



LUND UNIVERSITY

Språk för förverkligande av reglersystem

Elmqvist, Hilding

1980

Document Version:
Förlagets slutgiltiga version

[Link to publication](#)

Citation for published version (APA):
Elmqvist, H. (1980). *Språk för förverkligande av reglersystem*. (Research Reports TFRT-3161). 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

SPRÅK FÖR FÖRVERKLIGANDE AV REGLERSYSTEM

HILDING ELMQVIST

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA
DECEMBER 1980

RAPPORTREFERAT

Blanketten ifylles på svenska

STU programområde		part	ifylles ej
STU-rapport nr	Rapportdatum	Sidantal inkl bilagor	
79-3740	1980-12-30	10	
Anslagsmottagare (namn, institution, adress)		Institutionen för Reglerteknik	
Karl Johan Åström		Lunds Tekniska Högskola Box 725, 220 07 Lund	
Rapportförfattare (namn, institution, adress)		Institutionen för Reglerteknik	
Hilding Elmqvist		Lunds Tekniska Högskola Box 725, 220 07 Lund	
Projekttitel			
Språk för förverkligande av reglersystem			
Rapporttitel			
Språk för förverkligande av reglersystem			
Referat			
<p>Inom denna förstudie har befintliga hjälpmedel för implementering av reglersystem studerats. Det finns en klar trend att decentralisera reglerfunktionerna i större processreglersystem. Därvid används vanligen förprogrammerade mikrodatorbaserade regulatorer och PC-system. Minidatorer används i sådana system i huvudsak för att upprätthålla en databas för anläggningen samt för presentation av processscheman och trendkurvor på färgbildskärmar.</p> <p>Ett växande användningsområde för mikroprocessorer är reglering och styrning av diverse apparater som eventuellt massproduceras. Mikrodatorn ingår därvid som en komponent i apparaten. Vad mikrodatorn skall utföra varierar mycket mellan olika applikationer varför reelltidspråk med god flexibilitet erfordras. Språket Ada har utvecklats just för denna klass av applikationer.</p> <p>Ett stort problem vid beskrivning av process- reglersystem är den dåliga flexibiliteten. Det är t.ex. svårt att införa speciella beräkningar för vissa reglerkretsar. Vidare används helt skilda beskrivningsmetoder för reglering och styrning. Olika leverantörer har dessutom olika beskrivningssätt. Det är därför angeläget att utveckla mera enhetliga och kraftfulla beskrivningssätt som är leverantörs-oberoende. Dessa frågor kommer att studeras i ett nytt projekt: "Utveckling av användarorienterade språk och hjälpmedel för implementering av reglersystem" inom "Centrum för Industriella Datorsystem" som har bildats i samarbete med avd. för Digital- och Datorteknik.</p> <p>Följande pilotprojekt har utförts under förstudien: "Regula - Ett användarvänligt programpaket för implementering av reglersystem", "Ett enkelt reelltidsoperativsystem", "Pascal för programmering av PC-system" samt "Terminal med zoomning".</p>			

STU-rapport 79-3740

Språk för förverkligande av reglersystem

Hilding Elmqvist

Institutionen för Reglerteknik
Tekniska Högskolan i Lund
Box 725
220 07 Lund

December 1980
Slutrapport

Språk för förverkligande av reglersystem

1. INTRODUKTION

Stora processreglersystem har tidigare ofta implementerats med hjälp av s.k. DDC-paket på minidatorer. Mikrodatortvecklingen har emellertid lett till utformning av decentraliserade system där varje mikrodata implementerar endast ett fåtal regulatorfunktioner.

För såväl DDC-paket som mikrodatabaserade regulatorer gäller att reglersystemet måste byggas upp av ett antal förprogrammerade standardregulatorer. Detta förorsakar problem vid design av komplicerade regulatorer t.ex. för starkt olinjära system eller för flervariabla reglerkretsar. Bristen av flexibilitet försvårar även införandet av nya reglermetoder t.ex. adaptiv reglering. För att implementera mer komplicerade regulatorer har man därför varit hänvisad till att använda Fortran tillsammans med reelltidsoperativsystem. Vidare har processoperatören dåliga verktyg för att inkorporera sina processkunskaper i reglersystemen.

Den dåliga flexibiliteten hos DDC-paket och mikrodatabaserade regulatorer var ett motiv för att studera möjligheterna och behoven av speciella användarorienterade språk för implementering av reglersystem.

Vid institutionen för Reglerteknik har forskningen tidigare haft tyngdpunkt på processidentifiering och analys och syntes av reglersystem. Det är därför en naturlig fortsättning att i större utsträckning även studera implementeringsaspekter.

Simuleringspråket SIMNON har utvecklats vid institutionen. Det används bl.a. för att analysera komplexa regleralgoritmer. Modellspråket i SIMNON är enkelt och kompilatorn finns inkluderad i programmet. Detta visade att ett enkelt språk var tillräckligt även för att beskriva komplexa regleralgoritmer. Detta är viktigt att observera vid val av nivå hos ett implementeringspråk. Restriktioner i språket kan t.ex. utnyttjas för att öka säkerheten mot programmeringsfel.

2. VERKTYG FÖR IMPLEMENTERING AV REGLERSYSTEM

Under förstudien har flera studiebesök gjorts på industrier och hos leverantörer som har datorsystem för reglering och styrning. Vidare har olika verktyg som finns tillgängliga för att implementera reglersystem studerats. En sammanfattning ges nedan.

Reelltids-Fortran

Många minidatorleverantörer tillhandahåller reelltids-operativsystem (DEC - RSX, Nord - SINTRAN etc.). Dessa tillåter att flera program, skrivna i FORTRAN, kan exekvera jämlöpande i reell tid. "European Workshop on Industrial Computer Systems" (EWICS) har arbetat med standardisering av hjälpmedlen i dylika system samt deras snittyta mot användarens program. Detta har resulterat i en ANSI-standard [1]. En presentation av de olika aktiviteterna inom EWICS finns i [6].

Reelltids-Basic

Reelltids-Basic är en utvidgning av vanlig Basic med jämlöpande aktiviteter, in- och utmatning mot processen etc. och tillhandahålles exempelvis av Foxboro och Metromation. "European Workshop on Industrial Computer Systems" har arbetat med standardisering av dylika tillägg. Reelltids-Basic lämpar sig bäst för implementering av relativt små reglersystem. En fördel är därvid den goda interaktiviteten under programutvecklingen. Reelltids-Basic lider dock av alla nackdelar som vanlig Basic har, bl.a. brist på hjälpmedel för att strukturera kod och data.

Reelltidsspråk

Ytterligare exempel på reelltidsspråk är Coral 66 och RTL/2 från England och PEARL från Tyskland. Speciella språk för programmering av jämlöpande aktiviteter har utvecklats bl.a. för operativsystemtillämpningar. Exempel är Concurrent Pascal, Modula och Modula-2. Portal har utvecklats vid Landis & Gyr för reelltidsspråk. Datorkonsortium 80 i Sverige har utvecklat Pascal D/80 som mest är tänkt för militära tillämpningar. Försvarsdepartementet (DOD) i USA initierade utveckling av ett språk för "embedded systems". Detta språk kallas Ada och kommer troligen ha stor betydelse för implementering av reelltidssystem.

DDC-paket

Reglersystem har ofta en regelbunden struktur med likartade reglerkretsar. DDC-paket (Direct Digital Control) har utvecklats för att utnyttja detta och möjliggöra enkla beskrivningar av reglersystem. DDC-paket tillhandahålles bl.a. från IBM, Siemens, Foxboro, Comator och Tour & Andersson. De kännetecknas av goda möjligheter för att ändra parametrar i reglerkretsarna. Det är emellertid mycket besvärligt att införa nya regleralgoritmer i DDC-paketen.

Mikroprocessorbaserade regulatorer

Decentraliserade reglersystem implementeras ofta med hjälp av förprogrammerade mikroprocessorer. Sådana system har tagits fram bl.a. av ASEA, Saab-Scania, Honeywell och Fisher

& Porter. Regleralgoritmer måste väljas ur en fix uppsättning varför det är svårt att beskriva komplicerade regulatorer. Man kan skilja mellan enkretsregulatorer och flerkretsregulatorer. De senare innehåller små enkla DDC-paket. En översikt över mikroprocessorbaserade regulatorer ges i [11].

PC-system

PC-system har utvecklats för att ersätta relälogik och utgöres av enkla enbitsprocessorer. Programmering görs antingen med enkla assemblerliknande språk eller grafiskt, s.k. "ladder diagram". Det finns många tillverkare t.ex. Satt-Electronics, Asea och Alan Bradley. En översikt över PC-system ges i [10].

Mikrodatorsystem

Till mikrodatorsystem finns ofta reelltidshjälpmedel. Texas Instruments har t.ex. "Microprocessor Pascal" (MPP) till mikroprocessorn Texas 9900. MPP har processbegreppet, semaforer etc. Intel tillhandahåller reelltidsoperativsystemen RMX/80 och RMX/86. Motorola har Versados till MC 68000 vilket kan användas för reelltidstillämpningar. I Sverige har Saab-Scania utvecklat RTX86 för Intel 8086 samt Dataindustrier (DIAB) OS.8 för Zilog Z80 och Z8000.

Slutsatsen av studien är att det för närvarande knappast finns något idealiskt verktyg för att implementera reglersystem med de krav på säkerhet, flexibilitet och användarvänlighet som man kan ställa i dag.

3. PILOTPROJEKT

För att få en känsla för vad som skulle kunna göras och hur svårt det är att implementera olika faciliteter har några pilotprojekt utförts i samband med förstudien. Dessa beskrivs nedan.

REGULA - Ett användarvänligt programpaket för implementering av reglersystem

Ett enkelt språk för beskrivning av reglersystem har tagits fram. Vid utformningen har gjorts ett försök att på ett enhetligt sätt ta med notationer från olika beskrivningssätt. Det är möjligt att definiera generiska beskrivningar av t.ex. PID-regulatorer, självinställande regulatorer, förreglingsprogram och sekvensstyrningsprogram. Flera kopior kan sedan genereras och sammankopplas.

Stor vikt har lagts vid interaktivitet. Det är således möjligt att modifiera en regleralgoritm medan den exekveras.

Andringen verkställs i tiden mellan två exekverings-tillfällen. Editering sker med hjälp av bildskärm.

Implementeringen har gjorts i Pascal och Concurrent Pascal. En prototyp togs fram som ett projektarbete i en doktorandkurs "Moderna språk för implementering av reglersystem" vid institutionen för Reglerteknik (se [7]). Denna implementering har sedan vidareutvecklats.

I den ovan nämnda kursen gjordes ytterligare två projektarbeten. Det andra projektet bestod i att konstruera ett distribuerat DDC-paket (se [2]). Fyra LSI-11 datorer användes. Program som gör det möjligt att sända mätdata och kommandon mellan datorerna utvecklades.

Ett enkelt reelltidsoperativsystem

Det tredje kursprojektet bestod i att konstruera ett enkelt operativsystem s.k. kärna, för att göra det möjligt att använda jämlöpande aktiviteter (concurrent processes) i Pascal för LSI-11 datorer (se [5]). Detta hjälpmedel möjliggör att ett antal Pascal procedurer exekveras som jämlöpande aktiviteter. Kärnan innehåller hantering av semaforer för handhavande av gemensamma resurser etc. Kärnan skrevs till största delen i Pascal.

Detta arbete har sedan vidareutvecklats. Koden har omorganiserats så att den innersta delen består av fyra assemblerskrivna procedurer:

- initnucleus - initialisering
- newprocess - skapande av en ny process
- resume - återuppta exekveringen av en annan process
- ioresume - återuppta exekveringen av en annan process under väntan på en viss avbrottssignal

Implementeringen av dessa primitiver utgör ca fyra sidor assembler [3].

Med hjälp av dessa primitiver har algoritmer för "scheduling", semaforhantering, tidsköhantering etc. implementerats i Pascal [4].

Arbetet med detta reelltidsoperativsystem har gett goda kunskaper i programmering av jämlöpande aktiviteter. Systemets flexibilitet ger dessutom möjligheter till att testa nya primitiver såsom "rendezvous"-begreppet i Ada. Vidare är systemet mycket lämpligt för undervisning eftersom en vanlig Pascal-kompilator används som grund. Man kan därför enkelt visa exakt vilken kod som krävs för att åstadkomma flera processer, för att utföra "context switch" etc.

Pascal-kompilatorn för LSI-11 (OMSI-Pascal) som har använts

distribueras i Sverige av Sern Datakonsult AB i Göteborg. Diskussioner har förts med dem om distribution av reelltidsoperativsystemet.

För att få en återkoppling från industrins krav har kärnan implementerats på Texas 9900 i samarbete med Telemetric AB, Arlov. Inom ramen för detta samarbete kommer kärnan också att jämföras med Texas MPP och med ett speciellt operativsystem som används av Telemetric AB. Implementering av systemet på MC 68000 har diskuterats med Motorola, Stockholm. Preliminära tester har utförts.

Pascal för PC-system

PC-system programmeras för närvarande ofta i enkla assemblerliknande språk med instruktioner såsom AND, OR, NOT, SET samt timers och räknare. Ett PC-program kan uppgå till mer än 10000 rader varför struktureringsfaciliteter behövs. Sådana saknas för närvarande. Ett förslag till högnivåspråk för PC-system presenteras i [12].

En grundläggande slutsats inom projektet var att mera enhetlighet i beskrivningen borde eftersträvas. En naturlig ide var därför att möjliggöra att använda Pascal för programmering av PC-system. På så sätt skulle struktureringsfaciliteter fås automatiskt. Naturligtvis måste begränsningar i användningen av Pascal införas. Den enda primära datatyp som får användas är "boolean".

Heltal kan emellertid användas för beräkningar som kan utföras vid kompileringstillfället. På så sätt kan t.ex. "for"-slingor användas för generering av likartad kod för flera anläggningsdelar.

I samarbete med Satt-Electronics, Malmö har ett examensarbete utförts [9]. En Pascal-kompilator som genererar P-kod fanns tillgänglig. I arbetet ingick att från P-kod generera effektiv kod för PC-systemet PBS-mini från Satt-Electronics. Resultaten visar att dokumentationen av systemet blir avsevärt bättre med användande av Pascal samtidigt som nästan lika effektiv kod kan genereras. Dessutom ges bättre möjligheter till återanvändning av delar av program.

Terminal med zoomning

Utformningen av den interaktiva programutvecklingsmiljön är viktig. För närvarande används oftast en vanlig alfanumerisk terminal med bildskärm för 24 rader. I vissa fall finns en bildskärmsorienterad editor med vars hjälp en del av en textfil (24 rader) avbildas på skärmen och fås tillgänglig för modifiering.

En nackdel med denna typ av skärm är att man inte kan få någon överblick över innehållet i filen. Man skulle ibland vilja kunna betrakta innehållet "på avstånd" för att se t.ex. indragningar och var procedurer börjar etc.

Stor datorkapacitet krävs för bildgenerering eftersom det är önskvärt att zoomningen görs någorlunda kontinuerligt. Ett examensarbete [8] formulerades därför tillsammans med avdelningen för Digital- och Datorteknik för utveckling av en specialprocessor för generering av videosignaler. Denna processor samarbetar med en mikroprocessor via ett textminne. Resultaten hittills verkar lovande.

4. ANVANDARORIENTERADE SPRÅK OCH HJÄLPMEDEL FÖR IMPLEMENTERING AV REGLERSYSTEM

Kartläggningen av befintliga beskrivningsmetoder samt pilotprojekten visade att det fanns behov för, och möjligheter till, framtagande av användarorienterade språk för implementering av reglersystem. Ett projekt för fortsatt arbete formulerades i februari 1980 (STU-ansökan 80-3705): "Development of user oriented languages and software tools for implementation of control systems".

Huvudmålet med detta projekt är att ta fram programmeringsspråk med begrepp och notationer som är speciellt anpassade för implementering av datoriserade regler- och styrsystem. Genom att göra en lämplig avgränsning av problemklassen samt utnyttja de teoretiska begreppen och notationerna bör det vara möjligt att utforma ett enhetligt, kraftfullt och användarvänligt språk.

Ett exempel på problem som kommer att studeras är blandning av reglering (PID-regulatorer, adaptiva regulatorer, etc.) och styrning (förreglingar, sekvenser). Inom dessa områden används f.n. helt olika beskrivningssätt och att åstadkomma samverkan är ofta svårt.

Reglersystem blir alltmer decentraliserade med användande av mikroprocessorbaserade, förprogrammerade regulatorer. Därvid framkommer speciella problem vid beskrivning och dokumentation samt vad gäller kommunikation mellan olika moduler. Dokumentationsaspekter kommer speciellt att behandlas.

Andra viktiga problemområden är hjälpmedel för framtagande av systembeskrivningar samt operatörkommunikation. Det bör finnas kraftfulla hjälpmedel tillgängliga vid igångkörning och modifieringar som gör det möjligt att ändra systemet på ett enkelt sätt utan att störa processen. I decentraliserade system uppkommer därvid speciella problem vad gäller distribution av nya beskrivningar.

Det finns vidare planer på att utforma verktyg som ger

möjligheter till utförlig testning av regler- och styrsystemet mot en simulerad verklighet. Därvid kan systemets dynamiska egenskaper analyseras på ett säkert och billigt sätt. Detta är speciellt viktigt vad gäller onormala tillstånd hos processen som regler- och styrsystemet måste behandla på speciellt sätt.

5. CENTRUM FÖR INDUSTRIELLA DATORSYSTEM

I samband med diskussioner kring STU:s ramprogram informationsbehandling framkom idén till ett samarbete med Institutionen för Elautomatik (numera avdelningen för Digital- och Datorteknik) (STU-ansökan 80-3962). Ett sådant samarbete är naturligt bl.a. därför att mikroprocessorer blir allt viktigare verktyg för realisering av reglersystem.

Centret kommer att bedriva forskning och utveckling inom exempelvis områdena: datorarkitektur, tillämpningsanpassad hårdvara, real-tidsprogrammering, utformning av programmeringsspråk, programutvecklingshjälpmedel och operatörskommunikation. Inom centrets ram finns för närvarande tre projekt: "Multi-mikroprocessor-datorer", "Användarorienterade språk och hjälpmedel för förverkligande av reglersystem" samt "Konstruktion och tillämpning av associativ dator". Tanken är vidare att centret skall utgöra en bas för nya forskningsprojekt i samarbete med industrin.

6. SAMMANFATTNING

Det finns en klar trend att decentralisera reglerfunktionerna i större processreglersystem. Därvid används vanligen förprogrammerade mikrodatorbaserade regulatorer. Dessa innehåller ofta små DDC-paket med blockorienterad beskrivning. För förregling och sekvensstyrning används PC-system.

Minidatorer används i sådana system för att upprätthålla en databas för anläggningen samt för processpresentation. Färgbildskärmar används ofta för presentation av processschema som dynamiskt uppdateras samt för trendkurvor. I minidatorn görs även vissa avancerade beräkningar och optimeringar.

Ett växande användningsområde för mikroprocessorer är reglering och styrning av diverse apparater som eventuellt massproduceras. Mikrodatorn ingår därvid som en komponent i apparaten. Vad mikrodatorn skall utföra varierar mycket mellan olika applikationer och begreppet reglerkretsar är t.ex. inte så centralt som för processreglersystem. Av det skälet behövs reelltidsspråk med god flexibilitet. Språket Ada har utvecklats just för denna klass av applikationer ("embedded systems").

Det kan noteras att nivån på de beskrivningsmetoder som används i samband med processreglersystem är relativt låg. Det gäller t.ex. PC-system där beskrivningen utgöres antingen av en sekvens av elementära operationer eller av "ladder diagrams". För mikroprocessorbaserade regulatorer görs beskrivningen i form av koppling av "block" av typ PID-regulatorer, adderare, funktionsgeneratorer etc. Att t.ex. inkludera en speciell ekvation i en reglerkrets görs i sådana fall genom att koppla samman block. Detta leder till dåligt dokumenterade system.

En anledning till den låga nivån på beskrivningsmetoderna anges av leverantörer vara att "vem som helst" skall kunna göra en beskrivning. En annan anledning är att man vill ha programmeringshjälpmedel antingen i produkten själv eller i en liten mikrodatorbestyckad programmeringsenhet. Detta har lett till att man har varit tvingad att hålla nere komplexiteten. Denna begränsning håller på att försvinna i och med tillgängligheten av moderna mikroprocessorerna med större adresseringsområde. Vidare har förmodligen kunskapen om datalogi, t.ex. kompilator teknik, varit för låg hos många av de som utvecklat dessa system.

Man kan säga att det finns inget bra hjälpmedel för de lite mer kvalificerade användarna som arbetar med processer som kräver avancerade reglersystem.

När man diskuterar blockorienterad och algoritmorienterad beskrivning är det intressant att jämföra med utvecklingen av språk för simulering av dynamiska system i slutet av 60-talet. För att göra det enkelt för de som tidigare hade använt analogmaskin för simulering att övergå till användande av dator utvecklades det program som accepterade en dynamisk modell i form av integratorer, summatorer, potentiometrar samt vissa olinjära element. Numera används så gott som uteslutande ekvationsorienterade simuleringsprogram. Dessa har inbyggda integrationsalgoritmer samt hjälpmedel för resultatpresentation etc.

För beskrivning av reglersystem finns i princip två nivåer: den blockorienterade i DDC-paket och mikrodatorbaserade regulatorer samt den algoritmiska i reelltidsprogrammeringsspråk. Det finns även behov av en mellanliggande nivå där det finns tillräcklig flexibilitet för att införa t.ex. olinjäriteter i regulatorer, nya regleralgoritmer och olika reglermoder. Samtidigt skall det finnas tillgängligt moduler som kan hantera standardfallen.

Reelltidsspråk av typ Ada är onödigt flexibla och komplicerade för att passa för processreglersystem. Flexibiliteten medför att användaren har mycket litet stöd för hur problemet bör struktureras. Detta medför att en stor del av beskrivningen är irrelevant för problemet samt att kompilatorn inte kan utföra så omfattande kontroller. Vidare gör komplexiteten hos språket det svårt att göra en

kompiator som klarar ändringar i programmet "on-line".

Som angavs i avsnitt fyra kommer dessa problem att vidare studeras i projektet "Användarorienterade språk och hjälpmedel för implementering av reglersystem".

7. REFERENSER

- [1] ANSI/ISA-861.1: Industrial Computer System Fortran Procedures for Executive Functions, Process Input/Output, and Bit Manipulation. Standard, Instrument Society of America.
- [2] Breidegaard B. et.al.: A Distributed DDC-package. Report, Dept. of Automatic Control, Lund Inst. of Tech.
- [3] Elmqvist H., Mattsson S.E.: Implementation of Basic Primitives for Concurrent Programming in Pascal, (to be published).
- [4] Elmqvist H., Mattsson S.E.: A Kernel which facilitates Concurrent Programming in Pascal, (to be published).
- [5] Essebo T., et.al.: A Facility for Executing Concurrent Processes in Pascal, Report TFRT-7194, Dept. of Automatic Control, Lund Inst. of Technology.
- [6] European Workshop on Industrial Computer systems: Newsletter, Issue no 1-79, March 1979
- [7] Kruzela I. et.al.: REGULA - Ett användarvänligt programpaket för implementering av reglersystem, Rapport, Inst. för Reglerteknik, LTH.
- [8] Lindvall B., Lynner B.: Alfnumerisk bildskärm med zoom. Examensarbete, Avd. för Digital- och Datorteknik.
- [9] Minör B., Permval O.: Pascal för programmering av PC-system. Examensarbete, Inst. för Reglerteknik.
- [10] Normak P.: Programmerbara styrsystem, en översikt. Elteknik med aktuell elektronik, nr 15, 1979.
- [11] Normak P.: Mikroprocessorn i reglersystemet. Elteknik med aktuell elektronik, nr 16, 1980.
- [12] Takahashi H.: An Automatic Controller Description Language. IEEE Transactions on Software Engineering, vol SE-6, No 1, January 1980