen veranderen? Waarom?
15. Had je behoefte aan aanvullende communicatie met de andere deelnemers?
16. Gebruik je weleens aanvullende communicatie-middelen (bv Ichat)?
17. Zou je willen zien met wie je werkt?
18. Zou je willen zien waar de andere deelnemers vandaan komen?
19. Zou je willen zien hoe actief de andere deelnemers zijn?
20. Zou je willen zien wat de andere deelnemers doen?
21. Zou je willen zien wat de huidige activiteit is van anderen?
22. Zou je willen zien wanneer media worden toegevoegd of verwijderd?
23. Zou je willen zien wanneer verbindingen worden gemaakt?
24. Zou je willen zien wanneer parameters worden veranderd?
25. Zou je jezelf duidelijker willen zien?
26. Zou je elke deelnemer apart gerepresenteerd willen zien (bijvoorbeeld iedere deelnemer een eigen stukje scherm toewijzen)?
27. Zou je in staat willen zijn gradaties aan te geven in weer te geven informatie, zodat je soms meer en soms minder informatie ziet?
28. Heb je wel eens behoefte aan een overzicht van wat er tot nu toe is gebeurd ('history')? Zo ja, zou je dit dan willen zien voor een bepaald input-bestand, of een bepaalde persoon, of beiden, of anders?
29. Heb je ideeën over hoe de visualisatie van informatie gedaan zou kunnen worden?
30. Wat is je algemene indruk van deze sessie? (geef aan op schaal van 1 (negatief) tot 5 (positief))  
Waarom?



31. Nog aanvullende ideeën, suggesties?

### Bijlage 3: antwoorden op vragen achteraf

	A	B	C	D	E	F	G
1	ja	ja	nee	ja	nee	nee	nee
2	ja	ja	ja	ja	nee	ja	ja
3	vrij passief	vrij actief	vrij actief	erg actief	vrij actief	vrij passief	passief
4	minder	minder	meer	evenveel	evenveel	minder	minder
5	ja	nee	ja	nee	nee	ja	ja
6	vooraf geen duidelijke ideeën, tijdens sessie reageren op wat er gebeurt						
7a	nee	nee	ja	ja	redelijk	nee	nee
7b	nee	nee	nee	nee	nee	nee	nee
8	ja	ja	ja	ja	ja	ja	ja
9	ja	ja	ja	ja	ja	nee	ja
10	nee	nee	ja	ja	ja	ja	ja
11	vrij positief	neutraal	erg positief	vrij positief	neutraal	vrij positief	erg positief
12	vrij positief	erg positief	erg positief	vrij positief	neutraal	vrij negatief	erg positief
13	ja	ja	nee	ja	ja	ja	ja
14	nee	nee	nee	nee	nee	nee	nee
15	ja	ja	ja	ja	nee	ja	ja
16	ja	ja	ja	ja	nee	ja	nee
17	ja	ja	ja	ja	nee	ja	ja
18	ja	ja	ja	ja	nee	ja	ja
19	neutraal	nee	misschien	nee	ja	nee	ja
20	ja	ja	nee	nee	ja	ja	ja
21	ja	ja	nee	ja		ja	ja
22	ja	ja	nee	nee	ja	ja	ja
23	ja	ja	nee	nee	ja	nee	ja
24	ja	ja	nee	nee	ja	ja	ja
25	nee	ja	nee	nee	nee	ja	ja
26	nee	ja	nee	nee	nee	nee	nee
27	ja	ja	ja	ja	nee	ja	ja
28	ja	nee	ja	ja	nee	nee	ja
29	ja	nee	ja	ja	ja	ja	ja
30	neutraal	neutraal	neutraal	neutraal	erg positief	neutraal	vrij negatief

#### **Bijlage 4 : Vragen over het prototype**

1. Is it usefull to see the connections between modules?
2. Is the extra information on mouse over with the modules enough/too little/too much?  
If too much/too little: what should be removed/added?
3. Is the difference between the modules clear (through color and place in screen)?
4. The size of the module changes according to its activity. Is this usefull?
5. Is it usefull to be able to see the users doing actions?  
If so, now this is done by showing a faded, thick line. Is this ok, or should it be done another way?
6. Would you also like to see which users adds a connection?
7. When a user leaves the session, a message is shown in the right top. Is this necessary?
8. Is there any information missing that you would like to see represented?
9. Do you think this visualization of information is usefull?
10. Any other comments, ideas, suggestions or questions?