LUst

De uitwerking van een eenvoudige wens

R.A.C.M. Broekmeulen

Ten behoeve van de simulatieprojecten van de LUW vakgroep Wiskunde is een pakket ontwikkeld waarmee gebruikers via een "point-and-click"-interface een model kunnen bouwen. Door de openheid van het systeem en de integratie van discrete en continue processen is een bruikbaar hulpmiddel ontstaan voor het simuleren van logistieke processen.

Inleiding

Door de vakgroep Wiskunde van de LUW worden regelmatig projecten uitgevoerd in samenwerking met bedrijven. Het merendeel van deze bedrijven is actief in de agrarische sector. De probleemstelling is vaak van logistieke aard (reductie van doorlooptijden, bewaking van de kwaliteit van de veelal bederfelijke produkten, diversificatie van het produktenpakket). Simulatie is een belangrijk hulpmiddel om inzicht te krijgen in de vele factoren die bij complexe produktieprocessen een rol spelen.

Gedurende de afgelopen jaren is geconstateerd dat geavanceerde hulpmiddelen bij het bouwen van simulatiemodellen eerder een noodzaak zijn dan een luxe. Zeker als de projecten uitgevoerd moeten worden door studenten met een beperkte beschikbare tijd en een geringe kennis van simulatie(talen). Ook de logistieke managers in de bedrijven hebben in de praktijk nauwelijks tijd om een simulatietaal te leren en gelegenheid om daarmee een model te ontwikkelen. Vanuit het standpunt van de vakgroep, gelet op ervaringen met projecten, zijn de belangrijkste eisen aan een simulatiepakket:

- ondersteuning van discrete en eventueel continue processen;
- een grafische gebruikersinterface ("point-and-click");
- een open systeem dat eenvoudig uit te breiden is.

De DLO-projectgroep "Evaluatie Simulatie Software" heeft recentelijk gekozen voor de centrale aanschaf van Personal Prosim. Dit pakket fungeert als een tussenoplossing en is duidelijk gericht op onderzoekers. De taal Prosim lijkt sterk op SIMULA (met voorzieningen voor continue simulatie), heeft een beperkt grafisch gebruikersinterface (alleen voor de ontwikkelomgeving en animaties) en is allesbehalve open of uit te breiden door eindgebruikers.

Bij een zelf uitgevoerde inventarisatie bleek dat het ideale pakket inderdaad nog niet op de markt is. Daarom is besloten om zelf een pakket te ontwikkelen voor puur intern gebruik: LU Simulation Tool. LUst is in de eerste plaats bedoeld om discrete, job-shop-achtige systemen te ontwerpen en te evalueren welke volledig automatisch dienen te werken. In samenwerking met de vakgroep Levensmiddelentechnologie is het pakket uitgebreid met voorzieningen voor continue simulatie zodat ook microbiële groei en enzymatische activiteit gemodelleerd kunnen worden.

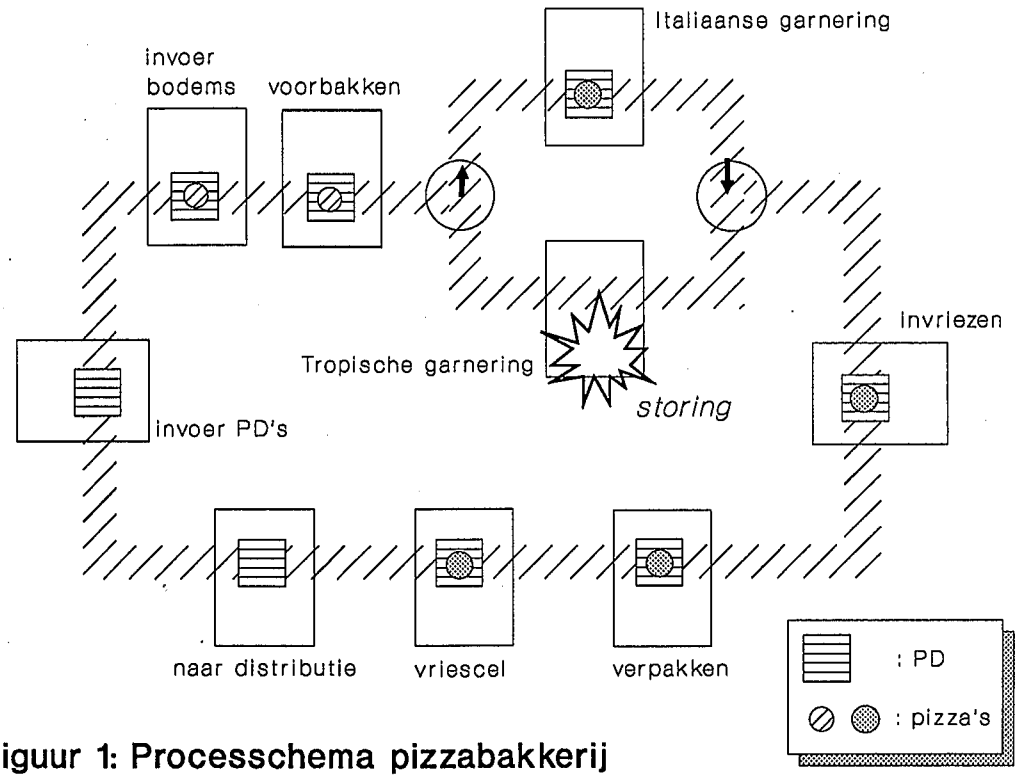
LUst verkeert nog in het ontwikkelstadium maar is desondanks reeds ingezet bij het modelleren van de assemblage van printplaten en televisies, het evalueren van beslissingen bij storingen in een glasconservenfabriek en het beoordelen van de risico's van microbiële besmetting bij de produktie van vleessnacks.

De gedachten achter LUst

De ideeën waarop het simulatiepakket is gebouwd zullen met behulp van een voorbeeld toegelicht worden. In figuur 1 is het processchema getekend voor de produktie van diepvriespizza's. Belangrijke onderdelen van het systeem zijn:

- een lopende band met daarop produktdragers (bijvoorbeeld bakplaten);
- produkten die verschillende bewerkingen ondergaan en daarnaast bloot staan aan besmetting met micro-organismen. De produkten worden op produktdragers (PD's) door het systeem getransporteerd;
- de invoer van PD's:
  deze is alleen bij het opstarten van het systeem actief.
- diverse modulen die:
  - produkten in het systeem brengen en na bewerkingen weer verwijderen;
  - bewerkingen uitvoeren op de produkten zoals voorbakken, garneren en verpakken;
  - produkten tijdelijk opslaan;
  - het proces besturen (bijvoorbeeld de produktmix).

  Het merendeel van deze modulen is gevoelig voor storingen;
- **werkbriefje:**
  De uit te voeren bewerkingen die een produkttype moet ondergaan worden vermeld op een zogenaamd werkbriefje. Dit werkbriefje plus de vereiste volgorde van de bewerkingen wordt bij de invoer van het produkt in het systeem meegegeven aan het produkt op de PD. Modulen die één van die bewerkingen uitvoeren, tekenen dit af op het werkbriefje. Aan de hand van de toestand van het produkt (vorderingen van de werkzaamheden) en informatie over de toestand van de modulen in de lijn (buffer vol, machine heeft storing, etc.) beslist een besturingsmodule m.b.v. ingebouwde regels welke weg het produkt door het systeem dient te volgen.



Figuur 1: Processchema pizzabakkerij

- **continue processen:**
  In een model kunnen continue aspecten opgenomen worden zoals de groei van micro-organismen. In het bovenstaande voorbeeld kan elke module een bron van microbiële besmetting zijn. De bijdrage van de afzonderlijke modulen is stochastisch. De groei van micro-organismen in en op het produkt wordt berekend met een integratieroutine die rekening houdt met de temperatuur, de verblijftijd in de module, de beginconcentratie en de aard van het organisme. Soortgelijke routines worden ontwikkeld voor vergelijkbare kwaliteitsfuncties (chemische modificatie, biodegradatie).

Implementatie van het concept

LUst presenteert zich als een werkblad waarop de gebruiker verschillende modulen kan plaatsen. Er wordt gebruik gemaakt van pull-down menu's en pop-up windows. Deze kunnen m.b.v. een beperkt aantal functietoetsen opgeroepen worden en hierin kan met de cursortoetsen (of de muis) geselecteerd worden. Het programma is grotendeels volgens een object-geörienteerde methode ontwikkeld (Cox & Hunt, 1986). Dit betekent onder meer dat de gegevens over o.a. modulen sterk geïntegreerd zijn met de subroutines die deze gegevens manipuleren (de objecten). Het gevolg is dat de verschillende onderdelen en mogelijkheden van LUst afzonderlijk besproken kunnen worden.

- **modulen:**
  De centrale bouwstenen van een simulatiemodel zijn de modulen. Om de programmatuur zo eenvoudig mogelijk te houden worden alle acties uitgevoerd door een twaalftal basisblokken. Alle denkbare processtappen zijn te modelleren door een juiste combinatie van deze basisblokken. Een aantal veel voorkomende procesapparaten zijn reeds vertaald in een combinatie van deze basisblokken en opgeslagen in een modulebibliotheek. M.b.v. deze voorgeprogrammeerde modulen zijn weer gecompliceerdere modulen te bouwen. Dit op dezelfde, gebruikersvriendelijke manier.
- **produkten en componenten:**
  - *typenummer:*
    Een produkt wordt in een proces gekarakteriseerd door zijn typenummer, de kwaliteit en de status van het werkbriefje. Produkten kunnen in LUst allerlei veranderingen ondergaan, zoals:

- *bewerkingen:*
  in het aan het produkt meegegeven werkbriefje wordt de uitgevoerde bewerking afgetekend.
- *assemblage:*
  dit is een speciaal geval van een bewerking. Naast het aftekenen van de bewerking wordt er een component (of meerdere) aan het produkt toegevoegd.
- *coderen:*
  door het wijzigen van de produktcode verandert een produkt van type. Voorbeelden van dergelijke overgangen zijn het afkeuren van een produkt en het geschikt maken van een produkt als component voor een assemblage.
- *kwaliteitsverlies:*
  bij de rondgang door het proces kunnen er allerlei omzettingen plaatsvinden in en op het produkt. Hierdoor veranderen de kwaliteitseigenschappen van het produkt.

- **produktdragers:**
  Op produktdragers worden de produkten door het proces getransporteerd. De verschillende typen produktdrager zorgen o.a. voor de indeling van de produkten in groepen en voor het batchgewijs verwerken van de produkten. Het is op deze wijze mogelijk met grote aantallen produkten te simuleren.
  - *kwaliteitsfuncties:*
    Dit zijn in essentie routines die op basis van in- en uitgangscondities plus de verblijftijd in een module de kwaliteitsverandering van een produkt berekenen.
  - *storingen:*
    In principe kan in elk type module een storing optreden. De storingstijden en de duur van de storingen kunnen van te voren opgegeven worden. Daarnaast kan er ook gewerkt worden met diverse kansverdelingen.
  - *besturingsregels:*
    Op plaatsen waar produktstromen uit elkaar gaan of zich samenvoegen zijn in LUst besturingsmodulen noodzakelijk die deze stromen regelen. De beslissing om een produkt een bepaalde richting in te sturen hangt af van de produktinformatie (bijvoorbeeld welke bewerkingen moeten nog uitgevoerd worden) en van de toestand van de modulen in het systeem (een overvolle buffer, een machinestoring). Iedere besturingsmodule kan voorzien worden van een eigen set regels. Er is een centraal "prikbord" gedefinieerd waarop iedere module zijn toestand kan bijhouden. Alleen de besturingsmodulen kunnen het "prikbord" lezen en interpreteren aan de hand van hun beslissingsregels ("message passing").
  - *rapportage en animatie:*
    Na afloop van een simulatie-experiment kan m.b.v. animatie snel inzicht gekregen worden in de werking van het model. Bottlenecks, problemen met de afstemming in een lijn, etc. worden eenvoudig zichtbaar gemaakt. Vrijwel alle relevante gegevens over het gedrag van het simulatiemodel worden in bestanden met een standaard format aangeboden aan de gebruiker. Het is mogelijk om de resultaten te evalueren m.b.v. statistische pakketten zoals SPSS. In LUst zijn een beperkt aantal faciliteiten beschikbaar om rapporten te produceren.

De stand van zaken en mogelijke verdere ontwikkelingen

Momenteel is de gebruikersinterface van LUst geprogrammeerd in FoxBase (een snelle kloon van dBase III). Nadat het complete model ingevoerd is, worden alle modulen vertaald naar basisblokken (compilatie) en alle gegevens worden weggeschreven naar een sequentiële file. De routine die m.b.v. deze dataset het simulatie experiment uitvoert, is geschreven in SIMULA. Als er behoefte bestaat aan het uitbreiden van de set van basisblokken, dan is het voldoende dat men een nieuw object (class) toevoegt aan het SIMULA-programma. Dit vergt uiteraard kennis van de taal SIMULA. Omdat LUst een open systeem is kan vervolgens eenvoudig het nieuwe basisblok geïntegreerd worden.

Het omzetten van het programma in Smalltalk-80 of naar een vergelijkbare object-geörienteerde ontwikkelomgeving staat nog ter discussie. Tegenover de grote voordelen op het gebied van het hergebruiken van code en het uitbreiden van het systeem (Wegner, 1989) staan de volgende nadelen (Guthery, 1989 en Ayers, 1989):

- lange rekentijden bij simulatie experimenten;
- niet alle opties zijn te vertalen in (aanpassingen van) objecten;
- slechte overdraagbaarheid aan eindgebruikers door de sterke samenhang met de ontwikkelomgeving. Dit maakt het ook bijna onmogelijk om met meerdere personen tegelijkertijd aan de ontwikkeling van LUst te werken;

Verantwoording

De ontwikkeling van LUst is een gezamenlijke inspanning van medewerkers en studenten van de vakgroep. Vooral de adviezen van dr. ir. M.P. Reinders zijn van grote waarde geweest bij de opzet van LUst en zijn voorlopers (Trip & van Wees, 1987 en Blankers, 1987).

Literatuur

Ayers, K.E., *An Object-oriented Logic Simulator*, Dr. Dobb's Journal 158:72, 1989

Blankers, I., *Een simulatieprogramma voor flexibele produktieautomatisering*, doctoraalverslag, 1988

Cox, B. & Hunt, B., *Objects, Icons and Software-ICs*, Byte 11-8:161, 1986

Guthery, S., *Are the Emperor's new Clothes Object Oriented?*, Dr. Dobb's Journal 158:80, 1989

Trip, H.J. & van Wees, J.A.M., *Simulatie-model van een automatic assembly line*, doctoraalverslag, 1987

Wegner, P., *Learning the Language*, Byte 14-3:245, 1989 ❑

Ir. R.A.C.M. Broekmeulen is universitair docent bij de Landbouwuniversiteit Wageningen, vakgroep Wiskunde, sectie Operationele Analyse, tel. 08370-82627, SURFnet:ROBBOCUUL@RCL.WAU.NL.