Nederlandse organisatie
voor toegepast
natuurwetenschappelijk
onderzoek

FEL

Fysisch en Elektronisch
Laboratorium TNO

rapport no.
FEL-89-B262

exemplaar no.
7

Verslag van de 15e conferentie over Very Large Data Bases van 22 t/m 25 augustus 1989 te Amsterdam

Nederlandse organisatie
voor toegepast
natuurwetenschappelijk
onderzoek

TNO-rapport

FEL

Fysisch en Elektronisch
Laboratorium TNO

Postbus 96864
2509 JG 's-Gravenhage
Oude Waalsdorperweg 63
's-Gravenhage

Telefoon 070 - 26 42 21

89-4702

1

rapport no.
FEL-89-B262

exemplaar no.
7

titel

Verslag van de 15e conferentie over Very Large Data Bases van 22 t/m 25 augustus 1989 te Amsterdam

auteur(s):

Ir. M.C. van Hekken

Ir. J.J.C.R. Rutten

rubricering

titel	: ongerubriceerd
samenvatting	: ongerubriceerd
rapport	: ongerubriceerd

oplage	: 26
aantal bladzijden	: 17
aantal bijlagen	: 1
datum	: 4 oktober 1989

TNO

rapport no. : FEL-89-B262

titel : *Verslag van de 15e conferentie over* Very Large Data Bases van 22 t/m 25 augustus te Amsterdam

auteur(s) : Ir. M.C. van Hekken, Ir. J.J.C.R. Rutten

instituut : Fysisch en Elektronisch Laboratorium TNO

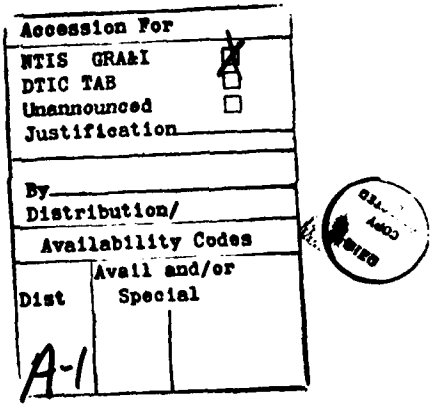
datum : 4 oktober 1989

hdo-opdr.no. :

no. in iwp '89 : 704

SAMENVATTING

Dit rapport bevat een verslag van de 15e conferentie over Very Large Data Bases (VLDB) die van 22 t/m 25 augustus 1989 in Amsterdam werd gehouden. Het belangrijkste gedeelte bestaat uit een samenvatting van een aantal lezingen die op de conferentie zijn gehouden. Het rapport wordt besloten met een aantal trends op database-gebied, zoals die uit de conferentie naar voren zijn gekomen.



report no.	:	FEL-89-B262
title	:	Report on the 15th conference on Very Large Data Bases in Amsterdam from August 22 till August 25 1989
author(s)	:	M.C. van Hekken MSc, J.J.C.R. Rutten MSc
institute	:	TNO Physics and Electronics Laboratory
date	:	October 4, 1989
NDRO no.	:	
no. in pow '89	:	704

ABSTRACT

This document contains a report on the 15th conference on Very Large Data Bases (VLDB) that was held in Amsterdam from August 22 till August 25 1989. The main part consists of abstracts from a number of tutorials, held on this conference. The report concludes with a number of trends in the area of databases, as have become evident from the conference.

1 INLEIDING

Van 22 t/m 25 augustus 1989 werd in de RAI in Amsterdam voor de 15e keer de conferentie over Very Large Data Bases (VLDB) gehouden. De VLDB conferentie is een jaarlijks terugkerend internationaal evenement waarin een overzicht wordt gegeven van de stand van zaken in het wetenschappelijk en praktijkgericht onderzoek op het gebied van grote database systemen. Het internationale karakter van de conferentie wordt benadrukt door het grote aantal deelnemende landen uit verschillende werelddelen. Dit jaar kwamen ruim 500 deelnemers uit zo'n 30 verschillende landen naar Nederland, dat voor het eerst de organisatie in handen had. Algemeen voorzitter van de conferentie was prof. dr. R.P. van de Riet van de Vrije Universiteit (VU) Amsterdam.

Onder de deelnemers waren twee medewerkers van groep 2-1 van het FEL-TNO. Dit rapport bevat hun verslag van de conferentie. In hoofdstuk 2 is een overzicht van het complete programma van de conferentie opgenomen. Hoofdstuk 3 bevat een samenvatting van de bijgewoonde lezingen. Het laatste hoofdstuk geeft een aantal conclusies en trends op het gebied van (grote) databases, zoals die uit de conferentie naar voren zijn gekomen.

2 PROGRAMMA VAN DE CONFERENTIE

De VLDB conferentie 1989 nam in totaal 4 dagen in beslag. Op dinsdag 22 augustus was er een middagprogramma, de twee daaropvolgende dagen een ochtend- en een middagprogramma, terwijl de conferentie op 25 augustus werd afgesloten met een ochtendprogramma. Met uitzondering van de eerste middag, die meer in het teken stond van de ontvangst en van een aantal inleidende verhalen, hadden de deelnemers steeds de keuze uit drie sessies die tegelijkertijd werden gehouden (zie tijdschema in bijlage A). Twee van de drie sessies waren wat meer wetenschappelijk van aard. In blokken van anderhalf uur werden daarin twee of drie lezingen met betrekking tot een specialistisch onderwerp gehouden. In een aantal gevallen vond er een discussie over een bepaald onderwerp plaats in de vorm van een panel. Naast de twee wetenschappelijke sessies was er een sessie waarin een aantal meer algemene tutorials werden gehouden. Deze tutorials, die steeds twee blokken van anderhalf uur duurden, bestreken een groot gedeelte van het huidige onderzoeksveld in de database-wereld.

Een overzicht van het complete programma is te vinden in bijlage A. Hierin zijn voor iedere sessie de onderwerpen en de sprekers aangegeven. De medewerkers van het FEL hebben voornamelijk de tutorials bijgewoond. Op de inhoud van deze tutorials wordt in het volgende hoofdstuk ingegaan.

3 HET TUTORIAL-PROGRAMMA

3.1 Algemeen

De sessie waarin het tutorial-programma werd afgewerkt trok over het algemeen de meeste deelnemers. Enerzijds omdat de meeste verhalen van dit programma een wat minder theoretisch karakter hadden dan de kortere verhalen uit de andere sessies. Anderzijds omdat een aantal gerenommeerde sprekers op het programma stonden, zoals professor G. Nijssen en C. Date. Het doel van de tutorials was het informeren van de toehoorders omtrent de stand van zaken op het huidige onderzoeksgebied van databases. Dat dit gebied vrij breed is bleek uit de aard van de lezingen. De verhalen van Nijssen en Date kwamen vanuit de 'traditionele' relationele database-praktijk, M. Brodie en J. Mylopoulos hielden een lezing over het integreren van AI en database technieken, terwijl de tutorial van S. Zdonik object-oriented databases als onderwerp had. De meest theoretische lezing was van K. Apt en ging over top-down versus bottom-up computing in deductieve databases. In de volgende paragrafen volgt een korte beschrijving van de tutorials. Voor de complete inhoud van de lezingen in de andere twee sessies wordt verwezen naar [VLDB89].

3.2 The foreign key Saga; C.J. Date

Het foreign key principe is een van de belangrijkste principes in een relationele database, evenals het sterk gerelateerde primary key principe. Voor een primary key gelden de volgende eisen:

- de primary key van een relatie/tabel (bestaande uit een of meer attributen) kan voor twee verschillende records nooit dezelfde waarde hebben (primary key is unieke identificatie van een record),
- de primary key is minimaal, d.w.z. geen enkel attribuut kan uit de primary key verwijderd worden zonder de vorige eis geweld aan te doen,
- geen enkel attribuut van de primary key mag de waarde NULL (onbekend) hebben (entiteitsintegriteit).

Voor een foreign key van een relatie/tabel (bestaande uit een of meer attributen) geldt dat de waarde van deze foreign key gelijk moet zijn aan:

- de waarde van de primary key van een bepaald record uit de relatie/table waarnaar verwezen wordt, òf
- de waarde NULL (onbekend).

Een relationele database voldoet aan de referentiële integriteit indien iedere foreign key waarde aan de bovenstaande eis voldoet. Het belang van de primary key en foreign key principes is gelegen in de mogelijkheid die ze bieden om de structuren van objecten in de database zelf vast te leggen en deze kennis buiten de applicaties te houden.

Het lijkt overdreven om aan een onderwerp als foreign keys een lezing van drie uur te besteden, omdat de principes op zichzelf vrij eenvoudig zijn. Niettemin zijn ze nog steeds een bron van misverstanden en discussies en worden ze door geen enkele van de huidige 'Relationele' DataBase Management Systemen (RDBMS) volledig ondersteund. De spreker wist de beschikbare tijd dan ook moeiteloos te vullen in een poging een aantal misverstanden weg te nemen en een aantal problemen te schetsen. Het verschaffen van duidelijkheid op het gebied van foreign keys en het zo volleding mogelijk ondersteunen van dit principe in RDBMS is belangrijk uit het oogpunt van integriteit, bruikbaarheid, toegankelijkheid en performance van grote databasesystemen.

Een aantal van de belangrijkste gedane uitspraken over regels, eisen en problemen t.a.v. foreign keys en foreign key support zijn:

- Primary keys en corresponderende foreign keys moeten op hetzelfde domein gedefinieerd worden. Geen van de huidige RDBMS biedt volledige ondersteuning van domeinen. Ook zonder dit is echter een zinvolle foreign key support mogelijk.
- Het eisen van referentiële integriteit voor een relationele database stelt ook (niet altijd triviale) eisen c.q. beperkingen aan de operatoren die de toestand van de database veranderen, met name het toevoegen van nieuwe records in tabellen die d.m.v. een foreign key verwijzen naar andere tabellen en het wijzigen c.q. verwijderen van records in tabellen waarnaar vanuit andere tabellen wordt verwezen. Er is nog geen eensluidendheid over de vraag welke vrijheden voor deze operatoren zouden moeten worden toegestaan en geimplementeerd in RDBMS.

- Het is een misverstand om foreign keys te zien als pointers. Foreign keys bevinden zich op een hoger abstractieniveau dan pointers. Ze zijn implementatie-onafhankelijk, kunnen bestaan uit meerdere attributen en vereisen geen speciale operatoren.
- Volgens de integriteitsregel voor entiteiten mag geen enkel attribuut van een primary key de waarde NULL (onbekend) aannemen, omdat er in een relationele database geen entiteiten mogen voorkomen die niet geïdentificeerd kunnen worden. Volgens de referentiële integriteitsregel voor foreign keys geldt dat òf geen enkel attribuut de waarde NULL moet hebben òf alle attributen. Omdat een foreign key als een ondeelbare identificatie van een entiteit moet worden beschouwd is het niet toegestaan dat een gedeelte van de attributen van een foreign key de waarde NULL hebben.
- Vanwege de hierboven beschreven regel van de ondeelbaarheid van foreign keys is het eveneens aan te bevelen om het gebruik van overlappende foreign keys in een relatie/tabel te beperken c.q. vermijden. Niet-samengestelde (primary and foreign) keys, d.w.z. keys die uit slechts een attribuut bestaan, kunnen hiervoor een betere garantie bieden.
- Er is nog veel discussie over de vraag of het toegestaan moet zijn om met een foreign key naar meerdere tabellen te wijzen, bijvoorbeeld in het geval van subtypes die in meerdere tabellen zijn opgeslagen. Mede vanwege de reeds eerder genoemde regels op het gebied van database-operatoren (het behouden van referetiële integriteit) acht de spreker het verwijzen naar meerdere tabellen niet wenselijk. Hij prefereert een oplossing met een zgn. 'master'-tabel waarnaar dan verwezen kan worden.

Tot slot gaf de spreker een overzicht van enkele ontwikkelingen op het gebied van foreign key c.q. integrity support. De SQL standaard is al enige tijd bezig met een poging de database integriteit beter te waarborgen, via het data definitie gedeelte van de taal. Begin 1988 werd de 'public review' periode afgesloten van de zgn. Integrity Enhancement Feature (IEF), een interim extensie van SQL1. De belangrijkste tekortkomingen van IEF zijn o.a. dat primary keys niet verplicht zijn, dat foreign keys behalve naar primary keys ook naar candidate keys kunnen wijzen en dat foreign keys niet ondeelbaar zijn, d.w.z. gedeeltelijk de waarde NULL kunnen hebben. Ook in de SQL2 definitie zitten een aantal eigenschappen t.b.v. referentiële integriteit, o.a. de mogelijkheid om expliciet (primary en foreign) keys te definiëren. Niettemin gelden ook

voor deze SQL2 definitie ongeveer dezelfde tekortkomingen. In DB2 versie 2 van IBM (met SQL/DS) wordt wel afgedwongen dat foreign keys altijd naar primary keys verwijzen, maar ook daar geldt dat primary keys optioneel zijn en foreign keys niet ondeelbaar. Zowel voor IEF, SQL2 als DB2 (SQL/DS) geldt dat er wel enige ondersteuning is om referentiële integriteit ook via de toegestane operaties op de database af te dwingen, maar dat het aantal regels dat voor die operatoren (met name update en delete) wordt ondersteund nog te gering is.

Al met al kan geconcludeerd worden dat het belang van foreign keys in een relationele database algemeen wordt onderkend, maar dat er nog steeds geen volledige overeenstemming bestaat over de eisen waaraan foreign key support zou moeten voldoen.

3.3 Top-down versus bottom-up computing in deductive databases; K.R. Apt

Wanneer men gecompliceerde soorten van informatie wil opslaan en opvraagbaar maken zijn de traditionele databases (waaronder relationele databases) op zichzelf vaak een onvoldoende oplossing. Steeds vaker doet zich de situatie voor dat een database in staat moet zijn om complexere vragen te beantwoorden door middel van logische redeneerstappen (afleidingen). Dergelijke databases noemt men deductieve databases.

In deductieve databases zijn in eerste instantie twee vormen van afleidingen te onderkennen: bottom-up computing en top-down computing. Bottom-up computing komt neer op het bottom-up evalueren van vragen (queries) aan de database. Dit gebeurt door het afleiden van nieuwe feiten uit bepaalde opgeslagen feiten. Top-down computing is gebaseerd op het SLD-resolution mechanisme dat wordt gebruikt in zgn. 'logic' programma's zoals PROLOG programma's. In de tutorial werd aan beide afleidingsvormen aandacht besteed.

In het gedeelte over bottom-up computing werden een aantal van de bekendste algoritmen van deze vorm van afleiden behandeld. Deze algoritmen houden zich bezig met het berekenen van de afsluiting van een eindige verzameling feiten onder een eindige verzameling regels. Door het bekijken van de eigenschappen van additieve operatoren in Boolean algebra blijkt het mogelijk de correctheid van de algoritmen te bewijzen. Voor een gedetailleerde theoretische uitwerking wordt verwezen naar [Apt1].

In het gedeelte over top-down computing lag, naast een uitleg van het SLD-resolution mechanisme, de nadruk op 'loop checking' mechanismes. In logic programma's zoals PROLOG programma's komt vaak divergentie voor omdat de (PROLOG) interpreter tijdens het zoekproces in een loop terecht kan komen. Het probleem van het ontdekken van mogelijke divergentie is onbeslisbaar omdat PROLOG alle eigenschappen van de recursie theorie bezit, d.w.z. er bestaan geen complete loop checks voor alle PROLOG programma's. Er bestaan echter een aantal soorten loop checks die voor een bepaalde klasse van (PROLOG) programma's (restricted programs) compleet zijn. In deze klasse is slechts een beperkte vorm van recursie toegestaan. Het SLD-resolutie mechanisme gecombineerd met een variant/instantie van een bepaald soort loop check zorgen voor een correcte berekening van queries in deze 'restricted' deductieve databases. Voor de gedetailleerde theoretische uitwerking wordt verwezen naar [Apt2].

3.4 An effective design method for relational databases; G.M. Nijssen

Voor de meeste informatiesystemen is een (relationele) database een van de belangrijkste componenten. Dit betekent dat het database ontwerp op zijn beurt een van de belangrijkste aspecten is in het ontwerptraject van deze informatiesystemen. Tot nu toe is Normalisatie de meest gebruikte methode om tot een relationeel database ontwerp te komen. In deze methode wordt een database beschreven als een aantal relaties. Op deze relaties worden een aantal normalisatie-stappen uitgevoerd totdat ze in de zgn. 3e, 4e of 5e NV (normaalvorm) zijn. Een andere methode die nogal eens wordt gebruikt is het specificeren van een Entity Relationship data model, dat vervolgens gebruikt wordt om tot genormaliseerde relaties te komen.

De spreker is van mening dat dit soort methoden in de praktijk maar matig voldoen. Ze zijn te ingewikkeld om te leren en de kwaliteit van het resultaat laat vaak te wensen over. Vandaar dat hij een methode prefereert die beter aansluit bij de praktijk, nl. de dagelijkse handelwijze van mensen. Kortweg komt deze methode er op neer dat de toekomstige gebruikers van informatiesystemen zelf een veel groter aandeel in het ontwerpproces van de (relationele) database krijgen. Hiertoe nemen deze gebruikers een paar significante voorbeelden van hun dagelijkse werkzaamheden ter hand. Zij weten immers het beste wat ze elke dag doen. Deze voorbeelden worden vervolgens expliciet verwoord ('verbalize')

en dienen zo als belangrijkste basis voor het conceptuele schema van de te ontwerpen database. Tenslotte wordt het conceptuele schema gevalideerd en kunnen er constraints aan worden toegevoegd, waarna het relationele schema van de database kan worden vervaardigd.

De belangrijkste stap in de hierboven beschreven methode is het kiezen van relevante voorbeelden en het expliciet verwoorden hiervan zodat geen informatie verloren gaat. Een hulpmiddel hierbij is de zgn. 'Aunt Nellie Heuristic' of 'Tante Nellie Methode' (TNM):

> 'Neem het voorbeeld ter hand. Ga ervan uit dat je het voorbeeld over de telefoon moet voorlezen aan je tante Nellie die niets weet van de specifieke feiten waarover ze gebeld wordt. Omdat je alleen de telefoon tot je beschikking hebt is het noodzakelijk alle feiten die in het voorbeeld zitten opgesloten duidelijk voor te lezen zonder naar de fysieke representatie te verwijzen. Door het goed luisteren naar het voorbeeld worden vanzelf alle feiten duidelijk.'

Indien de voorbeelden met behulp van de *TNM expliciet zijn verwoord is de structuur* van de zinnen die daarvoor zijn gebruikt de belangrijkste basis voor het conceptuele schema van de te ontwerpen database. Een conceptueel schema van een database kan immers als volgt worden gedefinieerd:

> Een conceptueel schema is een verzameling regels die specificeren welke toestanden een specifieke database mag aannemen
>
> òf
>
> welke zinnen in de communicatie tussen specifieke personen gebruikt mogen worden.

Er bestaat dus een duidelijke relatie tussen de expliciete zinnen die voor de voorbeelden zijn gebruikt en de te ontwerpen database.

In de meeste methoden om conceptuele schema's te definiëren (zoals NIAM, Nijssens Informatie Analyse Methode) bestaat de mogelijkheid om de structuur van de zinnen zoals verkregen met de TNM volledig in die schema's over te nemen. Daarna kunnen constraints, subtypes, etc. worden toegevoegd om tot volledige conceptuele schema's te komen. Via deze volledige conceptuele schema's kan men dan op eenvoudige wijze komen tot relationele database schema's. Voor deze stap bestaan zelfs een aantal geautomatiseerde hulpmiddelen. De ontworpen relationele databases zullen beter van

kwaliteit zijn en meer op de praktijk aansluiten dan de relationele databases die verkregen zijn via Normalisatie dan wel de Entity Relationship benadering.

3.5 Research Directions in Object-Oriented Databases; S.B. Zdonik

In een aantal nieuwe database toepassingsgebieden, zoals engineering en manufactoring databases (CAD/CAM) en geografische informatiesystemen (GIS) is het noodzakelijk om objecten (CAD-tekeningen, geografische objecten) in de database te kunnen opslaan. Uiteraard is dit in relationele databases mogelijk, maar in deze databases worden objecten min of meer 'platgeslagen', d.w.z. alleen de pure gegevens worden in de tabellen opgeslagen en de structuur van de objecten is slechts te reconstrueren, bijv. via het foreign key mechanisme (zie paragraaf 3.2). Bij toepassingen zoals hierboven genoemd leidt dit tot problemen. Men zoekt dan ook naar een manier om gehele objecten op een meer rechtstreekse manier in een database te kunnen opslaan. In Object Oriented DataBase (OODB) systemen worden gegevens en de bijbehorende elementaire operaties gegroepeerd in objecten (beschreven in objectklassen) en eenmalig vastgelegd. Hierbij zijn de gegevens van zo'n object alleen toegankelijk via de erbij gedefinieerde elementaire operaties. Het toepassen van deze benadering voor de ontwikkeling van GIS is beschreven in [Oost].

Dat OODB-systemen zich in een toenemende populariteit mogen verheugen bleek uit de grote opkomst bij de over dit onderwerp gehouden tutorial. In deze tutorial werd in eerste instantie een overzicht gegeven van de principes die aan OODB ten grondslag liggen. Deze principes komen gedeeltelijk voort uit de Software Engineering & Modularity principes zoals die bekend zijn uit de object georiënteerde programmeertalen:

- Object identiteit. Een object heeft een identiteit die onafhankelijk is van de huidige toestand en waarde.
- Data onafhankelijkheid. Het data model is onafhankelijk van de gebruikte opslagmethode.
- Data abstractie en information hiding. Een data type definieert een representatie en een verzameling operaties. Alleen deze operaties zijn voor dit data type toegestaan en de representatie van het data type is verborgen.
- Hierarchy and inheritance. Het is mogelijk om een hierarchie van objectklassen te definiëren. Een subklasse 'erft' de eigenschappen van zijn superklasse.

Daarnaast zijn er een aantal database principes waaraan OODB moeten voldoen, zoals persistentie (objecten bestaan langer dan de duur van het proces waarin ze zijn gedefinieerd), recovery, consistency en opvraagbaarheid van de opgeslagen objecten. De meest geschikte toepassingsgebieden voor OODB-systemen liggen op het gebied van ontwerp omgevingen (engineering applicaties, programmeeromgevingen). Naast de in de inleiding genoemde wens om objecten rechtsreeks in de database op te slaan zijn de belangrijkste redenen hiervoor o.a. de uitbreidbaarheid van data types (het creëren van nieuwe datatypes op basis van bestaande data types), de wens om de werkelijkheid die beschreven wordt nauwkeuriger te modelleren en andere Software Enigneering principes die goed aansluiten bij de principes waarop OODB-systemen zijn gebaseerd.

Aan het eind van de tutorial bleek echter dat er bij de object georiënteerde benadering van gegevens nog een groot aantal complexe problemen zijn op te lossen. Met verwijzing naar de titel van de tutorial zijn de belangrijkste 'research directions' voor OODB performance, query-talen, transactie-modellen en een aantal implementatie aspecten (architecturen, indexstructuren). Algemeen probleem van de object georiënteerde benadering is het ontbreken van een complete theoretische onderbouwing, zoals deze voor het relationele model wel bestaat. De conclusie is dan ook gewettigd dat, hoewel voor bepaalde toepassingen zeker nuttig, toekomstig gebruik van OODB-systemen op grote schaal vooralsnog onwaarschijnlijk is.

3.6 Integrating AI and Database Technologies; M.L. Brodie and J.Mylopoulos

Integratie van Artificial Intelligence (AI) en database technologieën is een van de steeds meer terrein winnende benaderingen om in de behoefte aan grote en geavanceerde informatiesystemen (Intelligente Informatie Systemen, IIS) te voorzien. Een recent aandachtsgebied (vanuit de AI) van deze integratie zijn Knowledge Based Systems (KBS) met een directe en efficiënte toegang tot een database. Vanuit de database-wereld zijn informatiesystemen met knowledge based processing zo'n aandachtsgebied. Volgens de sprekers van de op de afsluitende dag van de conferentie gehouden tutorial zullen beide gebieden convergeren en een basis vormen voor één technologie voor het ontwikkelen van toekomstige IIS. In de lezing werd een overzicht gegeven van de rol van AI-database technologie voor deze IIS.

Het idee om tot integratie van AI en databases te komen is al enige jaren oud. Het uitgangspunt is dat databases eigenschappen als persistentie, recovery, consistentie, distributie, efficiëntie, robuustheid, etc. kunnen toevoegen aan AI toepassingen (KBS) en dat ze op hun beurt kunnen profiteren van AI technieken t.b.v. een verhoogde database functionaliteit en performance ('intelligente' databases). In de loop van de tijd zijn de AI technologie (met name KBS) en de database technologie dichter bij elkaar komen te staan. Bij beiden vormen gegevens c.q. kennis de basis. Niettemin zijn de benaderingen nog ietwat verschillend. In de database benadering ligt over het algemeen de nadruk op de gegevensstructuren en de manier waarop deze efficiënt gemanipuleerd kunnen worden. De bekende abstractieniveau's zijn conceptueel model, logisch model en fysiek model. Bij AI (KBS) ligt de nadruk meer op de semantiek van gegevens en worden als abstactieniveau's het organisatorisch niveau (hoe mensen tegen informatie aankijken), het kennisniveau (representatie van en redeneren op basis van kennis) en het symbool niveau (representatie in symbolen, efficiëntie, integriteit, etc.) onderscheiden. De aandacht richt zich hierbij met name op het kennisniveau.

Momenteel speelt AI-database integratie onder andere een rol bij gegevens- c.q. kennisrepresentatie en -modellering. Kennisrepresentatietalen en -systemen vanuit de AI-wereld (semantische netwerken, logische programmeertalen zoals Prolog, productiesystemen, hybride representaties, etc.) hebben hun oorsprong gevonden vanuit de databasetheorie. Andersom zijn moderne ontwikkelingen op datamodelgebied, zoals semantische datamodellen en object georiënteerde datamodellen, geïnspireerd door AI (technologie en toepassingen). Ook toekomstige aandachtspunten van gegevensmodellering (geometrie, complexe objecten, causale verbanden, default gegevens, ontkennende gegevens) hangen in meer of mindere mate samen met AI aandachtspunten.

Ook op systeemniveau komt AI-database integratie steeds meer voor, zoals bij geavanceerde en uitbreidbare databasemanagementsystemen (DBMS), deductieve databases (zie par 3.3), object georiënteerde DBMS en knowledge base managementsystemen (KBMS). Een voorbeeld van een uitbreidbaar DBMS waarin AI uitgangspunten en database uitgangspunten zijn geïntegreerd is Postgres (Post Ingres). Voor deductieve databases wordt vaak een logische programmeertaal gebruikt in combinatie met een relationeel DBMS. KBMS is een nog in een beginstadium verkerende

ontwikkeling. Deze systemen zouden een knowledge base moeten beheren zoals een DBMS een database beheert. Momenteel is er echter nog weinig DBMS support voor specifieke eigenschappen van KBS. Speciale aandachtspunten bij de AI-database integratie op systeemniveau zijn recovery, concurrency control, het afdwingen van integriteit van databases / knowledge bases en het optimaliseren van database / knowledge bases queries.

Uit het voorgaande zal duidelijk zijn dat AI(KBS)-database integratie steeds belangrijker wordt bij de ontwikkeling van informatiesystemen. Niettemin zijn er nog de nodige problemen. Deze liggen o.a. op het gebied van architectuur (de heterogeniteit van AI-systemen/KBS en DBMS), performance (sommige AI functies/heuristieken zijn 'unbounded' m.b.t. zoektijd) en semantiek c.q. kennisrepresentatie (opnieuw heterogeniteit). Met name wat betreft de architectuur is het een probleem een optimum te vinden tussen de zgn. 'loosely coupled' KBS en DBMS enerzijds en de zgn. 'tightly coupled' KBS en DBMS anderzijds. Bij 'loosely coupled' systemen vraagt het AI systeem een grote deelverzameling uit de database in een keer op, bewerkt deze en stopt de resultaten in de vorm van een deelverzameling in een keer terug in de database. Probleem hierbij is dat de gegevensoverdracht veel te groot is in vergelijking met de hoeveelheid gegevens die daadwerkelijk wordt verwerkt. Bij 'tightly coupled' systemen (Postgres, Prolog-RDBMS) gebruikt het AI-systeem de database functies rechtstreeks om te redeneren en is de gegevensoverdracht minimaal. Probleem hierbij is echter de beperkte redeneercapaciteit van de database functies. Het vinden van een optimum tussen beide vormen van 'coupling' wordt door de sprekers als een van de belangrijkste uitdagingen beschouwd.

Integratie van AI technologie en database technologie betekent niet alleen het combineren van de voordelen van beide technologieën. De verwachting is dat ze elkaar ook zullen blijven beïnvloeden waardoor ze inderdaad 'convergeren', zoals al in de inleiding genoemd. Met name de diverse benaderingen van KBS shells/tools (systemen gebaseerd op regels, productiesystemen, hybride systemen) zullen leiden tot steeds meer geavanceerde DBMS (deductieve DBMS, object-georiënteerde DBMS en mogelijk ook KBMS), die op hun beurt de basis zullen vormen voor IIS.

4 SAMENVATTING EN TRENDS

Zoals uit de verschillende tutorials al bleek is het huidige onderzoeksveld in de database-wereld vrij breed. Het relationele model is in de jaren '80 behoorlijk ingeburgerd geraakt en heeft voor een groot aantal toepassingsgebieden ingang gevonden. Het huidige en toekomstige onderzoek zal zich voorlopig vooral blijven richten op het ontwikkelen van complete implementaties die alle principes van het relationele model ondersteunen.

Mede onder invloed van nieuwe database toepassingsgebieden (engineering en manufactoring, cartografie, knowledge bases, actieve databases) zijn er echter ook onderzoekers die nieuwe wegen proberen in te slaan. Twee van de belangrijkste nieuwe ontwikkelingen zijn deductieve databases en object-georiënteerde databases. De ontwikkelaars van deductieve databases zoeken naar methoden om het combineren en deduceren op basis van opgeslagen feiten te automatiseren. Deductieve databases bevatten niet alleen feiten maar ook regels op basis waarvan conclusies uit de opgeslagen feiten kunnen worden getrokken. In object-georiënteerde databases bestaat de mogelijkheid om objecten en hun eigenschappen rechtstreeks op te slaan, in plaats van in de vorm van 'platgeslagen' tabellen.

De beide ontwikkelingen zijn nogal uiteenlopend, hetgeen ook geldt voor nog andere nieuwe ontwikkelingen. Ook binnen de diverse ontwikkelingen variëren de experimenten nogal sterk in benadering en methode van technische implementatie, afhankelijk van het onderzochte deelgebied. Bovendien bleek uit de tutorials dat er op ieder van de gebieden nog een behoorlijk aantal complexe problemen zijn op te lossen. De conclusie kan dan ook luiden dat vooralsnog geen enkele nieuwe benadering eenzelfde rol in de jaren '90 kan claimen als de relationele databases in de jaren '80 hebben gedaan. Waarschijnlijker is het dat er in het komende decennium plaats zal zijn voor diverse uiteenlopende benaderingen, zoals semantische datamodellen, object-georiënteerde systemen, knowledge bases, deductieve databases en andere, nog onbekende, benaderingen.

REFERENTIES

[VLDB89] Peter M.G. Apers en Gio Wiederhold (eds.). Proceedings of the fifteenth International Conference on Very Large Data Bases. August 22-25 / 1989, Amsterdam.

[Apt1] Krysztof R. Apt. Efficient Computing of Least Fixpoints in Deductive Databases.

[Apt2] Krysztof R. Apt, Roland N. Bol en Jan Willem Klop. On the safe termination of PROLOG programs.

[Oost] Peter van Oosterom en Jan van den Bos. An Object-Oriented Approach to the Design of Geographical Information Systems.

OVERZICHT PROGRAMMA CONFERENTIE VLDB '89

Tijdschema

	Dinsdag 22 Aug.	Woensdag 23 Aug.	Donderdag 24 Aug.	Vrijdag 25 Aug.
9.00 - 10.30		TUTORIAL 1 SESSION 3A SESSION 3B	TUTORIAL 3 SESSION 7A SESSION 7B	TUTORIAL 5 SESSION 11A SESSION 11B
11.00 - 12.30		TUTORIAL 1 SESSION 4A SESSION 4B	TUTORIAL 3 SESSION 8A SESSION 8B	TUTORIAL 5 SESSION 12A SESSION 12B
14.00 - 15.30	KEY-NOTE SPEECH	TUTORIAL 2 SESSION 5A SESSION 5B	TUTORIAL 4 SESSION 9A SESSION 9B	
16.00 - 17.30	SESSION 2A SESSION 2B	TUTORIAL 2 SESSION 6A SESSION 6B	TUTORIAL 4 SESSION 10A SESSION 10B	

Programma 22 augustus

Key-note speech

'From a laguna to open waters: Another view on the next generations of databases'

H. Gallaire; directeur ECRC, West-Duitsland

Session 2A: Panel session

'Knowledge to Mediate from User's Workstations to Databases'

Session 2B: Data Models and Modelling

'On the design and implementation of information systems from deductive conceptual models'

A. Olive; Un. Politecna de Catalunya, Barcelona

'A deductive method for entity-relationship modeling'

G. Di Battista, M. Lenzerini; Un. degli Studi, Rome

'A family of incomplete relational database models

A. Ola, G. Ozsoyoglu; Case Western Reserve Un., Cleveland OH, USA

Programma 23 augustus

Tutorial 1

'The foreign key Saga'

C.J. Date; Codd and Date International, USA

Session 3A: Extensible Databases and Data Structures

'Gral: an extensible relational database system for geometric applications'

R.H. Güting; Un. Dortmund

'The LSD tree: spatial access to multidimensional point and non-point objects'

A. Henrich, H-W. Six and P. Widmayer; Fern Un., Hagen, West-Duitsland

'Managing Complex objects in an extensional relational DBMS'

G. Gardarin, J-P. Cheiney, G. Kiernan, D. Pastre and H. Stora; INRIA, Le Chesnay, France

Session 3B: Parallelism

'Effective resource utilization for multiprocessor join execution'

M.C. Murphy and D. Rotem; LBL, Berkeley CA, USA

'Optimization and dataflow algorithms for nested tree queries'

M. Muralikrishna; DEC, Colorada Springs CO, USA

'Parallel processing of recursive queries in distributed architectures'

G. Hulin; Philips Res. Lab. Brussel

Session 4A: Graphical Interfaces

'Pasta-3's graphical query language: direct manipulation, cooperative queries, full expressive power'

M. Kuntz and R. Melchert; ECRC München

'ENIAM: a more complete conceptual schema language'

P.N. Creasy; Un. of Queensland, St. Lucia, Australië

'Facekit: a database interface design toolkit'

R. King and M. Novak; Un. of Colorado, Boulder CO, USA

Session 4B: Parallelism

'A low communication sort algorithm for a parallel database machine'

R.A. Lorie and H.C. Young; IBM Research, Almaden CA, USA

'Percentile finding algorithm for multiple sorted runs'

B.R. Iyer, G.R. Ricard and P.J. Varman; IBM DTI, San Jose CA, USA

'A signature access method for the Starburst Database System'

W.W. Chang and H.J. Schlek; IBM Research Almaden CA, USA

Tutorial 2

'Top down versus bottom up computing in deductive databases'

K.R. Apt; Department of Computer Sciences, Texas University, USA

Session 5A: Recursive query optimization

'Commutativety and its role in the processing of linear recursion'

Y.E. Ioannidis; Un. of Wisconsin, Madision WI, USA

'Estimating the size of generalized transitive closures'

R.J. Lipton and J.F. Naughton; Princeton Un. NJ, USA

'Argument Reduction by Factoring'

J.F. Naughton, R. Ramakrishnan, Y. Sagiv and J.D. Ullman; Princeton Un. NJ, Un. of Wisconsin, Madison WI, Hebrew University, Jerusalem, Stanford Un. CA, USA

Session 5B: Panel session

'Database support for hypertext'

Session 6A: Recursive query optimization

'Finding regular simple paths in graph databases'

A.O. Mendelzon and P.T. Wood; CSRG, Un. of Toronto, Canada

'Towards an open architecture for LDL'

D. Chimenti, R. Gamboa and R. Krishnamurthy; MCC, Austin TX USA

Session 6B: Temporal Databases

'Event-join optimization in temporal relational databases'

A. Segev and H. Gunadhi; LBL, Berkeley CA, USA

'Achieving zero information-loss in a classical database environment'

G. Bhargava and S.K. Gadia; Iowa State Un., Ames IA, USA

Programma 24 augustus

Tutorial 3

'An effective design method for relational databases'

G.M. Nijssen; University of Queensland, Australië

Session 7A: Derived data and constraints

'Derived data update in semantic databases'

I.A. Chen and D. McLeod; USC, Los Angeles CA, USA

'Using integrity constraints to provide intensional answers to relational queries'

A. Motro; USC, Los Angeles CA, USA

Session 7B: Allocation and Optimization

'Integration of buffer management and query optimization in relational database environment'

D.W. Cornell and Ph.S. Yu; IBM Res., Yorktown Heights NY, USA

'The effect of bucket size tuning in the dynamic hybrid GRACE hash join method'

M. Kitsuregawa, M. Nakayama and M. Takagi; IIS, Un. of Tokyo

Session 8A: Panel Session

'Building knowledge-based applications with cooperating databases'

Session 8B: Statistics and Statistical Databases

'Random sampling from B+trees'

F. Olken and D. Rotem; LBL, Berkeley CA, USA

'Aggregate evaluability in statistical databases'

F.M. Malvestuto and M. Moscarini; Univ. degli Studi, ENEA, Rome

'Aggregates in Possibilistic Databases'

E.A. Rundensteiner and L. Bic; UC Irvine CA, USA

Tutorial 4

'Research Directions in Object-Oriented Databases'

S.B. Zdonik; Brown University USA

Session 9A: Complex Objects

'Extending the relational algebra to capture complex objects'

B. Mitschang; Un. Kaiserslautern

'Sorting, grouping and duplicate elimination in the advanced information management prototype'

G. Saake, V.Linnemann, P.Pistor and L. Wegner; IBM Scientific Center Heidelberg, West-Duitsland

'Optimization of relational schemas containing inclusion dependencies'

M.A. Casanova, L. Tucherman, A.L. Furtado and A.P. Braga; IBM Brazil, Rio de Janeiro

Session 9B: Recovery and Concurrency Control

'The case for safe RAM'

G. Copeland, T. Keller, R. Krishnamurthy and M. Smith; MCC Austin TX, USA

'ARIES/INT: a recovery method based on write-ahead logging for nested transactions'

K. Rothermel and C. Mohan; IBM Research, Almaden CA, USA

'Quasi Seriazability: a correctness criterion for global concurrency control in InterBase'

W. Du and A. Elmagarmid; Purdue Un. West Lafayette IN, USA

Session 10A: Object Management

'The O_2 object manager: an overview'

F. Velez, G. Bernard and V. Darnis; ALTAIR, Le Chesnay, Frankrijk

'On correctly configuring versioned objects'

R. Agrawal and H.V. Jagadish; AT&T Bell Labs. Murray Hill NJ, USA

'The starburst Long Field Manager'

T.J. Lehman and B.G. Lindsay; IBM Research, Almaden CA, USA

Session 10B: Priority Scheduling

'Scheduling real-time transactions with disk resident data'

R. Abbot and H. Garcia-Molina; Princeton Un. NJ, USA

'Priority in DBMS resource scheduling'

M.J. Carey, R. Jauhari, M. Livny; Un. of Wisconson, Madison WI, USA

Programma 25 augustus

Tutorial 5

'Integrating AI and Database Technologies'

M.L. Brodie; Intelligent Database System Department, GTE Laboratories Inc., USA

J. Mylopoulos; Department of Computer Science, University of Toronto, Canada

Session 11A: Languages for OODB

'The O_2 database programming language'

C. Lécluse and P. Richard; Altair, Le Chesnay, Frankrijk

'A model of queries for object-oriented databases'

W. Kim; MCC, Austin TX, USA

'OQL: a query language for manipulating object-oriented databases'

A.M. Alashqur, S.Y.W. Su and H. Lam; Un. of Florida, Gainesville FL, USA

Session 11B: Panel Session

'Database Tools and Interfaces'

Session 12A: Active databases

'Monitoring database objects'

T. Risch; Hewlett-Packard Labs., Palo Alto CA, USA

'Situation monitoring for active databases'

A. Rosenthal, U.S. Chakravarthy, B. Blaustein and J. Blakeley; Xerox AIT, Cambridge MA, USA

Session 12B: Panel Session

'Future research directions: Evidence from this conference'

REPORT DOCUMENTATION PAGE (MOD-NL)

1. DEFENSE REPORT NUMBER (MOD-NL)	2. RECIPIENT'S ACCESSION NUMBER	3. PERFORMING ORGANIZATION REPORT NUMBER
TD89-3878		FEL-89-B262
4. PROJECT/TASK/WORK UNIT NO.	**5. CONTRACT NUMBER**	**6. REPORT DATE**
20357	-	OCTOBER 4, 1989
7. NUMBER OF PAGES	**8. NUMBER OF REFERENCES**	**9. TYPE OF REPORT AND DATES COVERED**
17	4	FINAL REPORT

10. TITLE AND SUBTITLE
REPORT ON THE 15TH CONFERENCE ON VLDB IN AMSTERDAM FROM AUGUST 22 TILL AUGUST 25 1989
(VERSLAG VAN DE 15E CONFERENTIE OVER VLDB VAN 22 T/M 25 AUGUSTUS 1989 TE AMSTERDAM)

11. AUTHOR(S)
J.J.C.R. RUTTEN, M.C. VAN HEKKEN

12. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
PHYSICS AND ELECTRONICS LABORATORY TNO, P.O. BOX 96864, 2509 JG THE HAGUE
OUDE WAALSDORPERWEG 63, THE HAGUE, THE NETHERLANDS

13. SPONSORING/MONITORING AGENCY NAME(S)
TNO DIVISION OF NATIONAL DEFENSE RESEARCH, THE NETHERLANDS

14. SUPPLEMENTARY NOTES
THE PHYSICS AND ELECTRONICS LABORATORY IS PART OF THE NETHERLANDS ORGANIZATION FOR APPLIED SCIENTIFIC RESEARCH

15. ABSTRACT (MAXIMUM 200 WORDS, 1044 POSITIONS)
THIS DOCUMENT CONTAINS A REPORT ON THE 15TH CONFERENCE ON VERY LARGE DATA BASES (VLDB) THAT WAS HELD IN AMSTERDAM FROM AUGUST 22 TILL AUGUST 25 1989. THE MAIN PART CONSISTS OF ABSTRACTS FROM A NUMBER OF TURORIALS, HELD ON THIS CONFERENCE. THE REPORT CONCLUDES WITH A NUMBER OF TRENDS IN THE AREA OF DATABASES, AS HAVE BECOME EVIDENT FROM THE CONFERENCE.

16. DESCRIPTORS
DATABASES
RESEARCH & DEVELOPMENT
APPLIED SCIENTIFIC RESEARCH

IDENTIFIERS
PROCEEDINGS
OBJECT-ORIENTED
RELATIONAL DATABASES
DEDUCTIVE DATABASES

17a. SECURITY CLASSIFICATION (OF REPORT)	17b. SECURITY CLASSIFICATION (OF PAGE)	17c. SECURITY CLASSIFICATION (OF ABSTRACT)
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

18. DISTRIBUTION/AVAILABILITY STATEMENT	17d. SECURITY CLASSIFICATION (OF TITLES)
UNLIMITED AVAILABLE	UNCLASSIFIED