

Artikel 6

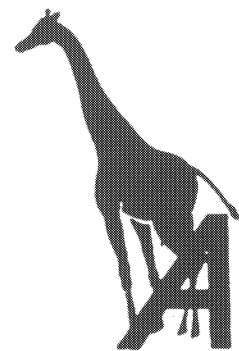
GIRAF

Een information retrieval systeem
voor universiteiten, hogescholen en
andere pluriforme organisaties

1984

*Het herkennen van de cultuur en het weten
hoe daarmee om te gaan zijn belangrijk om
te voorkomen dat de cultuur zich als een
virus gaat gedragen in de ontwikkelde
informatiesystemen.*

Mentink



Dit artikel is eerder verschenen in:

Informatie, mei 1984, jaargang 26 nr. 6 pag. 445 t/m 455

GIRAF

EEN INFORMATION RETRIEVAL SYSTEEM VOOR UNIVERSITEITEN, HOGESCHOLEN EN ANDERE PLURIFORME ORGANISATIES

door ir. A. J. van Dijk

In 1970 is bij de Technische Hogeschool Delft het (batch-)systeem 'BIBLIOSYSTEM' ontwikkeld. Over dit systeem is gepubliceerd in het blad 'Informatie' van oktober 1971. Hoewel het systeem bedoeld was om het samenstellen van bibliografieën met behulp van de computer sneller en prettiger te laten verlopen, is het in de periode 1970-1984 zowel binnen als buiten de TH-Delft ook gebruikt voor het bewerken van allerlei bestanden zoals adresbestanden, ledenbestanden, apparaatbestanden, diabeestanden, boekbestanden, bestanden met gegevens van congresgangers, etc. In 1976 ontwikkelde het Rekencentrum van de TH-Delft het on-line retrieval systeem 'BIBINFO'. Dit systeem sloot aan op 'BIBLIOSYSTEM'. De voortdurende vraag binnen de TH-Delft naar dergelijke faciliteiten heeft er toe geleid dat het, naar de normen van 1984, gebruiksvriendelijke information retrieval systeem 'GIRAF' is ontwikkeld. Bij de realisatie van 'GIRAF' heeft het database-management- en data-communicatie-systeem IDMS DB/DC als draagsysteem gefungeerd. Bovendien werd gebruik gemaakt van het ontwikkeltool 'Application Development System (ADS/ONLINE)' dat, evenals IDMS DB/DC, door de firma Cullinet Software geleverd wordt. Dit artikel behandelt de functionele aspecten van 'GIRAF'.

1 INLEIDING

In 1970 is bij de Technische Hogeschool Delft (kortweg TH-Delft) het systeem 'BIBLIOSYSTEM' [3] ontwikkeld. Aanleiding voor de bouw van dit systeem was de grote hoeveelheid tijd, die onderzoekers moesten besteden aan het verzamelen, registreren en ordenen van literatuur. Het bestuderen van literatuur kreeg nl. een steeds belangrijkere plaats in het onderzoek en het samenstellen van bibliografieën en literatuurlijsten werd onderdeel van vrijwel ieder researchproject van enige betekenis. Een grote flexibiliteit in het bijwerken en classificeren bleek daarbij van grote waarde.

De klassieke wijze van rapporteren van een literatuurstudie was zeer tijdrovend o.a. vanwege de omvang van het typewerk. Dit typewerk vormde een ware bottleneck. Bibliografieën waren soms op het moment dat ze gereed waren voor verspreiding alweer verouderd. Een en ander leidde tot het idee om het samenstellen van bibliografieën te automatiseren. 'BIBLIOSYSTEM' werd begin 1971 door de TH-Delft in gebruik genomen. Binnen de TH-Delft bleek het in een grote behoefte te voorzien. Maar niet alleen binnen de TH-Delft was men geïnteresseerd; na een publikatie in het blad 'Informatie' (oktober 1971) [3] kreeg de TH-Delft meer dan honderd verzoeken om de systeembeschrijving en programma's beschikbaar te stellen. In de loop der jaren bleek bovendien, dat door de 'abstracte' opzet het systeem bruikbaar was voor allerlei toepassingen. Naast bestanden met literatuurreferenties (zie figuur 2) werden met behulp van 'BIBLIOSYSTEM' adresbestanden, bestanden voor ledenadministraties, apparaatbestanden, diabeestanden, boekbestanden, etc. vervaardigd, bijgewerkt en afgedrukt.

In 1975 werd het Rekencentrum geconfronteerd met het

verzoek van een congresorganisatiecomité om tijdens een te organiseren congres in de zomer van 1976 een real-time congresinformatiesysteem beschikbaar te stellen. Dit leidde tot de ontwikkeling van 'BIBINFO' [4]. Dit systeem werd als het ware bovenop 'BIBLIOSYSTEM' gebouwd. Het opbouwen van de bestanden gebeurde met 'BIBLIOSYSTEM'. Het raadplegen en het on-line selecteren gebeurde met 'BIBINFO'. In 1976 was het systeem voor de congresorganisatie beschikbaar. In de periode 1976-1983 is 'BIBINFO' regelmatig gebruikt door de 'BIBLIOSYSTEM'-gebruikers en door congresorganisaties.

'BIBLIOSYSTEM' (1970) en 'BIBINFO' (1976) zijn ontwikkeld voor een beperkte doelgroep. De systemen zijn ontwikkeld met methoden en hulpmiddelen, die in die jaren modern waren. Het grote succes van deze systemen binnen de TH-Delft en de voortdurende vraag naar dergelijke faciliteiten heeft de directie van het Rekencentrum in 1982 doen besluiten een nieuw information retrieval systeem te laten ontwikkelen, dat op een ruime doelgroep moest worden gericht. Het moest de systemen 'BIBLIOSYSTEM' en 'BIBINFO' vervangen en zeer gebruiksvriendelijk zijn. Dit nieuwe systeem is het *programmatuursysteem* 'GIRAF' (General Information Retrieval Facilities) geworden.

In dit artikel zullen de *functionele* aspecten van 'GIRAF' worden behandeld. De technische aspecten komen slechts zeer zijdelings ter sprake. In een volgend artikel zal worden ingegaan op enkele technische aspecten van 'GIRAF'. Bovendien zal in dat artikel worden ingegaan op de ervaringen opgedaan met conversationale programmering ('BIBINFO' is volledig conversationeel geprogrammeerd), pseudo-conversationele programmering [5] en programmering met behulp van een 'ontwikkeltool' ('GIRAF' is ontwikkeld met behulp van het ont-

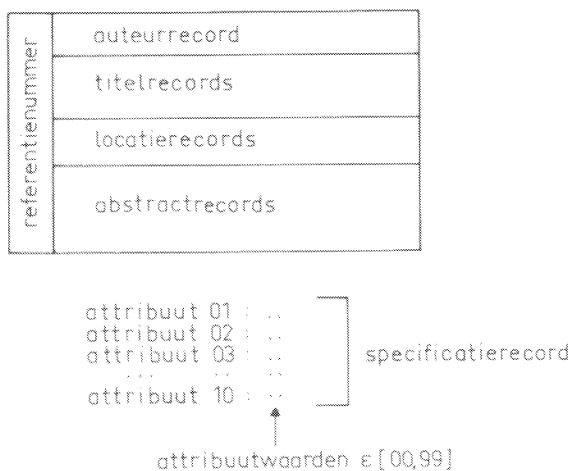
wikkeltool 'Application Development System (ADS/ ONLINE)' van de firma Cullinet Software [1, 2]).

2 UITGANGSPUNTEN 'GIRAF'

Uit het succes van de systemen 'BIBLIOSYSTEM' en 'BIBINFO' is de conclusie getrokken, dat de filosofie, waarop deze systemen waren gebaseerd, voor de TH-Delft een goede filosofie is. De eerste voorwaarde, die aan 'GIRAF' is gesteld, is het handhaven van deze filosofie. Dit betekende o.a., dat 'GIRAF' een eenvoudig en doeltreffend information retrieval systeem moest worden (*eis 1*).

De gegevenseenheid in de oude systemen was de referentie (zie figuren 1 en 2). Toegang tot een *individuele*

BIBLIOSYSTEM (1970)



Figuur 1: Gegevensmodel van een referentie binnen 'BIBLIOSYSTEM' (1970)

referentie geschiedde via het referentienummer of de auteur. Toegang tot een *verzameling* referenties was alleen mogelijk via een codesysteem. Dit codesysteem bestond uit een tiental attributen met voor ieder attribuut een waardenverzameling. Aan iedere referentie kon per

attribuut een waarde uit de betreffende waardenverzameling worden toegekend. Op deze manier kon een gebruiker aan een referentie kenmerken en kenmerkwaarden toekennen. Aan een boekreferentie kon bijvoorbeeld in attribuutnummer 2 de waarde 38 en in attribuutnummer 8 de waarde 17 worden toegekend (zie figuur 2). Een en ander betekende dat het boek in de Engelse taal was geschreven en door een bepaalde uitgever was uitgegeven. Het selecteren van een deelverzameling was te realiseren door de attributen, waarin men voor de betreffende selectie was geïnteresseerd, op te geven. Per attribuut werd de waardenverzameling, die voor deze selectie geldig was, opgegeven. Met behulp van EN- en OF-relaties op attribuutniveau was het mogelijk de gewenste deelverzameling te construeren.

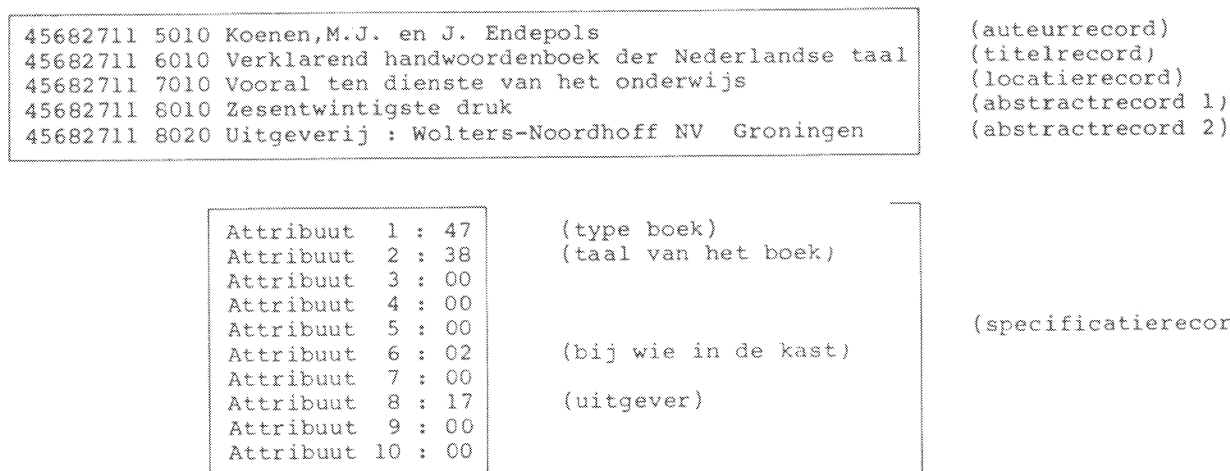
Aangezien 'GIRAF' voor een ruimere doelgroep moest worden gemaakt, moest het meer algemeen bruikbaar zijn dan de oude systemen (*eis 2*). Daaruit werd, mede naar aanleiding van verzoeken van gebruikers, de conclusie getrokken, dat het codesysteem ruimer zou moeten zijn (*eis 3*).

De TH-Delft is een pluriforme organisatie, die veel zelfstandig opererende groeperingen herbergt. Sommige van die groeperingen willen (delen van) hun informatievoorziening op een geïsoleerde manier realiseren (bijv. promovendi) (*eis 4*). Andere groeperingen willen graag samenwerken met groeperingen binnen of buiten de TH-Delft. Voor die groeperingen is het noodzakelijk om vanuit verschillende locaties, met behulp van real-timeverwerking, bestanden te kunnen opbouwen, raadplegen en/of muteren (*eis 5*).

'GIRAF' moet kunnen worden toegepast ten behoeve van veel informatiesystemen binnen de TH-Delft. Daarom moet 'GIRAF' worden aangeboden als *confectiesysteem*. De opzet moest echter dusdanig zijn, dat het met weinig inspanning kan worden aangepast. Op die manier kan 'GIRAF' voor sommige gebruikers als *maatconfectiesysteem* fungeren (*eis 6*).

Bij de TH-Delft werken en studeren ook buitenlanders. Het is daarom wenselijk dat 'GIRAF' kan werken met andere talen. Het systeem moest zodanig worden opgezet, dat in een later stadium *andere taalversies* snel kunnen worden geïmplementeerd (*eis 7*).

Het Rekencentrum van de TH-Delft is een dienstverlenende instantie. Zoals het een goede dienstverlenende



Figuur 2: Voorbeeld van een 'BIBLIOSYSTEM'-referentie

ment kan de TBB zijn GTD beheren. Dit betekent o.a., dat hij wachtwoorden op drie toegangsniveaus kan vaststellen. Het is de bedoeling, dat uitsluitend de TBB toegang heeft via toegangsniveau 1 (bijv. het wachtwoord 'ALLEENIK'). Geautoriseerde medewerkers hebben toegang via toegangsniveau 2 (bijv. het wachtwoord 'OOKANDER'). Toegangsniveau 3 kan worden vrijgegeven voor het raadplegen van de gegevens (bijv. het wachtwoord 'IEDEREEN'). Met behulp van functie 6, die alleen beschikbaar wordt gesteld, als een gebruiker via een wachtwoord van toegangsniveau 1 toelating tot de GTD heeft verkregen, kan de TBB nog meer maatregelen treffen. In 4.4.7 wordt hierop nader ingegaan.

Door systeemkenmerk en wachtwoorden niet vrij te geven kan een gebruiker geïsoleerd werken. Samenwerking tussen groeperingen is ook mogelijk. Zo kunnen bijvoorbeeld vakgroepbibliotheken samenwerken door wachtwoorden van toegangsniveau 3 uit te wisselen. De TBB houdt het heft in eigen hand, want hij kan op elk moment de wachtwoorden wijzigen. Op deze manier kan iedere TBB de muren om zijn GTD net zo hoog (of laag) maken, als hij dat wenst (zie figuur 7).

De overige functies zullen nu in het kort worden behandeld. Opgemerkt moet worden, dat functie 6 alleen beschikbaar is onder toegangsniveau 1 en de functies 4 en 5 alleen onder de toegangsniveaus 1 en 2. Het muteren in functie 2 en het verwijderen in functie 3 zijn eveneens alleen toegankelijk onder de toegangsniveaus 1 en 2.

4.4.2 *Helpfunctie (functie 1)*

De handleiding met betrekking tot het gebruik van 'GIRAF' is via deze functie beschikbaar. De gebruiker kan de gewenste hoofdstukken selecteren. TBB's, die over een eigen gebruikersafhankelijk subsysteem beschikken, kunnen een eigen hoofdstuk aan de handleiding toevoegen.

4.4.3 *Inbrengen van nieuwe referenties (functie 5)*

Het inbrengen van nieuwe referenties kan alleen gebeuren door de ToepassingsBestandsBeheerder (TBB) (toegangsniveau 1) of door geautoriseerde medewerkers (toegangsniveau 2). Er is veel aandacht besteed aan de gebruiksvriendelijkheid van deze functie. Allereerst kan de gebruiker kiezen tussen een serie schermbeelden voor zogenaamde standaardreferenties of een serie schermbeelden voor zeer uitgebreide referenties. De meeste gebruikers zullen kunnen werken met de serie schermbeelden voor standaardreferenties. Op alle schermbeelden verschijnen bij de gegevensgroepen de door de TBB gekozen namen. Op het standaardschermbeeld zijn 12 invoerregels beschikbaar, voor iedere gegevensgroep 3 regels. Dus bijvoorbeeld 3 regels voor namen, 3 regels voor de titel, 3 regels voor de locatie en 3 regels voor de abstract. Vanzelfsprekend kan er worden gewerkt met vervolgschermbeelden. Indien er echter wordt gewerkt met bijvoorbeeld een adressenbestand, is het niet zinvol om 3 regels te reserveren voor de naam. Het iedere keer overslaan van 2 regels wordt bovendien op den duur vervelend. Om dit probleem op te lossen kan een TBB via functie 6 (wijzigen bestandsgegevens) voor iedere gegevensgroep opgeven wat het gewenste aantal regels op het betreffende invoerschermbeeld is. Bovendien kan hij voor het invoerschermbeeld een initialiseringskarakter definiëren (bijvoorbeeld een '.' of een '-'). Indien dit initialiseringskarakter de letter 'H' ('H' van herhalen) is,

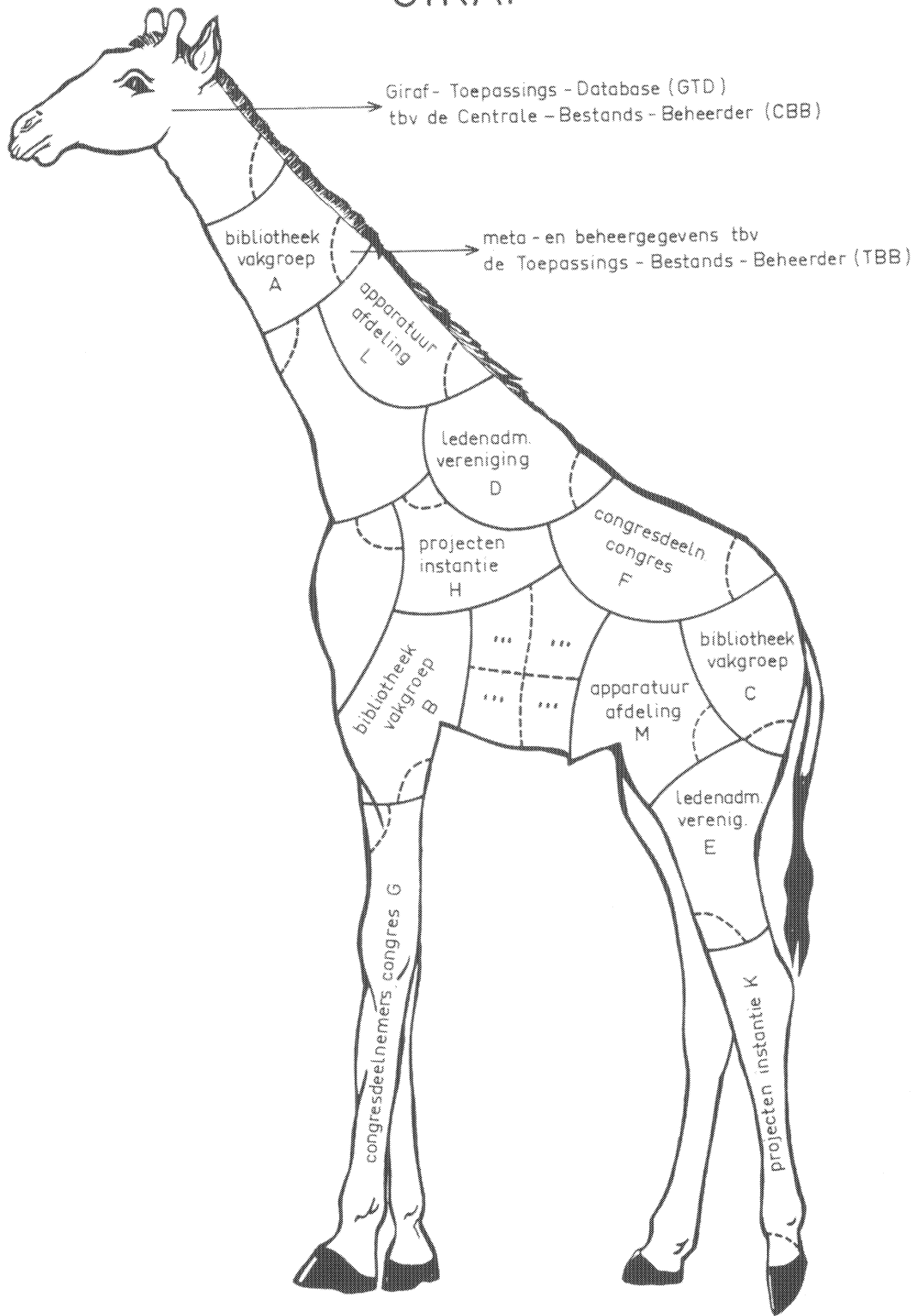
zullen, nadat de gegevens in de database zijn opgenomen, de gegevens op de invoerschermbeelden blijven staan. Op die manier kunnen veel gelijksoortige gegevens snel worden ingevoerd. In literatuurbestanden wordt vaak gewerkt met een jaartal (+ suffix) van publicatie. Een TBB kan definiëren of hij wel of niet met een jaartal + suffix wil werken. Een TBB kan op elk moment (on-line) wijzigingen aanbrengen. Aangezien de invoerschermbeelden op uitvoeringstijd worden opgebouwd, hebben wijzigingen onmiddellijk effect. Met behulp van bovengenoemde parameters kan een TBB zijn 'eigen' invoerschermbeelden definiëren. Indien men dan ook tien willekeurige 'GIRAF'-gebruikers referenties ziet invoeren, kan men tien 'eigen' series invoerschermbeelden zien (zie figuur 8). Een andere mogelijkheid bij het invoeren van gegevens is het eerst kopiëren van de gegevens van een reeds bestaande referentie. Na aanpassing kunnen de gegevens van de nieuwe referentie in de GIRAF-Toepassings-Database (GTD) worden opgenomen.

Sommige gebruikers willen de gegevens, die tot een gegevensgroep behoren, via een vaste, specifieke indeling inbrengen. Een hulpmiddel daarbij wordt geboden door de mogelijkheid als referentie een dummy-referentie met een initialiseringsmasker te definiëren. Voordat men nu de nieuwe referentiegegevens inbrengt, kan de dummy-referentie met het masker worden gekopieerd op de invoerschermbeelden.

4.4.4 *Ophalen/muteren van een individuele referentie (functie 2)*

Het ophalen van de gegevens van een individuele referentie kan onder elk toegangsniveau plaatsvinden. Via functie 6 (wijzigen van bestandsgegevens) kan de ToepassingsBestandsBeheerder (TBB) vastleggen welke gegevensgroepen onder toegangsniveau 1, respectievelijk toegangsniveau 2 en toegangsniveau 3 beschikbaar worden gesteld. Het ligt voor de hand, dat met name onder toegangsniveau 3 niet alle gegevens van een referentie ter beschikking komen van de betreffende gebruiker. Dit is bijvoorbeeld het geval, als in de gegevensgroepen 1, 2 en 3 gegevens worden vastgelegd, die door veel gebruikers mogen worden geraadpleegd, terwijl gegevensgroep 4 gegevens bevat, die alleen voor een beperkte groep gebruikers toegankelijk zijn. Het ophalen van de gegevens van een individuele referentie kan plaatsvinden op basis van een *referentiernaam* uit gegevensgroep 1 of het *referentienummer*. Een referentienummer is uniek en het resultaat is dan ook het beschikbaar komen van de referentiegegevens. Met een referentiernaam zijn er wat meer mogelijkheden. Enerzijds kunnen aan een referentie verschillende namen toegekend zijn, bijvoorbeeld bij tijdschriftartikelen, anderzijds kan een naam aan verschillende referenties zijn verbonden, omdat een auteur verschillende tijdschriftartikelen heeft geschreven. In het laatste geval zal van alle referenties, waaraan de betreffende naam is verbonden de eerste regel van gegevensgroep 2 op het betreffende schermbeeld verschijnen (bijvoorbeeld de eerste regel van de titel), zodat de gebruiker de gewenste referentie kan kiezen. Wanneer als referentiernaam een niet in de GIRAF-Toepassings-Database (GTD) aanwezige naam wordt opgegeven (of een *generieke naam*), wordt een schermbeeld geproduceerd met een lijst van in de GTD aanwezige referentiename

GIRAF



Figuur 7: Structuur van de 'GIRAF'-database

die in de (alfabetisch geordende) omgeving van de ingevoerde naam liggen.

Het *wijzigen* van gegevens is alleen toegestaan onder de toegangsniveaus 1 en 2. De wijzigingen worden direct in de GTD aangebracht.

4.4.5 *Selecteren/verwijderen van referenties (functie 3)*

Het on-line selecteren van een verzameling referenties kan alleen plaatsvinden met behulp van het codesysteem. In 3 is reeds vermeld, dat het codesysteem bij 'GIRAF' bestaat uit een (in principe onbeperkt) aantal attribuutcodes en per attribuutcode een attribuutwaardenverzameling. Iedere gebruiker bepaalt zelf een aantal attribuutcodes en de daarbij behorende attribuutwaardenverzamelingen.

Bij het invoeren van een referentie kunnen voor de betreffende attribuutcodes één of meer attribuutwaarden aan de referentie worden toegekend (zie figuren 3 en 4). Voor een literatuurreferentie kunnen bijvoorbeeld worden gekozen attribuutcode 'CORE' (afkorting van Computer REviews: een literatuurclassificatiemethode) met bijbehorende attribuutwaarden 'G.2.2' en 'G.3.3' en attribuutcode 'KAST' met bijbehorende attribuutwaarde 'KLAAS'. Een en ander betekent, dat de betreffende literatuurreferentie handelt over 'physical design database management' en 'information search and retrieval' en dat deze te vinden is in de kast bij medewerker 'KLAAS'. De gebruiker kan nu selecties maken door het opgeven van zogenaamde *items* en *relaties* tussen de items. Een item bestaat in dit geval uit een attribuutcode met één of meer attribuutwaarden. De relaties zijn EN- en OF-relaties.

'GIRAF' biedt de mogelijkheid om selectiecriteria d.m.v. *selectieprofielen* in de GIRAF-Toepassings-Database (GTD) op te slaan (zie 4.4.7). Door het intoetsen van het gewenste profielnummer kan het selectieprofiel op het beeldscherm verschijnen. Na het eventueel muteren van de criteria kan de selectie worden uitgevoerd. 'BIBINFO' bood de mogelijkheid om te selecteren met behulp van *trefwoorden*. Ook 'GIRAF' kent die mogelijkheid. Een ToepassingsBestandsBeheerder (TBB) kan een verzameling trefwoorden definiëren. Bij ieder trefwoord behoort een attribuutcode + attribuutwaarde. Op deze manier kunnen gebruikers selecties maken met trefwoorden en EN- en OF-relaties tussen trefwoorden. Nadat de selectie heeft plaatsgevonden, kan de gebruiker kiezen, wat hij met de geselecteerde referenties wil doen. Een van de mogelijkheden is het bekijken van (een deel van) de gegevens op het beeldscherm. Een andere mogelijkheid is het afdrukken van de referenties op een door de gebruiker op te geven printer. Het aantal af te drukken referenties kan door de TBB worden beperkt (zie 4.4.7). De verzameling geselecteerde referenties kan ook op een achtergrondgeheugen worden geplaatst. Op die manier kunnen gebruikers een *extract* uit de database ter beschikking krijgen. Met behulp van de 'GIRAF'-rapportagemogelijkheden kunnen dan de diverse overzichten worden vervaardigd. Interessant is echter ook de mogelijkheid om met behulp van door de gebruiker zelf ontwikkelde programmatuur de extractgegevens te bewerken. Opgemerkt moet worden dat gebruikers in alle bovenstaande gevallen alleen die gegevens ter beschikking krijgen, waartoe zij volgens hun toegangsniveau toegang hebben. Het verwijderen uit de database

van de verzameling geselecteerde referenties behoort eveneens tot de mogelijkheden.

4.4.6 *Opstarten batch-verwerking (functie 4)*

Functie 4 stelt de gebruiker in staat om, als onderdeel van on-line activiteiten, sommige standaard batch-tools op te starten. Met behulp van deze batch-tools kunnen diverse overzichten worden vervaardigd zoals de gebruikshandleiding, gebruiksstatistieken en bestandsstatistieken.

4.4.7 *Wijzigen bestandsgegevens (metagegevens) (functie 6)*

De TBB heeft een functie ter beschikking, waarmee hij zijn GTD kan beheren en *metagegevens* kan wijzigen. De volgende subfuncties kunnen worden onderkend:

4.4.7.1 *Wachtwoorden en autorisatiekenmerken*

Voor de toegangsniveaus 1, 2 en 3 zijn wachtwoorden en autorisatiekenmerken vastgelegd. Een van de autorisatiekenmerken betreft het *retrievalniveau*. De ToepassingsBestandsBeheerder (TBB) kan bijvoorbeeld bij toegangsniveau 3 opgeven, dat alleen de gegevens van de gegevensgroepen 1, 2 en 3 toegankelijk zijn. Bij de toegangsniveaus 1 en 2 kan hij dan alle gegevens beschikbaar stellen. Er zijn in 'GIRAF' een aantal combinaties van gegevensgroepen en attribuutgegevens gedefinieerd met een daarbij behorend nummer. De TBB kan per toegangsniveau het gewenste combinatie nummer opgeven. Het opstarten van *batch-programma's* (zie 4.4.6) kan alleen plaatsvinden als de TBB het betreffende autorisatiekenmerk op 'JA' heeft gezet. Ook het *wijzigen van referenties* of het afdrukken van geselecteerde referenties kan alleen plaatsvinden, als de betreffende autorisatiekenmerken op 'JA' staan. De TBB kan bovendien per toegangsniveau een *limiet* stellen aan het aantal af te drukken referenties. Soms is het gewenst om de toegang tot een GIRAF-Toepassings-Database (GTD) alleen toe te staan vanaf één of meer vaste terminals. De TBB kan een dergelijke *terminalidentificatiebescherming* realiseren door per toegangsniveau de namen van de betreffende terminals vast te leggen. Op deze manier kan hij bijvoorbeeld toegangsniveau 1 alleen toestaan op de terminal, die bij de TBB op de kamer staat. Toegangsniveau 3 kan bijvoorbeeld geheel worden vrijgegeven of kan worden beperkt tot een aantal opgegeven terminals.

4.4.7.2 *Bestandsidentificatie en -informatie*

'GIRAF' bevat zo'n 75 schermbeelden (zie figuren 6 en 8). Alle schermbeelden hebben eenzelfde indeling. Zo bevat regel 1 o.a. altijd de naam van de instelling waar het systeem is geïmplementeerd.

Bij de TH-Delft staat vermeld: *** TECHNISCHE HOGESCHOOL DELFT ***. Regel 2 bevat o.a. de tekst 'GIRAF' gevolgd door de naam van de GTD waarmee wordt gewerkt. De TBB kan deze *bestandsidentificatie* vastleggen en naar believen wijzigen. Op die manier is op ieder schermbeeld te zien bij welke instelling en met welke GTD de gebruiker werkt. Naast de bestandsidentificatie kan de TBB *bestandsinformatie* vastleggen. Deze bestandsinformatie verschijnt tijdens de toegangprocedure op het scherm. De TBB kan op die manier een korte beschrijving vastleggen om gebruikers op bepaalde zaken te attenderen. De TBB kan bovendien het *lettertype* definiëren. Dit houdt in, dat indien de TBB wenst te wer-

ken met uitsluitend hoofdletters of uitsluitend kleine letters, hij dit kan vastleggen. 'GIRAF' zorgt er dan voor, dat alvorens gegevens in de database worden opgeslagen alle letters worden omgezet in hoofdletters respectievelijk in kleine letters. De verstekwaarde is hoofd- en kleine letters. In het laatste geval worden de gegevens opgeslagen, zoals ze door de gebruiker zijn ingevoerd. In bijzondere gevallen kan het gewenst zijn, dat gegevens in een *versluisde* vorm op het achtergrondgeheugen worden geplaatst. De TBB kan dit echter niet zelf realiseren. Hij moet hiervoor een gemotiveerd verzoek indienen. Als op het verzoek positief wordt gereageerd, zal de CBB de betreffende optie activeren. Een en ander is op die manier geregeld, omdat het versluieren van gegevens nogal wat computertijd vraagt. Het versluieren van gegevens zal dan ook worden beperkt tot de werkelijk noodzakelijke gevallen.

4.4.7.3 *Systeemboodschappen*

Niet alleen de regels 1 en 2 van alle schermbeelden hebben een vaste indeling. Ook regel 24 heeft een vaste indeling. Deze regel is namelijk gereserveerd voor systeemboodschappen. De TBB kan *systeemboodschappen* versturen. Hij heeft op die manier als het ware een 'intercom' mogelijkheid tot zijn beschikking. De TBB kan per functie een systeemboodschap vastleggen. Dit betekent dat ieder schermbeeld van een functie de betreffende systeemboodschap zal bevatten. Pas nadat de TBB de betreffende systeemboodschap heeft verwijderd zal deze niet meer verschijnen. Naast de systeemboodschappen per functie heeft de TBB de mogelijkheid om een algemene systeemboodschap te definiëren. Deze heeft altijd voorrang boven de functiegerichte systeemboodschappen. Een algemene systeemboodschap '*** ER IS KOFFIE MET GEBAK BIJ ... ***' zal op alle schermbeelden van de betreffende GTD verschijnen en vermoedelijk leiden tot grote drukte bij de betreffende persoon.

4.4.7.4 *Presentatie van de gegevensgroepen*

In 4.4.3 (inbrengen nieuwe referenties) is reeds ingegaan op de mogelijkheden voor de ToepassingsBestandsBeheerder (TBB) om per gegevensgroep een eigen naam vast te leggen. Ook het vastleggen van een initialiserings-teken voor de invoerschermbeelden en het aantal regels per gegevensgroep alsmede het al dan niet gebruiken van een jaartal + suffix bij het invoeren van referenties is daar reeds aan de orde geweest.

4.4.7.5 *Attribuutwaardeomschrijvingen, trefwoorden*

De TBB heeft de mogelijkheid om aan een attribuutcode + attribuutwaarde een omschrijving en één of meer trefwoorden te koppelen. De *attribuutwaardeomschrijving* speelt een rol bij functie 2, het ophalen van individuele referenties. Hoewel het codesysteem, met alfanumerieke attribuutcodes van 4 posities en alfanumerieke attribuutwaarden van 6 posities, reeds mogelijkheden biedt om suggestieve codes en waarden te kiezen, is het soms prettig om bij het opvragen van een referentie van een betreffende attribuutcode + attribuutwaarde een ruimere omschrijving op het beeldscherm te krijgen. De TBB heeft een mogelijkheid om deze attribuutwaardeomschrijving te koppelen aan een attribuutcode + attribuutwaarde. Een andere rol spelen de *trefwoorden*. Bij het selecteren van referenties willen sommige gebruikers

graag werken met trefwoorden (zie 4.4.5). De TBB kan één of meer trefwoorden (synoniemen) koppelen aan een attribuutcode + attribuutwaarde. Gebruikers kunnen nu selecties maken met behulp van de trefwoorden. 'GIRAF' zal intern de trefwoorden omzetten in combinaties van attribuutcode + attribuutwaarde.

4.4.7.6 *Selectieprofiel, uitvoerlayout, batch-verstekwaarden*

Met behulp van *selectieprofielen* kan een ToepassingsBestandsBeheerder (TBB) selectiecriteria vastleggen in de GIRAF-Toepassings-Database (GTD). Dit is vooral van belang bij regelmatig terugkerende selecties. Ook kan de TBB op die manier een aantal selecties als het ware klaarzetten voor gebruikers van zijn GTD. Naast selectieprofielen kan een TBB ook *profielen voor uitvoerlayouts* in de GTD vastleggen. Ten behoeve van het opstarten van batch-verwerking door gebruikers kan de TBB een aantal *verstekparameterwaarden* en andere gegevens, o.a. met betrekking tot de Job Control Language, vastleggen. De TBB kan bovendien de on-line beschikbare *gebruikershandleiding* uitbreiden met zijn 'eigen' hoofdstuk. Met behulp van deze mogelijkheden kan een TBB een gebruiksvriendelijk klimaat scheppen rondom zijn GTD.

4.4.8 *Gebruikersafhankelijke on-line functie (functie 7)*

Het gebruikersafhankelijke subsysteem (zie 4.3) biedt o.a. de mogelijkheid om voor een GTD een 'eigen' on-line functie te maken. Zo heeft de Centrale BestandsBeheerder (CBB) een eigen on-line functie. Sommige gebruikersafhankelijke on-line functies zullen worden gemaakt voor bepaalde *groepen* GTD's. Daarbij kan bijvoorbeeld gedacht worden aan een eigen on-line functie ten behoeve van vakgroepbibliotheken en apparatuurbeheerders, die behoefte hebben aan een eenvoudige uitleenadministratie. Een andere, voor de hand liggende, toepassing zou een eenvoudige ledenadministratie zijn ten behoeve van bijvoorbeeld het Muzisch Centrum en andere groeperingen, die al met de oude systemen hun ledenadministratie hebben geautomatiseerd.

5 GEBRUIKSVRIENDELIJKHEID

De opdracht was om een gebruiksvriendelijk systeem te bouwen. In de diverse paragrafen is reeds aandacht besteed aan de gebruiksvriendelijkheid van 'GIRAF'. Samenvattend kunnen de volgende punten onder de term gebruiksvriendelijk worden gerangschikt:

- Het *opstarten* van 'GIRAF' door een gebruiker is eenvoudig. Door het woord 'GIRAF' in te toetsen wordt hij geconfronteerd met de toegangsschermbeelden van 'GIRAF'. Gebruikers merken er in principe niets van dat 'GIRAF' werkt met het draagsysteem IDMS DB/DC.
- De gebruiker kan bij de *invoerfunctie* (zie 4.4.3) kiezen of hij wil werken met een serie invoerschermbeelden voor standaardreferenties of met een serie invoerschermbeelden voor uitgebreide referenties. Iedere ToepassingsBestandsBeheerder (TBB) kan zijn eigen naamgeving kiezen voor de gegevensgroepen. Tevens kan hij het aantal regels vastleggen voor de respectievelijke gegevensgroepen op de invoerschermbeelden. Door daarnaast een initialiserings-teken op te geven en te vermelden

of hij al dan niet wil werken met een jaartal (+ suffix) van publicatie heeft de TBB voldoende mogelijkheden om de invoerschermbeelden een eigen gezicht te geven en af te stemmen op zijn eigen werkomgeving. Het invoeren van referenties wordt gemakkelijk door het kunnen werken met een invoermasker en de mogelijkheid om een bestaande referentie te kopiëren en na mutatie in de database op te nemen.

- c. Alle 75 schermbeelden hebben een systematische indeling. De regels 1, 2 en 24 hebben een *uniforme indeling*, terwijl bij de indeling van de regels 3 t/m 23 is gestreefd naar een systematische indeling. Zo heeft bijvoorbeeld de optie 'VORIG NIVEAU' op alle schermbeelden, waar deze optie voorkomt, hetzelfde nummer. Dit voorkomt vergissingen. Met deze optie kan een gebruiker teruggaan naar een vorig schermbeeld. Dit is bijvoorbeeld interessant, als hij een te grove selectie heeft gemaakt en alsnog een verfijning wil aanbrengen. De reeds ingebrachte selectiecriteria blijven bij de 'vorige schermbeelden' bewaard en hoeven dus niet opnieuw te worden ingetoetst.
- d. Gebruikers van 'GIRAF' kunnen doorgaans van het ene schermbeeld naar het andere schermbeeld komen door het intoetsen van een optienummer. Ervaren gebruikers kennen veel schermbeelden en willen daarom vaak één of meer schermbeelden van een serie schermbeelden overslaan. 'GIRAF' bevat op een aantal punten de mogelijkheid van '*command chaining*', waarmee één en ander bewerkstelligd kan worden.
- e. Het *op maat* maken van 'GIRAF' door het vervangen van dialogen of door toevoeging van een gebruikersafhankelijk subsysteem is reeds aan de orde geweest in 4.3 en 4.4.8.
- f. Een gebruiker kan desgewenst geïsoleerd werken ('eenzaam maar niet alleen') of kan *samenwerken* met anderen binnen de TH-Delft of daarbuiten (zie figuur 7). Hij bepaalt grotendeels zelf hoever die samenwerking gaat.
- g. Door te werken met *selectieprofielen*, profielen voor uitvoerlayouts en verstekwaarden voor batchverwerking kan een TBB het zichzelf en zijn gebruikers gemakkelijk maken.
- h. Het op een later tijdstip implementeren van andere *taalversies* van 'GIRAF' is eenvoudig te realiseren. Een Engelstalige versie zal op korte termijn beschikbaar komen.
- i. Een *gebruikshandleiding* is zowel op papier als on-line beschikbaar. Iedere TBB kan desgewenst zijn eigen hoofdstuk toevoegen. Naast deze gebruikshandleiding is er nog een verkorte gebruikshandleiding. Deze laatste is bedoeld voor gebruikers die, zonder zich uitgebreid te willen verdiepen in alle mogelijkheden van 'GIRAF', snel met hun (eenvoudige) toepassing van start willen gaan.

6 PRIVACYASPECTEN

Binnen 'GIRAF' is op een aantal punten aandacht besteed aan privacyaspecten:

- a. De on-line toegang tot een GIRAF-Toepassings-Database (GTD) wordt geregeld met behulp van *systeemkenmerken* en *wachtwoorden*. De systeem-

kenmerken en wachtwoorden staan versluierd op het achtergrondgeheugen. Per toegangsniveau kan een ToepassingsBestandsBeheerder (TBB) de toegestane handelingen vastleggen en definiëren tot welke gegevens toegang wordt verleend. Bovendien kan een TBB met behulp van *terminalidentificatiebescherming* toegang tot zijn GTD koppelen aan bepaalde terminals. Zelfs kan hij deze terminals fysiek beveiligen door ze van een *slot* te laten voorzien.

- b. Toegang tot een GTD via batch-verwerking is beveiligd met behulp van wachtwoorden.
- c. Printuitvoer kan via een *lokale printer* worden afgeleverd.
- d. Het *versluieren* van gegevens op het achtergrondgeheugen behoort tot de mogelijkheden.
- e. Een TBB kan desgewenst een *mutatieverslag* krijgen. Hierop staan alle mutaties vermeld, die op zijn GTD zijn aangebracht. Per mutatie worden o.a. vermeld: datum, tijdstip, wachtwoord en terminal via welke de mutatie is ingevoerd. Op die manier kan een TBB, weliswaar achteraf, nagaan wat er met zijn GTD is gebeurd.
- f. De *CBB* heeft toegang tot enkele metagegevens van de GTD's. Zo is de CBB bijvoorbeeld bekend met het systeemkenmerk van een GTD, omdat hij dit systeemkenmerk zelf aanbrengt. De CBB heeft echter zonder toestemming van een TBB geen toegang tot diens GTD.

Er kan worden geconcludeerd, dat binnen 'GIRAF' veel aandacht is besteed om de privacy van gegevens in een GTD te waarborgen. De genomen maatregelen zijn echter niet voldoende. De *omgeving* waarbinnen 'GIRAF' gaat functioneren, is mede bepalend voor het effect van de getroffen maatregelen. Immers, een gevangenis met enkele dikke muren, maar met een open tuinhekje aan de achterkant biedt de maatschappij niet die bescherming, die zij verwacht.

7 BEVEILIGINGSASPECTEN

'GIRAF' is ontwikkeld met behulp van het database-management- en datacommunicatie-systeem IDMS DB/DC. Dit draagsysteem heeft mogelijkheden om maatregelen te treffen tegen het verloren gaan van gegevens zoals recovery/restart, journal files, logging files, checkpointing, roll back, etc. Het Rekencentrum van de TH-Delft biedt mogelijkheden om GTD's te kopiëren en elders op te slaan. In het kader van dit artikel gaat het te ver om op deze beveiligingsaspecten nader in te gaan. Er kan worden geconcludeerd dat er voldoende mogelijkheden zijn om de gegevensbestanden te beschermen tegen calamiteiten. Wel is het voor sommige **informatiesystemen** (zie figuur 9) noodzakelijk dat er een noodprocedure beschikbaar is, zodat bij het uitvallen van de computer het informatiesysteem kan blijven functioneren.

8 TOEPASSINGEN

De bestanden van enkele tientallen gebruikers van de oude systemen 'BIBLIOSYSTEM' en 'BIBINFO' zijn inmiddels geconverteerd of worden binnenkort geconverteerd naar 'GIRAF'. Enkele daarvan zijn:

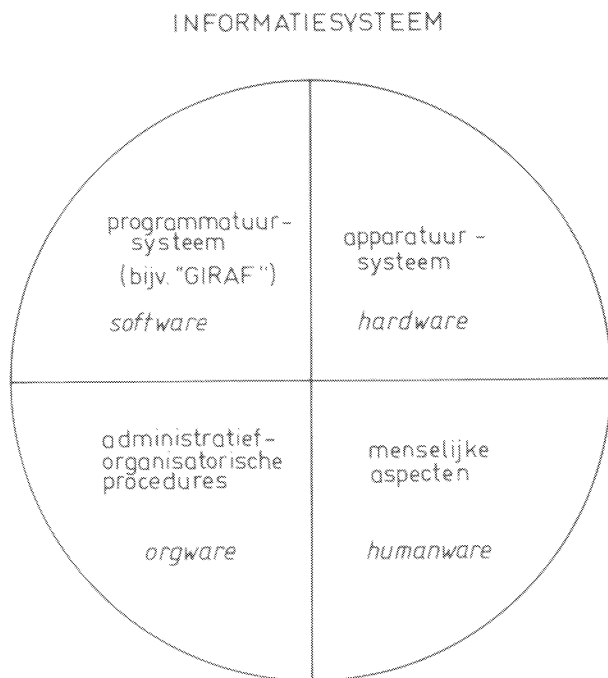
- a. Diverse GIRAF-Toepassings-Databases (GTD's) die literatuurreferenties bevatten.

- b. Diverse GTD's die de gegevens van een boekenverzameling bevatten.
- c. Een GTD die de ledenadministratie van het Muzisch Centrum bevat.
- d. Een GTD die de apparatuurgegevens bevat van de afdeling Scheikunde.
- e. Het Transferpunt van de TH-Delft functioneert als intermediair tussen de TH en het bedrijfsleven. Het heeft tot doel de overdracht van kennis en deskundigheid vanuit de TH naar met name kleine en

middelgrote ondernemingen te bevorderen. Het Transferpunt heeft enkele GTD's op het gebied van: adressen, beschikbare kennis binnen de TH-Delft, aanvragen tot kennisoverdracht, projecten, etc.

- f. Een GTD die de gegevens van alle correspondentie van de afdeling ISO van het Rekencentrum bevat. De secretaresse van de afdeling ISO kan daarmee o.a. de medewerkers van de afdeling maandelijks een rappelijst sturen van de correspondentie die zij nog niet afgehandeld hebben.

Met de ingebruikneming van het **programmatuursysteem** (zie figuur 9) 'GIRAF' is het Rekencentrum van de TH-Delft in staat om de komende jaren bepaalde **klassen van informatiesystemen** op zeer snelle wijze te ontwikkelen.



Figuur 9: Componenten van een informatiesysteem

9 LITERATUUR

1. Application Development System/Online. Summary Description. Cullinet Software, 1983
2. Application Development System/Online. Application Design Guide, release 1.1. Cullinet Software, augustus 1983
3. Dijk, A. J. van; 'BIBLIOSYSTEM'. Computerhulp bij het samenstellen van bibliografieën. In: *Informatie*, jaargang 13 nr. 10 (oktober 1971), pp. 448-455
4. Dijk, A. J. van: 'BIBINFO' – Een real-time retrieval systeem. Afstudeerverslag, vakgroep Informatica TH-Delft. Memorandum no. RC-DMS-76 002, november 1976, 236 pag.
5. Dijk, A. J. van; Ervaringen met het COMMAND-level van CICS/VS bij de ontwikkeling van een geautomatiseerde bibliotheek-toepassing. In: *Informatie*, jaargang 22 nr. 4, april 1980, pp. 313-322
6. Dijk, A. J. van en R. de Roos: 'GIRAF' – General Information Retrieval Facilities Functioneel ontwerp. Technische Hogeschool Delft, Rekencentrum Memorandum no. RC-ISO-82 012, december 1982

Artikel 7

ADS/ONLINE

Ervaringen met een "ontwikkeltool"

bij de ontwikkeling van het

information retrieval systeem

GIRAF

1984

Over kwaliteit valt te twisten.

AvD

Dit artikel is eerder verschenen in:

Informatie, november 1984, jaargang 26 nr. 11 pag. 912 t/m 927

ADS/ONLINE

ERVARINGEN MET EEN 'ONTWIKKELTOOL' BIJ DE ONTWIKKELING VAN HET INFORMATION RETRIEVAL SYSTEEM GIRAF

door ir. A. J. van Dijk

Bij de Technische Hogeschool Delft is het information retrieval systeem 'GIRAF' ontwikkeld. Over de functionele aspecten van dit programmatuursysteem is gepubliceerd in Informatie, mei 1984.

Bij de ontwikkeling van 'GIRAF' heeft het database-management- en datacommunicatiesysteem IDMS DB/DC als draagsysteem gefungeerd. Bovendien werd gebruik gemaakt van het ontwikkeltool 'Application Development System (ADS/ONLINE)', dat evenals IDMS DB/DC, door de firma Cullinet Software geleverd wordt.

In dit artikel komen enkele technische aspecten, die bij de ontwikkeling van 'GIRAF' een rol hebben gespeeld, aan de orde. Met name wordt ingegaan op de ervaringen die zijn opgedaan met ADS/ONLINE in relatie tot schermbeeldcommunicatie/dialogcommunicatie, multiple databases en prototyping.

1 INLEIDING

Het vervaardigen van een informatiesysteem is geen eenvoudige zaak. Er zijn diverse oorzaken die het ontwikkelen en invoeren van informatiesystemen doen mislukken [21]. Gebruikers van, met succes ontwikkelde en ingevoerde, informatiesystemen zorgen bovendien, doorgaans binnen korte tijd na oplevering van het informatiesysteem, voor wensen met betrekking tot uitbreidingen en/of aanpassingen van het informatiesysteem. Oonincx [21] rekent de mogelijkheden tot uitbreidingen en aanpassingen van een informatiesysteem tot de flexibiliteitsaspecten van een informatiesysteem:

'Tot het aspect der flexibiliteit moet ook gerekend worden de mogelijkheid het ontworpen informatiesysteem naderhand nog uit te breiden met nieuwe informatiesubsystemen die bij het ontwerp niet voorzien waren ('open-ended'-informatiesysteem). Als het informatiesysteem is opgezet kan het ook onder omstandigheden werken die verschillen met die ten tijde van de opzet ervan. Dit eist aanpassingsvermogen ('adaptability').

In het meinummer 1984 van *Informatie* [14] is een artikel gepubliceerd dat de functionele aspecten van het programmatuursysteem 'GIRAF' beschrijft. 'GIRAF' (General Information Retrieval Facilities) is een programmatuursysteem dat met behulp van de System Development Methodology (SDM) [17] is ontwikkeld door de afdeling Ontwikkeling Informatiesystemen (ISO) van het Rekencentrum van de Technische Hogeschool Delft (kortweg TH-Delft). Met behulp van 'GIRAF' is het Rekencentrum van de TH-Delft in staat om bepaalde klassen van informatiesystemen op zeer snelle wijze te ontwikkelen. Craenen [9] hanteert het begrip 'infosystemen' voor algemeen toepasbare programmatuur-/informatiesystemen voor eindgebruikers. 'GIRAF' kan tot de categorie der 'infosystemen' worden gerekend. In hoofdstuk 2 van dit artikel wordt een korte uiteenzetting van 'GIRAF' gegeven.

Bij de technische bouw van 'GIRAF' is o.a. gebruik gemaakt van een Amdahl 470V/7B computer en van het

'ontwikkeltool' **Application Development System/Online** (kortweg ADS/ONLINE) [1,2,3,4] van de firma Cullinet Software. Dit ontwikkeltool wordt ook gebruikt bij de ontwikkeling van een geautomatiseerd magazijnadministratie- en voorraadbeheersysteem ('GEMS') [16]. Leveranciers van programmatuur doen regelmatig aankondigingen van binnenkort te verschijnen nieuwe versies van hun produkten. Het ontwikkelen van een programmatuursysteem kan echter alleen plaatsvinden met de voor de gebruiker (systeemontwerper) beschikbare versies. De technische beslissingen inzake 'GIRAF' zijn genomen in **december 1983** op basis van de op **dat moment beschikbare** softwareprodukten. Eventuele latere aankondigingen van nieuwe versies zijn niet meer in de beschouwingen betrokken. In hoofdstuk 3 wordt een aantal basisbegrippen met betrekking tot ADS/ONLINE behandeld zoals het begrip 'dialoog', de samenhang tussen dialogen, het vervaardigen van schermbeelden en de relatie van ADS/ONLINE tot (pseudo-)conversationale programmering. De ervaringen, op systeemontwerpniveau, die zijn opgedaan met ADS/ONLINE bij de ontwikkeling van de systemen 'GIRAF' en 'GEMS' komen aan de orde in hoofdstuk 4. Ervaringen met betrekking tot responsietijden, geheugenbeslag, etc. komen niet aan de orde. Wellicht kunnen andere ADS/ONLINE gebruikers deze aspecten in een volgend artikel belichten. Paragraaf 4.2 handelt over de communicatie tussen dialogen. Daarbij wordt aangegeven op welke wijze aan de flexibiliteitsaspecten **open-ended** en **adaptability** gestalte is gegeven. Andere onderwerpen die aan bod komen zijn **prototyping** en het werken met **multiple databases**. Voor lezers die slechts in enkele van de in dit artikel behandelde onderwerpen geïnteresseerd zijn volgt hier de inhoudsopgave:

- 1 INLEIDING
- 2 DE TOEPASSING
- 3 ADS/ONLINE
 - 3.1 Inleiding
 - 3.2 Dialogen
 - 3.3 Samenhang tussen dialogen

- 3.4 OnLine Mapping (OLM)
- 3.5 ADS/ONLINE versus (pseudo-)conversationale programmering
- 4 ADS/ONLINE IN RELATIE TOT DE TOEPASSING
- 4.1 Inleiding
- 4.2 Communicatie tussen dialogen
- 4.3 Multiple databases
- 4.4 Prototyping
- 5 INDICES
- 6 CONCLUSIES
- 7 LITERATUUR/REFERENTIES

2 DE TOEPASSING

2.1 Inleiding

In 1970 is bij de TH-Delft het systeem 'BIBLIOSYSTEM' [11] ontwikkeld. Aanleiding voor de bouw van dit systeem was de grote hoeveelheid tijd, die onderzoekers moesten besteden aan het verzamelen, registreren en ordenen van literatuur. De klassieke wijze van rapporteren van een literatuurstudie was zeer tijdrovend o.a. vanwege de omvang van het typewerk. Bibliografieën waren soms op het moment dat ze gereed waren voor verspreiding alweer verouderd. Een en ander leidde tot het idee om het samenstellen van bibliografieën te automatiseren. 'BIBLIOSYSTEM' werd begin 1971 door de TH-Delft in gebruik genomen. In de loop der jaren bleek, dat door de 'abstracte' opzet het systeem bruikbaar was voor allerhande toepassingen. Naast bestanden met literatuurreferenties werden met behulp van 'BIBLIOSYSTEM' adresbestanden, bestanden voor ledenadministraties, apparatuurbestanden, diabeestanden, boekbestanden, etc. vervaardigd, bijgewerkt en afgedrukt.

In 1976 ontwikkelde het Rekencentrum op verzoek van een congresorganisatiecomité het real-time systeem 'BIBINFO' [12]. Dit systeem werd als het ware bovenop 'BIBLIOSYSTEM' gebouwd. Het opbouwen van de bestanden gebeurde met 'BIBLIOSYSTEM'. Het raadplegen en het on-line selecteren gebeurde met 'BIBINFO'. 'BIBLIOSYSTEM' (1970) en 'BIBINFO' (1976) zijn ontwikkeld voor een beperkte doelgroep met methoden en hulpmiddelen, die in die jaren modern waren. Het grote succes van deze systemen binnen de TH-Delft en de voortdurende vraag naar dergelijke faciliteiten heeft de directie van het Rekencentrum in 1982 doen besluiten een nieuw information retrieval systeem te laten ontwikkelen, dat op een ruimere doelgroep moest worden gericht. Het moest de systemen 'BIBLIOSYSTEM' en 'BIBINFO' vervangen en zeer gebruiksvriendelijk zijn. Dit nieuwe systeem is het programmatuursysteem 'GIRAF' geworden.

2.2 Uitgangspunten 'GIRAF'

Voordat met de bouw van 'GIRAF' is gestart is een aantal eisen geformuleerd waaraan het systeem moest voldoen [14]. De eisen kunnen als volgt worden samengevat:

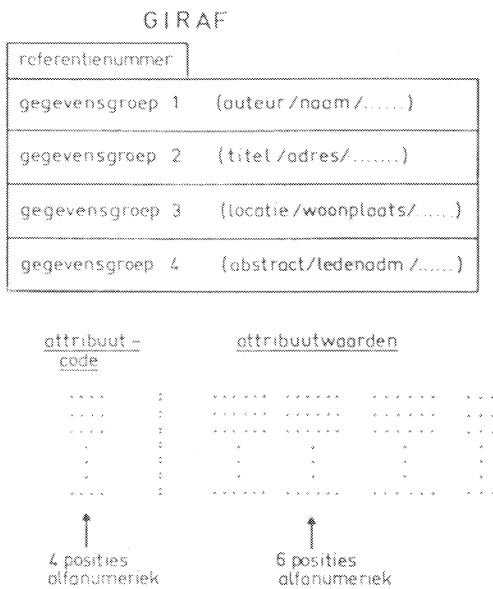
- Uit het succes van de systemen 'BIBLIOSYSTEM' en 'BIBINFO' is de conclusie getrokken, dat de filosofie, waarop deze systemen waren gebaseerd, voor de TH-Delft een goede filosofie is. De eerste voorwaarde, die aan 'GIRAF' is gesteld, is het handhaven van deze filosofie. Dit betekende

o.a., dat 'GIRAF' een **eenvoudig** en **doeltreffend** information retrieval systeem moest worden (eis 1).

- Aangezien 'GIRAF' voor een ruimere doelgroep moest worden gemaakt, moest het algemener bruikbaar zijn dan de oude systemen (eis 2).
- De gegevenseenheid in de oude systemen was de referentie. Toegang tot een **individuele** referentie geschiedde via het referentienummer of de auteur. Toegang tot een **verzameling** referenties was alleen mogelijk via een codesysteem. Dit codesysteem bestond uit een tiental attributen met voor ieder attribuut een waardenverzameling. Aan iedere referentie kon per attribuut één waarde uit de betreffende waardenverzameling worden toegekend. Binnen 'GIRAF' moest het codesysteem ruimer zijn (eis 3).
- De TH-Delft is een pluriforme organisatie, die veel zelfstandig opererende groeperingen herbergt. Sommige van die groeperingen willen (delen van) hun informatievoorziening op een geïsoleerde manier realiseren (bijv. promovendi) (eis 4). Andere groeperingen willen graag samenwerken met groeperingen binnen of buiten de TH-Delft. Voor die groeperingen is het noodzakelijk om vanuit verschillende locaties, met behulp van on-line-verwerking, bestanden te kunnen opbouwen, raadplegen en/of muteren (eis 5).
- 'GIRAF' moet kunnen worden toegepast ten behoeve van veel informatiesystemen binnen de TH-Delft. Daarom moet 'GIRAF' worden aangeboden als **confectiesysteem**. De opzet moest echter dusdanig zijn, dat het met weinig inspanning kan worden aangepast. Op die manier kan 'GIRAF' voor sommige gebruikers als **maatconfectiesysteem** fungeren (eis 6).
- Bij de TH-Delft werken en studeren ook buitenlanders. Het is daarom wenselijk dat 'GIRAF' kan werken met andere talen. Het systeem moest zodanig worden opgezet, dat in een later stadium **andere taalversies** snel kunnen worden geïmplementeerd (eis 7).
- 'GIRAF' moest voorzieningen hebben om de gebruikers van 'BIBLIOSYSTEM' en 'BIBINFO' op een zeer soepele wijze te kunnen laten overstappen naar het nieuwe systeem (eis 8).
- Door het Rekencentrum van de TH-Delft is als draagsysteem ten behoeve van database-management- en datacommunicatie-activiteiten gekozen voor IDMS DB/DC, dat wordt geleverd door de firma Cullinet Software. 'GIRAF' diende met behulp van IDMS DB/DC te worden ontwikkeld (eis 9).

2.3 Indeling van de referenties

De eenheid in de systemen 'BIBLIOSYSTEM' en 'BIBINFO' was de referentie. Hoewel oorspronkelijk bedoeld voor het behandelen van literatuurreferenties werden de systemen voor allerhande toepassingen gebruikt. De naamgeving was echter gebaseerd op literatuurreferenties. Zo was er sprake van een auteur, een titel, een locatie en een abstract. Daarnaast werd aan een referentie voor een aantal (maximaal 10) attributen een attribuutwaarde toegekend. Aangezien de term 'referentie' bij veel gebruikers van de oude systemen is ingeburgerd, is besloten deze term in 'GIRAF' over te nemen. 'GIRAF' moest een algemeen karakter hebben, daarom kon niet worden volstaan met de bestaande naamgeving.



Figuur 1: Gegevensmodel van een referentie binnen 'GIRAF'

Binnen 'GIRAF' bestaat een referentie uit vier gegevensgroepen (zie figuur 1). Iedere gebruiker die verantwoordelijk is voor een gegevensverzameling (in 'GIRAF'-termen: iedere ToepassingsBestandsBeheerder (TBB)) kan zelf de namen kiezen, die hij aan deze gegevensgroepen wenst te geven. Overal waar de gegevensgroepen worden gebruikt (op beeldscherm of papier) worden de door de TBB opgegeven namen gebruikt. 'GIRAF' kent zelf aan iedere nieuwe referentie een referentienummer toe.

Het codesysteem, waarmee attribuutcodes en attribuutwaarden aan een referentie kunnen worden toegekend, is uitgebreid (zie figuren 1 en 2). Het aantal attribuutcodes is nu onbeperkt. Bovendien kunnen per referentie aan een attribuutcode verschillende alfanumerieke attribuutwaarden worden toegekend.

2.4 Subsystemen

Het programmatuursysteem 'GIRAF' is opgedeeld in 4 subsystemen, te weten:

- subsysteem 1: conversie/opbouw database
- subsysteem 2: on-line activiteiten
- subsysteem 3: standaard batch-tools

Figuur 2: Voorbeeld van een 'GIRAF'-referentie

00000427		
Roox.R. de		
Speelgraagsingel 392 Hofje 3		
2659 KL Delft bij Den Haag		
Betaling	:	behoeft niet meer dan fl. 110,- te betalen
Telefonisch bereikbaar	:	079-896257
Organisatie	:	is in de toekomst beschikbaar voor een bestuursfunctie
Bijzonderheden	:	spreekt chinees

Attribuut- code	Attribuutwaarde	
BETA	B4-2	(betaling)
INST	PIRFO	(bespeelt instrument)
INST	GITAAR	
INST	KLASST	
CURS	BRMS	(volgt cursus)
CURS	JAZZ	
TAAL	REU	(spreekt taal)
TAAL	ENG	
TAAL	CHIN	
STAT	TH-NEE	(status)
JAAR	1980	(jaar van de inschrijving)

- subsysteem 4: gebruikersafhankelijke toepassingen

Met behulp van **substelsysteem 1** (conversie/opbouw database) kunnen bestanden in 'BIBLIOSYSTEM'-formaat worden geconverteerd naar een 'GIRAF-Toepassings-Database' (GTD).

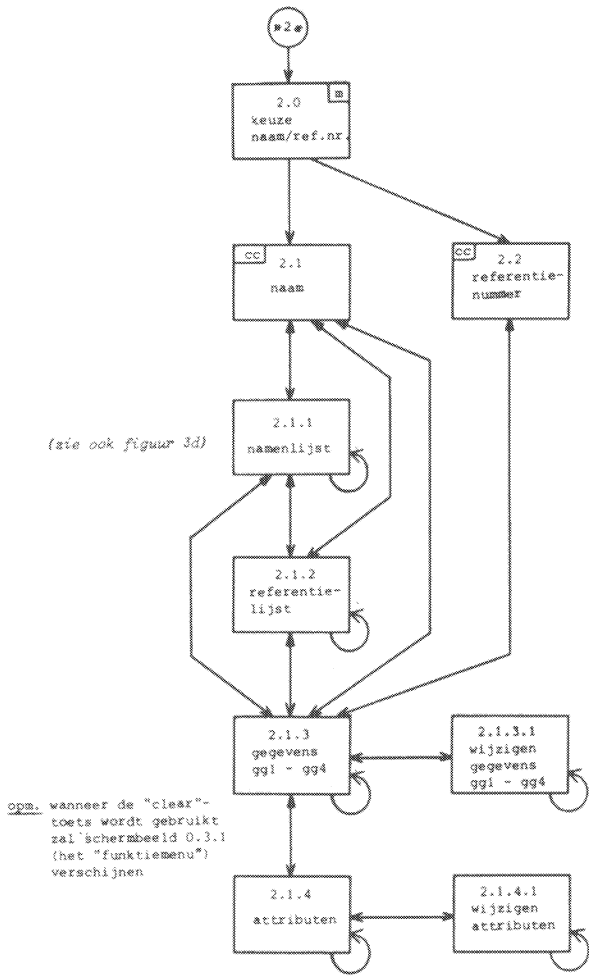
Het on-line subsysteem (**substelsysteem 2**) bestaat uit de volgende functies:

- *1* help (handleiding)
- *2* ophalen/muteren individuele referentie
- *3* selecteren/verwijderen verzameling referenties
- *4* opstarten serieverwerking (batch)
- *5* inbrengen nieuwe referentie(s)
- *6* wijzigen bestandsgegevens
- *7* *gebruikersafhankelijke functie*
- *9* afsluiten sessie

Indien voor de betreffende gebruiker een (on-line) gebruikersafhankelijk subsysteem beschikbaar is, dan kan via functie 7 toegang tot dit systeem worden verkregen. Naast deze functies is er een functie, die de toelating tot een GTD regelt. Om 'GIRAF' te kunnen gebruiken moet een gebruiker een zogenaamd systeemkenmerk en een wachtwoord intoetsen. Op basis van deze invoergegevens wordt onderzocht of de gebruiker toegang wordt verleend tot een GTD en zo ja welke faciliteiten hem worden toegestaan. Substelsysteem 2 bevat 75 schermbeelden. De communicatie tussen deze schermbeelden is weergegeven met behulp van SchermbeeldCommunicatieDiagrammen (SCD's) (zie figuren 3a, 3b, 3c en 3d).

Figuur 3a: Gebruikte symbolen in de SchermbeeldCommunicatieDiagrammen (SCD's)

symbool	verklaring
2.1.1 namenlijst	schermbeeld met schermbeeldnummer en schermbeeldomschrijving
cc	schermbeeld te bereiken via "command-chaining"
m	schermbeeld met "menu"
	schermbeeld verwijst via optie naar zichzelf (bijvoorbeeld "bladernogelijkheid")
	ingang via "clear"-toets
	uitgangverwijzing
	ingangverwijzing
	richting als gevolg van een optie-keuze
	richting als gevolg van optie "vorig niveau"



Figuur 3b: Voorbeeld van een SchermbeeldCommunicatieDiagram (SCD)

Gebruikers van 'GIRAF' krijgen door middel van **sub-systeem 3** een aantal gereedschappen voor batch-verwerking. Zo bevat dit subsysteem o.a. uitgebreide rapportagefaciliteiten, zoals statistische overzichten omtrent de omvang en samenstelling van de GTD, het afdrucken van een gebruikershandleiding, het afdrucken van een attribuutcodeoverzicht etc. Ook een batch-selectieprogramma, om grote deelbestanden te construeren, behoort tot dit subsysteem. 'GIRAF' wordt aangeboden als **confectiesysteem**. De opzet is echter dusdanig, dat met weinig inspanning wijzigingen kunnen worden aangebracht. Op die manier kan van een **maatconfectiesysteem** worden gesproken. Bepaalde categorieën van gebruikers zullen 'GIRAF' willen (laten) uitbreiden met een 'eigen' on-line systeem (**sub-systeem 4**). De structuur van 'GIRAF' maakt dit mogelijk. In 4.2 wordt hierop nader ingegaan.

3 ADS/ONLINE

3.1 Inleiding

Sinds 1973 is het database-managementsysteem IDMS beschikbaar. IDMS (Integrated Data Management System) is een database-managementsysteem, dat geba-

seerd is op het netwerkmodel, zoals dat gedefinieerd is in het 'CODASYL-report' [10]. IDMS wordt zowel in de Verenigde Staten van Amerika als daarbuiten door veel bedrijven en ondernemingen gebruikt als hulpmiddel bij activiteiten op het gebied van de geautomatiseerde gegevensverwerking. Gebruikers die on-line toepassingen construeren gebruiken IDMS in combinatie met een teleprocessing-monitor (kortweg TP-monitor), bijvoorbeeld de TP-monitor CICS/VS van IBM. In 1979 announceerde de firma Cullinet de TP-monitor IDMS/DC, die geïntegreerd was met IDMS. Het ontwikkelen van on-line toepassingen met behulp van een TP-monitor was aanvankelijk een tamelijk moeilijke en moeizame aangelegenheid waarvoor gespecialiseerde programmeurs nodig waren. Aan het eind van de jaren zeventig werd het ontwikkelen van on-line toepassingen wat vereenvoudigd toen de macro-level programmeerstructuren werden vervangen door command-level programmeerstructuren [13]. Het aantal on-line toepassingen nam sinds die tijd drastisch toe. Daarom werd gezocht naar hulpmiddelen, die het maken van toepassingsprogramma's verder vereenvoudigen. ADS/ONLINE is zo'n hulpmiddel. Het is als het ware een 'ontwikkeltool'. Met ADS/ONLINE kunnen TP-toepassingen niet alleen sneller worden ontwikkeld, maar ze zijn ook flexibeler m.b.t. het verrichten van aanpassingen. ADS/ONLINE moet gezien worden als een TP-toepassing onder IDMS/DC. Centraal staat het begrip dialoog (zie 3.2). Met behulp van ADS/ONLINE kunnen dialogen worden ontwikkeld. Belangrijk is dat deze dialogen ook op uitvoeringstijd werken onder

Figuur 3c: Voorbeeld van een Schermbeeld Definitie (SD)

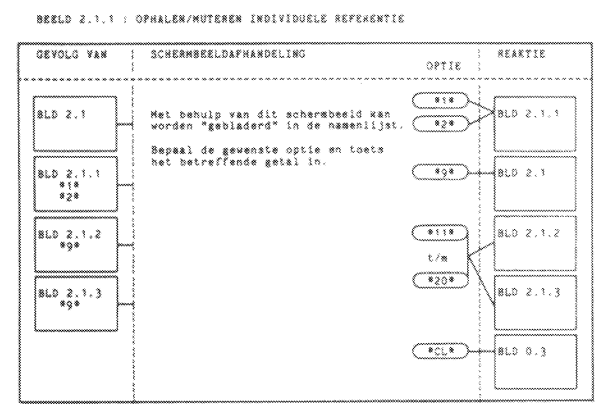
```

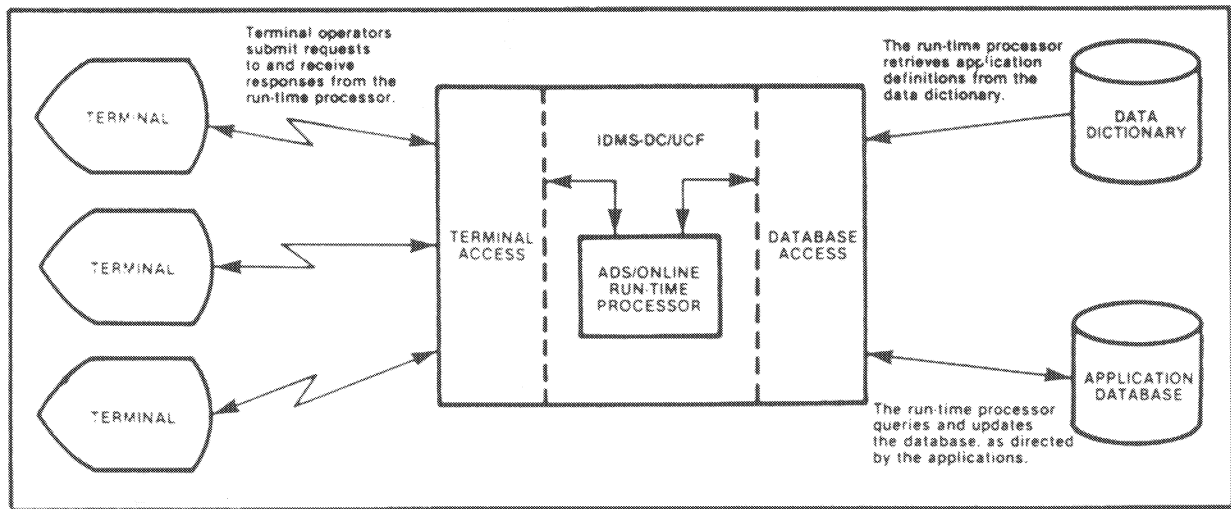
*****
*** TECHNISCHE HOOGESCHOOL DELFT *** FUNKTIE 2.1.1
*** GIRAF : "Bestands-ident" ***
*** LIJST VAN NAMEN ***
*11# *****
*12# *****
*13# *****
*14# *****
*15# *****
*16# *****
*17# *****
*18# *****
*19# *****
*20# *****

*1# OPVOLGENDE SERIE NAMEN
*2# VOORAFGAANDE SERIE NAMEN
*3# VORIG NIVEAU

MAAK UW OPTIE-KEUZE EN TOETS HET BETREFFENDE GETAL IN :
*****
"SYSTEEM-ROODSCHAPPEN"
*****
    
```

Figuur 3d: Voorbeeld van een Communicatie Diagram (CD)





Figuur 4: De omgeving waarbinnen ADS/ONLINE functioneert (Bron: ADS/ONLINE Summary Description)

ADS/ONLINE. Figuur 4 laat zien hoe op uitvoeringstijd ADS/ONLINE-dialogen communiceren met terminals en database. In het kader van dit artikel is het niet mogelijk om ADS/ONLINE in extenso te behandelen.

In de volgende paragrafen wordt ingegaan op enkele basisbegrippen van ADS/ONLINE, zoals het begrip dialoog en de samenhang tussen dialogen. Ook wordt aandacht besteed aan de manier, waarop schermbeldden kunnen worden vervaardigd. Het hoofdstuk wordt besloten met een paragraaf waarin een vergelijking wordt gemaakt tussen het ontwikkelen van on-line toepassingen onder ADS/ONLINE, het ontwikkelen van on-line toepassingen m.b.v. conversationale programmering en het ontwikkelen van on-line toepassingen m.b.v. pseudo-conversationale programmering.

3.2 Dialogen

Centraal in ADS/ONLINE staat de **dialoog**. Deze term verdient enige toelichting, omdat de begrippen achter een term nogal eens verschillen [7]. Zo verstaat Craenen [9] onder een dialoog: 'een interactieve vorm van communicatie tussen mens en computer'.

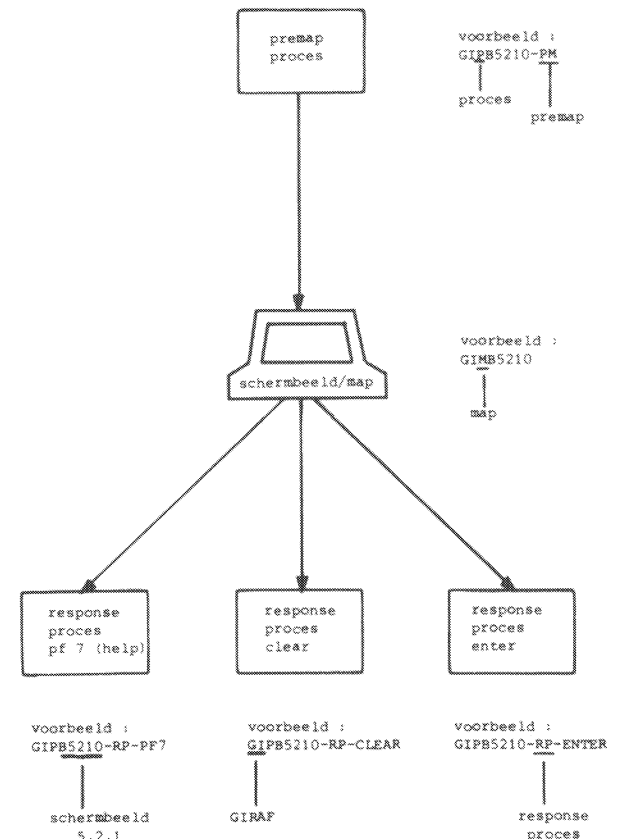
Janssens [20] onderkent drie elementaire dialoogstructuren, namelijk de sequentie, de selectie en de iteratie. Zijn opbouw van dialoogstructuren lijkt voornamelijk geschikt voor conversationale programmering. In een dialoog van ADS/ONLINE staat het **schermbeeld** centraal (zie figuur 5). Een schermbeeld wordt beschreven met behulp van een **map**. Een **map** is een soort tabel, waarin alle velden van een bepaald schermbeeld met een symbolische naam worden gedefinieerd. De velden bestaan uit tekstvelden en invoer-/uitvoervelden. Deze laatste velden worden gekoppeld aan de velden van één of meer records, de zogenaamde **maprecords**. Deze maprecords zijn gedefinieerd in de Integrated Data Dictionary (IDD) behorend tot IDMS.

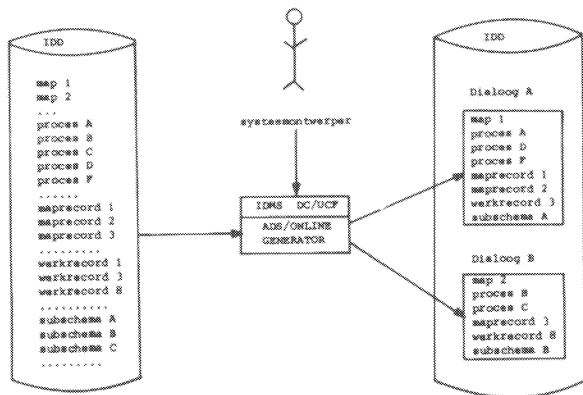
Naast schermbeeld (of map) kent een ADS/ONLINE-dialoog een zogenaamd **premap-proces** en **response-processen**. Een premap-proces is een programma, dat nadat de betreffende dialoog geactiveerd is, wordt uitgevoerd voordat de map (het schermbeeld) op het beeldscherm zichtbaar wordt gemaakt. Nadat de gebruiker van de dialoog, die onderdeel is van een of andere toepassing, gereageerd heeft op hetgeen hij op het beeldscherm heeft

gezien, wordt een bij de dialoog behorend response-proces gestart. De keuze van het response-proces wordt bepaald aan de hand van de door de gebruiker ondernomen actie. Zo zal het indrukken van de 'enter'-toets een ander response-proces kunnen doen activeren dan het indrukken van de 'clear'-toets. Processen hebben, naast maprecords, vaak werkrecords nodig.

Resumerend kan worden gesteld, dat een ADS/ONLINE-dialoog (doorgaans) bestaat uit: een premap-proces, een map met bijbehorende maprecords, één of meer response-processen en werkrecords. In figuur 5 wordt

Figuur 5: Voorbeeld van een DialoogStructuurDiagram (DSD)





Figuur 6: De ADS/ONLINE-generator (samenknoper)

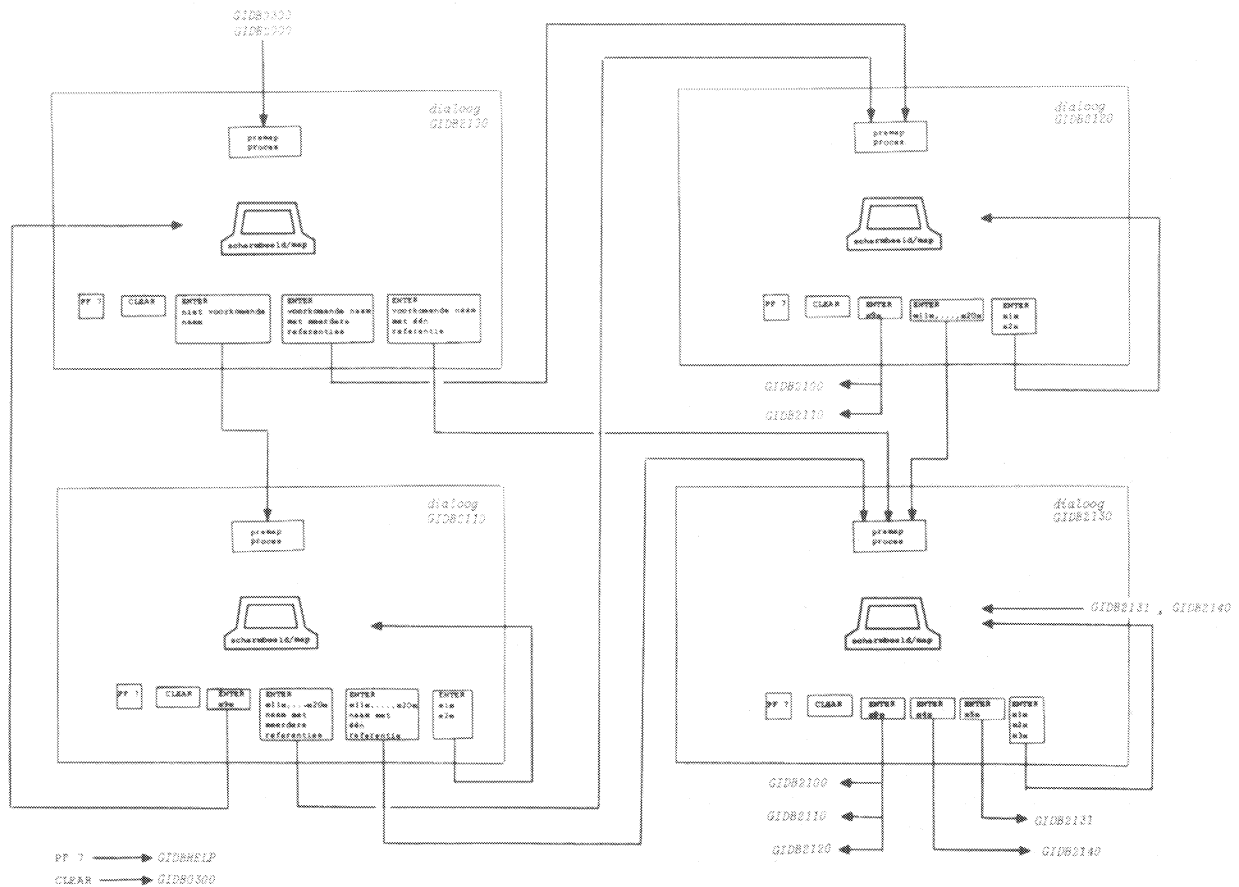
met een DialogStructuurDiagram (DSD) de samenhang tussen processen en map weergegeven. Maprecords, werkrecords en processen worden in IDD gedefinieerd. De map kan via batch-verwerking worden aangemaakt en in IDD worden geplaatst. Maps kunnen echter ook worden aangemaakt met het Cullinet-produkt On-Line Mapping (OLM). OLM biedt de mogelijkheid om, zoals de naam van het betreffende produkt aangeeft, on-line maps te definiëren. Op de ervaringen met OLM wordt in 3.4 nader ingegaan. Een dialoog wordt vervaardigd met behulp van de ADS/ONLINE-generator. Deze generator knoopt de juiste ingrediënten samen tot een ADS/ONLINE-dialoog (zie figuur 6). Invoergegevens voor de (on-line) generator zijn map, maprecords, pre-

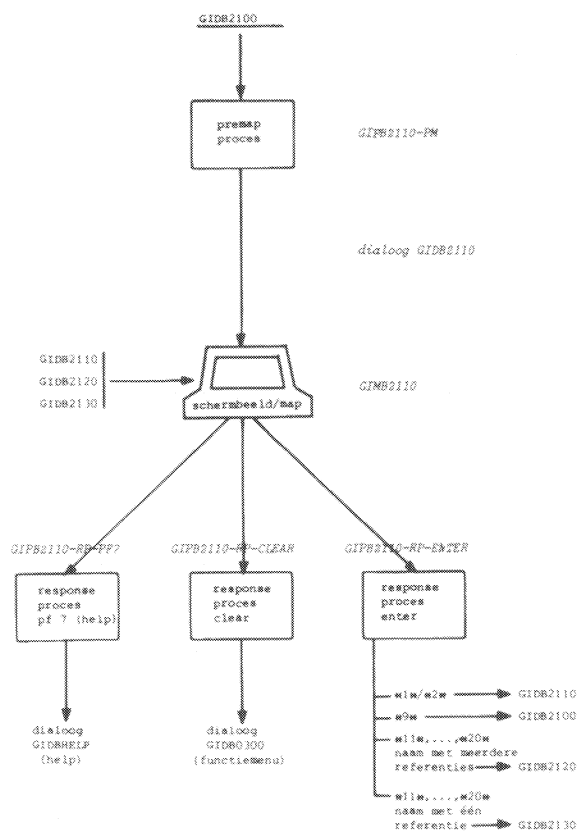
map-proces, response-processen, werkrecords en sub-schema.

3.3 Samenhang tussen dialogen

Een dialoog is doorgaans onderdeel van een verzameling dialogen. Deze verzameling dialogen vormt dan het online gedeelte van een toepassing. Hoe kan nu de samenhang tussen de betreffende dialogen worden geregeld? Eén of meer dialogen fungeren als **hoofddialoog/startdialoog**. Nadat een hoofddialoog is geactiveerd, kan een volgende dialoog worden gestart door in een proces een startcommando te geven. Voorbeelden van startcommando's zijn: INVOKE 'GIDB3121', INVOKE GIDB3121-NAAM, LINK 'GIDB3121'. De betreffende dialoognaam kan als literal worden opgegeven (eerste voorbeeld), maar kan ook als een variabele worden opgegeven (tweede voorbeeld). Deze laatste mogelijkheid is voor het systeem 'GIRAF' van zeer groot belang gebleken. In 4.2 zal hierop nader worden ingegaan. Het is wenselijk om goed inzicht te hebben hoe het verloop tussen de dialogen kan plaatsvinden. Het Dialogoog-CommunicatieDiagram (DCD) (zie figuur 7a) is één van de mogelijkheden om dit te realiseren. Vanuit de response-processen van een dialoog verwijst een pijl naar de dialoog, die door het betreffende response-proces wordt geactiveerd. Met het DCD liggen alle paden, die in een toepassing kunnen worden gevolgd, vast. In de figuren 3a en 3b is aangegeven op welke wijze in het functioneel ontwerp de schermbeelden en het verloop tussen de schermbeelden (m.b.v. SchermbeeldCommunicatie-

Figuur 7a: Voorbeeld van een Dialogoog-CommunicatieDiagram (DCD) (type 1)





Figuur 7b: Voorbeeld van een DialoogCommunicatieDiagram (DCD) (type 2)

Diagrammen (SCD)) wordt vastgelegd. Deze methode met SCD's werd reeds toegepast bij het systeem BIBINFO (1976) en is onafhankelijk van de verdere technische ontwikkeling van het systeem. Opvallend is nu dat de DCD's zeer nauw aansluiten bij de SCD's. Daaruit kan worden geconcludeerd dat met ADS/ONLINE in het technisch ontwerp op een hoger abstractieniveau kan worden gewerkt dan met bijvoorbeeld IDMS/DC of CICS/VS [13,19]. De overgang van functioneel ontwerp naar technisch ontwerp zal soepeler kunnen verlopen. Bovendien kunnen opdrachtgevers, die het functioneel ontwerp kennen, de vorderingen van het technisch ontwerp makkelijker volgen. Het vervaardigen van DCD's, zoals in figuur 7a weergegeven, is bij een on-line programmatuursysteem van enige omvang een nogal tijdrovende bezigheid. Een andere manier om DCD's te maken is het toevoegen van de communicatiegegevens aan het DialoogStructuurDiagram (DSD). In figuur 7b is een en ander weergegeven.

3.4 OnLine Mapping (OLM)

OnLine Mapping (OLM) is geen direct onderdeel van ADS/ONLINE, maar is een hulpmiddel van IDMS/DC. Met OLM is het mogelijk om on-line maps te vervaardigen. De ontwerper krijgt een 'blanco schermbeeld' te zien. Hij kan onmiddellijk het schermbeeld gaan ontwerpen door tekstvelden en invoer-/uitvoervelden te definiëren. OLM heeft een aantal conventies, waarmee een en ander bewerkstelligd kan worden. Een voordeel is dat de lay-out van het schermbeeld direct kan worden bekeken en desgewenst kan worden aangepast. Dit is vooral interessant bij de bouw van prototype (T) in de definitie-

studie. In 4.4 wordt hierop nader ingegaan. Nadat tekstvelden en invoer-/uitvoervelden op het schermbeeld zijn gedefinieerd krijgt de ontwerper de mogelijkheid om aan alle velden, zowel de tekstvelden als invoer-/uitvoervelden, attribootgegevens toe te kennen zoals protected/unprotected, lichtintensiteit normaal/helder/donker etc. De invoer-/uitvoervelden moeten worden gekoppeld aan een veld van een maprecord. Nadat alle velden voorzien zijn van attribootgegevens kan de map worden gegenereerd en in IDD worden opgenomen. Daarmee is de map beschikbaar voor het maken van een ADS/ONLINE-dialoog. OLM is op die manier een prettig hulpmiddel om maps te vervaardigen voor ADS/ONLINE-dialoogen.

De opgedane ervaringen met OLM geven aanleiding tot het maken van enkele opmerkingen:

- In diverse TP-monitoren is het niet mogelijk om bij het koppelen van invoer-/uitvoervelden, die tot de map behoren, aan velden van werkrecords (de maprecords) gebruik te maken van arrays. OLM biedt de mogelijkheid tot het werken met een-dimensionale arrays. Dit maakt de programmering eenvoudiger. Bij 'GIRAF' bestond bij diverse maps de behoefte om te kunnen werken met twee-dimensionale arrays. Helaas is het, werkend met OLM, niet mogelijk om met twee-dimensionale arrays te werken.
- Binnen 'GIRAF' wordt op diverse plaatsen gewerkt met maps, die 'dynamisch samengesteld' worden. Op één punt was het gewenst de lengte van een veld in bepaalde situaties te verkleinen. Hoewel attribootgegevens van een veld dynamisch kunnen worden gewijzigd, is het helaas niet mogelijk een veld (tijdelijk) korter te maken.
- Nadat met OLM een map vervaardigd is, ligt de koppeling tussen één of meer maprecords en de map vast. Een probleem is nu dat het loskoppelen van een maprecord van de map niet mogelijk is zonder eerst de map te elimineren. Aangezien dit doorgaans nogal wat werk met zich meebrengt blijft het niet meer gebruikte maprecord vaak gekoppeld aan de map. Indien er veel wijzigingen optreden komt een en ander de overzichtelijkheid niet ten goede.

Geconcludeerd kan worden dat OLM een goed produkt is om maps on-line te vervaardigen. Daarom verdient dit produkt het dat aan bovenstaande opmerkingen de nodige aandacht wordt geschonken.

3.5 ADS/ONLINE versus (pseudo-)conversationale programmering

Bij on-line informatiesystemen/programmatuursystemen [14] is sprake van onvertraagde (real-time) verwerking van logische eenheden te verwerken gegevens. Deze vorm van gegevensverwerking staat bekend als postgewijze/transactiegewijze gegevensverwerking (transaction processing). De postgewijze gegevensverwerking vindt plaats met behulp van conversaties. Onder een conversatie kan worden verstaan: een vraag- en antwoordspel tussen een gebruiker en een computer met behulp van een beeldscherm. Voor het laten functioneren van on-line postgewijze gegevensverwerkende systemen zijn twee ontwerpstrategieën beschikbaar: conversationale programmering en pseudo-conversationale programmering. Ook een combinatie van beide ontwerp-

strategieën is mogelijk. Allereerst wordt nu een korte beschrijving van conversationele programmering en pseudo-conversationele programmering gegeven. Daarna wordt de relatie met ADS/ONLINE aangegeven.

3.5.1 Conversationele programmering

Bij conversationele programmering worden meerdere conversaties tussen computer en gebruiker in één programma verwerkt (zie figuur 8a). Dit betekent dat het programma de invoer van de gebruiker inleest, deze invoer verwerkt en het resultaat op het beeldscherm plaatst. Daarmee is (een deel van) de post of de transactie met behulp van een conversatie verwerkt. Vervolgens is de gebruiker aan de beurt om gegevens in te toetsen. Dit kan echter enige tijd vragen, soms enkele seconden maar ook weleens enkele minuten. Gedurende deze reactietijd van de gebruiker staat het programma te wachten. Tijdens dat wachten is het programma als het ware 'slapend' maar vraagt wel aandacht van de TP-monitor. Nadat de gebruiker de 'enter'-toets, de 'clear'-toets of een functietoets heeft ingedrukt kan het programma weer in actie komen. Deze procedure herhaalt zich totdat de gebruiker er voor zorgt dat het programma kan worden beëindigd door via de invoergegevens te laten weten dat hij zijn sessie wil beëindigen. Aan veel programmatuursystemen wordt de eis gesteld dat de responsietijden < 4 seconden moeten zijn. Dit betekent dat bij conversationele programmering het programma slechts een zeer klein deel van de verblijftijd actief is.

Voor de komst van TP-monitoren zoals CICS/VS en IDMS/DC bestond een on-line programmatuursysteem vaak uit slechts één conversationeel programma. Een voorbeeld daarvan is het systeem 'BIBINFO' [12]. Dit on-line information retrieval systeem is in 1976 gemaakt. Het on-line gedeelte van dit systeem bestaat uit één groot COBOL-programma dat enkele duizenden instructies omvat. Met behulp van een tiental subroutines, zoals READSCR, WRTSCR etc., wordt de communicatie met de TP-monitor TSO (Time Sharing Option van IBM) geregeld om schermbeelden te maken, te wijzigen, te versturen en in te lezen. Dergelijke conversationele programma's lijken veel op grote batch-programma's. De ervaring heeft geleerd dat geroutineerde batch-programmeurs veel makkelijker kunnen overschakelen naar on-line programmering met behulp van de conversationele programmeringsstrategie dan met pseudo-conversationele programmering.

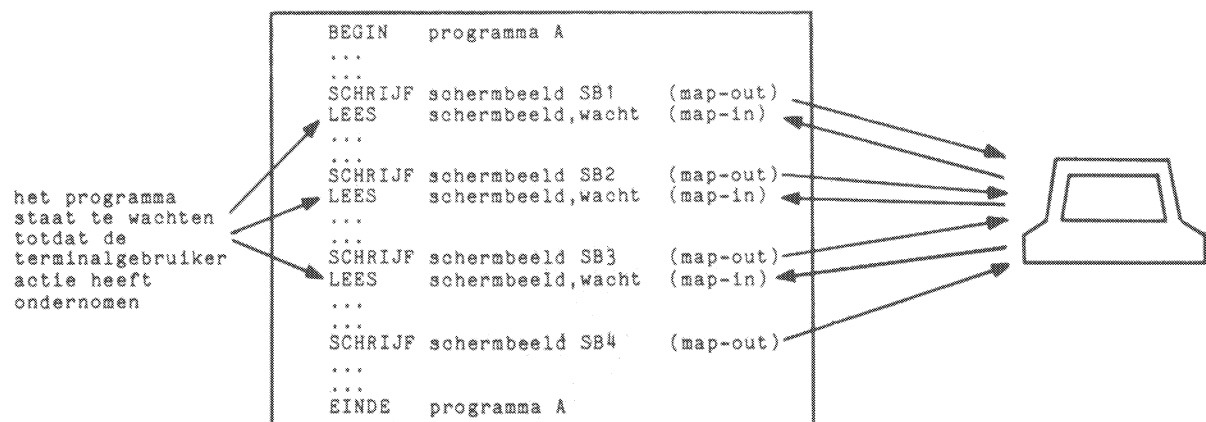
3.5.2 Pseudo-conversationele programmering

Conversationele programma's zijn slechts een zeer klein deel van de verblijftijd actief. Het doel van pseudo-conversationele programmering is nu om dergelijke programma's als het ware in stukjes te knippen. Ieder stukje handelt een (deel van een) transactie af. Het betreffende programma(deel) wordt pas 'opgeroepen' als een gebruiker op de 'enter'-toets, de 'clear'-toets of een functietoets heeft gedrukt (zie figuur 8b). Nadat de invoergegevens zijn verwerkt wordt het resultaat naar het beeldscherm verstuurd. Direct daarna verdwijnt het programma(deel) van het toneel. Het voordeel van deze werkwijze is dat er geen 'slapende' programma's in het geheugen aanwezig zijn. In plaats van één groot programma bestaat bij pseudo-conversationele programmering de applicatie uit een groot aantal kleine (en doorgaans eenvoudige) programma's. Deze programma's zijn als het ware subprogramma's die aan de TP-monitor worden toegevoegd. De TP-monitor verzorgt o.a. alle invoer- en uitvoeropdrachten [13, 19] en kan worden beschouwd als een soort besturingssysteem voor deze programma's. Voor het operating systeem is de TP-monitor echter een gewoon applicatieprogramma. Het vervaardigen van een programmatuursysteem met behulp van pseudo-conversationele programmering vergt een wat andere denkrant dan het maken van grote batch-programma's. Ervaren batch-programmeurs die moeten overschakelen naar pseudo-conversationele programmering hebben het gevoel dat ze niet zelf meer alles in de hand hebben. Het parallel kunnen werken van diverse programma's en de communicatie tussen alle kleine programma's zorgen in het begin voor een onzeker gevoel. Nadat deze programmeurs enkele programma's hebben gemaakt verdwijnt dat gevoel langzaam maar zeker.

3.5.3 ADS/ONLINE pseudo-conversationeel

ADS/ONLINE werkt, zoals aangegeven in 3.2, met dialogen. In een dialoog staat het schermbeeld (map) centraal. Naast het schermbeeld kent een dialoog een pre-map-proces en response-processen. Bij pseudo-conversationele programmering staat het proces, dat de transactie moet verwerken, centraal. Het programma start met het lezen van het schermbeeld (map-in), verwerkt de gegevens en verstuurt een schermbeeld naar het beeldscherm (map-out). ADS/ONLINE heeft dus een duidelijk afwijkende werkwijze voor de systeemontwerper/programmeur dan de pseudo-conversationele werkwij-

Figuur 8a: Conversationele programmering



ze. Toch werkt ADS/ONLINE intern pseudo-conversationeel. In dit artikel wordt niet verder ingegaan op de technische details van een en ander. Vastgesteld kan worden dat met ADS/ONLINE de systeemontwerper/programmeur op een hoger abstractieniveau kan werken dan met een pseudo-conversationele werkwijze (zie 3.3). Desondanks kan er worden geproefteerd van de voordelen van de pseudo-conversationele werkwijze.

4 ADS/ONLINE IN RELATIE TOT DE TOEPASSING

4.1 Inleiding

In dit hoofdstuk wordt ingegaan op de ervaringen die zijn opgedaan met ADS/ONLINE. Er wordt verslag gedaan van een ontwikkeld cassettesysteem. Met dit cassettesysteem kunnen de flexibiliteitsaspecten open-ended en adaptability worden gerealiseerd. Verder wordt er ingegaan op de ervaringen die zijn opgedaan met multiple databases en prototyping.

4.2 Communicatie tussen dialogen

Zoals in 3.3 beschreven is vindt de communicatie tussen dialogen plaats door middel van startcommando's in de response-processen van de dialogen. Als voorbeelden van zo'n startcommando zijn o.a. gegeven: INVOKE 'GIDB3121' en INVOKE GIDB3121-NAAM. In het laatste voorbeeld staat de naam van de dialoog die wordt gestart in de variabele GIDB3121-NAAM. Deze mogelijkheid is voor het programmatuursysteem 'GIRAF'

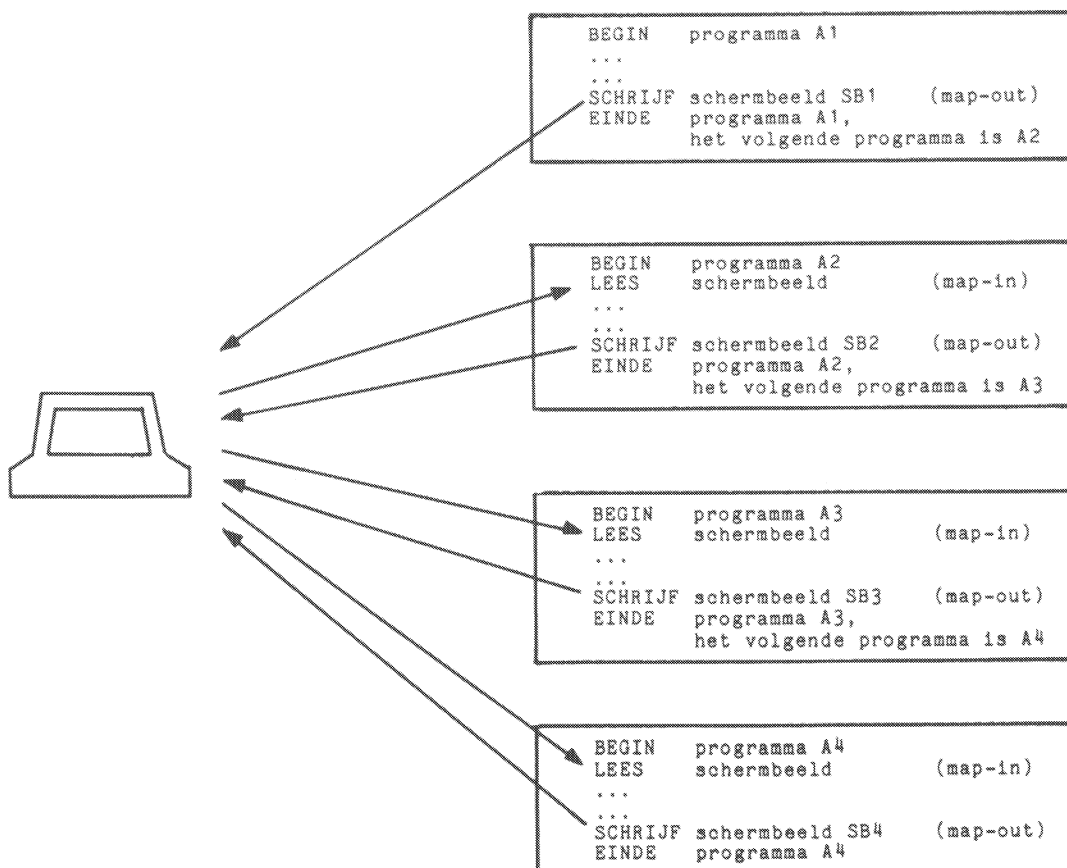
van groot nut gebleken. Binnen 'GIRAF' gebruiken alle response-processen startcommando's met een variabele als operand. De naam van de dialoognaamvariabele is gelijk aan de naam van de dialoog die volgens het DialoogCommunicatieDiagram (DCD) (zie figuren 7a,b) in de betreffende situatie moet worden gestart. De naam van de dialoognaam-variabele wordt gevolgd door de suffix '-NAAM'. Waarom is een en ander nu zo belangrijk? Binnen het programmatuursysteem 'GIRAF' is een cassettesysteem ontwikkeld. Dit houdt in dat, nadat de gebruiker van een GIRAF-Toepassings-Database (GTD) zich heeft gemeld, alle dialoognaam-variabelen door het systeem voor die gebruiker worden gevuld. Allereerst wordt nu ingegaan op de vraag waarom? Daarna zal aan de orde komen hoe een en ander geregeld is.

4.2.1 De motivatie van het cassettesysteem

De volgende motieven hebben een rol gespeeld bij de ontwikkeling van het cassettesysteem:

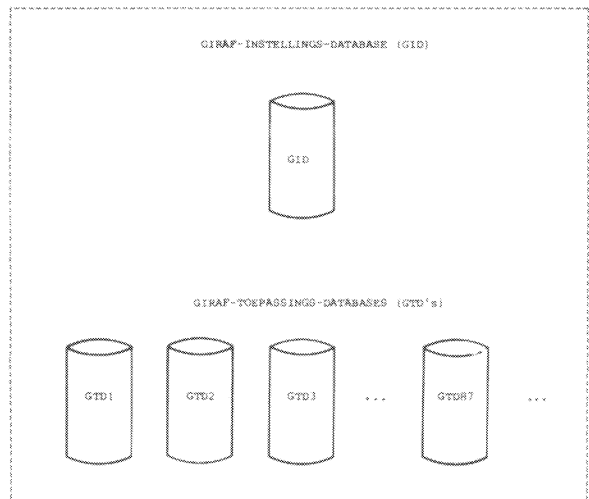
- 'GIRAF' is een algemeen toepasbaar programmatuursysteem en wordt gebruikt voor veel toepassingen. 'GIRAF' wordt aangeboden als confectiesysteem. Eén van de aan 'GIRAF' gestelde eisen was dat het met weinig inspanning moet kunnen worden aangepast. Op die manier kan 'GIRAF' voor sommige gebruikers fungeren als maatconfectiesysteem. Het op maat maken van 'GIRAF' voor een specifieke gebruiker kan nu worden gerealiseerd door de betreffende dialogen te kopiëren, van een andere naam te voorzien en aan te passen.

Figuur 8b: Pseudo-conversationele programmering



Nadat de betreffende gebruiker zich heeft aangemeld bij 'GIRAF', wordt er voor gezorgd dat in de betreffende dialoognaam-variabelen de afwijkende dialoognaam komt te staan. De oproepende dialoog gebruikt in het startcommando de dialoognaam-variabele. De inhoud van deze variabele bepaalt welke dialoog wordt gestart.

- b. Bij de TH-Delft werken en studeren ook buitenlanders. 'GIRAF' moet daarom kunnen werken met andere talen. De gebruiker geeft bij het aanmelden met behulp van een taalcode aan in welke taal hij wil werken. Het cassettesysteem zorgt er niet alleen voor dat de namen van de (gebruikersafhankelijke) dialogen worden geladen, maar zorgt er tevens voor dat, afhankelijk van de opgegeven taalcode, de juiste (taalafhankelijke) dialoognamen worden geladen. De gebruiker kan daarna werken met 'zijn' maatconfectiesysteem in de door hem gewenste taal.



Figuur 9a: Structuur van de GIRAF-DATABASE

4.2.2 De werking van het cassettesysteem

Een GIRAF-database bestaat (zie figuur 9a) uit een GIRAF-INSTELLINGS-DATABASE (GID) en voor iedere toepassing een GIRAF-TOEPASSING-DATABASE (GTD). De GID bevat een recordtype GI-DIANMALG-REC (zie figuur 9b). Een record-occurrence van dit recordtype bevat een dialoognaam als entrynaam en vervolgens tien dialoognaam-substituten. Afhankelijk van de taalcode wordt één van deze dialoognaamsubstituten gesubstitueerd voor de betreffende dialoognaam.

Voorbeeld:

- GIDB3121 (entrynaam)
 GIDB3121 (Ned. dialoog met hoofdletters) (subst. 1)
 GIDN3121 (Ned. dialoog met kleine letters) (subst. 2)
 GIDE3121 (Engelstalige dialoog) (subst. 3)
 GIDF3121 (Franstalige dialoog) (subst. 4)
 ...
 GIDH3121 (Hongaarstalige dialoog) (subst. 10)

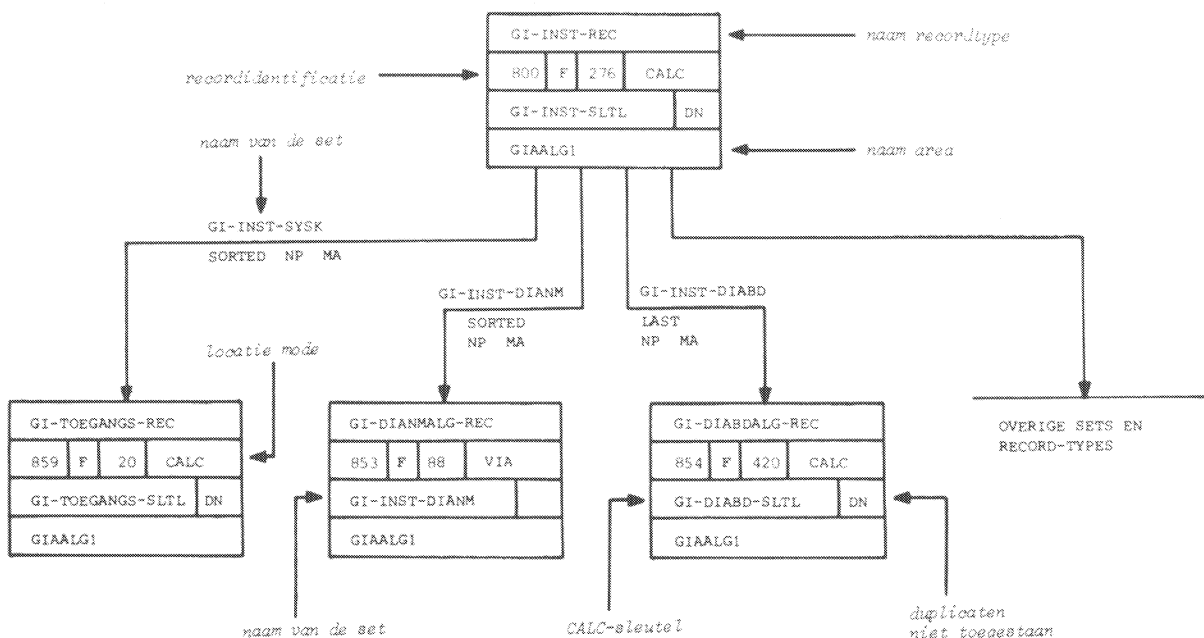
Ten behoeve van een GTD kunnen afwijkende dialogen zijn ontwikkeld. De GTD bevat een recordtype GI-DIANMAFW-REC (zie figuur 9c). De structuur van een record-occurrence is hetzelfde als de structuur van een record-occurrence van recordtype GI-DIANMALG-REC.

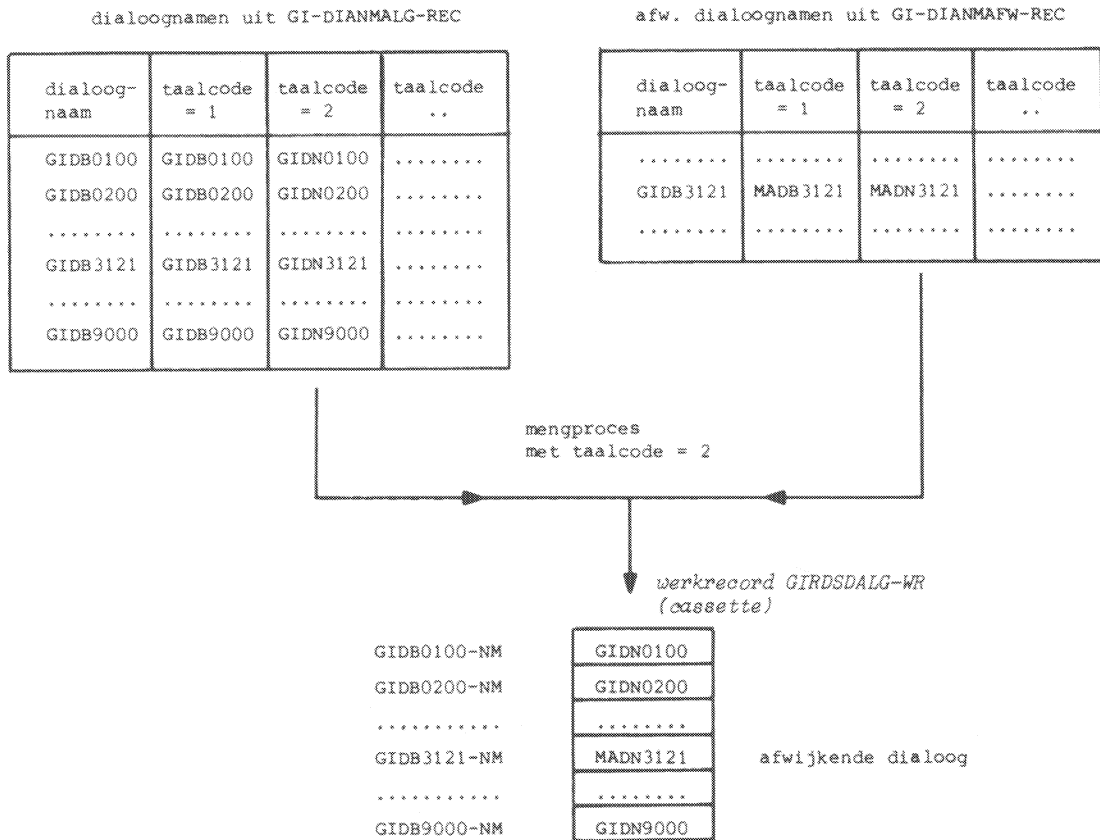
Voorbeeld:

- GIDB3121 (entrynaam)
 MADB3121 (afw. Ned. dialoog met hoofdletters)
 MADN3121 (afw. Ned. dialoog met kleine letters)
 MADE3121 (afwijkende Engelstalige dialoog)
 MADF3121 (afwijkende Franstalige dialoog)
 ...
 MADH3121 (afwijkende Hongaarstalige dialoog)

Vaak zullen afwijkende dialogen maar in één taal beschikbaar zijn. In dat geval kan de inhoud van de betreffende record-occurrence bijvoorbeeld zijn:

Figuur 9b: Structuur van de GIRAF-Instellings-Database (GID)





Figuur 10: De werking van het cassettesysteem

lijk van de taalcode een substituu van de afwijkende dialoognaam in het werkrecord (de cassette) geplaatst. Na afloop van het mengproces bevat het werkrecord GIRDSDALG-WR alle dialoognamen die voor de betreffende gebruiker op dat moment relevant zijn. Zowel met afwijkende dialogen als met de gewenste taalcode is rekening gehouden. Het werkrecord GIRDSDALG-WR wordt aan alle dialogen gekoppeld zodat alle dialogen over de voor de gebruiker relevante dialoognamen kunnen beschikken.

Iedere dialoog bevat processen. Deze processen kunnen mededelingen naar het beeldscherm zenden. Met name het zogenaamde \$-MESSAGE-veld speelt hierbij een belangrijke rol. Afwijkende dialogen produceren vaak ook afwijkende mededelingen. Bovendien moeten de processen onafhankelijk zijn van de door de gebruiker opgegeven taalcode. Om die redenen zijn alle mededelingen buiten de processen gehouden en op soortgelijke wijze als de dialoognamen in de GID en GTD opgeslagen. Ook met betrekking tot de mededelingen is een cassettesysteem ontwikkeld. Dit cassettesysteem vult de cassette (het werkrecord) niet direct nadat de gebruiker zich heeft gemeld, maar werkt op procesniveau. Ieder proces van een dialoog dat een mededeling wil versturen bepaalt eerst via het cassettesysteem wat de op dat moment relevante mededeling is.

4.3 Multiple databases

Verwacht wordt dat 'GIRAF' binnen de TH-Delft ten behoeve van tientallen informatiesystemen zal worden gebruikt. De toepassingen kunnen sterk variëren. Zo zijn er enerzijds individuele wetenschappers die hun literatuurreferenties in een GIRAF-TOEPASSINGS-

DATABASE (GTD) opslaan en anderzijds toepassingen die werken met tienduizenden referenties. Gezien het grote aantal toepassingen en het uiteenlopende karakter van de toepassingen is gezocht naar een mogelijkheid om alle GTD's fysiek te scheiden. Het vervaardigen van één grote database werd niet zinvol geacht. In het laatste geval zouden de gebruikers met kleine GTD's zeer vermoedelijk te veel worden gehinderd door gebruikers die 'GIRAF' gebruiken in een informatiesysteem dat dagelijks gedurende lange tijd met (zeer) veel referenties beschikbaar moet zijn. Door de GTD's fysiek te scheiden kan de verdeling van de GTD's over de verza-

Figuur 11: Voorbeeld van een Database Name Table (DNT)

DBNAME	formeel subschema		actueel subschema
GIGTD001	GIVT001A	maps to	GIVT001A
GIGTD002	GIVT001A	maps to	GIVT002A
GIGTD003	GIVT001A	maps to	GIVT003A
.....
GIGTD087	GIVT001A	maps to	GIVT087A
.....

meling schijven die het Rekencentrum van de TH-Delft bezit, beter worden geregeld. Een tweede belangrijk argument is de beheersbaarheid van de GTD's. Door een fysieke scheiding van de GTD's zal het eenvoudiger zijn om één of meer GTD's te reorganiseren. Van sommige GTD's zal vaker een back-up moeten worden gemaakt dan van andere GTD's. Ook dat is simpeler te realiseren als de GTD's gescheiden zijn.

Om een en ander te realiseren zou in het schema dat de database beschrijft een recordtype opgenomen kunnen worden waarvan een record-occurrence de 'owner' is van alle referenties van een toepassing. Het 'CODASYL-report' [10] staat toe dat record-occurrences van één recordtype in verschillende areas worden geplaatst. Door nu ieder record-occurrence in een aparte area te plaatsen en vervolgens aan iedere area een aparte file te koppelen zou de wens van scheiding van alle toepassingen gerealiseerd kunnen worden. IDMS staat echter niet toe dat record-occurrences van één recordtype in verschillende areas worden geplaatst zodat deze oplossing niet gerealiseerd kon worden. Gelukkig biedt IDMS een andere oplossing, die recentelijk ook binnen ADS/ONLINE kan worden gebruikt, namelijk de zogenaamde DBNAME-opdracht. Met behulp van de DBNAME-opdracht kan een programma dat met subschema A is vertaald op uitvoeringstijd worden gekoppeld aan subschema B. Om een en ander mogelijk te maken moet binnen IDMS een zogenaamde Database Name Table (DNT) zijn gedefinieerd (zie figuur 11). De DNT bevat voor 'GIRAF' een aantal ingangen (entries). Per ingang zijn gedefinieerd: een DBNAME, een formeel subschema en een actueel subschema. Binnen een proces van ADS/ONLINE kan met behulp van de DBNAME-opdracht het formele subschema worden 'vervangen' door het betreffende actuele subschema. De opdracht MOVE 'GIGTD087' TO DBNAME zorgt er voor dat via ingang GIGTD087 de DNT wordt benaderd. In de DNT staat aangegeven dat het formele subschema GIVT001A moet worden vervangen door het actuele subschema GIVT087A. Indien er geen DBNAME-opdracht wordt gegeven, dan is het formele subschema bij verstek het actuele subschema. Opgemerkt moet worden dat de formele en actuele subschema's alleen mogen verschillen v.w.b. de page ranges en files. Recordtype- en setbeschrijvingen moeten hetzelfde zijn.

Binnen 'GIRAF' is het zo geregeld dat de DBNAME van een toepassing is opgeslagen in het betreffende toegangsrecord GI-TOEGANGS-REC van de GID (zie figuur 9b). Nadat de gebruiker zich heeft gemeld wordt er met behulp van de DBNAME uit het betreffende GI-TOEGANGS-REC voor gezorgd dat het juiste actuele subschema (en dus ook de juiste database van de multiple databases) 'in stelling' wordt gebracht.

Het werken met bovengenoemde DNT levert een ongewenst neveneffect op. In IDMS is het namelijk mogelijk een relatie tussen een programma en subschema vast te leggen. De Data Base Administrator (DBA) legt die relatie vast om er voor te zorgen dat het betreffende programma uitsluitend met dat subschema kan werken. Het vertalen van het programma met een ander subschema is dan niet mogelijk. Deze koppeling is dus uniek. Het is niet mogelijk aan een programma meerdere subschema's te koppelen. Via het DBNAME-mechanisme kunnen, weliswaar via een omweg, wel meerdere subschema's aan een programma worden gerelateerd. Het is echter

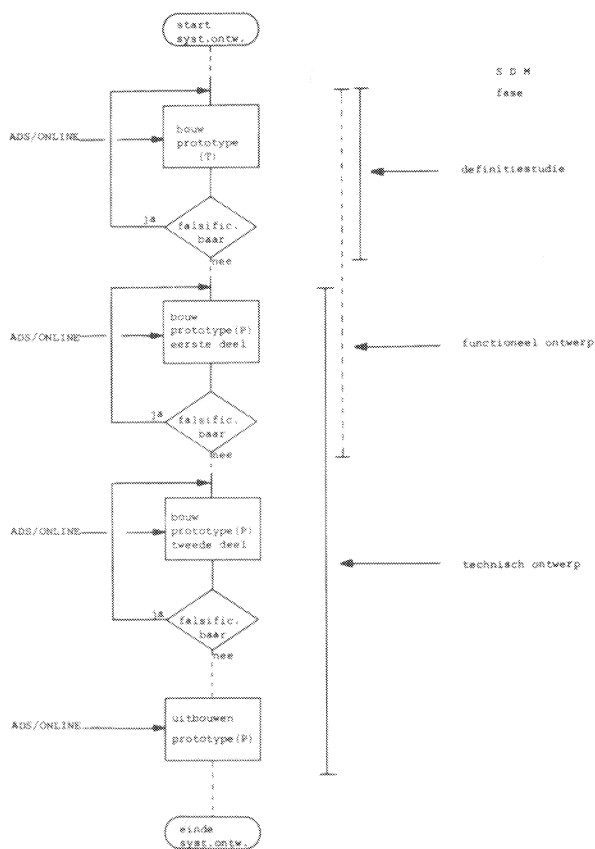
gebleken dat gebruik van het DBNAME-mechanisme alleen mogelijk is als de DBA geen vast subschema aan het programma koppelt. Dit betekent dat het betreffende programma eventueel door een programmeur met een ander subschema kan worden vertaald. Gebruik van het DBNAME-mechanisme gaat dus ten koste van een stukje beveiliging. Het is wenselijk om ook bij gebruik van het DBNAME-mechanisme een koppeling tussen een programma en een vast (formeel) subschema mogelijk te maken.

4.4 Prototyping

Bij de beschrijving van ADS/ONLINE [1] wordt aangegeven dat dit ontwikkeltool een goed hulpmiddel is om prototyping-activiteiten te ondersteunen. In deze paragraaf worden de ervaringen die zijn opgedaan met prototyping-activiteiten samengevat. Bij de beschouwingen zullen de artikelen van Sol [23] en Van Beek [5] worden betrokken. Van Beek stelt dat een werkwijze als SDM niet bruikbaar is voor een op prototyping gefundeerde systeemontwikkeling. Hij is van mening dat prototyping vooral geschikt is voor systeemontwikkelingsmethoden die zijn gebaseerd op de 'data analyse'-methode. In zijn artikel beschrijft Van Beek de Prototyping Development Methodology (PDM). Deze methode is vooral geschikt voor projecten waar de informatiebehoefte nogal onduidelijk is, zoals MIS en DSS. Van Beek concludeert dat bij administratieve automatisering de informatiebehoefte redelijk goed bekend is en daarom de baten van prototyping minder groot zijn. Bij de ontwikkeling van de systemen 'GIRAF' en 'GEMS' is met behulp van ADS/ONLINE op bescheiden schaal gebruik gemaakt van prototyping. Daarbij kunnen twee niveaus worden onderscheiden (zie figuur 12):

4.4.1 Prototype (T)

In de definitiestudie is met ADS/ONLINE een aantal dialogen alsmede de communicatie tussen deze dialogen geconstrueerd. De gegevens die een rol speelden in dit prototype waren gefingeerd. Het prototype bevatte geen database-activiteiten. Met het prototype werd beoogd de eindgebruiker/opdrachtgever op zeer korte termijn, enkele dagen, een aanschouwelijke indruk te geven over de wijze waarop zijn informatievoorziening gerealiseerd zou kunnen worden. Dit prototype had een tijdelijk karakter en wordt kortweg prototype (T) genoemd. Het heeft ook tijdens het functioneel ontwerp een rol gespeeld. Bij de systemen 'BIBINFO' (1976) [12], 'Aanvraagstelsysteem' (1980) [13], 'GIRAF' (1984) [14] en 'GEMS' (1984/1985) [16] heeft namelijk het ontwerpen van de schermbeelden en de communicatie tussen de schermbeelden plaatsgevonden tijdens de fase functioneel ontwerp. Dit in tegenstelling tot SDM, die aangeeft dat het ontwerpen van schermbeelden in de fase technisch ontwerp moet plaatsvinden. Persoonlijke opvattingen van de ontwerper, die aansluiten bij het motto (Van Rees [22]): 'De ontwerper ontwerpt en niet de methode', speelden hierbij de belangrijkste rol. Het beoordelen van op papier vastgelegde schermbeelden(dialogen) door eindgebruikers is een moeilijke zaak. Gebleken is dat het prototype (T) in deze fase een rol kan spelen. Door demonstraties met het prototype (T) is de eindgebruiker beter in staat om de op papier vastgelegde schermbeelden(dialogen) te kunnen beoordelen en eventueel te falsificeren.



Figuur 12: ADS/ONLINE in relatie tot prototyping en SDM

4.4.2 Prototype (P)

Een van de eerste activiteiten die in de fase technisch ontwerp is verricht was het bouwen van het prototype (P). Alle schermbeelden en schermbeelddialogen werden met behulp van ADS/ONLINE vervaardigd. De letter P geeft aan dat het een 'permanent prototype' betreft. Dit permanente karakter betekent dat het prototype in een latere fase wordt uitgebouwd tot een onderdeel van het gewenste informatiesysteem. De bouw van prototype (P) is uiteraard veel meer werk dan de bouw van een prototype (T). Opgemerkt kan nog worden dat, indien er voldoende menskracht beschikbaar is, reeds tijdens de fase functioneel ontwerp een gedeelte van prototype (P) kan worden gebouwd. Dit deel van prototype (P) kan dan de rol van het prototype (T) in het functioneel ontwerp overnemen. Ook bij het prototype (P) is gewerkt met gefingeerde gegevens zonder database-activiteiten. Na het realiseren van prototype (P) konden de eindgebruikers er uitgebreid mee 'spelen'. Dit heeft geleid tot enkele aanpassingen. Nadat de eindgebruikers het prototype (P) hadden goedgekeurd (d.w.z. zij konden geen fouten meer vinden, het aangepaste prototype kon voorlopig niet gefalsificeerd worden) konden de resterende bouwactiviteiten worden verricht.

Aan de hand van de opgedane ervaringen kan worden geconcludeerd dat:

- bij administratieve automatisering prototyping-activiteiten kunnen leiden tot een hogere betrokkenheid van de eindgebruiker/opdrachtgever;
- de ondersteuning van de fasen definitiestudie en functioneel ontwerp met een prototype de eindge-

bruiker ook meer mogelijkheden biedt zijn informatiebehoefte juist en volledig te specificeren. Het specificeren zal vaak door middel van enkele 'iteratie-stappen' tot stand komen;

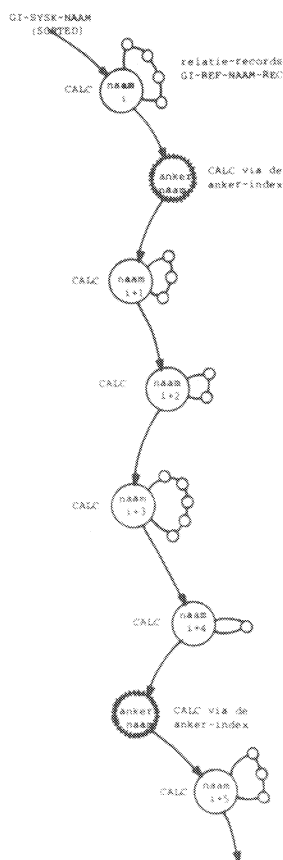
- sommige eindgebruikers het zeer merkwaardig vinden dat, nadat zij een goed functionerend prototype (P) hebben zien werken, zij nog lange tijd moeten wachten voordat het informatiesysteem wordt opgeleverd;
- het met eindgebruikers in zeer korte tijd opstellen van prototype (T) ertoe kan leiden dat het overzicht uit het oog wordt verloren. Sol [23] noemt dat het gevaar voor 'tunnelvisie';
- ADS/ONLINE een goed hulpmiddel is om deze prototyping-activiteiten te ondersteunen.

5 INDICES

Op twee plaatsen in het programmatuursysteem 'GIRAF' spelen indices een rol, namelijk bij de toegang op naam en bij de toegang op attribuutcode + attribuutwaarde.

5.1 Toegang op naam

Gegevensgroep 1 bevat doorgaans de namen van een referentie, bijvoorbeeld de auteursnamen van een literatuurreferentie, de artikelnaam van een produkt, de naam van een persoon die lid is van een vereniging of deelneemt aan een congres, etc. 'GIRAF' biedt de mogelijkheid om aan de hand van een naam de referenties te selecteren die de betreffende naam in gegevensgroep 1 opgenomen hebben. Zo zullen bijvoorbeeld na het intoetsen van de naam 'Pietersen, K.' alle in de GTD voorkomende literatuurreferenties van de heer K. Pietersen op het beeldscherm worden getoond. Indien de naam 'Pietersen, K.' niet voorkomt in de GTD, wordt op het beeldscherm een lijst van namen getoond rondom de naam 'Pietersen, K.'. De gebruiker kan dan kijken of hij wellicht een spelfout heeft gemaakt en kan alsnog, op eenvoudige wijze, de juiste naam kiezen. Ook bij het intoetsen van een generieke naam kan een dergelijke lijst van namen worden geproduceerd. Om zo'n (alfabetisch gesorteerde) serie namen op het beeldscherm te kunnen laten zien is het noodzakelijk om alle namen van de GTD op de een of andere wijze in een gesorteerde volgorde beschikbaar te hebben. In 'GIRAF' is daartoe een SORTED SET gedefinieerd met als 'owner' het record-occurrence van recordtype GI-SYSKEN-REC (zie figuur 9c), dat de kenmerkende gegevens voor de betreffende GTD bevat, en als 'members' de record-occurrences van het recordtype GI-NAAM-REC dat alle namen, die tot de GTD behoren, bevat. Op die manier kan de gewenste serie namen worden opgespoord. Hierbij komt wel een probleem om de hoek kijken. De record-occurrences van GI-NAAM-REC zijn namelijk CALC opgeslagen. Daarmee is een bestaande naam direct benaderbaar en kan de toegang via naam op zeer snelle wijze worden gerealiseerd. Het probleem is nu dat de genoemde SORTED SET een gesorteerde CALC-TO-CALC SET is. Het doorlopen van een grote gesorteerde CALC-TO-CALC SET kost onaantvaardbaar veel tijd omdat het lezen van elk member in de set een 'echte' leesopdracht betekent. Het zoeken in een set van 10 000 namen zou gemiddeld maar liefst 5000 leesopdrachten betekenen. De geëigende oplossing in dergelijke situaties is het ma-



Figuur 13: Voorbeeld van 'ankers' in de gesorteerde set met naam-records

ken van een index. IDMS biedt mogelijkheden om te kunnen werken met indices en heeft daarvoor het zogenaamde 'Sequential Processing Facility (SPF)' [18] beschikbaar. Na bestudering van SPF waren er nogal wat vraagtekens, met name bij het optreden van veel mutaties, over de efficiëntie van dit produkt. Uit een tiental gesprekken met Amerikaanse IDMS-gebruikers die SPF reeds gebruikten, bleek weinig waardering voor SPF. Ook Cullinet erkende dat SPF verbeterd kon worden door in het najaar 1983 aan te kondigen dat een nieuwe, met IDMS geïntegreerde, index-methode werd ontwikkeld.

Zoals in hoofdstuk 1 reeds vermeld is, moesten de technische beslissingen inzake 'GIRAF' in het najaar 1983 worden genomen op basis van de op dat moment beschikbare mogelijkheden. De conclusie werd getrokken dat SPF voor 'GIRAF' niet geschikt was. Daarom is binnen 'GIRAF' een eigen index-methode toegepast. De belangrijkste uitgangspunten bij de ontwikkeling van de index-methode waren enerzijds de noodzakelijke snelheid waarmee een lijst van namen kan worden geproduceerd en anderzijds de gewenste eenvoud van de index-methode. Daarom is gekozen voor een zogenaamde anker-index die niet on-line gemuteerd wordt. In de GTD is een recordtype GI-ANKNAAM-REC opgenomen. De record-occurrences van GI-ANKNAAM-REC bevatten zogenaamde ankers. Een anker wordt afgeleid van een bestaande naam en wordt opgenomen in zowel de index als in het recordtype GI-NAAM-REC (zie fi-

guur 9c). Er wordt voor gezorgd dat het anker altijd ongelijk is aan een naam die in de GTD voorkomt of kan voorkomen. De anker-index van een GTD wordt 's nachts met behulp van een batch-programma aangemaakt. Het aantal ankers is afhankelijk van een invoerparameter van het batch-programma dat de anker-index aanmaakt. Deze invoerparameter geeft het aantal namen aan dat tussen twee ankers voorkomt.

Indien nu een lijst van namen moet worden geproduceerd aan de hand van een generieke naam of een niet in de GTD aanwezige naam kan met behulp van de index GI-ANKNAAM-REC, die uit een moeder-index en een aantal blok-indices bestaat, het anker worden bepaald dat dicht in de buurt ligt van de naam die volgt op de ingevoerde naam. Aangezien het anker ook in de SORTED SET is opgenomen (zie figuur 13) kan via CALC dit anker direct worden bereikt en kan de gewenste serie namen worden bepaald zonder veel extra invoeropdrachten.

Samenvattend kunnen over de geschetste anker-indexmethode de volgende opmerkingen worden gemaakt:

- de ankers zijn niet bestaande namen en worden dus nooit on-line gemuteerd;
- het muteren van de anker-index gebeurt 's nachts met behulp van een batch-programma. Daarbij wordt de oude anker-index vervangen door een nieuwe. De ankers worden afgeleid van namen die op dat moment in de GTD aanwezig zijn en zijn dus geheel afgestemd op de betreffende GTD;
- met behulp van een parameter wordt het aantal namen tussen twee ankers bepaald. Indien voor een GTD zeer snelle responsietijden zijn vereist dan kan deze parameter bijvoorbeeld 3 of 5 zijn. Indien de responsietijden minder kritisch zijn kan een waarde 10 of groter worden genomen. Hoe kleiner de parameterwaarde hoe groter het aantal ankers;
- tijdens het opbouwen van een GTD zal de anker-index regelmatig vernieuwd moeten worden. Het sein daartoe wordt gegeven doordat de responsietijden substantieel gaan toenemen. Desgewenst kan met behulp van een hulpprogramma de SORTED SET GI-SYSK-NAAM (zie figuur 9c) regelmatig worden geïnspecteerd. Een groot voordeel van de anker-index-methode is dat als een GTD op grootte is gebracht, de anker-index doorgaans niet meer hoeft te worden vervangen. Mutaties op de namen kunnen plaatsvinden zonder dat de anker-index hoeft te worden gewijzigd.

Geconcludeerd kan worden dat de anker-index-methode een index oplevert die zowel met betrekking tot de namen van de ankers als de (met de index samenhangende) responsietijden aangepast is aan de gebruiker, on-line niet gemuteerd wordt en doorgaans na verloop van tijd in het geheel niet meer hoeft te worden vernieuwd.

5.2 Toegang op attribuutcode + attribuutwaarde

'GIRAF' biedt mogelijkheden om on-line een verzameling referenties te selecteren met behulp van een codesysteem (zie figuur 1). Het codesysteem bestaat uit een (in principe onbepaald) aantal attribuutcodes en per attribuutcode een attribuutwaardenverzameling. Evenals dat bij de namen het geval is, is per attribuutcode + attribuutwaarde vastgelegd welke referenties aan deze combinatie voldoen en is het recordtype GI-ATTRWD-REC met CALC (attribuutcode + attribuutwaarde) direct be-

naderbaar (zie figuur 9c). Selecteren op basis van een attribuutcode en een attribuutwaardeninterval is eveneens mogelijk. Ook hier ontstaat er een probleem als de grenzen van het attribuutwaardeninterval geen binnen de GTD bekende attribuutwaarden zijn. Om in dergelijke situaties toch snel de gewenste verzameling referenties te kunnen samenstellen is eveneens een anker-index geconstrueerd. Hoewel de constructie iets ingewikkelder is dan de anker-index voor namen zijn de uitgangspunten alsmede de voordelen en nadelen (ruimte voor de ankers) dezelfde.

6 CONCLUSIES

Uit de ervaringen met het ontwikkeltool ADS/ONLINE kunnen de volgende conclusies worden getrokken:

- Door de dialoogstructuur kan met ADS/ONLINE in het technisch ontwerp op een hoger abstractieniveau worden gewerkt dan met TP-monitoren zoals IDMS/DC en CICS/VS.
- De DialoogCommunicatieDiagrammen (DCD's) sluiten nauw aan bij de SchermbeeldCommunicatieDiagrammen (SCD's). Daardoor zal de overgang van functioneel ontwerp naar technisch ontwerp **soepeler** kunnen verlopen. Bovendien kunnen opdrachtgevers, die het functioneel ontwerp kennen, de vorderingen van het technisch ontwerp makkelijker volgen.
- Het ontwikkelen van programmatuursystemen met ADS/ONLINE zal de fase technisch ontwerp/programmering substantieel verkorten. Kwantitatieve gegevens zijn nog moeilijk te geven.
- De mogelijkheden van ADS/ONLINE, gecombineerd met een cassettesysteem zoals binnen 'GIRAF' is ontwikkeld, maken het mogelijk om ruim aandacht te besteden aan de flexibiliteitsaspecten 'open-ended' en 'adaptability'. Het leveren van maatconfectiesystemen behoort tot de mogelijkheden.
- Het werken met multiple databases kan binnen ADS/ONLINE op een soepele manier worden geregeld. Daarmee kunnen gebruikersdatabases fysiek worden gescheiden.
- ADS/ONLINE is een goed hulpmiddel om prototyping-activiteiten te ondersteunen. Zowel in de fasen definitiestudie, functioneel ontwerp als technisch ontwerp kunnen prototypes (T) en prototypes (P) een rol spelen. Dit kan leiden tot een hogere betrokkenheid van de eindgebruiker/opdrachtgever. Deze is bovendien beter in staat om zijn informatiebehoefte juist en volledig te specificeren. Systeemontwerpers moeten bedacht zijn op het gevaar voor 'tunnelvisie'.
- De gebruiksvriendelijkheid van het produkt On Line Mapping moet worden verbeterd.
- Hoewel Cullinet Software ernaar streeft om alle produkten te integreren zodat ze als het ware één geheel vormen, worden systeemontwerpers/programmeurs geconfronteerd met produkten die op gebruiksniveau niet altijd volledig op elkaar zijn afgestemd.

7 LITERATUUR/REFERENTIES

1. Application Development System / Online Summary Description. Cullinet Software, 1983
2. Application Development System / Online Application Design Guide, release 1.1. Cullinet Software, augustus 1983
3. Application Development System / Online User's Guide, release 1.1. Cullinet Software, augustus 1983
4. Application Development System / Online Reference Guide, release 1.1. Cullinet Software, september 1983
5. Beek, J. van; Een systeemontwikkelingsmethode gebaseerd op prototyping. In: *Informatie*, jaargang 24 nr. 12, december 1982, pp. 702-710
6. Bergstra, J. A. en J. J. Smeets; Formele ontwikkeling van schermverloopschema's en interactieve programma's. In: *Informatie*, jaargang 26 nr. 4, april 1984, pp. 318-324
7. Brussaard, B. K.; Collegedictaat 'Informatiesystemen'. Vakgroep Informatica. Technische Hogeschool Delft
8. Chapin, Ned; New format for flowcharts. In: *Software practice and experience*, vol. 4, 1974, pp. 341-357
9. Craenen, G.; Praten met de computer. Software voor eindgebruikers. Aula, 727, Spectrum, 1983
10. Database task group (DBTG). CODASYL report april 71
11. Dijk, A. J. van; 'BIBLIOSYSTEM'. Computerhulp bij het samenstellen van bibliografieën. In: *Informatie*, jaargang 13 nr. 10, oktober 1971, pp. 448-455
12. Dijk, A. J. van; 'BIBINFO'; Een real-time retrieval systeem. Afstudeerverslag, vakgroep Informatica TH-Delft, Memorandum no. RC-DMS-76002, november 1976, 236 pag.
13. Dijk, A. J. van; Ervaringen met het COMMAND-level van CICS/VS bij de ontwikkeling van een geautomatiseerde bibliotheek-toepassing. In: *Informatie*, jaargang 22 nr. 4, april 1980, pp. 313-322
14. Dijk, A. J. van; 'GIRAF'; Een information retrieval systeem voor universiteiten, hogescholen en andere pluri-forme organisaties. In: *Informatie*, jaargang 26 nr. 6, mei 1984, pp. 445-455
15. Dijk, A. J. van en R. de Roos; 'GIRAF' - General Information Retrieval Facilities. Functioneel ontwerp. Technische Hogeschool Delft, Rekencentrum. Memorandum no. RC-ISO-82012, december 1982, 213 pag.
16. Dijk, A. J. van, R. P. Karlas, P. Makkes en R. J. J. van Schie; 'GEMS' - Geautomatiseerd Magazijnadministratie- en voorraadbeheersysteem. Functioneel ontwerp. Technische Hogeschool Delft, Rekencentrum Memorandum no. RC-ISO-84004, juni 1984, 256 pag.
17. Eilers, H. B.; Systeemontwikkeling volgens SDM, Academic Service Den Haag, 1980, ISBN 90 623 3043 6
18. IDMS. Sequential Processing Facility. Cullinet Software, april 1983
19. IDMS-DC, Concepts and Facilities. Cullinet Software, april 1979
20. Janssens, H.; Gestructureerde conversaties. In: *Informatie*, jaargang 24 nr. 3, maart 1982, pp. 152-160
21. Ooninx, J. A. M.; Waarom falen informatiesystemen nog steeds? Samsom, Alphen a/d Rijn, 1982, ISBN 90 140 3201 3
22. Rees, J. R. van; Ontwikkelingsmethoden 22. De methode doet het niet. In: *Informatie*, jaargang 24 nr. 2, februari 1982, pp. 81-93
23. Sol, H. G.; Ervaringen rond prototyping. In: *Informatie*, jaargang 26 nr. 3, maart 1984, pp. 236-242
24. De auteur van dit artikel is dank verschuldigd aan de collega's R. de Roos en R. J. J. van Schie voor de vele interessante discussies die hebben bijgedragen aan de totstandkoming van dit artikel.