

Tilburg University

Voortgangsverslag integratieproject DenK

Ahn, R.; Beun, R.J.

Publication date:
1991

Document Version
Publisher's PDF, also known as Version of record

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Ahn, R., & Beun, R. J. (1991). *Voortgangsverslag integratieproject DenK: Augustus 1990 - januari 1991*. (ITK Research Memo). Institute for Language Technology and Artificial Intelligence, Tilburg University.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy


If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

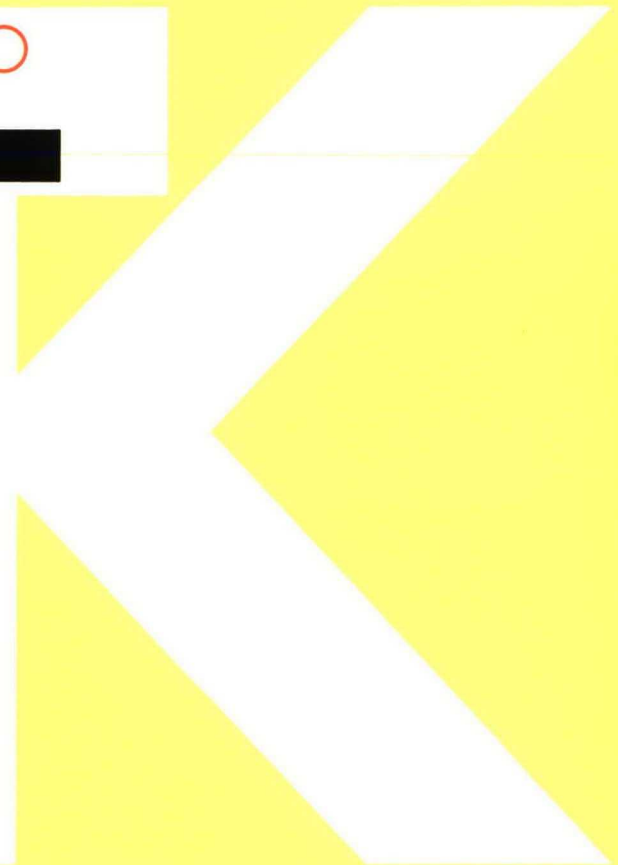
E. I.

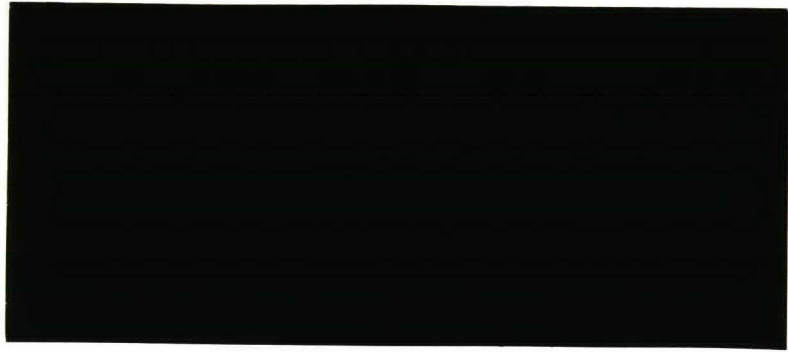
CBM
R
8419
1991
6
UNIVERSITY
UNIVERSITEIT
BRABANT



ITK

MEMO






 K.U.B.
BIBLIOTHEEK
TILBURG

I.T.K. Research Memo no. 6
February 1991

Voortgangsverslag
integratieproject DenK:
augustus 1990 - januari 1991

Rene Ahn
Robbert-Jan Beun

Dit ITK Research Memo verschijnt tevens als IPO rapport.

© 1990. Institute for Language Technology and Artificial Intelligence,
Tilburg University, The Netherlands

Samenvatting

In het kader van het zwaartepuntsproject 'Dialogvoering en Kennisopbouw' (DenK) van het Samenwerkingsorgaan Brabantse Universiteiten wordt vanaf mei 1989 gewerkt aan een aantal deelprojecten op het gebied van mens-machine communicatie en kennisrepresentatie. Anderhalf jaar na aanvang bleek duidelijke behoefte te bestaan om te starten met het zogenaamde 'integratieproject', waarbinnen de onderzoeksresultaten van de diverse deelprojecten in hun onderlinge samenhang belicht kunnen worden. Hierbij is het de bedoeling een interactief systeem te ontwerpen en implementeren dat zich beperkt tot de mogelijkheden die geboden worden door het huidige theoretische nivo.

Dit verslag is een weergave van de werkzaamheden die in het eerste half jaar verricht zijn in het kader van het integratieproject. Voordat een overzicht wordt gegeven van de globale architectuur, wordt eerst ingegaan op de achtergronden van het te ontwikkelen interactieve systeem. Het systeem kan hierbij voorgesteld worden als een coöperatieve assistent die niet alleen opgedragen commando's kan uitvoeren, maar tevens kan redeneren over de kennis en de intenties van de gebruiker. In de gepresenteerde architectuur hebben zowel de gebruiker als het systeem directe toegang tot het gespreksdomein. De gebruiker kan tevens d.m.v. talige uitingen via het systeem objecten wijzigen en vragen stellen over de eigenschappen van objecten. Het gebruikte formalisme voor de representatie van de uitingen (Constructieve Type Theorie) maakt het hierbij mogelijk om informatie die in de loop van de dialoog beschikbaar komt onmiddellijk te gebruiken in het vervolg van de dialoog.

Summary

Since May 1989, a number of subprojects is carried out in the framework of the project 'Dialogue and Knowledge Construction' of the Cooperation Center Tilburg and Eindhoven University. All projects are related to research in the field of man-machine communication and knowledge representation. After one and a half years the so called 'integrationproject' was started, where the results of the subprojects could be put together. This was supposed to be embodied in the design and implementation of an interactive system that was restricted to the possibilities of the current theoretical level of expertise.

In this paper, the activities are reported that were carried out in the first half year of the integration project. Before the architecture is described, attention is payed to some background details of the interactive system. The system can be represented as a cooperative agent who can reason about the knowledge and intentions of the user. In the architecture presented, both the user and the system have direct access to the domain of discourse. The user can also manipulate objects and examine the characteristics of objects indirectly by uttering speech acts to the system. The formalism used to represent the utterances (Constructive Type Theory) enables the system to use the information that comes available during the dialogue immediately in the sequel of the dialogue.

Inhoudsopgave

1	Primaire doelstelling van het integratieproject	1
2	Achtergrond: de communicatiemetafoor	2
2.1	De gemodelleerde wereld	2
2.2	De coöperatieve assistent	3
2.3	Het gemodelleerde domein en de coöperatieve assistent gecombineerd	5
3	Globale architectuur	7
3.1	Inleiding	7
3.2	Functionaliteit	8
3.3	Verdeling	8
3.4	Practische aspecten	9
4	De modules van het eerste prototype	10
4.1	WALT	10
4.2	USER	11
4.3	DABAS	11
4.4	KONTEX	12
5	Aandachtspunten en activiteiten voor de toekomst	15
5.1	Het eerste prototype	15
5.2	De verdere ontwikkeling	16
6	De geprojecteerde gebruikersgroep en gebruiksomgeving	18
7	Conditie voor het welslagen van het project	20
8	Referenties	22
A	De modules van het systeem en hun onderlinge samenhang	23
B	Lijst van gebruikte woorden	24

Voorwoord

Per 1 augustus 1990 hebben we de opdracht op ons genomen om voor een periode van een half jaar aan de integratie van het SO zwaartepuntproject 'Dialogvoering en Kennisopbouw' te werken.¹ Onze taak kon als volgt samengevat worden:

1. Het vaststellen van de eisen en wensen die gesteld kunnen worden aan het uiteindelijk te bouwen systeem.
2. Een globaal functioneel ontwerp van het te ontwikkelen systeem.
3. Terugkoppeling naar de DenK werkgroep van vragen m.b.t. de deelprojecten.
4. Activiteitenplan voor het komende half jaar.
5. Het bepalen van relevante kenmerken van de geprojecteerde gebruikersgroep en gebruiksomgeving.
6. Formulering van condities voor het welslagen van het integratieproject.

In dit verslag wordt allereerst geprobeerd de doelstellingen van het integratieproject vast te leggen (hoofdstuk 1) en de achterliggende problematiek van het DenK project te verduidelijken (hoofdstuk 2). Daarna zal worden ingegaan op de globale architectuur van het voorgestelde systeem (hoofdstuk 3), de daarbij behorende aparte modules (hoofdstuk 4) en enige aandachtspunten en activiteiten voor de toekomst (hoofdstuk 5). Vervolgens zullen de resterende punten van de bovengenoemde taakstelling worden behandeld.

We zijn ons zeer bewust van de communicatieproblemen die een multidisciplinair project als DenK met zich meebrengt. De integratie houdt dan voor een deel ook in dat de werkgroepleden op de hoogte raken van elkaars werk en de verbanden zien die hiertussen bestaan. Daarnaast is het vooral voor de uitvoerders van belang dat naar concrete doelen gewerkt kan worden, met behoud van de grootst mogelijke vrijheid voor het eigen onderzoek. Belangrijk is hierbij onder andere het gebruik van dezelfde formele taal (of talen) en het spreken over een gemeenschappelijk domein. Hiermee wordt een deel van de vaagheid en de abstractie over elkaars werk weggenomen

¹Voor een uitvoerige beschrijving van het project en de voortgang verwijzen we naar de Programma-beschrijving (1989), Beun, Dols, Jaspars & de Reus (1990) en Ahn, Beun, Dols, Jaspars & de Reus (1990).

hetgeen de communicatie zeker ten goede zal komen. Een en ander zal na een eerste verslag misschien nog niet lukken, maar we vertrouwen erop dat na de nodige interactie met de leden dit wel mogelijk zal zijn. Zowel positieve als negatieve kritiek in de vorm van vragen en opmerkingen zijn dan ook van harte welkom.

Wij wachten overigens nog steeds op electronic mail adressen, de onze zijn:
ahn@kub.nl
beun@heiipo5.bitnet

1 Primaire doelstelling van het integratieproject

Het hoofddoel van het integratieproject is het ontwikkelen van een coöperatieve multimodale mens-machine interface voor kennisoverdracht en -analyse door middel van talige en grafische interactie.

Om de interactie met het systeem taakgerichter en effectiever te laten zijn dan in gebruikelijke systemen zullen bij het ontwerp drie punten bijzondere aandacht krijgen:

- De uit de interactie resulterende dialoog heeft een *voortschrijdend* karakter, d.w.z. dat de kennis die opgenomen wordt tijdens de dialoog onmiddellijk gebruikt kan worden in het vervolg van de dialoog.
- Er zal gebruik gemaakt worden van *multimodale* interactie, d.w.z. dat verschillende interactie-modaliteiten geïntegreerd zullen worden, zodat men bijvoorbeeld in de taal kan verwijzen naar een object dat men op het scherm aanwijst.
- Het systeem dient niet alleen kennis op te kunnen bouwen over het eigenlijke gespreksdomein, maar ook over de gebruiker, d.w.z. over diens *kennis* en *intenties*.

2 Achtergrond: de communicatiemetafoor

We zullen in dit hoofdstuk een korte schets geven van de achterliggende problematiek zoals wij die zien. De globale architectuur zal hierop nauw aansluiten, zodat we hiermee tevens een onderbouwing hebben gegeven van de functie van de modules die in het volgende hoofdstuk aan de orde komen. We zullen er hierbij vanuit gaan dat een mens op twee manieren in interactie kan staan met zijn omgeving: een *fysische* en een *talige* interactie. Zo kan een mens voorwerpen verplaatsen, vastmaken, etc., maar ook door middel van taaluitingen een gesprekspartner informeren, vragen stellen of instructies geven. Analooq aan dit onderscheid zullen in het ontwerp twee aspecten een cruciale rol spelen: a. de gemodelleerde (fysische) wereld en b. de coöperatieve assistent.

2.1 De gemodelleerde wereld

In de wereld waarin we leven hebben we veelal te maken met objecten, hun eigenschappen en de handelingen die we kunnen uitvoeren op de objecten of de handelingen die de objecten zelf kunnen uitvoeren. Zo hebben fietsers een bepaalde snelheid, staan blokken naast of op elkaar, kunnen gegevensbestanden gecreëerd of verwijderd worden, etc. Wanneer we nu in staat zijn de samenhang tussen de eigenschappen van de wereld te formaliseren spreken we van een *gemodelleerde wereld*. In bepaalde gevallen is het zelfs mogelijk de eigenschappen van de wereld zodanig te verwoorden dat deze geïmplementeerd kunnen worden in een computer. Een eenvoudig voorbeeld hiervan is een data-base met aankomst- of vertrektijden van vliegtuigen zoals deze in het TENDUM-systeem wordt gehanteerd (zie bijv. Bunt, Beun, Dols, van der Linden & thoe Schwartzenberg 1984); in meer geavanceerde systemen kunnen ook fysische wetten worden geïmplementeerd, zodanig dat de objecten die hierin voorkomen voldoen aan eigenschappen die ze normaalgesproken ook zouden hebben in de reële wereld. Denk bij dit laatste bijvoorbeeld aan de weersvoorspellingen of aan het WALT-systeem (van Overveld 1990)

Doordat we nu in staat zijn een wereld te implementeren hebben we te maken met een belangrijk probleem. Voor de interactie met de gemodelleerde wereld kunnen we dikwijls niet meer gebruik maken van onze vertrouwde manier van *observeren* en *handelen*. Daar waar we in de reële wereld kunnen kijken naar de baan van een in beweging gebrachte voetbal, kunnen we nu in veel gevallen slechts kijken naar een uitvoer van getallen

die bijvoorbeeld de coördinaten of de snelheid van de bal weergeven²; waar we vroeger onze handen konden gebruiken om voorwerpen te grijpen of te duwen, moeten we nu dikwijls ingewikkelde talige commando's invoeren om hetzelfde resultaat te verkrijgen.

In de huidige generatie interfaces worden deze problemen slechts zeer ten dele ondervangen; met de muis kunnen we wel objecten aanwijzen of veranderen, maar de handelingen die men hiermee kan verrichten zijn dikwijls primitief en hebben in veel gevallen nauwelijks enige betekenis indien ze niet in samenhang met een menu worden uitgevoerd. Ook verandert soms de functie van de muis afhankelijk van de plaats op het scherm en sluit de symbolische weergave van de functie in veel gevallen niet aan met onze dagelijkse ervaringen (bijv. een 'pijl' om objecten te grijpen). De verplaatsing van onze dagelijkse ervaringswereld naar een geïmplementeerde wereld kan dus ernstige misconcepties tot gevolg hebben, welke op hun beurt voor een groot ongemak en verlaging van de efficiëntie tijdens de interactie kunnen zorgen. Het is daarom van essentieel belang dat de representatie van objecten met hun eigenschappen zo nauw mogelijk aansluit met de ervaringen in de reële wereld van de systeemgebruiker. (Zie bijv. ook van Nes 1990)

2.2 De coöperatieve assistent

In een fysische wereld zijn we niet in staat om door middel van taaluitingen directe observaties of manipulaties te doen; dit is alleen indirect mogelijk indien de taal geïnterpreteerd wordt door een gesprekspartner. Zo kunnen we met de taalhandeling "*Plaats object1 op object2!*" alleen het gewenste resultaat bereiken als we een gesprekspartner hebben die de talige elementen op de juiste wijze kan interpreteren, die in staat is de handeling uit te voeren en die de bereidwilligheid toont om dit daadwerkelijk te doen.³

In het algemeen gedraagt zo'n gesprekspartner zich *rationeel en coöperatief*. We zullen in het vervolg dan ook spreken over een *coöperatieve assistent* (CA). Het rationele gedrag houdt onder meer in dat een CA kan redeneren over de manier waarop hij zijn doelen op een efficiënte wijze kan bereiken (plannen). Coöperativiteit houdt in dat een CA probeert (tot op zekere

²Het voetbalspel zou, indien op deze manier weergegeven, geen onbelangrijk deel van zijn aantrekkingskracht verliezen.

³Merk overigens op dat door uitspraken als "*Hierbij ontsla ik je*" of "*Hierbij verklaar ik u man en vrouw*" primair veranderingen optreden in een sociale wereld en niet zozeer in een fysische (hoewel dat laatste natuurlijk onvermijdelijk is).

hoogte) de doelen van de gebruiker te verwezenlijken (Allwood 1976). Een CA kan dus niet alleen redeneren over zijn eigen kennis en intenties, maar ook over de intenties en kennis van zijn gesprekspartner. Dit impliceert dat een CA niet alleen de letterlijke betekenis van de uiting kan herkennen en de daarbijbehorende handelingen kan uitvoeren, maar dat tevens, afhankelijk van contextuele kennis, de achterliggende doelen van de gebruiker herkend en de bijbehorende handelingen uitgevoerd kunnen worden. Zo zal hij of zij op de zin "*Kunt u mij vertellen hoe laat het is?*" niet alleen in staat zijn in het positieve geval een letterlijk antwoord op de vraag te bedenken ("*Ja*"), maar afhankelijk van de situatie ook het juiste tijdstip verstrekken. In andere gevallen, zoals in "*Verwijder alle files!*", gedraagt een CA zich soms coöperatiever indien het commando juist niet of slechts gedeeltelijk wordt uitgevoerd, bijvoorbeeld indien de CA weet dat er een vergissing is gemaakt of indien men in een bepaalde subdirectory werkt.

Behalve kennis over rationeel en coöperatief gedrag, zal een CA ook beschikken over *contextuele* kennis. Iedere uiting wordt immers gedaan binnen een bepaalde context en de betekenis van iedere uiting is dan ook afhankelijk van die context. Deze contextuele kennis kan zowel afkomstig zijn van reeds aanwezige achtergrondkennis als van informatie die beschikbaar komt tijdens de gevoerde dialoog. Zo is de betekenis van de tweede zin in "*De auto reed tegen de muur. De bumper was ingedeukt.*" alleen te bepalen indien bekend is dat bumpers een onderdeel van auto's kunnen zijn (achtergrondkennis) en wanneer we ervan uitgaan dat gebeurtenissen die elkaar opvolgen in het algemeen na elkaar worden vermeld. Het ligt hier immers voor de hand om aan te nemen dat de deuk in de bumper zit van de auto die tegen de muur reed en dat tussen beide gebeurtenissen een causaal verband bestaat, alhoewel dit strikt genomen niet wordt meegedeeld.⁴

Verder kan een CA niet alleen declaratieve kennis communiceren, maar is in staat aan zijn uiting functionele kenmerken toe te voegen. In functioneel (of pragmatisch) opzicht heeft een CA in een natuurlijke taal dialoog drie basis-taalhandelingen tot zijn beschikking: 'beweren', 'commanderen' (of 'verzoeken') en 'vragen'. De taalhandelingen kunnen syntactisch tot uitdrukking gebracht worden door de drie zinstypes 'declaratief', 'imperatief' en 'interrogatief'. Het is echter belangrijk op te merken dat in natuurlijke dialogen de afbeelding van het syntactisch zinstype naar de pragmatische functie niet eenduidig is (Beun 1989).

⁴Strikt genomen weten we niet zeker of de bumper inderdaad de bumper van de eerder geïntroduceerde auto is. We spreken daarom liever van een *default-aanname*.

In een belangrijk deel van de huidige software lopen vrijwel alle observaties en handelingen die uitgevoerd moeten worden op de gemodelleerde wereld via een geïmplementeerde talige interface. De gebruiker kan dus geen directe veranderingen en observaties op het domein uitvoeren. Deze configuratie is vergelijkbaar met een telefoonsituatie tussen menselijke gesprekspartners, waarbij een van de partners directe toegang heeft tot het gespreksdomein.⁵ De huidige talige interfaces zijn echter, in tegenstelling tot een menselijke gesprekspartner, niet in staat een tekstuele context op te bouwen of coöperatief gedrag te vertonen.

2.3 Het gemodelleerde domein en de coöperatieve assistent gecombineerd

In de wereld waarin we leven maken de gesprekspartners dikwijls deel uit van elkaars omgeving. Dit heeft als belangrijke consequentie dat er voor beide partners een gemeenschappelijke werkruimte bestaat, zoals een schoolbord, werktekening of bouwplaats (zie ook Curtis & Mallone 1988). Deze werkruimte speelt een zeer belangrijke rol in de communicatie. De gesprekspartners weten namelijk hierdoor wat de ander ziet of hoort en kunnen dus allebei onmiddellijk op de hoogte zijn van de handelingen en observaties die op de wereld worden uitgevoerd. Dit stelt beiden onder andere in staat objecten aan te wijzen in plaats van te benoemen. Een belangrijke eigenschap van deze configuratie is verder dat bij de gesprekspartners een zo goed mogelijke overeenstemming bestaat over de kennis van de wereld.

In een dergelijke configuratie hebben we dikwijls de mogelijkheid om van verschillende kanalen gebruik te maken voor de communicatie over het gespreksdomein. We spreken in zo'n geval dan van een *multimodale dialoog* (Taylor, Neél & Bouwhuis 1989). Dezelfde informatie kan bijvoorbeeld op symbolische wijze of in de vorm van grafische data worden gerepresenteerd.⁶ Ieder van deze representaties heeft zo zijn eigen voor- en nadelen. Een symbolische (talige) weergave is bijvoorbeeld geschikt voor het stellen van vragen en het kwantificeren over objecten, maar bezit daarentegen dikwijls een grote mate van ambiguïteit (zelfs binnen een beperkt domein) of sluit niet aan bij de natuurlijke representatie zoals mensen die gewend zijn. Anderzijds is bijvoorbeeld grafische representatie bijzonder geschikt voor het

⁵Ook het TENDUM-systeem is een voorbeeld van deze configuratie, met als verschil dat als interactietaal gestreefd wordt naar natuurlijke taal.

⁶Merk op dat het multimodale aspect van een dialoog ook kan voorkomen in de configuratie zoals beschreven is onder 2.2.

weergeven van topologische of geometrische kennis, maar kunnen zinnen als *"Auto's kunnen in het algemeen harder rijden dan fietsers"* of *"Grafische representatie is niet geschikt voor het weergeven van de betekenis van deze zin"* niet of nauwelijks in grafische vorm worden weergegeven. Dikwijls ook zullen verschillende modaliteiten gecombineerd moeten worden zoals in de zin *"Zet dat daar!"*.

Afhankelijk van de toepassing kan men ook in mens-computer dialogen kiezen voor verschillende representaties voor de informatie-overdracht. Sommige gespreksdomeinen zullen zich daarbij beter lenen voor symbolische interactie, andere beter voor een grafische interactie. De kracht van een goede interface ligt echter zowel in de flexibiliteit in het gebruik van de verschillende modaliteiten ten opzichte van de verschillende domeinen, als in het gebruik van een combinatie van die modaliteiten.

In de door ons gepresenteerde architectuur stellen we de configuratie voor waarbij zowel de (geïmplementeerde) CA als de gebruiker in staat zijn direct te observeren en handelen op de gemodelleerde wereld en met elkaar kunnen communiceren door middel van taaluitingen en aanwijzen.

In principe zullen de CA en de gebruiker zoveel mogelijk symmetrische eigenschappen moeten bezitten. Beiden zullen bijvoorbeeld over linguïstische kennis (syntaxis en semantiek) en regels voor het gebruik van taal (pragmatic) moeten beschikken. Op een tweetal punten zal de symmetrie echter afwijken: a. de CA zal geen eigen doelen verzinnen en b. de CA zal andere mogelijkheden bezitten om te communiceren met de gemodelleerde wereld. In dit laatste geval kan men bijvoorbeeld voorstellen dat de CA bepaalde objecten kan creëren of doen oplichten, of dat de CA objecten kan 'zien' die op het scherm voor de gebruiker verborgen blijven. In praktijksituaties zal de asymmetrie voornamelijk veroorzaakt worden doordat een gebruiker van het systeem vragen heeft over het domein of een bepaalde toestand binnen het domein wil bereiken (bijv. *"Wat is de snelheid van auto A?"* of *"Zet de eerste alinea in hoofdletters!"*). De CA kan dan geacht worden in staat te zijn de vragen te beantwoorden of door het uitvoeren van bepaalde handelingen de gewenste toestand te bereiken. M.a.w. de gebruiker is expert over de gewenste toestand en de CA is expert over de manier waarop deze toestand verwezenlijkt kan worden. Ondanks deze verschillen is er in principe een duidelijke symmetrie die ook in de architectuur weerspiegeld dient te worden.

3 Globale architectuur

3.1 Inleiding

De eerste taakstelling is om te komen tot een globale architectuur van het systeem, een ontwerp wat voldoende flexibel is om latere wetenschappelijke inbreng uit de deelprojecten te incorporeren, en tegelijk voldoende concreet om op korte termijn vorm te geven aan een eerste prototype gebaseerd op reeds bestaande inzichten. Er kunnen verschillende aspecten worden onderscheiden die bij de ontwikkeling van belang zijn; de projecten binnen het DenK programma hebben alle betrekking op een van deze aspecten:

- *Het database-aspect.* De in het systeem aanwezige kennis over het gespreksdomein zal immers voor een belangrijk deel in een database zijn opgeslagen. (Projekt 2 en 3)
- *Het grafische weergave en interactie-aspect.* Zoals reeds vermeld, is het van groot belang dat er een rechtstreekse grafische weergave kan zijn van elementen van het gespreksdomein, zodanig dat de gebruiker deze elementen rechtstreeks kan manipuleren. (Projekt 7b)
- *Het gebruikers-aspect.* Het zal duidelijk zijn dat men alleen kan komen tot directe en natuurlijke dialogen tussen systeem en gebruiker, indien men bereid is aandacht te schenken aan de communicatieve vaardigheden en verwachtingen (bijv. protocollen) van deze laatste. (Projekt 5)
- *Het dialoog-aspect.* Het systeem moet door de gebruiker gedane bewerkingen rechtstreeks en incrementeel kunnen opnemen in zijn kenistoestand, en bijv. kunnen omgaan met latere referenties naar de inhoud van deze bewerkingen. Een verfijning hiervan is het doxastisch/boulemaïsch modelleren. Dat houdt in dat het systeem bijhoudt wat de gebruiker waarschijnlijk wel en niet weet, alsmede zijn vermoedelijke graad van expertise en zijn doelen, om efficiënte interactie mogelijk te maken. (Project 1, 6 en 7a)

In verband met tijdens de uitvoering van de deelprojecten opgedane kennis is het belangrijk dat deze aspecten geïntegreerd worden op een wijze die de grootst mogelijke vrijheid laat aan alle deelprojecten. In het vervolg zullen wij naar deze aspecten en de bijbehorende systemen of formalismen respectievelijk verwijzen als: DABAS, WALT, USER en KONTEX. (Zie ook appendix 1.)

3.2 Functionaliteit

Het eerste prototype van het systeem staat de gebruiker toe, om, gegeven een signatuur c.q. typering van een wereld, in deze wereld manipulaties en observaties uit te voeren, en een dialoog te voeren met een coöperatieve assistent met deze wereld als gespreksdomein. Bij gebrek aan een betere naam noemen we de coöperatieve assistent voorlopig WATSON.

De objecten binnen het gespreksdomein hebben een dynamisch gedrag dat doorlopend op het scherm zichtbaar wordt gemaakt. Zowel WATSON als de gebruiker kunnen objecten in de wereld creëren, manipuleren en observeren. In dit prototype kan de gebruiker wanneer het systeem is opgestart dus geen nieuwe types meer creëren, en ook geen nieuwe predicaten definiëren. Er kan bovendien niet gesproken worden over objecten die het systeem nog niet kent, en het systeem gedraagt zich dan ook feitelijk *alwetend* t.a.v. de bestaande toestand van het domein, evenals t.a.v. de geschiedenis ervan. De gebruiker kan wel vragen stellen over het gespreksdomein of opdracht geven er iets in te veranderen. Omdat het systeem alwetend is, zullen asserties geen kennis over het gespreksdomein toevoegen aan het systeem, maar door het systeem in principe alleen gecontroleerd kunnen worden, d.w.z. met “ja” of “nee” worden beantwoord.

3.3 Verdeling

WALT zal verantwoordelijk zijn voor de *representatie* van het gespreksdomein en biedt mogelijkheden om dit domein te inspecteren en te manipuleren.

WATSON is te verdelen in twee modules:

- DABAS, een database waarin *kennis* over het gespreksdomein is opgeslagen.
- KONTEX, waarin kennis over het verloop van de dialoog en over de gebruiker is opgeslagen.

DABAS zal eerste orde queries over de bestaande objecten kunnen beantwoorden, i.h.a. na ruggespraak met WALT. Ook eenvoudige vragen t.a.v. de historie moeten beantwoord kunnen worden.

KONTEX krijgt de vragen en opdrachten van de gebruiker binnen, moet trachten deze te interpreteren en er vervolgens adequaat op te reageren. Voorlopig wil dat zeggen dat er een antwoord wordt gegeven op vragen

(na ruggespraak met DABAS) en dat opdrachten door het systeem worden uitgevoerd (door het aanroepen van services van WALT), d.w.z. dat desgevraagd objecten worden verschoven, etc.

3.4 Practische aspecten

Voorlopig lijkt ons als software platform geschikt: C en PROLOG.

Window System: te bepalen door WALT; dat is nu Open Windows (de X window versie van SUN, gecombineerd met News). Het zou aan te raden zijn om News faciliteiten buiten het systeem te houden. News is een zinkend schip. Bovendien wil je gewoon dat software op X draait, dat is de de-facto standaard.

Hardware: SUN Workstation: SPARC.

4 De modules van het eerste prototype

4.1 WALT

WALT zal een object-georiënteerd systeem zijn, d.w.z. dat de objecten in de wereld die door WALT wordt gerepresenteerd allen deel uitmaken van bepaalde voorgedefiniëerde KLASSES. Bij elke klasse horen dan z.g. METHODS, dat zijn specificaties van gedrag die elementen uit die klasse op verzoek kunnen vertonen. Elementen uit een klasse kunnen verschillende *topologie* (d.i. een beschrijving van de bijbehorende ruimtelijke informatie) hebben. Een *signatuur* van het WALT systeem is een specificatie van de classes, methods en topologieën van een gewenste wereld. Deze specificatie dient aan het WALT systeem te worden aangeboden in een object-georiënteerde beschrijvingstaal (OGB). Gegeven zo een signatuur, biedt WALT de volgende diensten aan:

- Een continue grafische weergave van het domein op een display.
- De mogelijkheid om methods aan te roepen.
- De mogelijkheid om de aanroep van bepaalde methods direct te koppelen aan bepaalde muishandelingen.

In het algemeen zijn er drie belangrijke soorten van methods:

- Methods die, uitgaand van een combinatie van een klasse en een topologie⁷ een object creëren. Dit zijn de z.g. INSTANTIATION methods.
- Methods die bestaande objecten bepaalde gedragingen kunnen laten uitvoeren. We zullen deze methods voortaan aanduiden als ACTION methods. Deze methods brengen dus veranderingen in de toestand van het domein teweeg.
- Methods om gegevens over de objecten, i.h.b. de waarden van hun attributen, op te vragen of zelfs continu te observeren. We noemen dit INSPECTION methods. Deze laatste hebben als bijzonderheid dat ze dus geen enkele verandering aanbrengen in de toestand.

De combinatie van een tijdstip en een method aanroep noemen we een EVENT. De geschiedenis van een wereld kan dus volledig worden vastgelegd door alle events op te slaan. De attributen met bijbehorende action en

⁷Zo een combinatie heet in WALT een *prototype*.

inspection methods die we veronderstellen bij de meeste objecten t.b.v. de integratie van grafische en tekst interface zijn:

- **index**: boolean (geeft aan of de gebruiker ernaar wijst).
- **glow**: boolean (geeft aan dat de machine ernaar wijst middels highlighting).

4.2 USER

De gebruiker⁸ kan een tweetal verschillende interacties vertonen:

- **Ontwerpinteractie.**

Tijdens de ontwerpinteractie kan de gebruiker klassen, methods en topologieën toevoegen of wijzigen. Hier wordt dus de signatuur van het gespreksdomein veranderd. Deze interactiemodus zal in het eerste prototype verder niet ondersteund worden. Ontwerpen komt dan dus neer op het schrijven van een file in OGB. Dit betekent dat de signatuur na het opstarten niet meer kan worden gewijzigd.

- **Animatie-interactie.**

Hier kan de gebruiker binnen het raamwerk van een bepaalde signatuur de geschiedenis van de wereld beïnvloeden, maar hij kan deze wereld zelf niet veranderen. Deze interactiemodus zal in het eerste prototype worden ondersteund.

De gebruiker krijgt tijdens de animatie-interactie een overzicht van de situatie aangeboden via het scherm waarop een gedeelte van de actuele toestand van de wereld zichtbaar is. De gebruiker kan deze toestand desgewenst rechtstreeks via de muis manipuleren. Bovendien kan de gebruiker schriftelijk vragen stellen of opdrachten aan het systeem geven die betrekking mogen hebben op door hem op het scherm aangewezen objecten. Asserties van de gebruiker over het gespreksdomein worden als declaratieve vragen behandeld.

4.3 DABAS

DABAS krijgt van KONTEX zogenaamde QUERIES binnen en dient deze te beantwoorden. Daaronder vallen, behalve vragen over het nu, bijv. ook

⁸De termen 'gebruiker' en USER zullen verder doorelkaar gebruikt worden.

vragen over het verleden. De vragen zullen worden gesteld in getypeerde eerste orde logische taal (EOL). De typering dient overeen te komen met de signatuur van het gespreksdomein. DABAS kan bij het beantwoorden van de vragen gebruik maken van de services die WALT aanbiedt. DABAS hoeft geen rekening te houden met partialiteit, maar zal wel signaleren als een vraag buiten zijn bereik ligt. In het algemeen zal DABAS vragen alleen beantwoorden met TRUE, FALSE of een verzameling van entiteiten. Dit type antwoorden wordt verder RESULT genoemd.

DABAS zal de events die optreden binnen WALT opslaan, zodat de geschiedenis van WALT ten behoeve van vragen betreffende de historie later opnieuw kan worden nagespeeld. Dit naspelen kan men efficiënter maken met behulp van een checkpointing techniek.

Omdat sommige methods binnen WALT kunnen functioneren als 'watchdogs', kan men dan ook 'wanneer'-vragen beantwoorden. De complexiteit van dit soort vragen hangt af van de complexiteit van de watchdogs die worden aangeboden door WALT en is nog niet precies bekend. Vermoedelijk kunnen geen vragen worden behandeld waarin quantoren over objecten binnen een tijdsquantor voorkomen.

4.4 KONTEX

KONTEX krijgt van de gebruiker uitingen binnen en tracht daarop adequaat te reageren. Om dit te bewerkstelligen worden voorlopig de volgende taken verricht:

- De gemeenschappelijke context van systeem en gebruiker wordt t.g.v. deze uiting uitgebreid.
- De in deze context geïntroduceerde entiteiten dienen te worden gekoppeld aan entiteiten in het domein.
- Het systeem tracht een bij de uiting passende reactie te genereren.

We gaan er vanuit dat de in KONTEX binnenkomende teksten type theorie teksten zijn (voortaan TEXT), aangevuld met een label dat pragmatische informatie over de functie van de tekst bevat.

De gemeenschappelijke context is een context in Type Theorie. De initiële context wordt hierbij gegenereerd uit de signatuur van het gespreksdomein. Steeds als de gebruiker een uiting doet wordt deze aan de gemeenschappelijke context toegevoegd. Nieuwe contextvariabelen die zo ontstaan

vervullen de rol van *discourse markers*. Omdat het model van het systeem over de wereld DABAS is, moet het systeem de context interpreteren in DABAS. Dat is een kwestie van het vinden, in DABAS, van *satisfying assignments* (zie Ahn & Kolb 90), die aan de discourse markers entiteiten uit de wereld toevoegen, en dus een functie definiëren (de interpretatiefunctie: voortaan IF) van de variabelen naar de entiteiten in de wereld. Dit is recht toe recht aan, modulo ambiguïteiten. Als er geen satisfying assignment is, betekent dit, omdat de kennis over de wereld compleet is, dat de uitspraak geen juiste interpretatie kent. Het systeem zal de gebruiker daarop moeten wijzen. Als er wel zo een interpretatie is, levert deze de verdere uitbreiding voor de interpretatiefunctie. Dat stelt ons in staat om verder in de dialoog anaforsch naar de markers in de context, en dus indirect naar de entiteiten in DABAS te verwijzen. Overigens kunnen invullingen voor gegenereerde discourse markers niet alleen gegenereerd worden door DABAS, d.w.z. worden gekoppeld aan entiteiten in de wereld, maar ook door KONTEX, hetgeen wil zeggen dat ze worden gekoppeld aan eerdere introducties van entiteiten in de dialoog.

Om KONTEX te laten reageren op uitingen van de gebruiker, definiëren we drie pragmatische functies van de tekst: ASSERTION, INSTRUCTION en QUESTION.

- Bij assertions gebeurt er niets anders dan dat wat boven beschreven is.
- Instructions dienen als interpretatie een method-aanroep te hebben. Het is nog niet volledig duidelijk hoe zo een aanroep zou moeten worden gerepresenteerd. Men kan hier denken aan de introductie van een event in de context. In elk geval moet de gewenste method door het systeem worden aangeroepen.
- Questions verschillen weinig van assertions, behalve dat het bij WH-vragen gewenst is dat de gevonden satisfying assignments deels terug worden gemeld aan de gebruiker.

We kunnen echter niet alle antwoorden van DABAS rechtstreeks doorgeven aan de gebruiker. Het is wellicht niet zo moeilijk om de constanten TRUE en FALSE in een antwoord aan de gebruiker te vertalen, maar dit wordt moeilijker indien er sprake is van een verzameling van entiteiten. De manier waarop zo een verzameling aan de gebruiker dient te worden gepresenteerd kan verschillen. Dit hangt o.a. af van de cardinaliteit van de

verzameling. Een kleine, niet lege verzameling van entiteiten (zoals een singleton) kan worden aangegeven door enumeratie van de elementen. Sommige entiteiten hebben wellicht namen die aan de gebruiker bekend zijn, bijv. methods, de gebruiker zelf, het systeem, tijdstippen, etc. De gebruiker kent echter de interne objectnamen van DABAS niet. Om de identiteit van deze elementen aan de gebruiker duidelijk te maken dient te worden gezocht naar unieke eigenschappen van deze entiteiten die de gebruiker wel bekend zijn, zoals:

- ze waren reeds onderwerp van gesprek in de dialoog.
- ze hebben bepaalde opvallende eigenschappen.
- er wordt een in het oog springende eigenschap van het object gecreëerd (denk bijv. aan highlighting).

Merk op dat informatie die wordt verkregen via de gebruiker nooit de directe kennis over het domein kan beïnvloeden (d.w.z. DABAS), omdat het systeem immers alwetend is.

5 Aandachtspunten en activiteiten voor de toekomst

5.1 Het eerste prototype

- Een punt van zorg is de omgang met de dynamische aspecten binnen het systeem. De gebruiker kan het systeem instructies geven terwijl bijv. nog niet duidelijk is wat dit voor effecten zou moeten hebben op de gemeenschappelijke context. Wel zien we mogelijkheden om een in de tijd veranderende werkelijkheid te beschrijven m.b.v. contexten in type theorie. Door al die aspecten van de werkelijkheid die in de loop van de tijd kunnen veranderen te beschrijven d.m.v. predicaten, kunnen we, als objecten veranderen t.g.v. methods, eenvoudig een nieuwe passende context genereren door slechts betrokken 'atomaire' bewijsobjecten te vervangen. Conclusies die getrokken zijn op grond van niet langer geldige waarnemingen kunnen dan eenvoudig worden geïdentificeerd via het voorkomen van vrije variabelen in de samengestelde bewijsobjecten. Zo kunnen we dynamische wijziging van types vermijden en kunnen sommige eigenschappen van objecten veranderen terwijl andere eigenschappen van het betrokken object onaangetast blijven. We zullen ons het komende half jaar met name op dit dynamische aspekt richten.
- Er is op dit moment binnen het project geen ruimte voor de ontwikkeling van een parser. Er zal moeten worden nagegaan in hoeverre bestaande parsers of parser generatoren kunnen worden ingezet. Om wille van eenvoud zouden we voorlopig problemen t.a.v. parsing willen omzeilen door de textuele interface met de gebruiker in een TEXT-achtig formaat te doen plaatshebben. Dit betekent dus dat er ook nog geen lexicon is, maar dat de 'woordenschat' van de gebruiker bestaat uit de constanten in de initiële context, zoals die wordt gegenereerd uit de signatuur.
- In verband met bijvoorbeeld common knowledge zou het wenselijk zijn indien WALT kon vaststellen welke objecten t.g.v. hiding voor de gebruiker onzichtbaar zijn. Dit is echter vermoedelijk uiterst lastig te implementeren.

5.2 De verdere ontwikkeling

De vragen die we later nog kunnen tegenkomen zijn natuurlijk moeilijker:

- Kunnen we het *ontwikkelen* van klassen en topologieën, d.w.z. het ontwerpen van objecten, integreren binnen het systeem? Dit proces vindt natuurlijk niet plaats in dezelfde ‘ruimte’ als de simulatie. Als we objecten die nog niet ‘af’ zijn al willen displayen, komt hier onder meer het gebruik van defaults om de hoek kijken. Merk op dat het systeem tijdens zo’n interactie zijn alwetendheid verliest. Overigens is dat wel een bijzonder leuk aspect, omdat het betekent dat asserties niet meer als vragen dienen te worden opgevat, maar kennis aan het systeem kunnen toevoegen.
- In hoeverre ondersteunen we referenties aan de toekomst? En als we dit doen, gaan we dan ook ondersteuning geven voor counterfactuals?
- Modale kennisrepresentatie: We dienen niet alleen objectieve kennis te representeren maar moeten uiteindelijk ook in staat zijn attitudes van agenten t.a.v. semantische inhoud te representeren. Op die manier kunnen we namelijk de starre gedragscode m.b.v. de drie soorten uitingen doorbreken, uitgaande van de kennis over de doelen en verwachtingen van de gebruiker. Het gedrag van het systeem komt dan voort uit algemene principes, zoals: Tracht de verwachtingen van de gebruiker te vervullen of, indien dit onmogelijk is, te corrigeren. Dit veronderstelt dus de mogelijkheid van *modale* kennisrepresentatie. Het door Tijn Borghuis (project 7a) gepresenteerde voorstel om te werken met subcontexten, en bijbehorende ‘filters’ die het transport van informatie tussen deze contexten regelen, leidt o.i. tot een eenvoudige en flexibele architectuur om dit mogelijk te maken. De precieze uitwerking van zo een methode binnen het systeemontwerp is echter vooralsnog niet duidelijk.
- Focusering en Topicalisatie: Een bevredigende behandelingen van deze onderwerpen zou van groot belang kunnen zijn voor het door de gebruiker ervaren bedieningsgemak. Er is echter nog niet over nagedacht.
- Omdat de gebruiker verwacht dat hij door het systeem wordt begrepen, kan ook het generen van wedervragen t.b.v. het oplossen van ambiguïteiten e.d. worden gereduceerd tot het vervullen van verwachtingen.

gen. Om dit goed te kunnen doen, moet het systeem voor modale kennisverwerking natuurlijk al redelijk functioneren.

6 De geprojecteerde gebruikersgroep en gebruiksomgeving

Voordat we iets zeggen over de gebruikersgroep en gebruiksomgeving is het belangrijk het primaire doel in herinnering te brengen. Dit doel is de ontwikkeling is van een coöperatieve multimodale interface, hetgeen inhoudt dat het *specifieke* domein waarop de interface wordt toegepast geen wezenlijke rol speelt. Het betekent echter niet dat het domein in het gehele project een ondergeschikte rol zou spelen; integendeel, om de interface volwaardig te kunnen uittesten is de toepassing op een specifiek domein onontbeerlijk.

In het voorgestelde ontwerp wordt de representatie van het gespreksdomein door WALT verzorgd. De keuze van WALT als applicatie voor onze interface heeft een aantal belangrijke voordelen:

- WALT is een object-georiënteerd systeem. Dit vergemakkelijkt de uitwisselbaarheid met andere object-georiënteerde domeinen aanzienlijk, zodat de uniformiteit van de interface beter gegarandeerd kan worden.
- WALT bezit grafische eigenschappen. Dit heeft als voordeel dat verschillende communicatieve modaliteiten uitgetest en gecombineerd kunnen worden.
- WALT bevat spatiële en temporele aspecten. Het te ontwikkelen systeem biedt dus in potentie de mogelijkheid om een grote diversiteit aan semantische verschijnselen in natuurlijke taal te modelleren en deze modellering uit te testen (bijv. temporele verschijnselen in werkwoorden, voorzetsels, aanwijzende voornaamwoorden, etc.).

Het totale systeem (d.i. WALT gecombineerd met WATSON) zal ons inziens in eerste instantie dan ook gebruikt moeten worden als testcase voor fundamenteel onderzoek. Nadat de door ons gelegde basis klaar is kunnen de resultaten van de deelprojecten geleidelijk geïncorporeerd worden, wat inhoudt dat modelleringen die gemaakt zijn in de deelprojecten uitgetest kunnen worden binnen het systeem. De modulaire opbouw maakt het hierbij mogelijk dat de onderzoekers vanuit hun eigen expertise kunnen blijven werken aan verschillende aspecten van het systeem.

Het systeem kan ook gezien worden als belangrijke kapstok voor verder onderzoek. Alle betrokken groepen kunnen vanuit hun eigen interesse en expertise, in samenwerking met het integratieproject, projecten definiëren

waarvan de resultaten directe toepassing zouden kunnen vinden binnen het systeem. Een verdere ontwikkeling van het systeem kan derhalve een blijvende samenwerking garanderen tussen de deelnemende onderzoeksgroepen en vormt daarbij tevens een belangrijk platform voor de uitwisseling en toepassing van elkaars expertise.

Vanwege de modulariteit van het gehele systeem en de uniforme eigenschappen van WALT (zie ook 4.1) kan op langere termijn ook gedacht worden aan andere toepassingen dan WALT. WATSON zal immers een interactie kunnen ondersteunen met een object-georiënteerde applicatie die de volgende diensten aanbiedt:

- Methods voor de *creatie* van objecten.
- Methods voor de *transformatie* van de toestand van objecten.
- Methods voor de *observatie* van toestandsvariabelen van objecten.
- Mogelijkheden voor het aangeven van een selectie (op bijv. een beeldscherm).
- Mogelijkheden voor het uitvoeren van een selectie (bijv. met behulp van een muis)

Een voorbeeld van een systeem met dergelijke eigenschappen is het operating system van de Macintosh.

WATSON kan dus in principe dienen als uniforme interactie voor expertsystemen, leer- of ontwerpomgevingen. Specifieker kan gedacht worden aan een uniforme helpfaciliteit die gebruikt kan worden voor bijvoorbeeld applicaties met de bovengenoemde eigenschappen. De ontwerper van zo'n applicatie hoeft zich dan niet te concentreren op de interactieve eigenschappen van zijn systeem, maar dient de aandacht te richten op de specifieke eigenschappen van het domein. Een coöperatieve assistent bij de interactie kan dan bij zo een domein automatisch gegenereerd worden en hoeft dus niet apart te worden gemaakt.

Alhoewel verleidelijk, vinden we het prematuur om over de bovenstaande applicatie verdere speculaties te doen en verschuiven we daarom de uitwerking hiervan liever naar een latere datum. Gezien het algemene karakter van het domein en het rudimentaire stadium van het project op dit moment, lijkt het ons niet haalbaar om een commerciële versie te ontwikkelen binnen de nu nog beschikbare zes jaar.

7 Condities voor het welslagen van het project

Voor het welslagen van het project hebben we de volgende punten in gedachten.

1. *Ruime belangstelling van de uitvoerders t.a.v. elkaars projecten en het integratieproject.*

We willen hier niet de vrijheid van de onderzoekers inperken; we achten het echter onvermijdelijk dat de uitvoerders zich openstellen voor datgene wat zich in andere projecten afspeelt en actief meedenken over de problematiek bij de ontwikkeling van het door ons voorgestelde systeem.

2. *Goede informatie-uitwisseling over de deelprojecten tussen de verschillende uitvoerders.*

Een multidisciplinair project als dit brengt onherroepelijk communicatieproblemen met zich mee. Het creëren van de mogelijkheden om van elkaars werk op de hoogte te raken is o.i. dan ook van essentieel belang voor het welslagen van het project. We willen dan ook het volgende voorstellen om dit te verwezenlijken:

- Er komt regelmatige rapportage vanuit de deelprojecten naar het integratieproject zonder voortdurende initiatie vanuit de laatstgenoemde. Het is hierbij niet de bedoeling regelmatige voortgangsverslagen in te leveren, maar het gaat voornamelijk om (voor-)publicaties, aantekeningen, stage- of afstudeerverslagen, etc. Dit is alleen bedoeld om vanuit het integratieproject maximaal te profiteren van de resultaten van de deelprojecten en moet zeker niet gezien worden als controlemiddel. We kunnen dan vanuit het integratieproject bekijken of de inhoud zinvol opgenomen kan worden in het te ontwikkelen systeem.
- Om de twee maanden wordt een voordracht gehouden door een van de uitvoerders van een deelproject. Hierbij is het de bedoeling dat iemand vertelt over het werk dat hij het afgelopen half jaar heeft gedaan. Het verhaal hoeft niet aan te sluiten op het integratieproject, hoewel enige gedachtengang hierover natuurlijk altijd welkom is. Het is verder niet de bedoeling de voordrachten te vervangen door andere colloquia e.d., omdat de erop volgende discussie specifiek gericht kan zijn op het gehele DenK project.

De bijeenkomsten zijn toegankelijk voor alle leden van de werkgroep.

- Om iedereen op de hoogte te brengen van de werkzaamheden binnen het integratieproject, zal ook van ons regelmatige terugkoppeling komen in de vorm van mondelinge en schriftelijke verslaggeving. We streven ernaar ieder half jaar een rapport uit te brengen met tussentijdse 'draft-versies'. De werkgroepleden worden uitgenodigd hierop zoveel mogelijk commentaar te leveren. Dit rapport is een eerste voorbeeld van een half-jaarlijks verslag; een draft-versie is reeds uitgekomen in november.

3. *Gebruikmaking van gegevens uit praktijksituaties.*

We hebben voorlopig wel een idee welke aspecten belangrijk zijn voor de ontwikkeling van het systeem, maar uiteindelijk is het van cruciaal belang dat het systeem aansluit op datgene wat mensen daadwerkelijk doen in praktijk. Wordt in de voorgestelde configuratie bijvoorbeeld veel gebruik gemaakt van imperatieven? Wat is wel of geen relevante feed-back? Welke modaliteit prefereert men voor welke informatie-overdracht? Het zal duidelijk zijn dat er nog veel meer vragen te beantwoorden zijn en dat hierover pas zekerheid kan bestaan indien systematisch onderzoek verricht is naar praktijksituaties. We achten het dan ook noodzakelijk dat zo snel mogelijk gestart wordt met de uitvoering van project 5.

4. *Ondersteuning bij de implementatie van het voorgestelde prototype.*

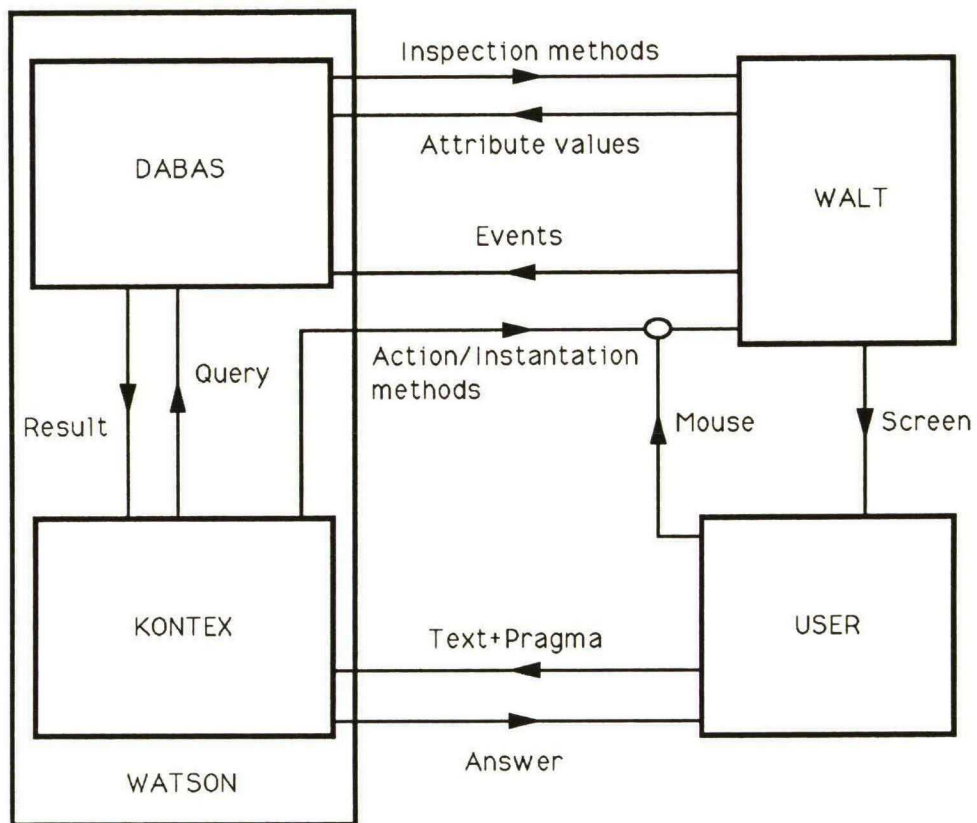
Alhoewel we nu nog in de ontwerpfasen zitten, zal het duidelijk zijn dat in de toekomst voorzien moet zijn in voldoende mankracht en middelen ter ondersteuning van het implementatie-activiteiten. Wij kunnen de implementatie in eerste instantie wel op gang zetten, maar we achten het niet wenselijk dit op de lange duur alleen door ons voort te laten zetten.

5. Als laatste punt willen we iedereen uitnodigen na te denken over een geschikte naam voor het systeem, zodat we in alle verdere stukken hiernaar kunnen verwijzen zonder steeds in algemene termen over 'het systeem' te hoeven spreken. We hebben zelf gedacht aan zoiets als COMIK (CoOperative Multimodal Interface for Knowledge analyses and transfer), maar we zijn er niet zo kapot van. Betere ideeën zijn van harte welkom.

8 Referenties

- Allwood, J. (1976) Linguistic communication as action and cooperation. *Gothenburg Monographs in Linguistics 1*.
- Ahn, Beun, Dols, Jaspars & de Reus (1990) Voortgang en Integratie van het Onderzoekswaartepunt Dialoogvoering en Kennisrepresentatie. Samenwerkingsorgaan Brabantse Universiteiten.
- Ahn, R. & Kolb, H.P. (1990) Discourse Representation meets Constructive Mathematics. *ITK Research Report, No. 16*. Katholieke Universiteit Tilburg.
- Beun, R.J., Dols, F.J.H., Jaspars, J.O.M. & de Reus, D. (1990) Terugblik en Vooruitzicht. DenK-rapport, Samenwerkingsorgaan Brabantse Universiteiten.
- Beun, R.J. (1989) *The recognition of declarative questions in information dialogues*. Proefschrift. Tilburg: Katholieke Universiteit Tilburg.
- Bunt, H.C., Beun, R.J., Dols, F.J.H., Linden, J.A. van der, & Schwarzenberg, G.O. thoe (1984) The TENDUM dialogue system and its theoretical basis. *IPO Annual Progress Report, 19*. 105-113.
- Curtis, B. & Malone, T. (1988) Introduction to this special issue on computer-supported cooperative work. *Human Computer Interaction, Vol. 3*.
- van Nes, F.L. (1990) *Boeken, Computers en Ergomomen*. Intreerede. Technische Universiteit Eindhoven.
- van Overveld, C.W.A.M. (1990) *The generalized display processor as an approach to real-time interactive 3-d computer animation*. Technical Report. Technische Universiteit Eindhoven.
- Programma-beschrijving 1989-1997 (1989) Dialoogvoering en Kennisopbouw: Onderzoekswaartepunt bij het Samenwerkingsorgaan Brabantse Universiteiten. ITK/IPO/Wsk&Inf TUE.
- Taylor, M.M., Néel, F., & Bouwhuis, D.G. (Eds.) (1989) *The structure of multimodal dialogues*. Amsterdam: North Holland.

**A De modules van het systeem en hun onderlinge
samenhang**



B Lijst van gebruikte woorden

DABAS : knowledge base voor directe objectieve kennis over het gespreksdomein.

ENTITEITEN : verzamelnaam voor dingen die in gesprek over een domein kunnen voorkomen, zoals objecten, tijdstippen, methods, classes, etc.

EOL : naam voor specifieke eerste orde logica (getypeerd) die wordt gebruikt in DABAS.

EVENT : combinatie van tijdstip en method.

IF : interpretatiefunctie; deze beeldt variabelen in gemeenschappelijke context af op namen van entiteiten in DABAS.

PRAGMATISCHE FUNCTIE : de taalhandelingen die uitgedrukt wordt met de tekst, hier zijn er drie van: ASSERTION, QUESTION, INSTRUCTION.

KLASSE : soort van type in object-georiënteerd programmeren;

KONTEX : module waarin kennis over de gebruiker en het verloop van de dialoog is opgeslagen.

METHOD : specificaties van gedrag die elementen uit een klasse op verzoek kunnen vertonen; hier zijn er drie van:

ACTION : verzoek om gepredefiniëerde mogelijke vorm van gedrag voor elementen uit een bepaalde klasse.

INSPECTION : verzoek om terugmelding van de attribuutwaarde van een object.

INSTANTIATION : verzoek tot creatie van object binnen een bepaalde klasse met gegeven topologie.

OBJECTEN : dat zijn de dingen die in WALT zijn gerepresenteerd, die kunnen veranderen, etc.

TEXT : een fragment van een context in type theorie.

TOPOLOGIE : specificatie van ruimtelijke informatie i.v.m. een object.

USER : de gebruiker.

WALT : systeem dat een domein simuleert op grond van object-georiënteerde beschrijving van dat domein.

WATSON : de coöperatieve assistent, bestaande uit KONTEX en DABAS.

Bibliotheek K. U. Brabant



17 000 01173165 1