

Centrale rol voor FCO IM

Geïntegreerd tool support voor datawarehouse-ontwerp

Guido Bakema en Elton Manoku

Het ontwerpproces van een groot corporate datawarehouse begint met een interview met toekomstige datawarehouse-gebruikers en het in kaart brengen van de bronsystemen. Vervolgens wordt het datawarehouse ontworpen, geïmplementeerd, worden ETL-processen ontworpen en gerealiseerd. Dimensionele data marts die op hun beurt weer gevoed moeten worden met data. Tenslotte is er de ingebruikneming om de gewenste Business Intelligence te realiseren.

Dit alles lijkt tamelijk lineair, maar in de praktijk is het dat allerm minst. Het vinden van een evenwicht tussen gebruikerswensen en de bronsystemen wordt gekarakteriseerd door een niet geringe kwantitatieve (het gaat om erg veel data) en kwalitatieve (het integreren van verschillende domeinen) complexiteit. De transformatie naar dimensionele structuren is een ander complex ontwerpproces [1]. Als gevolg van deze uitdagingen is het ontwerp van grote corporate datawarehouses in het algemeen een evolutionair proces met alle daarbij horende uitdagingen ten aanzien van versie-management. Daarbij is het vooral zaak te garanderen dat er een nauwe relatie blijft met de bedrijfsomgeving waarin het datawarehouse moet functioneren.

De in dit artikel besproken volledig *model driven aanpak*, is gebaseerd op het opstellen en onderhouden van een conceptueel model dat kan worden gepresenteerd in verschillende semantisch equivalente verschijningsvormen (elementair, genormaliseerd, gedimensionaliseerd) en in al deze verschijningsvormen eenvoudig kan worden gevalideerd. Deze modellen zijn puur gericht op de gebruikersomgeving, door middel van instantane bruggen naar het logische niveau (Entity Relationship-modellen of desgewenst UML Class-diagrammen), vanwaar vervolgens verder wordt afgedaald naar het fysieke niveau.

Een en ander wordt ondersteund door een geïntegreerde tool set, die het resultaat is van een langlopend R&D project in de Hogeschool van Arnhem en Nijmegen in samenwerking met VanderLek Advies, Atos Origin en Royal Dutch Airlines KLM. De werkwijze en de tool set werden uitvoerig getest in de praktijk en is met succes toegepast, onder meer in het datawarehouse-project KLM Passage. Ook werd de aanpak, inclusief de ondersteunende

tool set, integraal geïncorporeerd in het door Atos Origin gelanceerde Metadata Framework.

FCO-IM

De informatiemodelleringsmethode FCO-IM (Fully Communication Oriented Information Modeling) speelt een centrale rol: FCO-IM modellen zijn steeds het startpunt voor alle logische en fysieke data-modellen die een rol spelen. Om deze reden passeren de belangrijkste concepten en terminologie van FCO-IM (zie [2] en [3]) eerst de revue. FCO-IM is een feittypen-gebaseerde informatiemodelleringsmethode die is voortgekomen uit NIAM [4].

Een FCO-IM informatie-analyse gebeurt in dialoog tussen de analist en domeindeskundigen, waarbij op basis van concrete voorbeelddocumenten de binnen het domeingebied relevante feiten worden verwoord, dat wil zeggen uitgedrukt in zogenaamde feitexpressies. Deze feitexpressies worden vervolgens geanalyseerd: geklassificeerd en gekwalificeerd. Dit analyseproces levert de 'harde semantiek' (de structuuraspecten van het model: feittypen, objecttypen en labeltypen) van het model en vult deze aan met 'zachte semantiek' (naamgeving en feittype- en objecttype-expressies). Meer 'harde semantiek' wordt vervolgens toegevoegd in de vorm van integriteitsaspecten (restrictieve regels). Het resulterende FCO-IM model ligt daarmee dan vastgelegd in een gestandaardiseerde FCO-IM repository en kan diagrammatisch gepresenteerd worden in aan elkaar gerelateerde FCO-IM diagrammen die vanzelfsprekend niets anders zijn dan 'self-synchronizing auto-visualizations' van de populatie van de FCO-IM repository.

"There are 2 exits on floor number 1."

"There are 0 exits on floor number 2."

"Room 2.1 has 20 seats."

"Room 1.1 has 17 seats."

"Room 2.1 is equipped with a PC."

"Room 1.1 is equipped with a PC."

"There is a room 1.2."

"We have PC's available."

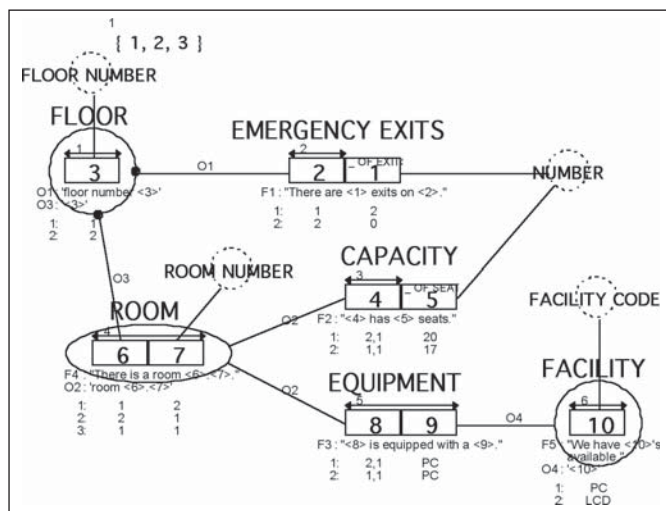
"We have LCD's available."

Afbeelding 1: Enkele FLOORS & ROOMS feitexpressies.

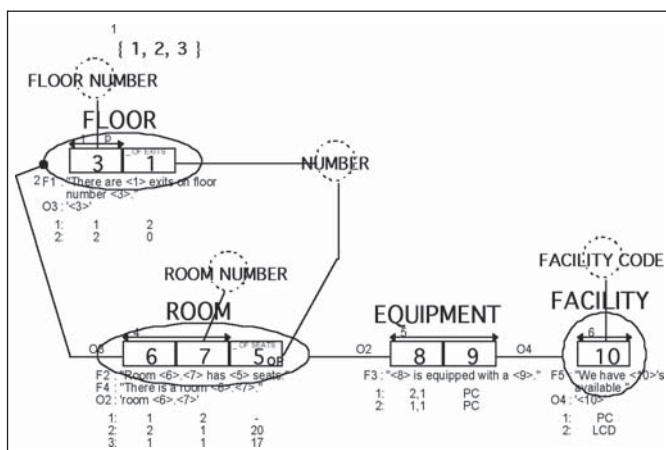
In afbeelding 1 is een aantal elementaire feiten verwoord. Het voorbeeld is ontleend aan het artikel Fully Communication Oriented Information Modeling [5], dat een uitvoeriger overzicht geeft van FCO-IM. Het gebruik van elementaire feiten heeft diverse voordelen. Het belangrijkste voordeel is dat – desgewenst – vrijheid van redundancy kan worden gegarandeerd.

Afbeelding 2 toont een elementair FCO-IM model dat ontstaat na klassificatie en kwalificatie van de in afbeelding 1 getoonde feiteexpressies en aanvulling met standaard constraints: uniciteitsregels, totaliteitsregels en een waardenregel. Merk op dat de drie objecttypen FLOOR, ROOM en FACILITY niets anders zijn dan tot objecttypen getransformeerde feitypen, die in deze geobjectveerde (met een linguïstische term: genominaliseerde) vorm zelf weer rollen kunnen spelen in de andere feitypen EMERGENCY EXITS, CAPACITY en EQUIPMENT. Rol 1 van het feitype EMERGENCY EXITS en rol 5 van het feitype CAPACITY zijn voorzien van postfixen 'OF EXITS' en 'OF SEATS' voor het later kunnen genereren van de gewenste attribuutnamen 'NUMBER OF EXITS' en 'NUMBER OF SEATS'.

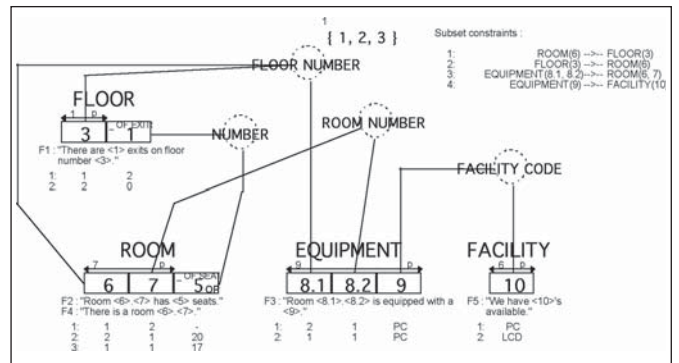
Toepassing van het groeperingsalgoritme resulteert in een gegroepeerd FCO-IM diagram, zie afbeelding 3.



Afbeelding 2: FLOORS & ROOMS elementair model.



Afbeelding 3: FLOORS & ROOMS gegroepeerd FCO-IM diagram.



Afbeelding 4: FLOORS & ROOMS GLR-IGD.

Het groeperings-algoritme voegt feitypen zoveel mogelijk samen zonder redundantie te introduceren. Na uitvoering van het lexicalisatie-algoritme ontstaat een gegroepeerd en gelexicaliseerd FCO-IM model, waarvan in afbeelding 4 het bijbehorende diagram wordt getoond, dat nu feitelijk een relationeel schema is in FCO-IM vermomming. De automatisch gegenereerde subset constraints representeren vreemde-sleutel-verwijzingen en eventuele andere deelverzamelingsregels, zoals mandatory-childness-constraints. Alle drie getoonde FCO-IM modellen zijn gebaseerd op dezelfde set elementaire feiteexpressies, die dan ook uit alle drie kunnen worden teruggegenereerd. Desgewenst kan tenslotte (voor of na de lexicalisatie maakt niet uit) eventueel nog een reductie-algoritme worden uitgevoerd, om waarschijnlijk overbodige domeintabellen te laten vervallen.

Een consequente repository based aanpak maakt de synchronisatie van deze drie semantisch equivalente FCO-IM modellen mogelijk. De getoonde FCO-IM modellen worden vastgelegd in de gestandaardiseerde relationele FCO-IM repository en het FCO-IM modelleringsproces en de transformatiealgoritmes schrijven in, lezen uit en wijzigen de populatie van deze relationele FCO-IM repository op basis van SQL. FCO-IM modellerings-tools, zoals CaseTalk (zie [6]) dat in het hoger onderwijs en in de praktijk veel wordt gebruikt, is feitelijk niets anders dan een heel mooi aangeklede FCO-IM repository.

Het relationele démasqué

Het beschreven transformatieproces (groeperen, lexicaliseren en reduceren) garandeert dat vanuit een elementair FCO-IM model een genormaliseerd (minimaal in BCNF) feitypenmodel met een minimum aantal feitypen wordt verkregen, terwijl de domeinrelevante feiteexpressies nog steeds kunnen worden gereproduceerd ter validatie. We mogen de FCO-IM termen feitype en labeltype vervangen door tabel en domein, want in feite is het FCO-IM diagram van afbeelding 4 een genormaliseerd relationeel schema. Database-beheerders en anderen die minder met FCO-IM vertrouwd zijn, zouden dit relationele schema wellicht dan ook liever in een voor hen meer vertrouwde diagramvorm gepresenteerd willen zien. Daarvoor moet het FCO-IM diagram van afbeelding 4 om zo te zeggen zijn FCO-IM masker laten vallen. CaseTalk heeft daarvoor, als aanvulling op de transformatie-algoritmen, een

Datamodelling

'FCO-IM format naar 'Tabel-format' 'démâsqué-algoritme'.

Omdat alle zachte semantiek (dat wil zeggen de feittypen-expressies) in het diagram van afbeelding 4 nog aanwezig is, kan dit démasqué-algoritme naast de harde semantiek (de structuur- en integriteitsaspecten van het relationele schema) ook deze zachte semantiek (betekenisvolle namen en predikaten) meegenereren, zie afbeelding 5.

Tenslotte kunnen DDL-scripts worden gegenereerd voor verschillende platforms. Zie afbeelding 6, waarin een ANSI SQL92 SQL DDL Script, zoals gegenereerd door het FCO-IM modellerings-tool CaseTalk. Merk op dat de mandatory child constraint hierbij is vertaald in een CHECK clause met een subquery, wat helaas nog niet of nauwelijks wordt ondersteund door hedendaags SQL. Voor Oracle SQL of SQL Server wordt dan ook in plaats van deze ALTER TABLE-instructie een TRIGGER gegenereerd.

Domains:
 FLOOR_NUMBER = NUMERIC(1,0)
 NUMBER = NUMERIC(3,0)
 ROOM_NUMBER = CHAR(3)
 FACILITY_CODE = CHAR(5)

Tables with verbalisation:

FLOOR	
FLOOR_NUMBER (FLOOR_NUMBER) NOT NULL	NUMBER_OF_EXITS (NUMBER)
← PK →	

"There are <NUMBER_OF_EXITS> exits on floor number <FLOOR_NUMBER>."

ROOM		
FLOOR (FLOOR_NUMBER) NOT NULL	ROOM_NUMBER (ROOM_NUMBER) NOT NULL	NUMBER_OF_SEATS (NUMBER) OPTIONAL
← PK →		

"There is a room <FLOOR>.<ROOM_NUMBER>."
 "Room <FLOOR>.<ROOM_NUMBER> has <NUMBER_OF_SEATS> seats."

EQUIPMENT		
FLOOR (FLOOR_NUMBER) NOT NULL	ROOM_NUMBER (ROOM_NUMBER) NOT NULL	FACILITY (FACILITY_CODE) NOT NULL
← PK →		

"Room <FLOOR>.<ROOM_NUMBER> is equipped with a <FACILITY>."

FACILITY	
FACILITY_CODE (FACILITY_CODE) NOT NULL	
← PK →	

"We have <FACILITY_CODE>'s available."

Foreign key references:
 ROOM (FLOOR) → FLOOR (FLOOR_NUMBER)
 EQUIPMENT (FLOOR, ROOM_NUMBER) → ROOM (FLOOR, ROOM_NUMBER)
 EQUIPMENT (FACILITY) → FACILITY (FACILITY_CODE)

Mandatory child constraint:
 FLOOR (FLOOR_NUMBER) → ROOM (FLOOR)

Afbeelding 5: FLOORS & ROOMS relationeel schema.

```

CREATE SCHEMA FLOORS_AND_ROOMS;

CREATE DOMAIN FLOOR_NUMBER AS NUMERIC(1,0);
CREATE DOMAIN NUMBER AS NUMERIC(3,0);
CREATE DOMAIN ROOM_NUMBER AS CHAR(3);
CREATE DOMAIN FACILITY_CODE AS CHAR(5);

CREATE TABLE FLOOR
(FLOOR_NUMBER FLOOR_NUMBER NOT NULL,
NUMBER_OF_EXITS NUMBER NOT NULL,
PRIMARY KEY (FLOOR_NUMBER) );

CREATE TABLE ROOM
(FLOOR FLOOR_NUMBER NOT NULL,
ROOM_NUMBER ROOM_NUMBER NOT NULL,
NUMBER_OF_SEATS NUMBER,
PRIMARY KEY (FLOOR, ROOM_NUMBER) );

CREATE TABLE EQUIPMENT
(FLOOR FLOOR_NUMBER NOT NULL,
ROOM_NUMBER ROOM_NUMBER NOT NULL,
FACILITY_CODE FACILITY_CODE NOT NULL,
PRIMARY KEY (FLOOR, ROOM_NUMBER, FACILITY_CODE) );

CREATE TABLE FACILITY
(FACILITY_CODE FACILITY_CODE NOT NULL,
PRIMARY KEY (FACILITY_CODE) );

ALTER TABLE ROOM
ADD FOREIGN KEY (FLOOR) REFERENCES FLOOR (FLOOR_NUMBER);

ALTER TABLE EQUIPMENT
ADD FOREIGN KEY (FLOOR, ROOM_NUMBER)
REFERENCES ROOM (FLOOR, ROOM_NUMBER);

ALTER TABLE EQUIPMENT
ADD FOREIGN KEY (FACILITY_CODE) REFERENCES FACILITY (FACILITY_CODE);

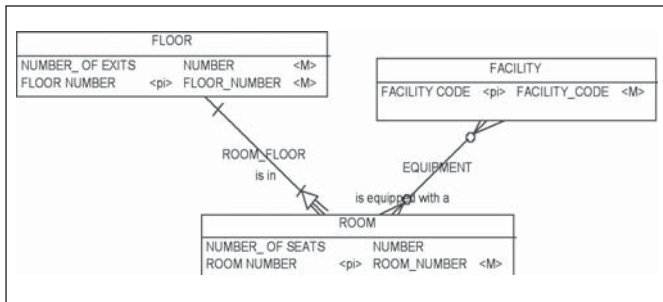
ALTER TABLE FLOOR
CHECK (FLOOR_NUMBER IN (SELECT FLOOR FROM ROOM));
  
```

Afbeelding 6: FLOORS & ROOMS SQL DDL-script.

Andere démasqué-algoritmen

Een entity relationship diagram (ERD) is in feite niets anders dan een op type-niveau redundantievrij relationeel schema. Bij een ERD probeert men zoveel mogelijk attributen (maar niet meer dan dat!) op te nemen in de entiteitstypen. Een consequentie van deze twee vaststellingen is dat het gegroepeerde FCO-IM diagram van afbeelding 3 in feite een ERD is, maar dan nog wel in FCO-IM vermomming. Een FCO-IM naar ERM démasqué-algoritme zorgt voor de conversie van een gegroepeerd (en dan bij voorkeur ook maar even gereduceerd) FCO-IM model naar een equivalent entity relationship format. Dit ERM kan dan (onder meer in XML-format) worden geëxporteerd en geïmporteerd in een ERM-tool (ERwin, PowerDesigner, Oracle Designer etcetera) en daarin dan worden gevisualiseerd, bijvoorbeeld in het Information Engineering format van afbeelding 7.

Een vergelijkbaar démasqué-algoritme converteert een gegroepeerd (en gereduceerd) FCO-IM model naar een equivalent UML class-format, waarbij alle voor de hand liggende operatoren meegegenereerd worden. In beide gevallen (ER diagram en UML class-diagram) kunnen de feittypen-expressies niet meer worden



Afbeelding 7: FLOORS & ROOMS ERD.

getoond naast en een-op-een met de diagrammen, maar bij de import van de metadata in ERM- of UML-tools worden ze wel meegeïmporteerd in de vorm van commentaar en/of beschrijvingen in structuur-elementen. Ook kan een voorbeeldpopulatie niet meer worden getoond, maar in het geval van een class-diagram kan een objectdiagram de populatie nog wel laten zien.

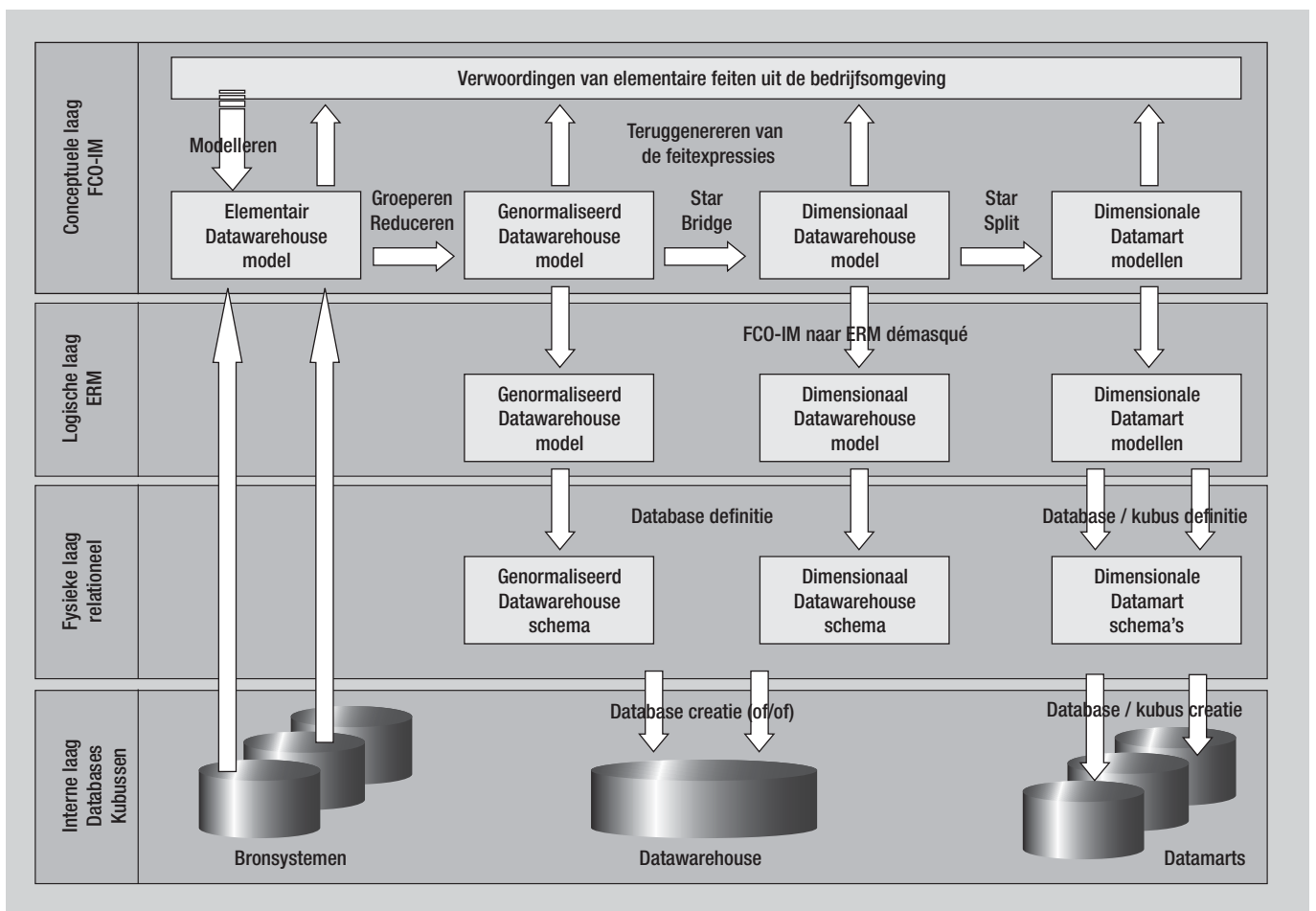
Beide démasqué-algoritmen zijn ondergebracht in een FCO-IM Bridge tool dat naadloos aansluit bij het FCO-IM modellerings-tool CaseTalk. Dit Bridge tool wordt in het hoger onderwijs en in de praktijk erg gewaardeerd, omdat het de brug slaat naar de wereldwijd zeer vertrouwde formats ERM en UML en de

bijbehorende tools. Binnen deze ERM- en/of UML-tools kunnen dan fysieke schema's (gegarandeerd in BCNF) en bijbehorende datadefinitie-scripts worden gegenereerd voor relationele (of zelfs OO) platforms met hun eigen interne schema's. Zo worden de pluspunten van FCO-IM en ERM (of class-diagramming) en ERM-tools ten volle benut. FCO-IM voor de conceptuele modellering- en ERM-tools voor de presentatie van de resultaten en het genereren van de fysieke databases.

Efficiënt metadata-beheer voor dataware-house-ontwerp

Voorgaande repository based-werkwijze maakt efficiënt metadata-beheer mogelijk in klassieke database-ontwerp of reengineering-projecten: recent werd het FCO-IM tool CaseTalk uitgebreid met een reverse-engineerings-algoritme dat een warm welkom kreeg. Met een uitbreiding van het algoritmisch scenario en daarop afgestemd metadata-beheer, konden ook de uitdagingen bij grote datawarehouse-projecten op vergelijkbare wijze tegemoet worden getreden. Het gaat daarbij om versie-op-versie management en de met model-naar-model-transformaties samenhangende complexiteit.

De metadata van het elementaire FCO-IM model van een datawarehouse krijgt weer de primus-inter-pares status en de



Afbeelding 8: De horizontale en verticale lagen-architectuur.

transformatie-algoritmen naar een genormaliseerd FCO-IM model werden uitgebreid met een veel complexer transformatie-algoritme dat zorg draagt voor de conversie van een genormaliseerd model naar een dimensioneel model in de vorm van families van sterren met conforme dimensies. Dit StarBridge-algoritme is gebaseerd op de in twee eerder in DB/M gepubliceerde artikelen van de hand van Harm van der Lek (zie [7] en [8]). De details werden verder uitgewerkt in masters-theses en post-masters assistentschappen [9][10] en vervolgens werd de FCO-IM Bridge tool set uitgebreid met dit algoritme [11]. Een familie van sterren met conforme dimensies kan nu eenvoudig worden opgesplitst (StarSplit-algoritme) in afzonderlijke data marts, elk bestaande uit één of enkele gerelateerde sterren.

Net als de eerdere FCO-IM transformatie-algoritmen, worden het StarBridge- en het StarSplit-algoritme uitgevoerd op het conceptuele niveau, dus binnen FCO-IM. Als gevolg hiervan blijft validatie door eindgebruikers mogelijk op basis van het teruggenereren van feitexpressies. De verticale lagen-architectuur en de FCO-IM naar ERM démasqué-algoritmen zorgen dat steeds een overstap naar het logische niveau van ERM-tools mogelijk is en dan van het logische naar het fysieke niveau van de relationele en dimensionele platforms, zie afbeelding 8.

Deze manier van werken garandeert het behoud van de afstemming van harde en zachte semantiek bij alle horizontale modeltransformaties en bij de verticale démasqué-algoritmen.

Eindgebruikers kunnen de conceptuele en logische modellen op ieder moment valideren, waardoor kan worden gezekerd dat het ontwerp-team goed bezig is en de afstemming met de bedrijfsomgeving niet is kwijtgeraakt. Op ieder moment kunnen via de logische laag van ERM-tools fysieke schema's worden gegenereerd, waarvan dan onmiddellijk zeker is dat ze in overeenstemming zijn met gevalideerde business requirements. Dit is erg waardevol gebleken in de praktijk.

Het startpunt voor de modellering van een datawarehouse is een set van elementaire feiten betreffende het gebruikersdomein. Goede afstemming is nodig tussen wensen en mogelijkheden, welke laatste worden gedictieerd door de bronsystemen. Indien de bronsystemen zelf relationeel zijn, kunnen concept FCO-IM deelmodellen via reverse engineering worden gegenereerd. Enkele cyclische verbeterlagen op en integratie van deze deelmodellen is dan de moeite van het proberen waard. Wel dient in het oog gehouden te worden dat het opstellen van een groot corporate elementair FCO-IM model voor een datawarehouse niet eenvoudig is. De projecten die tot nu toe zo zijn aangepakt tellen tot ongeveer 2500 feittypen. Wanneer het elementaire FCO-IM model echter eenmaal is opgesteld, dan is dat wel het drijvende model voor de hele verdere gang van zaken.

Het via het de groeperings- en reductie-algoritmen te genereren *genormaliseerde FCO-IM model* van het datawarehouse, biedt een meer compact overzicht van feittypen dan het elementair FCO-IM

model. De elementaire feittypen worden gebundeld, zonder dat daarbij redundantie wordt geïntroduceerd. Het resulterend genormaliseerd FCO-IM model is equivalent met een entity relationship model, dat kan worden geëxporteerd en geïmporteerd in een ERM-tool. Binnen het ERM-tool kan daaruit dan een fysiek relationeel schema worden verkregen. Indien het facultatieve reductie-algoritme (het reduceren van waarschijnlijk overbodige domeintabellen) achterwege blijft, dan kan uit het genormaliseerde model het elementaire model desgewenst nog weer worden terugverkregen.

Met behulp van het StarBridge-algoritme kan uit het genormaliseerde FCO-IM model een *dimensioneel FCO-IM model* voor het datawarehouse worden verkregen door de feittypen nog verder te bijeen te groeperen (we spreken van overgroeperen) tot een genormaliseerde view in de vorm van een familie van sterren met conforme dimensies, maar dan nog wel in FCO-IM stijl. Ook nu is het regenereren van de verwoordingen ter validatie nog weer mogelijk. Met behulp van de FCO-IM naar ERM démasqué-algoritmen kan weer een equivalent entity relationship-representatie worden verkregen en desgewenst een dimensioneel relationeel schema van het totale datawarehouse.

Genereren van dimensionele modellen met andere tools

Het geautomatiseerd genereren van dimensionele modellen werd recent ook ingebouwd in sommige ERM-tools: ERwin 4.1 en PowerDesigner 9. Bij ERwin 4.1 stelt dit niet veel voor, omdat slechts potentiële fact tables worden geïdentificeerd, maar de modeltransformatie zelf niet kan worden uitgevoerd.

Powerdesigner 9 gaat veel verder en is in staat om een dimensionele transformatie uit te voeren op fysieke datamodellen, waarbij een sneeuwvlokstructuur wordt gecomprimeerd tot een ster met als uiteindelijke resultaat een set kubussen. Dit 'rechttoe-rechtaan' transformatieproces is echter erg beperkt en biedt geen keuzen, zoals degeneratie van dimensies, snowflaking van een dimensie, het introduceren van mini- of aggregaat-dimensies, het introduceren van betekenisloze sleutels etcetera. Omdat de transformatie in Powerdesigner 9 in de fysieke laag wordt uitgevoerd, is het erg moeilijk het gegenereerde dimensionele model te valideren. Daarbij komt dat het erg kwetsbaar blijft voor versie-opversie aanpassingen door de ontwerpers.

Alle genoemde keuzemogelijkheden zijn wel beschikbaar in StarBridge en op dit moment kan dan ook worden vastgesteld dat de in dit artikel beschreven aanpak rijker en krachtiger is.

In het tweede artikel in de volgende DB/M wordt de praktische implementatie behandeld; het KLM Passage DWH, een corporate datawarehouse dat ontwikkeld is voor de Passenger Division van Royal Dutch Airlines KLM.

Literatuur

1. Ralph Kimball, *The Data warehouse Toolkit*. John Wiley & sons, 1996.
2. Guido Bakema, Jan Pieter Zwart, Harm van der Lek, *Volledig Communicatiegeoriënteerde Informatiemodellering*. TenHagenStam, 1996.

-
3. Marco Engelbart, Rob Arntz, Jorg Janssen, *Volledig Communicatie-georiënteerde Informatiemodellering. Opgaven en casussen. TenHagenStam, 2000.*
 4. G.M. Nijssen, T.A. Halpin, *Conceptual Schema and Relational Database Design: a fact oriented approach. Prentice-Hall, 1989.*
 5. G.P. Bakema, J.P.C. Zwart, H. van der Lek, *Fully Communication Oriented Information Modeling. 1994 NIAM-ISDM Working Conference, Albuquerque New Mexico USA, 1994, www.FCO-IM.com.*
 6. *FCO-IM tool CaseTalk, Bommeljé Crompvoets and partners i.s.m. FCO-IM Consultancy, 2002, www.CaseTalk.com.*
 7. Harm van der Lek, *Overdekkende Bos Stelling. Database Magazine, februari 2000.*
 8. Harm van der Lek, *Op jacht naar de sterren. DB/M, april 2000.*
 9. Jorg Janssens, Egi Rodriguez, *Extensions of FCO-IM, masters thesis Hogeschool van Arnhem en Nijmegen, augustus 1999.*
 10. Rob Arntz, *Algorithmische transformatie van Conceptuele Modellen naar Stermodellen, masters thesis Hogeschool van Arnhem en Nijmegen en Radboud Universiteit Nijmegen, augustus 2000.*
 11. Elton Manoku, *Architecture and implementation of a StarBridge tool, masters thesis Hogeschool van Arnhem en Nijmegen, augustus 2002.*

Guido Bakema en Elton Manoku

Guido Bakema (guido.bakema@han.nl) is lector Data Architectures & Metadata Management aan de Hogeschool van Arnhem en Nijmegen en studie leider van masters-opleidingen in Information Systems Development. Elton Manoku (elton.manoku@han.nl) is als docent/onderzoeker werkzaam bij de Hogeschool van Arnhem en Nijmegen aan de ontwikkeling van geïntegreerde tool sets.