



LUND UNIVERSITY

En on-linesimulator för operatörsstöd

Nilsson, Bernt

1991

Document Version:
Förlagets slutgiltiga version

[Link to publication](#)

Citation for published version (APA):
Nilsson, B. (1991). *En on-linesimulator för operatörsstöd*. (Research Reports TFRT-3209). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

General rights

Unless other specific re-use rights are stated the following general rights apply:
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

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

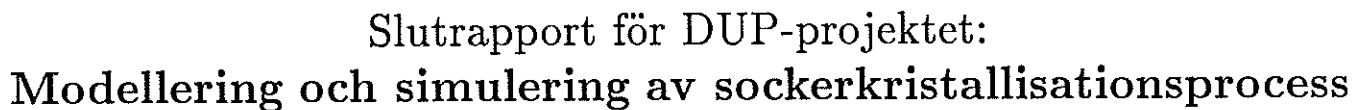
Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Bernt Nilsson



Institutionen för Reglerteknik
Lunds Tekniska Högskola
Maj 1991

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		Document name	
		Date of issue May 1991	
		Document Number CODEN: LUTFD2/(TFRT-3209)/1-44/(1989)	
Author(s) Bernt Nilsson		Supervisor Björn Wittenmark	
		Sponsoring organisation Swedish Board of Technical Development, contract 89-2740	
Title and subtitle An On-line Simulator for Operator Support. (En on-linesimulator för operatörsstöd.)			
Abstract Simulation is gaining more and more interest in process industry due to the increase of computer power and decrease in computer investment. In this report a simulator for operator support and training is presented and discussed. The simulator simulates a model of a process for crystallization of beet sugar. The process is composed of a large number of semibatch boiler pans, periodic centrifugals and buffertanks. The simulator can be used for i) training and education, ii) decision support, iii) production planning and iv) development of new control strategies. The simulator was found to be easy to use because of the development of a user friendly interface based on icons and colours, information zooming, multiple views, direct object manipulations and animation. The modell representation is based upon object orientation methodology. New ideas for batch process model description based on Grafset is presented.			
Key words Modelling; process models; hierarchical systems; computer simulation; simulation languages; software tools; object-oriented modelling; man-machine interaction			
Classification system and/or index terms (if any)			
Supplementary bibliographical information			
ISSN and key title			ISBN
Language Swedish	Number of pages 44	Recipient's notes	
Security classification			

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

En on-linesimulator för operatörsstöd

Slutrapport för DUP-projektet:
Modellering och simulering
av
sockerkristallisationsprocess

Bernt Nilsson

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 LUND
Sweden

© 1991 by Bernt Nilsson. All rights reserved
Published 1991
Printed in Sweden

Innehåll

Förord	5
Simulering - en teknik under förändring	6
1. Bakgrund	12
2. Beskrivning av sockerkristallisationsprocessen	14
2.1 Sockerbruket	14
2.2 Sockerhuset	15
2.3 Ett koksteg	16
2.4 Processtyrning	18
3. Sockerhussimulatoren	19
3.1 Introduktion till G2	19
3.2 Simulators användning	20
3.3 Presentationsformer	25
3.4 Interaktionsformer	28
4. Modellrepresentation	30
4.1 Strukturerad modellering	30
4.2 Processmodell i G2	33
5. Modellbygge	36
5.1 Beskrivning av kokarna	36
5.2 Modell av maischarna	40
5.3 Centrifugbeskrivningar	40
5.4 Modell av tankar och pumpar	42
5.5 Modelldiskussion	42
6. Erfarenheter	43
7. Referenser	44

Förord

Under 1990 bedrevs ett DUP-projekt med namnet *Modellering av sockerkristallisationsprocess* vid institutionen för Reglerteknik vid Lunds Tekniska Högskola. Detta är en presentation av den aktivitet som bedrevs inom projektet. STU-ramprogrammet DUP, Driftutvecklingssystem för processindustrin, har för avsikt att sprida ny teknik från till exempel informationsteknologin till kontrollrummen inom svensk processindustri. Detta projekt har studerat simulatorer vilket passar bra in i DUP:s målsättning.

Projektet kan delas upp i två faser. Den första fasen av projektet koncentrerades på objektorienterad modelleringsmetodik. Grunden för denna studie utgör ett tidigare STU-projekt, förkortat CACE, som har utvecklat nya idéer för datorstödd modellering och simulering. Projektets andra fas har fokuserats på en prototyp av en on-linesimulator av kristallisationssteget vid Hasslarps sockerbruk och prototypen är implementerad i G2.

Min förhoppning är att detta arbete kan tjäna som inspirationskälla för nya och framtida tillämpningar av simuleringsteknik. Denna rapport är därför riktad till morgondagens användare inom svensk processindustri. Nödvändig informationsteknologi, modelleringsmetodik och simuleringsteknik kommer att presenteras efterhand för att öka förståelsen och förhoppningsvis minimera behovet av förkunskap.

Jag är den som har varit mest aktiv i detta projekt men långt ifrån den enda. Projektledare har varit Björn Wittenmark, institutionen för Reglerteknik, och Claes Gudmundsson, Sockerbolaget. Stor hjälp har vi haft av Magnus Porle, Sockerbolaget, som har stått för praktiska kunskaper om driften av kristallisationsprocesser. Jag måste också få tacka Karl-Erik Årzén, Reglerteknik, för Grafcet-verktygslådan i G2 samt många och givande diskussioner om möjliga och omöjliga tillämpningar av G2. Speciellt tack till Åsa Nilsson, Magnus Porle, Björn Wittenmark och Karl-Erik Årzén för genomläsning av och kommentarer på manuskriptet.

Detta projekt har finansierats av STU under kontrakt 89-2740.

Simulering - en teknik under förändring

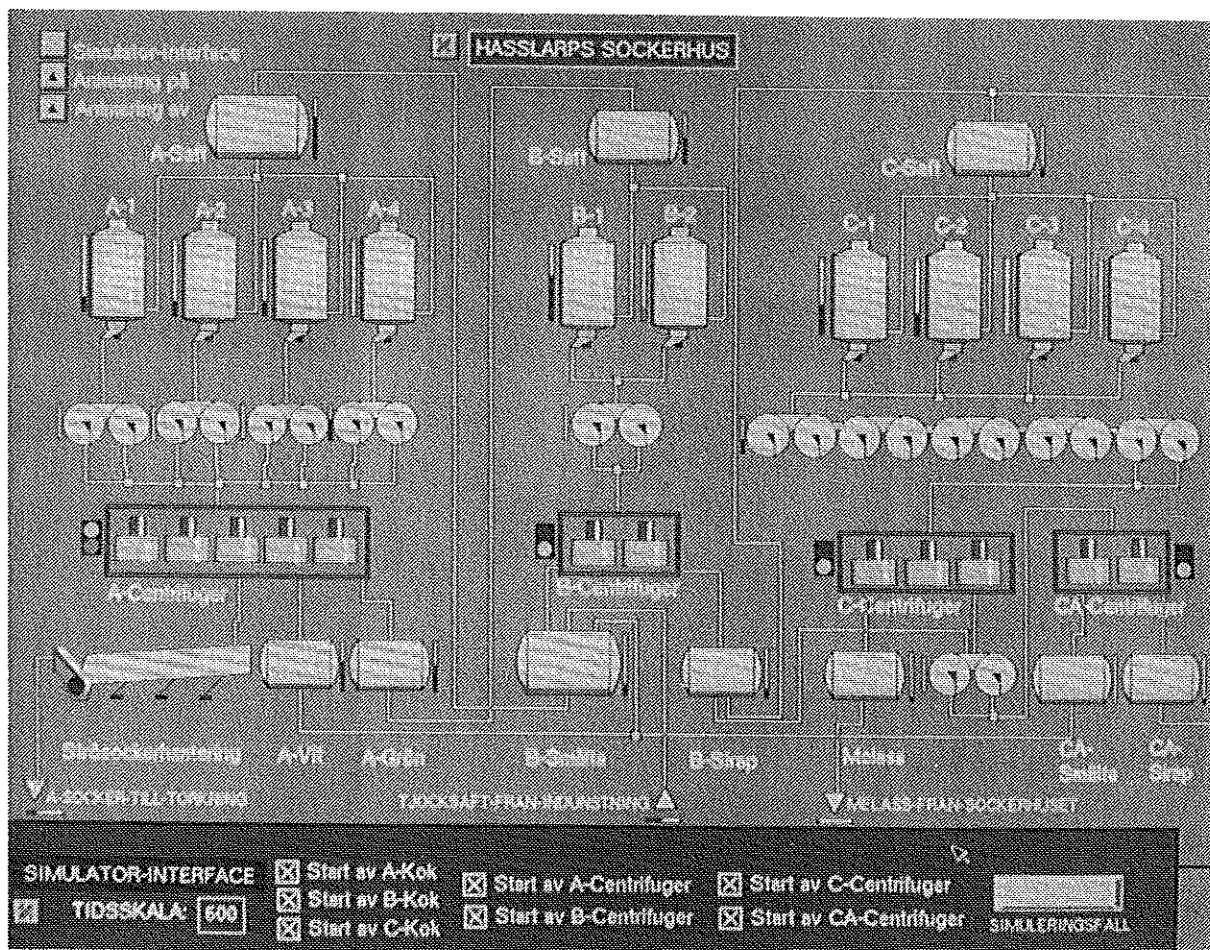
Den snabba utvecklingen inom informationsteknologin har inneburit att användandet av datorer har spridits till många nya användare och till nya tillämpningar. En nygammal tillämpning är simulering. Simulering är en gammal och beprövad teknik. Genom att simulera ett system, det vill säga härma eller imitera systemet, så kan vi studera hur detta beter sig i olika situationer utan att behöva utsätta systemet för dessa situationer i verkligheten. Simulering har därför länge använts där det av säkerhetsskäl varit befogat, till exempel vid simulering av flygplan och kärnkraftverk. Dessa simulatorer har ofta varit dyra men de ekonomiska faktorer har kommit i andra hand.

Det nyvunna intresset för simulering baseras på att ny informationsteknologi har sänkt den ekonomiska kostnaden för att bygga simulatorer vilket gör tekniken intressant för nya tillämpningar. Ett exempel på detta nygamla intresse är att introducera simulatorer i kontrollrummen inom processindustrin. Användare skall vara operatörer vid processen vilka är en ny grupp av användare av simuleringstekniken. I denna rapport presenteras just en sådan simulator.

Denna möjlighet att utnyttja en teknik i ny tillämpning och av en ny användargrupp ställer också krav på teknikutformningen. Viktiga aspekter blir hur slutanvändaren skall uppfatta simulatoren. Vad kan man utnyttja tekniken till och vad vinner man, blir frågor som måste besvaras. Det arbete som presenteras här är mer en utforskning av dagens simulator-teknik än en kartläggning av hur tekniken skall användas. Arbetet kan förhoppningsvis ge inspiration och ligga till grund för fortsatt arbete för spridning av simulatortekniken.

Sockerhussimulatorn

För att testa nya idéer och visa hur morgondagens simulatorer kan se ut byggdes en prototypsimulator. Simulatoren simulerar ett processteg vid Hasslarps sockerbruk, det så kallade sockerhuset, där vätska innehållande socker kokas så att det kristalliserar ut. I figur 1 ses hur processteget



Figur 1. Simulatorns presentation av sockerhuset med animering.

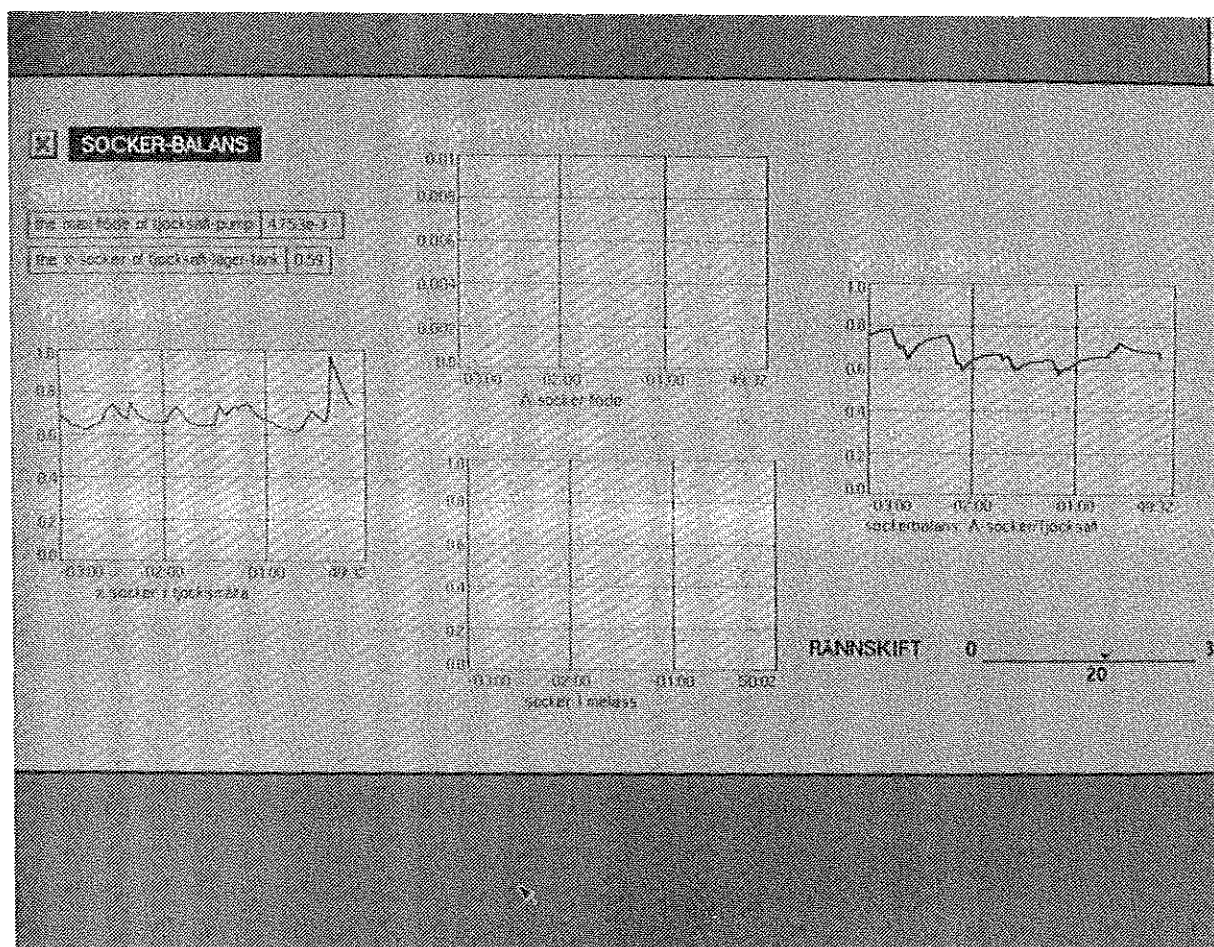
presenteras för operatören. Här är figuren tyvärr svart-vit men i verkligheten är bakgrunden kadettblå och alla stålkärl gråa med vitt ljusblänk och mörkgrå skuggning. Simulatoren kan användas på flera sätt i operatörens dagliga arbete. Denna prototyp är kapabel för i första hand fyra typer av tillämpningar:

- Utbildning och träning av operatörer vid nominell drift.
- Utveckling av operatörens styrstrategier vid felsituationer.
- Beslutsstöd on-line för styrstrategistudier vid nominell drift.
- Produktionsplanering vid förändring i produktionen.

Ett scenario

Simulatorn kan användas för att studera processens nominella drift. Operatören kan utbildas och tränas på de handgrepp och styråtgärder som krävs för att uppfylla alla krav. De vanligaste driftsstörningarna, i form av varierande koncentrationer och flöden, kan simuleras i simulatorn och operatören lär sig uppfatta och hantera dessa störningar. Operatören måste skapa sig en bild av processen och förstå dess beteende. Detta görs genom att interaktivt arbeta med simulatorn och aktivt finna information om den simulerade processens tillstånd.

En vidareutveckling av simulatorn är att utnyttja den för operatörsstöd vid driften av processen. Simulatorn kan då innehålla skräddarsydda bilder, eller vyer, för en given styråtgärd. Ett exempel på en sådan vy är rännskiftsstyrning. Processtegets interna recirkulation styrs med denna inställning och därmed dess ekonomi. Om simulatorn är kopplad on-line



Figur 2. En vy för simuleringsstudier av rännskiftsstyrning

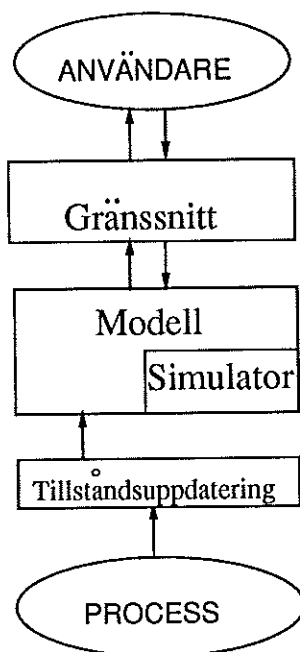
kan operatören prova några olika inställningar och utvärdera simuleringarna innan han eller hon gör det på den verkliga processen. Simulatoren används alltså som stöd vid olika beslut vid driften av processen.

On-linesimulering

Processmodeller som skall användas i och kring driften av en process kommer att nyttjas under en lång tid. Då processen är under ständig utveckling och förändring, innebär det att processmodellen också måste förändras. Processmodellen måste alltså underhållas för att passa de förändrade behoven.

I vårt fall skall simulatoren användas för två huvudtyper av simuleringar, off-line och on-line. Off-linetillämpning är till exempel att utbilda och träna operatörer på de olika handgrepp och beslut som de måste fatta vid driften av sockerhuset.

En annan tillämpning är att göra prediktionssimuleringar av olika styråtgärder som sedan kan ligga till grund för de beslut som operatören skall göra vid driften av processen. Här skall processens beteende under ett till två dygn presenteras direkt eller under kort tid (skalad realtid). Ett verktyg som detta ger underlag för studier av överordnade styrstrategier för hela processavsnittet och kan användas vid driften. Denna typ av on-linesimulator kan delas upp i olika delar: tillståndsuppdatering, modell, simulator samt användargränssnitt. Denna uppdelning åskådliggörs



Figur 3. Arkitekturen i en on-linesimulator.

i figur 3. För att simulatören skall kunna användas on-line så måste den ständigt ha rätt tillstånd. Processen ändrar sig hela tiden, vilket gör att modellen måste uppdateras varje gång en simulering skall göras. Efter-som processen alltid ändrar sig, kan tillstånden ej antas vara konstanta utan måste rekonstrueras dynamiskt från mätdata. Detta kan göras med till exempel Kalman-filter. Denna del av en on-linesimulator kommer inte att diskuteras i denna rapport och har inte varit en del av projektet.

Erfarenheter

Användandet av en dynamisk simulator som diskuteras i denna rapport har enligt Sockerbolaget stora möjligheter att utgöra ett användbart hjälpmedel för operatörerna i det dagliga arbetet vid processen. Detta på grund av att simulatorns användargränssnitt och organisation är väl anpassad till operatörerna och deras bild av processen. G2, den mjukvara som utnyttjas för prototypimplementering, visade sig vara ett bra hjälpmedel för snabb utveckling av prototyper som denna. Det har tagit 2 manmånader att göra modellbygge, simuleringstester och modellvalidering samt utveckling av användargränssnittet. Modellvalidering mot dynamiska processdata har inte gjorts men simuleringar visar att modellen predikterar processens beteende ganska bra. Modellrepresentationen som baseras på hierarkisk dekomponering och informationszoomning visar sig vara mycket användbar för presentation. Vid modellutveckling är denna organisation lite otyplig och ett alternativ är då separering av modellrepresentation och användargränssnitt, men detta har inte testats.

Rapportens uppläggning

I kapitel 1 görs en kortfattad presentation av projektet och dess bakgrund. I det följande kapitlet om sockerkristallisationsprocessen beskrivs detta processteg, det så kallade sockerhuset, dess komponenter samt de överväganden som görs under drift.

Simulering kan användas för flera olika ändamål och på många olika sätt. Hur dynamisk simulering kan göras med prototypsimulatören diskuteras och utvecklas i många scenarier i kapitel 3. Här presenteras också de idéer som ligger bakom simulatorns användargränssnitt. En förutsättning som gör simulatören lättanvänd är att presentationen av och interaktion med simuleringen görs naturlig för operatören. Den moderna informationsteknologin ger stora möjligheter till skraddarsydda och tillämpningsanpassade användargränssnitt. G2, den mjukvara som utnyttjas för prototypsimulatören, är ett bra exempel på state-of-the-art inom detta område.

Processen och kunskapen om processen förändras med tiden. Detta gör det nödvändigt att ständigt utveckla, förfinas och underhålla processmodellen. Det får inte vara kostsamt och svårt utan skall kunna göras av personalen vid processen. De som bygger om processen skall också bygga om processmodellen. Objektorienterad modellering underlättar just detta vilket diskuteras i kapitel 4, Modellrepresentation.

Slutligen återfinns modelltekniska överväganden och modellernas interna detaljer i det näst sista kapitlet om modellbygge och i det sista kapitlet diskuteras de erfarenheter som projektet givit.

1. Bakgrund

Denna rapport är en presentation av projektet *Modellering av sockerkristallisationsprocess*. Projektet är en del av det STU-finansierade ramprogrammet DUP - Driftutvecklingssystem för processindustrin. Projektet har i huvudsak studerat processmodeller och modelleringsmetodik samt gjort en prototyp av en online-simulator. Arbetet förväntas fortsätta i en andra etapp, ett fortsättningsprojekt, mer koncentrerat mot utbyggnaden och användandet av en prototypsimulator vid drift.

Målsättning

Detta projekt har fokuserats på hur ny simuleringsteknik kan användas vid on-linesimulering av ett satsvis processteg, sockerkristallisation. Modellen skall vara underlag för olika typer av studier; utbildning av nya operatörer, utveckling för nya styrstrategier samt ge ökad förståelse av processen.

Arbetssätt

I princip kan arbetet indelas i två olika delar. I den första delen av projektet studerades objektorienterad modelleringsmetodik. Här testades och utnyttjades verktyg från det STU-finansierade ramprogrammet "Datorbaserade hjälpmedel för utveckling av styrsystem" (förkortat CACE). De idéer som utvecklades inom CACE-projektet är nu under implementering i ett annat STU-projekt. Eftersom detta modelleringsprojekt låg före implementeringen av CACE-prototypen så gjordes denna studie konceptuell och ej i någon mjukvara. En första version av CACE-prototypen kommer våren 1991.

Den andra delen av projektet utgjordes av byggandet av en prototyp. Denna prototyp utgör en testbänk för nya idéer samt ger en vision av hur morgondagens simulatorer kan se ut och hur de kan fungera. I ett första steg gjordes utveckling och strukturering av modell och simulator för att underlätta framtida underhåll. I ett andra steg koncentrerades arbetet på gränssnittet till en tänkt slutanvändare. För att simulatoren skall nå ända ut till operatören krävs ytterligare ett steg där den anpassas till och

integreras i organisationen vid processen. Det kan utgöra ett eventuellt fortsättningsprojektet.

Val av simulatormiljö

Andra delen av projektet utgjordes av byggandet av en prototyp. Två exempel på moderna mjukvaror som utgör tänkbara kandidater är för denna tillämpning Satt Line och G2. Satt Line, från SattControl, är ett regler- och styrsystem baserat på objektorientering och grafisk programmering. G2 är ett expertsystem för realtidsapplikationer från Gensym Corporation (Gensym, 1990).

Vid institutionen för Reglerteknik har G2 använts aktivt under ett par år. Det finns stor erfarenhet och goda kunskaper om G2. Institutionen hade ingen erfarenhet av och hade inget fungerande Satt Line-system vid tidpunkten då valet gjordes. Projektet valde därför G2 av tids- och kunskapsmässiga skäl. G2 är ursprungligen ett expertsystem för realtidsstillämpningar men ger också goda möjligheter för modellbygge och simulering. Modellerna och modelleringsmetodiken har därför modifierats för att passa G2. Vi har alltså blivit tvungna att skilja på den önskade modelleringsmetodiken och själva implementeringen av prototypsimulatorn i G2.

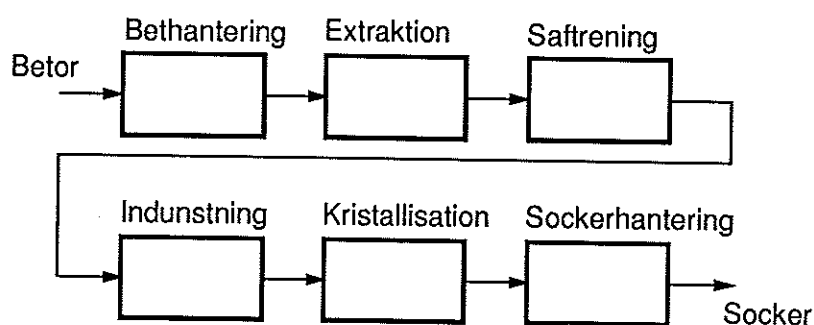
G2 bör köras på Unix-arbetsstationer med extra primärminne, stor disk, färgskärm och under X Windows. Vid institutionen för Reglerteknik körs G2 på Sun-arbetsstationer och då i första hand på Sparc2-maskiner med färgskärm, extra primärminne på 24 Mega byte. G2 kan också köras på Compac 386 eller 486 med Unix och X Windows.

2. Beskrivning av socker-kristallisationsprocessen

I Sverige tillverkas strösocker av sockerbeter i ett antal sockerbruk. Dessa sockerbruk är lokaliserade i Skåne, på Öland och Gotland. Det sker en ständig rationalisering och effektivisering av dessa bruk. Tillverkning av socker räknas till livsmedelsindustrin men processtekniskt liknar det mer en ren processindustri. Den framtida utvecklingen kommer att göra detta än mer tydligt med större enheter med helt kontinuerlig produktion.

2.1 Sockerbruket

Ett sockerbruk är uppdelat i ett antal processteg; bethantering, extraktion, saftrening, indunstning, kristallisation och strösockerhantering. I bethantering sorteras, tvättas och strimlas betan till så kallade snitslar. Vid extraktionen extraheras sockret ur betsnitslarna. Betmassan går till djurfodertillverkning medan vätskan, den så kallade råsaften går vidare i processen. Vid saftreningen blandas råsaften med kalk som gör att föroreningar faller ut. Utfällda föroreningar kan sedan filtreras bort. Den renade tunnsaften går nu till indunstningssteget. Här indunstas tunnsaften till tjocksaft som sedan går till kristallisation i sockerhuset. Socker kristalliseras och separeras från restprodukten melass. Sockret torkas och siktas innan det lagras i silo. Runt dessa processteg finner man också en del stödprocesser för vattenrening, betmassahantering, kalkbränning, ång- och elproduktion.



Figur 2.1 Processtegen vid betsockertillverkning.

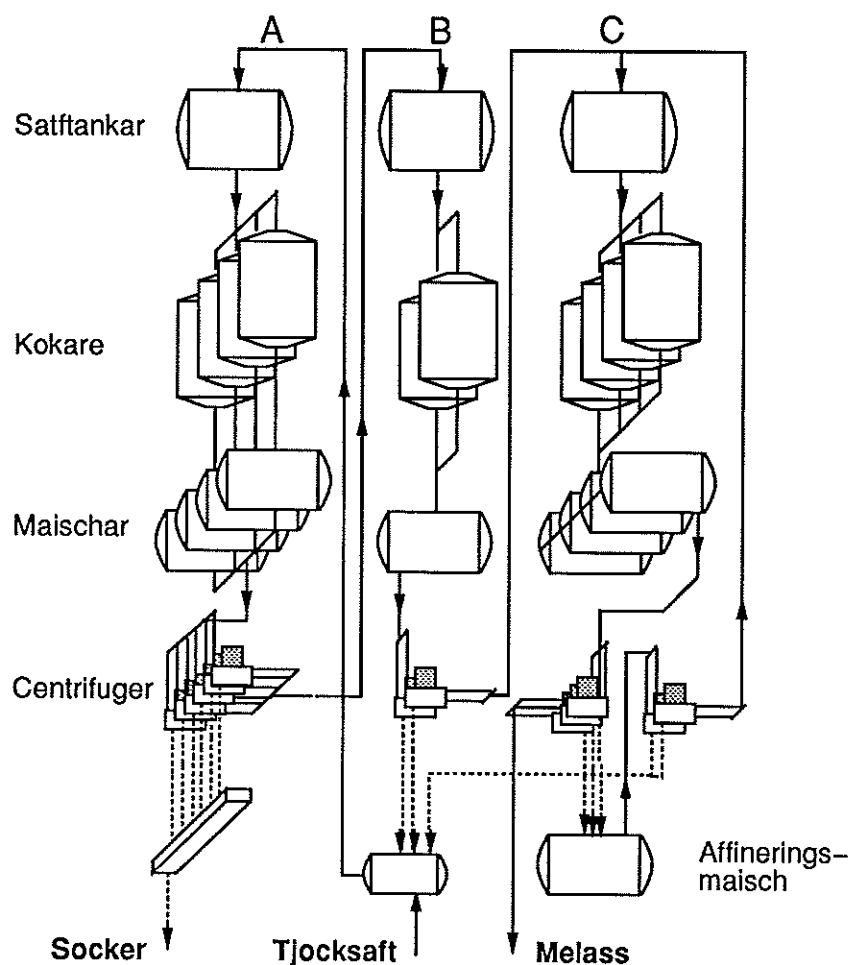
Strösockertillverkning är en livsmedelsprocess som påminner mer om en kontinuerligt tillverkande processindustri än en livsmedelsindustri. Alla processteg i ett sockerbruk, utom ett, är kontinuerliga, det vill säga har ett kontinuerligt inflöde och utflöde. Kristallisationen är det processteg som är uppbyggt på satsvisa processkomponenter, så som kokare, maischar och centrifuger. Kristallisationssteget, eller sockerhuset, är det svåraste processteget att styra bland annat på grund av mycket långa dötider och variationer i tjocksaftens kvalitet. Mer om sockertillverkning kan finnas i Svenskt socker (Larsson, 1989).

2.2 Sockerhuset

Processteget för sockerkristallisation brukar kallas sockerhuset. I sockerhuset kokas tjocksaften så att sockret kristalliserar; detta görs i kokpannor. Den kokade massan lagras i maischar och eventuellt kylas i maischar, innan den nu så kallade fyllmassan går vidare till centrifugerna. Här separeras kristaller och vätska, eller sirap, från varandra. Kristallerna är den önskade produkten och vätskan är restprodukten. Denna procedur kallas ett koksteg och detta första steg, A-kok. Restprodukten, från A-koket, innehåller fortfarande mycket socker, därför låter man sirapen gå igenom en likadan procedur som ovan med kokning, lagring och centrifugering och detta kallas B-kok. Sockret från B-koket recirkuleras till A-koket, medan vätskan, B-sirap, går till ytterligare ett koksteg, C-kok. Precis som i B-koket recirkuleras C-sockret till A-koket medan vätskan, melass, är processtegets restprodukt. Processens olika koksteg ses tydligt i figur 2.2.

För att förstå processen krävs ytterligare några detaljer. Sockret från C-koket recirkulerar inte direkt utan det går till affineringen. C-sockret smältes delvis med tunnsaft i affineringsmaischen. Fyllmassan centrifugeras och CA-sockret recirkuleras till A-koket medan vätskan, CA-sirap, recirkuleras internt till C-koket. Man kan säga att C-socker inte recirkuleras direkt utan tvättas först.

A-sirap delas upp i två fraktioner, A-grön och A-vit. A-grön-sirap går till B-koket medan A-vit-sirap recirkuleras direkt till A-koket. Detta görs vid centrifugeringen genom att sätta det så kallade rännskiftet. Rännskiftet är den viktigaste styrvariabeln för överordnad styrning av processen. Vi återkommer till detta i 2.4 Processtyrning.



Figur 2.2 Processchema över sockerhuset i Hasslarp.

2.3 Ett koksteg

Ett koksteg består av tre huvudkomponenter: kokare, maisch och centrifuger. I kokaren kristalliseras sockret och i maischen mellan-lagras och eventuellt kylls (C-maischar) koket. Separationen av kristaller och resterande vätska sker i centrifugerna. Före kokarna, som är satsvisa, finns buffertar i form av safttankar.

Kokare

Själva kristallisationen sker alltså i kokarna. Den saft som skall användas dras in i kokaren från safttanken. Därefter förkokas saften till en given temperatur och sockerinnehåll, den så kallade ymppunkten. Under förkoket sker samtidigt påfyllning av saft för att vätskenivån ska hållas konstant.

Kristallisationen startar genom att satsen ympas, det vill säga små sock-

erkristaller tillsättes. För att få en jämn sockerkvalitet skall kristallerna vara lika stora vilket innebär att de tillsatta kristallerna skall växa lika fort. Detta åstadkoms med en så jämn övermättning av socker som möjligt. Hastigheten i kristalltillväxten ökar med ökande övermättning. Vid hög övermättning finns risk för kristallbildning vilket orsakar ojämn sockerkvalitet vilket inte är önskvärt. Efter ympningen fortsätter förångningen av vatten. Koksafte tillsätts under kokfasen för att hålla en jämn övermättningen samt öka kokets massa till det fyrdubbla. Denna kokfas är den som i regel tar längst tid, från 3 till 10 timmar (A- resp. C-kok). Vid en given kokmängd avslutas påfyllnaden och koket stramas in. Efter instramningen kan koket tömmas i en maisch och efter tömmningen kan kokaren startas igen.

I A-koket används fyra parallella kokare. En kokare startas när tillgången på saft bedöms som tillräcklig. Normalt är alla kokare alltid i drift och endast små väntetider förekommer. På motsvarande sätt styrs B-kokets två kokare och C-kokets fyra.

Maisch

Kokmassan förvaras i en maisch under några timmar. Maischen fungerar också som bufferttank mellan de satsvisa kokarna och de efterföljande centrifugerna. I A-koket finns dubbla maischar till varje kokare. I B-koket finns bara en trippel-maisch till de två kokarna. I C-koket finns ett batteri av maischar i serie (17 stycken) till alla kokarna. I C-maischarna kyls fyllmassan och kristallerna fortsätter därför att växa. Uppehållstiderna i A- och B-maischar är cirka 2 timmar medan i C-maischarna cirka 36 timmar (1,5 dygn).

Centrifuger

Fyllmassan från maischarna tas till ett batteri av centrifuger. I centrifugerna separeras kristaller och resterande vätska. Under centrifugeringen spolas kristallerna rena med däckvätska. Detta gör att kvaliteten på sockret ökar men nackdelen är att socker löses upp och sänker kvantiteten.

Antal och typ av centrifuger kan vara olika. I A-koket utnyttjas fem stycken satsvisa centrifuger. Antalet centrifuger som används samtidigt beror av hur mycket fyllmassa det finns i maischarna. Målet är ett kontinuerligt utflöde av socker och sirap från centrifugsteget. B-koket har en satsvis och en kontinuerlig centrifug. Kontinuerliga centrifuger används i C-koket, tre stycken, och i affineringen, två stycken.

2.4 Processtyrning

De tre huvudmålen för processtyrningen är att garantera kvaliteten hos produkten socker, att optimera det ekonomiska utbytet av socker samt att få en kontinuerlig produktion.

Kvaliteten på socker brukar uttryckas i vithet, det vill säga sockret skall vara så fritt från föroreningar som möjligt. Det ekonomiska kravet är att kristallisera ut så mycket sockret som möjligt men samtidigt minimera förbrukningen av ånga och koktider. Ett produktionstekniskt krav är önskan att köra processen kontinuerligt utan svängningar i volymsflöde och sockerkvalitet. Det innebär att sockerhuset skall klara av att ta hand om all tjocksaft från indunstningen.

Det tredje huvudmålet är alltså att garantera kontinuerlig produktion. Varje koksteg har en stor safttank som buffert. En kokare startas när tillräckligt mycket saft nått till safttanken för ett kok. Det är alltså av fundamentalt intresse att inte låta dessa tankar rinna över. Varje kokare körs enligt sitt kokprogram. Beroende på saftens kvalitet tar koken olika lång tid och innehåller olika mängd föroreningar.

Styrning av kvalitet och ekonomi görs i första hand med hjälp av centrifugernas däckningsflöde samt rännskiftet hos A-centrifugerna. Principiellt kan man säga att ökade däckflöden förbättrar kvaliteten men ökar både sockerförluster och ångkonsumtionen på grund av större recirkulationsflöden och vattentillsats. Rännskiftet används för att recirkulera en del av A-sirapen direkt tillbaka till A-koket, det vill säga andelen A-vit-sirap. Om andel A-vit-sirap ökar så försämras kvaliteten men mindre mängd sirap går till B- och C-koken vilket sparar energi och ger bättre utsockring av melassflödet.

3. Sockerhussimulatorn

Den andra delen av projektet har inriktas på en prototyp av en simulator. Avsikten har varit att studera hur state-of-the-art inom datorstödd ingenjörskonst kan utnyttjas för online-simulering. Prototypen kallas SOS, SOckerhusSimulator och är implementerad i G2, ett expertsystem för realtidsapplikationer, från Gensym Corporation (Gensym, 1990).

I detta kapitel beskrivs användandet av sockerhussimulatorn. Innan detta görs en mycket kort presentation av G2. I nästa sektion beskrivs simulatorn samt dess användning översiktligt. Därefter diskuteras de tankar som har styrt utformningen av principer för presentation och interaktion med simulatorn.

3.1 Introduktion till G2

G2 är ett expertsystemskal för realtidsapplikationer med simuleringskapacitet. G2 baseras på objektorienterad representation och utnyttjar grafik för presentation av datastrukturer. G2 innehåller avancerade editorer för definiering av nya objekt och dess ikoner. För interaktion med ett objekt utnyttjas direkt manipulering. Man klickar på ett objekt och får upp en meny med aktioner som kan utföras. Kopplingar och relationer mellan objekten görs genom att dra grafiska förbindelser från ett objekt till det andra.

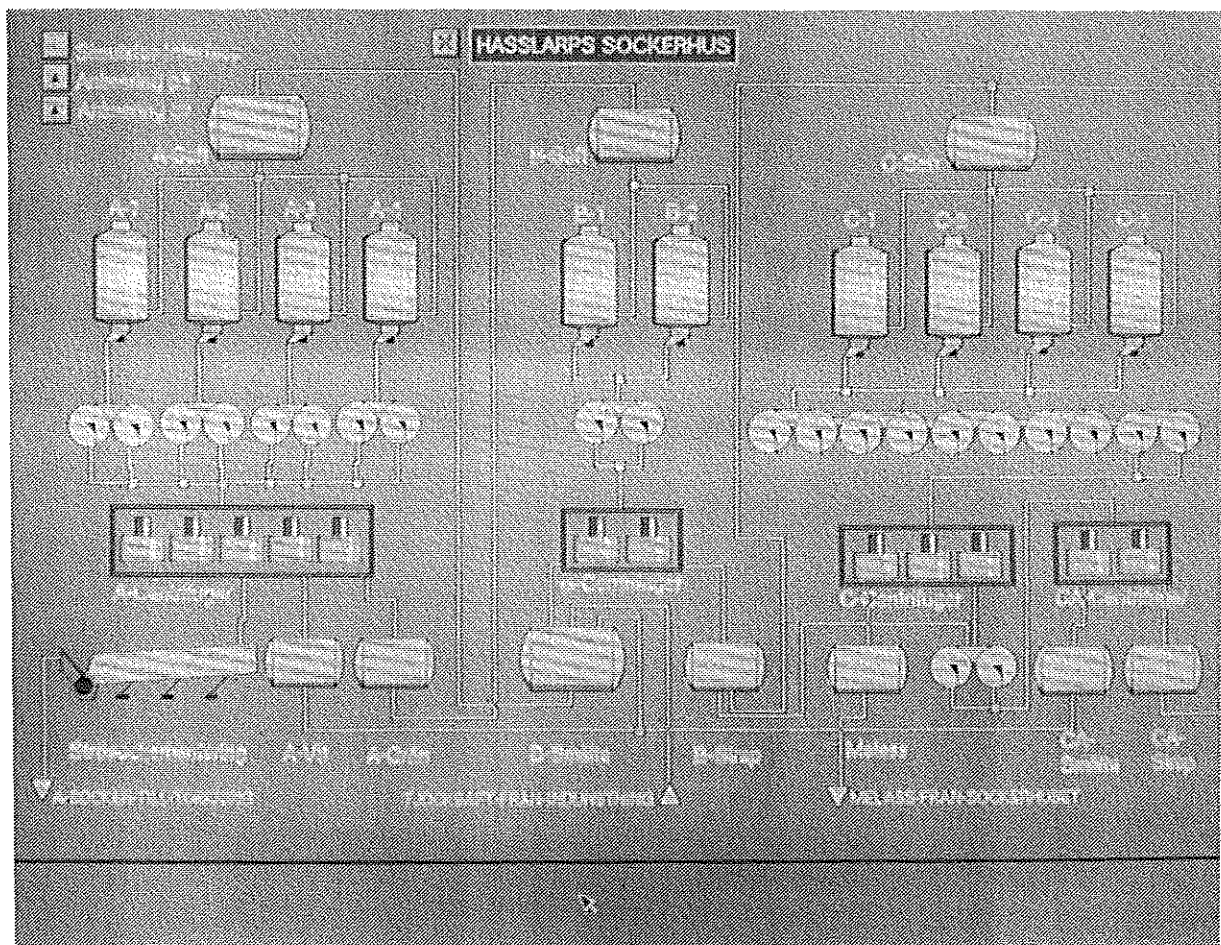
Allting i G2 representeras som objekt med enkel ärvning. I G2 finns färdiga objektklasser för interaktion och presentation, såsom aktionsknappar (action-buttons), skriv-in-rutor (type-in-boxes), skriv-ut-rutor (read-outs) och trendkurvor (graphs). För modellering finns klasser för ekvationer, regler och procedurer. Dessa kan styras med en inferensmaskin med realtidskapacitet eller en realtidssimulator. För att göra egna objekt kan man specialisera gamla objekt så att de passar ens tillämpning. På detta sätt kan man bygga dina modellobjekt och presentationsobjekt. G2 skiljer inte på presentation och representation. Tillsammans med modellobjekt kan man ha grafobjekt. Presentation av modeller och simuleringar kan därför göras på ett naturligt och enkelt sätt.

Det utvecklas ständigt nya verktyglådor till G2. I projektet har en G2-verktyglåda för Grafcet utnyttjats (Årzén, 1991). Här finns färdiga primitiver för byggandet av sekvenser, till exempel styrsystem, med hjälp av steg och övergångar.

3.2 Simulators användning

Simulatorn kan användas för olika typer av arbeten och av olika persontyper. Den primära användaren är i vårt fall processoperatören, som kan använda simulatorn för i) utbildning och träning inför det dagliga arbetet, ii) beslutstöd vid nominell drift, iii) stöd för viss produktionssplanering, samt iv) utveckling av operatörens kompetens vid abnorma driftstillstånd

När simulatorn är klar för användning visas grundbilden hos simulatorn,



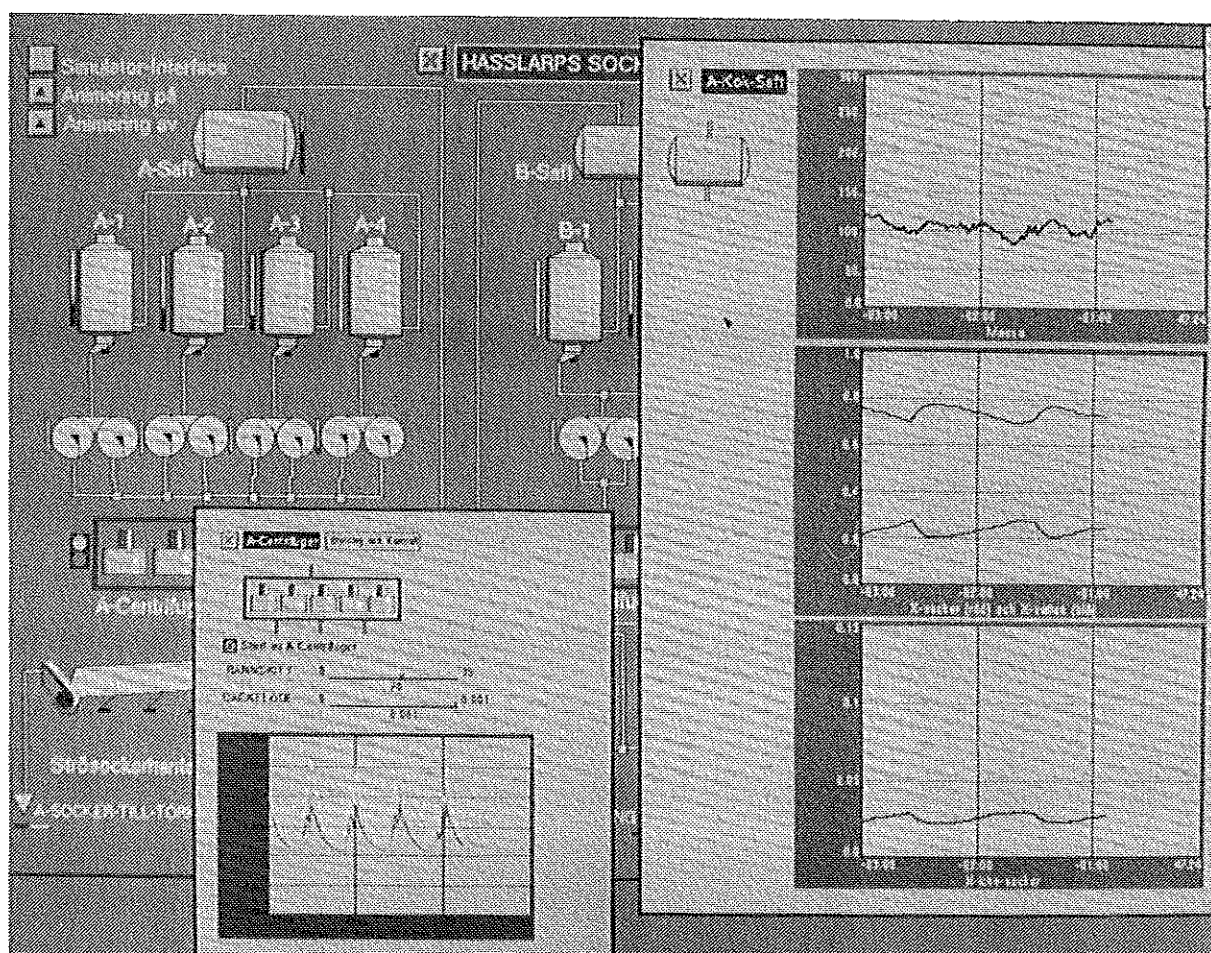
Figur 3.1 Processchema i SOS. Översiktsnivån.

vilket är ett processschema som beskriver processteget översiktligt. Processschemat kan ses i figur 3.1. Denna presentation ger ett helhetsintryck av processen men visar långt ifrån all information som finns tillgänglig.

Nedan diskuteras först användningen av simulatoren samt några olika användningsscenarior. Presentationsformer och interaktionsmöjligheter redovisas i kommande sektioner.

Utbildning och träning

Eftersom produktionen av socker görs kampanjvis finns det alltid behov av uppfräschning och träning av erfarna operatörer samt utbildning av nya operatörer. Kraven på simulatoren är här att beskriva processens nominella beteende på ett bra sätt. Operatören skall kunna studera olika typer av information så att hon/han kan skapa sig en så bra bild av processen som möjligt. Operatören utnyttjar samma styrvariabler under



Figur 3.2 Scenario 1: Ökad processförståelse med hjälp av simulering.

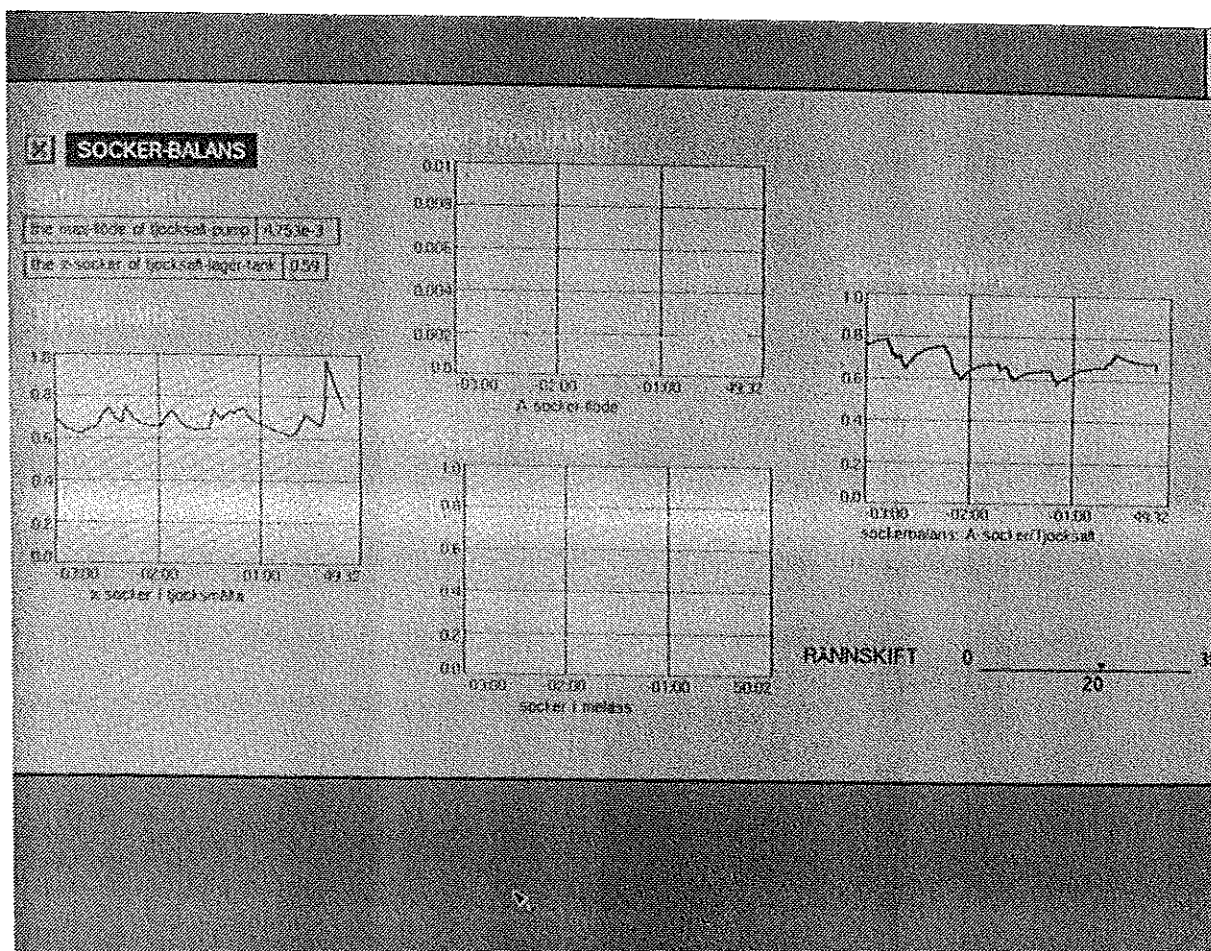
arbetet med simulatören som kommer att vara aktuella vid själva processstyrningen. Tidsskalan kan variera från verklig tid till 10-100 gånger snabbare än verkligheten.

Scenario 1 kan vara att studera processens beteende vid nominell drift. Detta görs i en tidsskala som är måttligt snabbare än verkligheten, 10 gånger snabbare. I denna tidsskala kan operatören manuellt starta alla kokare och centrifuger samt ställa in lämpliga värden på rännskift och däckflöde. Operatören skall alltså finna nödvändig information för att få förståelse för samt göra rätt bedömning om lämpligt styringrepp. Till en början kan denna simulering koncentreras på ett koksteg, till exempel på A-koket. I figur 3.2 ses ett exempel på hur simulatören används för studie av A-kokets beteende. Ett modifierat scenario för träning av erfarna operatörer kan vara att simulera hela processen (100 ggr snabbare) och studera flöden och kvalitet som funktion av vissa styrvariabler (rännskift, däckflöde etc.). Här måste dock vissa manuella styringrepp automatiseras för att den korta tidsskalan ska kunna åstadkommas.

Beslutsstöd

Att utnyttja simulatorer on-line vid processen som hjälpmedel vid beslutsfattande är en attraktiv tanke. Sockerhuset har en stor uppehållstid och det tar lång tid innan styråtgärder syns. Detta gör det tidsmässigt möjligt att studera simuleringar innan olika beslut verkställs. Tränings-simulatören, som diskuterades ovan, simulerar processens nominella drift, kan enkelt utnyttjas även för detta ändamål. Här bör dock viss information och vissa styrvariabler var lättillgängliga för att snabbt och enkelt kunna användas för beslut. Tidsskala vid simulering för beslutsstöd bör vara kort (flera hundra gånger snabbare än verkligheten).

Scenario 2 liknar därför scenario 1. Viss information komprimeras för att ge en bra överblick som förenklar beslutsfattning. Tillsammans med den komprimerade informationen finns de primära styrvariablerna. En sådan här sammanställning med information för ett givet syfte kan vi kalla en vy, se nästa sektion. Man vill också kunna stoppa simuleringen och starta om. I figur 3.3 kan vi se en sådan skräddarsydd vy för en viss typ av beslutsstöd vid styrning av sockerbalansen i processen. Här visas information om kvantitet och kvalitet samt styrvariabeln rännskift.



Figur 3.3 Scenario 2: Beslutsstöd för styrning med rännskift.

Produktionsplanering

Operatörernas yrkesroll kan också breddas genom att de blir delaktiga vid produktionsplanering, processanalys och utveckling av nya styrstrategier. En simulator är ett bra verktyg som gör detta möjligt.

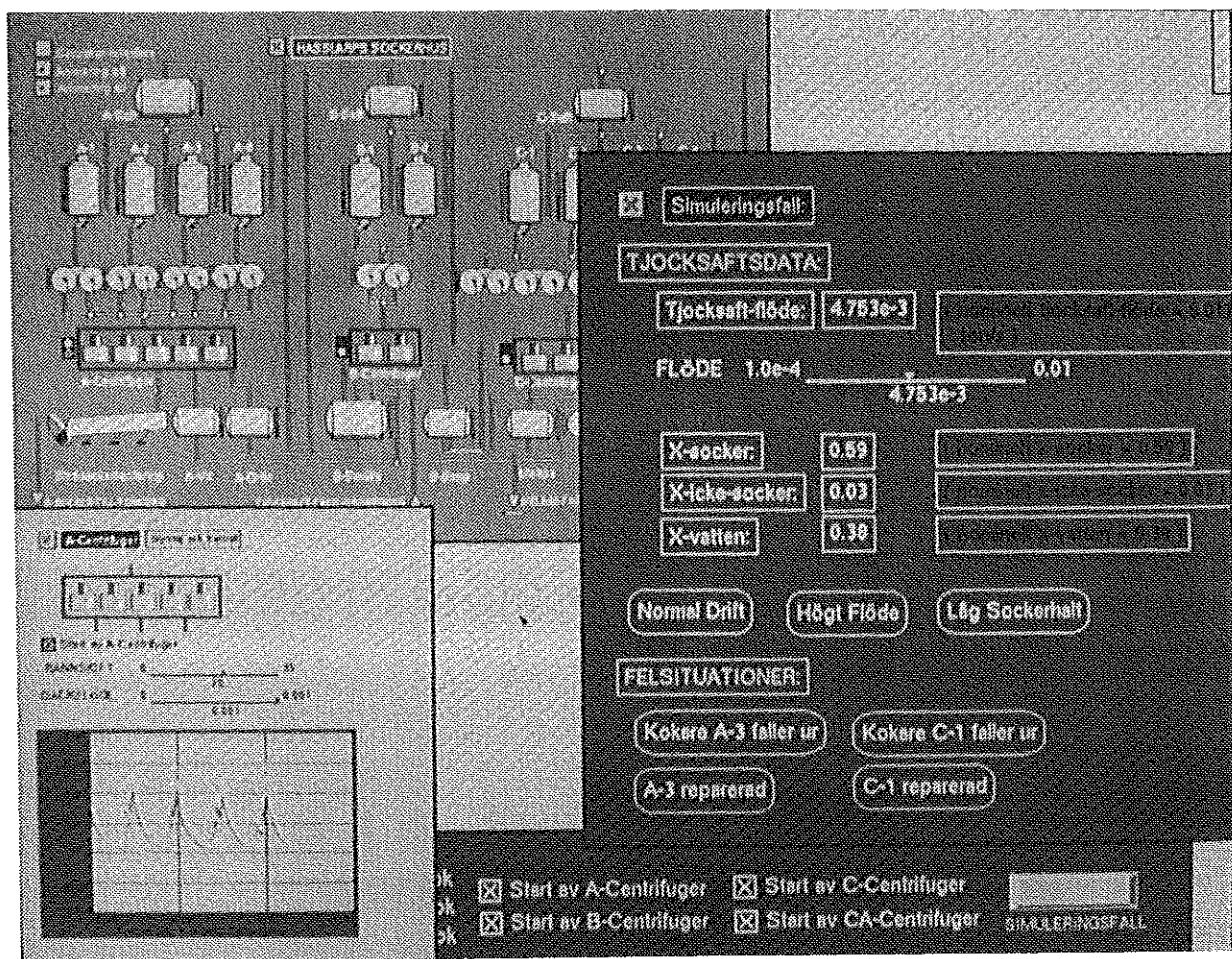
Scenario 3: Under drift måste vissa produktionsförändringar ske planerat. Hur dessa förändringar skall gå till kan tas fram med hjälp av simulering. I sockerhuset skall de kontinuerliga centrifugerna i B- och C-koken spolas rena med jämna mellanrum. Detta innebär stora produktionsförändringar som kan simuleras i simulatoren.

Kompetensutveckling

En fjärde form av simulatortillämpning är studier av abnorma driftstillstånd och felsituationer. Här kan operatörer få studera större fel i processen och dess inverkan på produktionen. Operatörerna själva kan ta fram

dessa styrstrategier eller bli tränade att hantera "katastrofsituationer". Detta innebär att det skall gå att generera olika fel i processen. En instruktör skall obemärkt kunna gå in i simulatören och lägga in fel för att maximera träningseffekten.

Scenario 4 för hur simulatören kan användas är studier om vad som händer om till exempel en kokare faller ur produktion och dess inverkan på produktionskapaciteten. Utveckling och inläring av rätt styrstrategier görs genom att simulera processen med kända fel. Simuleringen skall kunna stoppas och göras om med nya styringrepp. En träningsversion av detta scenario kan sedan utvecklas till ett datorspel där man får poäng för snabba och korrekta åtgärder. Det som är positivt med datorspel är den direkta återkopplingen av styringreppens resultat, både kvantitativt och kvalitativt, men nackdelen kan vara att det ger en oseriös inställning till problemen.



Figur 3.4 Scenario 4: Studie av felsituationer.

3.3 Presentationsformer

För att göra dessa olika tillämpningar möjliga så kommer vi att diskutera filosofin bakom användargränssnittet till simulatorn. Det finns alltså mycket mer information i simulatorn än vad som kan presenteras på en skärm. I simulatorn har vi testat flera sätt för informationssökning och presentation.

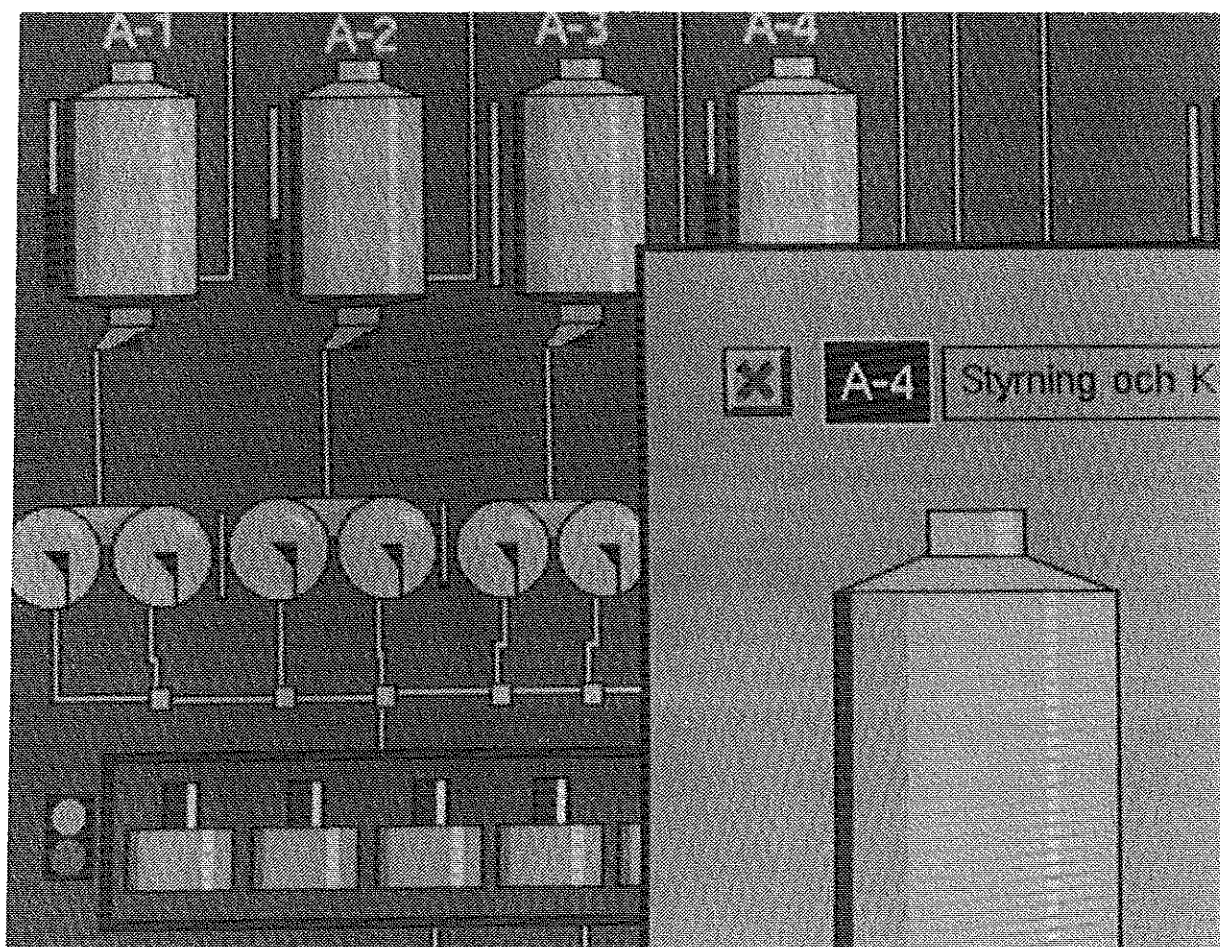
- Ikoner och färger. Färger har valts så att det känns neutralt och lugnt. Ikoner skall vara så naturtrogna som möjligt.
- Informationszoomning. Mer detaljerad information finner man "under" det objektet man vill studera. Detta bildar en informationshierarki.
- Animering. För att göra information mer begriplig animeras viss information såsom nivåer och flöden.
- Multipla vyer. Information av en viss typ samlas i en vy eller "bild".

Nedan kommer varje punkt att diskuteras mer ingående.

Ikoner och färger

Avsikten har varit att ge presentationsobjekten ett så naturligt utseende som möjligt. Detta mynnade ut i något som jag kallar "slutna ikoner". Det vill säga ikoner som påminner om tredimensionella objekt och som inte är genomskinliga eller enbart konturer. Den tredimensionella känslan har åstadkommits med hjälp av ljusblänk i ikonens ena sida och skuggning i den andra. Eftersom nästan alla objekt i verkligheten är metalliska tankar så har de flesta ikoner och kopplingar fått grå färg, ljusblänk är vita och skuggning mörkgrå och dimgrå i två steg. Det har varit förvånansvärt enkelt att göra med hjälp av G2:s ikoneditor. Notera dock att ikonerna i simulatorn är ganska enkla (mest tankar), se figur 3.1 och figur 3.5.

Färger har valts så att de känns neutrala och lugna tillsammans. Uppseende väckande färger skall bara användas till sådant som är extraordinärt (alarm, start/stopp förlopp, etc.). De flesta ikonerna och kopplingarna är grå. Enligt mitt förmenande är den bästa bakgrundsfärgen till grå ikoner kadettblå. Det förstärker också den tredimensionella känslan hos ikonerna. Om man byter bild bytes också bakgrundsfärgen. Vid informationszoomning är bakgrundsfärgen ljusgrå och blir mörkare för varje ny nivå man går ner i. Samma sak gäller vid byte till ny vy. Man kanske skall indikera byte av vy med någonting annat än färg, såsom symboler



Figur 3.5 En närbild av simulatören.

eller texttrutor.

Informationszoomning

På den översta nivån, den så kallade *översiktsnivån*, presenteras den principiella strukturen hos processen. Denna ses i figur 1. För att finna mer detaljerad information så måste man zooma in ett objekt.

Det görs genom att klicka på det och då kommer det upp en nytt fönster med intern information. Denna andra nivå kallas *presentationsnivån*. Här återfinns trendkurvor och interaktionsobjekt (knappar etc.).

På nästa nivå, *modellobjektnivån*, återfinns modellobjekt såsom tankar, pumpar, kokare och styrsystem. Här finns också attributtabell med aktuella värden på alla variabler och parametrar hos objekten. Denna informationshierarki åskådliggörs tydligt i figur 5.1.

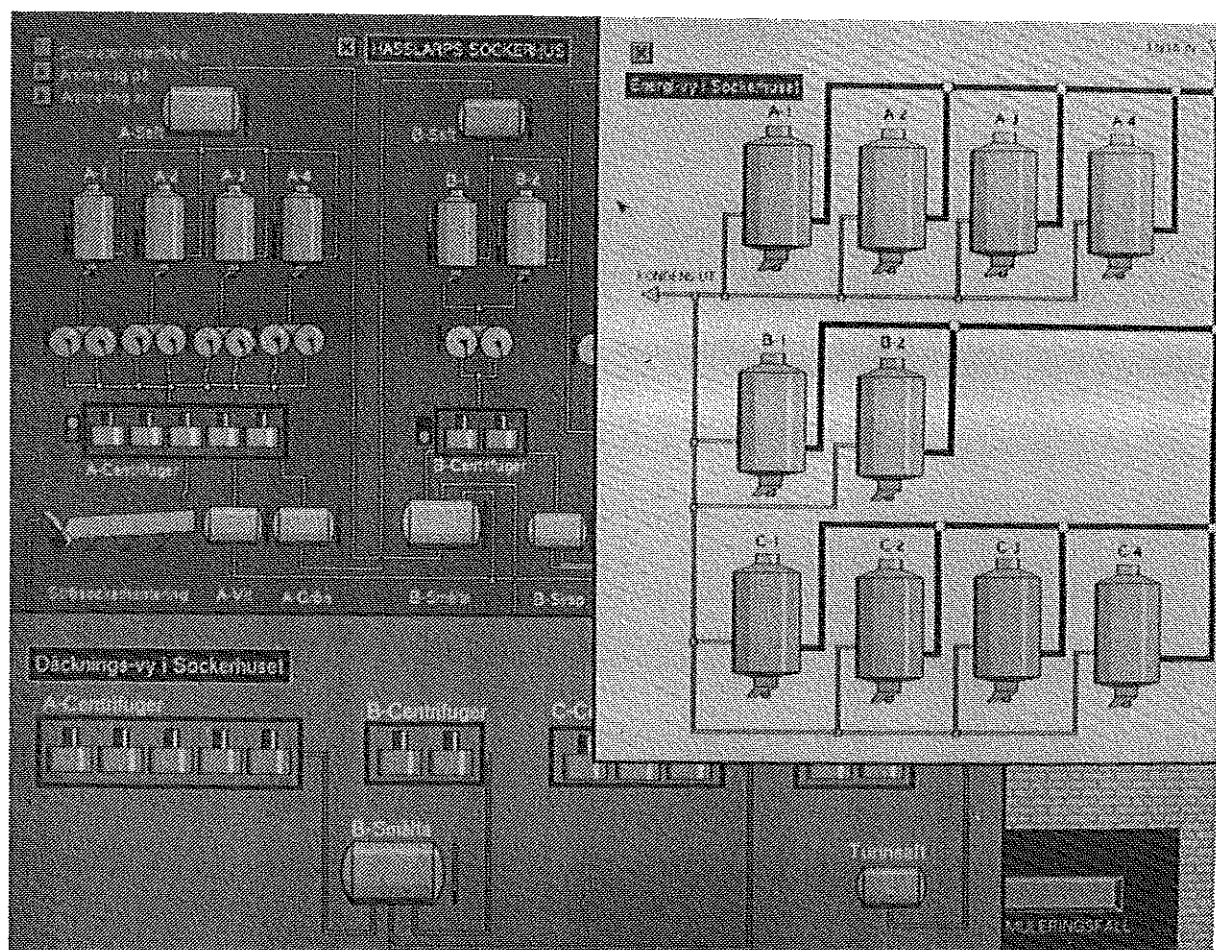
Lägre nivåer innehåller själva modellbeskrivningarna. Detta görs i två

nivåer, *modellnivå* och *ekvationsnivå*. Detta diskuteras mer utförligt i kapitel 4.

Multipla vyer

All information kan inte presenteras på en enda skärm utan mer detaljer finner man genom att zooma in på det man vill studera. Men även översiktlig information vill vi dela upp och inte blanda hur som helst. En översiktlig processbeskrivning kan vi kalla en vy. Den översiktsbild som ses i figur 3.1 kan vi kalla en topologisk produktvy, på grund av att den är koncentrerad på produktens väg genom processen. Ett annat exempel på en vy är en topologisk energivy som beskriver processens behov av energi. Även sammanställningar av information för att presentera processens status görs i en vy.

Multipla vyer av en process är likt hypertextteknik där man kan hoppa mellan olika bilder och presentationer. I simulatören finns vyer för doku-



Figur 3.6 Presentation med multipla vyer.

mentation och alternativa processpresentationer med bland annat översiktsbilder på hela sockerbruket och på sockerhuset, se figur 3.6.

Animering

På den översta nivån kan en enkel animering göras. Denna animering visar nivån i tankar och kärler samt om centrifuger går eller inte. Nivåanimering görs med smala stigrör, se figur 3.1. Ett alternativ är det vanligare sättet med färgläggning av tankens innandöme, det vill säga genomskinligt ytterhölje med väskan indikerad med färg. Denna typ av animering blir mycket iögonfallande. Stora tankar med animering kommer då att dominera bildskärmen. Här har meningen varit att minska animeringens betydelse. Den har därför avsiktligt gjorts så diskret som möjligt. Animeringen i sig skall inte ligga till grund för beslut utan man skall studera de bakomliggande variablerna. Animeringen och dess objekt kan plockas bort och tas fram med hjälp av knappar.

3.4 Interaktionsformer

En operatör måste utföra en mängd olika sturingrepp för att köra processen. Hur skall operatören interagera med simulatoren? Vissa handgrepp kan direkt översättas till simulatoren, medan andra är svårare. Exempel på sturingrepp är följande.

- Start och stopp av styrsystem för koksteg, enskilda kokare och centrifuger.
- Viktiga styrvariabler kan ställas in med hjälp av "glidare".
- Tidsskala i simulatoren kan skrivas in i en "skriv-in-ruta".
- Inställning och ändring av initialtillstånd, inflödesdata under simulering samt generering av felsituationer skall kunna göras osynligt av en instruktör.

I G2 finns färdiga knappar som genererar aktioner då man trycker på den. Flera knappar kan göra samma sak. Till exempel finns knappar för start och stopp av kokare och hela koksteg både lokalt i modellhierarkin (presentationsnivån) och centralt på simulator-interface. Motsvarande finns för andra styrsignaler till exempel rännskiftsinställning. Här finns en inställare lokalt i A-centrifugerna men också centralt på vyn för nominell styrning av processens interna flöden. I simulatoren använd mest förde-

finierade interaktionsobjekt med det är fullt möjligt att definera användarspecifika objekt.

Till G2 finns ett kompletterande program som heter Telewindows. Med detta program kan flera användare koppla upp sig mot samma G2. I G2 kan man också ha olika användaridentiteter vilket gör att man kan ha olika befogenheter för operatörer, instruktörer och utvecklare. Detta gör att en instruktör kan gå in via Telewindows och skapa ett fel som görs på en speciell instruktörsvy vilken operatören (eleven) inte kan komma åt. På denna vy kan instruktören generera individuella förändringar men också starta hela förlopp som spelas upp för operatören.

4. Modellrepresentation

Den första delen av projektet utnyttjades för att studera objektorienterad modelleringsmetodik. Framförallt studerades den teknik som utvecklades inom CACE-ramprogrammet. Här utvecklades ett nytt modellerings-språk, Omola, för objektorienterad modellering (Andersson, 1990). En datormiljö som kan simulera Omola-modeller är under implementering. Det innebär att detta projekt ej har kunnat göra en prototyp-simulator med denna teknik. Den har istället gjorts i en annan mjukvara, G2, som är en god ersättare i denna tillämpning. Nedan följer en kort beskrivning av denna metodik. Därefter presenteras strukturen hos G2-modellen.

4.1 Strukturerad modellering

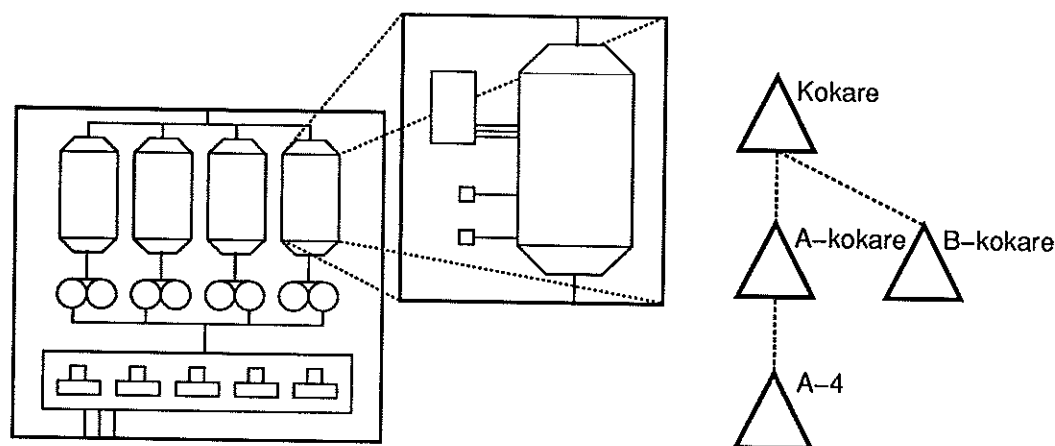
I CACE-projektet utvecklades koncept för strukturerad processmodellering based på objektorienterad metodik (Nilsson, 1989). I det arbetet studerades uteslutande kontinuerliga processer. Nedan följer några definitioner och begrepp inom strukturerad modellering.

Modellstrukturer

En modell består av ett antal modellkomponenter. Det finns tre huvudtyper av modellkomponenter: terminaler, parameterar och beteenden. Varje modell är hårt inkapslad och terminaler beskriver hur modellen kommunicerar med omgivande modeller. En modells parameter används av modellanvändaren för att adaptera modellen till en speciell tillämpning. Den tredje typen av modellkomponent är modellens beteendebeskrivning. En beteendekomponent kan vara av två olika typer: primitiv eller strukturerad. En primitiv beteendebeskrivning är en matematisk modell, till exempel ett system av differentialekvationer. I en strukturerad beteendebeskrivning är beteendet distribuerat i ett antal submodeller som är hopkopplade. Det sista innebär att en modell kan bestå av andra delmodeller som i sin tur kan bestå av delmodeller etc. Vi får något som kallas hierarkisk modelldekomponering.

Ärvning

En modell med modellkomponenter kan beskrivas som en klass med attribut (komponenterna). Denna klass är definierad i en klasshierarki. En klass ärver alla attribut (komponenter) hos sin superklass. Genom att definiera en modell som en subclass av en annan modellklass ärver den nya modellklassen superklassens attribut, det vill säga dess modellkomponenter. Den nya modellen kan specialiseras genom att man lägger till nya attribut eller om man definierar om ett gammalt. Man kan alltså återanvända gamla modeller och förändra dem lite för att passa en ny tillämpning. Ärvning och klasser är idéer från objektorienterad programmering.



Figur 4.1 Exempel på modell- och klasshierarkier.

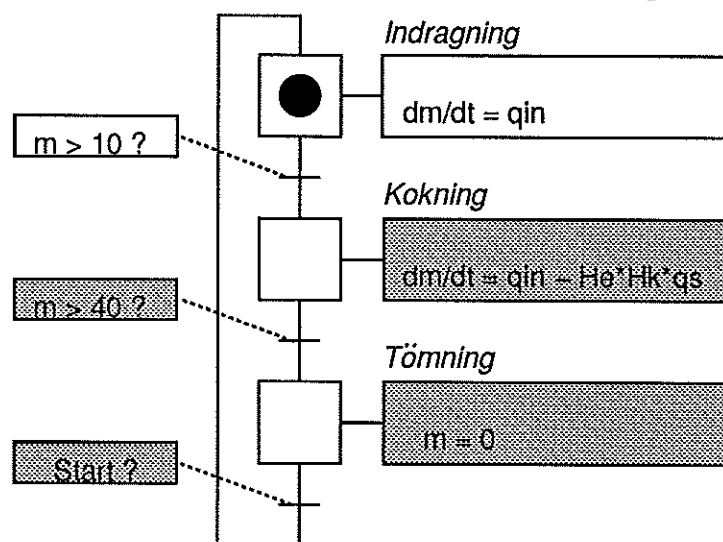
Satsvisa processer

Omola kan idag beskriva kontinuerliga processer och kontinuerliga regler-system med hjälp av de koncept som diskuteras ovan. Det innebär att de så kallade primitiva realiseringarna, det vill säga matematiska modeller av beteendet, måste vara av formen differential-algebraiskt ekvationssystem. Styrsystem och satsvisa processer kan eller bör ej beskrivas så. Därför diskuteras en utvidgning av Omola med primitiver för beskrivning av diskreta händelser.

Karaktäristiskt beteende hos satsvisa processer är de drastiska förändringar som processen gör när den går över i en ny fas. Det går att beskriva alla faser i en och samma beteendebeskrivning, men den blir otympligt stor och obegriplig för en utomstående. Troligen kommer den också att ställa till numeriska problem. Därför vore det önskvärt att kunna separera olika fasbeskrivningar för att öka begriplighet och återanvändbarhet. Fasbeskrivningarna kan å andra sidan inte innehålla villkoren för fasövergång utan den bör lämpligast ske i en separat beskrivning. Denna su-

perbeskrivning styr vilken fasbeskrivning som skall vara aktiv. En sådan superbeskrivning kan göras med hjälp av Grafcet. Grafcet-beskrivningar i Omola med diskreta händelser är under utveckling.

I figur 4.2 ses en sådan här beskrivning. Fasbeskrivningarna till höger är aktiverade och deaktiverade av en Grafcet-sekvens. I figuren är indragning den fas som är aktiv och de andra är inaktiva. När massan m blir större än 10 så byter sekvensen tillstånd och kokningsfasen blir aktiv.



Figur 4.2 Modellstrukturen i satsvisa processer.

Modellegenskaper

Strukturerad modellering med objektorientering ger goda möjligheter för återanvändning, utveckling, förfining och underhåll av modeller.

Genom att modeller är klasser i klasshierarki med ärvning kan man återanvända modeller. Om modellen är lämpligt parameteriserad ökar återanvändbarheten. Genom specialisering kan samma kokaremodell användas i olika koksteg och i olika processsimulatorer. Samma kokaremodell kan användas i en simulator för Hasslarp och i en för Örtofta. Inkapsling gör det också möjligt att göra lokala förändringar utan att det berör helheten. Man kan till exempel gå in och modifiera en ekvation i A-kokarmodellklassen som sedan ärvt ner till varje individuell A-kokarklass vilken i sin tur ärvt ner till själva kokarmodellerna i simulatören. Dessa egenskaper underlättar utveckling och återanvändning.

Egenskaperna underlättar också möjligheterna att göra förfining av modeller och att underhålla modellerna, eftersom detta arbete kan göras lokalt i separata modellklasser. Detta är mycket viktigt eftersom underhåll och förfining av modeller i en framtid måste göras av personal som

arbetar vid processen. Om simulatorer skall ha ett stort inflytande på produktionssättet måste modellens fortsatta användning efter utvecklingskedet vara i händerna på de som använder den.

4.2 Processmodell i G2

G2 innehåller inte alla de egenskaper som diskuterades ovan. G2 har objektorienterad modellrepresentation samt goda möjligheter att bygga grafiska beskrivningar av modellerna. Olika typer av utvecklingsstöd, såsom editorer, är väl utvecklade och gör det enkelt och snabbt att bygga prototyper. Det sätt på vilket G2 används i detta projekt är inte vad det egentligen är avsett för. Emellertid visar detta programmets styrka och flexibilitet.

G2 är därför ett bra redskap vid idéutveckling och för att bygga prototyper, men som mjukvara är det stort, dyrt och kräver en kraftfull arbetsstation.

Modellstruktur i G2

I kapitel 3 introducerades konceptet informationszoomning. Då beskrevs den hierarkiska informationsstrukturen med översiktsnivå, objektpresentationnivå, modellobjektnivå samt modell- och ekvationsnivå. De två översta nivåerna utgör enbart simulatorns presentation och interaktion. I detta sammanhang koncentrerar vi oss på de tre lägre nivåerna. Det finns alltså en klar gräns mellan användargränssnittet och själva modellbeskrivningen.

Den hierarkiska modellstrukturen åstadkoms i G2 genom att associera ett fönster, i G2 kallat subworkspace, till ett objekt. På detta fönster kan nu nya objekt och ny information lagras. På modellobjektnivån i hierarkin återfinns själva modellobjekten. Modellobjekten är de moduler som innehåller de simulerade egenskapernas variabler och parametrar.

Ett exempel är kokarhierarkin, där alla kokare finns representerade med presentationsobjekt på översiktsnivån. Genom att klicka på en speciell kokare kommer dess fönster för objektpresentation upp, innehållande trendkurvor och styrvariabler. Genom att klicka på kokaren på detta subworkspace kommer modellobjektnivån upp med själva kokarmodellobjektet, dess styrsystemobjekt och en attributtabel.

I vissa fall finns ytterligare två nivåer där själva modellbeskrivningarna finns. På de subworkspace som tillhör modellobjekten återfinns Grafcet-modeller av styrsystem och den satsvisa kokaren. Styrsystemet avgör i vilken fas kokaren skall vara, medan kokarens Grafcet byter aktiv fasbeskrivning beroende på styrsystemet. Detta är möjligt så som Grafcet är implementerat i G2. När ett steg i en Grafcetsekvens är aktivt är dess subworkspace också aktivt, och om ett steg blir inaktivt deaktiveras också dess subworkspace. Genom att lägga varje enskild fasbeskrivning på separata subworkspace så kommer Grafcet automatiskt att styra bytet av fasbeskrivning.

Detta låter enkelt, men ett problem är dock att varken objekt för differentialekvationer eller för differensekvationer kan deaktiveras (i dagens G2). Detta problem har lösts genom att implementera differensekvationer med hjälp av samplade regler som kan deaktiveras. Detta blir vid simulering exakt samma sak som differensekvationer. I dessa samplade regler görs alla beräkningar som behövs för att uppdatera tillstånd samt beräkna andra variabler. Differentialekvationer Euler-approximeras, det vill säga derivator beskrivs med framåtapproximation.

$$\frac{dm}{dt} = q \Rightarrow m_{t+h} = m_t + q_t h$$

Alla regler samplas med tidsintervallet 1 sekund, vilket är det kortaste intervallet i G2. Detta kan orsaka stabilitetsproblem men i denna tillämpning har inga sådana problem dykt upp. För att variera tidsskalan i simuleringen finns en global variable, TimeScale, som är tidssteget i framåtapproximationen ($h = \text{TimeScale}$). Genom att ställa tidsskalan kan simuleringen göras lika många gånger snabbare än verkligheten som denna faktor indikerar. Upplösningen i tiden kommer naturligtvis att försämrats i motsvarande grad. Stabiliteten hos en modell som Euler-approximeras är också begänsande för hur stor faktorn kan vara.

Klasshierarki och ärvning i G2

G2 har objektorientering med enkel ärvning. En klass ärver alla attribut från dess superklass, så som det diskuterades ovan. Detta är inbyggt i G2 och det finns några grundläggande objektklasser för trendkurvor, knappar etc. I simulatoren har alla presentationsobjekt och modellobjekt sina klassdefinitioner i ett klassbibliotek. I simulatoren har detta bibliotek delats upp i bokhyllor innehållande klasser för olika objekttyper. Man

kan också lyfta in färdiga klassbibliotek eller verktyglådor. I detta fall har jag utnyttjat en färdig verktyglåda med Grafcetkomponenter.

Sammanfattning

G2 ger goda möjligheter för konstruktion av interaktiva program med bra användargränssnitt. Modellrepresentation i G2 baseras på klassdefinitioner med enkel ärvning. En klar begränsning i denna tillämpning har varit att en klass ej kan vara sammansatt av andra objekt. Ovan diskuterades hur informationshierarkin var implementerad via så kallade subworkspaces. Subworkspaces är endast en egenskap hos instansen och inte hos klassen. Detta innebär att de ej går att ärva.

5. Modellbygge

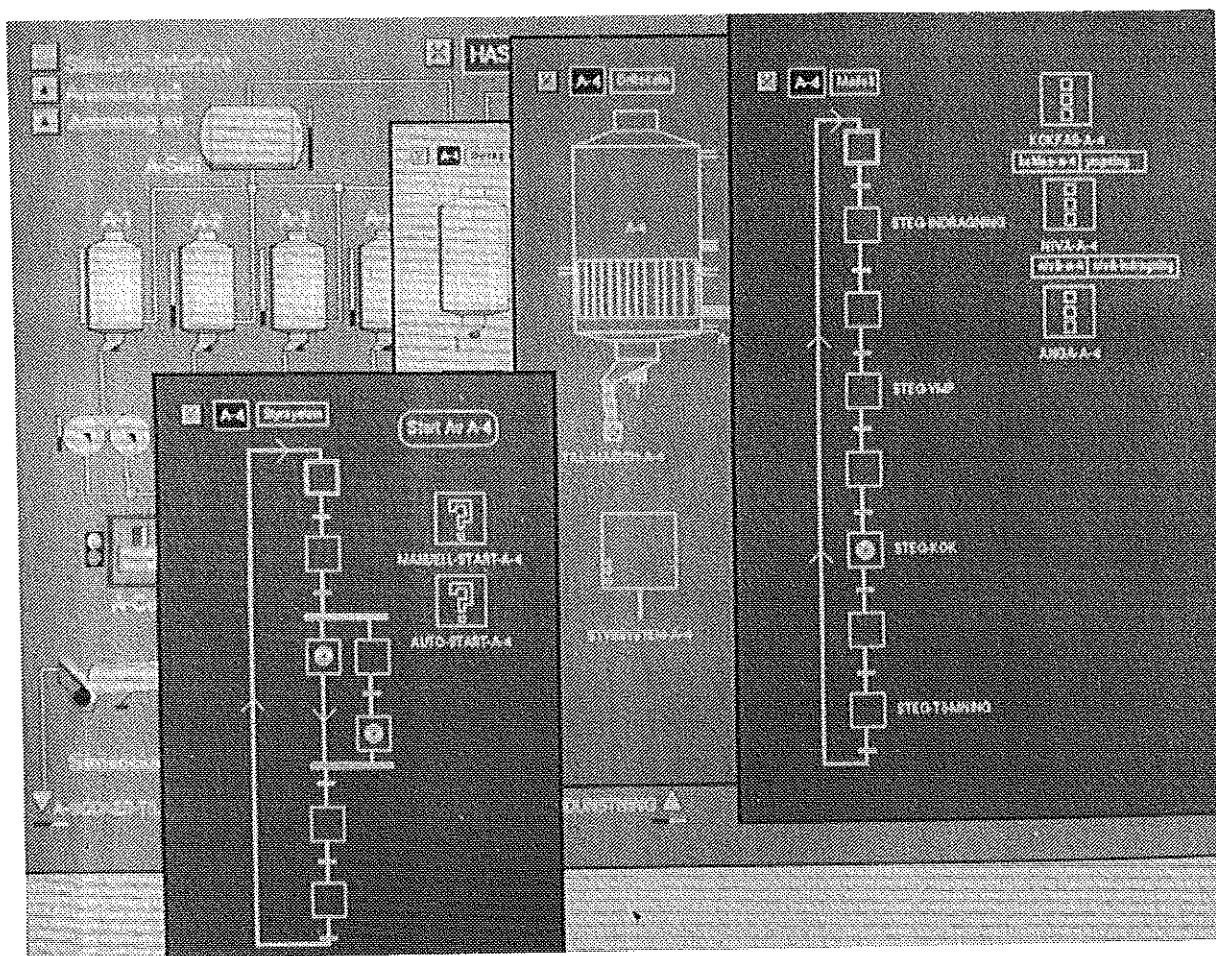
I detta kapitel kommer de modeller som utnyttjas vid simulering av sockerhuset och dess komponenter att presenteras. Simulatoren utgör själv den bästa dokumentationen och G2:s sätt att beskriva beteenden med naturligt språkliknande syntax underlättar detta ytterligare. Detta innebär att delmodellerna presenteras och diskuteras översiktligt för att indikera delmodellernas giltighet och omfattning.

5.1 Beskrivning av kokarna

Beteendet hos en kokare beskrivs med två objekt, en kokarmodell och en styrsystemmodell, vilka återfinns på modellobjektnivån. Styrsystemet är beskrivet med hjälp av Grafset och är en sekvens av olika ingrepp. Ingreppen har förenklats till att avgöra i vilken fas kokaren skall befinna sig. Kokarmodellen är också en sekvens, vilkens tillstånd innehåller beskrivningar av varje fas för sig. Alla faserna har gemensamma tillstånd, massa och massfraktioner. Dessa tillstånd är attribut i kokarmodellobjektet. Komponenterna är löst socker, sockerkristaller, föroreningar samt vatten. Summan av massan av alla komponenterna är lika med den totala kokarmassan. Detta innebär att vi har ett algebraiskt villkor på tillstånden vilket genomgående har gjort att fraktionen vatten beräknas enligt $x_v = 1 - x_s - x_k - x_i$. Det dynamiska beteendet beskrivs nu alltså med fyra massbalanser, en total massbalans och tre komponentbalanser.

Grafset i G2

Med Grafset kan man beskriva sekvenser grafiskt. I simulatoren utnyttjas ett bibliotek med Grafset-objekt för beskrivningar av satsvisa förlopp och styrsystem (Årzén, 1991). Två Grafsetdiagram ses i figur 5.1. Tillstånden, eller stegen, visas med en fyrkant och om steget är aktivt så indikeras det med en markör. Vad som händer när steget är aktivt beskrivs på stegets subworkspace. Beskrivningen kan vara av två typer, initialregler eller samplade regler. Initialregler exekveras bara initialt, det vill säga då



Figur 5.1 En kokaren och dess styrsystem beskrivna med Grafset.

steget blir aktivt, medan samplade regler har ett samplingsintervall som avgör när regeln skall exekveras. Samplingen görs en gång per sekund.

Efter varje steg finns det alltid en övergång som beskriver när det föregående steget blir inaktivt och nästa blir aktivt. Övergångsvillkoret återfinns på övergångens subworkspace. De regler som beskriver övergång är samplade.

Kokarstyrning

Styrsekvensen för en kokare har förenklats jämfört med de verkliga styrsystemen. Antalet faser har bantats till sex stycken; indragning, ympning, kokning, kokhållning, tömning samt vänta-på-ny-sats. De aktioner som sekvensen gör har också förenklats drastiskt genom att endast bestämma vilken fas kokaren befinner sig i. Detta indikeras i en global variabel, kokfas-A-1. I styrsystemet bokas den massa som ett kok behöver i saft-tankens styrsystem. Om det ej finns tillräckligt med saft för ett helt kok

så startar kokaren inte.

Kokarmodell

Kokaren kan delas upp i fyra huvudfaser; indragning, ympning, kokning samt tömning. Alla dessa steg är följt av ett vänta på nästa-fas-steg. Nedan diskuteras de olika stegen mer ingående.

Indragning är den första fasen där en given mängd saft dras in i kokaren. Mängd och tid för indragning bestäms av parametrar i kokarmodellobjektet. I indragningssteget utnyttjas därför initialregler för att ta den givna mängden saft från safttanken till kokaren samt sätta sammansättningarna i kokaren lika med dem i safttanken. Indragning deaktiveras då tiden för indragning har gått ut.

Ympning sker då förkoket har nått ett givet tillstånd vad gäller temperatur och viskositet. För att nå detta tillstånd måste kokmassan förkokas. Under förkoket förångas vatten, koncentrationen av socker ökar, men samtidigt hålls nivån i kokaren konstant. Förkoket beskrivs i ympsteget med dynamiska massbalanser:

$$\begin{aligned}x_{t+1} &= (x_t m_t + x_{t,in} q_{in} T) / m_{t+1} \\ m_{t+1} &= m_t + q_{in} - q_{ut}\end{aligned}$$

x är massfraktioner, m är kokarmassa, q är massflöde in till eller ut från kokaren samt T är tidskalan. Vid förkoket är $q_{in} = q_{ut}$, det vill säga massan är konstant. Detta implementeras med hjälp av en samplad regel. Mängden vattenånga som kondenseras ut beskrivs på enklaste sätt.

$$q_{ut} = \Delta H_{evap} P_{energi}$$

ΔH_{evap} är ångbildningsvärmets medan P är den tillförda energin i form av ånga.

Kokning är det viktigaste steget i modellen och beskrivs med tre samplade regler. Den första beräknar alla blandningseffekter då en given mängd tjocksaft tas in i koket. Detta blir identiskt med förkokbeskrivningen ovan. Nästa regel beskriver kokningen det vill säga vad som händer då en given mängd vatten kokas bort. Den tredje regeln beskriver hur mängden kristaller växer proportionellt mot övermättningen, vilket definieras som skillnaden mellan mängden löst socker och en jämviktskoncentration av löst socker. Mängden förångat vatten beskrivs likadant

som i förkoket medan mängden inkommande saft beräknas. Här har detta flöde beräknats genom att den totala kokmassan dividerats med koktiden. Koktiden varierar alltså inte utan är en parameter. Koket avbryts då tiden för koksteget har gått ut.

De tre reglerna kan slås ihop och bildar då följande system av differensekvationer.

$$\begin{aligned}x_{s,t+1} &= \frac{x_{s,t}m_t + x_{s,in}q_{in}T}{m_t + q_{in}T - q_{ut}T}(1 - K_{krist}) + K_{krist}x_{jmv} \\x_{k,t+1} &= \frac{x_{k,t}m_t}{m_t + q_{in}T - q_{ut}T} + K_{krist}\left(\frac{x_{s,t}m_t + x_{s,in}q_{in}T}{m_t + q_{in}T - q_{ut}T} - x_{jmv}\right) \\x_{i,t+1} &= \frac{x_{i,t}m_t + x_{i,in}q_{in}T}{m_t + q_{in}T - q_{ut}T} \\x_{v,t+1} &= 1 - x_{s,t+1} - x_{i,t+1} - x_{k,t+1} \\m_{t+1} &= m_t + q_{in}T - q_{ut}T\end{aligned}$$

x är de olika massfraktionerna, m den totala kokarmassan, q_{in} inflödet av saft, q_{ut} de förångade vattenflödet samt T är tidsskalan. Själva kristallisationen beskrivs med en förstärkningsfaktor K_{krist} och jämviktskoncentrationen med x_{jmv} .

Tömning Då det finns en tillräckligt tom maisch töms nu koket i maischen. Detta beskrivs med initialregler vilket gör att det endast tar ett sampel att tömma kokaren ner i maischen. Tillstånden sättes helt abrupt till noll.

Kokstartsautomation

Man kan simulera hela processen och styra den med enbart manuella styringrepp. För att hinna med och göra rätt bedömning måste simuleringen gå ganska långsamt. Om man vill öka simuleringshastigheten måste vissa manuella styringrepp automatiseras. Tillsammans med safttanken finns därför ett styrsystem som så fort det finns tillräckligt med saft i safttanken automatiskt startar den kokare som står på tur. Detta styrsystem kan startas och stoppas under simuleringen.

5.2 Modell av maischarna

Maischarna är beskrivna med ett modellobjekt med attribut som beskriver tillstånden, massa och massfraktioner. Den innehåller ingen ytterligare information om sitt beteende utan är en ren integrator. Kokarna kan tömma sitt innehåll i maischarna medan centrifugerna kan ta ur fyllmassa ur dem. Maischarna har också ett attribut som indikerar hur länge koket har kylts i maischen samt en parameter för bestämd kyltid. Efterkommande centrifuger skall inte kunna tömma en maisch som inte har kylt koket tillräckligt länge.

5.3 Centrifugbeskrivningar

Centrifugering görs alltid med flera stycken centrifuger parallellt. Tidskraven vid simuleringarna gör att vi ej kan modellera varje centrifug för sig utan alla centrifuger i en station slås samman och beskrivs gemensamt. Detta har ingen större inverkan på simuleringens kvalitet. I A-koket används periodiska centrifuger och i B- och C-koket används kontinuerliga. Eftersom man studerar hela centrifugstationer och försummar dynamiken, eventuell uppehållstid eller tidsfördröjning i centrifugerna, kan alla stationer ha samma beskrivning. De beskrivs statistiskt med ekvationer som delar upp ett inkommande flöde i två eller tre utgående flöden. Alla centrifugstationer beskrivs överst av ett Grafcet som startar och stoppar centrifugerna.

Centrifugeringsmodellen består av tio regler. Två regler beräknar massflödet in till centrifugerna då centrifugerna fylls med fyllmassa från maischen. Dessa båda regler kompletterar varandra. Den ena beskriver den nominella driften medan den andra beskriver undantaget då maischen helt och hållet töms under detta sampel. Därefter beräknas de statiska komponentmassbalanserna samt utflödet av socker och sirap (vätska). Sockerflödet bestäms genom en effektivitetsfaktor som är ett mått på hur mycket av sockerflödet som består av sockerkristaller. En effektivitetsfaktor på 0.99 indikerar att sockerflödet består av 99% kristaller och 1% vätska. Vi antar också att det inte finns några kristaller i sirapen. Under dessa antaganden kan alla flöden och deras inbördes koncentrationer bestämmas med statiska massbalanser.

Däckflöde utnyttjas under centrifugeringen för att tvätta kristallerna och därmed öka kvaliteten hos sockret. I modellen finns beskrivning av

däckflödet och några av dess effekter. De effekter som är modellerade är utspädningseffekten samt upplösning av kristaller. Utspädningseffekten beskrivs direkt i de statiska massbalanserna. Den sista effekten beskrivs på enklaste sätt genom en upplösningssfaktor med avseende på mängden kristaller. Upplösning på grund av varierande däckflöde har inte modellerats. Däckflödets tvättande och där med kvalitetshöjande effekt har inte heller modellerats.

Flöde av inkommande fyllmassa till centrifugerna är beroende på hur mycket fyllmassa det finns i maischarna. Om den totala fyllmassan i maischarna ökar så ökas centrifugerna kapacitet. I A-koket betyder det att fler periodiska centrifuger går samtidigt. I kontinuerliga centrifuger kan inflödet varieras något och detta gör att konsumtionen av fyllmassa kan styras. Flödet in till centrifugstationen är proportionell mot mängden fyllmassa i maischarna.

Rännskiftet är processens viktigaste styrvariabel. Denna modelleras i två steg. I de statiska massbalanserna finns tre parametrar för beskrivning av rännskifteffekter. Dessa koefficienter bestämmer förhållandet mellan massa, löst socker och föroreningar i A-grönsirap och i A-vitsirap. I en ytterligare regel bestäms massaförhållandet som en funktion av det så kallade rännskiftstiden enligt följande.

$$K_{g/v-massa} = (1 + \frac{T_{cc}}{T_{cc} - T_r}) \frac{T_r}{T_{cc}}$$

Här betyder T_{cc} den totala centrifugeringstiden och T_r rännskiftstiden, det vill säga den styrvariabel som ställs in av operatören. I dagen modell finns det ingen kvalitativ skillnad mellan vit- och grönsirap. Men det är enkelt att komplettera modellen med $K_{g/v-socker}$ och $K_{g/v-icke-socker}$ som funktioner av T_r och q_{df} . Hur dessa funktioner skall se ut är dock en annan sak.

Val av maisch för tömning måste göras i A-koket. I A-koket finns nämligen fyra parallella maischar före centrifugerna. Detta innebär att vi måste tömma en maisch i sänder. Maischarna kan bli fyllda oberoende av varandra vilket gör att när centrifugerna har tömt en maisch måste de söka upp nästa fyllda maisch, vilket kan vara vilken som helst. Detta görs genom att kontrollera om nästa maisch i ordning kan tömmas, om inte så kontrolleras nästa igen och så vidare. Detta tar lite väl många sampel om det är långt emellan maischarna. En annan metod, som inte är testad,

är att skapa en kö där varje maisch kommer i rätt tömningsordning och centrifugerna sedan läser av kön.

5.4 Modell av tankar och pumpar

Utöver de tidigare beskrivna processkomponenterna beskrivs också tankar och pumpar individuellt i simulatoren.

Tankar utnyttjas som buffertkärl i processen och modelleras precis som maischarna. Tankobjekten innehåller alltså attribut som beskriver deras interna tillstånd av massa och koncentrationer. Andra objekt kan fylla på eller ta ur massa ur tankarna.

Pumpar modelleras med ett enkelt Grafcet som startar och stoppar pumpningen. Pumpstyrningen modelleras som ett relä. Om en nivå överstiger eller understiger ett visst referensvärde så startas pumpen, som då går för fullt. Denna modell är mycket grov och en ganska dålig beskrivning av verkligheten.

5.5 Modelldiskussion

Varje enskild komponent är ganska enkel men helheten med ett trettiotal komponenter, med interna styrsystem och fasbeskrivningar, blir automatiskt komplex. I dagens simulator bör i första hand alla processparametrar ställas in så bra som möjligt, innan modellernas giltighet och eventuella förändring diskuteras. Av de befintliga modellerna anser jag att beskrivningar av varierande flöden bör förbättras, såsom styrning av fyllmassaflöde till centrifuger samt pumpstyrning.

Beskrivningen av kristallisationsförloppet bör också modifieras något. Idag beräknas den teoretiska övermättningen som sedan ligger till grund för beräkningen av mängden som kristalliserar. Istället kan man basera beskrivningen på hastighetssamband där kristallisationshastigheten är proportionell mot den drivande kraften, övermättningen, $R_{krist} = K_r(x_{s,t} - x_{s,jmv})$. Faktorn K_r är en funktion av arean som i sin tur varierar med kristallstorleken (Wilson, 1990).

Andra brister är krav som inte har modellerats, till exempel dynamik i energibalanser samt begränsad tillgång på ånga och vakuum. Att ställa upp modeller för tillgången på ånga är dock mycket svårt och kräver insyn även i indunstningssteget.

6. Erfarenheter

Erfarenheterna från projektet är många och de mest intressanta diskuteras nedan.

Användandet av en dynamisk simulator, såsom det är indikerat i rapporten, anser Sockerbolaget som realistiskt och användbart. Denna typ av simulator har stora möjligheter att bli ett bra hjälpmedel för operatören i det dagliga arbetet. Kommentarer har framförts om att simulatorns organisation och användargränssnitt är väl anpassat för operatörer. Simulatorens har inte visats för en större grupp operatörer då beslut om fortsatt utveckling inte har fattats. Något större arbete med modellvalidering mot dynamiskt processdata har inte gjorts. Simuleringar visar ändå att modellen beskriver processens förväntade beteende.

Användargränssnittet eller kontaktytan med simulatorens måste göras enkel men också intressant och spännande. Formerna för presentation och interaktion i simulatorens anses som realistiska och anpassade för operatörer och deras bild av processen. En användare skall känna att hon eller han har full kontroll och överblick över simuleringen. Samtidigt skall det finnas möjlighet att finna ny intressant detaljerad information i simulatorens.

G2 är en mycket användbar och flexibel mjukvara, trots en del brister. Det är ett ypperligt hjälpmedel för snabb prototypkonstruktion och för byggandet av användargränssnitt. De brister som finns gäller i vårt fall hur man representerar strukturerade processobjekt. Sockerbolaget överväger att köpa in G2 men som on-linesimulator i sockerhuset i alla bruk kommer den troligen inte att installeras. Om man inte direkt kan peka på konkreta vinster på grund av on-linesimulatorns användning så är investeringen för stor.

Modellrepresentationen i G2 har en del brister då det gäller representation av strukturerade modeller. En annan kommentar gäller uppdelningen i användargränssnitt och modellbeskrivningar. I sockerhussimulatorens är modellen decentraliserad och utspridd "under" presentationssubjekten. Ett alternativ är att separerar modell och presentation. Man bygger först modellen och sedan konstrueras presentationvyerna som hämtar information från modellen.

7. Referenser

- ANDERSSON, M. (1990): "Omola - An Object-Oriented Language for Model Representation," Licentiate thesis CODEN: LUTFD2/TFRT-3208, Department of Automatic Control, Lund Institute of Technology.
- GENSYM. (1990): *G2 Reference Manual*, Gensym Corporation, Cambridge, Mass.
- LARSSON, H. (1989): *Svenskt socker*, Sockerbolaget, Arlöv.
- NILSSON, B. (1989): "Structured Modelling of Chemical Processes - An Object Oriented Approach," Licentiate thesis CODEN: LUTFD2/TFRT-3203, Department of Automatic Control, Lund Institute of Technology.
- WILSON, D. (1990): "Advanced Control of a Batch Raw Sugar Crystalliser," Ph D Thesis,, Department of Chemical Engineering, University of Queensland.
- ÅRZÉN, K.E. (1991): "Sequential function charts for knowledge-based, real-time applications," *Submitted to 3rd IFAC Workshop on Artificial Intelligence in Real Time Control*.

