

# Agenterne styrer

**Agentbaseret styring vinder hurtigt frem inden for programmeringsverdenen. Disse systemer består af adskillige helt eller delvist autonome softwareprogrammer - de såkaldte agenter - der kommunikerer og samarbejder for at løse specifikke opgaver. Nyt projekt forsøger at få teknologien helt ud på produktionsgulvet, hvor agenterne skal agere på vegne af de enkelte maskinenheder**

## Af lektor Klaus Holth og koordinator Michael Wehner Rasmussen

Kigger man ind i en dansk produktionsvirksomhed, vil man ofte se et komplekst samspil mellem processer, produkter og medarbejdere. Målet er at producere så effektivt som muligt og samtidig optimere processerne. Det er en vanskelig opgave, især hvis tingene ikke går som planlagt.

### Meget kan gå galt

Og der er mange ting, som kan gå galt. En råvare kan være forsinket, der kan opstå fejl i produkter, en maskine kan bryde ned, der kan komme en hasteordre eller en medarbejder kan blive syg. Så det er nødvendigt konstant at ændre planer og træffe beslutninger.

-For at begrænse produktionsstop og spildtid lægger virksomheder derfor stadig flere beslutninger ud til de medarbejdere, som styrer maskinerne. Men hvad nu hvis en stor del af disse beslutninger blev placeret helt ude i de enkelte maskinenheder? spørger erhvervsforsker Claus Risager, der er projektleder for DECIDE.

DECIDE er et IT-Korridor-projekt iværksat under den jysk-fynske IT-satsning, hvor forskere og førende danske virksomheder er gået sammen om at udvikle nye metoder til at styre virksomheders interne transport og produktion.

Partnerne har arbejdet sammen om at udvikle agentbaseret styring, der hurtigt og kompetent kan tage stilling til ændringer i transport- og produktionssystemer. Agenter er således softwareprogrammer, der kan forholde sig til pludselige forandringer, forhandle indbyrdes og træffe nye beslutninger.

## Multi-agent systemer vinder frem

Gennem IT-systemer og avanceret datalogi er det nemlig muligt at decentralisere beslutningsgange, så transport- og produktionssystemer kan blive mere robuste over for uforudsete og pludseligt opståede ændringer.

En af de metoder, der vinder kraftigt frem inden for programmeringsverdenen, er multi-agent systemer, hvor adskillige helt eller delvist autonome softwareprogrammer - de såkaldte agenter - kommunikerer og samarbejder for at løse specifikke opgaver eller nå veldefinerede mål.

- Multi-agent systemet har sit ud-spring i kunstig intelligens, og metoden er baseret på elementer fra forhandlings- og beslutningsteori. Den har typisk været anvendt i større IT-systemer, f.eks. til overvågning af aktiekurser, hvor agenterne selv forhandler og træffer beslutninger om køb og salg af aktier. Og nu vil vi forsøge at få teknologien helt ud på produktionsgulvet, hvor agenterne skal agere på vegne af de enkelte maskinenheder, siger adjunkt og ph.d. Kasper Hallenborg fra Mærsk Institutet på Syddansk Universitet.

### Transport af bagage

Projektets første mål har været at bygge en nøjagtig digital kopi af et anlæg til bagagetransport for at vise, hvordan multi-agent systemer kan

Den første del af forfatterduoen, Klaus Holth, er lektor ved Mærsk Mc-Kinney Møller Institutet og Michael Wehner Rasmussen er koordinator i RoboCluster.

anvendes i en konkret industriel sammenhæng.

FKI Logistex A/S (det tidligere Crisplant), som blandt andet leverer bagagetransportanlæg til internationale lufthavne, har stillet viden og scenarier til rådighed.

FKI Logistex har programmer, som automatisk skriver styrekoder til transportanlæggene, men de kan ikke fungere optimalt uden manuel finpudsning af programmerne.

-Vi bruger mange ressourcer på at undgå flaskehalse i transporten. Det kan være en krævende proces, for flaskehalsene har det med at skjule hinanden, og de kan derfor kun fjernes én ad gangen, siger Lars Kristensen, der er product manager hos FKI Logistex A/S. Og han fortsætter:

-Gennem DECIDE har vi fået indsigt i, hvordan vi kan få et bedre og mere effektivt system til at udvikle styringer i vores anlæg.

Projektets partnere har haft meget realistiske data at arbejde med, idet metoderne er blevet testet på et scenario, der er lig det anlæg, som FKI Logistex er i færd med at levere til en større international lufthavn.

### Emulering - softwaremodel af transportbåndet

De programmer, der skal udvikles til lufthavnsanlægget, skal i sidste instans bruges til at styre flypassagerers bagage på et virkeligt, håndgribeligt transportbånd.

Den eneste måde at teste programmer og principper på er at sende nogle kufferter ud på transportbåndet og se, om de kommer rettidigt frem til de rette destinationer.

I praksis har man selvsagt ikke et komplet transportanlæg inklusive nogle tusinde kufferter og en snes

hjælpere til rådighed, når man skal teste sine programmer. Man må derfor opbygge en softwaremodel af transportbåndet.

Denne model skal ikke blot gengive hovedtrækkene af det virkelige transportbånd - dens opførsel skal være en nøjagtig kopi af det virkelige transportbånds. Når den nye software lægges i produktion, skal den nemlig fungere 100 procent korrekt med det samme.

En lufthavn kan ikke aflyse samtlige afgangene et par dage, mens programmørerne får rettet de sidste fejl - og der er altid fejl i ny styringssoftware.

En softwaremodel, der er en nøjagtig kopi af noget virkeligt forekommende, kaldes en emulering.

Lufthavnsmodellen skal ikke blot gengive det virkelige anlægs geometri og elementer korrekt, den skal også nøjagtigt efterligne adfærden. Når anlægget i virkelighedens verden gives besked om at styre en kuffert til højre ved forgrening, skal den virtuelle kuffert også føres til højre ved den virtuelle forgrening.

Man har i samarbejde med Simcon med stort udbytte brugt AutoMod fra Brooks Software til at opbygge en model for lufthavnsanlægget. Med AutoMod kan man opbygge 3D real-time modeller. AutoMod er velegnet til at emulere produktions- og transportprocesser.

### Programstruktur

Et fundamentalt træk ved den nuværende styring er anvendelsen af telegrammer, som er betegnelsen for en række tekstbaserede, strukturerede meddelelser, som sendes frem og tilbage mellem det fysiske anlæg og styringssoftwaren. Når der f.eks. læses en kuffert på transportbåndet

## Donaldson. Torit<sup>®</sup> DCE<sup>®</sup>

Torit<sup>®</sup> - støvfilteranlæg med *ovale* patroner

- Kompakt design
- Højere effektivitet
- Lavere energiforbrug
- Længere levetid

Donaldson Scandinavia ApS  
Ådalsvej 50, 2970 Hørsholm • Tlf.: 45 57 00 77 • Fax: 45 57 00 44  
ifs-dk@emea.donaldson.com  
www.donaldson.com



## PRODUKTER TIL INDUSTRIEN



- Strategisk afklaring
- Myndighedsrelationer
- Nye produktionsfaciliteter
- Opstart og idriftsættelse
- Test, validering og dokumentation
- Optimering

Carl Bro Gruppen · Granskoven 8 · 2600 Glostrup  
hea@carlbro.dk · Tlf. 4348 6367 · www.carlbro.com

Carl Bro   
Intelligent Solutions

# Dispenser



Manuel dispensering  
Timerstyret dispensering  
Sekvensdispensering  
Leveres med tilbehørskit

**Kr. 2.998,00**



**I&J FISNAR Scandinavia**

Lindegade 64 • 6070 Christiansfeld  
Tel.: 7631 2181 • Fax: 7631 2152 • E-mail: info@ijscandinavia.dk  
[www.dispensering.dk](http://www.dispensering.dk)

det, sendes der besked herom til softwaren. Tilsvarende sender softwaren instruktioner om, hvordan »skiftesporene« på transportbåndet skal indstilles for at bringe kufferten frem til dens bestemmelsessted.

Dette telegrambaserede interface mellem transportbåndet og styringssoftware bevarer i de nye programmer, men repræsentationen af telegrammerne internt i styringssoftware er ganske anderledes.

Artiklen vil ikke komme nærmere ind på, hvordan programmerne er struktureret, men vil nøjes med at anføre, at de i hovedtrækkene består af tre moduler.

- Der er ét modul, der dels tager sig af indlæsning og repræsentation af transportbåndet, og dels foretager ruteberegninger.
- Der er ét modul til logning af kufferternes vej gennem systemet. Dette modul bruges primært til statistik og test af performance.
- Og der er ét modul, der tager sig af hele kommunikationen med transportbåndet, omformning af telegrammer til objekter, validering af beskeder osv. Kort sagt hele den konkrete styring af transporten. Dette modul er langt det mest omfattende både i antal linjer kode og i arbejdsindsats.

De tre moduler kan naturligvis kommunikere med hinanden, men der er tale om en meget løs kobling. Programmerne er alle skrevet i Java.

I det følgende ses nærmere på, hvor den nuværende styring har sine begrænsninger, og hvor der med fordel kan sættes ind med forbedringer. Vi vil også foretage en vurdering af, om det faktisk er lykkedes for os at skabe markant bedre resultater. Analysen vil foregå på et overordnet niveau, hvor der vil blive diskuteret styringsprincipper, men ikke hvordan disse konkret implementeres i programpakkerne.

### Styringsprincipper

Når et fly foretager en mellemlanding, forventer passagererne, at deres kufferter bliver bragt ombord på næste fly. Bagagen fra det netop landede fly skal derfor dirigeres hen til en række forskellige afgange, vel at mærke i god tid inden næste fly letter.

Kufferterne læses på transportbåndet ved en toploader. Her aflæses også dens destination. Kufferterne føres nu gennem en række forgreninger til et afkast, hvor den læses af. Man kan sammenligne transportbåndet med en modeljernbane,

Arbejdet med lufthavnsmodellen er blevet udført inden for rammerne af IT-korridor projektet DECIDE, som har følgende deltagere: RoboCluster, FKI Logistex, Bang & Olufsen, Lego Koncernen, Odense Staal-skibsværft, Grundfos, Simcon, Mesh-Technologies, Teknologisk Institut, Fyns ErhvervsCenter, Mærsk Instituttet og Institut for Matematik og Datalogi ved Syddansk Universitet, samt Ingeniørhøjskolen Odense Teknikum.

hvor forgreningerne svarer til skiftesporene.

Det drejer sig nu i grove træk om at få en kuffert så hurtigt gennem systemet som muligt. Hvis en kuffert er helt alene på transportbåndet, kan man foretage en matematisk beregning, der viser, hvilken rute der er den hurtigste. Man kan altså med andre ord på forhånd sige, hvordan forgreningerne skal være indstillet for at føre en kuffert fra toploader A til afkast B. Denne metode kaldes *shortest path*.

Så længe belastningen er lav, giver dette ikke problemer. Men hvis der er mange kufferter på transportbåndet er der en meget reel risiko for kødannelse. Det tager lidt tid, hver gang en kuffert fjernes fra transportbåndet ved et afkast, så de efterfølgende kufferter må vente, til de kan komme videre eller selv blive afkastet. Hvis kuffert-tætheden er tilstrækkeligt stor, breder køen sig langt bagud, og den kan blokere en forgrening, så adgangen til en ellers fri bane bliver blokeret. Herved kan kødannelse brede sig som ringe i vandet. Man kalder dette fænomen for tilsanding.

Man kan til en vis grad undgå kødannelserne ved at tage højde for, hvor og hvornår køerne typisk opstår, når man skriver programmerne, der styrer kufferttransporten. En sådan kodning tager imidlertid udgangspunkt i den konkrete udformning af transportbåndets geometri. Man kan ikke umiddelbart overføre resultaterne fra én lufthavn til en anden.

Der er derfor to problemer med den nuværende metode. For det første er *shortest path* princippet for ufleksibelt, og for det andet er udgifterne til

vedligehold af styringsprogrammerne for høje. Dybest set skyldes problemerne, at man anvender en statisk metode til at løse et dynamisk problem. Selv om man ved, hvornår et fly lander, ved man ikke, hvor meget bagage det har med sig, og hvordan bagagen skal fordeles på videre afgange. Hvis et fly er forsinket, kan det være, at det lander ved en anden gate end planlagt.

### Kuffertens vej

En oplagt måde at forbedre styringen på er at anvende dynamiske algoritmer til styringen af den enkelte kufferts vej fra toploader til afkast. *Shortest path* algoritmen går ud fra, at det altid tager lige lang tid at komme fra én bestemt toploader til ét bestemt afkast. Denne antagelse er urealistisk, når der opstår køer, så i denne situation, er det sandsynligvis en god idé, hvis kufferten vælger en alternativ rute. Herved kan kufferten måske komme til at genere andre kufferter i deres færd, så de må justere deres ruter.

Optimering af den enkelte kufferts rute kan derfor føre til sub-optimering af den samlede transport, medmindre man foretager en central beregning, der tager hensyn til alle kufferternes behov.

Og det er netop, hvad man har gjort. Metoden kaldes *routing on way*, fordi der kan foretages dynamiske justeringer undervejs.

Metoden forudsætter, at man til ethvert tidspunkt ved, hvor samtlige kufferter befinder sig, og hvor de er på vej hen. Den foretager så beregninger, der viser, hvilken rute hver eneste kuffert skal følge. Kufferterne bevæger sig nu en stund efter de nye resultater, hvorefter der atter foretages genberegninger, osv.

For at implementere algoritmen må man konstant indsamle oplysninger om kufferternes tilstand, foretage beregninger og give instrukser om, hvor kufferterne nu skal føres hen.

Der skal med andre ord oprettes et kommunikationsframework, som man kan bruge til at opsamle data og sende beskeder med. Man har i dette tilfælde valgt at bruge JADE-frameworket, som er udviklet af Telecom Italia. JADE er skrevet i Java og giver en basiskommunikationsstruktur for multiagentsystemer.

### Test af strategier og algoritmer

For i praksis at sammenligne *shortest path* og *routing on way* algoritmerne foretog parterne i DECIDE projektet en række simulerede kørsler på lufthavnsanlægget. ►

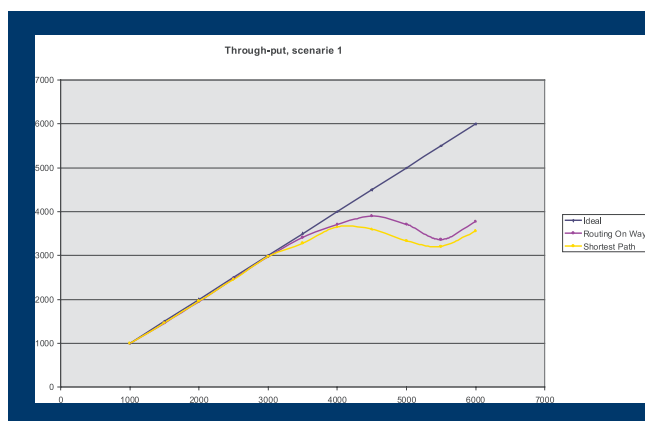
Der findes forskellige mål for effektiviteten af bagagetransporten. Her ses på through-put, som er antallet af afkastede kufferter pr. tidsenhed som funktion af antal loadede kufferter pr. tidsenhed.

Umiddelbart kan man sige, at der gerne skulle komme lige så mange kufferter ud af systemet, som føres ind i det. Tingene er imidlertid lidt mere komplicerede. Ved lav belastning vil der ikke opstå køer, og der er derfor ingen problemer med through-put. Hvis belastningen øges, så der opstår køer, falder det relative through-put. Dette er ikke nødvendigvis fatalt, fordi erfaringen viser, at der i løbet af en dag er lange perioder med lav belastning af anlægget. Hvis belastningen derfor snart falder, kan en kø opløses »af sig selv«, og kufferterne når alligevel frem. Hvis perioder med høj belastning afløses af stille perioder, kommer alle kufferter ud af systemet, selv om det går langsommere af få dem ud end at få dem ind på transportbåndet.

På den anden side kan det være, at de når for sent frem. Jo større køerne er, jo større er sandsynligheden herfor. Det er derfor en god idé under alle omstændigheder at søge at undgå kødannelse ved at bruge forbedrede routningsalgoritmer.

Desuden er transportanlæggene meget dyre, og der er ingen grund til at overdimensionere dem, hvis man kan klare sig med mindre.

Man kan altså forvente, at det relative through-put falder efterhånden som belastningen af systemet øges. Der er naturligvis en øvre grænse for, hvor mange kufferter i timen, der kan læses ind i systemet, men holder man sig under denne grænse, er



*Som det fremgår af graferne, er dynamisk routing faktisk bedre end statisk, især hvis der er »vanskelige« begyndelsesbetingelser. Graferne viser imidlertid også, at der endnu er et stykke vej til en tilfredsstillende løsning er opnået.*

det vores forhåbning, at man med forbedrede algoritmer kan forøge through-put.

Når en kuffert lægges på transportbåndet, bliver den lagt ned i en stor balje. Der blev foretaget sammenligninger af shortest path og routing on way på to scenarier. I det ene er der rigeligt med tomme baljer til rådighed, når de første kufferter føres ind i systemet. I det andet får kufferterne lov til at hobe sig op ved to af i alt syv toploadere, inden der kommer tomme baljer. Resultaterne af kørslerne er vist på de nedenstående grafer. Den øvre grænse for antal kufferter i timen er 6500.

Når routing on way ikke giver et tilfredsstillende through-put, skyldes det i det væsentlige ikke, at der opstår køer ved afkastene. Når en kuffert er smidt af systemet ved et afkast, befinder den nu tomme balje sig stadig på transportbåndet. Den føres tilbage til toploader-området, så den kan benyttes til en ny kufferttransport. Problemet er imidlertid, at de tomme baljer på deres vej tilbage blokerer for kufferter, der netop er blevet lagt på en balje.

Routing on way forhindrer således tilsanding i afkaster-området, men ikke i toploader-området. En mulig løsning på dette problem er at styre toploadere, sådan at man midlertidigt nedsætter deres hastighed, mens en hel række tomme baljer passerer forbi. En anden mulighed er at styre kufferter på vej ind i systemet hen til en toploader, der ikke er overbelastet eller blokeret.

Denne mulighed er endnu ikke implementeret. Derimod har man efterprøvet toploaderstyringen med gode resultater. Det har også givet markant forbedrede kørsler at lave en mere effektiv tilbagestrømning af tomme baljer til netop de toploadere, hvor der hober sig kufferter op.

Det er desværre ikke »gratis« at afvikle alle disse forbedringer i kørslerne. Nogle af dem er meget cpu-tunge, og der ligger et stort arbejde forude med at optimere programmerne. Hvordan kan man formindske memoryforbruget, hvor tit skal man udføre genberegningerne af ruterne, hvor tit skal man kontrollere belastningen af toploadere osv?

## Sammenlægning af SRO-anlæg på renseanlæg

Dansk Styringsteknik A/S er klar med nyt koncept for trinvis sammenlægning af SRO-anlæg på renseanlæg i de nye storkommuner.

Den forestående kommunalreform skaber mange udfordringer for landets forsyningsvirksomheder - eksempelvis i forbindelse med drift af renseanlæg. Forskellige teknologiplatforme skal samkøres, der skal etableres fælles vagtsystemer dækkende større geografiske områder, medarbejderne skal uddannes i nye systemer og der er behov for centralovervågning med decentral sty-

ring og sammenhold af data fra flere anlæg med henblik på driftsoptimering. I den forbindelse er Dansk Styringsteknik A/S klar med et nyt koncept, Aqua-Cept, som sikrer de nye storkommuner en trinvis overgang til et samlet komplet SRO-anlæg med central overvågning af alle renseanlæg og pumpestationer samt mulighed for decentral styring via web-klienter.

Aqua-Cept tager højde for de udfordringer, som sammenlægning af forskellige systemer medfører, og giver samtidig mulighed for kundetilpas-

set overbygning. En af fordelene ved konceptet er, at det kan integreres efter behov. Eksempelvis startende med et fælles vagtsystem, som også kan hente data fra de eksisterende SRO-anlæg, til løbende at udskifte og implementere pumpestyringer og renseanlæg ét efter ét.

En anden fordel er, at systemet bygger på standard software som iFix, Crystal Reports, Windows og MS-SQL. Det er med til at sikre den rigtige markedspris og fritstiller kunden med hensyn til valg af fremtidige systempartnere.