

Produktspecificatie in productiebeheersingsystemen door middel van variant-stuklijsten

Citation for published version (APA):

Muntslag, D. R., & Veen, van, E. A. (1993). Produktspecificatie in productiebeheersingsystemen door middel van variant-stuklijsten. *Informatie*, 35(januari), 44-58.

Document status and date:

Gepubliceerd: 01/01/1993

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

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

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

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Produktspecificatie in productiebeheersingsystemen door middel van variant-stuklijsten

D.R. Muntslag en E.A. van Veen

Dit artikel schetst een beeld van de ontwikkelingen op het gebied van de registratie van produkt- en produktstructuurgegevens in informatiesystemen voor productiebeheersing. Er wordt ingegaan op de problematiek van de registratie van produktstructuurgegevens, in situaties waar sprake is van een grote variëteit aan produkten. Traditionele oplossingen in bestaande softwarepakketten bieden hiervoor geen goede oplossing. Nieuwe concepten, de zogenaamde generatieve stuklijsten, bieden deze oplossing wel. In dit artikel wordt met name ingegaan op een specifiek type hiervan, de variant-stuklijst.

1 Inleiding

De opkomst van de concurrentiefactor 'produktflexibiliteit' – de mogelijkheid produkten te leveren die optimaal voldoen aan de specifieke wensen van een klant – heeft een dominerende invloed op de inrichting van productiesystemen en tal van besturende functies. De productiesystemen zelf worden steeds meer opgebouwd uit 'Flexibele Fabricage Systemen', 'Flexibele Assemblage Systemen', enzovoorts. Deze ontwikkelingen zijn mede gericht op de vervaardiging van een grotere variëteit aan produkten met behoud van de bestaande efficiency.

Ook de concepten die ten behoeve van productiebeheersing worden toegepast, worden sterk beïnvloed door de toenemende produktvariëteit. Is het met een beperkt assortiment eindprodukten met een relatief hoge omzet nog mogelijk op basis van voorspellingen eindprodukten op voorraad te produceren, bij een toenemend aantal eindproduktvarianten wordt dit steeds moeilijker. Men ziet zich steeds meer genoodzaakt het klantorderontkoppelpunt stroomopwaarts te verleggen (Hoekstra & Romme, 1985) en specifieke eindprodukten pas te assembleren als daarvoor een klantorder is ontvangen. Deze verschuiving van een productiebeheersingsomgeving volgens het principe 'produceer-op-voorraad' naar een omgeving met het principe 'assembleer-op-order', 'produceer-op-order' of zelfs 'engineer-op-order' (Sari, 1981) heeft geleid tot de ontwikkeling van nieuwe concepten voor productiebeheersing en bijbehorende ondersteunende informatiesystemen. Eén van de problemen bij de ontwikkeling van nieuwe informatiesystemen voor productiebeheersing is het vastleggen van de zeer grote hoeveelheid produkt(structuur-)gegevens (stuklijsten). De toenemende complexiteit van en variëteit binnen het produktassortiment vereist meer geavanceerde concepten voor de representatie van produkt(structuur-)gegevens dan in het merendeel van de bestaande standaard-software beschikbaar is.

Dit artikel schetst een beeld van de ontwikkelingen op het gebied van de registratie van produkt- en produktstructuurgegevens in informatiesystemen voor productiebeheersing. Met name zal worden ingegaan op een recente ontwikkeling, namelijk op systemen waarmee produkt(structuur-)gegevens kunnen worden gegenereerd op basis van klantspecificaties (Van Veen, 1992; Hegge en Wortmann, 1990). Dit soort systemen wordt *generatieve stuklijstsystemen* genoemd.

Paragraaf 2 is gewijd aan de conventionele stuklijstregistratie en de oplossingen die de bestaande literatuur aanreikt om grote produktvariëteit te representeren. Paragraaf 3 beschrijft vervolgens de functionele architectuur van zogenaamde generatieve stuklijstsystemen. Er zijn verschillende soorten generatieve stuklijstconcepten denkbaar. Hier behandelen wij het variant-stuklijstconcept. Steeds meer standaard-softwareleveranciers rusten nieuwe softwarepakketten uit met een variant-stuklijststelsel. Zonder twijfel zal dit aantal in de nabije toekomst sterk groeien. In de paragrafen 4 en 5 worden de twee hoofdfuncties van een variant-stuklijststelsel achtereenvolgens behandeld. In paragraaf 6 volgt tenslotte een slotbeschouwing.

2 Conventionele stuklijstregistratie

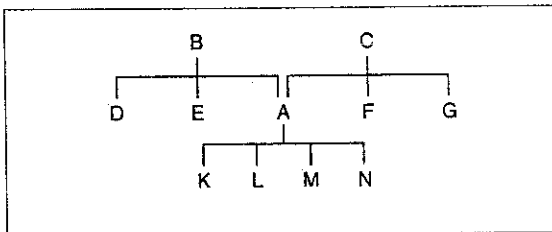
2.1 Inleiding

Gegevens over produkten en processen vormen de kern van een informatiesysteem voor productiebeheersing. Verreweg de meeste bestaande stuklijstsystemen vormen onderdeel van zogenaamde MRP II-georiënteerde standaard-softwarepakketten. MRP II (Manufacturing Resources Planning) is een integraal productiebeheersingsconcept voor de seriematige productie van standaardartikelen (zie voor een nadere beschrijving: Muntslag en Verstegen, 1990). Een belangrijk onderdeel van een MRP II-systeem is de materiaalbehoefteberekening (zie Wortmann, 1984). Hierbij wordt op basis van de behoefte voor een samenstelling de materiaalbehoefte voor de onderdelen berekend. De stuklijst van een produkt speelt een centrale rol bij de materiaalbehoefteberekening: de stuklijst definieert namelijk uit welke onderdelen een produkt is opgebouwd. Er wordt bij het gebruik van de diverse begrippen van uitgegaan dat de lezer enigszins vertrouwd is met begrippen rondom stuklijsten en stuklijststructuren.

Bij het vastleggen van stuklijsten wordt in de regel het

modulaire opslagprincipe toegepast (zie figuur 1) Bij dit principe hoeft de stuklijst van een artikel (*A*) slechts éénmaal te worden vastgelegd, onafhankelijk van het aantal keer dat de betreffende samenstelling zelf als onderdeel wordt toegepast in de stuklijst van 'hoger gelegen' samenstellingen (*B* en *C*).

Ondanks dit modulaire opslagprincipe ontstaan problemen bij het registreren van produkt(structuur-)gegevens bij zeer grote produktvariëteit. Veel produkten, in het bijzonder kapitaalgoederen, worden op de markt gebracht met een groot aantal eigenschappen waarvoor alternatieve keuzen beschikbaar zijn. In het geval van een personenwagen kan meestal uit vele verschillende kleuren worden gekozen, verschillende motoren, verschillende carrosserieën en minder ingrijpende zaken, zoals getint glas, centrale deurvergrendeling, enzovoort. Het uiteindelijke aantal varianten dat wordt aangeboden, kan worden benaderd door de vele combinaties van de mogelijke keuzen. Zo loopt het aantal varianten al snel in de miljoenen. Het behoeft weinig toelichting dat het in die gevallen ongewenst en zelfs praktisch onmogelijk is elke variant expliciet met een artikelnummer te identificeren en voor dit artikelnummer een stuklijst vast te leggen.



Figuur 1: Het modulaire opslagprincipe

In de volgende paragrafen worden twee oplossingen beschreven, die met name in de MRP-georiënteerde literatuur worden voorgesteld om met conventionele stuklijstsystemen een zeer grote produktvariëteit vast te leggen. Achtereenvolgens komen aan de orde de *add-and-delete-techniek* en het *modulair structureren* van stuklijsten.

2.2 De add-and-delete-techniek

Uitbreiding van het produktassortiment wordt zelden bereikt door steeds volledig verschillende nieuwe produkten te ontwikkelen. In het algemeen worden reeksen van produktvarianten ontwikkeld, die onderling slechts op een beperkt aantal punten verschillen. Dit leidt tot zeer veel verschillende stuklijsten, die echter wel sterk op elkaar lijken. Een oplossing die met name enkele jaren geleden nog wel eens werd voorgesteld om het aantal

stuklijstrelaties te beperken, is bekend geworden als de *'add-and-delete'-techniek*. Bij deze techniek wordt alleen van een basisprodukt de volledige stuklijst vastgelegd en worden voor de varianten afzonderlijke stuklijsten vastgelegd, waarin alléén de onderdelen zijn gespecificeerd die afwijken ten opzichte van het basisprodukt (zie figuur 2). Door in de stuklijst van een variant aan het basisprodukt te refereren maakt men het achterhalen van de stuklijst van dit basisprodukt mogelijk. In de overige stuklijstrelaties van de variant staat door middel van negatieve aantallen aangegeven welke onderdelen van het basisprodukt *niet* voorkomen in de betreffende variant. De relaties met de positieve aantallen specificeren de onderdelen die juist *additioneel* in de variant voorkomen ten opzichte van het basisprodukt.



Figuur 2: De add-and-delete-techniek

In geval van variant X_1 wordt de betreffende stuklijst 'geëxplodeerd' naar de onderdelen x , s en t . De stuklijst van x wordt verder geëxplodeerd naar de onderdelen p , q , r en s . In de materiaalbehoefteberekening valt de positieve behoefte voor s weg tegen de even grote negatieve behoefte voor s .

De add-and-delete-techniek wordt om een aantal redenen steeds vaker afgeraden. Ten eerste heeft deze techniek haar intrede gedaan in een tijd dat men het aantal stuklijstrelaties wilde beperken in verband met het geheugenbeslag bij de beschikbare hardware. Dit is echter steeds minder een doorslaggevend argument. Ten tweede kon de tijd nodig voor het invoeren en wijzigen van stuklijsten worden beperkt. Bij praktische toepassingen blijkt echter dat ook dit een twijfelachtig argument is. Het blijkt namelijk zeer moeilijk te zijn add-and-delete-stuklijsten consistent te houden, met name bij frequente stuklijstwijzigingen. Bijvoorbeeld in figuur 2 mag onderdeel s in produkt X niet worden vervangen door een ander onderdeel zonder dat eerder de stuklijstrelatie X_1-s verwijderd of aangepast is. In sommige gevallen is voor het behoud van de consistentie zeer complexe en kostbare ondersteunende software ontwikkeld. Ook blijkt dat bij de ontwikkeling van geavanceerde programmatuur voor manipulatie en opvraging van stuklijsten het add-and-delete-verschijnsel een sterk complicerende factor is.

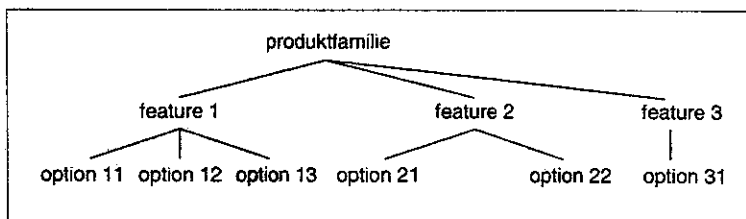
Tot slot moet worden opgemerkt dat met de add-and-delete-techniek weliswaar het aantal stuklijstrelaties kan worden beperkt, maar het biedt geen oplossing voor de noodzaak de mogelijk miljoenen produktvarianten expliciet vast te leggen. Immers, in de add-and-delete-techniek wordt nog steeds verondersteld dat iedere produktvariant uniek door middel van een codenummer is geïdentificeerd.

Voor een uitgebreide beschrijving van de gevaren die kleven aan het gebruik van de add-and-delete-techniek vanuit het gezichtspunt van de materiaalplanning, verwijzen we naar Mather (1982).

2.3 Het modulair structureren van stuklijsten: produktfamilies, features en options

Zoals gezegd ontstaat een grote produktvariëteit in de regel doordat, met een basisprodukt als uitgangspunt, varianten kunnen ontstaan via de keuze uit tal van 'options'. Teneinde het keuzeproces te structureren verdeelt men de volledige produktvariëteit in produktfamilies. Vervolgens worden per produktfamilie elkaar onderling uitsluitende options toegewezen aan een 'feature'. Zo'n feature wordt in de regel zodanig gekozen dat deze een voor de markt herkenbare keuze uit de produkteigenschappen representeert. Bijvoorbeeld de feature 'kleur' kan als options hebben: 'rood', 'wit' en 'blauw'. Bij het definiëren van een produkt kan zo voor iedere feature van dat produkt een keuze worden gemaakt uit de bij die feature behorende options. Veronderstel een basisprodukt heeft 10 features elk met 2 options, dan zouden zo theoretisch 2^{10} eindproduktvarianten gespecificeerd kunnen worden.

Op basis van features en options is het dus mogelijk binnen een produktfamilie eindproduktvarianten uniek te identificeren zonder elke variant a priori met een artikelnummer te definiëren. De volgende stap is de relatie te leggen naar stuklijsten die de materiaalinhoud van de eindprodukten beschrijven. Hiervoor wordt in de MRP-literatuur een techniek voorgesteld die bekend staat als het *modulariseren van stuklijsten* (niet te verwarren met de modulaire opslag van stuklijsten). Bij het modularise-



Figuur 3: Typische opbouw van een Super Bill

ren van stuklijsten van eindprodukten worden die onderdelen waarvan het voorkomen in de stuklijst van een eindprodukt wordt bepaald door dezelfde option, bij elkaar gegroepeerd. Zo'n groep krijgt een apart codenummer (een 'pseudo') en wordt gezien als onderdeel van het betreffende eindprodukt. De onderdelen die tezamen de pseudo vormen, worden door middel van stuklijstrelaties gekoppeld aan de betreffende option. Zo ontstaan voor tal van options verzamelingen onderdelen die nodig zijn voor de realisering van een eindprodukt met die option. Het moge duidelijk zijn dat zoveel mogelijk moet worden vermeden dat onderdelen afhankelijk zijn van een combinatie van options, omdat hier weer het gevaar van de combinatorische explosie van option-combinaties schuilt. Voor een meer gedetailleerde beschrijving van de modulariseringstechniek willen we verwijzen naar Orlicky e.s. (1972).

In de conventionele stuklijstsystemen kan in de regel de structuur van features en options worden gerepresenteerd met behulp van stuklijstrelaties. Daartoe kunnen in de stuklijststructuur – naast items die fysieke materialen en artikelen representeren – verschillende *pseudo-items* worden gedefinieerd die een produktfamilie, feature of option kunnen representeren. De stuklijststructuur die hiervoor wordt gebruikt, wordt in de Engelstalige literatuur vaak *Super Bill* genoemd. Figuur 3 toont de typische opbouw van een Super Bill.

De stuklijstop wordt gevormd door een pseudo-item dat de produktfamilie representeert. Voor dit pseudo-item wordt een stuklijst gedefinieerd bestaande uit feature-items. Voor elk feature-item wordt met stuklijstrelaties naar option-items aangegeven welke options voor die feature kunnen worden gekozen. De option kan een fysiek produkt zijn, maar ook een groepering van option-specifieke onderdelen, die echter fysiek niet kunnen worden samengesteld. In het laatste geval wordt de option gerepresenteerd door een pseudo-item met een stuklijst die deze option-specifieke delen bevat. Bij het specificeren van een eindproduktvariant, bijvoorbeeld voor een klantorder, wordt de Super Bill geëxplodeerd. Vervolgens moet ook voor ieder feature-item de stuklijst worden geëxplodeerd naar de onderliggende option-items, waarna per feature één van de options geselecteerd moet worden. De stuklijsten van deze options leveren eenduidig de benodigde onderdelen op.

In deze structuur is het dus niet mogelijk aan te geven dat één of meer onderdelen afhankelijk is van een combinatie van options (Wortmann, 1987). In de standaard MRP-literatuur wordt ervan uitgegaan dat dit niet voorkomt door een 'slimme' invulling van produktfamilies, features

en options. Dit kan een herdefinitie betekenen van features, waarbij combinaties van options worden samengetrokken tot één option en evenzo worden de bijbehorende features samengetrokken tot één feature. Men dient zich echter te realiseren dat als *alle* features zouden worden samengetrokken tot één feature, de bijbehorende options ieder eenduidig één eindproduktvariant identificeren. Met andere woorden, in dat geval is toch weer elke produktvariant expliciet vastgelegd, zij het via een option.

Het keuzeproces levert na stuklijstexplosie een lijst van onderdelen op die ieder op zich een stuklijst kunnen hebben. Er is echter geen stuklijststructuur opgesteld die aangeeft of de componenten zelf in een bepaalde volgorde tot een eindprodukt moeten worden samengesteld. Er wordt dus impliciet verondersteld dat het assemblageproces van de eindproduktvarianten dermate eenvoudig is, dat nadat de benodigde onderdelen via options zijn vastgesteld, geen additionele informatie (bijvoorbeeld in de vorm van een stuklijststructuur) nodig is om het assemblageproces te beschrijven.

De reden dat het modulaar structureren van stuklijsten in de praktijk vaak niet succesvol is, is dat het succes bepaald wordt door de mate waarin men erin slaagt 'geschikte' produktfamilies en options te definiëren, zodanig dat aan de volgende voorwaarden is voldaan (Van Veen, 1992).

1. Materialen en onderdelen zijn binnen een produktfamilie altijd afhankelijk van maximaal één option en niet van een combinatie van options.
2. Als de benodigde materialen en onderdelen zijn bepaald via de selectie van options zijn geen additionele stuklijststructuren nodig voor de beschrijving van het assemblageproces.
3. De options kunnen onafhankelijk van elkaar worden geselecteerd. De selectie van een bepaalde option heeft geen enkele invloed op de mogelijkheid andere options te selecteren.

In de praktijk wordt zelden aan al deze veronderstellingen tegelijk voldaan. We hebben al aangegeven dat door herdefinitie van features en options kan worden bereikt dat materialen afhankelijk zijn van maximaal één option. Men dient zich hierbij te realiseren dat dit kan resulteren in options die voor vele functies, waaronder de order entry-functie, niet meer herkenbaar zijn. Bovendien zal elke wijziging in het produktontwerp een volledige herdefinitie van features en options teweeg kunnen brengen.

Zo ook kunnen de onderdelen die nodig zijn om een bepaalde option te realiseren fysiek op zeer verschillende plaatsen in het eindprodukt voorkomen. Voor een fabri-

kant van kantoormeubelen kan het eindassemblageproces relatief eenvoudig zijn, zodat er geen additionele gegevens nodig zijn die de assemblage van een specifieke eindproduktvariant beschrijven. Voor producenten van personenwagens, vrachtwagens, vliegtuigen en andere complex samengestelde producten is dit echter wel degelijk nodig.

Tot slot, de implementatie van de mogelijkheid options onderling willekeurig te combineren stelt zeer hoge eisen aan de modulariteit van het produktontwerp. In de meeste gevallen wordt dit ideaal dan ook niet bereikt en zijn produktvarianten met bepaalde combinaties van options technisch niet realiseerbaar. Bij het specificeren van een specifieke eindproduktvariant moet dan terdege rekening worden gehouden met dergelijke verboden combinaties van options. In de meeste standaard-MRP-software is het niet mogelijk om – gegeven een verzameling features en options – formeel de verboden combinaties van features en options vast te leggen.

3 Generatieve stuklijstsystemen

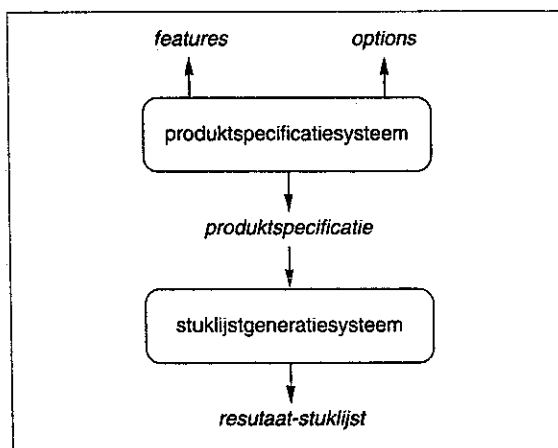
We hebben enkele voorwaarden genoemd die gelden voor de traditionele oplossingen voor produktregistratie in conventionele stuklijstsystemen, in situaties met een grote produktvariëteit. In de praktijk blijken deze zeer moeilijk te verwezenlijken. De steeds verder toenemende produktvariëteit en produktcomplexiteit heeft dan ook geleid tot de noodzaak stuklijstsystemen te ontwikkelen waarin meer complexe relaties tussen features, options en produktstructuren kunnen worden gerepresenteerd. Dit heeft in eerste instantie geresulteerd in stuklijstconcepten, en later ook in de daarop gebaseerde systemen, waarin (Van Veen en Wortmann, 1987; Schönsleben, 1985; Daniëls, 1989).

- verboden combinaties van options kunnen worden vastgelegd ten behoeve van de specificatie van valide produktvarianten;
- stuklijsten kunnen worden gedefinieerd met complexe deterministische afhankelijkheden tussen stuklijstrelaties en options, waarmee het automatisch genereren van specifieke stuklijsten van eindprodukten mogelijk wordt.

Gezocht is naar mogelijkheden om stuklijsten van produkten te genereren, zonder dat de stuklijsten van alle produktvarianten a priori expliciet zijn vastgelegd. Dit type systemen wordt *generatieve* stuklijstsystemen genoemd. In sommige grote bedrijven is dit type systemen specifiek voor de eigen bedrijfssituatie ontwikkeld. Pas zeer recent doen deze nieuwe stuklijstsystemen in allerlei uitvoeringsvormen schoorvoetend hun intrede in de – voornamelijk in Europa ontwikkelde – standaard-soft-

warepakketten voor productiebeheersing. Bij bestudering blijkt dat veel van deze softwarepakketten en bedrijfsspecifieke systemen een overeenkomstige functionele architectuur hebben. Deze valt uiteen in twee hoofd-functies (zie figuur 4), namelijk.

- een produktspecificatiesysteem: een ondersteunende functie voor het specificeren van eindproduktvarianten. Dat wil zeggen het selecteren van options met inachtneming van option-combinaties die om technische of andere redenen niet zijn toegestaan;
- een stuklijstgeneratiesysteem: een functie voor het genereren van een stuklijststructuur van een produktvariant op basis van complexe relaties die kunnen worden gedefinieerd tussen options en stuklijstrelaties.



Figuur 4: Functionele architectuur van een generatief stuklijst-systeem

Het produktspecificatiesysteem maakt het opstellen van een valide produktspecificatie mogelijk (een produktspecificatie is een verzameling van gekozen options). Dit systeem is gebaseerd op gegevens over verboden combinaties van options, de zogenaamde produktspecificatiegegevens. Uitgangspunt is dat elke produktspecificatie die met behulp van het produktspecificatiesysteem is opgesteld, technisch realiseerbaar is en – indien gebruikt in het order entry-proces – leverbaar is. Als een produktvariant uniek is beschreven door middel van een valide produktspecificatie, kan met behulp van het stuklijstgeneratiesysteem de gewenste stuklijststructuur worden gegenereerd.

Generatieve stuklijstsystemen waarbij de stuklijstgeneratie is gebaseerd op complexe relaties tussen *stuklijstrelaties* en *options*, worden in de regel *variant-stuklijstsystemen* genoemd. In dit artikel zullen wij ons beperken tot het variant-stuklijstconcept. Wij zullen achtereenvolgens

ingaan op een aantal praktische ontwerpoverwegingen voor de deelsystemen 'stuklijstgenerator' (paragraaf 4) en 'produktspecificator' (paragraaf 5)

4 Stuklijstgeneratie bij variant-stuklijstsystemen

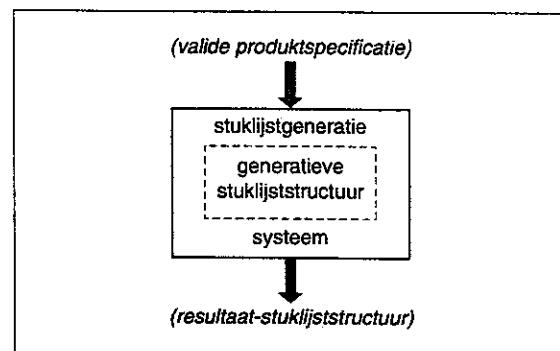
4.1 De 'generatieve stuklijst' en 'resultaat-stuklijst'

Generatieve stuklijstsystemen zijn ontstaan uit twee schijnbaar tegenstrijdige wensen. enerzijds wil men niet alle stuklijsten van produktvarianten expliciet vastleggen, anderzijds wil men wel de stuklijst van elk van deze varianten beschikbaar hebben op het moment dat deze nodig is.

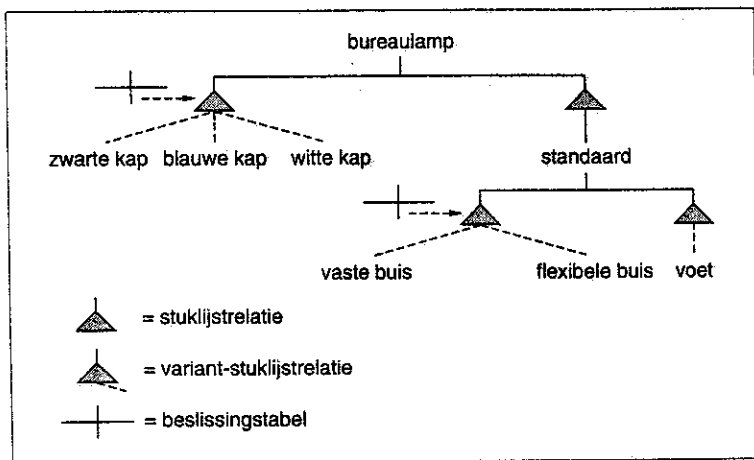
Bij een generatief stuklijstsysteme spelen vier begrippen een rol.

- 1 de 'generatieve stuklijst' Dit is de stuklijst van een produktfamilie waarvan eenduidige stuklijsten van de individuele produkten uit die familie kunnen worden afgeleid;
- 2 de specificatie van de produktvariant waarvoor de stuklijst gegenereerd moet worden. Deze kan zijn verkregen met behulp van een produktspecificatiesysteem waarmee gewaarborgd is dat de specificatie een valide specificatie is;
- 3 de applicaties waarmee de gevraagde stuklijst gegenereerd kan worden, gegeven een generatieve stuklijst en een valide specificatie;
- 4 de gewenste stuklijst, welke wij 'resultaat-stuklijst' zullen noemen.

Figuur 5 laat schematisch zien hoe deze begrippen met elkaar samenhangen in het stuklijstgeneratieproces. De verschillende begrippen zullen in het navolgende nader worden toegelicht.



Figuur 5: Het stuklijstgeneratieproces



Figuur 6: Variant-stuklijstrelaties

4.2 De 'generatieve stuklijst' in het variant-stuklijstconcept

In de conventionele stuklijsttheorie wordt ervan uitgegaan dat elk item (dat wil zeggen knooppunt in de stuklijstgraaf) één produkt(variant) representeert. De stuklijstrelaties van een dergelijk item gelden derhalve alleen voor die bepaalde produktvariant. Kenmerkend voor een generatieve stuklijst is dat de items meer dan één produktvariant kunnen representeren. De stuklijstrelaties zijn nog steeds gedefinieerd voor dat bepaalde item. Dit betekent dat ofwel de geldende stuklijstrelaties geldig zijn voor *elk* van de tot het item behorende produktvarianten (bijvoorbeeld de stuklijstrelatie 'bureaulamp-standaard' in figuur 6), of dat sommige stuklijstrelaties alleen voor bepaalde produktvarianten binnen dat item geldig zijn (bijvoorbeeld de stuklijstrelatie 'bureaulamp-zwarte kap', die alleen geldig is voor zwarte bureaulampvarianten).

De variant-stuklijst is een soort generatieve stuklijst. Dit concept is reeds enige tijd bekend (Schönsleben, 1985; Daniëls, 1988, Van Veen, 1987). Het basisprincipe van de variant-stuklijst is dat een stuklijstrelatie van een parent-item niet meer eenduidig naar één component-item verwijst, maar naar een verzameling van component-items. Daarbij is de conventie dat in iedere resultaat-stuklijst precies één component-item uit de verzameling is geselecteerd. De relatie tussen één parent-item en één component-item wordt in het variant-stuklijstconcept *variant-stuklijstrelatie* genoemd. Een stuklijstrelatie van een parent-item is dan een verzameling elkaar onderling uitsluitende variant-stuklijstrelaties.

Als voor een stuklijstrelatie slechts één variant-stuklijst-

relatie is gedefinieerd, dan geldt deze relatie blijkbaar voor alle produktvarianten die tot het parent-item behoren. Als voor een stuklijstrelatie 2 of meer variant-stuklijstrelaties zijn gedefinieerd, geldt elk van deze variant-stuklijstrelaties slechts voor een beperkt aantal produktvarianten van het parent-item.

Om op basis van de generatieve stuklijst automatische generatie van een resultaat-stuklijst mogelijk te maken, moeten in de generatieve stuklijst deterministische relaties zijn gedefinieerd tussen de variant-stuklijstrelaties en specificaties van produktvarianten. Conceptueel kunnen deze relaties worden gemodelleerd door middel van beslissingstabellen. Neem bijvoorbeeld de verzameling lampen uit figuur 7. Door het leggen van een relatie tussen de variant-stuklijstrelatie 'bureaulamp-zwarte kap' en de feature-option (kleur, zwart), kan worden bereikt dat voor elke produktvariant die wordt geïdentificeerd door een specificatie met de option zwart voor de feature kleur, een resultaat-stuklijst wordt gegenereerd waar de variant-stuklijstrelatie 'bureaulamp-zwarte kap' in voorkomt (zie figuur 7). Juist door dit type relatie wordt de variant-stuklijst een generatieve stuklijst.

4.3 Kanttekeningen bij het variant-stuklijstconcept

Bij het hierboven beschreven concept van de variant-stuklijst kan een aantal kanttekeningen worden geplaatst met betrekking tot.

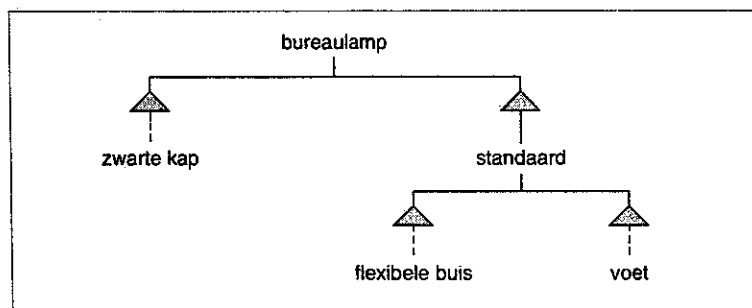
- de definitie van variant-stuklijsten voor produktfamilies,
- identificatie van een produktvariant.

Variant-stuklijsten worden gedefinieerd voor produktfamilies

De concepten en stuklijstsystemen die zijn gebaseerd op het variant-stuklijstprincipe, kennen de beperking dat de options waaraan gerefereerd wordt bij het kiezen van variant-stuklijstrelaties niet aan bepaalde items gekoppeld kunnen worden. In de regel wordt aangenomen dat de options kenmerken van eindprodukten beschrijven. Dit doet echter enigszins afbreuk aan de voordelen van het modulaire opslagprincipe van stuklijststructuren (zie paragraaf 2). Immers, als de options van een produktfamilie worden gewijzigd, kan dit wijzigingen in beslissingstabellen op willekeurige niveaus in de stuklijststructuur tot gevolg hebben. Dit is een grote belemmering bij het onderhoud van een variant-stuklijststructuur als deze uit meer niveaus bestaat.

Identificatie van een produktvariant

Een tweede beperking van variant-stuklijsten betreft de identificatie van produktvarianten. Een van de doelstellingen van het ontwikkelen van generatieve stuklijstsystemen



Figuur 7: Een resultaat-stuklijst

men was dat men niet langer elke produktvariant expliciet van een identificerend nummer hoefde te voorzien. Weliswaar wordt tijdens het stuklijstgeneratieproces voor de relevante items impliciet één produktvariant bepaald met zijn eigen specifieke stuklijst, maar deze produktvariant wordt in de regel geïdentificeerd door een ordernummer of een identificerend nummer voor het generatieproces in combinatie met het itemnummer. De bepaalde produktvariant is niet herkenbaar onder een identificerend artikelnummer, onafhankelijk van een bepaalde order. Dit maakt anonieme voorraadregistratie onmogelijk. Maar in die gevallen waarbij een enorme produktvariëteit wordt geboden, waarvan echter maar een beperkt aantal produktvarianten een echte snelloper is, is het wenselijk juist de snellopende produktvarianten wel te identificeren door een codenummer, zonder dat hiervoor dan noodgedwongen ook een afzonderlijke stuklijst expliciet moet worden gedefinieerd. Hierdoor zou anonieme voorraadregistratie voor dat produkt mogelijk worden, waardoor het logistieke klantorder-ontkoppelpunt voor dat specifieke produkt stroomafwaarts kan worden verlegd.

Onderhoud van variant-stuklijsten

In de praktijk blijkt het onderhoud van variant-stuklijsten een zeer complexe taak te zijn. Als op verschillende niveaus in de variant-stuklijststructuur beslissingstabellen voorkomen, kunnen deze vaak niet onafhankelijk van elkaar worden beschouwd. De uitkomst van een hoger gelegen beslissingstabel kan de relevantie van een lager gelegen beslissingstabel bepalen. Zo ook bestaat er een nauwe samenhang tussen de (gecombineerde) beslissingstabellen in de variant-stuklijst en de validiteit van eindproduktspecificaties. Alle specificaties die ertoe zouden leiden dat geen enkele variant-stuklijstrelatie voor een stuklijstrelatie kan worden bepaald, of juist meer dan precies één variant-stuklijstrelatie geldig is, moeten door de produktspecificatie gegevens worden uitgesloten. Dat wil zeggen dat er geen valide specificaties mogen zijn. De

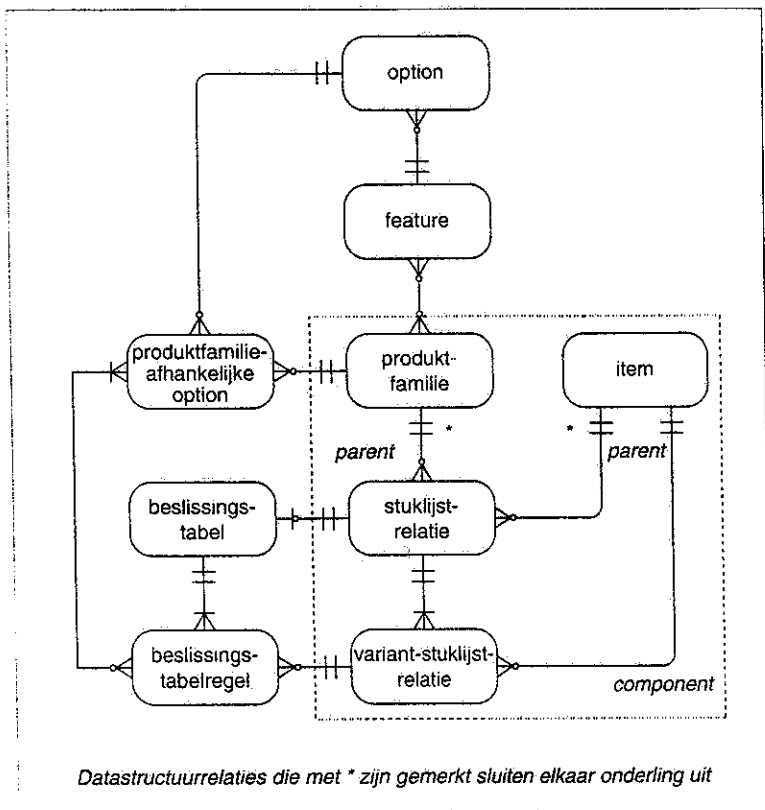
conventie van de variant-stuklijst is immers dat de variant-stuklijstrelaties van één stuklijstrelatie elkaar onderling uitsluiten. Dit betekent dat produktspecificatiegegevens vaak niet kunnen worden gewijzigd zonder dat ook de beslissingstabellen in de variant-stuklijst geanalyseerd en eventueel aangepast moeten worden en vice versa.

In het als opvolger van het variant-stuklijstconcept ontwikkelde generieke stuklijstconcept (Van Veen, 1992; Hegge en Wortmann, 1990) wordt getracht de voorgaande problemen te elimineren. Voor veel bestaande produktie-omgevingen zullen de variant-stuklijsten, gezien de daarin voorkomende complexiteit, echter een prima hulpmiddel blijven als men zich bewust is van de beperkingen.

4.4 Praktische overwegingen bij het structureren van variant-stuklijsten

De structuur van een resultaat-stuklijst die vanuit een generatieve stuklijst wordt gegenereerd, is bepaald door de structuur van de generatieve stuklijst. De generatieve stuklijst dient dus gestructureerd te worden in overeenstemming met het toepassingsgebied waarvoor men de resultaat-stuklijst wil gebruiken. Bij deze toepassingsgebieden kan worden gedacht aan de bekende gezichtspunten vanuit welke men een stuklijst kan structureren, namelijk: constructie, berekening van de materiaalbehoefte, assemblage, enzovoort (Van Rijn, 1985). Bij het ontwikkelen of kiezen van een generatief stuklijststelsel is het met name van belang te weten of de resultaat-stuklijsten een structuur van meer niveaus moeten vormen of dat alleen maar single-level resultaat-stuklijsten gegenereerd hoeven te worden. Voor het weergeven van een assemblage- of constructiestructuur van een eindprodukt is een multi-level stuklijststructuur nodig. Als het echter alleen gaat om het bepalen van de materiaalinhoud van een eindprodukt, kan worden volstaan met een single-level resultaat-stuklijst en generatieve stuklijst.

Voorts speelt de wijze waarop produktfamilies, features en options worden gekozen, een belangrijke rol. Naarmate meer verschillende produktfamilies worden gedefinieerd, ontstaan weliswaar meer generatieve stuklijsten, maar daar staat tegenover dat iedere afzonderlijke generatieve stuklijst minder complex wordt. Binnen iedere produktfamilie blijven immers minder features over die van invloed zijn op de keuze van variant-stuklijstrelaties. De beslissingstabellen in de generatieve stuklijst worden dus eenvoudiger. In het meest extreme geval definieert men een afzonderlijke produktfamilie voor iedere produktvariant. Er zijn in dat geval weliswaar geen beslissingstabellen meer in de generatieve stuklijst, maar men



Figuur 8: Datastructuur van een variant-stuklijstgenerator

is weer beland bij de bezwaren van de conventionele stuklijstsystemen (zie paragraaf 2). Vanuit het oogpunt van stuklijstonderhoud is het aan te bevelen als criterium voor het onderscheiden van produktfamilie die features te kiezen waarvan de options leiden tot grote afwijkingen in de stuklijststructuur

Bij het definiëren van features en options zijn er vaak tegengestelde belangen tussen de eenvoud waarmee een specificatie van een produktvariant kan worden opgesteld en de complexiteit van de beslissingstabellen in de variant-stuklijst. De meest eenvoudige beslissingstabel ontstaat als een één-op-één-relatie wordt gedefinieerd tussen een option en de keuze van een variant-stuklijstrelatie. Dat wil zeggen, een option valt samen met een bepaald onderdeel. Dat kan echter leiden tot zeer veel options met zeer veel onderlinge relaties (zoals uitgesloten combinaties). Ook zal in veel gevallen een onderdeel niet een eenvoudig herkenbare produkteigenschap zijn, hetgeen het produktspecificatieproces bemoeilijkt (met name als de produktspecificatie wordt uitgevoerd door een persoon die niet volledig bekend is met de constructietechnische aspecten van het produkt)

4.5 Typische datastructuur van variant-stuklijsten

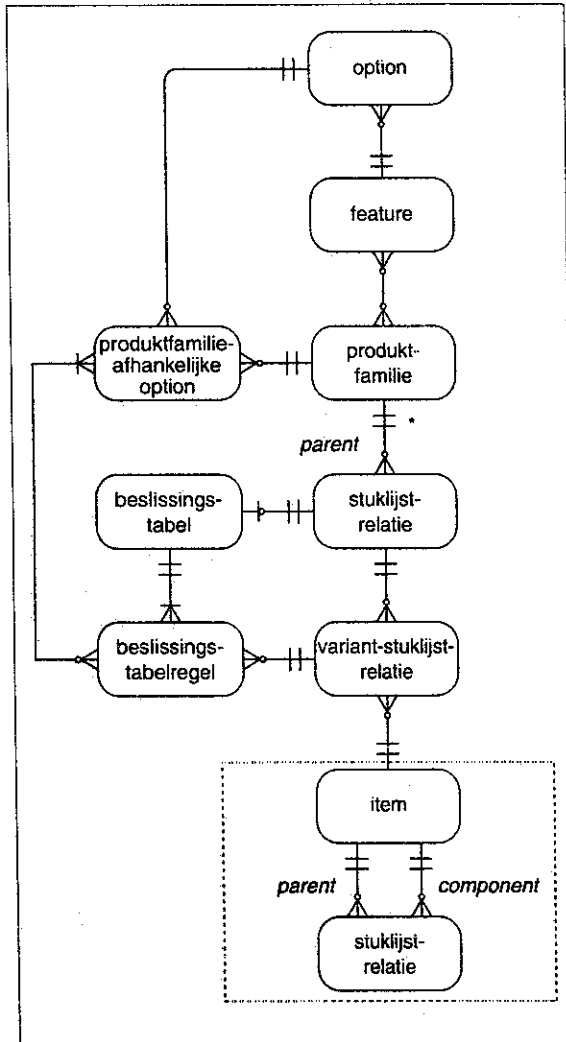
Bij het ontwikkelen van een productie-informatiesysteem speelt de datastructuur die als uitgangspunt wordt genomen een dominerende rol. Zo ook bij een variant-stuklijststelsel. Figuur 8 toont de typische datastructuur van een variant-stuklijstgenerator waarin multi-level generatieve stuklijsten kunnen worden opgebouwd. Het omliggende gedeelte uit deze datastructuur geeft de kern van het variant-stuklijstconcept weer. Indien een item in een stuklijst voorkomt als component-item, dan is deze niet direct gerelateerd aan een stuklijstrelatie, maar via een variant-stuklijstrelatie; deze variant-stuklijstrelatie wordt bestuurd door een beslissingstabel die uniek voor de betreffende stuklijstrelatie is gedefinieerd.

Voor de datastructuur is het verschil van belang tussen het bestaan van multi-level variant-stuklijststructuren of van alleen een single-level variant-stuklijst (zie paragraaf 4.4). Figuur 9 toont de typische datastructuur van een variant-stuklijststelsel waarin alleen single-level variant-stuklijsten kunnen worden opgebouwd. Uit deze figuur komt duidelijk naar voren, dat voor single-level variant-stuklijsten deze datastructuur een relatief geïsoleerde uitbreiding is van de datastructuur van een conventioneel stuklijststelsel (het omliggende gedeelte in figuur 9).

5 Produktspecificatie bij variant-stuklijstsystemen

Zoals in paragraaf 3 is aangegeven, bestaat het eerste deel van de functionele architectuur van een generatief stuklijststelsel uit een produktspecificatiesysteem. In een produktspecificatiesysteem ligt vast welke produktspecificaties wel en welke niet toegelaten zijn. Dit gebeurt in termen van options en relaties tussen options.

Het wezen van een produktspecificatiesysteem is niets anders dan de vastlegging van een verzameling toegelaten combinaties van options, die tezamen de toegelaten varianten binnen een produktfamilie representeren. Er zijn in principe twee vormen waarin options en relaties tussen options onderling in een informatiesysteem kunnen worden gerepresenteerd, namelijk met behulp van grafen of regels. Elke vorm van representatie heeft een aantal voor- en nadelen. Afhankelijk van de specifieke situatie zal bij het ontwerp van een produktspecificator gekozen worden voor één van beide concepten. In de volgende paragraaf zal elk van deze twee representatievormen worden uitgewerkt.



Figuur 9: Datastructuur voor een single level variant-stuklijst

5.1 Produktspecificatie met behulp van een graaf

5.1.1 Het graafconcept

Het produktspecificatieproces bestaat uit een aantal keuzen van geschikte options voor elk produkttype. In de praktijk zijn echter in het algemeen bepaalde combinaties van options om technische of commerciële redenen niet toegestaan. Daartoe dient in het produktspecificatiesysteem te worden aangegeven welke option-keuzen elkaar wel en niet mogen opvolgen. Het produktspecificatieproces is dus in wezen een speciaal soort keuzeprocess, dat kan worden gerepresenteerd met behulp van een gerichte graaf.

Veronderstel een produktfamilie waarbij alle varianten

kunnen worden beschreven met behulp van twee features, ieder met twee alternatieve options.

$$F_1: O_{11}, O_{12}$$

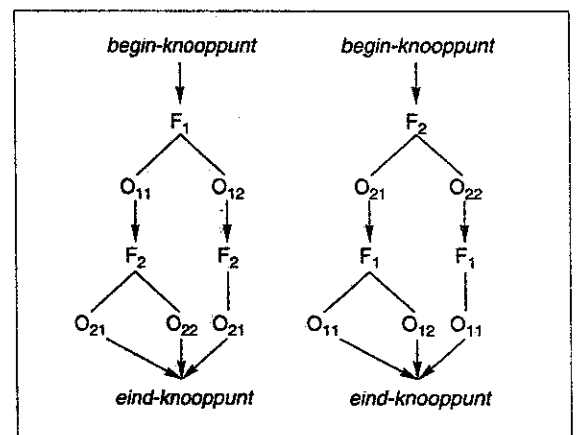
$$F_2: O_{21}, O_{22}$$

Stel nu dat geen enkele combinatie van options is uitgesloten, met andere woorden dat de produktfamilie vanuit dit oogpunt volledig modulaair is. Dan kan de bijbehorende specificatiegraaf op twee manieren worden gestructureerd (zie figuur 10), afhankelijk van de volgorde waarin de features worden weergegeven.

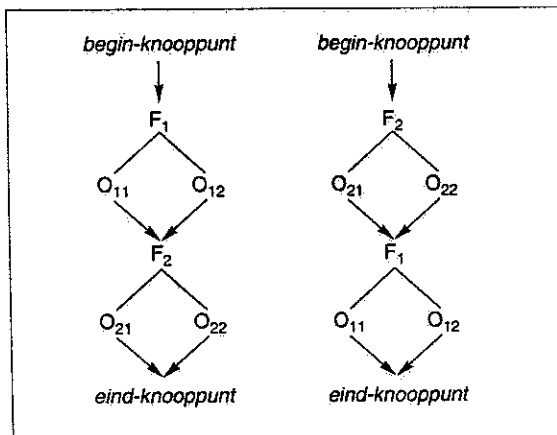
Een gerichte graaf is opgebouwd uit twee elementen, namelijk knooppunten en relaties tussen knooppunten (de pijlen). Bij de toepassing van een graaf voor het modelleren van een produktspecificatieproces, representeert een knooppunt een keuzemogelijkheid (een feature met één of meer options die kunnen worden gekozen). Een pijl representeert een daadwerkelijke keuze (een option).

Bij elk knooppunt wordt een option gekozen, die weer wordt gevolgd door een nieuw knooppunt. Een produktvariant wordt gespecificeerd door een verzameling options (dat wil zeggen een verzameling pijlen) samen te stellen. Elke verzameling options die een toegestane specificatie vormt, kan in de graaf worden teruggevonden doordat de corresponderende pijlen tezamen een pad vormen tussen beginknooppunt en eindknooppunt.

Stel nu dat in het voorgaande voorbeeld de options O_{12} en O_{22} elkaar uitsluiten. Dit betekent dat de corresponderende verzameling pijlen niet een pad in de structuur van de graaf mogen vormen. Ofwel na de keuze voor option O_{12} kan option O_{22} niet meer worden geselecteerd en vice versa. De twee alternatieve graafstructuren zijn



Figuur 10: Alternatieve specificatiegrafen



Figuur 11: Alternatieve graafstructuren bij uitgesloten optie-combinaties

weergegeven in figuur 11. Om de uitgesloten option-combinaties te kunnen representeren zijn twee afzonderlijke keuzemogelijkheden noodzakelijk voor dezelfde feature F_2 (respectievelijk F_1).

5.1.2 De toepassing van grafen in de praktijk

Bij de produktspecificatiefunctie spelen in de praktijk twee aspecten een voorname rol, namelijk.

- 1 de wijze waarop met behulp van het produktspecificatiesysteem een produktvariant gespecificeerd kan worden,
- 2 de wijze waarop produktspecificatiegegevens kunnen worden onderhouden

1 Het specificeren van een produktvariant

De charme van het vastleggen van produktspecificatiegegevens in de vorm van een graaf is dat het specificatieproces vrij eenvoudig geautomatiseerd kan worden als een interactief vraag- en antwoordspel. Het proces start bij het beginknooppunt en vraagt de gebruiker welke optie hij wenst, ofwel via welke pijl het keuzeprocess moet worden voortgezet. Als de keuze is gemaakt, wordt de optie die bij deze pijl hoort, toegevoegd aan de produktspecificatie en het keuzeprocess belandt bij het volgende knooppunt, waar opnieuw een vraag aan de gebruiker wordt gesteld. Dit proces herhaalt zich tot de eindknoop is bereikt. Zo'n rudimentaire ondersteuning van het specificatieproces heeft ook een aantal nadelen, die het op bepaalde aspecten gebruikersvriendelijk maken, namelijk.

- de gebruiker heeft geen overzicht van alle features en options die nog later in het proces voorkomen,
- de gebruiker kan ook niet eenvoudig zien of een keuze die hij nu maakt, gevolgen heeft voor de mogelijkheid om andere options te kiezen in een later stadium. Hij

wordt steeds geconfronteerd met één keuze. Hij heeft geen inzicht in de paden die hij afsluit door het maken van een bepaalde keuze;

- de gebruiker kan bij deze eenvoudige ondersteuning ook niet in een willekeurige volgorde de features bepalen

Het is natuurlijk mogelijk op basis van het graafconcept meer complexe applicaties te ontwikkelen, waarbij bijvoorbeeld wél de options in een willekeurige volgorde kunnen worden bepaald. In dat geval zou bijvoorbeeld na elke gekozen optie moeten worden bepaald of er nog steeds een pad tussen het beginknooppunt en het eindknooppunt kan worden gecreëerd waarin alle options (of pijlen) uit de nieuwe specificatie voorkomen. Daarmee verliest deze oplossing echter het voordeel van de eenvoudige realisatie van een interactieve specificatie-ondersteuning.

2 Het onderhouden van produktspecificatiegegevens

Bij het onderhoud van produktspecificatiegegevens spelen de volgende twee aspecten een belangrijke rol.

- a inzichtelijkheid van de produktspecificatiegegevens,
- b de eenvoud waarmee produktspecificatiegegevens kunnen worden gewijzigd.

a Inzicht in bestaande restricties

Bij toepassing van een graaf komen de niet-toegelaten option-combinaties niet expliciet tot uiting, maar zijn deze impliciet in de graafstructuur verwerkt. Een graaf beschrijft niet alleen relaties tussen de verschillende options, maar de relaties tussen verschillende keuzesituaties in het totale keuzeprocess. Hierdoor moet een relatie tussen features en options nogal eens meer dan één keer in een graaf worden gerepresenteerd (zie figuur 11). Just hierdoor kan het moeilijk zijn om - gegeven een graaf - een helder inzicht te krijgen in de structuur en achtergrond van de graaf. Vreemd genoeg blijkt echter uit onze ervaring in de praktijk dat het opstellen van een graaf zeer behulpzaam kan zijn bij het inventariseren van alle mogelijke relaties tussen options (dit is in de regel beperkt tot grafen die nog eenvoudig gevisualiseerd kunnen worden).

b De eenvoud waarmee produktspecificatiegegevens kunnen worden gewijzigd

Zoals eerder aangegeven, brengt het gebruik van grafen een grote redundantie van gegevens met zich mee. Dit betekent dat in een complexe graaf een (groot) aantal knooppunten de keuze van één en dezelfde feature representeren. In het geval van wijzigen betekent dit dat de graaf moet worden nagelopen om alle knooppunten te vinden die door de wijziging worden beïnvloed. Het toe-

voegen of verwijderen van features en/of options betekent dan ook dat de graaf op vele plaatsen moet worden aangepast en geherstructureerd. In het algemeen kan dan ook gezegd worden dat grafen met complexe restricties veel onderhoudsinspanning vereisen.

5.1.3 De representatie in een informatiesysteem

Om in productiebeheersingssystemen operationeel te kunnen worden toegepast dient het graafconcept te worden vertaald naar informatiestructuren, zodat realisatie in een informatiesysteem mogelijk wordt. Hierbij wordt ervan uitgegaan dat per produktfamilie een graaf in de vorm van beslisknooppunten (features) en keuzeknooppunten (options) wordt vastgelegd, waarmee alle mogelijke eindprodukten binnen die produktfamilie kunnen worden gerepresenteerd. Dat wil zeggen: er wordt nu van uitgegaan dat ook een option door een knooppunt wordt gerepresenteerd. Hierdoor kunnen gegevens van een option eenmalig worden vastgelegd in plaats van dat dit voor elke pijl die deze option representeert opnieuw zou moeten gebeuren. Voordat ingegaan wordt op de benodigde datastructuur voor het graafconcept, wordt voor een beter begrip eerst een aantal ontwerpoverwegingen die aan de datastructuur ten grondslag liggen, nader uitgewerkt.

Een feature is een kenmerk van een eindprodukt en kan als zodanig in verschillende produktfamilies voorkomen. Een option geeft een mogelijke invulling van een bijbehorende feature aan. Als zodanig behoort bij een feature een verzameling van één of meer options. Daar een specifieke feature bij één of meer produktfamilies kan voorkomen en de graaf van een produktfamilie één of meer features kan bevatten, worden produktfamilie-afhankelijke features gedefinieerd. Bij elke produktfamilie-afhankelijke feature dient een set van options te worden benoemd. Dit dient een deelverzameling te zijn van de options die bij de bijbehorende feature zijn benoemd. Deze options worden in analogie *produktfamilie-afhanke-*

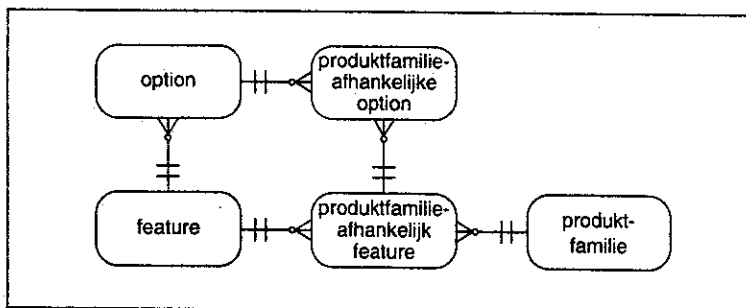
lijke options genoemd. De hierboven genoemde relaties zijn weergegeven in figuur 12.

Wil men nu een produktspecificatiegraaf voor een produktfamilie vastleggen, dan dient men relaties tussen de features en options aan te brengen. Elke graaf begint met een beginknooppunt, die het begin van de graaf representeert. Vanuit het beginknooppunt worden de produktfamilie-afhankelijke features op het eerste niveau in de graaf gevonden. Per feature liggen de bijbehorende options vast. Vanuit een option kunnen nul of meer features in de graaf worden gevonden. Indien geen features worden gevonden, is het eindpunt van de graaf bereikt. Met de hier genoemde relaties zijn de belangrijkste relaties in de datastructuur voor de produktspecificatie met het graafconcept behandeld. De volledige structuur is weergegeven in figuur 13.

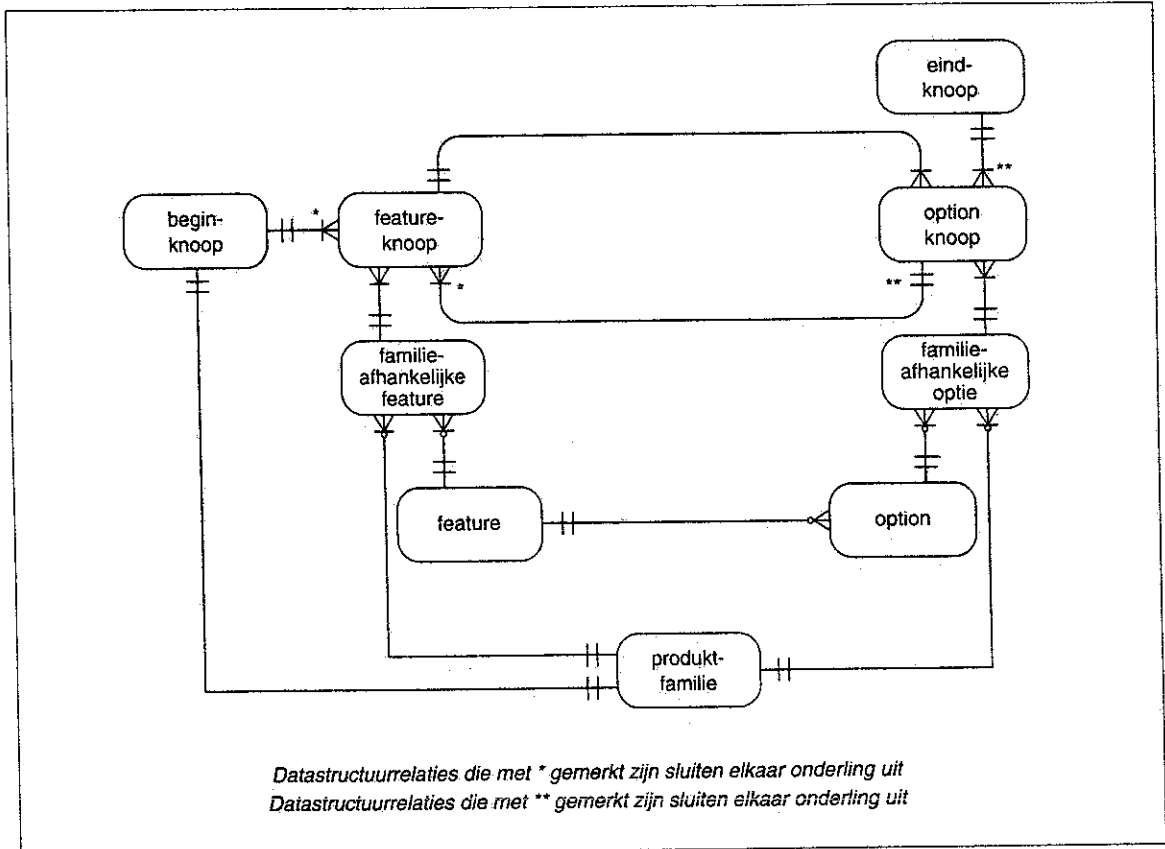
Het opbouwen van produktfamiliegraaf is, gegeven de in figuur 13 weergegeven datastructuur, aan de volgende regels gebonden:

- een (produktfamilie-afhankelijk) feature-knooppunt heeft óf het beginknooppunt óf precies één (produktfamilie-afhankelijk) option-knooppunt als voorganger;
- een (produktfamilie-afhankelijk) feature-knooppunt heeft altijd één of meer (produktfamilie-afhankelijke) option-knooppunten als opvolger. Dit omdat anders geen keuzemogelijkheid voor deze feature zou zijn vastgelegd;
- produktfamilie-afhankelijke features en options mogen maar éénmaal als knooppunt in een pad van de beslissingsboom voorkomen;
- bij een feature-knooppunt mag alleen een option-knooppunt als opvolger worden vastgelegd als er ook een relatie bestaat tussen de betreffende feature en de option.

Indien van een produktfamilie de graaf volgens de structuur uit figuur 13 is vastgelegd, kan een unieke produktspecificatie als volgt worden gegenereerd. Allereerst wordt het beginknooppunt van de graaf geselecteerd. Via een explosie kunnen de bijbehorende produktfamilie-afhankelijke features worden gevonden. Elke feature wordt vervolgens geëxplodeerd (via de relatie tussen feature en option) naar de onderliggende option. Bij de explosie van een feature wordt precies één van de onderliggende options gekozen. Deze option kan op haar beurt weer worden geëxplodeerd. Bij de explosie van een option moeten de onderliggende features *allemaal* één voor één worden geëxplodeerd.



Figuur 12: Vastleggen van een produktfamilie-afhankelijke graaf



Figuur 13: Produktspecificatie met behulp van een graaf

De op deze manier gevonden set van options identificeert het geselecteerde eindproduct binnen de productfamilie uniek. Deze set wordt in het stuklijstgeneratiesysteem (zie paragraaf 4) gebruikt om een specifieke 'result'-stuklijst te genereren.

5.2 Produktspecificatie met behulp van regels

5.2.1 Het regelconcept

In paragraaf 5.1 hebben we gezien hoe een gerichte graaf kan worden gebruikt voor de specificatie van producten. Dit wordt gerealiseerd door toegestane combinaties van options expliciet vast te leggen in de vorm van paden in een graaf. Naast *grafen* kunnen ook *regels* worden gebruikt voor het vastleggen van produktspecificatiegegevens.

Een regel is in feite een booleaanse relatie tussen twee of meer variabelen (options). Hieraan moet zijn voldaan

In het voorbeeld van figuur 11 zou de daar geldende restrictie als volgt in een regel kunnen worden weergegeven:

$$\text{NOT} ((F_1, O_{12}) \text{ AND } (F_2, O_{22}))$$

Dergelijke regels worden *exclusions* genoemd (Koenders, 1988). De definitie van features, de daarbij behorende options en eventuele exclusions die aangeven dat bepaalde combinaties van options niet zijn toegestaan, zijn voldoende om een willekeurige variëteit van producten te representeren.

In de praktijk worden naast exclusions soms ook andere typen regels toegepast. De aanleiding hiervoor is meestal een sterk toenemend aantal exclusions dat nodig is om combinaties van options uit te sluiten. Stel een option O_{ij} van feature F_i kan niet worden gecombineerd met elk van de options van feature F_p , behalve met option O_{pq} . In dat geval is het efficiënter om aan te geven dat als die ene option O_{ij} wordt gekozen, option O_{pq} automatisch onder-

deel moet uitmaken van de produktspecificatie. Dit type regel wordt *inclusion* genoemd en heeft in het voorbeeld de volgende vorm.

$$O_{ij} \rightarrow O_{pq}$$

Een regel kan in een aantal gevallen ook uit twee delen bestaan, het IF-statement (de conditie) en het THEN-statement (de daadwerkelijke constraint), bijvoorbeeld de volgende exclusion:

$$\text{IF } (F_3, O_{32}) \text{ THEN } (\text{NOT } ((F_1, O_{11}) \text{ AND } (F_2, O_{21})))$$

Er kunnen naast de genoemde twee nog vele andere soorten regels worden geconstrueerd om produktvariëteit te beschrijven. Het doel van de toepassing van verschillende typen regels is meestal de algehele complexiteit van de produktspecificatiegegevens te reduceren. Het is echter goed te bedenken dat vermindering van het aantal regels niet hetzelfde is als reductie van de complexiteit. Het introduceren van nieuwe soorten regels (dat is: relaties tussen options) kan tot gevolg hebben dat weliswaar het aantal regels afneemt, maar de volledige complexiteit voor de mens toch toeneemt.

5.2.2 De toepassing van regels in de praktijk

Evenals voor grafen kan ook voor regels een aantal overwegingen worden gegeven met betrekking tot.

- 1 de wijze waarop met behulp van de produktspecificator een produktvariant gespecificeerd kan worden,
- 2 de wijze waarop produktspecificatiegegevens kunnen worden onderhouden

1 Het specificeren van een produktvariant

We hebben laten zien dat een graaf het mogelijk maakt op eenvoudige wijze een interactief specificatieproces te realiseren. De structuur van regels leent zich daar niet zo eenvoudig voor. Bij het toepassen van regels is de meest rudimentaire benadering van het produktspecificatieproces die, waarbij een volledige produktspecificatie moet worden opgesteld en deze pas achteraf wordt gecontroleerd op de aanwezigheid van eventuele verboden combinaties van options. Hiermee kan het produktspecificatieproces echter een 'trial and error'-karakter krijgen, hetgeen ongewenst is. Daarom zal bij produktspecificatiesystemen die zijn gebaseerd op regels, meestal een afzonderlijk user-interface-module nodig zijn die het mogelijk maakt interactief een valide produktspecificatie op te stellen.

Bij een eenvoudig interactief specificatieproces kan voor een willekeurige feature steeds worden getoond welke van de bij de feature behorende options nog toegestaan

zijn. Bij de toepassing van regels vereist dit een complex redeneerproces, omdat niet alleen een option op basis van enkelvoudige exclusions niet toegelaten kan zijn, maar ook bepaalde option-combinaties kunnen zijn uitgesloten op grond van bepaalde combinaties van exclusions (zie de zogenaamde impliciete relaties onder punt 2). Dit moet ten opzichte van de graafrepresentatie als een nadeel worden beschouwd.

De semantiek van de booleaanse regels is wel zeer geschikt om individuele relaties tussen options duidelijk aan de eindgebruiker te tonen. De regels kunnen gebruikt worden om de gebruiker te informeren over de achtergrond van een restrictie van de gekozen (gedeeltelijke) produktspecificatie. Zo kan bijvoorbeeld aan zo'n relatie een reden worden gekoppeld, zoals 'technisch niet mogelijk', 'wettelijk niet toegestaan', enzovoorts.

Produktspecificatiesystemen waarin op basis van regels interactief, in willekeurige volgorde, een produktspecificatie kan worden opgesteld, waarbij de gebruiker ook nog geïnformeerd wordt over uitgesloten options, is niet eenvoudig te realiseren. Voor zover bekend bestaan zij alleen als prototype-software (Euwe en Schuwer, 1992)

2 Het onderhouden van produktspecificatiegegevens

a Inzicht in bestaande restricties

Indien het aantal regels groot wordt en het aantal verschillende soorten regels toeneemt, zal het inzicht in de relaties tussen options afnemen, mede doordat zogenaamde impliciete restricties gaan ontstaan. Dit begrip zal aan de hand van een eenvoudig voorbeeld worden toegelicht.

Veronderstel een produktfamilie waarbij alle varianten kunnen worden gespecificeerd met behulp van drie features, ieder met twee alternatieve options.

$$F_1: O_{11}, O_{12}$$

$$F_2: O_{21}, O_{22}$$

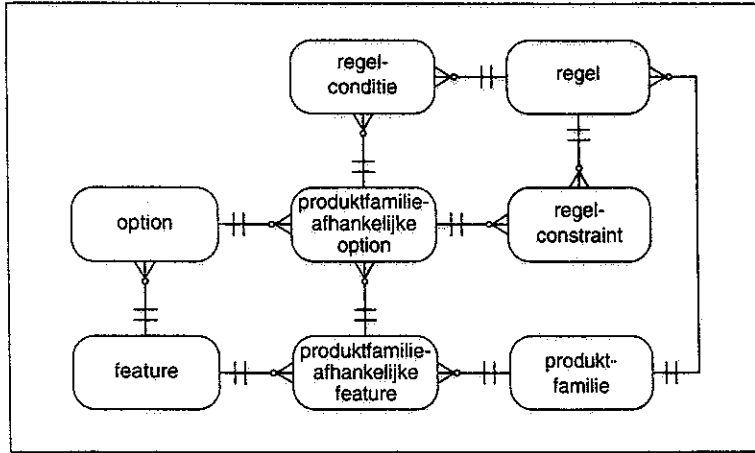
$$F_3: O_{31}, O_{32}$$

Met betrekking tot de relaties tussen options in deze produktfamilie gelden de volgende twee restricties.

$$\text{NOT } (O_{11}, O_{31}) \quad (1)$$

$$\text{NOT } (O_{21}, O_{32}) \quad (2)$$

Veronderstel: bij feature F_1 wordt option O_{11} gekozen. Deze gedeeltelijke produktspecificatie kan nog worden uitgebreid tot een valide produktspecificatie. Vervolgens wordt option O_{21} bij feature F_2 gekozen. Ondanks het



Figuur 14: Produktspecificatie met behulp van regels

feit dat de gedeeltelijke specificatie van deze twee options volgens de expliciete exclusions (1) en (2) op het eerste gezicht nog valide gemaakt kan worden, had de keuze van O_{21} niet toegelaten mogen worden. Het is nu namelijk niet meer mogelijk nog één van de options O_{31} of O_{32} te kiezen. Er is hier sprake van een *impliciete* restrictie die voortkomt uit de gecombineerde werking van de twee expliciete restricties en de conventie dat bij een valide produktspecificatie voor iedere feature een option moet zijn gekozen. Het ontstaan van dergelijke impliciete restricties maakt het doorzien van alle relaties tussen options in de praktijk erg moeilijk. Hierdoor wordt het onderhoud van regels moeilijker. Daarnaast wordt het ook moeilijker te detecteren of een specificatie niet valide is en wordt het navenant moeilijker uit te leggen *waarom* de specificatie niet valide is. Speciaal voor de ondersteuning van het onderhoud van regels kan software worden ontworpen waarmee het mogelijk is alle impliciete restricties op te sporen en expliciet te tonen (Koenders, 1988).

b De eenvoud waarmee produktspecificatiegegevens kunnen worden gewijzigd

In tegenstelling tot bij de toepassing van grafen kunnen veranderingen in features en options bij de toepassing van regels met relatief geringe moeite worden aangebracht. Aangezien de compositie van de specificatie bij het gebruik van regels gescheiden is van de validatie van de specificatie hebben wijzigingen in de specificatiegegevens geen directe invloed op de wijze waarop de specificatie wordt gecomponeerd. Uiteraard moeten bij wijziging wel alle restricties waarin betrokken features of options een rol spelen, op consequenties worden geanalyseerd.

5.2.3 De representatie in een informatiesysteem

Naar analogie met de toepassing van grafen wordt in deze paragraaf ingegaan op een mogelijke representatie van het regelconcept voor een produktspecificatiesysteem in een informatiesysteem. Hierbij zal ervan worden uitgegaan dat de verzameling toegelaten varianten binnen een produktfamilie wordt gedefinieerd in termen van features en options en door uitsluitend vast te leggen welke option-combinaties elkaar uitsluiten (exclusions), dan wel welke option-(combinatie) het voorkomen van een andere option-(combinatie) vereist (inclusions).

De relatie tussen produktfamilie enerzijds en features en options anderzijds is analoog aan die bij de toepassing van grafen. Echter, in plaats van relaties te leggen tussen produktfamilie-afhankelijke features en options onderling, worden nu relaties gelegd tussen produktfamilie-afhankelijke options en produktfamilie-afhankelijke regels (constraints). Door de introductie van een regelconditie kan de relevantie van een constraint afhankelijk worden gemaakt van één of meer options (zie paragraaf 5.2.1).

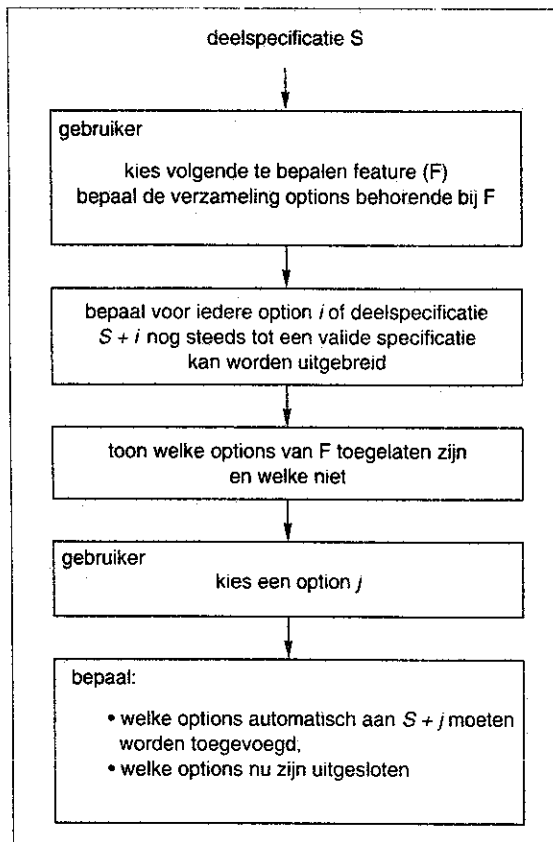
De relatie tussen regels en produktfamilie-afhankelijke options is weergegeven in figuur 14.

Het proces waarmee volledig interactief een valide produktspecificatie kan worden gegenereerd, is schematisch weergegeven in figuur 15.

Hieruit blijkt ook het verschil tussen het specificatieproces bij het gebruik van een graaf en bij het gebruik van regels. Bij het gebruik van regels moet tijdens het produktspecificatieproces een ingewikkelde analyse worden uitgevoerd om te bepalen of uitgaande van de huidige deelspecificatie, uitgebreid met een option, nog steeds minimaal één valide specificatie kan worden verkregen. Met andere woorden, tijdens het specificatieproces worden op basis van de regels 'paden' gereconstrueerd. Bij een graaf liggen deze paden expliciet vast in de graafstructuur. Hierdoor is bij een graaf de validatie van een specificatie tijdens het specificatieproces vanuit de programmatuur bezien eenvoudiger. De prijs die daarvoor wordt betaald, is echter de grotere onderhoudsinspanning die voortkomt uit de noodzaak expliciet alle mogelijke paden vast te leggen.

6 Slotbeschouwing

Dit artikel is ingegaan op de problematiek van registratie van produktstructuregegevens in situaties waar sprake is van grote produktvariëteit. Er is aangegeven dat traditionele oplossingen in bestaande softwarepakketten hiervoor geen goede oplossing bieden. Vervolgens is ingegaan op nieuwe concepten die hiervoor wel een oplossing



Figuur 15: Het produktspecificatieproces

bieden, de zogenaamde generatieve stuklijstsystemen. Er is uitgebreid ingegaan op het variant-stuklijststelsel, een specifiek type generatief stuklijststelsel. Er is hierbij met name aandacht besteed aan ontwerpoverwegingen en de representatie van dergelijke systemen in informatiesystemen.

Variant-stuklijsten verschijnen de laatste jaren in toenemende mate in standaard-softwarepakketten. Ervaring met deze pakketten in de komende jaren zal uitwijzen in welke omstandigheden dit stuklijstconcept een wezenlijke bijdrage kan leveren aan de problematiek van de grote produktvariëteit.

Naast het variant-stuklijstconcept is meer recent het generieke stuklijstconcept geïntroduceerd. Typisch voor

het generieke stuklijstconcept is dat niet alleen relaties kunnen worden gelegd tussen options van eindproducten en stuklijstrelaties, maar ook tussen options van willekeurige (lager gelegde) items en hun component-items (Van Veen, 1992).

Literatuur

- Daniëls, G. (1989). *Een geïntegreerd industrieel productieplanning- en besturingssysteem met een 'generieke' produktstructuur*. Datanorm Software GmbH Freiburg.
- Euwe, M. en R. Schuur (1992). *Verkoopondersteunende systemen (deel I en II)*. Doelmatige Bedrijfsvoering.
- Hegge, H.M.H. en J.C. Wortmann. *Generic bill-of-material: A new product model*. Intern. rapport. Faculteit Bedrijfskunde, Technische Universiteit Eindhoven.
- Hoekstra, S.J. en J.H.J.M. Romme (1985). *Op weg naar integrale logistieke structuren*. Kluwer, Deventer/NIVE, Amsterdam.
- Koenders, H.A.M. (1988). *Registratie van een produktbeschrijving*. Alstudeerverslag Technische Universiteit Eindhoven.
- Mather, H. (1982). *Bills of materials, recipes and formulations*. Wright Publishing Company Inc., Atlanta (Georgia).
- Muntslag, D.R. en M.F.G.M. Versteegen (1990). 'Manufacturing resources planning'. *Financieel Handboek*. Kluwer, Deventer, september 1990.
- Orlicky, J.A., G.W. Plossl en O.W. Wight (1972). 'Structuring the bill-of-material for MRP'. *Production and Inventory Management*, Vol. 13, Nr. 4, winter 1972, blz. 22-27.
- Rijn, Th.M.J. van (1985). *Producteren door informeren*. Kluwer, Deventer.
- Sari, J.F. (1981). *The MPS and the bill of material go hand-in-hand*. Richard C. Ling Inc.
- Schönleben, P. (1985). *Flexibele Produktionsplanung und -Steuerung mit dem Computer*. CW-Publikationen, München.
- Veen, E.A. van (1987). 'Generic bills-of-material', in: *Proceedings of the Conference on Management and New Production Systems*. Enschede, maart 1987, blz. 429-458.
- Veen, E.A. van, en J.C. Wortmann (1987). 'Generic bills-of-material in assemble-to-order manufacturing'. *International Journal of Production Research*, Vol. 25, Nr. 11, blz. 1645-1658.
- Veen, E.A. van (1992). 'Modelling product structures by generic bills-of-material', Elsevier, Amsterdam.
- Wortmann, J.C. (1985). 'Materiaalbehoefteberekening'. *MRPI Informatie*, jaargang 27, speciaal nummer, blz. 333-448.
- Wortmann, J.C. (1987). 'Information systems for assemble-to-order production. An application'. *Engineering Costs and Production Economics*, nr. 12.
- Ir. D.R. Muntslag is werkzaam als senior-consultant bij Moret Ernst & Young management consultants, bij de adviesgroep Logistiek en Informatieverzorging. Daarnaast is hij als wetenschappelijk onderzoeksmidwerker part-time verbonden aan de faculteit Technische Bedrijfskunde aan de Technische Universiteit Eindhoven.*
- Dr. ir. E.A. van Veen heeft aan de Technische Universiteit Eindhoven promotie-onderzoek verricht naar stuklijstsystemen en is nu werkzaam als Informatiemanager Product Development bij Van Doorne's Bedrijfswagenfabriek DAF B.V.*